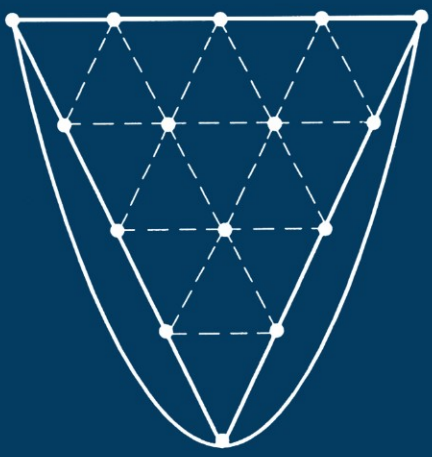
SOLID MECHANICS AND ITS APPLICATIONS

Raphael T. Haftka and Zafer Gürdal

Elements of Structural Optimization

Third Revised and Expanded Edition



SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

ELEMENTS OF STRUCTURAL OPTIMIZATION

SOLID MECHANICS AND ITS APPLICATIONS
Volume 11

Series Editor: G.M.L. GLADWELL
Solid Mechanics Division, Faculty of Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids.

The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

The median level of presentation is the first year graduate student. Some texts are monographs defining the current state of the field; others are accessible to final year undergraduates; but essentially the emphasis is on readability and clarity.

For a list of related mechanics titles, see final pages.

Elements of Structural Optimization

Third revised and expanded edition

by

RAPHAEL T. HAFTKA

Department of Aerospace and Ocean Engineering,
Virginia Polytechnic Institute and State University,
Blacksburg, Virginia, U.S.A.

and

ZAFER GÜRDAL

Department of Engineering Science and Mechanics,
Virginia Polytechnic Institute and State University,
Blacksburg, Virginia, U.S.A.

SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

Library of Congress Cataloging-in-Publication Data

Haftka, Raphael T.
Elements of structural optimization / by Raphael T. Haftka and Zafer Gürdal. -- 3rd rev. and expanded ed.
p. cm. -- (Solid mechanics and its applications ; v. 11)
Includes bibliographical references and indexes.
ISBN 978-0-7923-1505-6 ISBN 978-94-011-2550-5 (eBook)
DOI 10.1007/978-94-011-2550-5
1. Structural optimization. I. Gürdal, Zafer. II. Title.
III. Series.
TA658.8.H34 1991
624.1'7--dc20 91-37690
CIP

ISBN 978-0-7923-1505-6

Printed on acid-free paper

This book is dedicated to

Rose

Pınar and Erin

Contents

Preface xiii

Chapter 1. Introduction 1
1.1 Function Optimization and Parameter Optimization 1
1.2 Elements of Problem Formulation 3
Design Variables 3
Objective Function 5
Constraints 9
Standard Formulation 9
1.3 The Solution Process 12
1.4 Analysis and Design Formulations 14
1.5 Specific Versus General Methods 15
1.6 Exercises 16
1.7 References 19

Chapter 2. Classical Tools in Structural Optimization 23
2.1 Optimization Using Differential Calculus 23
2.2 Optimization Using Variational Calculus 29
Introduction to the Calculus of Variations 29
2.3 Classical Methods for Constrained Problems 33
Method of Lagrange Multipliers 34
Function Subjected to an Integral Constraint 37
Finite Subsidiary Conditions 40
2.4 Local Constraints and the Minmax Approach 44
2.5 Necessary and Sufficient Conditions for Optimality 49
Elastic Structures of Maximum Stiffness 50

Optimal Design of Euler-Bernoulli Columns 52
Optimum Vibrating Euler-Bernoulli Beams 57
2.6 Use of Series Solutions in Structural Optimization 61
2.7 Exercises 64
2.8 References 66

Chapter 3. Linear Programming 71

3.1 Limit Analysis and Design of Structures Formulated as LP Problems 72
3.2 Prestressed Concrete Design by Linear Programming 81
3.3 Minimum Weight Design of Statically Determinate Trusses 83
3.4 Graphical Solutions of Simple LP Problems 86
3.5 A Linear Program in a Standard Form 88
Basic Solution 89
3.6 The Simplex Method 90
Changing the Basis 91
Improving the Objective Function 93
Generating a Basic Feasible Solution—Use of Artificial Variables 94
3.7 Duality in Linear Programming 96
3.8 An Interior Method—Karmarkar's Algorithm 100
Direction of Move 101
Transformation of Coordinates 103
Move Distance 104
3.9 Integer Linear Programming 104
Branch-and-Bound Algorithm 107
3.10 Exercises 110
3.11 References 113

Chapter 4. Unconstrained Optimization 115

4.1 Minimization of Functions of One Variable 115
Zeroth Order Methods 116
First Order Methods 121
Second Order Method 122
Safeguarded Polynomial Interpolation 123
4.2 Minimization of Functions of Several Variables 123
Zeroth Order Methods 123
First Order Methods 132
Second Order Methods 137
Applications to Analysis 142
4.3 Specialized Quasi-Newton Methods 143
Exploiting Sparsity 143
Coercion of Hessians for Suitability with Quasi-Newton Methods 144
Making Quasi-Newton Methods Globally Convergent 145
4.4 Probabilistic Search Algorithms 145
Simulated Annealing 146
Genetic Algorithms 149
4.5 Exercises 152
4.6 References 154

Chapter 5. Constrained Optimization 159
5.1 The Kuhn-Tucker Conditions 161
General Case 161
Convex Problems 166
5.2 Quadratic Programming Problems 169
5.3 Computing the Lagrange Multipliers 170
5.4 Sensitivity of Optimum Solution to Problem Parameters 173
5.5 Gradient Projection and Reduced Gradient Methods 176
5.6 The Feasible Directions Method 182
5.7 Penalty Function Methods 186
Exterior Penalty Function 187
Interior and Extended Interior Penalty Functions 190
Unconstrained Minimization with Penalty Functions 193
Integer Programming with Penalty Functions 195
5.8 Multiplier Methods 198
5.9 Projected Lagrangian Methods (Sequential Quadratic Prog.) 201
5.10 Exercises 205
5.11 References 206

Chapter 6. Aspects of the Optimization Process in Practice 209
6.1 Generic Approximations 211
Local Approximations 211
Global and Midrange Approximations 219
6.2 Fast Reanalysis Techniques 222
Linear Static Response 222
Eigenvalue Problems 226
6.3 Sequential Linear Programming 228
6.4 Sequential Nonlinear Approximate Optimization 236
6.5 Special Problems Associated with Shape Optimization 239
6.6 Optimization Packages 242
6.7 Test Problems 244
Ten-Bar Truss 244
Twenty-Five-Bar Truss 245
Seventy-Two-Bar Truss 246
6.8 Exercises 248
6.9 References 249

Chapter 7. Sensitivity of Discrete Systems 255
7.1 Finite Difference Approximations 256
Accuracy and Step Size Selection 256
Iterative Methods 259
Effect of Derivative Magnitude on Accuracy 261
7.2 Sensitivity Derivatives of Static Displacement and Stress Constraints 263
Analytical First Derivatives 263
Second Derivatives 268
The Semi-Analytical Method 269
Nonlinear Analysis 273

Sensitivity of Limit Loads 274
7.3 Sensitivity Calculations for Eigenvalue Problems 276
Sensitivity Derivatives of Vibration and Buckling Constraints ... 276
Sensitivity Derivatives for Non-Hermitian Eigenvalue Problems . 283
Sensitivity Derivatives for Nonlinear Eigenvalue Problems 290
7.4 Sensitivity of Constraints on Transient Response 291
Equivalent Constraints 291
Derivatives of Constraints 293
Linear Structural Dynamics 298
7.5 Exercises 301
7.6 References 302

Chapter 8. Introduction to Variational Sensitivity Analysis 305
8.1 Linear Static Analysis 306
The Direct Method 308
The Adjoint Method 312
Implementation Notes 317
8.2 Nonlinear Static Analysis and Limit Loads 318
Static Analysis 318
Limit Loads 323
Implementation Notes 327
8.3 Vibration and Buckling 327
The Direct Method 328
The Adjoint Method 331
8.4 Static Shape Sensitivity 334
The Material Derivative 334
Domain Parametrization 337
The Direct Method 339
The Adjoint Method 343
8.5 Exercise 345
8.6 References 345

Chapter 9. Dual and Optimality Criteria Methods 347
9.1 Intuitive Optimality Criteria Methods 348
Fully Stressed Design 348
Other Intuitive Methods 353
9.2 Dual Methods 353
General Formulation 354
Application to Separable Problems 355
Discrete Design Variables 357
Application with First Order Approximations 361
9.3 Optimality Criteria Methods for a Single Constraint 365
The Reciprocal Approximation for a Displacement Constraint ... 366
A Single Displacement Constraint 368
Generalization for Other Constraints 370
Scaling-based Resizing 372
9.4 Several Constraints 375

Reciprocal-Approximation Based Approach 375
Scaling-based Approach 380
Other Formulations 382
9.5 Exercises 383
9.6 References 384

Chapter 10. Decomposition and Multilevel Optimization 387
10.1 The Relation between Decomposition and Multilevel Formulation 387
10.2 Decomposition 388
10.3 Coordination and Multilevel Optimization 399
10.4 Penalty and Envelope Function Approaches 401
10.5 Narrow-Tree Multilevel Problems 404
Simultaneous Analysis and Design 404
Other Applications 406
10.6 Decomposition in Response and Sensitivity Calculations 406
10.7 Exercises 412
10.8 References 412

Chapter 11. Optimum Design of Laminated Composite Materials 415
11.1 Mechanical Response of a Laminate 415
Orthotropic Lamina 416
Classical Laminated Plate Theory 418
Bending, Extension, and Shear Coupling 420
11.2 Laminate Design 422
Design of Laminates for In-plane Response 422
Design of Laminates for Flexural Response 430
11.3 Stacking Sequence Design 438
Graphical Stacking Sequence Design 438
Penalty Function Formulation 440
Integer Linear Programming Formulation 442
Probabilistic Search Methods 450
11.4 Design Applications 451
Stiffened Plate Design 451
Aeroelastic Tailoring 459
11.5 Design Uncertainties 460
11.6 Exercises 462
11.7 References 464

Name Index 469

Subject Index 475

Preface

The field of structural optimization is still a relatively new field undergoing rapid changes in methods and focus. Until recently there was a severe imbalance between the enormous amount of literature on the subject, and the paucity of applications to practical design problems. This imbalance is being gradually redressed. There is still no shortage of new publications, but there are also exciting applications of the methods of structural optimizations in the automotive, aerospace, civil engineering, machine design and other engineering fields. As a result of the growing pace of applications, research into structural optimization methods is increasingly driven by real-life problems.

Most engineers who design structures employ complex general-purpose software packages for structural analysis. Often they do not have any access to the source program, and even more frequently they have only scant knowledge of the details of the structural analysis algorithms used in this software packages. Therefore the major challenge faced by researchers in structural optimization is to develop methods that are suitable for use with such software packages. Another major challenge is the high computational cost associated with the analysis of many complex real-life problems. In many cases the engineer who has the task of designing a structure cannot afford to analyze it more than a handful of times.

This environment motivates a focus on optimization techniques that call for minimal interference with the structural analysis package, and require only a small number of structural analysis runs. A class of techniques of this type, pioneered by Lucien

Schmit, and which are becoming widely used, are referred to in this book as sequential approximate optimization techniques. These techniques use the analysis package for the purpose of constructing an approximation to the structural design problem, and then employ various mathematical optimization techniques to solve the approximate problem. The optimum of the approximate problem is then used as a basis for performing one or more structural analyses for the purpose of updating or refining the approximate design problem. Most of the approximate design problems are based on derivatives of the structural response with respect to design parameters.

In the new environment the structural designer is typically called upon to provide the interface between a commercially available analysis program, and a commercially available optimization software package. The three most important ingredients of the interface are: sensitivity derivative calculation, construction of an approximate problem, and evaluation of results for the purpose of fine-tuning the approximate problem or the optimization method for maximum efficiency and reliability.

This textbook is organized so that its middle part—Chapters 6, 7 and 8 deal with the two issues of constructing the approximate problem and obtaining sensitivity derivatives. Evaluating the results of the optimization calls for a basic understanding of optimality conditions and optimization methods. This is dealt with in Chapters 1 through 5. The last three chapters deal with the specialized topics of optimality criteria methods, multi-level optimization, and applications to composite materials.

The material in the textbook can be used in various ways in teaching a graduate course in structural optimization, depending on the available amount of time, and whether students have prior preparation in optimization techniques.

Without prior preparation in optimization techniques it is suggested that the minimum time requirement is one semester. It is suggested to cover Chapter 1, sections 2.1, 2.2 and 2.3 of Chapter 2, Sections 3.1 and 3.4 of Chapter 3, some material from Chapters 4 and 5 depending on the instructor's favorite optimization methods, most of Chapter 6 and the first two sections of Chapter 7. With a two-quarter sequence it is suggested to cover Chapters 1 and 2, selected topics of Chapters 3 to 5 and Chapter 6 in the first quarter, and Chapters 7, 9, 11 and either Chapter 8 or Chapter 10 in the second quarter. Finally, in a two-semester sequence it is recommended to cover Chapters 1 through 6 in the first semester, and Chapters 7 through 11 in the second semester.

With a preparatory course in mathematical optimization a one quarter and a one semester versions of the course can be considered. A one-quarter version could include Chapters 1 and 2, sections 3.1, 3.2, 3.3 and 3.7 of Chapter 3, and Chapters 6, the first two sections of Chapter 7, and Chapter 9 or 11.. A one-semester version could include the same part of Chapters 1 through 7 and then Chapters 9 through 11.

The authors gratefully acknowledge the assistance of Drs. H. Adelman, B. Barthelemy, J-F. Barthelemy, L. Berke, R. Grandhi, D. Grierson, E. Haug, R. Plaut, J. Sobieski, and J. Starnes in reviewing parts of the manuscript and offering critical comments.

Introduction 1

Optimization is concerned with achieving the best outcome of a given operation while satisfying certain restrictions. Human beings, guided and influenced by their natural surroundings, almost instinctively perform all functions in a manner that economizes energy or minimizes discomfort and pain. The motivation is to exploit the available limited resources in a manner that maximizes output or profit. The early inventions of the lever or the pulley mechanisms are clear manifestations of man's desire to maximize mechanical efficiency. Innumerable other such examples abound in the saga of human history. Douglas Wilde [1] provides an interesting account of the origin of the word optimum and the definition of an optimal design. We will paraphrase Wilde and offer the definition of an optimal design as being 'the best feasible design according to a preselected quantitative measure of effectiveness'.

As it is beyond the scope of this text to trace the historical development of optimization, we list a few of the more recent references on the subject of structural optimization. These references [2–19] trace the development of the field of structural optimization dating back to the eighteenth century. The importance of minimum weight design of structures was first recognized by the aerospace industry where aircraft structural designs are often controlled more by weight than by cost considerations. In other industries dealing with civil, mechanical and automotive engineering systems, cost may be the primary consideration although the weight of the system does affect its cost and performance. A growing realization of the scarcity of raw materials and a rapid depletion of our conventional energy sources is being translated into a demand for lightweight, efficient and low cost structures. This demand in turn emphasizes the need for engineers to be cognizant of techniques for weight and cost optimization of structures. The objective of this text is to acquaint students and practicing engineers with these techniques.

1.1 Function Optimization and Parameter Optimization

Before the advent of high speed computation most of the solutions of structural analysis problems were based on formulations employing differential equations. These

differential equations were solved analytically (e.g., by using infinite series) with occasional use of numerical methods at the very end of the solution process. The unknowns were functions (representing displacements, stresses, etc.) defined over a continuum.

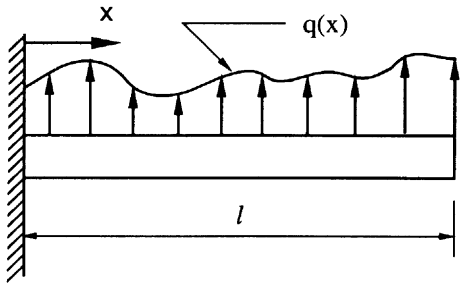


Figure 1.1.1 Beam example.

The early beginning of structural optimization followed the same route, in that the unknowns were functions defining the optimal structural properties. Consider, for example, the beam shown in Figure 1.1.1. Structural analysis is concerned with finding the displacement $w(x)$ of the beam by solving the well-known governing equation

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right) = q(x). \tag{1.1.1}$$

The structural designer may want to find the optimum distribution of the moment of inertia $I(x)$ of the beam along its length. Of course, the notion of optimality requires that we have an objective function that we wish to maximize or to minimize. For example, the objective function may be the mass of the beam. For many common beam cross sections the mass m is given as

$$m = c\int_0^l I^p(x)dx, \tag{1.1.2}$$

where the exponent p is usually between 0.4 and 0.5, and c is a known constant. An optimization problem typically involves a number of constraints. Without any constraint the optimum beam would have zero moment of inertia and zero mass. In the design of a beam, a typical constraint would be to limit the maximum displacement of the beam to some specified allowable w_0,

$$w_{max} = \max_{0\le x\le l} w(x) \le w_0. \tag{1.1.3}$$

It is possible to obtain the necessary conditions for optimality in the form of a differential equation in $I(x)$ and $w(x)$. The mathematical discipline that deals with this type of problem is called the *calculus of variations*, and is briefly discussed

in Chapter 2. The class of structural optimization problems that seeks an optimum structural *function* is called *function or distributed parameter* structural optimization.

In the late fifties and early sixties high speed electronic computers had a profound effect on structural analysis solution procedures. Techniques that were well suited to computer implementation, in particular the finite element method (FEM), became dominant. The finite element method discretizes the structure at the very beginning of the analysis, so that the unknowns in the analysis are discrete values of displacements and stresses at nodes of the finite element model, rather than functions. The differential equations solved by earlier analysts are replaced by systems of algebraic equations for the variables that describe the discretized system.

The same transformation began to take hold in the early sixties in the field of structural optimization. When optimizing a structure discretized by finite elements it is natural to discretize the structural properties which are optimized. Consider again the beam example of Figure 1.1.1. A finite element solution for the displacements starts by dividing the beam into a number of constant-property segments or finite elements. An optimization of the same beam would naturally use the moments of inertia of the segments as design parameters. Thus, instead of searching for an optimum function, we will be looking for the optimum values of a number of parameters. The mathematical discipline that deals with *parameter optimization* is called *mathematical programming.* The bulk of this text (Chapters 3-7, 9-11) is concerned, therefore, with mathematical programming techniques and their application to structural optimization problems defined by discretized models. In particular, it is often implicitly assumed that the structural analysis is based on the finite element method.

1.2 Elements of Problem Formulation

1.2.1 *Design Variables*

The notion of improving or optimizing a structure implicitly presupposes some freedom to change the structure. The potential for change is typically expressed in terms of ranges of permissible changes of a group of parameters. Such parameters are usually called *design variables* in structural optimization terminology and denoted by a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in this book. Design variables can be cross-sectional dimensions or member sizes, they can be parameters controlling the geometry of the structure, its material properties, etc. Design variables may take *continuous* or *discrete* values. Continuous design variables have a range of variation, and can take any value in that range. For example, in the design problem of Figure 1.1.1 the moment of inertia of any segment of the beam may be considered a continuous design variable. Discrete design variables can take only isolated values, typically from a list of permissible values. Material design variables are often discrete. If we consider five materials in the design of the beam, then we can define a design variable that can take any integer value from one to five to represent the material choice. Design variables that

are commonly treated as continuous are often made discrete due to manufacturing considerations. For example, if the beam of Figure 1.1.1 is designed to minimize cost, then we may need to limit ourselves to commercially available cross sections. The moment of inertia would then cease to be a continuous design variable, and would become a discrete one.

In most structural design problems we tend to disregard the discrete nature of the design variables in the solution of the optimization problem. Once the optimum design is obtained, we then adjust the values of the design variables to the nearest available discrete value. This approach is taken because solving an optimization problem with discrete design variables is usually much more difficult than solving a similar problem with continuous design variables. However, rounding off the design to the closest integer solution works well when the available values of the design variables are spaced reasonably close to one another, so that changing the value of a design variable to the nearest integer does not change the response of the structure substantially. In some cases the discrete values of the design variables are spaced too far apart, and we have to solve the problem with discrete variables. This is done by employing a branch of mathematical programming called *integer programming.* In this text it is assumed that design variables are continuous unless otherwise stated.

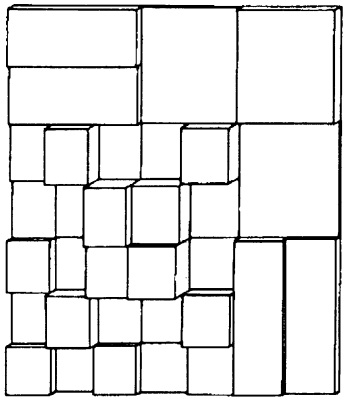
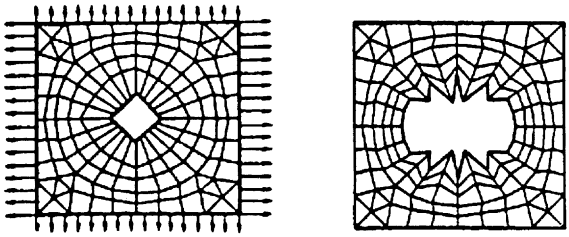


Figure 1.2.1 Optimal thickness distribution of a plate.

The choice of design variables can be critical to the success of the optimization process. In particular it is important to make sure that the choice of design variables is consistent with the analysis model. Consider, for example, the process of discretizing a structure by a finite element model and applying the optimization procedure to the model. If the design variable distribution has a one-to-one correspondence with the finite element model we can encounter serious accuracy problems. For example, the plate shown in Figure 1.2.1 was analyzed [20] by a 7×7 finite element mesh, with most design variables specifying the thickness of individual elements. While the 7×7

model was adequate for the initial design which had uniform thickness, it was not adequate for the final design shown in the Figure.



(a) *(b)*

Figure 1.2.2 Optimized shape of a hole in a plate, (a) initial design, (b) final design.

A similar problem may be encountered when the coordinates of nodes of the finite element model are used as design variables. For example, the shape of the hole in the plate shown in Figure 1.2.2 was optimized [21] to reduce the stress concentration near the hole with the coordinates of the boundary nodes serving as design parameters. Again, the finite element model was adequate for the analysis of the initial circular shape of the hole, but not the "optimal shape" obtained. In general, the distribution of design variables should be much coarser than the distribution of finite elements (except for skeletal structures where often each element corresponds to a physical member of the structure)

1.2.2 *Objective Function*

The notion of optimization also implies that there are some merit function $f(\mathbf{x})$ or functions $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_p(\mathbf{x})]$ that can be improved and can be used as a measure of effectiveness of the design. The common terminology for such functions is *objective functions.* Optimization with more than one objective is generally referred to as *Multicriteria Optimization.* For structural optimization problems, weight, displacements, stresses, vibration frequencies, buckling loads, and cost or any combination of these can be used as objective functions. Consider, for example, the three-bar truss of Figure 1.2.3. Our design problem may be to vary the horizontal locations of the three support points so as to minimize the mass of the truss and the stresses in its members. We have four objective functions: the mass and the three stresses.

Dealing with multiple objective functions is complicated and is usually avoided. There are two intuitive ways commonly used for reducing the number of objective functions to one. The first way is to generate a composite objective function that replaces all the objectives. For example, if the mass of the structure is denoted m and

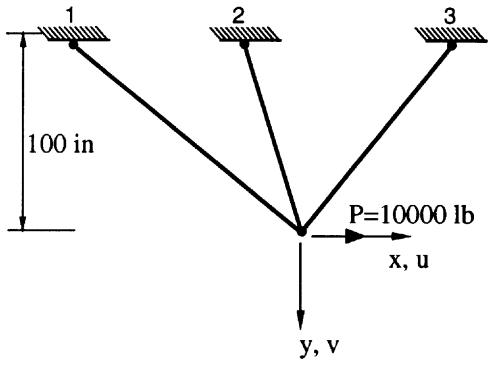


Figure 1.2.3 Three-bar truss example.

the stresses in the three bars as σ_i, $i = 1, 2, 3$, then a composite objective function f could be

$$F = \alpha_0 m + \alpha_1 \sigma_1 + \alpha_2 \sigma_2 + \alpha_3 \sigma_3, \tag{1.2.1}$$

where the α_i are weighting coefficients selected to reflect the relative importance of the four objective functions.

The second intuitive way to reduce the number of objective functions is to select the most important as the only objective function and to impose limits on the others. Thus we can formulate the three-bar truss design problem as minimization of mass, subject to upper limits on the values of the three stresses.

When it is not intuitively clear how to weight or choose between the objective functions, a systematic approach to the problem is through a branch of mathematical programming called *Edgeworth-Pareto optimization* that deals with multiple objective functions [22-24]. Stadler [25,26] was probably the first to apply Edgeworth-Pareto optimality to structural design. More recent applications can be found in Refs. 27–31.

A vector of design variables $\mathbf{x}^*$ is said to be Edgeworth-Pareto optimal if, for any other vector $\mathbf{x}$, either the values of all the objective functions remain the same, or at least one of them worsens compared to its value at $\mathbf{x}^*$. When it is not possible to specify intuitively the relative importance of the objective functions in an equation such as (1.2.1), the values of the weights α_i, $i = 0, 1, 2, 3$ in Eq. (1.2.1) can be decided by studying various Edgeworth-Pareto optimal designs. Thus the design process is an interactive process, and the imposition of constraints is postponed until knowledge of the optimum performance is gained by studying Edgeworth-Pareto optimal designs.

One of the approaches for generating a pareto-optimal solution to multiple objective function optimization problems is based on the minimization of the deviation of the individual objective functions from their individual minimum values. If the independent minimizations of each of the objective functions result in function values of $f_1^*, f_2^*, \ldots, f_p^*$ associated with design points $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_p^*$, then for an arbitrary value of the design variable vector $\mathbf{x}$ the normalized distance of each of the objective

functions from its individual optimum is given by

$$d_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^*}{f_i^*} \qquad i = 1, \ldots, p \ . \tag{1.2.2}$$

It is then possible to pose the problem either as the minimization of the largest deviation of the objective functions from their individual minima (l_∞ norm),

$$\textbf{minimize} \quad \max_{i=1,\ldots,p} [d_i(\mathbf{x})] \ , \tag{1.2.3}$$

or of the distance (i.e., the l_2 or Euclidean norm) from the reference point $\mathbf{f}^* = (f_1^*, f_2^*, \ldots, f_p^*)$ to $\mathbf{f} = (f_1, f_2, \ldots, f_p)$;

$$\textbf{minimize} \quad \sum_{i=1}^{p} d_i^2 \ . \tag{1.2.4}$$

It is also possible to use weighting coefficients in Eq. (1.2.4) for the contributions of the individual objective functions. A more detailed discussion of the methods for solving multicriteria optimization problems and their design applications is given by Eschenauer et al. [31].

Example 1.2.1

Consider the design of cross-sectional dimensions of a rectangular beam so as to minimize the area. At the same time it is desired to minimize the maximum shear stress in the beam corresponding to a unit shear force. Based on some physical constraints, the two variables, w and h, which are the width and height of the cross-section are limited to be in the range $0.5 \leq w, h \leq 5$ units.

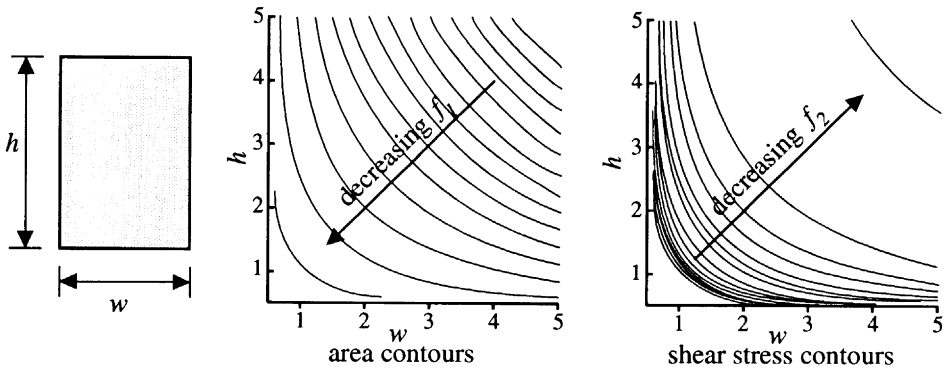


Figure 1.2.4 Design of a beam cross-section for minimum area and minimum shear stress.

The contour lines for the two objective functions,

$$f_1 = A = wh \ , \qquad \text{and} \qquad f_2 = \tau = \frac{3}{2wh} \ , \tag{a}$$

are shown in Figure 1.2.4 . The individual minima for the two functions are at the opposite corners of the design space, $w_1^* = h_1^* = 0.5$ and $w_2^* = h_2^* = 5.0$, with associated function values of $f_1^* = 0.25\ in^2$ and $f_2^* = 0.06\ lb/in^2$.

The weighted objective function approach with equal weights results in minimization of the function

$$F = wh + \frac{3}{2wh}. \tag{b}$$

Since design variables w and h appear everywhere in the form of a product, we can treat this product as a single variable. Minimization of Eq. (b) with respect to the product results in $w^*h^* = \sqrt{3/2} = 1.225$ with objective function values of $f_1^* = f_2^* = 1.225$. If, on the other hand, we use the minimization of the Euclidean norm of the distance from the individual minima, the function that needs to be minimized is

$$F = \left(\frac{hw - 0.25}{0.25}\right)^2 + \left(\frac{\dfrac{3}{2hw} - 0.06}{0.06}\right)^2 . \tag{c}$$

The resulting design is $w^*h^* = 2.5$ with objective function values of $f_1^* = 2.5$ and $f_2^* = 0.6$.

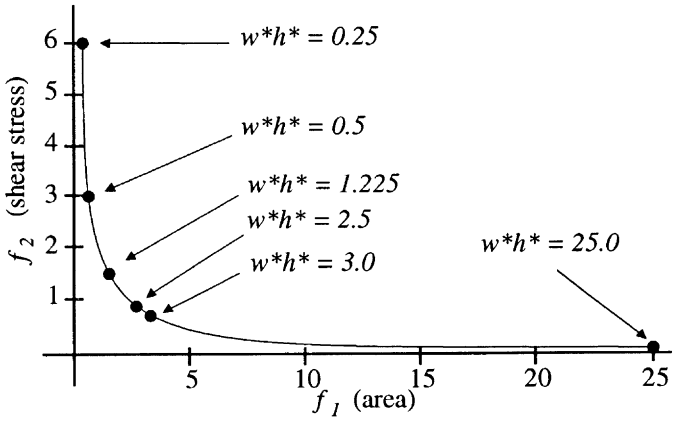


Figure 1.2.5 Pareto-optimum solutions for the beam design problem.

The two designs obtained above and the designs corresponding to the minimization of the individual functions constitute a pareto-optimum. There are other solutions that satisfy the condition for pareto-optimality. These solutions can be obtained either by varying the weighting coefficients of the individual objectives, or by

imposing one of the objectives as a constraint and varying the desired level of this constraint. For example, if the second objective function is turned into a constraint by imposing a condition that $f_2 \leq 0.5$ while minimizing the area, we would obtain a design $w^*h^* = 3.0$ with objective function values of $f_1^* = 3.0$ and $f_2^* = 0.5$. Similarly, if we minimize f_2 by imposing a constraint that $f_1 \leq 0.5$, we obtain $w^*h^* = 0.5$ with $f_1^* = 0.5$ and $f_2^* = 3$. All of these solutions lie on a curve in the function space that connect the two individual minima as shown in Figure 1.2.5 . This curve is usually called the efficiency curve. • • •

1.2.3 *Constraints*

The formulation of the three-bar truss example where the stresses are subject to upper limits, and the beam cross-section design problem where the height and width variables are limited to take values only in a certain range, introduces the notion of limits on the design variables. Because of their simplicity, these upper and lower limit constraints on the values of the design variables are often treated in a special way by solution procedures, and are refereed to as *side constraints.* Constraints which impose upper or lower limits on quantities are by their very nature *inequality constraints.* Sometimes we need *equality constraints.* For example, the three-bar truss may be designed subject to a requirement that the vertical component of the displacement at the point of application of the force be zero. Another example of equality constraints is provided by the equations of equilibrium that a structure must satisfy in terms of its design variables.

Some strategies for the solution of nonlinear optimization problems are unable to handle equality constraints, but are limited to inequality constraints only. In such instances it is possible to replace the equality constraint with two inequality constraints that form upper and lower bound constraints with a same limiting value. However, it is usually undesirable to increase the number of constraints. Another way of handling equality constraints in such situations will be discussed later in Chapter 5.

1.2.4 *Standard Formulation*

The notation adopted in this text for design variables, objective function and constraints is summarized in the following formulation of the optimization problem. In this text we deal only with problems formulated to have a single objective function.

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && g_j(\mathbf{x}) \geq 0 \ , \qquad j = 1, \ldots, n_g \ , \\ &&& h_k(\mathbf{x}) = 0 \ , \qquad k = 1, \ldots, n_e \ , \end{aligned} \tag{1.2.5}$$

where $\mathbf{x}$ denotes a vector of design variables with components x_i, $i = 1, \ldots, n$. The equality constraints $h_j(\mathbf{x})$ and the inequality constraints $g_j(\mathbf{x})$ are assumed to be transformed into the form (1.2.5). The fact that the optimization problem is assumed to be a minimization rather than a maximization problem is not restrictive

since instead of maximizing a function it is always possible to minimize its negative. Similarly, if we have an inequality of opposite type, that is

$$g_j(\mathbf{x}) \leq 0 \tag{1.2.6}$$

we can transform it to a greater-than-zero type by multiplying Eq. (1.2.6) by -1. However, while most optimization texts deal with minimization rather than maximization problems, many of them prefer less-than inequalities to greater-than ones. This choice affects the sign convention in some of the results obtained in this textbook, and the reader should be alert to this fact when comparing results with texts that use the opposite inequality convention.

An optimization problem is said to be linear when both the objective function and the constraints are linear functions of the design variables x_i, i.e., they can be expressed in the form

$$f(\mathbf{x}) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \mathbf{c}^T \mathbf{x} \ . \tag{1.2.7}$$

Linear optimization problems are solved by a branch of mathematical programming called *linear programming.* The optimization problem is said to be nonlinear if either the objective function or the constraints are nonlinear functions of the design variables.

Example 1.2.2

Consider the three-bar truss of Figure 1.2.3. Assume that it is made of steel (density 0.29 lb/in^3), and that we want to minimize the mass subject to the constraint that the stress in any member does not exceed 30,000 psi in tension or compression. We also impose a side constraint that the minimum area of any member is 0.1 in^2. The design variables are the member cross-sectional areas A_1, A_2, and A_3, and the horizontal coordinates x_1, x_2 and x_3 of the support points. The point of application of the force is assumed to be fixed. We seek to formulate this optimization problem in the standard form of (1.2.5).

The objective function is easy to write in terms of the design variables.

$$m = 0.29(A_1 L_1 + A_2 L_2 + A_3 L_3) \ ,$$

where

$$L_i = \sqrt{x_i^2 + 100^2} \ , \qquad i = 1, 2, 3 \ .$$

To calculate the stress constraint it is convenient to introduce the displacements u and v at the point of application of the force as intermediate variables. It can be verified that the equations governing u and v are

$$\begin{aligned} k_{11}(\mathbf{x})u + k_{12}(\mathbf{x})v &= 10,000 \ , \\ k_{12}(\mathbf{x})u + k_{22}(\mathbf{x})v &= 0 \ , \end{aligned}$$

where

$$k_{11}(\mathbf{x}) = E \sum_{i=1}^{3} \frac{A_i x_i^2}{L_i^3} ,$$

$$k_{12}(\mathbf{x}) = -E \sum_{i=1}^{3} \frac{100 A_i x_i}{L_i^3} ,$$

$$k_{22}(\mathbf{x}) = E \sum_{i=1}^{3} \frac{10,000 A_i}{L_i^3} ,$$

and where E is Young's modulus for steel (30×10^6psi). In terms of u and v, the stresses in the members are given as

$$\sigma_i = E(-u x_i / L_i^2 + 100 v / L_i^2) , \qquad i = 1,2,3 .$$

Based on the above analysis, one way of formulating the optimization problem in the standard form is to add u and v to the list of design variables. The formulation is

$$\begin{aligned} \textbf{minimize} \quad & m = 0.29(A_1 L_1 + A_2 L_2 + A_3 L_3) \\ \textbf{such that} \quad & h_1 = k_{11} u + k_{12} v - 10\,000 = 0 , \\ & h_2 = k_{12} u + k_{22} v = 0 , \end{aligned}$$

and

$$\begin{aligned} &\text{(tension constraints)} \quad g_i = 30\,000 - E(-u x_i + 100 v)/L_i^2 \geq 0 , \\ &\text{(compression constraints)} \quad g_{i+3} = E(-u x_i + 100 v)/L_i^2 + 30\,000 \geq 0 , \\ &\text{(minimum gage constraints)} \quad g_{i+6} = A_i - 0.1 \geq 0 , \qquad i = 1,2,3 . \end{aligned}$$

We then have a problem with eight design variables (A_i, x_i, $i = 1,2,3$ and u, v), two equality constraints and nine inequality constraints. This formulation including the response variables u and v together with the structural dimensions as design variables is called *simultaneous analysis and design.* Most structural optimization formulations eliminate the response variables by using the equations of equilibrium. In this problem we can solve for u and v from the equality constraints, thus eliminating two equality constraints and two design variables. The new formulation, which does not include the displacements as design variables, is much more common in structural optimization. As a result it is rare to encounter formulations of structural optimization problems which include equality constraints.• • •

While the above formulation of Example 1.2.2 conforms to our standard formulation, we may expect to encounter numerical difficulties when we solve this example using many standard solution techniques. The reason for the expected numerical difficulties is the large discrepancy between the magnitudes of the different design variables and constraints. Consider first the design variables. The area design variables may be expected to be of the order of the ratio of the applied force to the allowable stress, that is between 0.1 and 1 in^2. The coordinate design variables, on the other hand, may be expected to be of the order of 100 in.

Next consider the constraints. If the displacements u and v are about ten percent below or above their optimal values we can expect the equality constraints h_1 and h_2 to be of the order of magnitude of ten percent of the applied load. Similarly the inequality constraints g_1 through g_6 will be of the order of ten percent of the allowable stress, 30000 psi. However, the minimum gage constraints g_7 through g_9 will be of the order of 0.1 in^2.

Because many optimization software packages are not numerically robust, it is a good idea to eliminate such wide variations in the magnitudes of design variables and constraints by normalization. Design variables may be normalized to order 1 by scaling. In Example 1.2.2 the coordinate design variables may be normalized by the given vertical distance (100 in), and the area design variables by a nominal area, $A_0 = 1/3$ in^2, which is the ratio of the applied load to the allowable stress.

The constraints may be similarly normalized. Usually, inequality constraints can be normalized by the allowable value which is used to form them. Thus a constraint that a stress component σ be smaller than an allowable stress σ_{al} is often written as

$$g = \sigma_{al} - \sigma \geq 0 \ . \tag{1.2.8}$$

The value of the constraint depends on the units used, and can be large or small. Instead the constraint can be normalized as

$$\bar{g} = 1 - \frac{\sigma}{\sigma_{al}} \geq 0 \ . \tag{1.2.9}$$

Now the constraint values are of order one, and do not depend on the units used.

1.3 The Solution Process

The optimization methods discussed in this text are mostly numerical search techniques. These techniques start from an initial design and proceed in small steps to improve the value of the objective function, or the degree of compliance with the constraints, or both. The search is terminated when no progress can be made in improving the objective function without violating some of the constraints. Some optimization methods terminate when progress in improving the objective function becomes very slow. Others check for optimality by employing the necessary conditions, called the *Kuhn-Tucker conditions* (see Chapter 5), that must be satisfied at a minimum. We will typically use n to denote the number of design variables, so that the search for the optimum is carried out in the n-dimensional space of real variables R^n. Every point in this space constitutes a possible design.

In structural optimization problems the constraints imposed on the design, such as stress, displacements or frequency constraints, are important. That is, such constraints will affect the final design and force the objective function to assume a higher value than it would take without the constraints. For example, in Example 1.2.2, if the stress constraints were removed all the cross-sectional areas would be reduced to their minimum-gage values of 0.1 in^2, and the coordinates of points 1, 2 and 3, would

lie directly above point 4, so that the lengths of all three members would take the minimum value, 100 in., corresponding to a total mass of 8.7 lb. The resulting stresses in the members would tend to infinity. Since we cannot tolerate infinite stresses, we impose stress constraints, and we may expect that the optimum mass will be heavier than 8.7 lb., and that, at the optimum design, the stress in at least one member will be equal to the maximum allowable stress of 30,000 psi.

In general, we divide the space of design variables into a *feasible domain* and *infeasible domain.* The feasible domain contains all possible design points that satisfy all the constraints. The infeasible domain is the collection of all design points that violate at least one of the constraints. Because we expect that the constraints influence the optimum design, we expect that some constraints will be critical at the optimum design. This is equivalent to the optimum being on the boundary between the feasible and infeasible domains. Inequality constraints in our standard formulation, Eq. (1.2.5), are critical when they are equal to zero. These constraints are also called *active* constraints, while the rest of the constraints are *inactive* or *passive.* For example, consider the minimum gage constraint g_7 of Example 1.2.2. For $A_1 = 0.1\text{in}^2$ the constraint is active, for $A_1 = 0.11\text{in}^2$ the constraint is passive, and for $A_1 = 0.09\text{in}^2$ the constraint is violated.

It may be intuitively assumed that all the constraints which are active at the optimum design influence it; that is, if they were removed the objective function could be improved. This is not always true. It is possible to have constraints that are active and can be removed without any impact on the optimum design. Many optimization procedures calculate, along with the optimum design, a set of numbers, one for each active constraint, called the *Lagrange multipliers* (see Chapter 5) which measure the sensitivity of the optimum design to changes in each constraint. When the Lagrange multiplier associated with a constraint is zero, it indicates that, to a first order approximation, removing this constraint will not have any effect on the optimum value of the objective function. These multipliers also provide very important design information because in many structural optimization applications there is some degree of arbitrariness in the choice of parameters that determine the constraints such as stress limits or minimum gage values. For example, when we impose stress constraints on a steel structure we typically select ahead of time the grade of steel to be used. We can use the Lagrange multipliers to estimate the effect of a change in the stress limit on the objective function. If we find that the optimum design is very sensitive to this value we may consider using a better grade of steel.

One of the major problems in almost all optimization solution procedures is the determination of the set of active constraints. If the solution procedure attempts to consider all constraints during the search process the computational cost of the optimization may be significantly increased. If, on the other hand, the procedure deals only with constraints that are active or near active for the trial design, the convergence of the optimization process may be endangered due to oscillation in the set of active constraints. Most optimization procedures are usually complemented by an *active set* strategy used to determine the set of constraints to be considered at each trial design.

During the optimization process we move from one design point to another. While

there are many optimization techniques, most of them proceed through four basic steps in performing the move. The first step is the selection of the active constraint set discussed above. The second step is the calculation of a search direction based on the objective function and the active constraint set. Some methods (such as the gradient projection method) look for a direction which is tangent to the active constraint boundaries. Other methods, such as the feasible direction or the interior penalty function method seek to move away from the constraint boundaries. The third step is to determine how far to go in the direction found in the previous step. This is often done by a process called a *one dimensional line search* because it seeks the magnitude of a single scalar which is the distance to be travelled along the given direction. The last step is a convergence step which determines whether additional moves are required.

1.4 Analysis and Design Formulations

In a practical design situation it is not always clear which mathematical formulation of the structural design problem should be used. Consider, for example, the beam of Figure 1.1.1, and assume that the designer wants to achieve a high-stiffness, low-mass design. One option that the designer may elect is to employ a multiple objective function formulation where both the mass m in Eq. (1.1.2) and the maximum displacement w_{max}, Eq. (1.1.3), are to be minimized simultaneously. A second approach is to assign some weights α_1 and α_2 to the two objectives and use them to form a composite objective function $\alpha_1 m + \alpha_2 w_{max}$. Third, it is possible to set the mass as the objective function and constrain the magnitude of w_{max}. Finally, it is possible to prescribe an upper limit on the mass and use the maximum displacement as the objective function.

All of the above formulations may be acceptable for the design goal of producing a strong, lightweight design. However, the mathematical formulation and the solution difficulties may be quite different. For example, if $p = 1$ in Eq. (1.1.2), the mass is a linear function of the design variable $I(x)$ while the maximum displacement w_{max} is not. Some nonlinear optimization procedures work better when the objective is linear and the constraints are nonlinear, and others work better when the situation is reversed. The choice of the formulation may be decided, therefore, on the basis of the optimization software available to the designer.

The formulation and the solution of the structural optimization problem is also important. First, because the analysis has to be repeated many times during the optimization process it may be crucial to use a solution method that is computationally inexpensive. Thus a detailed finite element model that is typically used for a single analysis of the structure may not be affordable for optimization, and it may have to be replaced with a cruder model.

The choice of the structural analysis solution process may be similarly influenced by the optimization environment. For example, vibration frequencies and modes of a structure are typically calculated by an eigenvalue solution procedure. Some of these procedures benefit from good initial approximations for the eigenvectors and some

do not. For applications in structural optimization the former procedure gains an advantage because the eigenvectors (vibration modes) change only gradually as the design is modified. Therefore the eigenvectors from an earlier design can serve as good initial approximations for the current eigenvectors.

Finally, in some cases it may be worthwhile to integrate the analysis and design procedures. This happens when structural analysis is iterative in nature, as in the case of nonlinear structural behavior. The analysis and design iterations may be then integrated so that the analysis iteration is only partially converged for each design iteration (e.g. [32, 33]). In some cases it may be worthwhile to combine the analysis and design iterations into a single iterative process. This *simultaneous analysis and design* approach is discussed in Chapter 10.

1.5 Specific Versus General Methods

The solution methods commonly used for obtaining optimum designs in structural optimization may be divided into different categories. An important classification of solution methods considers specific versus general methods. Specific methods are used exclusively in structural optimization (even if they could be applicable also in other disciplines). General methods apply to optimization problems in several other fields. In the early stages of the development of structural optimization, specific methods enjoyed great popularity. These included methods tailored to some special structural optimization problems which they could solve more efficiently than any general method.

The most successful of these specific methods was the *fully stressed design* technique described in Chapter 9. It is a method applicable to the design of a structure subject to stress constraints only, and it works well for lightly-redundant single-material structures.

The popularity of specific methods is currently waning as their limitations become increasingly apparent. The approach taken in this text is to emphasize general methods rather than specific ones. General methods not only have the advantage of wider applicability but also a wider base of resources. Researchers in many disciplines are constantly improving these methods and developing efficient and reliable software implementations.

Besides playing down the role of specialized methods for structural design we also do not discuss some mathematical programming methods applicable to *problems of specialized form* such as dynamic programming, geometric programming and optimal control techniques. These methods have been applied successfully to structural design problems, but because of space considerations they are not covered here. The reader is referred to Refs. 34-36 for information on the application of these methods to structural design.

The important considerations for a structural analyst using general optimization methods have to do with providing an interface between structural analysis software and optimization software. This interface includes the three major components of

formulation, sensitivity and approximation, and is one of the major thrusts of this text.

The formulation of a structural design problem is of crucial importance for the success of the design process. A poor formulation can lead to poor results or prohibitive computational cost. Chapter 3, for example, describes various structural design problems that can be formulated with a linear objective function and linear constraints. The reason for the usefulness of a linear formulation is the highly advanced state of methodology and software for solving such linear problems.

The efficient calculation of derivatives of the constraints and objective function with respect to design variables, often referred to as *sensitivity derivatives*, is discussed in Chapters 7 and 8. Most general purpose optimization algorithms require such derivatives, and their calculation is often the major computational expense in the optimization of structures modeled by complex finite element models. These derivatives can also be used to form constraint approximations which can then be employed instead of costly exact constraint evaluations during portions of the optimization process. The use of constraint approximations is discussed in Chapter 6.

The importance of efficient and accurate calculation of sensitivity derivatives and of employing constraint approximations is now recognized by most structural optimization specialists. We believe that it affects the success and overall computational cost of the optimization process even more than the choice of the optimization method.

1.6 Exercises

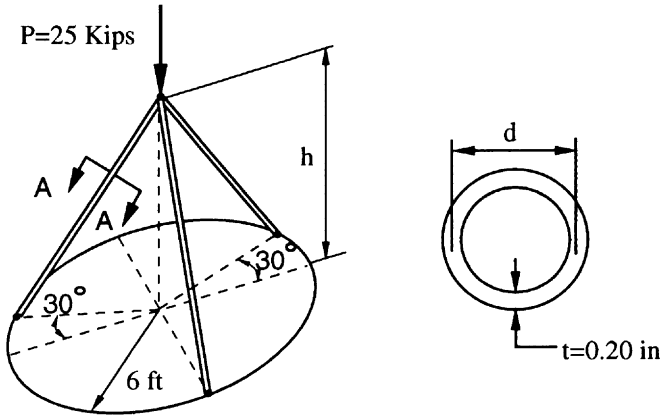


Figure 1.6.1 A tripod under a vertical load.

1. A tripod is made from three steel pipes as shown in Figure 1.6.1. The ends of these pipes are placed 120° apart on a circle of radius 6 ft. A vertical downward force of 25 kips is applied at the top. It is required to minimize the weight of the tripod such that the tripod is safe with respect to Euler buckling, local buckling and yielding. Assume $E = 30 \times 10^6$ psi, $\sigma_{yield} = 60 \times 10^3$ psi and calculate the local buckling stress in psi by the formula

$$\sigma_{cr} = 36 \times 10^6 \left(\frac{t}{d}\right) .$$

Sketch the constraints in the two-dimensional design space of d (mean diameter of pipe) and h. Identify the feasible and infeasible domains; plot the contours of the objective function and locate the optimum solution.

2. A narrow rectangular beam with cross-sectional dimensions b and h is cantilevered over 20 ft and subjected to an end load $p = 10$ kips (Figure 1.6.2). In addition to a flexural failure such beams can collapse through lateral instability by twisting. The critical load for such a beam of length l is given by

$$p_{cr} = 4.013\sqrt{EI_{least}c}/l^2 ,$$

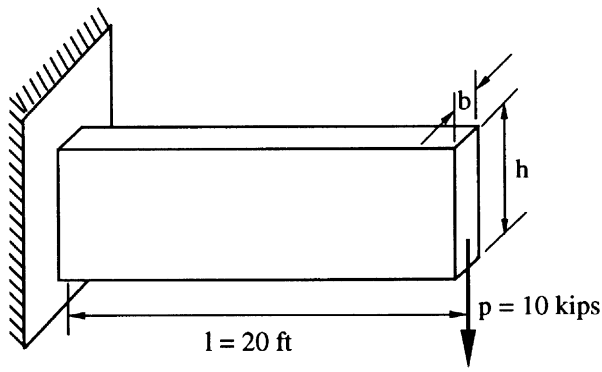


Figure 1.6.2 A narrow rectangular cantilever beam.

where E is Young's modulus, I_{least} is the smallest moment of inertia and c is the torsional rigidity of the beam given by $0.312hb^3G$, G being the shear modulus. Design a minimum weight beam so as to prevent failure in both flexure and twisting. Assume $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi and $\sigma_{al} = 75$ ksi in tension and compression. Locate the optimum solution graphically.

3. Consider the design of the cross-section of an I-beam shown in Figure 1.6.3 with the objectives of minimizing the cross-sectional area and minimizing the normal stresses resulting from bending about the horizontal neutral axis. The thicknesses of the flange and the web of the cross-section are fixed at $t = 0.1in$. The design variables are the width w and the height h of the cross-section. Determine graphically the designs which minimize the individual objectives if the width and the height are

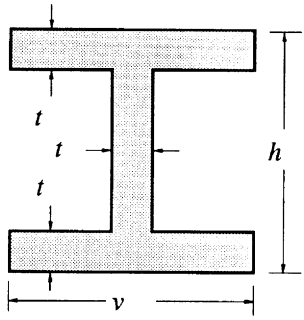


Figure 1.6.3 Cross-sectional design of an I-beam.

constrained to remain in the range $0.1 \leq w, h \leq 10$. Also find the designs by using weighting function approach with equal weights, and using Eq. (1.2.4).

4. The elastic grillage of Figure 1.6.4 consists of two uniform beams with cross-sectional areas A_1 and A_2. Both beams are subjected to a uniformly distributed load of 1000 lb/in. The minimum weight design of such a structure was first proposed by Moses and Onada [37]. Develop expressions for the maximum stresses in tension and compression at sections 1, 2 and 3 in terms of A_1 and A_2. Assume that the section modulus z and moment of inertia I are related to the cross-sectional area as

$$z = \left(\frac{A}{1.48}\right)^{1.82}, \qquad I = 1.007\left(\frac{A}{1.48}\right)^{2.65}.$$

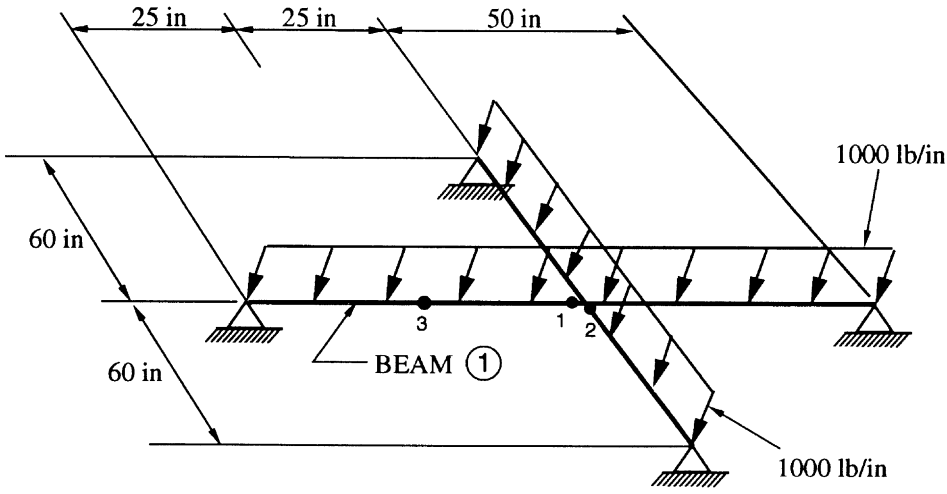


Figure 1.6.4 An elastic grillage under uniform load.

Assuming an allowable stress of 20,000 psi in tension and compression, formulate the five constraints and the objective function. Plot the constraints and the objective function. Identify the feasible and infeasible domains. Comment on the characteristics of the feasible domain in contrast with those of the previous two problems. Determine the *best* design for the grillage.

1.7 References

[1] Wilde, D.J., Globally Optimal Design, John Wiley and Sons, New York, 1978.

[2] Wasiutynski, Z., and Brandt, A., "The Present State of Knowledge in the Field of Optimum Design of Structures," Appl. Mech. Rev., 16 (5), pp. 341–348, May 1963.

[3] Sheu, C.Y., and Prager, W., "Recent Developments in Optimum Structural Design," Appl. Mech. Rev., 21 (10), pp. 985–992, Oct. 1968.

[4] Schmit, L.A. Jr., "Structural Synthesis 1959–1969: A Decade of Progress," in Recent Advances in Matrix Methods of Structural Analysis and Design, University of Alabama Press, Huntsville, pp. 565–634, 1971.

[5] Pierson, B.L., "A Survey of Optimal Structural Design Under Dynamic Constraints," Int. J. Num. Meth. Eng., 4, pp. 491–499, 1972.

[6] Niordson, F.I., and Pedersen P., "A Review of Optimal Structural Design," in Theoretical and Applied Mechanics, Proceedings of the Thirteenth International Congress of Theoretical and Applied Mechanics, E. Becker and G. K. Mikhalov (eds.), pp. 264–278, Springer-Verlag, Berlin, 1973.

[7] Rao, S.S., "Optimum Design of Structures under Shock and Vibration Environment," Shock Vibr. Digest, 7 (12), pp. 61–70, Dec. 1975.

[8] Olhoff, N. J., "A Survey of Optimal Design of Vibrating Structural Elements, Parts I and II," Shock Vibr. Digest, 8 (8&9) , pp. 3–10,1976.

[9] Venkayya, V. B., "Structural Optimization: A Review and Some Recommendations," Int. J. Num. Meth. Eng., 13, pp. 203–228, 1978.

[10] Lev, O. E., (ed.), Structural Optimization—Recent Developments and Applications, ASCE Committee on Electronic Computation, New York, 1981.

[11] Schmit, L.A., "Structural Synthesis–its Genesis and Development," AIAA J., 19 (10), pp. 1249–1263, 1981.

[12] Haug, E.J., "A Review of Distributed Parameter Structural Optimization Literature," in Optimization of Distributed Parameter Structures, E.J. Haug and J. Cea (eds.), Vol. 1, pp. 3–74, Sijthoff and Noordhoff, Alphen aan den Rijn, the Netherlands, 1981.

[13] Ashley, H., "On Making Things the Best—Aeronautical Uses of Optimization," J. Aircraft, 19 (1), pp. 5–28, 1982.

[14] Kruzelecki, J., and Zyczkowski, M., "Optimal Structural Design of Shells—A Survey," SM Archives, 10, pp. 101–170, 1985.

[15] Haftka, R. T., and Grandhi, R. V., "Structural Shape Optimization—A Survey," Computer Methods in Applied Mechanics and Engineering, 57, pp. 91–106, 1986.

[16] Bushnell, D., Holmes A. M. C., Flaggs, D. L., and McCormick, P. J., "Optimum Design, Fabrication and Test of Graphite-Epoxy, Curved, Stiffened, Locally Buckled Panels Loaded in Axial Compression", in Buckling of Structures (ed. I. Elishakoff et al.) Elsevier Science Publishers B. V., Amsterdam, pp. 61–131, 1988.

[17] Kirsch, U., "Optimal Topologies of Structures," Appl. Mech. Rev., 42, No. 8, pp. 223–239, 1989.

[18] Friedmann, P. P., "Helicopter Vibration Reduction Using Structural Optimization with Aeroelastic/Multidisciplinary Constraints—A Survey," J. Aircraft, 28, No. 1, pp. 8–21, 1991.

[19] Sobieszczanski-Sobieski, J., "Structural Optimization: Challenges and Opportunities," Int. J. Vehicle Design, 7, pp. 242–263, 1986.

[20] Prasad, B., and Haftka, R. T., "Optimal Structural Design with Plate Finite Elements," ASCE J. Structural Division, 105, pp. 2367–2382, 1979.

[21] Braibant, V., Fleury, C., and Beckers, P., "Shape Optimal Design: An Approach Matching C.A.D. and Optimization Concepts," Report SA-109, Aerospace Laboratory of the University of Liege, Belgium, 1983.

[22] Edgeworth, F. Y., Mathematical Physics, London, England, 1881.

[23] Pareto, V., Manuale di Economia Politica, Societa Editrice Libraria, Milano, Italy, 1906. Translated into English by A.S. Schwier as Manual of Political Economy, MacMillan, New York, 1971.

[24] Zeleny, M., Multiple Criteria Decision Making, McGraw-Hill Book Company, New York, 1972.

[25] Stadler, W., "Natural Structural Shapes of Shallow Arches," J. Appl. Mech, 44, pp.291–298, 1977.

[26] Stadler, W., "Natural Structural Shapes (The Static Case)," Q. J. Mech. Appl. Math., 31, pp. 169–217, 1978.

[27] Adali, S., "Pareto Optimal Design of Beams Subjected to Support Motions," Computers and Structures, 16, pp. 297–303, 1983.

[28] Bendsøe, M.P., Olhoff, N., and Taylor, J.E., "A Variational Formulation for Multicriteria Structural Optimization," J. Struct. Mech., 11 (4), pp. 523–544, 1984.

[29] Stadler, W., "Applications of Multicriterion Optimization in Engineering and the Sciences," in MCDM—Past decade and Future Trends, (Zeleny M., ed.), JAI Press, Greenwich, Conn., 1984.

[30] Stadler, W., (ed.), Multicriteria Optimization in Engineering and in the Sciences, Plenum Press, New York, 1988.

[31] Eschenauer, H., Koski, J., and Osyczka, A., Multicriteria Design Optimization: Procedures and Applications, Springer-Verlag, New York, 1990.

[32] Wu, C.C., and Arora, J.S., "Simultaneous Analysis and Design Optimization of Nonlinear Response," Engineering with Computers, 2, pp. 53–63, 1987.

[33] Haftka, R.T., "Integrated Analysis and Design", AIAA J., 27, 11, pp.1622-1627, 1989.

[34] Carmichael, D.G., Structural Modeling and Optimization, Halstead Press, England, 1981.

[35] Palmer, A.C., "Optimal Structural Design by Dynamic Programming," J. Struct. Div. ASCE, 94, No. ST8, pp. 1887–1906, 1968.

[36] Hajela, P., "Geometric Programming Strategies in Large-Scale Structural Synthesis", AIAA J., 24 (7), pp. 1173–1178, 1986.

[37] Moses, F., and Onoda, S., "Minimum Weight Design of Structures with Applications to Elastic Grillages", Int. J. Num. Meth. Eng., 1, pp. 311–331, 1969.

Classical Tools in Structural Optimization 2

Classical optimization tools used for finding the maxima and minima of functions and functionals have direct applications in the field of structural optimization. The words 'classical tools' are implied here to encompass the classical techniques of ordinary differential calculus and the calculus of variations. Exact solutions to a few relatively simple unconstrained or equality constrained problems have been obtained in the literature using these two techniques. It must be pointed out, however, that such problems are often the result of simplifying assumptions which at times lack realism, and result in unreasonable configurations. Still, the consideration of such problems is not a purely academic exercise, but is very helpful in the process of solving more realistic problems.

In recent years there has been an increased interest in the application of classical tools, especially variational methods, in structural optimization. Mathematical formulations of broad classes of structures as optimization problems have been achieved by adopting variational methods. In addition, the study of classical problems not only serves to portray the underlying principles of the techniques of classical methods, but it serves an even more basic need in structural optimization. Closed form exact solutions to classical problems serve to validate solutions obtained using more general but approximate numerical techniques. More importantly, classical optimization is perhaps the best vehicle for letting a student of structural optimization appreciate fully the questions of the existence and uniqueness of the optimum designs, and the establishment of the necessity and sufficiency of the optimality conditions. Such questions can be rigorously answered for only the simplest problems of optimization similar to those considered in this chapter.

2.1 Optimization Using Differential Calculus

In the absence of constraint equations a continuously differentiable objective function $f(x_1, x_2, \ldots, x_n)$ of n independent design variables attains a maximum or a minimum value in the interior of the design space $\mathbf{R}^n$ only at those values of the design variables

$\mathbf{x}^*$ for which the n partial derivatives

$$\frac{\partial f}{\partial x_1}, \quad \frac{\partial f}{\partial x_2}, \quad \ldots\ldots, \quad \frac{\partial f}{\partial x_n}, \tag{2.1.1}$$

vanish simultaneously. This is the necessary condition for the point $\mathbf{x}^*$ to be a stationary point. We will see in later chapters that this property proves to be a valuable tool in locating the optimum solution. For a scalar valued function, the vector of first derivatives is referred to as the gradient vector ∇f and is used for finding search directions in optimization algorithms.

Development of a sufficient condition for a stationary point $\mathbf{x}^*$ to be an extreme point requires the evaluation of the matrix of second derivatives $\mathbf{H}$ of the objective function. The matrix of second derivatives is also referred as the Hessian matrix and defined as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \tag{2.1.2}$$

It can be proved that if the matrix of second derivatives evaluated at $\mathbf{x}^*$ is positive-definite then the stationary point is a minimum, if it is negative-definite then the stationary point is a maximum point [1]. A symmetric matrix $\mathbf{H}$ is said to be positive (negative)-definite if the quadratic form $Q = \mathbf{x}^T\mathbf{H}\mathbf{x}$ is positive (negative) for every $\mathbf{x}$, and is equal to zero if and only if $\mathbf{x} = \mathbf{0}$. A computational check for the positive and negative definiteness of a matrix involves determinants of the principal minors, $\mathbf{H}_i (i = 1, \ldots, n)$. A principal minor $\mathbf{H}_i$ is a square sub-matrix of $\mathbf{H}$ of order i whose principal diagonal lies along the principal diagonal of the matrix $\mathbf{H}$. The matrix $\mathbf{H}$ is positive-definite if the determinants of all the principal minors located at the top left corner of the matrix are positive; and negative-definite if $-\mathbf{H}$ is positive definite. Alternatively, $-\mathbf{H}$ is positive definite if $\mathbf{H}_1$ is negative and the following principal minors, $\mathbf{H}_2, \mathbf{H}_3, \ldots, \mathbf{H}_n$, are alternately positive and negative [1]. Another property of positive (negative)-definite matrices can be used as a test. A symmetric matrix is positive (negative)-definite if and only if all its eigenvalues are positive (negative).

A symmetric matrix $\mathbf{H}$ is called positive semi-definite if the quadratic form $Q = \mathbf{x}^T\mathbf{H}\mathbf{x}$ is non-negative for every $\mathbf{x}$. This happens when the eigenvalues of the matrix are non-negative. Unfortunately, the expected condition that the principal minors are non-negative is not sufficient for positive semi-definiteness. If a matrix is positive semi-definite but not positive-definite, then there exist at least one $\mathbf{x} \neq \mathbf{0}$ such that the quadratic form is zero, at least one of the principal minors is zero, the matrix is singular, and at least one of the eigenvalues is zero. In that case higher order derivatives of the function f are needed to establish sufficient conditions for a minimum. Similarly, when $-\mathbf{H}$ is positive semi-definite then $\mathbf{H}$ is negative semi-definite. If $\mathbf{H}$ is negative semi-definite but not negative-definite, we need higher order derivatives to establish sufficient conditions for a maximum. Finally when $\mathbf{H}$ is neither positive semi-definite nor negative semi-definite, it is called indefinite. In that case the stationary point is neither a minimum nor a maximum but a saddle point.

Two simple examples demonstrate the use of differential calculus in finding optimum structural configurations.

Example 2.1.1

The symmetric statically determinate truss structure shown in Figure (2.1.1) is to be designed for minimum weight by varying the heights h_1 and h_2 of the vertical members. Because the truss is statically determinate the forces in the members are independent of the cross-sectional area, so that the areas can be reduced until each member is fully stressed (its stress is equal to the allowable stress σ_o).

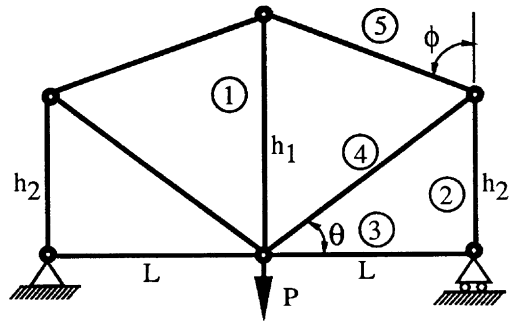


Figure 2.1.1 Fully stressed minimum weight truss.

For the loading shown in the figure, the forces in each of the members can be expressed in terms of the geometry of the structure as

$$F_1 = \frac{h_1 - h_2}{h_1} P, \qquad F_2 = -\frac{P}{2}, \qquad F_3 = 0, \tag{2.1.3}$$

$$F_4 = \frac{(h_2^2 + L^2)^{\frac{1}{2}}}{2h_1} P, \qquad F_5 = -\frac{[(h_1 - h_2)^2 + L^2]^{\frac{1}{2}}}{2h_1} P\,. \tag{2.1.4}$$

If each member is to be fully stressed the cross sectional areas of the members A_i can be related to the forces carried by the members as

$$A_i = \frac{|F_i|}{\sigma_o}, \qquad i = 1, \ldots, 9\,. \tag{2.1.5}$$

From Eq. (2.1.3) the cross-sectional area A_3 of the horizontal members vanishes. However, based on stability considerations these members may be assumed to have a minimum area of A_{min}. The contribution of the weight of these members to the total weight of the structure is independent of the design variables h_1 and h_2, and

will be ignored for the mini .zation problem. The total volume of material in the remaining truss structure is ıe sum of the products of the cross sectional areas and the member lengths that can be expressed in terms of the unknown variables. It can be shown that the remaining total volume is

$$V = 2\frac{P}{\sigma_o}[h_1 - h_2 + \frac{h_2^2}{h_1} + \frac{L^2}{h_1}]. \tag{2.1.6}$$

Differentiating the volume with respect to the unknown variables we obtain

$$\frac{\partial V}{\partial h_1} = 2\frac{P}{\sigma_o}\left(1 - \frac{h_2^2}{h_1^2} - \frac{L^2}{h_1^2}\right) = 0, \qquad \frac{\partial V}{\partial h_2} = 2\frac{P}{\sigma_o}\left(-1 + \frac{2h_2}{h_1}\right) = 0\,. \tag{2.1.7}$$

The resulting optimum values for the heights are

$$h_1^* = \frac{2}{\sqrt{3}}L, \qquad h_2^* = \frac{1}{\sqrt{3}}L\,, \tag{2.1.8}$$

and the cross sectional areas of the members are equal to

$$A_1 = A_2 = A_4 = A_5 = \frac{P}{2\sigma_o}\,. \tag{2.1.9}$$

The matrix of second derivatives of the objective function for the problem is

$$\mathbf{H} = 2\frac{P}{\sigma_o}\begin{bmatrix} (2/h_1^3)(h_2^2 + L^2) & -2h_2/h_1^2 \\ -2h_2/h_1^2 & 2/h_1 \end{bmatrix}, \tag{2.1.10}$$

which, evaluated at the optimum values of the design variables, is

$$\mathbf{H}^* = 2\frac{P}{\sigma_o}\frac{\sqrt{3}}{L}\begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix}. \tag{2.1.11}$$

The matrix $\mathbf{H}^*$ is positive definite (check principal minors), thereby, proving the sufficiency condition for the optimality of the design. • • •

Example 2.1.2

Consider an inextensible structural cable with zero bending stiffness. The cable is stretched by applying a horizontal force F_h at the ends of the cable, two points separated by a distance L, and carries a vertical distributed load of intensity p(x), Figure (2.1.2).

If the cross-sectional area of the cable is allowed to vary along its length so that the axial stress is equal to the allowable stress σ_o, determine the optimum value of the horizontal pull F_h that will minimize the total volume of material of the cable for a uniform load of $p(x) = p_o$.

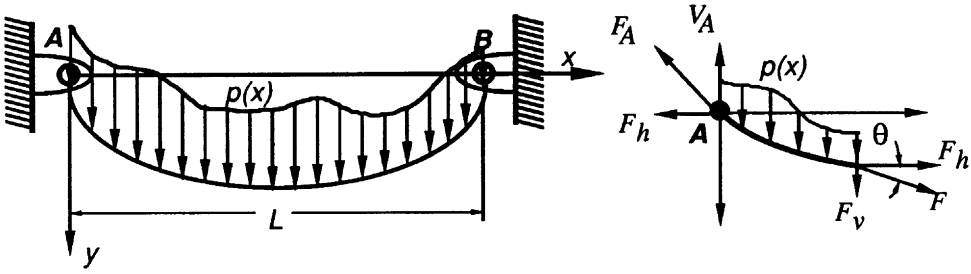


Figure 2.1.2 Structural cable design.

Neglecting the weight of the cable, we obtain the equilibrium equations in the horizontal and vertical directions of a cable as

$$F\cos\theta = F_h = \text{constant}, \qquad \text{and} \quad F_h\frac{d^2y}{dx^2} = -p(x)\,, \tag{2.1.12}$$

where θ is the angle between the horizontal coordinate axis x and the tangent to the arc length coordinate s such that $\cos\theta = ds/dx$. For a uniform loading, the second equilibrium equation can be solved for the vertical displacement along the length of the cable by integrating twice and making use of the zero displacement conditions at the two ends to yield

$$y = \frac{p_o L^2}{2F_h}[(\frac{x}{L}) - (\frac{x}{L})^2]\,. \tag{2.1.13}$$

The total volume of material in the cable to be minimized is

$$V = \int_0^L dV\,, \tag{2.1.14}$$

where

$$dV = A(s)ds\,. \tag{2.1.15}$$

With the assumption that the cross-sectional area is to be fully stressed, $A(s) = F/\sigma_o$, the total volume can be expressed as

$$V = \int_0^L \frac{F_h}{\sigma_o}(\frac{ds}{dx})^2 dx. \tag{2.1.16}$$

Since

$$ds = [1 + (\frac{dy}{dx})^2]^{\frac{1}{2}}dx\,, \tag{2.1.17}$$

Eq. (2.1.16) can be written as

$$V = \frac{F_h}{\sigma_o}\int_0^L [1 + (\frac{dy}{dx})^2]dx\,. \tag{2.1.18}$$

Substituting the first derivative of the displacement function of Eq. (2.1.13) into the above equation, we can show that the volume of the material is related to the horizontal pull as,

$$V = \frac{L}{\sigma_o}[F_h + \frac{{p_o}^2 L^2}{12 F_h}]. \tag{2.1.19}$$

If the horizontal pull is small, the volume increases because the cable becomes longer. If, on the other hand, the horizontal pull is very large the cross-sectional area has to be large in order to keep the stress level at σ_o, although the length of the cable approaches the minimum distance between the support points.

The optimum value of the horizontal pull can be obtained from

$$\frac{dV}{dF_h} = 0, \tag{2.1.20}$$

which produces

$${F_h}^* = \frac{p_o L}{\sqrt{12}}. \tag{2.1.21}$$

This corresponds to a minimum total volume of

$$V^* = \frac{p_o L^2}{\sqrt{3}\sigma_o}, \tag{2.1.22}$$

and an optimal cross-sectional area distribution of

$$A^*(x) = \frac{F_h}{\sigma_o \cos\theta} = \frac{F_h}{\sigma_o}\sqrt{1 + (\frac{dy}{dx})^2} = \frac{p_o L}{\sigma_o}\sqrt{\frac{1}{12} + (\frac{1}{2} - \frac{x}{L})^2}. \tag{2.1.23}$$

• • •

Although applications of classical calculus can be demonstrated for many other structures such as beams and arches, it is appropriate to mention the aspects and assumptions which make these problems tractable using ordinary calculus. The truss example discussed above, for example, could be treated by ordinary calculus because of several simplifying assumptions. First, some of the potential design variables, such as the cross-sectional areas of the truss members, were eliminated by assuming the stresses in each member to be equal to the maximum allowable value. Second, the analysis was simplified by neglecting the effect of selfweight of the truss members on structural response, and by ignoring possible buckling of those members loaded in compression. Most realistic structural optimization problems cannot be simplified to the point where they can be solved by ordinary calculus.

2.2 Optimization Using Variational Calculus

Some structural design problems, when formulated as optimization problems, have an objective function in the form of a definite integral involving an unknown function and some of its derivatives. Such forms, called functionals, assume a specific numerical value for each function that is substituted into it. The task of the designer is to find a suitable function that minimizes the functional. The branch of mathematics that deals with the maxima and minima of functionals is called the Calculus of Variations. Certain aspects of the methods used in the calculus of variations are analogous to procedures used for differential calculus, and are discussed in this section.

2.2.1 *Introduction to the Calculus of Variations*

Consider the problem of determining a function y(x) given at two points, $y(a) = y_a$ and $y(b) = y_b$, for which the integral

$$J = \int_a^b F(x, y, y')dx\,, \tag{2.2.1}$$

assumes a minimum or a maximum value ($y' \equiv dy/dx$). The end conditions on $y(a)$ and $y(b)$ are referred to as *kinematic boundary conditions* for the problem. In a more general case F can be a function of more than one function $(y_1, y_2, ..., y_p)$, and each of these functions can depend on n independent variables $(x_1, x_2, ..., x_n)$. Also, higher order derivatives of these functions with respect to the independent variables may be included in F. This brief introduction, however, is limited to a functional expressed in terms of a single function with one independent variable. A more general discussion of the methods of variational calculus is available in many textbooks (e.g., [2-4]).

Assuming $y^*(x)$ to be the function that minimizes our integral, consider another function $y(x)$ obtained by a small variation δy from $y^*(x)$,

$$y(x) \quad = \quad y^*(x) + \delta y \quad = \quad y^*(x) + \epsilon\eta(x)\,, \tag{2.2.2}$$

where ϵ is a small amplitude parameter and $\eta(x)$ a shape function. The function $\eta(x)$ must satisfy the kinematic boundary conditions

$$\eta(a) = 0\,, \quad \text{and} \quad \eta(b) = 0\,, \tag{2.2.3}$$

so that $y(a)$ and $y(b)$ will remain unchanged. We substitute Eq. (2.2.2) into the integral (2.2.1), so that J becomes a function of only the perturbation parameter ϵ

$$J(\epsilon) = \int_a^b F(x, y^* + \epsilon\eta, y^{*\prime} + \epsilon\eta')dx\,. \tag{2.2.4}$$

Knowing that the value of the integral J attains an extremum for $\epsilon = 0$, one can use ordinary calculus to write the necessary condition

$$\frac{dJ(\epsilon)}{d\epsilon}\Big|_{\epsilon=0} = \int_a^b \left(\frac{\partial F}{\partial y}\frac{dy}{d\epsilon} + \frac{\partial F}{\partial y'}\frac{dy'}{d\epsilon}\right) dx = 0 . \tag{2.2.5}$$

Using Eq. (2.2.2) and defining $\epsilon\,(dJ/d\epsilon\,|_{\epsilon=0})$ to be the first variation of the functional J denoted by δJ we obtain

$$\delta J = \int_a^b (\frac{\partial F}{\partial y}\delta y + \frac{\partial F}{\partial y'}\delta y')dx = 0 . \tag{2.2.6}$$

The variational operator δ is analogous to the differential operator in ordinary calculus, and the same rules that apply to the differential operator apply to the variational operator. The property of interchangeability of the two operators

$$\epsilon\eta' = \epsilon\frac{d\eta}{dx} = \frac{d}{dx}\epsilon\eta = \frac{d}{dx}\delta y = \delta(\frac{dy}{dx}) = \delta y' , \tag{2.2.7}$$

has been used in order to arrive at Eq. (2.2.6).

In the more general case F depends on more than one function and on higher order derivatives of these functions with respect to the independent variable x. For example, if

$$J = \int_a^b F(x, y_1, y_2, y_1', y_2', y_2'')dx , \tag{2.2.8}$$

then the condition that variation of the functional is zero may be written as

$$\delta J = \int_a^b (\frac{\partial F}{\partial y_1}\delta y_1 + \frac{\partial F}{\partial y_1'}\delta y_1' + \frac{\partial F}{\partial y_2}\delta y_2 + \frac{\partial F}{\partial y_2'}\delta y_2' + \frac{\partial F}{\partial y_2''}\delta y_2'')dx = 0 . \tag{2.2.9}$$

The necessary condition for extremum expressed in the form of Eq. (2.2.6) or (2.2.9) is usually not very useful. The terms that involve variation of derivatives can be integrated by parts in order to obtain more useful conditions. For example, integrating the second term of Eq. (2.2.6) and rearranging we write

$$\delta J = \frac{\partial F}{\partial y'}\delta y\Big|_a^b + \int_a^b \left[\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right)\right]\delta y dx = 0 . \tag{2.2.10}$$

For our problem the first term on the right hand side of Eq. (2.2.10) vanishes due to the fact that the arbitrary function $\eta(x)$ satisfies the boundary conditions, $\eta(a) = \eta(b) = 0$. By the definition of the variation it follows that

$$\delta y(a) = \delta y(b) = 0 . \tag{2.2.11}$$

Thus, the necessary condition for the extremum of J reduces to

$$\delta J = \int_a^b \left[\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right)\right] \delta y dx = 0\,. \tag{2.2.12}$$

Finally, since δy is arbitrary, we conclude that the coefficient of δy in Eq. (2.2.12) must vanish identically over the interval of integration. Therefore, if $y(x)$ is to minimize (or maximize) J, it must satisfy the following condition, known as the Euler-Lagrange equation,

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0\,. \tag{2.2.13}$$

If the value of the unknown function is not specified at either or both ends, then the variation of $y(x)$ need not vanish at those points. However, the first term on the right hand side of Eq. (2.2.10) must still vanish independently, in order for the relation to hold. That is if $y(x)$ is not prescribed at the end points the following conditions, often called the *natural boundary conditions*, must be satisfied.

$$\left[\frac{\partial F}{\partial y'}\right]_{x=a} = 0\,, \qquad \text{and} \qquad \left[\frac{\partial F}{\partial y'}\right]_{x=b} = 0\,. \tag{2.2.14}$$

Example 2.2.1

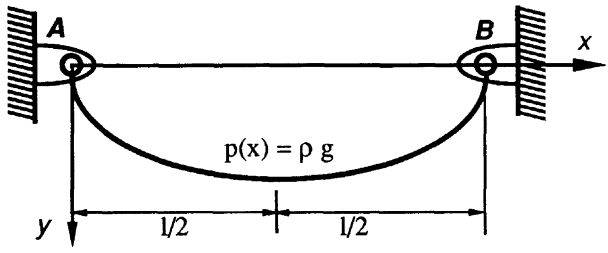


Figure 2.2.1 Supported cable under its own weight.

Consider the problem of determining the equilibrium configuration $y(x)$ of a flexible, constant cross-section cable hanging under its own weight between two points, a distance l apart, as shown in Figure (2.2.1). This is a rather well-known fixed point problem of the calculus of variations.

The cable assumes a position that is consistent with its potential energy being a minimum. Hence, to determine the equilibrium shape $y(x)$ we need to minimize the potential energy functional which can be expressed in terms of the unknown shape function as

$$J = \int \rho g y ds\,, \tag{2.2.15}$$

where ρg is the weight per unit length and ds is an element of arc length of the cable. Relating the arc length to the horizontal coordinate x, with the origin at the center, we rewrite Eq. (2.2.15) as

$$J = \rho g \int_{-l/2}^{l/2} y\sqrt{1+y'^2}dx\,. \tag{2.2.16}$$

At this point one can either take the variation of Eq. (2.2.16) or, since this is a fixed-end-point problem, apply the Euler-Lagrange equation of Eq. (2.2.13) derived previously. The resulting necessary condition for the potential energy to be minimum reduces to the following ordinary differential equation

$$\sqrt{1+y'^2} - \frac{d}{dx}\left(\frac{yy'}{\sqrt{1+y'^2}}\right) = 0\,. \tag{2.2.17}$$

Expanding the second term and rearranging terms, we simplify Eq. (2.2.17) to

$$yy'' - y'^2 - 1 = 0\,. \tag{2.2.18}$$

Introducing $dy/dx = t$ and $d^2y/dx^2 = tdt/dy$, we rewrite Eq. (2.2.18) as

$$\frac{tdt}{t^2+1} = \frac{dy}{y}\,. \tag{2.2.19}$$

Integrating Eq. (2.2.19) once we obtain

$$t = \frac{dy}{dx} = \sqrt{\frac{y^2}{c_1^2} - 1}\,. \tag{2.2.20}$$

Finally, one more integration yields

$$y(x) = c_1 \cosh\left(\frac{x}{c_1} + c_2\right)\,, \quad t = \frac{dy}{dx} = \sinh\left(\frac{x}{c_1} + c_2\right)\,. \tag{2.2.21}$$

The condition

$$\left.\frac{dy}{dx}\right|_0 = 0 \tag{2.2.22}$$

yields $c_2 = 0$, while c_1 can be determined from the condition

$$y(-l/2) = y(l/2) = c_1 \cosh(l/(2c_1))\,. \tag{2.2.23}$$

Equation (2.2.21) is the equation of a catenary. • • •

2.3 Classical Methods for Constrained Problems

Most practical structural optimization problems have constraints on design variables in the form of limits or algebraic relations in terms of these design variables. These constraints may be related to the functional requirements of the design, geometry, availability of the resources, or appearance and esthetic appeal. In this section we will consider problems with equality constraints on the design variables. Although these constraints mentioned above appear most often as inequality constraints, they can be converted into equivalent equality constraints as will be discussed later on.

The general form of the equality constrained problem can be expressed in the following form.

$$\begin{aligned} \textbf{Minimize} \quad & f(\mathbf{x}), \qquad \mathbf{x} = (x_1, \ldots, x_n)^T, \\ \textbf{subject to} \quad & h_j(\mathbf{x}) = 0, \qquad j = 1, \ldots, n_e, \end{aligned} \tag{2.3.1}$$

where the number of independent equality constraints n_e is less than or equal to the number of design variables n. If the number of constraints is larger than the number of design variables, then the problem is over constrained and, in general, there is no solution.

There is more than one approach to solving problems posed in the form of Eq. (2.3.1). If the equality constraints can be solved explicitly for n_e design variables in terms of a set of $n - n_e$ independent design variables, then the objective function can be written in terms of the $n - n_e$ independent design variables. The new objective function will not be subject to any constraints and can be minimized using techniques discussed in the previous section.

For example, for a minimization problem with two design variables subject to a single equality constraint

$$\begin{aligned} \textbf{Minimize} \quad & f(x_1, x_2) \\ \textbf{subject to} \quad & h(x_1, x_2) = 0, \end{aligned} \tag{2.3.2}$$

we can solve for one of the design variables from the constraint relation,

$$x_1 = h_c(x_2), \tag{2.3.3}$$

and substitute into the objective function. The resulting new objective function,

$$f_r(x_2) = f[h_c(x_2), x_2], \tag{2.3.4}$$

can be differentiated with respect to the independent design variable x_2, and df_r/dx_2 can be set to zero to determine the optimum value of the x_2. The optimum value of x_1 can then be obtained from Eq. (2.3.3).

The procedure outlined above is called *variable-elimination* or *direct substitution* method. For problems in which the constraint equations cannot be solved explicitly, for example, when the constraints are defined in terms of integrals, another method called *the method of Lagrange multipliers* is used.

2.3.1 *Method of Lagrange Multipliers*

In essence, the method of Lagrange multipliers in calculus of variations is a direct extension of the method for constrained minimization in differential calculus. We start by reviewing the method as used in differential calculus. For an objective function $f(\mathbf{x})$ of n design variables to be a minimum, the differential change in the objective function must still vanish.

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \ldots\ldots\ldots + \frac{\partial f}{\partial x_n} dx_n = 0 \,. \tag{2.3.5}$$

However, now the derivative terms can not be set to zero individually because the differential changes in the design variables $(dx_1, dx_2, \ldots, dx_n)$ are dependent on one another through the constraint equations.

For simplicity, assume only a single constraint relation $h(\mathbf{x}) = 0$, the differential changes in the design variables are related through

$$dh = \frac{\partial h}{\partial x_1} dx_1 + \frac{\partial h}{\partial x_2} dx_2 + \ldots\ldots\ldots + \frac{\partial h}{\partial x_n} dx_n = 0 \,. \tag{2.3.6}$$

We can multiply Eq. (2.3.6) by an arbitrary (for the time being) constant, λ, and add to the Eq. (2.3.5) to obtain (see [4])

$$\left(\frac{\partial f}{\partial x_1} + \lambda \frac{\partial h}{\partial x_1}\right) dx_1 + \left(\frac{\partial f}{\partial x_2} + \lambda \frac{\partial h}{\partial x_2}\right) dx_2 + \ldots\ldots + \left(\frac{\partial f}{\partial x_n} + \lambda \frac{\partial h}{\partial x_n}\right) dx_n = 0 \,. \tag{2.3.7}$$

Let λ be determined so that the quantities inside each of the parenthesis vanish to satisfy the previous equation. This leads to n equations for the $n + 1$ unknowns, the n design variables, and the unknown multiplier λ called *the Lagrange multiplier.* The constraint relation $h(\mathbf{x}) = 0$ provides the requisite $(n + 1)$th relation. Equations (2.3.7) and (2.3.2) are exactly what one would obtained by an unconstrained minimization of an auxiliary function $f + \lambda h$ with respect to the design variables and the Lagrange multiplier λ.

For multiple constraint functions, one has to introduce a *Lagrange multiplier* for each of the constraint functions. Therefore, in general an optimization problem with an objective function with n design variables plus n_e equality constraints stated in Eq. (2.3.1) is equivalent to an unconstrained problem with an auxiliary function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^{n_e} \lambda_j h_j \,. \tag{2.3.8}$$

The optimum values of the design variables can be obtained by solving a system of $n + n_e$ equations

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_i} &= 0 \,, \qquad i = 1, \ldots, n \,, \\ \frac{\partial \mathcal{L}}{\partial \lambda_j} &= 0 \,, \qquad j = 1, \ldots, n_e \,, \end{aligned} \tag{2.3.9}$$

for $n + n_e$ unknowns.

Example 2.3.1

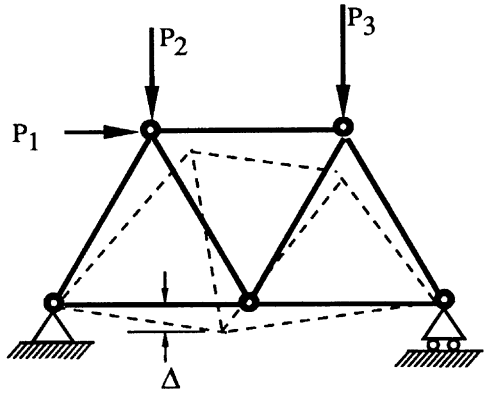


Figure 2.3.1 Design of trusses with displacement constraint.

Consider a general truss structure with n members under concentrated loads acting at the junctions of the members. The objective of the design is to minimize the total volume of material used for the members while specifying the displacement Δ in a given direction, and at a given point of the truss. In general, displacement constraints at more than one location can be imposed, but for simplicity only one displacement constraint is considered. Since the overall topology of the structure is fixed, the only design variables are the cross-sectional areas A_i, $(i = 1, \ldots, n)$ of the members.

In order to pose the problem as a constrained minimization problem, we need to express the displacement constraint in terms of the design variables. The dummy-load method (*Method of Virtual Load*), which is a special case of the principle of virtual forces or the principle of complementary virtual work, is used for this purpose. The principle of complementary virtual work states that *the strains and displacements in a deformable body are compatible and consistent with the support conditions if and only if the total complementary virtual work is zero* [5]

$$\delta W_I^* + \delta W_E^* = 0\,. \tag{2.3.10}$$

Here δW_I^* is the internal complementary virtual work and δW_E^* is the complementary virtual work of the external forces. The dummy-load method starts by applying a unit virtual load at the point of unknown displacement along the displacement component of interest. The internal complementary virtual work under this loading can be expressed as

$$\delta W_I^* = -\delta U^* = -\int_V \delta\sigma_{ij}\epsilon_{ij} dV\,, \tag{2.3.11}$$

where δU^* is the complementary strain energy, ϵ_{ij} is the strain field under the actual loads, and $\delta\sigma_{ij}$ is the virtual stress field due to the dummy-load. In absence of the

body forces the external complementary virtual work is

$$\delta W_E^* = \int_S \hat{u}_i \delta t_i dS \,, \tag{2.3.12}$$

where $\hat{u}_i$ are the components of the surface displacements and δt_i are the components of the applied virtual tractions. For a two dimensional truss structure with n constant-cross section members, Eqs. (2.3.10), (2.3.11), and (2.3.12) yield

$$\Delta \times 1 = \sum_{i=1}^{n} \delta\sigma_i \epsilon_i L_i A_i \,, \tag{2.3.13}$$

where L_i is the length of the ith member, ϵ_i is the strain due to the actual loads, and $\delta\sigma_i$ is the dummy stress in the ith member. Relating the stresses and strains to the design variables, we can rewrite Eq (2.3.13) as

$$\Delta = \sum_{i=1}^{n} \frac{f_i F_i}{A_i E_i} L_i \,, \tag{2.3.14}$$

where f_i and F_i are the dummy and actual internal forces in the ith member, respectively, and E_i is the elastic modulus of the ith member.

We can now formulate the design problem in the standard form of Eq. (2.3.1) as

$$\begin{aligned} \textbf{Minimize} \quad & V(\mathbf{A}) = \sum_{i=1}^{n} A_i L_i \\ \textbf{subject to} \quad & \sum_{i=1}^{n} \frac{f_i F_i}{A_i E_i} L_i - \Delta = 0 \,. \end{aligned} \tag{2.3.15}$$

Introducing the Lagrange multiplier, we write the auxiliary objective function as

$$\mathcal{L}(\mathbf{A}, \lambda) = \sum_{i=1}^{n} A_i L_i + \lambda \left(\sum_{i=1}^{n} \frac{f_i F_i}{A_i E_i} L_i - \Delta \right) . \tag{2.3.16}$$

Then the necessary conditions for extremum are given by the following set of equations.

$$\frac{\partial \mathcal{L}}{\partial A_i} = L_i - \lambda \frac{f_i F_i}{A_i^2 E_i} L_i = 0 \,, \tag{2.3.17}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{i=1}^{n} \frac{f_i F_i}{A_i E_i} L_i - \Delta = 0 \,. \tag{2.3.18}$$

Solving for the cross-sectional areas from Eq. (2.3.17) in terms of the Lagrange multiplier and substituting back into Eq. (2.3.18), we can determine the value of the Lagrange multiplier in terms of the specified displacement Δ by

$$\lambda^{\frac{1}{2}} = \frac{1}{\Delta} \sum_{i=1}^{n} \left[\frac{f_i F_i}{E_i} \right]^{\frac{1}{2}} L_i \,. \tag{2.3.19}$$

Then, the optimum values of the cross-sectional areas are

$$A_i^* = \frac{1}{\Delta}\left[\sum_{j=1}^{n}\left(\frac{f_j F_j}{E_j}\right)^{\frac{1}{2}} L_j\right]\sqrt{\frac{f_i F_i}{E_i}}. \tag{2.3.20}$$

Note that the term inside the square brackets is a constant. We determine the corresponding total volume of material by substituting Eq. (2.3.20) into the objective function to obtain

$$V^* = \frac{1}{\Delta}\left[\sum_{j=1}^{n}\left(\frac{f_j F_j}{E_j}\right)^{\frac{1}{2}} L_j\right]^2. \tag{2.3.21}$$

• • •

2.3.2 *Function Subjected to an Integral Constraint*

For problems in which the unknown design variables are functions constrained by functionals, variational calculus also employs Lagrange multipliers. Recall that for the supported cable problem the Euler-Lagrange equation was obtained by allowing the variation of the cable shape function δy to be arbitrary, or in other words by allowing $y(x)$ to be completely unconstrained except for the kinematic boundary conditions. However, if the function $y(x)$ is required to satisfy a subsidiary integral constraint of the form

$$\int_a^b g[y(x)]dx = c, \tag{2.3.22}$$

then the extremum of the functional $J[y(x)]$ can be determined by the use of the Lagrange multiplier technique. In this case the necessary condition for an extremum is the vanishing of the first variation of an auxiliary functional

$$\mathcal{L} = J + \lambda\left[\int_a^b g[y(x)]dx - c\right]. \tag{2.3.23}$$

In the following example we illustrate the use of this technique for determination of the cross-sectional area distribution of minimum weight beams for a specified displacement at a point along the span.

Example 2.3.2

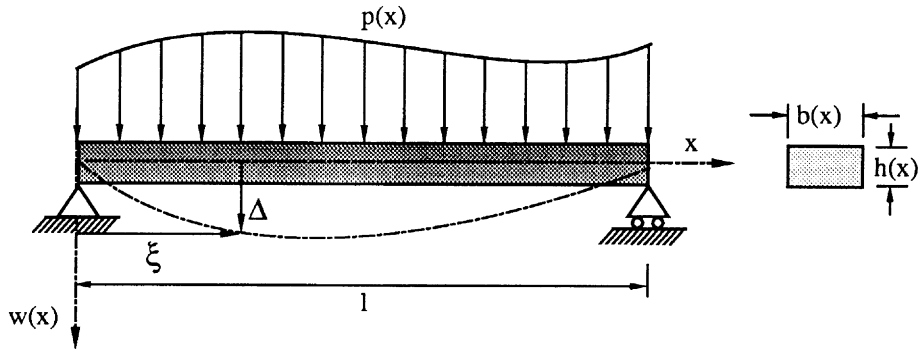


Figure 2.3.2 Design of beams for a specified displacement [6].

Consider a statically determinate beam of variable cross section $A(x)$ loaded by a concentrated and/or distributed loads and moments which produce a moment distribution $M(x)$ along the beam. We want to minimize the volume V of the beam subject to the requirement that the displacement at a point $x = \xi$ is equal to a specified value Δ [6]. This problem, studied by Barnett [6], is formulated as

$$\begin{aligned} \textbf{Minimize} \quad & V = \int_0^l A(x)dx \\ \textbf{subject to} \quad & w(\xi) - \Delta = 0\,. \end{aligned} \tag{2.3.24}$$

A convenient expression for the displacement of a beam at a point $x = \xi$ is obtained again by using the method of virtual load discussed in the previous example problem. That is

$$w(\xi) = \int_0^l \frac{M(x)m(x)}{EI(x)}dx\,, \tag{2.3.25}$$

where $m(x)$ is the moment distribution generated by a unit load applied at $x = \xi$, E is the elastic modulus of the beam material, and $I(x)$ is the cross-sectional moment of inertia. Since the cross-sectional area distribution function of the beam is the design variable, the moment of inertia term has to be expressed in terms of the area. Commonly, the beam moment of inertia function is related to the cross-sectional area function as

$$I(x) = \alpha[A(x)]^n\,, \tag{2.3.26}$$

where α is a constant related to some physical dimension of the cross-section, and n is a constant that depends on the physical relation between the two functions.

Here we limit the constant n to the integer values of 1,2, or 3. The case of $n = 1$ is for a rectangular cross-section beam of constant depth whose width varies along the length. Such a beam is sometimes referred to as a plane-tapered beam. The case $n = 2$ is obtained when both the width and the depth of the cross-section vary without changing its aspect ratio, and finally the case $n = 3$ is for a cross-section with a variable depth and a constant width. The latter may be referred to as the depth-tapered beam.

The auxiliary functional for the minimization problem, Eq. (2.3.24) takes the following form.

$$\mathcal{L} = \int_0^l A(x)dx + \lambda \left[\int_0^l \frac{M(x)m(x)}{EI(x)} dx \; - \Delta \right] . \tag{2.3.27}$$

The necessary condition for the constrained minimum is the vanishing of the first variation of this auxiliary functional. At this point we set $n = 1$ in order to simplify the following derivation. The first variation of Eq. (2.3.27) becomes

$$\delta\mathcal{L} = \int_0^l \left[1 - \lambda \frac{M(x)m(x)}{\alpha E A^2(x)} \right] \delta A dx = 0 . \tag{2.3.28}$$

The corresponding Euler-Lagrange equation is

$$1 - \lambda \frac{M(x)m(x)}{\alpha E A^2(x)} = 0 , \qquad \text{or} \qquad A(x) = \lambda^{\frac{1}{2}} \left(\frac{Mm}{\alpha E} \right)^{\frac{1}{2}} . \tag{2.3.29}$$

The unknown Lagrange multiplier in Eq. (2.3.29) must be determined from the displacement constraint in Eq. (2.3.24). That is, using Eqs. (2.3.25), (2.3.26), and (2.3.29) in Eq. (2.3.24) we can extract

$$\lambda^{\frac{1}{2}} = \frac{1}{\Delta} \int_0^l \left(\frac{Mm}{\alpha E} \right)^{\frac{1}{2}} dx . \tag{2.3.30}$$

Then, the optimal area distribution and the corresponding volume are given by

$$A^*(x) = \frac{1}{\alpha E \Delta} \left[\int_0^l \Big(M(\eta)m(\eta) \Big)^{\frac{1}{2}} d\eta \right] \Big(M(x)m(x) \Big)^{\frac{1}{2}} , \tag{2.3.31}$$

and

$$V^* = \frac{1}{\alpha E \Delta} \left[\int_0^l \Big(M(\eta)m(\eta) \Big)^{\frac{1}{2}} d\eta \right]^2 , \tag{2.3.32}$$

respectively. • • •

2.3.3 *Finite Subsidiary Conditions*

The problems discussed in the previous section involve a rather simple integral constraint that require a constant Lagrange multiplier in the auxiliary functional. In a more general case, as mentioned earlier, we are interested in extremizing functionals of several functions and their derivatives with respect to more than one independent variable [see Eq. (2.2.8)]. In addition, there may be m finite subsidiary constraints of the form

$$h_i\left(x_1,\ldots,x_n,y_1,\ldots,y_p,\frac{\partial y_1}{\partial x_1},\ldots,\frac{\partial y_p}{\partial y_n}\right)=0\,,\qquad i=1,\ldots,m\,, \tag{2.3.33}$$

imposed on the problem. These constraints may range from simple algebraic equations to highly complicated differential equations that must be satisfied at every point over the entire domain of the problem.

The Lagrange multiplier method, in this case, still reduces to extremizing an auxiliary functional of the form

$$\mathcal{L}=\int_v\left(f+\sum_{i=1}^{m}\lambda_i h_i\right)dv\,. \tag{2.3.34}$$

The Lagrange multipliers, however, are no longer constants but functions of the coordinates $x_1,\ldots,x_n$.

Example 2.3.3

The problem described above can be best illustrated by a design example of a cantilever beams of prescribed volume and prescribed loads for minimum deflection. Except for a slight change of notation, this example is based upon Makky and Ghalib's solution [7].

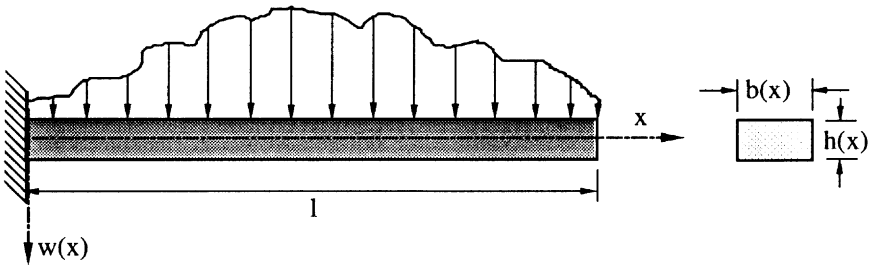


Figure 2.3.3 Optimum Design of a Beam for Minimum Deflection.

Figure 2.3.3 shows an elastic cantilever beam fixed at the end $x = 0$, free at the end $x = l$, and acted upon by a specified distribution of transverse loading $q(x)$ per unit of length. The objective is to minimize some norm of the transverse displacement of the beam for a given total volume, V_0. The norm we choose is the integral of the transverse displacement w over the length of the beam. The loading $q(x)$ is restricted to be unidirectional in order to render the norm appropriate.

The functional to be minimized, in this case, is an integral of the displacement field $w(x)$ which must satisfy the equation of equilibrium of the beam as well as the constraint on the total volume of material. The equation of equilibrium is expressed as

$$[s(x)w'']'' - q(x) = 0\,, \tag{2.3.35}$$

with boundary conditions

$$\text{at } x = 0: \quad w = 0, \quad \text{and} \quad w' = 0. \tag{2.3.36}$$

$$\text{at } x = l: \quad sw'' = 0, \quad \text{and} \quad s'w'' + sw''' = 0\,, \tag{2.3.37}$$

s(x) being the bending stiffness of the beam that can be related, through Eq. (2.3.26), to the cross-sectional area of the beam by

$$s(x) = EI(x) = \alpha E A^n(x)\,, \qquad n = 1, 2, \text{or } 3\,. \tag{2.3.38}$$

In addition to the subsidiary condition of Eq. (2.3.35), we must specify an integral constraint on the total volume, namely

$$\int_0^l A(x)dx = V_0\,. \tag{2.3.39}$$

The auxiliary functional is

$$\mathcal{L}\left(w(x), s(x), A(x), \lambda_1, \lambda_2(x)\right) = \int_0^l w(x)dx + \lambda_1 \left[\int_0^l A(x)dx - V_0\right]$$

$$- \int_0^l \lambda_2(x)[sw'''' + 2s'w''' + s''w'' - q]dx\,, \tag{2.3.40}$$

which must be stationary with respect to the functions $w(x), s(x), A(x), \lambda_2(x)$, and the parameter λ_1. We note, however, that $A(x)$ depends on $s(x)$ through Eq. (2.3.38). Hence,

$$\delta A = \left(\frac{dA}{ds}\right)\delta s\,. \tag{2.3.41}$$

The first variation of $\mathcal{L}$

$$\delta\mathcal{L} = \int_0^l \delta w dx + \lambda_1 \left[\int_0^l \delta A dx\right] + \delta\lambda_1 \left[\int_0^l A dx - V_0\right]$$

$$-\int_0^l \delta\lambda_2(x)[sw'''' + 2s'w''' + s''w'' - q]dx \tag{2.3.42}$$

$$-\int_0^l \lambda_2(x)[\delta sw'''' + s\delta w'''' + 2\delta s'w''' + s'\delta w''' + \delta s''w'' + s''\delta w'']dx = 0\,,$$

can be simplified by several integrations by parts. Collecting the terms multiplied by arbitrary variations $\delta w, \delta s, \delta\lambda_1$, and $\delta\lambda_2$ and equating them to zero independently we obtain the following Euler-Lagrange equations

$$\delta w: \quad 1 - (\lambda_2''s)'' = 0\,, \tag{2.3.43}$$

$$\delta s: \quad \lambda_1\frac{dA}{ds} - \lambda_2''w'' = 0\,, \tag{2.3.44}$$

$$\delta\lambda_1: \quad \int_0^l A(x)dx - V_0 = 0\,, \tag{2.3.45}$$

$$\delta\lambda_2: \quad sw'''' + 2s'w''' + s''w'' - q(x) = 0\,, \tag{2.3.46}$$

together with the associated boundary conditions at $x = 0$ and $x = l$

Either	*Or*	
$\delta s = 0\,,$	$\lambda_2 w''' - \lambda_2' w'' = 0\,,$	(2.3.47)
$\delta s' = 0\,,$	$\lambda_2 w'' = 0\,,$	(2.3.48)
$\delta w = 0\,,$	$\lambda_2''' s + \lambda_2'' s' = 0\,,$	(2.3.49)
$\delta w' = 0\,,$	$\lambda_2'' s = 0\,,$	(2.3.50)
$\delta w'' = 0\,,$	$-\lambda_2 s' + \lambda_2' s = 0\,,$	(2.3.51)
$\delta w''' = 0\,,$	$\lambda_2 s = 0\,.$	(2.3.52)

Equations (2.3.43) through (2.3.46) together with the associated boundary conditions are general enough that they apply to simply supported as well as to clamped beams. For the cantilever beam the boundary conditions are Eqs. (2.3.36) and (2.3.37). Since the bending moment and the shear force at $x = 0$ cannot vanish because of the unidirectional nature of the applied loading, the above conditions reduce to

$$\lambda_2(0) = 0\,, \qquad \lambda_2'(0) = 0\,, \tag{2.3.53}$$

$$\lambda_2''(l)s(l) = 0\,, \qquad \lambda_2'''(l)s(l) + \lambda_2''s'(l) = 0\,. \tag{2.3.54}$$

We can integrate Eqs. (2.3.43) and (2.3.46) twice and make use of both boundary conditions of Eqs. (2.3.37) and (2.3.54) to get

$$s\lambda_2'' = \frac{1}{2}(x - l)^2\,, \tag{2.3.55}$$

and

$$sw'' = \int_l^x \left(\int_l^r q(\xi) d\xi \right) dr \equiv p(x) , \tag{2.3.56}$$

from which

$$\lambda_2'' w'' = \frac{1}{2} \frac{(x-l)^2 p(x)}{(s(x))^2} . \tag{2.3.57}$$

Combining the last equation with the second Euler-Lagrange equation (2.3.44), we obtain

$$s^2(x) \frac{dA}{ds} = \frac{(x-l)^2 p(x)}{2\lambda_1} . \tag{2.3.58}$$

Specialization to Plane-Tapered Beams. The remainder of this problem will be specialized to plane-tapered beams, $n = 1$, under a uniformly distributed load of intensity $q(x) = q_0$. Evaluating the distribution of $p(x)$, we find that Eq. (2.3.56) becomes

$$sw'' = q_0 \frac{(l-x)^2}{2} . \tag{2.3.59}$$

Also, for a plane-tapered beam

$$\frac{dA}{ds} = \frac{1}{\alpha E} = c^2 = constant . \tag{2.3.60}$$

Hence, Eq. (2.3.58) becomes

$$s(x) = \frac{(x-l)^2}{2c} \sqrt{\frac{q_0}{\lambda_1}} , \tag{2.3.61}$$

and, therefore, the optimum distribution of the cross-sectional area is

$$A^*(x) = \frac{c(x-l)^2}{2} \sqrt{\frac{q_0}{\lambda_1}} . \tag{2.3.62}$$

The unknown Lagrange multiplier can be evaluated from the volume constraint of Eq. (2.3.45) to be

$$\lambda_1 = \frac{c^2 q_0 l^6}{36 V_0^2} . \tag{2.3.63}$$

The resulting optimal area and bending stiffness distributions are

$$A^*(x) = \frac{3V_0(x-l)^2}{l^3} , \quad \text{and} \quad s^*(x) = \frac{3V_0(x-l)^2}{c^2 l^3} . \tag{2.3.64}$$

Substituting Eq. (2.3.64) into Eq. (2.3.59) and integrating it twice, we obtain the deflection function corresponding to the optimal beam as

$$w(x) = \frac{c^2 q_0 l^3}{12 V_0} x^2 , \tag{2.3.65}$$

where the boundary conditions in Eq. (2.3.36) were used. The constant c for a rectangular plane-tapered beam with constant thickness h and varying width $b(x)$ is

$$c^2 = \frac{12}{Eh^2}\,. \tag{2.3.66}$$

The resulting deflection function is

$$w(x) = \frac{q_0 l^3 x^2}{EV_0 h^2}\,. \tag{2.3.67}$$

For comparison, consider an equivalent uniform beam of the same total volume V_0, length l, constant thickness h, but a constant width

$$b_0 = \frac{V_0}{hl}\,. \tag{2.3.68}$$

It is easy to verify that its deflection $w_0(x)$ satisfies

$$\frac{\int_0^l w(x)dx}{\int_0^l w_0(x)dx} = \frac{5}{9}\,. \tag{2.3.69}$$

That is, the optimal beam is 1.8 times stiffer than the uniform beam of the same volume.

Several other cases of loading with different types of beams, $n = 1, 2$, and 3 may be found in Ref. 7. Some of these cases form a part of the exercises at the end of the chapter. • • •

2.4 Local Constraints and the Minmax Approach

In many problems of structural optimization we have constraints that are local in nature, such as stress constraints. In a beam design problem, for example, we may require that the stresses do not exceed the yield limit anywhere in the beam. Such constraints can be expressed as subsidiary conditions similar to Eq. (2.3.33), except that the equalities are replaced by inequalities

$$g_i\left(x_1, \ldots, x_n, y_1, \ldots, y_p, \frac{\partial y_1}{\partial x_1}, \ldots, \frac{\partial y_p}{\partial y_n}\right) \geq 0\,, \qquad i = 1, \ldots, m\,. \tag{2.4.1}$$

We can transform the inequalities back to equalities by subtracting slack functions, t_i's, and rewrite Eq. (2.4.1) as

$$g_i\left(x_1, \ldots, \frac{\partial y_p}{\partial y_n}\right) - t_i^2\,(x_1, \ldots, x_n) = 0\,, \qquad i = 1, \ldots, m\,. \tag{2.4.2}$$

The auxiliary functional of Eq. (2.3.34) becomes

$$\mathcal{L} = \int_v \left[f + \sum_{i=1}^{m} \lambda_i (g_i - t_i^2) \right] dv \,. \tag{2.4.3}$$

When we take the variation of $\mathcal{L}$, the variation of t_i will contribute $-2 \int_v \lambda_i t_i \delta t_i dv$. Setting the coefficient of δt_i to zero we get

$$t_i \lambda_i = 0 \,, \qquad i = 1, \ldots, m \,. \tag{2.4.4}$$

This equation implies that the Lagrange multipliers are equal to zero when the slack variables are not zero. That is, the Lagrange multipliers are zero at points in the design space where the corresponding constraint is not critical. Equation (2.4.4) may also be written as

$$\lambda_i g_i = 0 \,, \qquad i = 1, \ldots, m \,, \tag{2.4.5}$$

because $t_i = 0$ if and only if $g_i = 0$. It can be shown that if we use Eq. (2.4.5), which is called a *constraint qualification* equation, we can dispense with the slack functions in the auxiliary functional. When we do that, we also dispense with the variation of the auxiliary functional with respect to the Lagrange multiplier, and instead add the inequality constraints to the optimality conditions. This treatment of inequality constraints is demonstrated in the following example.

Example 2.4.1

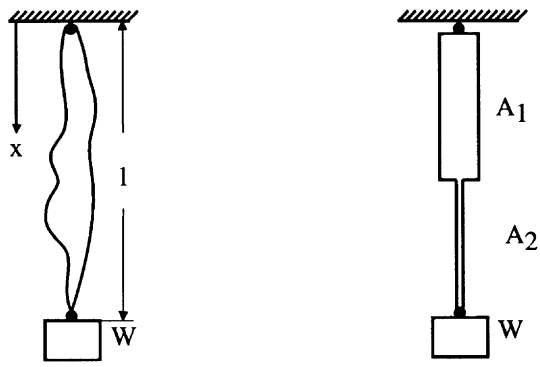


Figure 2.4.1 Hanging cable: (a) general cross section; and (b) two constant-area segments.

The cable in Figure 2.4.1(a) is loaded by a hanging weight W plus its own self-weight. The cross-sectional area $A(x)$ of the cable is to be designed for minimum volume, subject to the constraint that the stress does not exceed an allowable value of σ_0, and the cross-sectional area is not less than a minimum, A_0. We assume that

the load W is small enough to be supported by a minimum-area cable if the selfweight is neglected. That is

$$W \leq A\sigma_0 \,. \tag{2.4.6}$$

We also assume that the cable is long enough so that selfweight requires the top part of the cable to have a cross-sectional area larger than A_0.

The problem is statically determinate, with the axial load in the cable satisfying

$$P' + \rho A = 0 \,, \qquad P(l) = W \,, \tag{2.4.7}$$

where ρ is the weight density. The problem can then be formulated as

$$\begin{aligned} \textbf{minimize} \quad & \int_0^l A(x)dx \\ \textbf{such that} \quad & A(x)\sigma_0 - P(x) \geq 0 \,, \\ & A - A_0 \geq 0 \,, \\ \textbf{and} \quad & P' + \rho A = 0 \,. \end{aligned} \tag{2.4.8}$$

The Lagrangian functional is given as

$$\begin{aligned} \mathcal{L}(A(x), P(x), \lambda_1(x), \lambda_2(x), \lambda_3(x)) = & \int_0^l A dx + \int_0^l \lambda_1 (A\sigma_0 - P) dx \\ & + \int_0^l \lambda_2 (A - A_0) dx + \int_0^l \lambda_3 (P' + \rho A) dx \,. \end{aligned} \tag{2.4.9}$$

We take the variation of $\mathcal{L}$ to obtain

$$\delta\mathcal{L} = \int_0^l \delta A dx + \int_0^l \lambda_1 (\delta A \sigma_0 - \delta P) dx + \int_0^l \lambda_2 \delta A dx + \int_0^l \lambda_3 (\delta P' + \rho \delta A) dx \,. \tag{2.4.10}$$

We integrate the term including $\delta P'$ by parts to convert it to δP, and then set the coefficients of δA and δP to zero to obtain

$$1 + \lambda_1 \sigma_0 + \lambda_2 + \rho \lambda_3 = 0 \,, \tag{2.4.11}$$

$$\lambda_1 + \lambda_3' = 0 \,, \qquad \lambda_3(0) = 0 \,. \tag{2.4.12}$$

These equations are augmented with the two inequalities

$$A\sigma_0 - P \geq 0 \,, \tag{2.4.13}$$

$$A - A_0 \geq 0 \,, \tag{2.4.14}$$

the constraint qualification equations

$$\lambda_1 (A\sigma_0 - P) = 0 \,, \tag{2.4.15}$$

$$\lambda_2 (A - A_0) = 0 \,. \tag{2.4.16}$$

and with Eq. (2.4.7).

To solve the above equations we note that near $x = 0$ we assumed that $A > A_0$, so that from Eq. (2.4.16) $\lambda_2 = 0$. We can substitute λ_1 from Eq. (2.4.12) into Eq. (2.4.11) to get

$$1 - \lambda_3'\sigma_0 + \rho\lambda_3 = 0\,, \qquad \lambda_3(0) = 0\,. \tag{2.4.17}$$

This equation is easily solved to yield

$$\lambda_3 = (e^{\rho x/\sigma_0} - 1)/\rho\,, \tag{2.4.18}$$

and then from Eq. (2.4.12)

$$\lambda_1 = -(1/\sigma_0)e^{\rho x/\sigma_0}\,. \tag{2.4.19}$$

These two equations are valid as long as $A > A_0$. From Eq. (2.4.19) we see that λ_1 is nonzero, so that from Eq. (2.4.15)

$$A(x) = P(x)/\sigma_0 \qquad \text{when} \quad A > A_0\,. \tag{2.4.20}$$

We can now construct the entire solution for $A(x)$. At the bottom of the cable $A = A_0$, and from Eq. (2.4.7)

$$P = W + \rho(l - x)A_0\,. \tag{2.4.21}$$

This solution becomes invalid when P exceeds $A_0\sigma_0$, which from Eq. (2.4.21) happens at $x = x_t$,

$$x_t = l - \frac{A\sigma_0 - W}{\rho A_0}\,. \tag{2.4.22}$$

For $x < x_t$ we have $A > A_0$, so that $P = A\sigma_0$, and Eq. (2.4.7) can be replaced by

$$A'\sigma_0 + \rho A = 0\,, \qquad A(x_t) = A_0\,. \tag{2.4.23}$$

This equation is easily solved to yield

$$A(x) = A_0 e^{\rho(x_t - x)/\sigma_0}\,, \qquad x < x_t\,. \tag{2.4.24}$$

• • •

Another formulation of the problem in Example 2.4.1 is to find a cable with a given volume that has the lowest possible stress. The objective function is

$$\min_{A(x)} \max_{0 \le x \le l} \sigma(x)\,. \tag{2.4.25}$$

This is an example of the so-called min-max problems that are common in structural optimization. Min-max problems present a difficulty in that the maximum of a function does not have continuous derivatives. This can be seen by considering even the simplest case of the maximum of a function at two points. Consider, for example, the case when the cross-sectional area of the cable has to be piecewise constant to keep down manufacturing cost. Figure 2.4.1(b) shows a case where the number of segments is limited to two, and the design variables are the two cross-sectional areas A_1 and A_2.

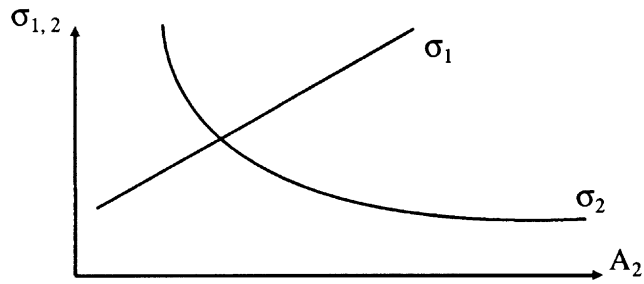


Figure 2.4.2 Discontinuity of maximum function.

Figure 2.4.2 shows a possible variation of the maximum stresses at each segment as a function of A_2. Increasing A_2 reduces the maximum stress in the bottom segment, but increases it in the top segment. It is seen that the maximum of the stress over the beam has a discontinuity in slope at the point where the the location of the maximum jumps from one segment to the other.

For the cable of Example 2.4.1, the two formulations of minimizing volume or minimizing the maximum stress for a given volume V_0 are equivalent. That is, we can guess a stress allowable and minimize the volume for this stress allowable. If the resulting optimal volume is larger than V_0 we increase the stress allowable and repeat the optimization. Similarly, if the optimal volume is smaller than V_0 we reduce the stress allowable and reoptimize. However, for many problems it is not possible to find an equivalent formulation that does not involve the minimum of a maximum. For example, when we optimize the shape of a hole so as to reduce stress concentration, we often do not have any constraint on volume. Therefore, we cannot transform the problem to one of minimizing volume with a constraint on stress. Taylor and Bendsøe [8] suggested an elegant solution to the problem of discontinuous derivatives. They suggest replacing the objective function of Eq. (2.4.25) with another objective plus a constraint equation

$$\begin{aligned} &\min_{A(x),\beta} \quad \beta \\ &\text{such that} \quad \sigma(x) \leq \beta\,, \qquad 0 \leq x \leq l\,. \end{aligned} \tag{2.4.26}$$

The additional design variable β is the unknown stress limit that we want to keep as low as possible. Now the objective function is equal to one of the design variables, so that it is perfectly smooth. This tactic for converting a min-max problem to a smooth problem by using an additional variable is very useful in many applications. We demonstrate it for the cable problem of Example 2.4.1.

Example 2.4.2

Formulate the problem of designing the cable of Figure 2.4.1(a) for minimum maximum stress, subject to a limit V_0 on the available volume, and a lower limit A_0 on the cross-sectional area.

The problem is first formulated as

$$\begin{aligned} \textbf{minimize} \quad & \max_{0 \le x \le l} \sigma(x) \\ \textbf{such that} \quad & A - A_0 \ge 0\,, \\ & \int_0^l A(x)dx = V_0\,, \\ \textbf{and} \quad & P' + \rho A = 0\,. \end{aligned} \tag{2.4.27}$$

Note that we have formulated the volume constraint as an equality rather than an inequality constraint, because common sense tells us that the volume will be fully utilized in order to minimize the stress. Next we replace the min-max formulation with the Taylor-Bendsøe 'beta' formulation

$$\begin{aligned} \textbf{minimize} \quad & \beta \\ \textbf{such that} \quad & A(x)\beta - P(x) \ge 0\,, \\ & A - A_0 \ge 0\,, \\ & \int_0^l A(x)dx = V_0\,, \\ \textbf{and} \quad & P' + \rho A = 0\,. \end{aligned} \tag{2.4.28}$$

The solution of this problem is left as an exercise to the reader (Exercise 4).• • •

2.5 Necessary and Sufficient Conditions for Optimality

In the absence of inequality constraints, the Euler-Lagrange equations provide the necessary conditions for optimality (Optimality Conditions). As opposed to the use of Differential Calculus it is, however, not an easy task to verify whether such necessary conditions are also sufficient conditions for optimality.

Sufficiency of optimality conditions can sometimes be established on the basis of the variational principles of continuum mechanics. For a discretized model, using the techniques of mathematical programming for optimization (Chapter 5), we can establish sufficiency of the optimality conditions on the basis of convexity of the objective function and the constraints. In general, however, establishing convexity of the objective function and the constraints is again not an easy task. A vast majority of the optimization problems are non-convex.

Thus, establishing the sufficiency of the optimality conditions or identifying local and global optima is a question that cannot be answered fully for most situations. Often we have to rely either upon engineering intuition or upon trial and error techniques for answering such questions.

Using the variational principles of mechanics, we illustrate how the sufficiency of the optimality conditions can be established for a select class of classical optimization problems.

2.5.1 *Elastic Structures Maximum Stiffness*

The development in this section is based on the work of Prager and his collaborators (see Refs. 9,10).

Consider an elastic structure being acted upon by a load $2P$ at a point X. Assume the load is such that it produces a unit displacement in its direction. Then by the principle of conservation of energy [11]

External work = Internal energy stored,

or

$$\frac{1}{2}(2P \times 1) = P = \int_v s(X)e[Q(X)]dv\,, \tag{2.5.1}$$

where $e[Q(X)]$ is the specific elastic strain energy or the strain energy in a structure of unit stiffness due to a strain field $Q(X)$ produced by the prescribed unit displacement at X, and $s(X)$ is the specific stiffness of the structure at X. That is, $s(X)$ is the stiffness per unit length for one-dimensional structures and stiffness per unit of area for two-dimensional structures.

Thus $s(X)$ specifies the design of the structure while the function $e[Q(X)]$ is independent of design parameters. For instance $s(X)$ and $e[Q(X)]$ for a one-dimensional beam element would be $EI(x)$ and $1/2(\text{curvature})^2$, respectively.

We wish to design a structure of a given total stiffness so as to maximize the magnitude P of the load producing the prescribed unit displacement at X. From Eq. (2.5.1) it is clear that maximizing P subject to the integral constraint on specific stiffness

$$\int_v s(X)dv = s_0\,, \tag{2.5.2}$$

can be performed by seeking a stationary point of the auxiliary functional

$$\mathcal{L} = \int_v s(X)e[Q(X)]dv - \lambda\left[\int_v s(X)dv - s_0\right]\,. \tag{2.5.3}$$

The necessary condition for $\mathcal{L}$ to be stationary is given by

$$\delta\mathcal{L} = 0 = \int_v \left(e[Q(X)] + s(X)\frac{\partial e}{\partial s}\right)\delta s dv - \lambda\left(\int_v \delta s dv\right)\,. \tag{2.5.4}$$

Since the structure is required to satisfy the equations of equilibrium for every structural design, then by the principle of minimum strain energy (which is a special

case of the principle of minimum potential energy for prescribed displacements) the second term within the first integral vanishes yielding

$$\int_v (e[Q(X)] - \lambda)\, \delta s dv = 0\,. \tag{2.5.5}$$

Thus for arbitrary variation δs we have

$$e[Q(X)] = \lambda = \text{constant}\,. \tag{2.5.6}$$

Equation (2.5.6) is the necessary condition for optimality. That is, the the stiffness of an elastic structure is stationary for a given structural design if the specific elastic strain energy is constant throughout the structure. We wish to examine if it is also sufficient. To answer this question we assume two distinct designs s and $\bar{s}$ with associated specific strain energies $e[Q(X)]$ and $e[\bar{Q}(X)]$, both satisfying the constant total stiffness constraint

$$\int_v s(X)dv = \int_v \bar{s}(X)dv = s_0\,. \tag{2.5.7}$$

The loads P and $\bar{P}$ that correspond to s and $\bar{s}$ are

$$P = \int_v s(X)e[Q(X)]dv\,, \quad \text{and} \quad \bar{P} = \int_v \bar{s}(X)e[\bar{Q}(X)]dv\,, \tag{2.5.8}$$

respectively. Subtracting $\bar{P}$ from P, we have

$$P - \bar{P} = \int_v s(X)e[Q(X)]dv - \int_v \bar{s}(X)e[\bar{Q}(X)]dv\,. \tag{2.5.9}$$

Since $Q(X)$ is also a kinematically admissible strain field for the design s, if we replace $\bar{Q}(x)$ in the definition of $\bar{P}$ with $Q(X)$ we are guaranteed by the principle of minimum potential energy that

$$\int_v \bar{s}(X)e[\bar{Q}(X)]dv \leq \int_v \bar{s}(X)e[Q(X)]dv\,. \tag{2.5.10}$$

Thus

$$P - \bar{P} \geq \int_v s(X)e[Q(X)]dv - \int_v \bar{s}(X)e[Q(X)]dv\,. \tag{2.5.11}$$

If the design s satisfies the optimality condition, Eq. (2.5.6), then

$$P - \bar{P} \geq \lambda \int_v [s(X) - \bar{s}(X)]dv\,. \tag{2.5.12}$$

Finally, in view of Eq. (2.5.7) we have

$$P - \bar{P} \geq 0\,, \quad \text{or} \quad P \geq \bar{P}\,. \tag{2.5.13}$$

This implies that condition (2.5.6) is not only a necessary but also a sufficient condition for optimality.

2.5.2 *Optimal Design of Euler-Bernoulli Columns*

We consider the problem of maximizing the buckling load of an Euler-Bernoulli column of a given volume or weight with cross-sections obeying $I(x) = \alpha[A(x)]^n, n = 1, 2,$ or 3. It is well known that the buckling load of a structure is given by the minimum value of the Rayleigh quotient over all kinematically admissible displacement fields [11]. For an optimum column we want to maximize this minimum value by varying the distribution of material along the length of the column. Hence the present problem can be posed as one of maximizing the minimum value of the Rayleigh quotient for the buckling load

$$p = \max_{I(x)} \min_{w(x)} \frac{\int_0^l EI(x)w''^2 dx}{\int_0^l w'^2 dx} = \max_{A(x)} \min_{w(x)} \frac{\int_0^l E\alpha[A(x)]^n w''^2 dx}{\int_0^l w'^2 dx}, \tag{2.5.14}$$

subject to the constant volume constraint

$$\int_0^l A(x)dx = V_0 . \tag{2.5.15}$$

Using the Lagrange multiplier technique, we have

$$\mathcal{L} = \max_{A(x)} \min_{w(x)} \frac{\int_0^l E\alpha[A(x)]^n w''^2 dx}{\int_0^l w'^2 dx} - \lambda \left[\int_0^l A(x)dx - V_0 \right] . \tag{2.5.16}$$

The necessary conditions for stability and optimality can be determined by requiring the first variation of the Lagrangian to vanish, that is

$$\delta\mathcal{L} = \frac{2\int_0^l E\alpha[A(x)]^n w''\delta w'' dx}{\int_0^l w'^2 dx} - \frac{2\int_0^l E\alpha[A(x)]^n w''^2 dx}{\left[\int_0^l w'^2 dx\right]^2} \left[\int_0^l w'\delta w' dx \right]$$

$$+ \frac{\int_0^l nE\alpha[A(x)]^{n-1} w''^2 \delta A dx}{\int_0^l w'^2 dx} - \lambda \left[\int_0^l \delta A(x)dx \right] = 0 . \tag{2.5.17}$$

The terms involving the variations of derivatives of w need to be integrated by parts. After a rearrangement of terms, the coefficients of δw and $\delta w'$ yield the stability equation and the associated boundary conditions for every design $A(x)$ while the coefficient of δA yields the optimality condition.

Stability Equation : $[E\alpha A^n(x)w'']'' + pw'' = 0 .$ (2.5.18)

Boundary Conditions : $\delta w = 0 ,$ or $[E\alpha A^n(x)w'']' + pw' = 0,$ (2.5.19)

$\delta w' = 0 ,$ or $E\alpha A^n(x)w'' = 0 .$ (2.5.20)

Optimality Conditions : $nE\alpha A^{n-1} w''^2 - \lambda \int_0^l w'^2 dx = 0 .$ (2.5.21)

Since the second term in equation (2.5.21) is a constant, the equation can be simplified to

$$A^{n-1}w''^2 = c^2 = \text{constant}\,, \tag{2.5.22}$$

and this can be verified to be a statement of constant strain energy density in the buckled mode shape of the optimum column.

The sufficiency of the optimality condition can be very easily established for the case n=1. For this case Eq. (2.5.22) reduces to

$$w''^2 = c^2\,. \tag{2.5.23}$$

We begin by assuming two distinct designs $A(x)$ and $\bar{A}(x)$ both of which satisfy the constant volume constraint (2.5.15) to yield

$$\int_0^l (A - \bar{A})dx = 0\,. \tag{2.5.24}$$

The corresponding buckling loads p_{cr} and $\bar{p}_{cr}$ with associated buckling modes w and $\bar{w}$ are given by

$$p_{cr} = \frac{\int_0^l E\alpha A w''^2 dx}{\int_0^l w'^2 dx}\,, \qquad \bar{p}_{cr} = \frac{\int_0^l E\alpha \bar{A} \bar{w}''^2 dx}{\int_0^l \bar{w}'^2 dx}\,. \tag{2.5.25}$$

Since the buckling mode w is also kinematically admissible for design $\bar{A}(x)$, by the Rayleigh quotient Eq. (2.5.14), the magnitude of the quantity $\bar{\bar{p}}$ defined by

$$\bar{\bar{p}} = \frac{\int_0^l E\alpha \bar{A} w''^2 dx}{\int_0^l w'^2 dx}\,, \tag{2.5.26}$$

has the property that

$$\bar{\bar{p}} \geq \bar{p}_{cr}. \tag{2.5.27}$$

Subtracting p_{cr} from both sides of Eq. (2.5.27) and rearranging we have

$$p_{cr} - \bar{p}_{cr} \geq p_{cr} - \bar{\bar{p}}\,. \tag{2.5.28}$$

Thus,

$$p_{cr} - \bar{p}_{cr} \geq \frac{\int_0^l E\alpha w''^2 (A - \bar{A})dx}{\int_0^l w'^2 dx}\,. \tag{2.5.29}$$

If the design A(x) satisfies the optimality condition (2.5.23), then by virtue of Eq. (2.5.24)

$$p_{cr} - \bar{p}_{cr} \geq 0, \tag{2.5.30}$$

meaning that of all the designs with different cross-sectional shapes the one that satisfies the optimality condition has the largest value of the critical load, thereby establishing the sufficiency of the optimality condition.

Prager and Taylor [9] provide a similar sufficiency proof for the dual problem namely the case of minimizing the volume or weight of an Euler-Bernoulli column for a given buckling load.

Although it is difficult to prove the sufficiency of the optimality condition for values of n other than 1, explicit solutions for the optimum designs for all classical boundary conditions are well known and are available from Refs. 12-16. Approximate numerical solutions using the finite element displacement models have also been reported by Refs. 17-20 for elastically supported columns with a very general distributed axial loading and for portal frames.

Earlier works, especially those of Tadjbaksh and Keller [13], assumed unimodal behavior and did not allow for discontinuity in the slope and the shear force at places where the area of cross-section vanished. Olhoff and Rasmussen [21] have shown that the design of Tadjbaksh and Keller [13] for the clamped column is non-optimal and have outlined more accurate bimodal numerical solutions with a constraint on the minimum cross-sectional area. Olhoff and Rasmussen identify a threshold value for the minimum area constraint below which the optimum clamped columns exhibit a bimodal behavior. Papers by Masur [22,23], Olhoff [24], and by Plaut, Johnson, and Olhoff [25] outline less approximate and properly formulated multi-modal solutions for the elastically supported columns.

Example 2.5.1

By way of illustration we outline the solution for one of the classical cases here while relegating others to the exercises. Consider maximizing the critical load of a simply-supported column of length l subject to the constant volume constraint, Eq. (2.5.15). An explicit solution to this problem was first outlined in [19]. We begin by listing the governing equations and boundary conditions of the problem.

$$\textbf{Stability Equation}: \quad [E\alpha A^3 w'']'' + pw''. \tag{2.5.31}$$

$$\textbf{Boundary Conditions}: \quad w(0) = w(l) = 0, \tag{2.5.32}$$

$$A^3(0)w''(0) = A^3(l)w''(l) = 0. \tag{2.5.33}$$

$$\textbf{Optimality Conditions}: \quad A^2 w''^2 = c^2, \quad \text{or} \quad w'' = \pm c/A. \tag{2.5.34}$$

A consequence of the boundary condition of (2.5.33) and the optimality condition (2.5.34) is

$$A(0) = A(l) = 0. \tag{2.5.35}$$

Substituting the optimality condition into the stability equation we obtain

$$A''^2 + \frac{\beta^2}{A} = 0, \tag{2.5.36}$$

where

$$\beta^2 = p/E\alpha. \tag{2.5.37}$$

The differential equation of (2.5.36) and the associated boundary conditions can be solved by using a change of variables. Letting

$$A = u^{1/2}, \tag{2.5.38}$$

we can integrate once the differential equation to obtain

$$u' = \pm(c_1 - 4\beta^2 u^{1/2})^{\frac{1}{2}}, \tag{2.5.39}$$

c_1 being a constant of integration. The above equation can be integrated once more giving

$$|x - c_2| = -\int \frac{du}{(c_1 - 4\beta^2 u^{1/2})^{\frac{1}{2}}}. \tag{2.5.40}$$

Using another change of variables with $c_1 - 4\beta^2 u^{1/2} = t$ we can integrate the right-hand-side of this equation once more to give

$$|x - c_2| = \frac{1}{6\beta^4}[(c_1 - 4\beta^2 A)^{\frac{1}{2}}(c_1 + 2\beta^2 A)]. \tag{2.5.41}$$

The two constants of integration, namely c_1 and c_2, can be determined by using the boundary condition given in Eq. (2.5.35) which yields

$$c_2 = \frac{1}{6\beta^4}c_1^{3/2}, \quad \text{and} \quad |l - c_2| = \frac{1}{6\beta^4}c_1^{3/2}. \tag{2.5.42}$$

consequently

$$c_1 = (3l\beta^4)^{2/3}, \quad \text{and} \quad c_2 = l/2. \tag{2.5.43}$$

The optimal value of the cross-sectional area at any point along the length of the column can, therefore, be determined from Eq. (2.5.41).

To determine the critical load parameter β we use the volume constraint

$$\int_0^l A(x)dx = 2\int_0^{l/2} A(x)dx = 2\int_0^{l/2} u^{1/2}dx = 2\int_0^{u(l/2)} u^{1/2}\frac{dx}{du}du = V_0, \tag{2.5.44}$$

or from Eq. (2.5.39)

$$\int_0^l A(x)dx = V_0 = 2\int_0^{u(l/2)} \frac{u^{1/2}}{(c_1 - 4\beta^2 u^{1/2})^{1/2}}du. \tag{2.5.45}$$

The right-hand side of this equation can be integrated to obtain

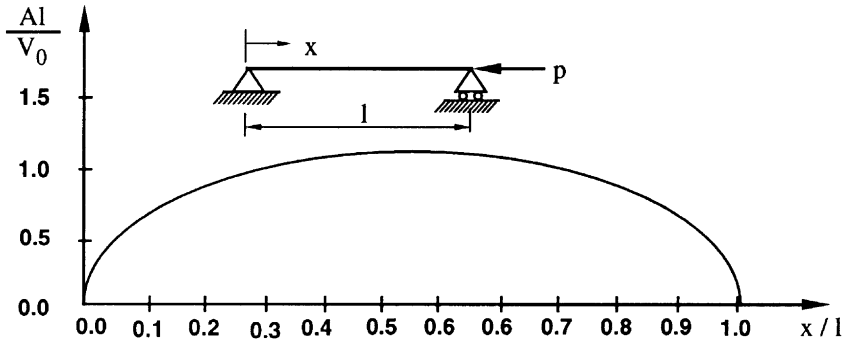
$$V_0 = \frac{(c_1 - 4\beta^2 u^{1/2})^{1/2}}{8\beta^6}\left[c_1^2 - \tfrac{2}{3}c_1(c_1 - 4\beta^2 u^{1/2}) + \tfrac{1}{5}(c_1 - 4\beta^2 u^{1/2})^2\right]_0^{u(l/2)}. \tag{2.5.46}$$

Recalling the definition of u, we can find the value of $u(l/2)$ from Eqs. (2.5.41) and (2.5.43) as

$$u(l/2) = \frac{1}{16\beta^4}(3l\beta^4)^{4/3}. \tag{2.5.47}$$

Substituting Eq. (2.5.47) and the value of the constant c_1 from Eq. (2.5.43) into Eq. (2.5.46) we determine the optimum value of the load parameter and the critical load to be

$$\beta^2_{\text{opt}} = \frac{(15V_0)^3}{243l^5}, \quad \text{and} \quad (p_{cr})_{\text{opt}} = \frac{125}{9}\frac{E\alpha V_0^3}{l^5}. \tag{2.5.48}$$



x / l	0.0 1.0	0.05 0.95	0.10 0.90	0.20 0.80	0.30 0.70	0.40 0.60	0.50
Al/V0	0.0	0.58540	0.78730	1.02345	1.15651	1.22751	1.25000

Figure 2.5.1 Area distribution for the column.

In comparison to a constant area beam with $A_0 = V_0/l$ and $I_0 = \alpha V_0^3/l^3$

$$(p_{cr})_{\text{opt}} = \frac{125}{9}\frac{EI_0}{l^2} = 1.41\frac{\pi^2 EI_0}{l^2} = 1.41 p_{0cr}. \tag{2.5.49}$$

That is the optimum depth-tapered column is 41% stronger than the corresponding uniform column of the same volume. With c_1, c_2, and β known, $A(x)$ is completely known from Eq. (2.5.41). Figure 2.5.1 shows this area distribution along the length of the column. Notice the undesirable feature of zero areas of cross-section at the two ends of the column. This is a consequence of not having specified a lower bound constraint on the area distribution. • • •

Optimum design of Thin Plates for Stability. For a column, the axial stress-resultant in the prebuckling state is independent of changes in cross-sectional areas

along the length of the column. However, this is not true for thin plates. The in-plane stress-resultants in the prebuckled state of a thin plates are indeed functions of the thickness distribution. The problem of optimizing thin plates for stability is, therefore, significantly more complicated than that for a column.

The situation is not as bad for thin circular plates, for which the resulting governing equations (stability and optimality) are ordinary nonlinear differential equations which can be solved approximately by some numerical schemes like those proposed by Frauenthal [26].

The problem is more complicated for thin rectangular plates which are governed by nonlinear partial differential equations. For instance, questions about the uniqueness of solutions are not as easily answered. Under the assumption of inextensional prebuckling deformations, which lead to thickness-independent in-plane stress-resultants in the pre-buckled state, a condition of uniform strain energy density has been established as being the optimality condition for such plates [27]. Even so, optimization of plates on the basis of such assumptions have led to unsatisfactory solutions for plates with aspect ratios close to unity.

Armand and Lodier [28] have attempted to explain this difficulty in optimizing plates by linking it to the existence of infinitely many local extrema rather than a single global optimum. According to this explanation, the solution obtained by Frauenthal [26] is only a local optimum in the class of continuous thickness distributions. Simitses [29] has shown that for the same volume, stiffened circular plates yield much higher buckling loads than Frauenthal's optimum plate. Similarly, Kamat [27] who optimized finite element models of rectangular plates, observed discontinuous thickness distributions that exhibit a tendency toward formation of ribs, and suspected that stiffened plates would be superior. Haftka and Prasad [30], in their survey paper on optimum structural design with plate bending elements, explain the radically different designs obtained for the same problem by different researchers by offering the conjecture that rib-stiffened plates are better than optimum plates with continuous thickness distributions. Olhoff [31] provides a mathematical justification for this behavior and for the questions of singularities and local optima in plates. The reader is referred to the monograph by Gajewski and Zyczkowski [32] for additional references on this topic.

2.5.3 *Optimum Vibrating Euler-Bernoulli Beams*

The fundamental frequency of free vibration of a beam is given by the minimum value of the Rayleigh quotient

$$\omega^2 = \max_{A(x)} \min_{w(x)} \frac{\int_0^l E\alpha[A(x)]^n w''^2 dx}{\int_0^l \rho A(x) w^2 dx}, \tag{2.5.50}$$

over all kinematically admissible displacement fields [11]. However, even though both stability and vibration of Euler-Bernoulli beams are governed by a similar eigenvalue system, the criteria for optimization of freely vibrating Euler-Bernoulli beams are

different from those for Euler-Bernoulli columns. Unlike the case of columns, the denominator of the Rayleigh quotient for a freely vibrating beam involves the structural mass which is a function of the cross-sectional area.

Consider the problem of the maximization of the fundamental frequency of a freely vibrating Euler-Bernoulli beam of a specified volume V_0 and specific mass ρ. We again assume that $I(x) = \alpha[A(x)^n]$, $n = 1, 2, \text{or } 3$.

The equation of motion of the beam and the necessary optimality condition are then obtained by maximizing the minimum value of the Rayleigh quotient, ω^2, subject to the constant volume constraint. In other words starting with the Lagrangian

$$\mathcal{L} = \max_{A(x)} \min_{w(x)} \frac{\int_0^l E\alpha[A(x)]^n w''^2 dx}{\int_0^l \rho A(x) w^2 dx} - \lambda \left[\int_0^l A(x)dx - V_0 \right] , \tag{2.5.51}$$

which is a functional of the functions $w(x)$ and $A(x)$, and setting its total variation with respect to both functions to zero we get

$$\delta\mathcal{L} = \frac{2\int_0^l E\alpha[A(x)]^n w''\delta w'' dx}{\int_0^l \rho A(x) w^2 dx} - \frac{2\int_0^l E\alpha[A(x)]^n w''^2 dx}{\left[\int_0^l \rho A(x) w^2 dx\right]^2} \left[\int_0^l \rho A(x) w \delta w dx \right]$$

$$+ \frac{\int_0^l nE\alpha[A(x)]^{n-1} w''^2 \delta A dx}{\int_0^l \rho A(x) w^2 dx} - \frac{\int_0^l E\alpha[A(x)]^n w''^2 dx}{\left[\int_0^l \rho A(x) w^2 dx\right]^2} \left[\int_0^l \rho w^2 \delta A(x) dx \right]$$

$$+ \lambda \left[\int_0^l \delta A(x) dx \right] = 0 . \tag{2.5.52}$$

Integrating by parts the first term on the right-hand side of the above equation and collecting the coefficients of δw and δA, we obtain the

Equation of Motion : $$[E\alpha A^n w'']'' - \omega^2 \rho A w = 0 . \tag{2.5.53}$$

Boundary Conditions : $$\delta w = 0 , \quad \text{or} \quad [E\alpha A^n w'']' = 0 , \tag{2.5.54}$$

$$\delta w' = 0 , \quad \text{or} \quad [E\alpha A^n w''] = 0 . \tag{2.5.55}$$

Optimality Condition : $$nE\alpha A^{n-1} w''^2 - \omega^2 \rho w^2 = \text{constant} . \tag{2.5.56}$$

Equation (2.5.56) can be interpreted to imply that the Lagrangian energy density must be uniform in the fundamental mode of an optimum vibrating beam.

As with columns the sufficiency of this optimality condition can be easily demonstrated for the case $n = 1$. For this case Eq. (2.5.56) reduces to

$$E\alpha w''^2 - \omega^2 \rho w^2 = c . \tag{2.5.57}$$

We begin by assuming two designs $A(x)$ and $\bar{A}(x)$ both of which satisfy the constant volume constraint, Eq. (2.5.15) and hence also Eq. (2.5.24). Assume ω and $\bar{\omega}$ to be the fundamental frequencies and w and $\bar{w}$ to be the associated fundamental modes corresponding to the two designs $A(x)$ and $\bar{A}(x)$, respectively. Thus

$$\omega^2 = \frac{\int_0^l E\alpha A w''^2 dx}{\int_0^l \rho A w^2 dx}, \qquad \bar{\omega}^2 = \frac{\int_0^l E\alpha \bar{A} \bar{w}''^2 dx}{\int_0^l \rho \bar{A} \bar{w}^2 dx}. \tag{2.5.58}$$

Since w is kinematically admissible for $\bar{\omega}^2$, by the Rayleigh quotient [11] we are guaranteed that

$$\bar{\omega}^2 = \frac{\int_0^l E\alpha \bar{A} w''^2 dx}{\int_0^l \rho \bar{A} w^2 dx} \geq \bar{\omega}^2 . \tag{2.5.59}$$

But

$$\bar{\omega}^2 \int_0^l \rho \bar{A} w^2 dx = \int_0^l E\alpha \bar{A} w''^2 dx , \tag{2.5.60}$$

and

$$\omega^2 \int_0^l \rho A w^2 dx = \int_0^l E\alpha A w''^2 dx . \tag{2.5.61}$$

Subtracting Eq. (2.5.61) from Eq. (2.5.60) we get

$$\bar{\omega}^2 \int_0^l \rho \bar{A} w^2 dx - \omega^2 \int_0^l \rho A w^2 dx = \int_0^l E\alpha (\bar{A} - A) w''^2 dx . \tag{2.5.62}$$

Now assume that the design $A(x)$ is one that satisfies the optimality condition, Eq. (2.5.57). Equation (2.5.62) can then be written as

$$\bar{\omega}^2 \int_0^l \rho \bar{A} w^2 dx - \omega^2 \int_0^l \rho A w^2 dx = \int_0^l (\bar{A} - A)(c + \omega^2 \rho w^2) dx , \tag{2.5.63}$$

which upon simplification and use of Eq. (2.5.25) yields

$$\bar{\omega}^2 - \omega^2 = 0 . \tag{2.5.64}$$

In light of Eq. (2.5.59) it follows that

$$\omega^2 \geq \bar{\omega}^2 , \tag{2.5.65}$$

thereby establishing the sufficiency of the optimality condition of Eq. (2.5.57).

It should be noted t the same optimality condition can be shown to hold for the dual problem of the n imum weight design of the beam for a specified frequency. Several similar examples optimization with frequency constraints may be found in Refs. 33-38. In particu Turner [34] and Taylor [35] provide exact solutions for axially vibrating minim nass structures at specified natural frequencies.

As in the case of columns, several approximate numerical solutions using the finite element displacement method are available for maximum fundamental frequency of elastically supported vibrating beams of fixed weight carrying a combination of concentrated and distributed non-structural masses and subjected to upper and lower bounds on cross-sectional areas. For examples of this kind of approximate designs, see Refs. 39 and 40. By comparison, published literature on the more practical dual problem of minimizing the weight of beams for specified lower bounds on natural frequencies and upper and lower bounds on design variables appear to be limited [41]. It is not clear whether the primal and dual problems in this case are always equivalent [41, 42].

In closing this topic of vibrating beams, it is appropriate to point out that the same optimality condition, Eq. (2.5.57) also applies to the optimum design of sandwich beams under the constraint of prescribed deflection at the point of application of a single concentrated periodic load (e.g. see Icerman [43]). A more general optimality condition for the constraint of a prescribed deflection at a specified point under a general distributed loading has been provided by Plaut [44]. For sandwich beams, Plaut has shown that it is possible to establish the sufficiency of the optimality condition on the basis of the principle of stationary mutual potential energy introduced by Shield and Prager [45]. A mathematically more rigorous study of this problem using the dynamic compliance of the structure as a constraint has been provided by Mróz [46]. A very extensive bibliography on the topic of optimization for dynamic response may be found in the survey papers referenced in Chapter 1.

Optimum design of Thin Plates for Vibration. The problem of the optimum design of thin plates for vibration is not beset with the difficulty (encountered in the design for buckling) associated with the dependence of the prebuckling stress-resultants on the thickness distribution. That may explain why the problem of the optimum design of thin plates for vibration appears to have received a greater attention than the corresponding problem for stability. Haftka and Prasad [30] have provided an extensive bibliography on the optimum design of plate bending elements for vibration.

The solution to the problem of the optimum design of a circular plate for vibration was first provided by Olhoff [47]. Olhoff showed (see Exercise 8) that under the assumption of a rotationally symmetric lowest mode, the problem reduces to an ordinary, fourth-order, nonlinear, singular but homogeneous eigenvalue problem. An approximate numerical solution to this problem was generated, but the solution so obtained is only a local optimum belonging to the class of continuous thickness distributions. For the same volume, it is easy to devise stiffened circular plate configurations that possess far higher fundamental vibration frequencies than that of Olhoff's original solution [47].

For rectangular plates, the optimum designs of finite element models that allow discontinuous thickness distributions again exhibit a tendency to distribute the material of the plate along discrete ribs [48-50]. For the same volume, a stiffened rectangular plate can be expected to have much higher fundamental frequency of vibration than that of a plate optimized on the basis of a continuous thickness distribution.

2.6 Use of Series Solutions in Structural Optimization

The methods of calculus of variations discussed in the previous sections are ideally suited for simple problems where the unknowns are design functions such as area distributions. These problems are called distributed parameter optimization problems.

Another approach for solving distributed parameter problems which are not simple enough to be attacked by the methods of Variational Calculus is the use of series solutions. The basic idea is to assume a series representation of the unknown design function within the domain of the structure along with the assumed response functions such as displacements. In general, therefore, the series solution method reduces continuous mathematical programming problems to discrete ones with a finite number of design variables. These variables are the coefficients of the series representation of the unknown design function. This idea was initially presented by Balasubramanyam and Spillers [51] who solved various vibration and buckling problems using Fourier series representation of the cross-sectional area of beam and column structures. A similar procedure was recently used by Parbery [52] to obtain minimum-area shapes for desired torsional and flexural rigidity. The method will be demonstrated by the following example.

Example 2.6.1

The optimum design of a buckling critical simply supported column is repeated in this example [51] to demonstrate the use of Fourier series approach. As in the examples discussed earlier there is a fixed material volume constraint, see Eq. (2.5.15). The objective is to find the cross-sectional area distribution of a plane-tapered column that maximizes the buckling load. That is, the cross-sectional area distribution is assumed to be related to a change in width (direction perpendicular to the deformation direction) of a rectangular section with constant depth. This corresponds to $n = 1$ in Eq. (2.3.26).

We start with the governing stability equation for the problem

$$E\alpha A(x)w'' + pw = 0 \; . \tag{2.6.1}$$

Expanding the unknown quantities in two-term truncated Fourier series we have

$$w = a_1 \sin \frac{\pi x}{L} + a_3 \sin \frac{3\pi x}{L} \, , \tag{2.6.2}$$

$$A(x) = \beta_0 - \beta_2 \cos \frac{2\pi x}{L} . \tag{2.6.3}$$

Note that the boundary conditions of the column eliminate the need for a constant term in Eq. (2.6.2). Because of the expected symmetry of the mode shape and the cross-sectional area distribution a_2 and β_1 terms are omitted from the Fourier series. The selection of the cosine representation for the cross-sectional area makes it possible to reduce the products of Fourier series (Aw'') directly to a single series.

The key strategy in this application is to reduce the number of unknown terms by substituting these assumed forms into the appropriate equations. Equating the coefficients of the similar trigonometric functions one obtains algebraic equations that must be satisfied by these coefficients. For example, using Eq. (2.4.15) we can show that

$$\beta_0 = \frac{V_0}{L} . \tag{2.6.4}$$

Substituting β_0 back to Eq. (2.6.3) and then using Eq. (2.6.2) we obtain the following product

$$\alpha A(x) w'' = -(\frac{\pi}{L})^2 \left[(\frac{\alpha V_0}{L} a_1 + \frac{\alpha \beta_2}{2} a_1 - \frac{9\alpha\beta_2}{2} a_3) \sin \frac{\pi x}{L} \right.$$

$$\left. + (\frac{-\alpha\beta_2}{2} a_1 + \frac{9\alpha V_0}{L} a_3) \sin \frac{3\pi x}{L} - \frac{9\alpha\beta_2}{2} a_3 \sin \frac{5\pi x}{L} \right] , \tag{2.6.5}$$

where the trigonometric identity

$$2 \cos A \sin B = \sin(A + B) - \sin(A - B) , \tag{2.6.6}$$

has also been used. Using Eq. (2.6.5) in the equilibrium equation and equating the coefficients of the sine terms we obtain the following algebraic equations.

$$-E(\frac{\pi}{L})^2 (\frac{\alpha V_0}{L} a_1 + \frac{\alpha\beta_2}{2} a_1 - \frac{9\alpha\beta_2}{2} a_3) + p a_1 = 0 , \tag{2.6.7}$$

$$-E(\frac{\pi}{L})^2 (\frac{9\alpha V_0}{L} a_3 - \frac{\alpha\beta_2}{2} a_1) + p a_3 = 0 , \tag{2.6.8}$$

$$E(\frac{\pi}{L})^2 (\frac{9\alpha\beta_2}{2} a_3) = 0 . \tag{2.6.9}$$

For a nontrivial solution, the determinant of the coefficient matrix for the unknown mode shape $(a_1, a_3)^T$ must vanish. This results in the following quadratic relation for the buckling load p in terms of the only unknown coefficient β_2 left in the problem

$$p^2 - E(\frac{\pi}{L})^2 (\frac{10\alpha V_0}{L} + \frac{\alpha\beta_2}{2}) p + 9\alpha^2 E^2 (\frac{\pi}{L})^4 (\frac{V_0^2}{L^2} + \frac{V_0\beta_2}{2L} - \frac{\beta_2^2}{4}) = 0 . \tag{2.6.10}$$

The expression for the critical load is given by

$$p = \frac{E(\frac{\pi}{L})^2\alpha}{4}\left[\frac{20V_0}{L} + \beta_2 \pm (256\frac{V_0^2}{L^2} - 32\frac{V_0\beta_2}{L} + 37\beta_2^2)^{\frac{1}{2}}\right] . \tag{2.6.11}$$

In order to determine the value of β_2 that maximizes the buckling load we take the derivative of Eq. (2.6.11) with respect to the unknown parameter β_2 and equate it to zero. Resulting optimum value of β_2^* is

$$\beta_2^* = \frac{32}{37}\frac{V_0}{L} , \tag{2.6.12}$$

and the corresponding optimum value of the buckling load is

$$p_{\rm cr}^* = \frac{45}{37}\frac{\pi^2 E\alpha V_0}{L^3} = \frac{45}{37}p_{0cr} . \tag{2.6.13}$$

where p_{0cr} is the buckling load of the constant-cross-section column of volume V_0. Although 22% stronger than the constant cross-section column of the same volume, this design is inferior to the design obtained in Example 2.5.1. In that example the change in area was achieved by varying the depth of the cross section keeping the width constant ($n = 3$). Clearly modifying the depth of the cross section is a more effective way of achieving increased buckling resistance. Example 2.5.1 repeated with $n = 1$ results in a quadratic distribution of the cross-sectional area with a critical load of $p_{\rm cr}^* = 12/\pi^2 p_{0cr}$, which is almost identical to the result obtained in Eq. (2.6.13). Moreover, the advantage of this method over other classical methods is in its ability to deal with more general structural problems under a variety of load conditions that may not be possible to solve using variational calculus. • • •

The success of the series solution in optimization is closely related to the form of the series chosen for the representation of the unknown function. In order to keep the number of design variables to a minimum, only few terms in the series representation should be used. But, with a small number of terms used in the series, the approximation of the solution of the governing differential equations of the problem may be poor. Selection of the two-term approximation for the mode shape in the example just covered makes it possible to come up with a one parameter solution for the maximum buckling load in a closed form. However, it is important to note that the two-term solution shown above does not satisfy the equilibrium equation exactly. The last term in Eq. (2.6.5), when substituted into the equilibrium equation does not vanish. If, on the other hand, one uses too many terms in the series finding the optimum values of the coefficients of the terms becomes difficult, and may require the use of a formal search technique. A simple way of reducing the number of design variables without the loss of accuracy is to use possible symmetry inherent to the problem so that only a part of the geometry needs to be modelled. A good example of this approach is demonstrated in [52] where three-fold symmetry is used for the cross-sectional shape of a bar in torsion.

2.7 Exercises

1. The equations of equilibrium and the associated boundary conditions of an elastic structure can be obtained by requiring that the potential energy of the structure be a minimum. Illustrate this for a cantilever Euler-Bernoulli beam. Comment on the types of boundary conditions at the two ends of the beam. Assume that the potential energy of such a beam is given by

$$\Pi = \frac{1}{2}\int_v \sigma\epsilon \, dv - \int_0^l q(x)w dx \,,$$

where

$$\epsilon = -y\frac{d^2w}{dx^2}, \quad \text{and} \quad \sigma = E\epsilon \,,$$

and $q(x)$ is the distributed external transverse loading acting along the beam.

2. Solve Example Problem 2.3.2 for $q = q_0$, $\xi = l/2$, assuming $n = 2$ and $n = 3$.

3. Solve Example Problem 2.3.3 for the following cases

$a)n = 1; \quad q(x) = q_0(l - x)/l.$

$b)n = 1; \quad q(x) = 4q_0(lx - x^2)/l^2.$

$c)n = 2; \quad q(x) = q_0.$

$d)n = 2; \quad q(x) = q_0(l - x)/l.$

$e)n = 3; \quad q(x) = q_0.$

$f)n = 3; \quad q(x) = q_0(l - x)/l.$

4. Solve Example 2.4.2.

5. Determine the optimum area distribution and corresponding buckling loads of the following Euler-Bernoulli columns subject to the constant volume constraint;

a) cantilever column, n=1, 2, and 3.

b) simply-supported column, n = 1,2.

6. The Rayleigh quotient for an axially vibrating bar with an attached non-structural mass m at the free end $x = l$ is given by

$$\omega^2 = \frac{\int_0^l EA(x)u'^2 dx}{\int_0^l \rho A(x)u^2 dx + m[u(l)]^2} \,.$$

a) Derive the equation of motion and the optimality condition for the minimum mass design of the bar with a specified fundamental frequency ω_o.

b) Verify Turner's solution [34] that for such a bar

$$A(x) = \frac{\beta m}{\rho} \tanh \beta l \left[\frac{\cosh \beta l}{\cosh \beta x}\right]^2, \quad \text{and} \quad V_0 = \frac{m}{\rho} \sinh^2 \beta l,$$
$$u(x) = sinh\beta x / sinh\beta l.$$

where $\beta = \sqrt{\omega_0^2 \rho / E}$.

7. Begin with a Rayleigh quotient similar to that of the previous problem for a vibrating cantilever beam of sandwich construction. Assume that the beam carries a distributed non-structural mass $m(x) \gg \rho A(x)$. Verify Taylor's solution [35] that the area distribution $A(x)$ for the optimum beam with a specified fundamental frequency ω_0 is given by

$$A(x) = \frac{\omega_0^2}{Ec^2} \int_x^l (\xi - x)\xi^2 m(\xi) d\xi,$$

where $2c^2 = I(x)/A(x)$, and $\xi = x/l$.

8. Show that the fundamental frequency for a thin circular plate of radius a and thickness distribution function $t(r)$ is given by

$$\omega^2 = \min_{w(x)} \frac{\int_0^1 h^3(\xi)[w''^2 + 2\nu w'' w'/\xi + (w'/\xi)^2]\xi d\xi}{\int_0^1 h(\xi) w^2 \xi d\xi},$$

where

$$\xi = \frac{r}{a}, \quad h = \frac{ta^2}{V_0}, \quad \text{and} \quad V_0 = \int_0^a 2\pi t r dr,$$

with primes denoting differentiation with respect to the non-dimensional radial coordinate ξ.

Also, show that the optimality condition for maximizing the fundamental frequency of such a plate with a specified volume V_0,

$$3h^2[w''^2 + 2\nu w'' w'/\xi + (w'/\xi)^2] - \omega^2 w^2 = \text{constant}, \quad 0 \le \xi \le 1,$$

can be reduced to imply a constant Lagrangian density in the fundamental mode.

9. Solve Example Problem 2.6.1 assuming $n = 2$.

10. The governing equation of motion for the steady-state forced vibration of a simply-supported Euler-Bernoulli beam is given

$$(EIw'')'' - \rho A \omega^2 w = q(x,t),$$

where the applied transverse load $q(x,t) = q_0 \sin \omega t$ and the area distribution is related to the moment of inertia of the cross-section by $I(x) = \alpha[A(x)]^n$. Determine the optimal distribution of the cross-sectional area for $n = 1$ and 2 such that the center displacement, $w(l/2)$ is minimized subject to a specified constant volume, V_0, constraint. Assume two-term symmetric solution for the displacement and the area distribution.

2.8 References

[1] Hancock, H., Theory of Maxima and Minima. Ginn and Company, New York, 1917.

[2] Gelfand, I.M. and Fomin, S.V., Calculus of Variations. Prentice Hall, Inc., Englewood Cliffs, NJ, 1963.

[3] Pars, L.A., An Introduction to the Calculus of Variations. Heinmann, London, 1962.

[4] Hildebrand, F. B., Methods of Applied Mathematics. Prentice–Hall, New Jersey, 1965.

[5] Reddy, J.N., Energy and Variational Methods in Applied Mechanics. John Wiley and Sons, New York, 1984.

[6] Barnett, R.L., "Minimum Weight Design of Beams for Deflection," J. EM Division, ASCE, Vol. EM1, 1961, pp. 75–95.

[7] Makky, S.M. and Ghalib, M.A., "Design for Minimum Deflection," Eng. Opt., 4, pp. 9–13, 1979.

[8] Taylor, J.E., and Bendsøe, M.P.,, "An Interpretation for Min-Max Structural Design Problems Including a Method for Relaxing Constraints," International Journal of Solids and Structures, 30, 4, pp. 301–314, 1984.

[9] Prager, W. and Taylor, J.E., "Problems of Optimal Structural Design," J. Appl. Mech. 35, pp. 102–106, 1968.

[10] Prager, W., "Optimization of Structural Design," J. Optimization Theory and Applications, 6, pp. 1–21, 1979.

[11] Washizu, K., Variational Methods in Elasticity and Plasticity. 2nd ed. Pergamon Press, 1975.

[12] Keller, J.B., "The Shape of the Strongest Column," Arch. Rat. Mech. Anal. 5, pp. 275–285, 1960.

[13] Tadjbaksh, I. and Keller, J.B., "Strongest Columns and Isoperimetric Inequalities for Eigenvalues," J. Appl. Mech. 29, pp. 159–164, 1962.

[14] Keller, J.B. and Niordson, F.I., "The Tallest Column," J. Math. Mech., 29, pp. 433–446, 1966.

[15] Huang, N.C. and Sheu, C.Y., "Optimal Design of an Elastic Column of Thin-Walled Cross Section," J. Appl. Mech., 35, pp. 285–288, 1968.

[16] Taylor, J.E., "The Strongest Column — An Energy Approach," J. Appl. Mech., 34, pp. 486–487, 1967.

[17] Salinas, D., On Variational Formulations for Optimal Structural Design. Ph.D. Dissertation, University of California, Los Angeles, 1968.

[18] Simitses, G.J., Kamat, M.P. and Smith, C.V., Jr., "The Strongest Column by the Finite Element Displacement Method," AIAA Paper No: 72–141, 1972.

[19] Hornbuckle, J.C., On the Automated Optimal Design of Constrained Structures. Ph.D. Dissertation, University of Florida, 1974.

[20] Turner, H.K. and Plaut, R.H., "Optimal Design for Stability under Multiple Loads," J. EM Div. ASCE 12, pp. 1365–1382, 1980.

[21] Olhoff, N.J. and Rasmussen, H., "On Single and Bimodal Optimal Buckling Modes of Clamped Columns," Int. J. Solids and Structures, 13, pp. 605–614, 1977.

[22] Masur, E.F., "Optimal Structural Design under Multiple Eigenvalue Constraints," Int. J. Solids Structures, 20, pp. 211–231, 1984.

[23] Masur, E.F., "Some Additional Comments on Optimal Structural Design under Multiple Eigenvalue Constraints," Int. J. Solids Structures, 21, pp. 117–120, 1985.

[24] Olhoff, N.J., "Structural Optimization by Variational Methods," in Computer Aided Structural Design: Structural and Mechanical Systems (C.A. Mota Soares, Editor), Springer Verlag, pp. 87–164, 1987.

[25] Plaut, R.H., Johnson, L.W. and Olhoff, N., "Bimodal Optimization of Compressed Columns on Elastic Foundations," J. Appl. Mech., 53, pp. 130–134, 1986.

[26] Frauenthal, J.C., "Constrained Optimal Design of Circular Plates against Buckling," J. Struct. Mech., 1, pp. 159–186, 1972.

[27] Kamat, M.P., Optimization of Structural Elements for Stability and Vibration. Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA, 1972.

[28] Armand, J.L. and Lodier, B., "Optimal Design of Bending Elements," Int. J. Num. Meth. Eng., 13, pp. 373–384, 1978.

[29] Simitses, G.J., "Optimal Versus the Stiffened Circular Plate," AIAA J., 11, pp. 1409–1412, 1973.

[30] Haftka, R.T. and Prasad, B., "Optimum Structural Design with Plate Bending Elements — A Survey," AIAA J., 19, pp. 517–522, 1981.

[31] Olhoff, N., "On Singularities, Local Optima and Formation of Stiffeners in Optimal Design of Plates," In: Optimization in Structural Design, A. Sawczuk and Z. Mróz (eds.). Springer-Verlag, 1975, pp. 82–103.

[32] Gajewski, A., and Zyczkowski, M., Optimal Structural Design under Stability Constraints, Kluwer Academic Publishers, 1988.

[33] Niordson, F.I., "On the Optimal Design of a Vibrating Beam," Quart. Appl. Math., 23, pp. 47–53, 1965.

[34] Turner, M.J., "Design of Minimum-Mass Structures with Specified Natural Frequencies," AIAA J., 5, pp. 406–412, 1967.

[35] Taylor, J.E., "Minimum-Mass Bar for Axial vibration at Specified Natural Frequency," AIAA J. , 5, pp. 1911–1913, 1967.

[36] Zarghamee, M.S., "Optimum Frequency of Structures," AIAA J., 6, pp. 749–750, 1968.

[37] Brach, R.M., "On Optimal Design of Vibrating Structures," J. Optimization Theory and Applications, 11, pp. 662–667, 1973.

[38] Miele, A., Mangiavacchi, A., Mohanty, B.P. and Wu, A.K., "Numerical Determination of Minimum Mass Structures with Specified Natural Frequencies," Int. J. Num. Meth. Engng., 13, pp. 265–282, 1978.

[39] Kamat, M.P. and Simitses, G.J., "Optimum Beam Frequencies by the Finite Element Displacement Method," Int. J. Solids and Structures, 9, pp. 415–429, 1973.

[40] Kamat, M.P., "Effect of Shear Deformations and Rotary Inertia on Optimum Beam Frequencies," Int. J. Num. Meth. Engng., 9, pp. 51–62, 1975.

[41] Pierson, B.L., "A Survey of Optimal Structural Design under Dynamic Constraints," Int. J. Num. Meth. Engng., 4, pp. 491–499, 1972.

[42] Kiusalaas, J., "An Algorithm for Optimal Structural Design with Frequency Constraints," Int. J. Num. Meth. Engng., 13, pp. 283–295, 1978.

[43] Icerman, L.J., "Optimal Structural Design for given Dynamic Deflection," Int. J. Solids and Structures, 5, pp. 473–490, 1969.

[44] Plaut, R.H., "Optimal Structural Design for given Deflection under Periodic Loading," Quart. Appl. Math., 29, pp. 315–318, 1971.

[45] Shield, R.T. and Prager, W., "Optimal Structural Design for given Deflection," Z. Angew. Math. Phys., 21, pp. 513–523, 1970.

[46] Mróz, Z., "Optimal Design of Elastic Structures subjected to Dynamic, Harmonically Varying Loads," Z. Angew. Math. Mech., 50, pp. 303–309, 1970.

[47] Olhoff, N., "Optimal Design of Vibrating Circular Plates," Int. J. Solids and Structures, 6, pp. 139–156, 1970.

[48] Olhoff, N., "Optimal Design of Vibrating Rectangular Plates," Int. J. Solids and Structures, 10, p. 93–109, 1974.

[49] Kamat, M.P., "Optimal Thin Rectangular Plates for Vibration," Recent Advances in Engineering Science, Vol. 3. Proceedings of the 10th Annual Meeting of the Society of Engineering Science, pp. 101–108, 1973.

[50] Armand, J.L., Lurie, K.A. and Cherkaev, A.V., "Existence of Solutions of the Plate Optimization Problem," Proceedings of the International Symposium Structural Design, Tucson, AZ, pp. 3.1–3.2, 1981.

[51] Balasubramanyam, K. and Spillers, W.R., "Examples of the Use of Fourier Series in Structural Optimization," Quart. of Appl. Math., 3, pp. 559–566, 1986.

[52] Parbery, R.D., "On Minimum-Area Convex Shapes of given Torsional and Flexural Rigidity," Eng. Opt., 13, pp. 189–196, 1988.

Linear Programming 3

Mathematical programming is concerned with the extremization of a function f defined over an n-dimensional design space $\mathbf{R}^n$ and bounded by a set $\mathbf{S}$ in the design space. The set $\mathbf{S}$ may be defined by equality or inequality constraints, and these constraints may assume linear or nonlinear forms. The function f together with the set $\mathbf{S}$ in the domain of f is called a *mathematical program* or a mathematical programming problem. This terminology is in common usage in the context of problems which arise in planning and scheduling which are generally studied under operations research, the branch of mathematics concerned with decision making processes. Mathematical programming problems may be classified into several different categories depending on the nature and form of the design variables, constraint functions, and the objective function. However, only two of these categories are of interest to us, namely *linear* and *nonlinear programming* problems (commonly designated as LP and NLP, respectively).

The term linear programming (LP) describes a particular class of extremization problems in which the objective function and the constraint relations are linear functions of the design variables. Because the necessary condition for an interior minimum is the vanishing of the first derivative of the function with respect to the design variables, linear programming problems have a special feature. That is, the derivatives of the objective function with respect to the variables are constants which are not necessarily zeroes. This implies that the extremum of a linear programming problem cannot be located in the interior of the feasible design space and, therefore, must lie on the boundary of the design space described by the constraint relations. Since the constraint relations are also linear functions of the design variables the optimum design must lie at the intersection of two or more constraint functions, unless the bounding constraint is parallel to the contours of the objective function. This special feature of the linear programming problems makes it possible to devise effective algorithms that are suitable for reaching optimum solutions. Linear programming problems involving large number of design variables and constraints are usually solved by an extremely efficient and reliable method known as the simplex method.

Unfortunately, however, very few physically meaningful problems in structural design, if any, can be formulated directly as LP problems without involving a degree

of simplification. Most structural design problems involve highly nonlinear objective function and constraint relations. Nevertheless, the category of LP is of interest to us because of several reasons. First of all, many nonlinear constrained problems can be approximated by linear ones which can be solved efficiently by using standard LP algorithms. Using such approximations opens up a possibility for solving NLP problems. That is, almost all NLP problems can be solved as a sequence of repetitive approximate LP problems which converge to the exact solution of the original NLP problem provided that the procedure is repeated enough number of times. This powerful procedure is called sequential linear programming (SLP) and is discussed in Chapter 6. Also, methods intended for nonlinear constrained problems often utilize linear programming as an intermediate step. For example, Zoutendijk's method of feasible directions (see Chapter 5) employs a LP to generate a search direction.

Whether a given nonlinearly constrained problem in structural optimization can be replaced by an approximately equivalent linearly constrained problem depends to a great extent on the intuition of the designer and his knowledge and experience with the given problem. Such approximations must usually be made so as not to alter the overall character of the problem radically. The trade-off between a higher value of the objective function attained because of the approximation and a lower computational cost must be weighted carefully. Fortunately, there are a few classes of problems in structural analysis and design in which such approximations have found to be indeed reasonable. In the following sections some of those problems will be presented as LP problems, and graphical solution of a simple LP problem will be demonstrated. Next, the standard formulation of the mathematical LP problems will be presented, and solution techniques for LP problems will be discussed. Finally, we would discuss a special case of LP problems that require the design variables to assume values from a set of discrete or integer values.

3.1 Limit Analysis and Design of Structures Formulated as LP Problems

In many structural design problems the initiation of yielding somewhere in the structure is considered to be a criterion for failure, but this is not always reasonable. In many cases we are not interested in the initiation of failure but in the maximum load, called the *limit load* or the *collapse load*, that a structure may carry without losing its functionality. The collapse load can be defined as the load required to generate enough local plastic yield points (referred as plastic hinges for bending type members) to cause the structure to become a mechanism and develop excessive deflections. While the exact calculation of the collapse load of a structure requires the solution of a costly nonlinear problem, for ductile materials it is possible to obtain a conservative estimate of that load by making the assumption that the material behaves as an elastic-perfectly plastic material. That is, the material is assumed to follow the stress-strain diagram shown in Fig. 3.1.1, yielding at stress level σ_0 but functioning as a constant stress carrying medium beyond the elastic limit. It is this important assumption that allows the limit analysis and design problems to be formulated as LP problems.

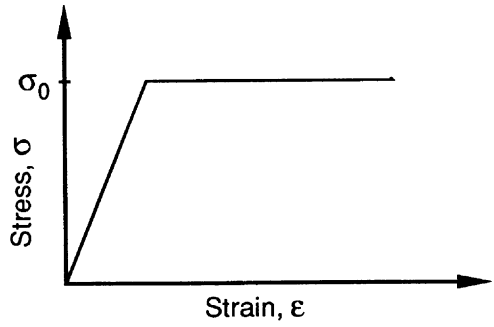


Figure 3.1.1 The stress-strain curve for an elastic-perfectly plastic material.

A simple example of a three bar truss is used in the following example to illustrate the difference between the calculation of the load which initiates yielding and the estimate of the collapse load.

Example 3.1.1

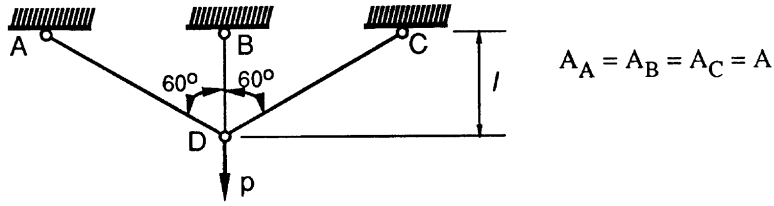


Figure 3.1.2 Collapse of a three bar truss subject to a single load.

We perform the collapse analysis of a three bar pin jointed truss under a vertical load as shown in Fig. 3.1.2. All three bars have the same cross-sectional area A, and are made of material having Young's modulus E and yield stress σ_0. We start by calculating the load p at which the first bar yields. Denoting the vertical displacement at the common joint D by v, we obtain the strains in the three members

$$\epsilon_B = \frac{v}{l}, \qquad \epsilon_A = \epsilon_C = \frac{v}{4l} \, . \tag{3.1.1}$$

The corresponding member forces are

$$n_B = \frac{EA}{l} v, \qquad n_A = n_C = \frac{EA}{4l} v = 0.25 n_B \, . \tag{3.1.2}$$

Using the two equations of equilibrium at joint D, we get

$$n_A = n_B, \qquad p = n_B + \frac{1}{2}(n_A + n_C) = 1.25 n_B \, , \tag{3.1.3}$$

and the internal forces in the three members are determined as

$$n_A = n_C = 0.2p, \qquad n_B = 0.8p\ . \tag{3.1.4}$$

Clearly, as the load is increased from zero member B yields first, when

$$n_B = \sigma_0 A, \qquad \text{or} \qquad p = 1.25A\sigma_0\ . \tag{3.1.5}$$

The structure does not collapse, however, at $p = 1.25A\sigma_0$ since members A and C can still carry the applied load without experiencing excessive deformations. We may increase the load until member A or C yields. Since we have assumed elastic-perfectly plastic material behavior, the stress in member B will remain at σ_0 as we increase the load beyond the initial yield load. Due to the symmetry in this problem, the next yielding takes place simultaneously in members n_A and n_C. Therefore, at collapse all three members will be at the yield point so that

$$n_A = n_B = n_C = A\sigma_0\,, \tag{3.1.6}$$

and from the equations of equilibrium Eq. (3.1.4) we have

$$p_{\text{collapse}} = 2A\sigma_0\ . \tag{3.1.7}$$

This is a 60% increase over the load when first yielding starts. • • •

In example 3.1.1 it was easy to identify the sequence of yielding of the members and determine the state of stress in the members at collapse. This fact permitted us to determine the collapse load without difficulty. In general, it is not easy to determine the combination of members that will yield at collapse, and the stress distribution at the collapse is not known. Fortunately, it is possible to cast the problem as an LP problem in order to determine the collapse load [1] based on a general theorem of the theory of plasticity. This theorem is the lower bound theorem, and it is quoted below from Calladine Ref. 2.

The Lower Bound Theorem: If any stress distribution throughout the structure can be found which is everywhere in equilibrium internally and balances the external loads, and at the same time does not violate the yield conditions, these loads will be carried safely by the structure.

The application of this theorem will now be demonstrated for a problem where the choice of stress at collapse is not as trivial as it was in example 3.1.1. We use the same structure used in the previous example, but with an added horizontal load at point D.

Example 3.1.2

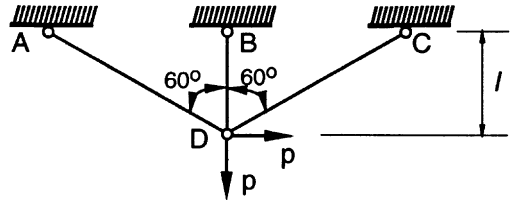


Figure 3.1.3 Limit analysis of a three bar truss subjected to two loads.

Consider the limit analysis of the three bar truss of Figure 3.1.3 under the combined vertical and horizontal loads of equal magnitude, p. The equations of equilibrium in this case are

$$\begin{aligned} n_B + \frac{1}{2}(n_A + n_C) - p = 0\,, \\ \frac{\sqrt{3}}{2}(n_A - n_C) - p = 0\,, \end{aligned} \tag{3.1.8}$$

and we have the constraints

$$-A\sigma_0 \quad \leq \quad n_A, n_B, n_C \quad \leq \quad A\sigma_0\,. \tag{3.1.9}$$

It is no longer easy to know which two of the three bars yield at the collapse. However, we may try different combinations of $n_A, n_B,$ and n_C that satisfy the equations of equilibrium in order to obtain a lower bound to the collapse load. For example, if we try $n_C = 0$, we obtain from the equilibrium relations (3.1.8)

$$n_A = \frac{2}{\sqrt{3}}p = 1.155p, \quad \text{and} \quad n_B = 0.423p\,. \tag{3.1.10}$$

Clearly in this case n_A reaches its yield value of $A\sigma_0$ before n_B so that

$$n_A = A\sigma_0, \quad n_B = 0.366A\sigma_0, \quad \text{and} \quad p = \frac{\sqrt{3}}{2}A\sigma_0 = 0.866A\sigma_0\,. \tag{3.1.11}$$

Having satisfied all the requirements for the lower bound theorem, we thus know that the collapse load is bounded below by $0.866A\sigma_0$. We can now try different combinations of member force distribution until we obtain a higher value of p than the one obtained in Eq. (3.1.11). To get the best estimate, we cast the problem as a maximization problem

$$\begin{aligned} &\textbf{maximize} \quad p \\ &\textbf{such that} \quad \text{Eqs. (3.1.8) and Eqs. (3.1.9) are satisfied.} \end{aligned} \tag{3.1.12}$$

This is clearly a LP problem in the variables n_A, n_B, n_C and p , and may be solved using any LP algorithm. It is also simple enough to admit a graphical solution if required (see Exercise 1). • • •

The general formulation of the calculation of the limit load for truss structures is similar to the procedure used in example 3.1.2 . It is assumed that no part of the truss structure fails by buckling before the plastic collapse load is reached. If we have a truss structure with r members loaded by a system of loads $\lambda \mathbf{p}$, where $\mathbf{p}$ is a given load vector and λ is a scalar, the limit load can be determined by finding the largest value of λ that the structure can support. The equations of equilibrium are written as

$$\sum_{j=1}^{r} e_{ij} n_j = \lambda p_i, \qquad i = 1, \ldots, m\,, \tag{3.1.13}$$

where n_j $(j = 1, \ldots, r)$ are the forces in each of the truss members, e_{ij} are direction cosines, and m is the number of equilibrium equations. The yield constraints are written as

$$A_j \sigma_{Cj} \leq n_j \leq A_j \sigma_{Tj}\,, \tag{3.1.14}$$

where A_j, σ_{Cj}, and σ_{Tj} are the cross-sectional areas, and the yield stresses in compression and tension, respectively. The limit or collapse load is then the solution to the following linear programming problem:

$$\begin{aligned} &\textbf{maximize} && \lambda \\ &\textbf{such that} && \text{Eq. (3.1.13) and Eq. (3.1.14) are satisfied,} \end{aligned} \tag{3.1.15}$$

where λ and the member forces n_j are treated as the design variables.

A related problem is the problem of *limit design* where the collapse load is specified and the optimal cross-sectional areas are sought. Often, the objective is to minimize the total mass of the structure

$$\textbf{minimize} \qquad m = \sum_{j=1}^{r} \rho_j A_j l_j\,, \tag{3.1.16}$$

where ρ_j and l_j are the mass density and the length of member j, respectively. The minimization problem of Eq. (3.1.16) has the same set of constraints, Eqs. (3.1.13) and (3.1.14), that applies to the limit analysis problem, but both n_j and A_j are treated as design variables. This time, however, the load amplitude λ is specified.

Example 3.1.3

Formulate the limit analysis and design of the five bar truss shown in Figure (3.1.4) as linear programs. Assume that all bars are made of the same material and that $\sigma_C = -\sigma_T = \sigma_0$.

The vertical and horizontal equations of equilibrium at the unrestrained nodes of the structure are

$$n_{13} + \frac{\sqrt{2}}{2} n_{23} = 0, \qquad n_{24} + \frac{\sqrt{2}}{2} n_{14} = 0\,, \tag{3.1.17a}$$

$$n_{34} + \frac{\sqrt{2}}{2} n_{23} = 0, \qquad n_{34} + \frac{\sqrt{2}}{2} n_{14} = p\,. \tag{3.1.17b}$$

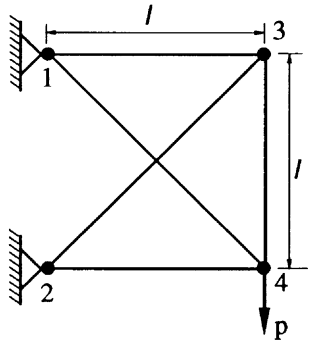


Figure 3.1.4 Limit analysis and design of a five bar truss.

The yield constraints are

$$\begin{aligned} &-A_{13}\sigma_0 \le n_{13} \le A_{13}\sigma_0, && -A_{23}\sigma_0 \le n_{23} \le A_{23}\sigma_0, \\ &-A_{14}\sigma_0 \le n_{14} \le A_{14}\sigma_0, && -A_{24}\sigma_0 \le n_{24} \le A_{24}\sigma_0, \\ &-A_{34}\sigma_0 \le n_{34} \le A_{34}\sigma_0 . \end{aligned} \qquad (3.1.18)$$

The limit load problem is specified as defined previously: maximize p, by varying the member forces, such that the equations of equilibrium and the yield constraints are satisfied. The limit design problem is

$$\begin{aligned} &\textbf{minimize} && \frac{m}{\rho l} = A_{13} + A_{24} + A_{34} + \sqrt{2}(A_{14} + A_{23}) \\ &\textbf{such that} && \text{Eq. (3.1.17) and Eq. (3.1.18) are satisfied.} \end{aligned} \qquad (3.1.19)$$

For the limit design problem both the cross-sectional areas and the member forces are treated as design variables. • • •

The analysis and design of structures that include members under bending may be formulated as LP problems as in Refs. 3-5. Cohn, Ghosh, and Parimi [3] provide an excellent unified approach to both the analysis and design of beams, frames, and arches of given configurations under fixed, alternating, and variable repeated or shakedown loadings. We focus our attention here only on simple examples in this class of problems.

The basic hypothesis regarding the material is that the beam or frame is elastic-perfectly plastic. The fully plastic moment, m_p, of a beam cross-section is defined as the bending moment, m, required to make the entire cross-section yield so as to form a hinge with constant bending resistance.

Example 3.1.4

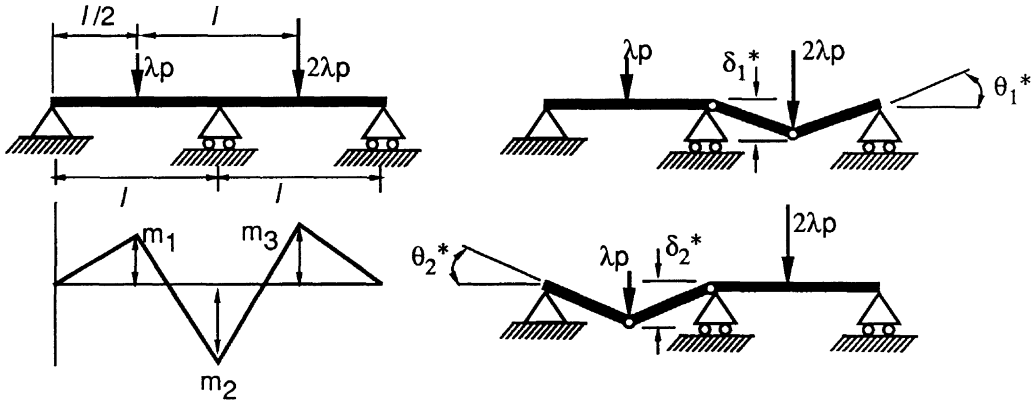


Figure 3.1.5 Limit analysis of a two-span beam.

Limit analysis of bending members is illustrated by using a two-span continuous beam under the loading shown in Figure 3.1.5. Following the general formulation presented earlier, the limit load is the largest value of λ that the structure can support without forming a mechanism. As in the case of Example 3.1.2 the sequence of hinge formation to form a beam mechanism and the distribution of the bending moments along the span of the beam is not obvious. In fact, there are infinitely many statically admissible bending moment distributions that satisfy the equilibrium equations. However, there are only two possible collapse mechanisms. The two elementary mechanisms and the moment distribution for the beam are presented in Figure 3.1.5.

The LP problem for the plastic analysis is

$$\begin{aligned} &\textbf{maximize} && \lambda \\ &\textbf{subject to} && -m_p \leq m_i \leq m_p, \qquad i = 1,2,3, \end{aligned} \tag{3.1.20}$$

where $m_1, m_2,$ and $, m_3$ are the magnitudes of the bending moment at those points along the beam which have the potential to form plastic hinges; at these points the bending moments have local maxima. These three moments are also unknowns for the problem and need to be determined. At the onset of either of the collapse mechanisms shown in Figure 3.1.5, we can write down two equations of equilibrium by using the principle of virtual displacements. The basic assumption in writing the virtual displacements is that the hinges in the figure are not plastic hinges, but are introduced to permit the small displacements that are assumed to take place while the members between them remain straight. The resulting equilibrium relations are

$$2m_3\theta_1^* + m_2\theta_1^* = 2\lambda p(l/2)\delta_1^*, \tag{3.1.21}$$

$$2m_1\theta_2^* + m_2\theta_2^* = \lambda p(l/2)\delta_2^*, \tag{3.1.22}$$

where θ_1^*, θ_2^* are the virtual rotations of the member at the expected plastic joints and δ_1^*, δ_2^* the virtual displacements of the beam under the load points. The virtual displacements and the rotations are related to one another through kinematic relations, and can be eliminated from the equations. Furthermore, using the two equilibrium equations, we can eliminate the two variables, m_1 and m_3, to reduce the LP problem of 3.1.20 to finding the λ and m_2 such that

$$\begin{aligned} &\textbf{maximize} && \lambda \\ &\textbf{subject to} && -m_p \le (\frac{pl}{4}\lambda - \frac{1}{2}m_2) \le m_p, \\ & && -m_p \le m_2 \le m_p, \\ & && -m_p \le (\frac{pl}{2}\lambda - \frac{1}{2}m_2) \le m_p . \end{aligned} \tag{3.1.23}$$

This is a simple two variable (m_2 and λ) LP problem that can be solved graphically.

• • •

Example 3.1.5

As an illustration of limit design for bending type problems, consider the well-known problem of minimizing the weight of a plane frame to resist a given set of ultimate loads. A single bay, single story portal frame is loaded by a horizontal and a vertical load of magnitude p as shown in Figure 3.1.6. For this design problem the top horizontal member is assumed to be different from the two vertical columns. Accordingly, we assume the beam and the column cross-sections to have associated fully plastic moments m_{pB} and m_{pC}, respectively. These two plastic moments depend on the cross-sectional properties of their respective members and, therefore, are the design variables for the problem.

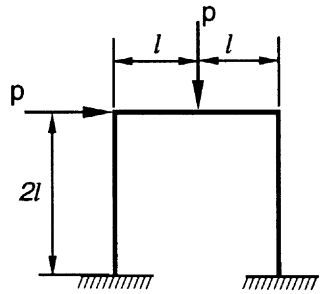


Figure 3.1.6 Portal frame design against plastic collapse.

In order to pose the problem as a weight minimization design problem, we need to relate the design variables and the structural weight. Massonet and Save [6] have shown that for beam sections in bending there is an approximate linear relation

between the weight per running foot, w_l, and the plastic section modulus, m_p/σ_0. Over the relevant range of sections that may be expected to be used for a given frame the error involved in this linearization is of the order of 1%. It is this single assumption which renders the plastic design problem linear.

We will, therefore, assume that the problem of minimizing the weight of a frame for a set of ultimate loads reduces to minimizing a function

$$w = 2m_{p_C} l_C + m_{p_B} l_B = 2m_{p_C}(2l) + m_{p_B}(2l) \ . \tag{3.1.24}$$

In the interest of non-dimensionalization we divide both sides of Eq. (3.1.24) by $2pl^2$ to obtain the weight proportional objective function

$$f(x_1, x_2) = (\frac{w}{2pl^2}) = 2\frac{m_{p_C}}{pl} + \frac{m_{p_B}}{pl} = 2x_1 + x_2 \ . \tag{3.1.25}$$

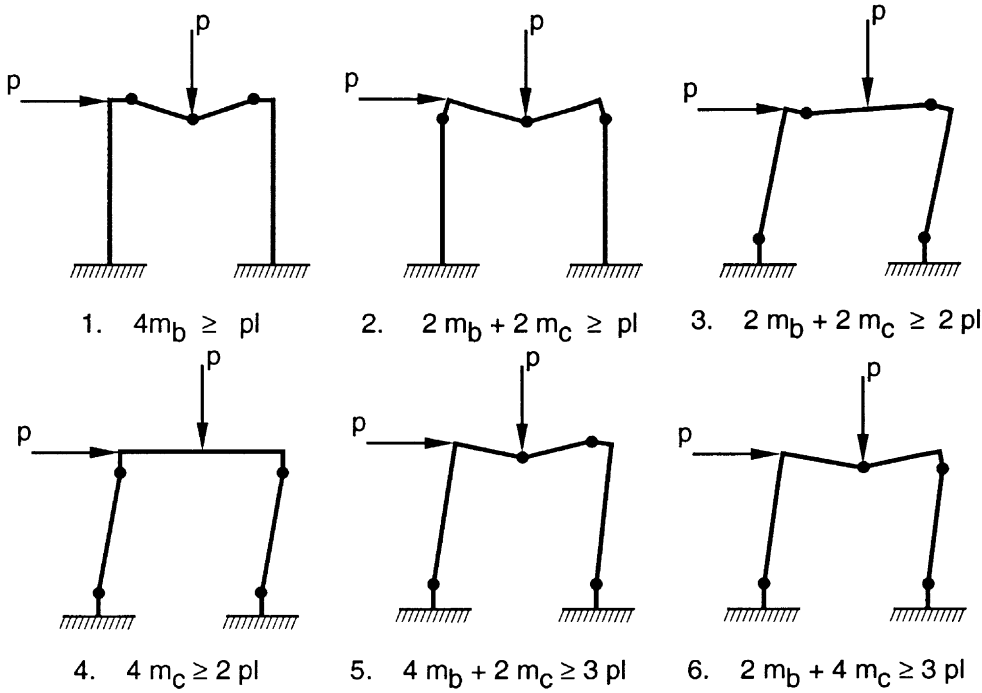


Figure 3.1.7 Collapse mechanisms for the portal frame of Figure 3.1.6.

The equations of equilibrium can be obtained by using the same approach used in the previous example. Figure 3.1.7 shows all possible collapse mechanisms for the frame. The ultimate load carrying capacity of the structure for any given collapse mechanism is obtained by the virtual work equivalence between the external work of the applied loads and the internal work of the fully plastic moments experienced

while undergoing virtual rotations of the plastic hinges. Thus a permissible design is one for which the capacity for internal virtual work is greater than or equal to the external work. It is left as an exercise (see Exercise 4) to verify that behavioral constraints associated with the collapse mechanism of Figure 3.1.7 reduce to

$$4x_2 \geq 1, \quad \text{beam mechanism 1}, \tag{3.1.26}$$

$$2x_1 + 2x_2 \geq 1, \quad \text{beam mechanism 2}, \tag{3.1.27}$$

$$x_1 + x_2 \geq 1, \quad \text{sway mechanism 1}, \tag{3.1.28}$$

$$2x_1 \geq 1, \quad \text{sway mechanism 2}, \tag{3.1.29}$$

$$2x_1 + 4x_2 \geq 3, \quad \text{combined mechanism 1}, \tag{3.1.30}$$

$$4x_1 + 2x_2 \geq 3, \quad \text{combined mechanism 2}. \tag{3.1.31}$$

Furthermore since x_1 and x_2 represent cross-sectional variables it is required that

$$x_1 \geq 0, \qquad \text{and} \qquad x_2 \geq 0\,. \tag{3.1.32}$$

Thus the problem of weight minimization under a set of ultimate load has been reduced to the determination of those non-negative values of $x_1$and x_2 for which f as given by Eq. (3.1.25) is minimized subject to constraints Eqs. (3.1.26 - 3.1.32). The problem is clearly an LP problem. We will defer the analytical solution of this problem until later. • • •

3.2 Prestressed Concrete Design by Linear Programming

Since concrete is weak in tension, prestressing helps to eliminate undesirable tensile stresses in concrete and thereby improve its resistance in bending. A prestressing cable or a tendon exerts an eccentrically applied compressive load to the beam cross-section giving rise to an axial load and possibly a bending moment due to an offset in the cable. In evaluating the total stresses at any given cross-section we must superimpose the stresses due to dead and live loads on the stresses due to the eccentrically applied prestressing forces of the tendons. For a beam of fixed cross-sectional dimensions, the total cost of the beam may be assumed to be approximately proportional to the cost of building in a desired prestressing force. The optimization problem for the design of a prestressed beam thus reduces to minimizing the magnitude of the prestressing force f_0.

Consider the following simple problem of the optimum design of the simply-supported beam shown in Figure 3.2.1 . The initial value of the prestressing force f_0 and the eccentricity f_e is to be determined such that f_0 is a minimum subject to constraints on normal stress, transverse displacement, and upper and lower bound constraints on the design variables. Additionally, in designing a prestressed concrete beam which is expected to remain in service for a number of years, we must allow for the loss of prestressing force through time dependent shrinkage and creep effects in concrete. To simplify design considerations it is frequently assumed that the realizable

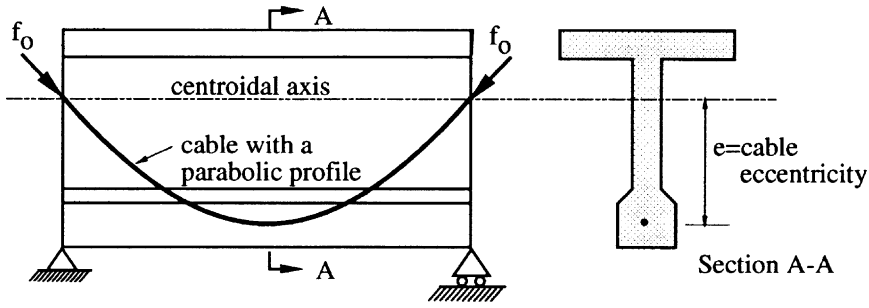


Figure 3.2.1 Simply supported post-tensioned beam.

prestressing force in service is a constant fraction α of the initial prestressing force f_0. In calculating the bending moment distribution or the deflected shape of a prestressed beam, in addition to the usual dead and live loads, we must allow for the equivalent distributed loading (see Exercise 6a) and the end loads resulting from the curved profile of the eccentrically placed tendons. It can be shown [7,8] that for parabolic profiles of the cables (see Figure 3.2.1) the induced moments and deflections are linearly related to the quantity $f_0 e$ with the constant of proportionality k being a function of the known material and cross-sectional properties. With this assumption maximum stresses and the deflections of a simply supported beam occur at the center of the beam. If the maximum positive bending moment and maximum deflection at the center of the simply-supported beam of Figure 3.2.1 due to external loads in the ith loading condition are denoted by m_{ei} and δ_{ei}, respectively, then the beam optimization problem reduces to

$$\textbf{minimize} \qquad f(f_0, e) = f_0 \tag{3.2.1}$$

$$\textbf{subject to} \qquad \sigma^{li} \leq -\frac{\alpha f_0}{a} \pm \frac{m_{ei} - \alpha f_0 e}{z} \leq \sigma^{ui}\,, \tag{3.2.2}$$

$$\delta^{li} \leq \delta_{ei} + \alpha k f_0 e \leq \delta^{ui}\,, \tag{3.2.3}$$

$$e^l \leq e \leq e^u\,, \tag{3.2.4}$$

$$f_0 \geq 0, \qquad i = 1, \ldots, nl\,. \tag{3.2.5}$$

Here nl denotes the number of different loading conditions; $\sigma^l, \sigma^u, \delta^l, \delta^u, e^l$, and e^u denote lower and upper bounds on stress, deflections and the tendon eccentricity; a and z denote the effective area and the section modulus of the cross-section.

The problem as formulated by Eqs. (3.2.1) through (3.2.5) is not an LP problem because it includes the product $f_0 e$ of the two variables. However, it can be easily cast as one by letting

$$m = f_0 e\,, \tag{3.2.6}$$

and expressing the problems in terms of the new design variables f_0 and m. The transformed problem thus reduces to the following LP problem

$$\textbf{minimize} \qquad f(f_0, m) = f_0 \tag{3.2.7}$$

$$\textbf{subject to} \qquad \sigma^{li} \leq -\frac{\alpha f_0}{a} \pm \frac{m_{ei} - \alpha m}{z} \leq \sigma^{ui}\,, \tag{3.2.8}$$

$$\delta^{li} \leq \delta_{ei} + \alpha k m \leq \delta^{ui}\,, \tag{3.2.9}$$

$$m^l \leq m \leq m^u\,, \tag{3.2.10}$$

$$f_0 \geq 0, \qquad i = 1, \ldots, nl\,, \tag{3.2.11}$$

with m^l and m^u being the upper and lower bounds on $f_0 e$.

Morris [9] has treated a similar problem, but with additional constrains on ultimate moment capacity. He also modified the constraint (3.2.11) to allow the American Concrete Institute's limit on the prestressing force intended to prevent premature failure of the beam by pure crushing of the concrete. Morris linearizes part of the problem by using the reciprocal of the prestressing force as one of the design variables; this transformation however fails to linearize the constraint on the ultimate moment capacity. In the interest of linearization, this nonlinear constraint is replaced by a series of piecewise linear connected chords with true values at chord intersections. Kirsch [10] has shown that appropriate transformations can also be used to reduce the design of continuous prestressed concrete beams to equivalent linear programming problems. These problems involve not only the optimization of the prestressing force and the tendon configuration, but also the optimization of the cross-sectional dimensions of the beam.

3.3 Minimum Weight Design of Statically Determinate Trusses

As another example of the design problems that can be turned into LP problems we consider the minimum weight design of statically determinate trusses under stress and deflection constraints. The difficulty in these problems arises due to the nonlinear nature of the deflections as a function of the design variables which are the cross-sectional areas of the truss members. This type of problem, however, belongs to the class of what is known as separable programming [11] problems. In this class of programming the objective function and the constraints can be expressed as a sum of functions of a single design variable. Each such function can be approximated by a piecewise linear function or a set of connected line segments or chords interpolating the actual function at the chord intersections.

A nonlinear separable function of n design variables,

$$f = f(x_1, \ldots, x_n) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)\,, \tag{3.3.1}$$

can be linearized as

$$f = \sum_{k=0}^{m} \eta_{1k} f_{1k} + \sum_{k=0}^{m} \eta_{2k} f_{2k} + \ldots + \sum_{k=0}^{m} \eta_{nk} f_{nk}\,, \tag{3.3.2}$$

with

$$x_1 = \sum_{k=0}^{m} \eta_{1k} x_{1k}, \quad \ldots \quad , x_n = \sum_{k=0}^{m} \eta_{nk} x_{nk}\,, \tag{3.3.3}$$

$$\sum_{k=0}^{m} \eta_{1k} = \sum_{k=0}^{m} \eta_{2k} = \ldots = \sum_{k=0}^{m} \eta_{nk} = 1\,, \tag{3.3.4}$$

$$\eta_{jk} \geq 0, \qquad j = 0, 1, \ldots, n\,, \quad \text{and} \quad k = 0, 1, \ldots, m\,. \tag{3.3.5}$$

Here f_{jk} and x_{jk} are the values of the functions $f_1, f_2, \ldots, f_n$ and the design variables $x_1, x_2, \ldots, x_n$ at $m+1$ preselected points along each of the design variables, and η_{nk}'s are the interpolation functions for the design variables. Note that the number, m, of points selected for each design variable can, in general, be different ($m_1, m_2, \ldots, m_n$, etc.), but for the sake of simplicity they are taken to be equal here.

The purpose of using m intervals with $m+1$ values of the design variables is to cover the entire range of the possible design space accurately. The number of segments m decides the degree of approximation to the original problem— the larger the m the closer the solution of the linear problem will be to the true solution. However, at any given design point, a linear approximation to a nonlinear function requires only the value of the function at two values of a design variable. We, therefore, require that for every design variable $j (j = 1, \ldots, n)$, at most two adjacent η_{jk} be positive. This implies that if, for example, η_{pq} and $\eta_{p(q+1)}$ are non-zero with all other η_{pk} zero, then the value of x_p is in the interval between x_{pq} and $x_{p(q+1)}$ and is given by

$$x_p = \eta_{pq} x_{pq} + \eta_{p(q+1)} x_{p(q+1)}, \qquad \text{with} \qquad \eta_{pq} + \eta_{p(q+1)} = 1\,. \tag{3.3.6}$$

The variables, $(x_1, \ldots, x_n)$, of the function have thus been replaced by the interpolation functions, η_{jk}, only two of which are constrained to be non-zero for each of the design variables. Therefore, we have a linear approximation to the function at every design variable.

Example 3.3.1

As an illustration we consider a problem similar to the one solved by Majid [12]. The objective is the minimum weight design of the four bar statically determinate truss shown in Figure 3.3.1 with stress constraints in the members and a displacement constraint at the tip joint of the truss. In order to simplify the problem we assume members 1 through 3 to have the same cross-sectional area A_1, and the member 4 the area A_2. Under the specified loading, the member forces and the vertical displacement at joint 2 can easily verified to be

$$F_1 = 5p, \quad F_2 = -p, \quad F_3 = 4p, \quad \text{and} \quad F_4 = -2\sqrt{3}p\,, \tag{3.3.7}$$

$$\delta_2 = \frac{6pl}{E}\left(\frac{3}{A_1} + \frac{\sqrt{3}}{A_2}\right)\,, \tag{3.3.8}$$

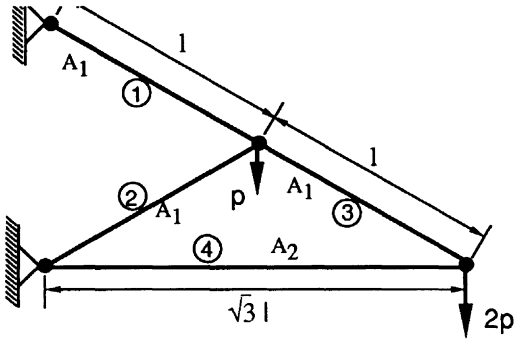


Figure 3.3.1 Four bar statically determinate truss.

where negative values for the forces denote compression. Allowable stresses in tension and compression are assumed to be $7.73 \times 10^{-4}E$ and $4.833 \times 10^{-4}E$, respectively and the vertical tip displacement is constrained to be no greater than $3 \times 10^{-3}l$. The problem of the minimum weight design subject to stress and displacement constraints can be formulated in terms of the non-dimensional variables

$$x_1 = \left(\frac{p}{A_1 E}\right) 10^3, \qquad \text{and} \qquad x_2 = \left(\frac{p}{A_2 E}\right) 10^3, \tag{3.3.9}$$

as

$$\textbf{minimize} \qquad f(x_1, x_2) = \frac{3}{x_1} + \frac{\sqrt{3}}{x_2} \tag{3.3.10}$$

$$\textbf{subject to} \qquad 18x_1 + 6\sqrt{3}x_2 \leq 3, \tag{3.3.11}$$

$$0.05 \leq x_1 \leq 0.1546, \tag{3.3.12}$$

$$0.05 \leq x_2 \leq 0.1395, \tag{3.3.13}$$

where lower bound limits on x_1 and x_2 have been assumed to be 0.05. Except for the objective function which is a separable nonlinear function, the rest of the problem is linear. The objective function can be put in a piecewise linear form by using Eqs. (3.3.2) and (3.3.3). For the purpose of demonstration, we divide the design variable intervals of Eqs. (3.3.12) and (3.3.13) into two equal segments ($m = 2$) resulting in

$$x_{10} = 0.05, \quad x_{11} = 0.1023, x_{12} = 0.1546,$$

$$\text{and } x_{20} = 0.05, \quad x_{21} = 0.09475, x_{22} = 0.1395 .$$

Objective function values corresponding to these points are

$$f_{10} = 20, \quad f_{11} = 9.76, f_{12} = 6.47,$$

$$\text{and } f_{20} = 34.64, \quad f_{21} = 18.28, f_{22} = 12.42 .$$

Therefore, the linearized objective function is

$$f(x_1, x_2) = 20\eta_{10} + 9.76\eta_{11} + 6.47\eta_{12} + 34.64\eta_{20} + 18.28\eta_{21} + 12.42\eta_{22} .$$

After substituting

$$x_1 = 0.05\eta_{10} + 0.1023\eta_{11} + 0.1546\eta_{12}\,,$$

$$\text{and } x_2 = 0.05\eta_{20} + 0.09475\eta_{21} + 0.1546\eta_{22}\,,$$

into the constraint equations of (3.3.11) through (3.3.13), a standard LP algorithm can be applied with the additional stipulation that only two adjacent η_{ik} for every design variable x_i be positive. • • •

3.4 Graphical Solutions of Simple LP Problems

For simple problems with no more than two design variables a graphical solution technique may be used to find the solution of a LP problem. A graphical method not only gives a solution, but also helps us to understand the nature of LP problems. The following example is included in order to illustrate the nature of the design space and the optimal solution.

Example 3.4.1

Consider the portal frame limit design problem of example 3.1.5. The problem was reduced to minimizing the objective function

$$f(x_1, x_2) = 2x_1 + x_2\,, \tag{3.4.1}$$

subject to inequality constraints Eqs. (3.1.26) through (3.1.32).

Since the problem is an LP problem in two-dimensional space it is possible to obtain a graphical solution. Constraints (3.1.32) imply that we can restrict ourselves to the non-negative quadrant of the $x_1 - x_2$ plane in Figure 3.4.1. We plot all the straight lines corresponding to Eqs. (3.1.26) through (3.1.31) as strict equalities (these lines identify the constraint boundaries). To identify the feasible and the infeasible portions on either side of a given constraint line we choose a point on either side and substitute its coordinates in the inequality. If the inequality is satisfied then the portion on the side of the constraint line which contains this point is the feasible portion, if not it is infeasible. For example, if the coordinates $x_1 = 0$ and $x_2 = 0$ are substituted into the inequality (3.1.27), the inequality is violated, implying that the origin does not belong to the feasible domain. If we continue this process for all the inequality constraints we will soon end up with a feasible region that is a convex polygon; the corners are called extreme points. The feasible region corresponding to the constraints is illustrated in Figure 3.4.1.

Next, we plot the contours of the objective function by setting the function $2x_1 + x_2$ equal to a constant and plotting the lines corresponding to various values of this constant. The optimum point is obtained by finding the contour of the objective function which just barely touches the feasible region. The direction of decreasing f is shown in Figure 3.4.1 with the optimum solution identified as

$$x_1 = x_2 = 1/2\,, \tag{3.4.2}$$

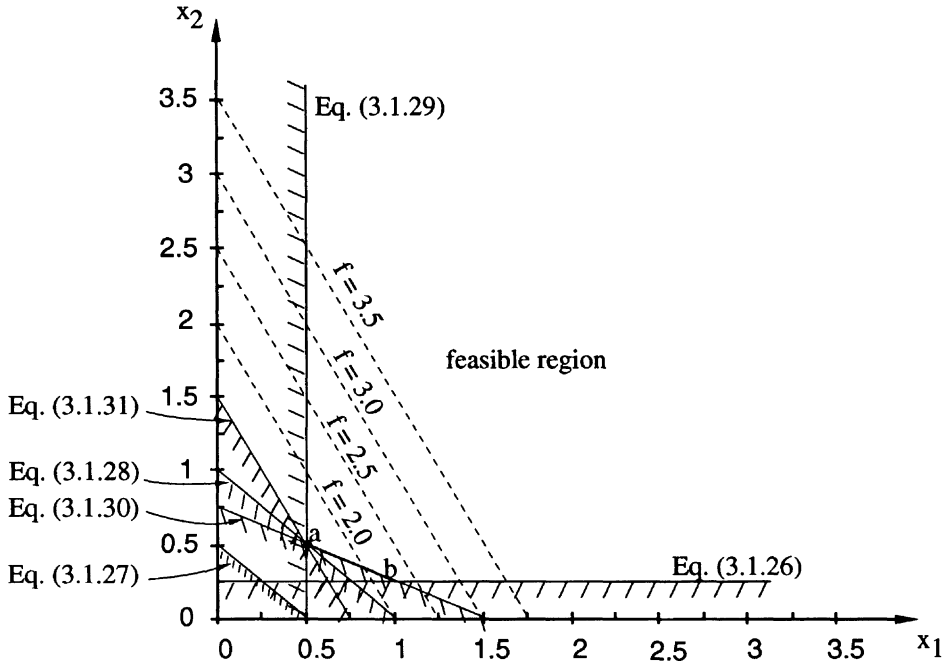


Figure 3.4.1 Graphical solution of the portal frame LP problem.

with $f_{\min} = 1.5$. • • •

Barring degeneracy, the optimum solution in an LP problem will always lie at a corner or an extreme point. The degenerate case may occur when the gradient of the objective function is a constant multiple of the gradient of one of the constraints along which the optimum solution lies. Then, every point along this constraint including the extreme points constitutes an optimum solution. For example if the problem just discussed had an objective function of the type

$$f = c(2x_1 + 4x_2)\,, \tag{3.4.3}$$

with c being a constant, then every point along the line [a,b] in Figure 3.4.1 would constitute an optimum solution.

The concept of a convex polygon with corners or vertices in two dimensions generalizes to a *convex polytope* with *extreme points* in $\mathbf{R}^n$. For example, a convex polytope [11] is defined to be the set which is obtained by the intersection of a finite number of closed half-spaces. Similarly, an extreme point of a set is defined to be a point $\mathbf{x}$ in $\mathbf{R}^n$ which cannot be expressed as a convex combination $\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ $(0 < \alpha < 1)$ of two distinct points $\mathbf{x}_1$ and $\mathbf{x}_2$ belonging to the set. Finally, as in the two-dimensional case of Figure 3.4.1, barring degeneracy, a linear objective function in $\mathbf{R}^n$ achieves its minimum only at an extreme point of a bounded convex polytope.

Interested readers are advised to consult either Ref. 11 or 13 for a comprehensive treatise on this subject.

It is obvious that the above graphical procedure cannot be used for linear programming problems involving more than two variables. We have to look at alternative means of solving such problems. The simplex method first proposed by Dantzig [13] is an efficient method for solving problems with a large number of variables and constraints. We will study the simplex method next and to this end we outline a few definitions and some very important concepts in linear programming.

3.5 A Linear Program in a Standard Form

A linear program is said to be in a standard form if it is posed as

$$\textbf{minimize} \qquad f = \mathbf{c}^T\mathbf{x} \tag{3.5.1}$$

$$\textbf{subject to} \qquad \mathbf{A}\mathbf{x} = \mathbf{b}\,, \tag{3.5.2}$$

$$\mathbf{x} \geq \mathbf{0}\,, \tag{3.5.3}$$

where $\mathbf{c}$ is an $n \times 1$ vector, $\mathbf{A}$ is a $m \times n$ matrix, and $\mathbf{b}$ is a $m \times 1$ vector. Any linear program including inequality constraints can be put into the standard form by the use of what are known as *slack* and *surplus* variables. Consider, for example, the linear program defined by Eqs. (3.1.26) through (3.1.32). We can transform those inequalities into strict equalities as

$$4x_2 - x_3 = 1\,, \tag{3.5.4}$$

$$2x_1 + 2x_2 - x_4 = 1\,, \tag{3.5.5}$$

$$x_1 + x_2 - x_5 = 1\,, \tag{3.5.6}$$

$$2x_1 - x_6 = 1\,, \tag{3.5.7}$$

$$2x_1 + 4x_2 - x_7 = 3\,, \tag{3.5.8}$$

$$4x_1 + 2x_2 - x_8 = 3\,, \tag{3.5.9}$$

by the addition of the surplus variables x_3 through x_8, provided that these variables are restricted to be non-negative, that is

$$x_i \geq 0, \qquad i = 1, \ldots, 8\,. \tag{3.5.10}$$

If the inequalities in Eqs. (3.1.26) through (3.1.31) were of the opposite kind we would add non-negative variables x_3 through x_8 to achieve equality constraints. In this case the variables x_3 through x_8 would be referred to as the slack variables. If the original values of the design variables are not required to be non-negative we can still convert the problem to a standard form of Eqs. (3.5.1) through (3.5.3) by defining either

$$x_1 = u_1 - v_1, \qquad \text{and} \qquad x_2 = u_2 - v_2\,, \tag{3.5.11}$$

where $u_1, u_2, v_1, v_2 \geq 0$, or by adding a large enough positive constant M to the design variable

$$\bar{x}_1 = M + x_1\,, \tag{3.5.12}$$

so that the new variable never becomes negative during the design. Such artificial variables are often used in structural design problems where quantities such as stresses are used as design variables. Stresses can be both positive or negative depending upon the loading condition. It is clear from Eq. (3.5.11) that putting LP program in a standard form may cause an increase in the dimension of the design space. Using Eq. (3.5.12) does not increase the dimension of the problem but it may be difficult to know a priori the value of the constant M that will make the design variable positive (the choice of a very large number may result in numerical ill-conditioning).

Going back to Eq. (3.5.2) we notice that if $m = n$ and all the equations are linearly independent, we have a unique solution to the system of equations, whereas with $m > n$ we have, in general, an inconsistent system of equations. It is only when $m < n$ that we have many possible solutions. Of all these solutions we seek the one which satisfies the non-negativity constraints and minimizes the objective function f.

3.5.1 *Basic Solution*

We assume the rank of the matrix $\mathbf{A}$ to be m and select from the n columns of $\mathbf{A}$ a set of m linearly independent columns. We denote this $m \times m$ matrix by $\mathbf{D}$. Then $\mathbf{D}$ is non-singular and we can obtain the solution

$$\underset{m \times 1}{\mathbf{x_D}} = \underset{m \times m}{\mathbf{D}^{-1}} \; \underset{m \times 1}{\mathbf{b_D}}, \tag{3.5.13}$$

where $\mathbf{x_D}$ is the vector of independent variables and $\mathbf{b_D}$ is the corresponding right-hand vector. Thus it can easily be verified that

$$\mathbf{x} = \begin{Bmatrix} \mathbf{x_D} \\ \cdots \\ \mathbf{0} \end{Bmatrix}, \tag{3.5.14}$$

is a solution of the system of Eqs. (3.5.2). Such a solution is known as a basic solution, and $\mathbf{x_D}$ is called the vector of basic variables. A basic solution, however, need not satisfy the non-negativity constraints (3.5.3). Those basic solutions which do indeed satisfy these constraints are known as basic feasible solutions and can be shown to be extreme points. In other words all basic feasible solutions to Eqs. (3.5.2) will correspond to corners or extreme points of the convex polytope [13].

The total number of possible basic solutions to Eqs. (3.5.2) can be estimated by identifying the number of possibilities for selecting m variables arbitrarily from a group of n variables. From the theory of permutations and combinations we know this number to be

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}. \tag{3.5.15}$$

Not all of these possibilities will however be feasible.

3.6 The Simplex Method

The idea of the simplex method is to continuously decrease the value of the objective function by going from one basic feasible solution to another until the minimum value of the objective function is achieved. We will postpone the discussion of how to generate a basic feasible solution and assume that we have a basic feasible solution to start the algorithm. Indeed, if we had the following inequality constraints

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i, \qquad i = 1, \ldots, m, \tag{3.6.1}$$

$$x_j \geq 0, \qquad j = 1, \ldots, n, \tag{3.6.2}$$

where $b_i \geq 0$ for every constraint, then the process of converting the constraint set to the standard form yields the following

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + y_i = b_i, \qquad i = 1, \ldots, m, \tag{3.6.3}$$

$$x_j \geq 0, \qquad j = 1, \ldots, n, \tag{3.6.4}$$

$$y_i \geq 0, \qquad i = 1, \ldots, m, \tag{3.6.5}$$

and we immediately recognize

$$y_i = b_i, \quad i = 1, \ldots, m, \quad \text{and} \quad x_j = 0, \quad j = 1, \ldots, n, \tag{3.6.6}$$

as a basic feasible solution. A formal scheme for generating a basic feasible solution will be discussed later in this section. The question of immediate interest at this point is how to go from one basic feasible solution to another basic feasible solution. Without loss of generality let us assume that we have a system of equations in the canonical form shown below (such forms can always be obtained through the well-known Gauss elimination scheme for a matrix $\mathbf{A}$ with rank m).

$$\begin{array}{ccccccccccccc}
x_1 & +0 & +\ldots & +0 & +\ldots & +0 & +a_{1,m+1}\,x_{m+1} & +\ldots & +a_{1,n}\,x_n & = & b_1 \\
0 & +x_2 & +\ldots & +0 & +\ldots & +0 & +a_{2,m+1}\,x_{m+1} & +\ldots & +a_{2,n}\,x_n & = & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & +0 & +\ldots & +x_s & +\ldots & +0 & +a_{s,m+1}\,x_{m+1} & +\ldots & +a_{s,n}\,x_n & = & b_s \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & +0 & +\ldots & +0 & +\ldots & +x_m & +a_{m,m+1}\,x_{m+1} & +\ldots & +a_{m,n}\,x_n & = & b_m
\end{array} , \tag{3.6.7}$$

with a basic feasible solution

$$x_1 = b_1, \quad x_2 = b_2, \quad \ldots \quad x_s = b_s, \quad \ldots \quad x_m = b_m,$$

$$x_{m+1} = x_{m+2} = \ldots = 0. \tag{3.6.8}$$

The variables x_1 through x_m are called basic and the x_{m+1} through x_n are called non-basic variables.

3.6.1 *Changing the Basis*

The simplex procedure changes the set of basic variables while improving the objective function at the same time. However, for the purpose of clarity we will first demonstrate the approach for going from one basic feasible solution to another. The objective function improvement will be discussed in the following section.

We wish to make one of the current non-basic variables of Eq. (3.6.7), say x_t ($m < t \leq n$), basic and in the process cause a basic variable, $x_s (1 \leq s \leq m)$, to become non-basic. At this point we assume that we know the variable x_t which we will bring into the basic set. We only need to decide which variable to drop from the basic set. Consider the selected terms shown below for the coefficients of the sth equation and an additional arbitrary ith equation.

$$\begin{array}{cccccccccc} & i & & s & & t & & & \\ i & 1 & \ldots & 0 & \ldots & a_{it} & \ldots & = & b_i \\ & \vdots & & \vdots & & \vdots & & & \vdots \\ s & 0 & \ldots & 1 & \ldots & a_{st} & \ldots & = & b_s \end{array} \tag{3.6.9}$$

Since we want to make x_t basic, we need to eliminate it from the rest of the equations except the sth one by reducing the coefficients a_{it} ($i = 1, \ldots, n; \quad i \neq s$) to zeroes, and making the coefficient a_{st} unity by dividing the sth equation throughout by a_{st}. We can do this only if a_{st} is non-zero. Also, unless a_{st} is positive, the process of dividing the sth equation by a_{st} will produce a negative term on the right-hand side since b_s is positive because the current solution is a basic feasible solution. To eliminate the new basic variable x_t from the ith equation ($i = 1, \ldots, n; \quad i \neq s$) we have to multiply the sth equation by the factor (a_{it}/a_{st}) and subtract the resulting equation from each of these equations. The resulting coefficients on the right-hand side of the ith equation will be

$$b_i' = b_i - b_s(\frac{a_{it}}{a_{st}}) \, . \tag{3.6.10}$$

To guarantee that the resulting solution is a basic feasible solution we must require that $b_i' \geq 0$, or rearranging Eq. (3.6.10) we have

$$(\frac{b_s}{a_{st}}) \leq (\frac{b_i}{a_{it}}) \, . \tag{3.6.11}$$

Equation (3.6.11) together with the condition

$$a_{st} > 0 \, , \tag{3.6.12}$$

are the two conditions which identify possible sth rows in changing from one basic feasible solution to another. Thus for a given non-basic variable x_t that is to be made basic we check the coefficients of all the terms in the tth column. We eliminate from consideration all elements in the tth column with non-positive coefficients as violating condition (3.6.12). Among those with positive coefficients we compute the ratios b_i/a_{it} ($i = 1, \ldots, n$). We select the row, s, for which the ratio b_i/a_{it} has the smallest value and call it b_s/a_{st}, Eq. (3.6.11). It is the basic variable corresponding to that row which will become non-basic in the process of making x_t basic.

Example 3.6.1

We illustrate the foregoing discussion with an example. Consider the system of equations

$$\begin{aligned} 2x_1 + 2x_2 + x_3 &= 6\,, \\ 3x_1 + 4x_2 + x_4 &= 10\,, \\ x_1 + 2x_2 + x_5 &= 4\,. \end{aligned} \tag{3.6.13}$$

The system is already in the canonical form with a basic feasible solution being

$$x_1 = x_2 = 0, \qquad x_3 = 6, \quad x_4 = 10, \quad x_5 = 4\,. \tag{3.6.14}$$

The variables x_1 and x_2 are the non-basic variables, whereas $x_3, x_4,$ and $, x_5$ are the basic variables. Now, let us assume that we want to make x_1 basic. Rewriting Eqs. (3.6.13) in a matrix form we have

$$\begin{bmatrix} 2 & 2 & 1 & 0 & 0 \\ 3 & 4 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 10 \\ 4 \end{Bmatrix}\,. \tag{3.6.15}$$

Since x_1 is to made basic we consider the first column. To chose the variable to be made non-basic we form the ratios (b_i/a_{i1}), $i = 1, 2, 3$.

$$\frac{b_1}{a_{11}} = 3, \qquad \frac{b_2}{a_{21}} = 3\frac{1}{3}, \qquad \frac{b_3}{a_{31}} = 4\,.$$

The smallest ratio is b_1/a_{11} and so we pivot on a_{11}. Thus the new system of equations is

$$\begin{bmatrix} 1 & 1 & 0.5 & 0 & 0 \\ 0 & 1 & -1.5 & 1 & 0 \\ 0 & 1 & -0.5 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix} = \begin{Bmatrix} 3 \\ 1 \\ 1 \end{Bmatrix}\,, \tag{3.6.16}$$

and the process of making x_1 basic has resulted in the variable x_3 being non-basic. The new feasible solution is

$$x_2 = x_3 = 0, \qquad x_1 = 3, \quad x_4 = 1, \quad x_5 = 1\,.$$

It may be verified by the reader that by using a pivot other than a_{11} we would end up with an infeasible basic solution. For example, if a_{13} is a pivot we obtain

$$x_2 = x_5 = 0, \qquad x_1 = 4, \quad x_3 = -2, \quad x_4 = -2\,,$$

which is not feasible since $x_3 < 0$ and $x_4 < 0$. • • •

3.6.2 *Improving the Objective Function*

In the preceding section we considered making a particular non-basic variable x_t basic without losing feasibility. We also need to decide the variable that we make basic. We should seek to bring into the basis only that variable which will decrease the objective function while yielding at the same time a basic feasible solution. Notice that the objective function is a linear equation just like the other equations and hence it can be included with the others. The objective function equation may be written as

$$\mathbf{c}^T\mathbf{x} = f \ . \tag{3.6.17}$$

Assume the system of equations (3.5.2) is in the canonical form, and append Eq. (3.6.17) at the end of all other equations. The form of the equations that includes the objective function is often referred as the *simplex tableau.* We now eliminate all the basic variables from this last equation by subtracting c_i times each of the equations in the canonical form. Then the right-hand of Eq. (3.6.17) becomes $(f - c_1b_1 - c_2b_2 - c_3b_3 - \ldots - c_mb_m)$. Thus if we ignore the presence of f, the right-hand side represents the negative of the value of the objective function since $x_{m+1} = x_{m+2} = \ldots = x_n = 0$. The left-hand side of this last equation will contain only non-basic variables. Next, assume that the coefficient of one of the non-basic variables on the left-hand side of the last equation is negative. If we make this variable basic then we will increase the value of this variable from its present value of zero to some positive value. Since the last equation is just one of the equations, when we pivot on one of the equations (sth) and eliminate the corresponding variable (x_s) from the basic set we perform the operations described in the previous section on all the $m + 1$ equations. When the particular variable with the negative coefficient in the last equation is eliminated, the right-hand side of this equation will increase since the variable has increased in value from zero to a positive value. Since the right-hand side represents the negative of the value of the objective function, a function decrease is therefore guaranteed. Thus the criterion for guaranteeing an improvement of the objective function is to bring into the basis a variable that has a negative coefficient in the objective function equation after it has been cleared of all the basic variables. This can be verified by the following example.

Example 3.6.2

$$\begin{aligned}
&\textbf{minimize} && f = x_1 + x_2 + x_3 && (3.6.18)\\
&\textbf{subject to} && 2x_1 + 2x_2 + x_3 = 6\,, && (3.6.19)\\
& && 3x_1 + 4x_2 + x_4 = 10\,, && (3.6.20)\\
& && x_1 + 2x_2 + x_5 = 4\ . && (3.6.21)
\end{aligned}$$

As mentioned above we rewrite the constraint equations (3.6.21) in the matrix form together with the objective function appended as the last row of the matrix

$$\left[\begin{array}{ccccc} 2 & 2 & 1 & 0 & 0 \\ 3 & 4 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \\ -- & -- & -- & -- & -- \\ 1 & 1 & 1 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array}\right\} = \left\{\begin{array}{c} 6 \\ 10 \\ 4 \\ -- \\ 0 \end{array}\right\} \ . \tag{3.6.22}$$

A basic solution is

$$x_1 = x_2 = 0, \qquad x_3 = 6, \quad x_4 = 10, \quad x_5 = 4\ . \tag{3.6.23}$$

The variable x_3 is a basic variable that appears in the last equation of Eqs. (3.6.22) and must be eliminated from it so that its right-hand side yields the negative of the current value of the objective function.

$$\left[\begin{array}{ccccc} 2 & 2 & 1 & 0 & 0 \\ 3 & 4 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \\ -- & -- & -- & -- & -- \\ -1 & -1 & 0 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array}\right\} = \left\{\begin{array}{c} 6 \\ 10 \\ 4 \\ -- \\ -6 = -f \end{array}\right\}\ . \tag{3.6.24}$$

We can pivot either on column (1) or column (2). That is to say the objective function will decrease in value by bringing either x_1 or x_2 into the basis. If we pivot on column (1) (bringing x_1 into the basis) the pivot element is a_{11} because it yields the smallest (b_i/a_{i1}) ratio. The new simplex tableau becomes

$$\left[\begin{array}{ccccc} 1 & 1 & 0.5 & 0 & 0 \\ 0 & 1 & -1.5 & 1 & 0 \\ 0 & 1 & -0.5 & 0 & 1 \\ -- & -- & -- & -- & -- \\ 0 & 0 & 0.5 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array}\right\} = \left\{\begin{array}{c} 3 \\ 1 \\ 1 \\ -- \\ -3 = -f \end{array}\right\}\ . \tag{3.6.25}$$

The value of the objective function has been reduced from 6 to 3. Since the last equation contains no non-basic variable with a negative coefficient, it is no longer possible to decrease the value of the objective function further. Thus the minimum value of the objective function is 3 and corresponds to the basic solution

$$x_2 = x_3 = 0, \qquad x_1 = 3, \quad x_4 = 1, \quad x_5 = 1\ . \tag{3.6.26}$$

If we had decided to bring x_2 into the basis first, we would have reduced the objective function from 6 to 4, and there would have been a negative number in the last equation in the first column indicating the need for another round of pivoting to bring x_1 into the basis. • • •

This would have completed the discussion of the simplex method except for the fact that we need a basic feasible solution to start the simplex method and we may not have one readily available. This is our next topic.

3.6.3 *Generating a Basic Feasible Solution—Use of Artificial Variables*

In the process of converting an LP problem given in the form of Eqs. (3.6.4) and (3.6.5)

$$\mathbf{Ax} \leq \mathbf{b}, \qquad \text{where} \qquad \mathbf{b} > \mathbf{0}, \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}, \tag{3.6.27}$$

into the standard form by adding slack variables we obtained a basic feasible solution to start the simplex method. However, when we have a linear program which is

already in the standard form of Eqs. (3.5.2) and (3.5.3) we cannot, in general, identify a basic feasible solution. The following technique can be used in such cases.

Consider the following minimization problem

$$\textbf{minimize} \quad \sum_{i=1}^{m} y_i \tag{3.6.28}$$

$$\textbf{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b}, \tag{3.6.29}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \text{and} \quad \mathbf{y} \geq \mathbf{0}, \tag{3.6.30}$$

where $\mathbf{y}$ is a vector of artificial variables. There is no loss of generality in assuming that $\mathbf{b} > \mathbf{0}$ so that the LP problem (3.6.29) has a known basic feasible solution

$$\mathbf{y} = \mathbf{b}, \quad \text{and} \quad \mathbf{x} = \mathbf{0}, \tag{3.6.31}$$

so that the simplex method can be easily applied to solve the LP problem of Eqs. (3.6.30). Note that if a basic feasible solution to the original LP problem (3.6.28) exists then the optimum solution to the modified problem (3.6.30) must have $\mathbf{y}_i$'s as non-basic variables ($\mathbf{y} = \mathbf{0}$). However, if no basic feasible solution to the original problem exists then the minimum value of Eq (3.6.29) will be greater than zero.

Example 3.6.3

We illustrate the use of artificial variables with the following example for which we seek a basic feasible solution to the system

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &= 13, \\ x_1 + 2x_2 + x_3 &= 7, \\ x_i \geq 0, \qquad i &= 1,2,3\,. \end{aligned} \tag{3.6.32}$$

Introduce the artificial variables $y_1$and y_2 and pose the following minimization problem.

$$\textbf{minimize} \quad f = y_1 + y_2 \tag{3.6.33}$$

$$\begin{aligned} \textbf{subject to} \quad & 2x_1 + x_2 + 3x_3 + y_1 = 13, \\ & x_1 + 2x_2 + x_3 + y_2 = 7, \\ & x_i \geq 0, \quad i = 1,2,3, \text{ and} \qquad y_i \geq 0, \; j = 1,2\,. \end{aligned} \tag{3.6.34}$$

With the basic feasible solution, $y_1 = 13$, $y_2 = 7$, and $x_1 = x_2 = x_3 = 0$ known, we append the objective function (3.6.33) and clear the basic design variables y_1 and y_2 from it to obtain the initial simplex tableau

$$\left[\begin{array}{ccccc} 2 & 1 & 3 & 1 & 0 \\ 1 & 2 & 1 & 0 & 1 \\ -- & -- & -- & -- & -- \\ -3 & -3 & -4 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{array}\right\} = \left\{\begin{array}{c} 13 \\ 7 \\ -- \\ -20 \end{array}\right\}. \tag{3.6.35}$$

Since it has the largest negative number we choose column (3) for pivoting with a_{13} as the pivot element since $13/3 < 7/1$,

$$\begin{bmatrix} 2/3 & 1/3 & 1 & 1/3 & 0 \\ 1/3 & 5/3 & 0 & -1/3 & 1 \\ -- & -- & -- & -- & -- \\ -1/3 & -5/3 & 0 & 4/3 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} 13/3 \\ 8/3 \\ -- \\ -8/3 \end{Bmatrix} . \tag{3.6.36}$$

Next we choose a_{22} as the pivot element to obtain

$$\begin{bmatrix} 9/15 & 0 & 1 & 6/15 & -1/5 \\ 1/5 & 1 & 0 & -1/5 & 3/5 \\ -- & -- & -- & -- & -- \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} 19/5 \\ 8/5 \\ -- \\ 0 \end{Bmatrix} . \tag{3.6.37}$$

The process has converged to the basic feasible solution

$$x_1 = 0, \quad x_2 = 8/5, \quad \text{and} \quad x_3 = 19/5 . \tag{3.6.38}$$

to the original problem. • • •

3.7 Duality in Linear Programming

It was shown by Dantzig [13] that the primal problem of minimization of a linear function over a set of linear constraints is equivalent to the dual problem of the maximization of another linear function over another set of constraints. Both the dual objective function and constraints of the dual problem are obtained from the objective function and constraints of the primal problem. Thus if the primal problem is defined to be

$$\begin{aligned} \textbf{minimize} \quad & f_p = c_1 x_1 + \ldots\ldots + c_n x_n = \mathbf{c}^{\mathbf{T}}\mathbf{x} && (n \text{ variables}) \\ \textbf{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m , && (m \text{ constraints}) \\ & x_j \geq 0, \quad j = 1, \ldots, n , && (3.7.1) \end{aligned}$$

then the dual problem is defined to be

$$\begin{aligned} \textbf{maximize} \quad & f_d = b_1 \lambda_1 + \ldots\ldots + b_m \lambda_m = \mathbf{b}^{\mathbf{T}}\boldsymbol{\lambda} && (m \text{ variables}) \\ \textbf{subject to} \quad & \sum_{i=1}^{m} a_{ij} \lambda_i \leq c_j, \quad j = 1, \ldots, n , && (n \text{ constraints}) \\ & \lambda_i \geq 0, \quad j = 1, \ldots, m . && (3.7.2) \end{aligned}$$

The choice of the primal or dual formulation depends on the number of design variables and the number of constraints. The computational effort in solving an LP

problem increases as the number of constraints increases. Therefore, if the number of constraint relations is large compared to the number of design variables then it may be desirable to solve the dual problem which will require less computational effort. The classification of problems into the primal and dual categories is, however, arbitrary since if the maximization problem is defined as the primal then the minimization problem is its dual. It can be shown [13] that the optimal values of the basic variables of the primal can be obtained from the solution of the dual and that $(f_p)_{\min} = (f_d)_{\max}$. Thus if x_j is a basic variable in the primal problem, then it implies that the jth constraint of the dual problem is active and vice versa.

If the primal problem is stated in its standard form; namely with equality constraints

$$\begin{aligned} \textbf{minimize} \quad & f_p = c_1 x_1 + \ldots\ldots + c_n x_n = \mathbf{c}^{\mathbf{T}}\mathbf{x} && (n \text{ variables}) \\ \textbf{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m\,, && (m \text{ constraints}) \\ & x_j \geq 0, \quad j = 1, \ldots, n\,, && (3.7.3) \end{aligned}$$

then the corresponding dual problem is

$$\begin{aligned} \textbf{maximize} \quad & f_d = b_1 \lambda_1 + \ldots\ldots + b_m \lambda_m = \mathbf{b}^{\mathbf{T}}\boldsymbol{\lambda} && (m \text{ variables}) \\ \textbf{subject to} \quad & \sum_{i=1}^{m} a_{ij} \lambda_i \leq c_j, \quad j = 1, \ldots, n\,, && (n \text{ constraints}) \\ & && (3.7.4) \end{aligned}$$

with the variables λ_i being unrestricted in sign [11].

It should be noted that, in practice, it is rare for a LP problem to be solved either as a primal or as a dual problem. Most state-of-the-art LP software employ what is known as a primal-dual algorithm. This algorithm begins with a feasible solution to the dual problem that is successively improved by optimizing an associated restricted primal problem. The details of this algorithm are beyond the scope of this book and interested readers should consult Ref. [11].

Example 3.7.1

As an example of the simplex method for solving an LP problem via the dual formulation we use the portal frame problem formulated in Example 3.1.5 with a slightly different loading condition. The new loading condition is assumed to correspond to a 25% increase in the magnitude of the horizontal load while keeping the magnitude of the vertical load the same. The corresponding constraint equations have different right-hand sides than those given in Eqs. (3.5.4) through (3.5.9), namely

$$\begin{aligned} 4x_2 &\geq 1\,, \\ 2x_1 + 2x_2 &\geq 1\,, \\ x_1 + x_2 &\geq 1.25\,, \\ 2x_1 &\geq 1.25\,, \\ 2x_1 + 4x_2 &\geq 3.5\,, \\ 4x_1 + 2x_2 &\geq 3.5\,. \end{aligned} \qquad (3.7.5)$$

However, when put into the standard form, not only does the problem involve a total of 8 variables, but also a basic feasible solution to the problem is not immediately obvious. Because the objective function (3.1.25) involves only two variables x_1 and x_2 the solution of the dual problem may be more efficient. The dual problem is

$$\begin{aligned} \textbf{maximize} \quad & f_d = \lambda_1 + \lambda_2 + 1\frac{1}{4}\lambda_3 + 1\frac{1}{4}\lambda_4 + 3\frac{1}{2}\lambda_5 + 3\frac{1}{2}\lambda_6 && (3.7.7) \\ \textbf{subject to} \quad & 2\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 + 4\lambda_6 \leq 2\,, \\ & 4\lambda_1 + 2\lambda_2 + \lambda_3 + 4\lambda_5 + 2\lambda_6 \leq 1\,, && (3.7.8) \\ & \lambda_i \geq 0, \quad i = 1, \ldots, 6\,. \end{aligned}$$

Maximizing f_d is same as minimizing $-f_d$ and the process of converting the above linear problem to the standard form yields

$$\begin{aligned} \textbf{minimize} \quad & -f_d = -\lambda_1 - \lambda_2 - 1\frac{1}{4}\lambda_3 - 1\frac{1}{4}\lambda_4 - 3\frac{1}{2}\lambda_5 - 3\frac{1}{2}\lambda_6 && (3.7.9) \\ \textbf{subject to} \quad & 2\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 + 4\lambda_6 + \lambda_7 = 2\,, \\ & 4\lambda_1 + 2\lambda_2 + \lambda_3 + 4\lambda_5 + 2\lambda_6 + \lambda_8 = 1\,, && (3.7.10) \\ & \lambda_i \geq 0, \quad i = 1, \ldots, 8\,, \end{aligned}$$

with the basic feasible solution

$$\lambda_i = 0, \quad i = 1, \ldots, 6\,, \quad \text{and} \qquad \lambda_7 = 2, \quad \lambda_8 = 1\,.$$

We can begin with the initial simplex tableau with the basic variables cleared from the last equation which represents the objective function.

$$\left[\begin{array}{cccccccc} 0 & 2 & 1 & 2 & 2 & 4 & 1 & 0 \\ 4 & 2 & 1 & 0 & 4 & 2 & 0 & 1 \\ -- & -- & -- & -- & -- & -- & -- & -- \\ -1 & -1 & -5/4 & -5/4 & -7/2 & -7/2 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{array}\right\} = \left\{\begin{array}{c} 2 \\ 1 \\ -- \\ 0 \end{array}\right\}\,. \tag{3.7.11}$$

Although we should perhaps be choosing fifth or sixth column for pivoting, since it has the largest negative value, pivoting on third column produces the same final answer with one less simplex tableau. Pivoting on element a_{23} we have

$$\left[\begin{array}{cccccccc} -4 & 0 & 0 & 2 & -2 & 2 & 1 & -1 \\ 4 & 2 & 1 & 0 & 4 & 2 & 0 & 1 \\ -- & -- & -- & -- & -- & -- & -- & -- \\ 4 & 3/2 & 0 & -5/4 & 3/2 & -1 & 0 & 5/4 \end{array}\right] \left\{\begin{array}{c} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{array}\right\} = \left\{\begin{array}{c} 1 \\ 1 \\ -- \\ 5/4 \end{array}\right\}\,. \tag{3.7.12}$$

Because of the presence of negative terms in the last equation, it is clear that the objective function can still be decreased further. Pivoting on element a_{14} we obtain

$$\begin{bmatrix} -2 & 0 & 0 & 1 & -1 & 1 & 1/2 & -1/2 \\ 4 & 2 & 1 & 0 & 4 & 2 & 0 & 1 \\ -- & -- & -- & -- & -- & -- & -- & -- \\ 3/2 & 3/2 & 0 & 0 & 1/4 & 1/4 & 5/8 & 5/8 \end{bmatrix} \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{Bmatrix} = \begin{Bmatrix} 1/2 \\ 1 \\ -- \\ 15/8 \end{Bmatrix} . \quad (3.7.13)$$

Hence we conclude that $(f_d)_{\min} = -15/8$ or $(f_d)_{\max} = (f_p)_{\min} = 15/8$ with the solution

$$\lambda_1 = \lambda_2 = \lambda_5 = \lambda_6 = \lambda_7 = \lambda_8 = 0, \quad \text{and} \quad \lambda_3 = 1, \quad \lambda_4 = 1/2 . \quad (3.7.14)$$

The non-zero λ's indicate that the active constraints in the primal problem are the third and fourth, namely

$$2x_1 = 1.25, \quad \text{and} \quad x_1 + x_2 = 1.25 , \quad (3.7.15)$$

Solution of Eqs. (3.7.15) yields $x_1 = x_2 = 5/8$. • • •

In closing this section, it is interesting to point out that the dual variables can be interpreted as the prices of the constraints. For a given variation on the right hand side $\mathbf{b}$ of the constraint relations of Eq. (3.7.5), the change in the optimum value of the objective function can be determined from

$$\Delta f^* = \boldsymbol{\lambda}^T \Delta \mathbf{b} . \quad (3.7.16)$$

For Eq. (3.7.16) to hold, however, the changes in the $\mathbf{b}$ vector must be such that it does not result in a change in the active constraint set. The dual problem can also be viewed as one of maximization of a *profit* subject to limitations on availability of *resources*. It is clear then that the non-negative dual variables can be interpreted as increased costs which would ensue from a violation of given constraints on resource availabilities. Similarly a primal problem can be viewed as one of minimization of total *cost* while satisfying *demand*. The full significance of dual variables, however, can be brought out more clearly only in the context of the Kuhn-Tucker conditions and the sensitivity of the optimum solutions to changes in design parameters which will be discussed in Chapter 5. The following example demonstrates the use of dual variables to find the sensitivity of the optimal solution to a change in a problem parameter.

Example 3.7.2

Consider the portal frame design problem solved in Example 3.7.1 using dual variables. We will determine the change in the value of the optimum objective function $f^* = 1.875$ corresponding to a 25% reduction in the value of the horizontal force,

keeping the vertical force at p. These loads correspond to the problem formulated in Example 3.1.5 and solved graphically in Example 3.4.1 .

From Eqs. (3.7.5) and (3.1.26) through (3.1.31) the change in the right-hand side is $\Delta b_3 = \Delta b_4 = -\frac{1}{4}$, and $\Delta b_5 = \Delta b_6 = -\frac{1}{2}$. Using the values of the dual variables from Example 3.7.1 in Eq. (3.7.15) we obtain

$$\Delta f^* = -\left(\frac{1}{4}\right) 1 + -\left(\frac{1}{4}\right)\left(\frac{1}{2}\right) = -0.375 \, .$$

Therefore the optimum value of the objective function under this new loading configuration would be $f^* = 1.5$, of course, assuming that the active constraints (the ones associated with non-zero dual variables) remain active. Fortunately, that assumption is correct for the present example. However, beside the two constraints that are active initially there are two more constraints which become active at the new design point (see Fig. 3.4.1). Any reduction larger than 25% in the value of the horizontal load would have caused a change in the active constraint set and resulted in an incorrect answer.

We, therefore, emphasize the fact that in applying Eq. (3.7.15) one has to be cautious not to perturb the design parameter to an extent that the active constraint set changes. This is generally achieved by limiting the parameter perturbations to be small. However, if we had used the design in Example 3.4.1 as our nominal design, no matter how small the perturbation of the magnitude of the horizontal force, the active constraint set would have changed. This is due to the redundancy of the constraints at the optimal solution of Example 3.4.1. • • •

3.8 An Interior Method — Karmarkar's Algorithm

In using the simplex algorithm discussed in section 3.6, we operate entirely along the boundaries of the polytope in $\mathbf{R}^n$ moving from one extreme point (vertex) to another following the shortest path between them, an edge of the polytope. Of all the possible vertices adjacent to the one at which we start, the selection of the next vertex is based on the maximum reduction in the objective function. With these basic premises, the simplex algorithm is only a systematic approach for identifying and examining candidate solutions to the LP problem. The number of operations needed for convergence grows exponentially with the number of variables. In the worst case, the number of operations for convergence for an n variable problem with a set of s constraints can be $s!/n!(s-n)!$. However, it is possible to choose a move direction different from an edge of the polytope, be consistent with the constraint relations, and attain larger gains in the objective function. Although such a choice can lead to a rapid descent toward the optimal vertex, it will do so through intermediate points which are not vertices.

Interior methods of solving LP problems have drawn serious attention only since the dramatic introduction of Karmarkar's algorithm [14] by AT&T Bell Laboratories. This new algorithm was originally claimed to be 50 times faster than the simplex

method. Since then, much work has been invested in improvements and extensions of Karmarkar's algorithm. Developments include demonstration of how dual solutions can be generated during the course of this algorithm [15], and extension of Karmarkar's algorithm to treat upper and lower bounds more efficiently [16] by eliminating the slack variables which are commonly used for such bounds in the Simplex algorithm.

Because some of the recent developments of the algorithm are mathematically involved and beyond the scope of this book, only a general outline of Karmarkar's algorithm are presented in the following sections. At this point we would like to warn the reader that the tools used in the algorithm were originally introduced for minimization of constrained and unconstrained nonlinear functions which are covered in Chapters 4 and 5. Therefore, the reader is advised to read these chapters before proceeding to the next section.

3.8.1 *Direction of Move*

The direction of maximum reduction in the objective function is the direction of steepest descent, which is the direction of the negative of the gradient of the objective function ∇f (see section 4.2.2). For an LP problem posed in its standard form, see Eq. (3.5.1), the gradient direction is,

$$\nabla f = \mathbf{c} \, . \tag{3.8.1}$$

Although we are not limiting the move direction to be an edge of the polytope formed by the constraint surfaces, for an LP problem the move direction cannot be selected simply as the negative of the gradient direction. The direction must be chosen such that the move leads to a point in the feasible region. This can be achieved by using the projection matrix $\mathbf{P}$

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T \, , \tag{3.8.2}$$

derived in section 5.3, where the columns of the matrix $\mathbf{N}$ correspond to the gradient of the constraint equations. Since the constraints are linear functions of the variables, we have $\mathbf{N} = \mathbf{A}^T$. Operating on the gradient vector $-\mathbf{c}$, $\mathbf{P}$ projects the steepest descent direction onto the nullspace of the matrix $\mathbf{A}$. That is, if we start with an initial design point $\mathbf{x}_0$ which satisfies the constraint equation $\mathbf{A}\mathbf{x}_0 = \mathbf{b}$, and move in a direction $-\mathbf{P}\mathbf{c}$ we will remain in the subspace defined by that constraint equation. Note that in numerical application of this projection the matrix product $\mathbf{A}\mathbf{A}^T$ may not actually be inverted, but rather the linear system $\mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{A}\mathbf{c}$ may be solved and then the projected gradient may be calculated by using $\mathbf{P}\mathbf{c} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$. A more efficient and better conditioned procedure based on QR factorization of the matrix $\mathbf{A}$ for the solution of the projection matrix is described in section 5.5 . The following simple example by Strang from reference [17] illustrates graphically the move direction for a three dimensional design space.

Example 3.8.1

Consider the following minimization problem in three design variables,

$$\textbf{minimize} \qquad f = -x_1 - 2x_2 - 3x_3 \tag{3.8.3}$$

$$\textbf{subject to} \quad x_1 + x_2 + x_3 = 1\,, \tag{3.8.4}$$

$$\mathbf{x} \geq \mathbf{0}\,. \tag{3.8.5}$$

Starting at an initial point $\mathbf{x}^{(0)} = (1/3, 1/3, 1/3)^T$ determine the direction of move.

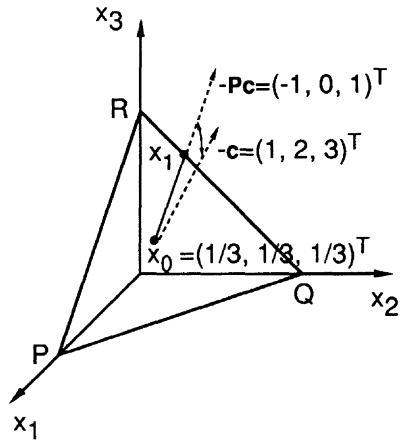


Figure 3.8.1 Design space and move direction.

The design space and the constraint surface for the problem are shown in Figure (3.8.1). The direction corresponding to the negative of the gradient vector is marked as $-\mathbf{c}$. The projection matrix for the problem can be obtained from Eq. (3.8.2) where $\mathbf{A} = [1 \;\; 1 \;\; 1]$. The system $\mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{A}\mathbf{c}$ produces a scalar for $\mathbf{y}$,

$$\{1 \;\; 1 \;\; 1\} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} y = \{1 \;\; 1 \;\; 1\} \begin{Bmatrix} -1 \\ -2 \\ -3 \end{Bmatrix}, \tag{3.8.6}$$

$$y = -2\,.$$

The projected direction $\mathbf{Pc}$ is then given by

$$\mathbf{Pc} = \mathbf{c} - y\mathbf{A}^T\,, \tag{3.8.7}$$

$$\mathbf{Pc} = \begin{Bmatrix} -1 \\ -2 \\ -3 \end{Bmatrix} - \begin{Bmatrix} -2 \\ -2 \\ -2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix}\,. \tag{3.8.8}$$

Moving in a direction $-\mathbf{Pc}$ guarantees maximum reduction in the objective function while remaining in the plane PQR formed by the constraint equation. The minimum value of the objective function for this problem is achieved at the vertex R which, clearly, can not be reached in one iteration. Therefore, the move has to be terminated before the non-negativity requirement is violated (which is at $\mathbf{x}^{(1)} = (2/3, 1/3, 0)^T$),

and the procedure has to be repeated until a reasonable convergence to the minimum point is achieved. • • •

In the preceding example no explanation is provided for the selection of the initial design point, and for the distance travelled in the chosen direction. Karmarkar [14] stops the move before hitting the polytope boundary, say at $\mathbf{x}^{(1)} = (19/30, 1/3, 1/30)^T$ in the previous example, so that there will be room left to move in the next iteration. That is, starting either at the polytope or close to it increases the chances of hitting another boundary before making real gains in the objective function. The solution to this difficulty is accomplished by transforming the design space discussed in the next section.

3.8.2 *Transformation of Coordinates*

In order to focus on the ideas which are important for his algorithm, Karmarkar [14] makes several assumptions with respect to the form of the LP problem. In his canonical representation, the LP problem takes the following form,

$$\textbf{minimize} \qquad f = \mathbf{c}^T\hat{\mathbf{x}} \tag{3.8.9}$$

$$\textbf{subject to} \qquad \mathbf{A}\hat{\mathbf{x}} = 0\,, \tag{3.8.10}$$

$$\mathbf{e}^T\hat{\mathbf{x}} = 1\,, \tag{3.8.11}$$

$$\hat{\mathbf{x}} \geq \mathbf{0}\,, \tag{3.8.12}$$

where $\mathbf{e}$ is a $1 \times n$ vector, $\mathbf{e} = (1, \ldots, 1)^T$. The variable $\hat{\mathbf{x}}$ represents the transformed coordinate such that the initial point is the center, $\hat{\mathbf{x}}^{(0)} = \mathbf{e}/n$, of a unit simplex, and is a feasible point, $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{0}$. A simplex is a generalization to n dimensions of a 2-dimensional triangle and 3-dimensional tetrahedron. A unit simplex has edges of unit length along each of the coordinate directions. Karmarkar also assumes that $\mathbf{c}^T\hat{\mathbf{x}} \geq 0$ for every point that belongs to the simplex, and the target minimum value of the objective function is zero. Conversion of the standard form of an LP problem into this new canonical form can be achieved through a series of operations that involve combining the primal and dual forms of the standard formulation, introducing of slack and artificial variables, and transforming coordinates. The combination of the primal and dual formulations is needed to accommodate the assumption that the target minimum value of the objective function be zero. Details of the formation of this new canonical form is provided in Ref. [14]. In this section we will demonstrate the coordinate transformation which is referred as *projective rescaling transformation.* This is the same transformation that helps to create room for move as we proceed from one iteration to another.

Consider an arbitrary initial point $\mathbf{x}^{(a)}$ in the design space, and let

$$\mathbf{D}_x = \text{Diag}\,(x_1^{(a)}, \ldots, x_n^{(a)})\,. \tag{3.8.13}$$

The transformation, T_x, used by Karmarkar maps each facet of the simplex given by $x_i = 0$ onto the corresponding facet $\hat{x}_i = 0$ in the transformed space, and is given by

$$\hat{\mathbf{x}} = \frac{1}{\mathbf{e}^T\mathbf{D}_x^{-1}\mathbf{x}}\mathbf{D}_x^{-1}\mathbf{x}\,. \tag{3.8.14}$$

While mapping the unit simplex onto itself, this transformation moves the point $\mathbf{x}^{(a)}$ to the center of the simplex, $\hat{\mathbf{x}}^{(0)} = (1/n)\mathbf{e}$. Karmarkar showed that repeated application of this transformation, in the worst case, leads to convergence to the optimal corner in less than $\mathcal{O}(n^{\frac{7}{2}})$ arithmetic operations.

Karmarkar's transformation is nonlinear and a simpler form of this transformation has been suggested. A linear transformation,

$$\hat{\mathbf{x}} = \mathbf{D}_x^{-1}\mathbf{x}, \tag{3.8.15}$$

has been shown to perform as well as Karmarkar's algorithm in practice and to converge in theory [18].

3.8.3 *Move Distance*

Following the transformation, Karmarkar optimizes the transformed objective function over an inscribed sphere of radius $r = 1/(\sqrt{n(n-1)}$ centered at $\hat{\mathbf{x}}^{(0)}$. This is the largest radius sphere that is contained inside the simplex. For the three dimensional design space of Example 3.8.1, for example, where there is one constraint surface, the 'sphere' is a circle in the plane of the constraint equation. In practice, the step length along the projected direction used by Karmarkar is a fraction, α, of the radius. Thus, the new point at the end of the move is given by

$$\mathbf{x}^{(k+1)} = \hat{\mathbf{x}}^{(k)} - \alpha r^{(k)}\mathbf{P}\mathbf{c}^{(k)}, \tag{3.8.16}$$

where $0 < \alpha < 1$. A typical value of α used by Karmarkar is $1/4$.

During the course of the algorithm the optimality of the solution is checked periodically by converting the interior solution to an extreme point solution at the closest vertex. If the extreme point solution is better than the current interior, then, it is tested for optimality.

3.9 Integer Linear Programming

Solution techniques for the LP problems considered so far have been developed under the assumption that the design variables are positive and continuously-valued; they can thus assume any value between their lower and upper bounds. In certain design situations, some or all of the variables of a LP problem are restricted to take discrete values. That is, the standard form of the LP problem of Eq. (3.5.1-3.5.3) takes the form

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ &\textbf{such that} && \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & && x_i \in X_i = \{d_{i1}, d_{i2}, \ldots, d_{il}\}, \qquad i \in I_d, \end{aligned} \tag{3.9.1}$$

where I_d is the set of design variables that can take only discrete values, and X_i is the set of allowable discrete values. Design variables such as cross-sectional areas of

trusses and ply thicknesses of laminated composite plates often fall in this category. Those problems with discrete-valued design variables are called discrete programming problems.

In general, a discrete programming problem can be converted to a form where design variables can assume only integer values. This conversion can be achieved by having the design variable x_i to represent the index j of the $d_{ij}, j = 1, \ldots, l$, Eq. (3.9.1). If the values in the discrete set are uniformly spaced, it is possible to scale the set to form a set of integer values only. The problem is then called an *integer linear programming* (ILP) problem,

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = \mathbf{c}_1^T\mathbf{x} + \mathbf{c}_2^T\mathbf{y} \\ \textbf{such that} \quad & \mathbf{A}_1\mathbf{x} + \mathbf{A}_2\mathbf{y} = \mathbf{b}, \\ & x_i \geq 0 \text{ integer}, \\ & y_j \geq 0 \ . \end{aligned} \tag{3.9.2}$$

This form, where certain design variables are allowed to be continuous, is referred to as *mixed integer linear programming* (MILP) problem. Problems where all variables are integer are called pure ILP problems or in short ILP problems. It is also common to have problems where design variables are used to indicate a 0/1 type decision making situation. Such problems are referred to as *zero/one* or *binary* ILP problems. For example, a truss design problem where the presence of a particular member or the lack of it is represented by a binary variable falls into this category. Any ILP problem with an upper bound on the design variable x_i of $2^K - 1$ can be posed as binary ILP problem by replacing the variable with K binary variables $x_{i1}, \ldots, x_{iK}$ such that

$$x_i = x_{i1} + 2x_{i2} + \ldots + 2^{K-1}x_{iK} \ . \tag{3.9.3}$$

It is also possible to convert the linear discrete programming problem to a binary ILP by using binary variables ($x_{ij} \in \{0, 1\}, j = 1, \ldots, l$) such that

$$x_i = d_{i1}x_{i1} + d_{i2}x_{i2} + \cdots + d_{il}x_{il} \ , \tag{3.9.4}$$

$$\text{and} \qquad x_{i1} + x_{i2} + \cdots + x_{il} = 1 \ . \tag{3.9.5}$$

Most of the following discussion assumes problems to be pure ILP.

A practical approach to solving ILP problems is to round-off the optimum values of the variables, obtained by assuming them to be continuous, to the nearest acceptable integer value. For problems with n design variables there are 2^n possible rounded-off designs, and the problem of choosing the best one is formidable for large n. Furthermore, for some problems the optimum design may not even be one of these rounded-off designs, and for others none of the rounded-off designs may be feasible. A more systematic way of trying possible combinations of variables that will satisfy the requirements of a given problem can be explained by using the *enumeration tree* example of Garfinkel and Nemhauser [19].

Example 3.9.1

Consider the binary ILP problem of choosing a combination of five variables such that the following summation is satisfied

$$f = \sum_{i=1}^{5} ix_i = 5 \ .$$

A decision tree representing the progression of solution of this problem is composed of nodes and branches that represent the solutions and the combinations of variables that lead the those solutions, respectively (Figure 3.9.1). The top node of the tree corresponds to a solution which all the variables are turned off ($x_i = 0, i = 1, \ldots, 5$) with a function value of $f = 0$. Branching off from this solution are two paths corresponding to the two alternatives for the first variable. The branch which has $x_1 = 1$ has a function value of $f = 1$ and tolerates turning additional variables on without running into the risk of exceeding the required function value of 5. Of course the other branch is same as the initial solution, and can be branched further. Next, these two nodes are branched by considering the on and off alternatives for the second variable. The node arrived by taking $x_1 = x_2 = 1$ has $f = 3$ and is terminated as indicated by a vertical line. Such a vertex is said to be *fathomed*, because further branching would mean adding a number that would cause f to exceed its required value of 5. The other three vertices are said to be live, and can be branched further by considering the alternatives for the remaining variables in a sequential manner until either the created nodes are fathomed or the branches arrive at feasible solutions to the problem.

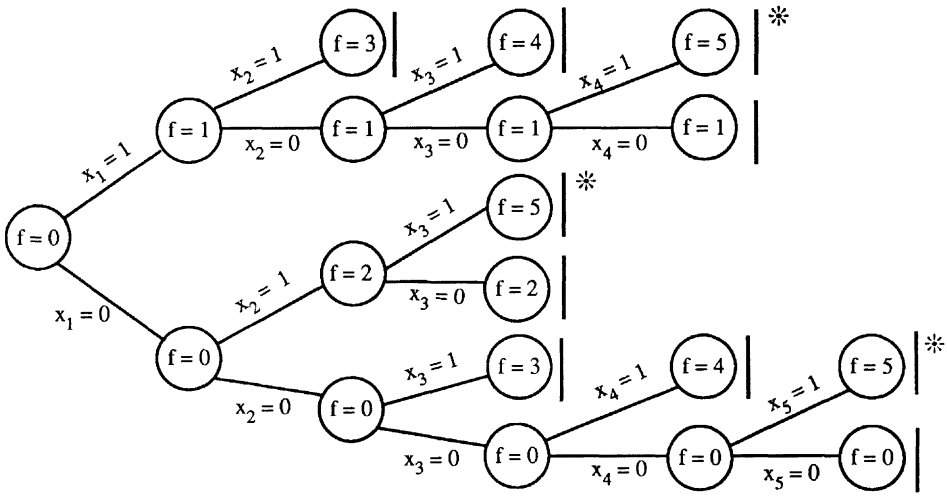


Figure 3.9.1 Enumeration tree for binary ILP problem of $f = \sum_{i=1}^{5} ix_i = 5$.

For the present problem, after considering 19 possible combinations of variables, we identified 3 feasible solutions which are marked by an asterisk. This is a 40% reduction in the total number of possible trials, namely $2^5 = 32$, needed to identify all feasible solutions. For a structural design problem in which trials with different combinations of variables would possibly require expensive analysis an enumeration tree can yield substantial savings. • • •

3.9.1 *Branch-and-Bound Algorithm*

The basic concept behind the enumeration technique forms the basis for this powerful algorithm suitable for MILP problems as well as nonlinear mixed integer problems [20,21]. The original algorithm developed by Land and Doig [22] relies on calculating upper and lower bounds on the objective function so that nodes that result in designs with objective functions outside the bounds can be fathomed and, therefore, the number of analyses required can be cut back. Consider the mixed ILP problem of Eq. (3.9.4). The first step of the algorithm is to solve the LP problem obtained from the MILP problem by assuming the variables to be continuous valued. If all the x variables for the resulting solution have integer values, there is no need to continue, the problem is solved. Suppose several of the variables assume noninteger values and the objective function value is f_1. The f_1 value will form a lower bound $f_L = f_1$ for the MILP since imposing conditions that require any of the noninteger valued variables to take integer values can only cause the objective function to increase. This initial problem is labeled as LP-1 and is placed in the top node of the enumeration tree as shown in Figure (3.9.2). For the purpose of illustration, it is assumed that only two variables x_k and x_{k+1} violate the integer requirement with $x_k = 4.3$ and $x_{k+1} = 2.8$.

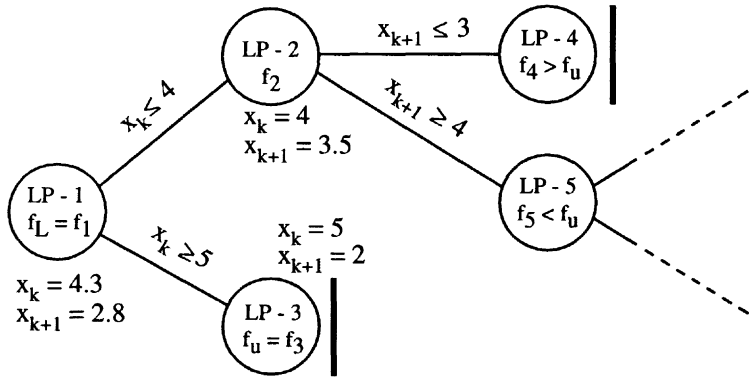


Figure 3.9.2 Branch-and-bound decision tree for ILP problems.

The second step of the algorithm is to branch from the node into two new LP problems by adding a new constraint to the LP-1 that would involve only one of the noninteger variables, say x_k. One of the problems, LP-2, will require the value of the branched variable, x_k to be less than or equal to the largest integer smaller than x_k,

and the other, LP-3, will have a constraint that x_k is larger than the smallest integer larger than x_k. As will be demonstrated later in Example 3.9.2, these two problems actually do branch the feasible design space of the LP-1 into two segments. There are several possibilities for the solution of these two new problems. One of these possibilities is to have no feasible solution for the new problem. In that case the new node will be fathomed. Another possibility is to reach an all integer feasible solution (see LP-3 of Figure 3.9.2) in which case the node will again be fathomed but the value of the objective function will become an upper bound f_U for the MILP problem. That is, beyond this solution point, any node that has an LP solution with a larger value of the objective function will be fathomed, and only those solutions that have the potential of producing an objective function between f_L and f_U will be pursued. If there are no solutions with an objective function smaller than f_U, then the node is an optimum solution. If there are other solutions with an objective function smaller than f_U, they may still include noninteger valued variables (LP-2 of Figure 3.9.2), and are labeled as *live* nodes. Live nodes are then branched again by considering one of the remaining noninteger values and resulting solutions are analyzed until all the nodes are fathomed.

Example 3.9.2

Consider the portal frame problem problem of Example 3.1.5 (see Eqs. (3.1.25) through (3.1.31)) with the requirement that $x_i \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}, i = 1, 2$. We rescale the design variables by a factor of 5 to pose the problem as an integer linear programming problem,

$$\begin{aligned} \textbf{minimize} \quad & f = \frac{1}{5}(2x_1 + x_2) \\ \textbf{such that} \quad & x_2 \geq 1.25\,, \\ & x_1 + x_2 \geq 2.5\,, \\ & x_1 + x_2 \geq 5\,, \\ & x_1 \geq 2.5\,, \\ & x_1 + 2x_2 \geq 7.5\,, \\ & 2x_1 + x_2 \geq 7.5\,, \\ & x_i \geq 0 \text{ integer}, \qquad i = 1, 2\,. \end{aligned}$$

Graphical solution of this scaled problem (presented in Example 3.4.1 without the integer design variable requirement before scaling) is

$$x_1 = x_2 = 2.5, \qquad f = 7.5\,,$$

and forms a lower bound for the objective function, $f_L = 7.5$. That is, the optimal integer solution cannot have an objective function smaller than $f_L = 7.5$. Next, we choose x_1 and investigate solutions for which $x_1 \leq 2$ and $x_1 \geq 3$ by forming two new LP's by adding each one of these constraints to the original set of constraints. Since the original set has a constraint that requires $x_1 \geq 2.5$, the first LP problem with $x_1 \leq 2$ has no solution. The solution of the second LP is shown graphically in Figure (3.9.3). The active constraints at the optimum are, $x_1 \geq 3$ and $x_1 + 2x_2 \geq 7.5$, and the solution is,

$$x_1 = 3, \qquad x_2 = 2.25, \qquad f = 8.25\,.$$

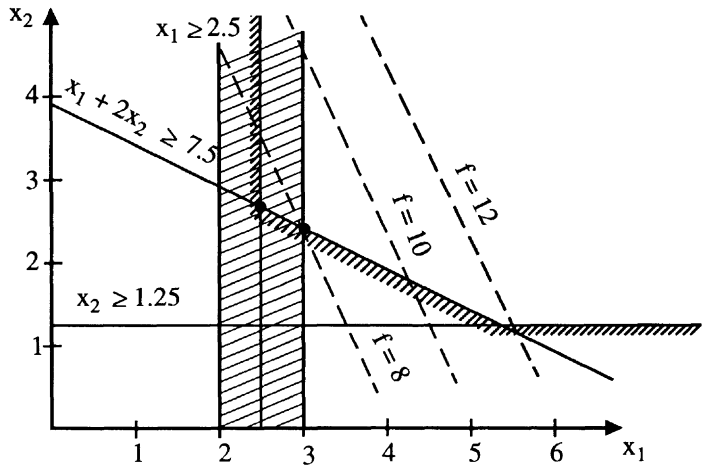


Figure 3.9.3 Branch-and-bound solution for $x_1 \leq 2$ and $x_1 \geq 3$ of Example 3.9.2 .

Since x_2 is still non integer, we create two more LP's, this time by imposing $x_2 \leq 2$ and $x_2 \geq 3$, respectively. Graphical solutions of the new LP's are shown in Figure (3.9.4). The solution for the case $x_2 \geq 3$ is at the vertex $x_1 = 3$ and $x_2 = 3$, and is a feasible solution for the integer problem with an objective function value of $f = 9$. This value of the objective function, therefore, establishes an upper bound, $f_U = 9$ for the problem. The solution for the case $x_2 \leq 2$, on the other hand is at the intersection of $x_2 = 2$ and $x_1 + 2x_2 = 5$ leading to

$$x_1 = 3.5, \qquad x_2 = 2, \quad \text{and} \quad f = 9 .$$

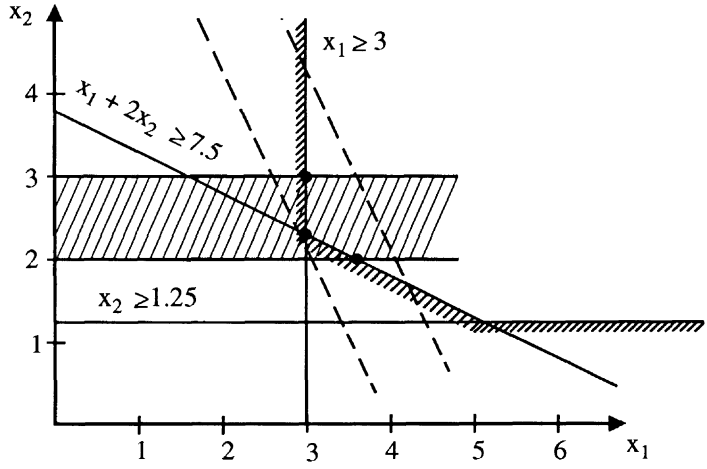


Figure 3.9.4 Branch-and-bound solution for $x_2 \leq 2$ and $x_2 \geq 3$ of Example 3.9.2 .

This solution is not discrete and can be interrogated further by branching on x_1 (that is creating new LP's by adding $x_1 \leq 3$ and $x_1 \geq 4$). However, since its objective function is equal to the upper bound, we cannot improve the objective function any further. To do so would necessitate introducing a further constraint which could only increase the objective function. Therefore, the optimal solution is the one with $x_1 = x_2 = 3$, and $f = 9$. • • •

As can be observed from the example, performance of the Branch-and-Bound algorithm relies heavily on the choice of noninteger variable to be used for branching, and the selection of node to be branched. If a selected node and branching variable leads to an upper bound close to the objective function of the LP-1 early in the enumeration scheme, then substantial computational savings can be obtained because of the elimination of branches that would not be capable of generating solutions lower than the upper bound. A rule of thumb for choosing the noninteger variable to be branched is to take the variable with the largest fraction. For the selection of the node to be branched, we choose, among all the live nodes, the LP problem which has the smallest value of the objective function; that node is most likely to generate a feasible design with a tighter upper bound.

Branch-and-Bound is only one of the algorithms for the solution of ILP or MILP problems. However, because of its simplicity it is incorporated into many commercially available computer programs [23, 24]. There are a number of other techniques which are capable of handling general discrete-valued problems (see, for example, Ref. [25]). Some of these algorithms are good not only for ILP problems but also for NLP problems with integer variables. Particularly, methods based on probabilistic search algorithms are emerging for many applications, including structural design applications, that involve linear and nonlinear programming problems. Two of such techniques, namely simulated annealing and genetic algorithms, are discussed in Chapter 4. Another approach, which is based on an extension of the penalty function approach for constrained NLP problems, is presented in Chapter 5. Finally, the use of dual variables (which are presented to be useful as prices of constraints in section 7.3) in ILP problems are discussed in Chapter 9.

One of the interesting design applications of the ILP was introduced by Haftka and Walsh [26] for the stacking sequence design of laminated composite plates for improved buckling response. Since the formulation of this problem involves material introduced in Chapter 11, discussion and demonstration of this application is presented in that chapter.

3.10 Exercises

1. Estimate the limit load for the three bar truss example 3.1.2 using a graphical approach. Verify your solution using the simplex method.

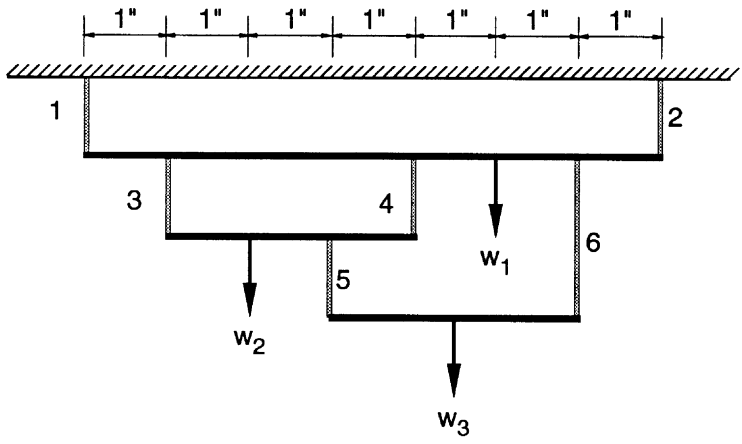


Figure 3.10.1 Platform support system

2. Consider the platform support system shown in Figure 3.10.1 in which cables 1 and 2 can support loads up to 400 lb each; cables 3 and 4 up to 150 lb each and cables 5 and 6 up to 75 lb each. Neglect the weight of the platforms and cables, and assume the weights w_1, w_2, and w_3 at the positions indicated in the figure. Also neglect the bending failure of the platforms. Using linear programming determine the the maximum total load that the system can support.

3. Solve the limit design problem for the truss of Figure 3.1.4 using the simplex algorithm. Assume $A_{13} = A_{24} = A_{34}$, $A_{14} = A_{23}$, and use appropriate nondimensionalization.

4. Using the method of virtual displacements verify that the collapse mechanisms for the portal frame of Figure 3.1.6 lead to Eqs. (3.1.26) through (3.1.31) in terms of the nondimensional variables x_1 and x_2.

5. The single bay, two story portal frame shown in Figure (3.10.2) is subjected to a single loading condition consisting of 4 concentrated loads as shown. Following Example 3.1.5 formulate the LP problem for the minimum weight design of the frame against plastic collapse.

6. Consider the continuous prestressed concrete beam shown in Figure (3.10.3),

a) Verify that the equivalent uniformly distributed upward force exerted on the concrete beam by a prestressing cable with a force f and a parabolic profile defined by eccentricities y_1, y_2, and y_3 at the three points $x = 0, x = l/2$, and $x = l$ respectively is given by

$$q = \frac{4f}{l^2}(y_3 - 2y_2 + y_1) \ .$$

b) The beam in the figure is subjected to two loading conditions: the first consisting of a dead load of 1 kip/ft together with an equivalent load due to a parabolic

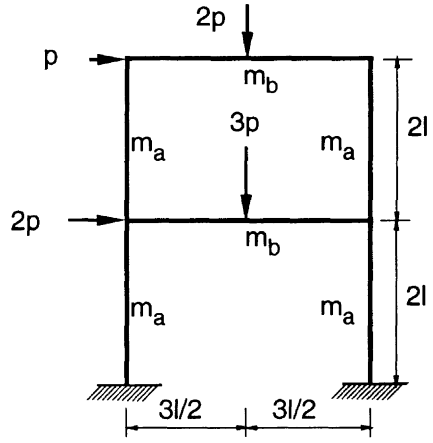


Figure 3.10.2 Two story portal frame

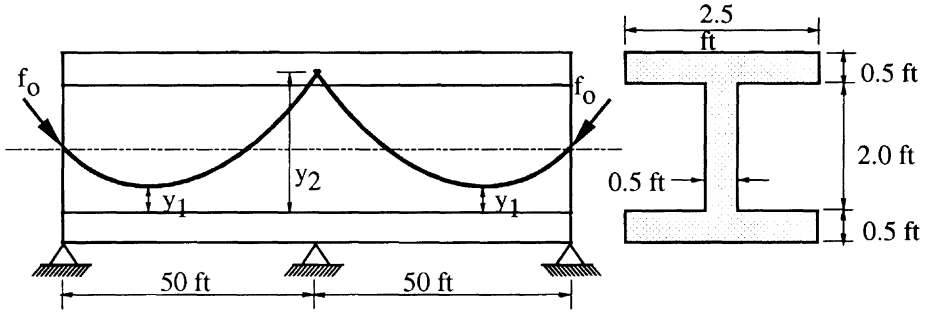


Figure 3.10.3 A continuous prestressed concrete beam

prestressing cable with a force f, and the second due to an additional live load of 2.5 kips/ft in service. It is assumed, however, that in service a 15% loss of prestressing force is to be expected. Formulate the LP problem for the minimum cost design of beam assuming f, y_1, and y_2 as design variables. Assume the allowable stress for the two loading conditions to be $\sigma_1^u = 200$ psi, $\sigma_1^l = -3000$ psi, $\sigma_2^u = 0$ psi, $\sigma_2^l = -2000$ psi and the upper and lower bound limits on the eccentricities y_1 and y_2 to be $0.4ft \leq y_i \leq 2.6ft$, $i = 1, 2$.

c) Solve the LP problem by the simplex algorithm and obtain the solution for the minimum prestressing force and the tendon profile.

7. Consider the statically determinate truss of Figure 3.3.1 and its minimum weight design formulation as described by Eqs. (3.3.9) through (3.3.13). Use the linearization scheme implied by Eqs. (3.3.2) through (3.3.5) to formulate the LP problem for m=3. Solve the LP by the simplex algorithm and compare the approximate solution with

the graphical or an exact solution to the problem.

8. Use Branch-and-Bound algorithm to solve the limit design problem of Exercise 3 by assuming the cross-sections of the members to take values from the following sets

a) $\{0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$.

b) $\{0.0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1\}$.

3.11 References

[1] Charnes, A. and Greenberg, H. J., "Plastic Collapse and Linear Programming," Bull. Am. Math. Soc., 57, 480, 1951.

[2] Calladine, C.R., Engineering Plasticity. Pergamon Press, 1969.

[3] Cohn, M.Z., Ghosh, S.K. and Parimi, S.R., "Unified Approach to Theory of Plastic Structures," Journal of the EM Division, 98 (EM5), pp. 1133–1158, 1972.

[4] Neal, B. G., The Plastic Methods of Structural Analysis, 3rd edition, Chapman and Hall Ltd., London, 1977.

[5] Zeman, P. and Irvine, H. M., Plastic Design, An Imposed Hinge–Rotation Approach, Allen and Unwin, Boston, 1986.

[6] Massonet, C.E. and Save, M.A., Plastic Analysis and Design, Beams and Frames, Vol. 1. Blaisdell Publishing Co., 1965.

[7] Lin, T.Y. and Burns, N.H., Design of Prestressed Concrete Structures, 3rd ed. John Wiley and Sons, New York, 1981.

[8] Parme, A.L. and Paris, G.H., "Designing for Continuity in Prestressed Concrete Structures," J. Am. Concr. Inst., 23 (1), pp. 45–64, 1951.

[9] Morris, D., "Prestressed Concrete Design by Linear Programming," J. Struct. Div., 104 (ST3), pp. 439–452, 1978.

[10] Kirsch, U., "Optimum Design of Prestressed Beams," Computers and Structures 2, pp. 573–583, 1972.

[11] Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass., 1973.

[12] Majid, K.I., Nonlinear Structures, London, Butterworths, 1972.

[13] Dantzig, G., Linear Programming and Extensions, Princeton University Press, Princeton, NJ, 1963.

[14] Karmarkar, N., "A New Polynomial–Time Algorithm for Linear Programming," Combinatorica, 4 (4), pp. 373–395, 1984.

[15] Todd, M. J. and Burrell, B. P., "An Extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables," Algorithmica, 1, pp. 409–424, 1986.

[16] Rinaldi, G., "A Projective Method for Linear Programming with Box–type Constraints," Algorithmica, 1, pp. 517–527, 1986.

[17] Strang, G., "Karmarkar's Algorithm and its Place in Applied Mathematics," The Mathematical Intelligencer, 9, 2, pp. 4–10, 1987.

[18] Vanderbei, R. F., Meketon, M. S., and Freedman, B. A., "A Modification of Karmarkar's Linear Programming Algorithm," Algorithmica, 1, pp. 395–407, 1986.

[19] Garfinkel, R. S., and Nemhauser, G. L., Integer Programming, John Wiley & Sons, Inc., New York, 1972.

[20] Lawler, E. L., and Wood, D. E., "Branch-and-Bound Methods—A Survey," Operations research, 14, pp. 699–719, 1966.

[21] Tomlin, J. A., "Branch-and-Bound Methods for Integer and Non-convex Programming," in Integer and Nonlinear Programming, J. Abadie (ed.), pp. 437–450, Elsevier Publishing Co., New York, 1970.

[22] Land, A. H., and Doig, A. G., "An Automatic Method for Solving Discrete Programming Problems," Econometrica, 28, pp. 497–520, 1960.

[23] Johnson, E. L., and Powell, S., "Integer Programming Codes," in Design and Implementation of Optimization Software, Greenberg, H. J. (ed.), pp. 225–240, 1978.

[24] Schrage, L., Linear, Integer, and Quadratic Programming with LINDO, 4th Edition, The Scientific Press, Redwood City CA., 1989.

[25] Kovács, L. B., Combinatorial Methods of Discrete Programming, Mathematical Methods of Operations Research Series, Vol. 2, Akadémiai Kiadó, Budapest, 1980.

[26] Haftka, R. T., and Walsh, J. L., "Stacking-sequence Optimization for Buckling of Laminated Plates by Integer Programming," AIAA J. (in press).

Unconstrained Optimization 4

In this chapter we study mathematical programming techniques that are commonly used to extremize nonlinear functions of single and multiple (n) design variables subject to no constraints. Although most structural optimization problems involve constraints that bound the design space, study of the methods of unconstrained optimization is important for several reasons. First of all, if the design is at a stage where no constraints are active then the process of determining a search direction and travel distance for minimizing the objective function involves an unconstrained function minimization algorithm. Of course in such a case one has constantly to watch for constraint violations during the move in design space. Secondly, a constrained optimization problem can be cast as an unconstrained minimization problem even if the constraints are active. The penalty function and multiplier methods discussed in Chapter 5 are examples of such indirect methods that transform the constrained minimization problem into an equivalent unconstrained problem. Finally, unconstrained minimization strategies are becoming increasingly popular as techniques suitable for linear and nonlinear structural analysis problems (see Kamat and Hayduk[1]) which involve solution of a system of linear or nonlinear equations. The solution of such systems may be posed as finding the minimum of the potential energy of the system or the minimum of the residuals of the equations in a least squared sense.

4.1 Minimization of Functions of One Variable

In most structural design problems the objective is to minimize a function with many design variables, but the study of minimization of functions of a single design variable is important for several reasons. First, some of the theoretical and numerical aspects of minimization of functions of n variables can be best illustrated, especially graphically, in a one dimensional space. Secondly, most methods for unconstrained minimization of functions $f(\mathbf{x})$ of n variables rely on sequential one-dimensional minimization of the function along a set of prescribed directions, $\mathbf{s}_k$, in the multi-dimensional design space $\mathbf{R}^n$. That is, for a given design point $\mathbf{x}_0$ and a specified search direction at that point $\mathbf{s}_0$, all points located along that direction can be expressed in terms of a single variable α by

$$\mathbf{x} = \mathbf{x}_0 + \alpha \mathbf{s}_0 \,, \tag{4.1.1}$$

where α is usually referred to as the step length. The function $f(\mathbf{x})$ to be minimized can, therefore, be expressed as

$$f(\mathbf{x}) = f(\mathbf{x}_0 + \alpha \mathbf{s}_0) = f(\alpha) \ . \tag{4.1.2}$$

Thus, the minimization problem reduces to finding the value α^* that minimizes the function, $f(\alpha)$. In fact, one of the simplest methods used in minimizing functions of n variables is to seek the minimum of the objective function by changing only one variable at a time, while keeping all other variables fixed, and performing a one-dimensional minimization along each of the coordinate directions of an n-dimensional design space. This procedure is called the *univariate search* technique.

In classifying the minimization algorithms for both the one-dimensional and multi-dimensional problems we generally use three distinct categories. These categories are the zeroth, first, and second order methods. Zeroth order methods use only the value of the function during the minimization process. First order methods employ values of the function and its first derivatives with respect to the variables. Finally, second order methods use the values of the function and its first and second derivatives. In the following discussion of one-variable function minimizations, the function is assumed to be in the form $f = f(\alpha)$. However, the methods to be discussed are equally applicable for minimization of multivariable problems along a preselected direction, $\mathbf{s}$, using Eq. (4.1.1).

4.1.1 *Zeroth Order Methods*

Bracketing Method. As the name suggests, this method brackets the minimum of the function to be minimized between two points, through a series of function evaluations. The method begins with an initial point α_0, a function value $f(\alpha_0)$, a step size β_0, and a step expansion parameter $\gamma > 1$. The steps of the algorithm [2] are outlined as

1. Evaluate $f(\alpha_0)$ and $f(\alpha_0 + \beta_0)$.

2. If $f(\alpha_0 + \beta_0) < f(\alpha_0)$, let $\alpha_1 = \alpha_0 + \beta_0$ and $\beta_1 = \gamma\beta_0$, and evaluate $f(\alpha_1 + \beta_1)$. Otherwise go to step 4.

3. If $f(\alpha_1 + \beta_1) < f(\alpha_1)$, let $\alpha_2 = \alpha_1 + \beta_1$ and $\beta_2 = \gamma\beta_1$, and continue incrementing the subscripts this way until $f(\alpha_k + \beta_k) > f(\alpha_k)$. Then, go to step 8.

4. Let $\alpha_1 = \alpha_0$ and $\beta_1 = -\xi\beta_0$, where ξ is a constant that satisfies $0 < \xi < 1/\gamma$, and evaluate $f(\alpha_1 + \beta_1)$.

5. If $f(\alpha_1 + \beta_1) > f(\alpha_1)$ go to step 7.

6. Let $\alpha_2 = \alpha_1 + \beta_1$ and $\beta_2 = \gamma\beta_1$, and continue incrementing the subscripts this way until $f(\alpha_k + \beta_k) > f(\alpha_k)$. Then, go to step 8.

7. The minimum has been bracketed between points $(\alpha_0 - \xi\beta_0)$ and $(\alpha_0 + \beta_0)$. Go to step 9.

8. The last three points satisfy the relations $f(\alpha_{k-2}) > f(\alpha_{k-1})$ and $f(\alpha_{k-1}) < f(\alpha_k)$, and hence, the minimum is bracketed.

9. Use either one of the two end points of the bracket as the initial point. Begin with a reduced step size and repeat steps 1 through 8 to locate the minimum to a desired degree of accuracy.

Quadratic Interpolation. The method known as quadratic interpolation was first proposed by Powell [3] and uses the values of the function f to be minimized at three points to fit a parabola

$$p(\alpha) = a + b\alpha + c\alpha^2 , \tag{4.1.3}$$

through those points. The method starts with an initial point, say, $\alpha = 0$ with a function value $p_0 = f(\mathbf{x}_0)$, and a step size β. Two more function evaluations are performed as described in the following steps to determine the points for the polynomial fit. In general, however, we start with a situation where we have already bracketed the minimum between $\alpha_1 = \alpha_l$ and $\alpha_2 = \alpha_u$ by using the bracketing method described earlier. In that case we will only need an intermediate point α_0 in the interval (α_l, α_u).

1. Evaluate $p_1 = p(\beta) = f(\mathbf{x}_0 + \beta\mathbf{s})$

2. If $p_1 < p_0$, then evaluate $p_2 = p(2\beta) = f(\mathbf{x}_0 + 2\beta\mathbf{s})$. Otherwise evaluate $p_2 = p(-\beta) = f(\mathbf{x}_0 - \beta\mathbf{s})$. The constants a, b, and c in equation Eq. (4.1.3) can now be uniquely expressed in terms of the function values p_0, p_1, and p_2 as

$$a = p_0 ,$$

$$b = \frac{4p_1 - 3p_0 - p_2}{2\beta}, \quad \text{and} \quad c = \frac{p_2 + p_0 - 2p_1}{2\beta^2}, \quad \text{if } p_2 = f(\mathbf{x}_0 + 2\beta\mathbf{s}) , \tag{4.1.4}$$

or

$$b = \frac{p_1 - p_2}{2\beta}, \quad \text{and} \quad c = \frac{p_1 - 2p_0 + p_2}{2\beta^2}, \quad \text{if } p_2 = f(\mathbf{x}_0 - \beta\mathbf{s}) . \tag{4.1.5}$$

3. The value of $\alpha = \alpha^*$ at which $p(\alpha)$ is extremized for the current cycle is then given by

$$\alpha^* = -\frac{b}{2c} . \tag{4.1.6}$$

4. α^* corresponds to a minimum of p if $c > 0$, and the prediction based on Eq. (4.1.3) is repeated using $(\mathbf{x}_0 + \alpha^*\mathbf{s})$ as the initial point for the next cycle with $p_0 = f(\mathbf{x}_0 + \alpha^*\mathbf{s})$ until the desired accuracy is obtained.

5. If the point $\alpha = \alpha^*$ corresponds to a maximum of p rather than a minimum, or if it corresponds to a minimum of p which is at a distance greater than a prescribed maximum $\beta_{\max}$ (possibly meaning α^* is outside the bracket points), then the maximum allowed step is taken in the direction of decreasing f and the point furthest away from this new point is discarded in order to repeat the process.

In step 4, instead of starting with $(\mathbf{x}_0 + \alpha^*\mathbf{s})$ as the initial point and repeating the previous steps, there is a cheaper alternative in terms of the number of function evaluations. The point $(\mathbf{x}_0 + \alpha^*\mathbf{s})$ and the two points closest to it from the left and

right can be used in another quadratic interpolation to give a better value of α^*. Other strategies for improving the accuracy of the prediction will be discussed later in Section 4.1.4.

Fibonacci and the Golden Section Search. Like bracketing, the Fibonacci and the golden section search techniques are very reliable, if not the most efficient, line search techniques for locating the unconstrained minimum of a function $f(\alpha)$ within the interval $a_0 \leq \alpha \leq b_0$. It is assumed that the function f is *unimodal*, or that it has only one minimum within the interval. Unimodal functions are not necessarily continuous or differentiable, nor convex (see Figure 4.1.1). A function is said to be unimodal [3] in the interval $\mathcal{I}_0$ if there exist an $\alpha^* \in \mathcal{I}_0$ such that α^* minimizes f on $\mathcal{I}_0$, and for any two points $\alpha_1, \alpha_2 \in \mathcal{I}_0$ such that $\alpha_1 < \alpha_2$ we have

$$\alpha_2 \leq \alpha^* \quad \text{implies that} \quad f(\alpha_1) > f(\alpha_2)\,, \tag{4.1.7}$$

$$\alpha_1 \geq \alpha^* \quad \text{implies that} \quad f(\alpha_2) > f(\alpha_1)\,. \tag{4.1.8}$$

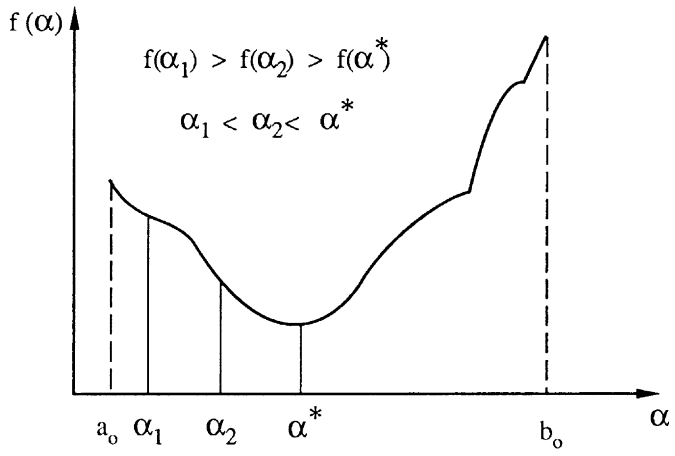


Figure 4.1.1 A typical unimodal function.

The assumption of unimodality is central to the Fibonacci search technique which seeks to reduce the interval of uncertainty within which the minimum of the function f lies.

The underlying idea behind the Fibonacci and the golden section search techniques can be explained as follows. Consider the minimization of f in the interval (a_0, b_0). Let us choose two points in the interval (a_0, b_0) at $\alpha = \alpha_1$ and at $\alpha = \alpha_2$ such that $\alpha_1 < \alpha_2$, and evaluate the function f at these two points. If $f(\alpha_1) > f(\alpha_2)$, then since the function is unimodal the minimum cannot lie in the interval (a_0, α_1). The new interval is (α_1, b_0) which is smaller than the original interval. Similarly, if $f(\alpha_2) > f(\alpha_1)$, then the new interval will be (a_0, α_2). The process can be repeated to

reduce the interval to any desired level of accuracy. Only one function evaluation is required in each iteration after the first one, but we have not specified how to choose the locations where f is evaluated. The best placement of these points will minimize the number of function evaluations for a prescribed accuracy requirement (i.e., reduction of the interval of uncertainty to a prescribed size). If the number of function evaluations is n the most efficient process is provided by a symmetric placement of the points provided by the relations [4]

$$\alpha_1 = a_0 + \frac{f_{n-1}}{f_{n+1}} l_0 \,, \tag{4.1.9}$$

$$\alpha_2 = b_0 - \frac{f_{n-1}}{f_{n+1}} l_0 \,, \tag{4.1.10}$$

and

$$\alpha_{k+1} \quad = \quad a_k + \frac{f_{n-(k+1)}}{f_{n-(k-1)}} l_k \quad = \quad b_k - \frac{f_{n-(k+1)}}{f_{n-(k-1)}} l_k \,, \tag{4.1.11}$$

where f_n are Fibonacci numbers defined by the sequence $f_0 = 1$, $f_1 = 1$, $f_n = f_{n-2} + f_{n-1}$, and l_k is the length of the kth interval (a_k, b_k). The total number of required function evaluations n may be determined from the desired level of accuracy. It can be shown that the interval of uncertainty after n function evaluations is $2\epsilon l_0$ where

$$\epsilon = \frac{1}{f_{n+1}} \,. \tag{4.1.12}$$

A disadvantage of the technique is that the number of function evaluations has to be specified in advance in order to start the Fibonacci search. To eliminate this undesirable feature a quasi-optimal technique known as the golden section search technique has been developed. The golden section search technique is based on the finding that for sufficiently large n, the ratio

$$\frac{f_{n-1}}{f_{n+1}} \to 0.382 \,. \tag{4.1.13}$$

Thus, it is possible to approximate the optimal location of the points given by Eqs. (4.1.9 - 4.1.11) by the following relations

$$\alpha_1 = a_0 + 0.382 l_0 \,, \tag{4.1.14}$$

$$\alpha_2 = b_0 - 0.382 l_0 \,, \tag{4.1.15}$$

and

$$\alpha_{k+1} \quad = \quad a_k + 0.382 l_k \quad = \quad b_k - 0.382 l_k \,. \tag{4.1.16}$$

Example 4.1.1

Determine the value of α, to within $\epsilon = \pm 0.1$, that minimizes the function $f(\alpha) = \alpha(\alpha - 3)$ on the interval $0 \leq \alpha \leq 2$ using the golden section search technique.

From Eqs. (4.1.14) and (4.1.15) we can calculate

$$\alpha_1 = 0 + 0.382(2) = 0.764, \qquad f(\alpha_1) = -1.708,$$
$$\alpha_2 = 2 - 0.382(2) = 1.236, \qquad f(\alpha_2) = -2.180 .$$

Since $f(\alpha_2) < f(\alpha_1)$ we retain $(\alpha_1, 2)$. Thus, the next point is located at

$$\alpha_3 = 2 - 0.382(2 - 0.764) = 1.5278, \qquad f(\alpha_3) = -2.249 .$$

Since $f(\alpha_3) < f(\alpha_2)$ we reject the interval (α_1, α_2). The new interval is $(\alpha_2, 2)$. The next point is located at

$$\alpha_4 = 2 - 0.382(2 - 1.236) = 1.7082, \qquad f(\alpha_4) = -2.207 .$$

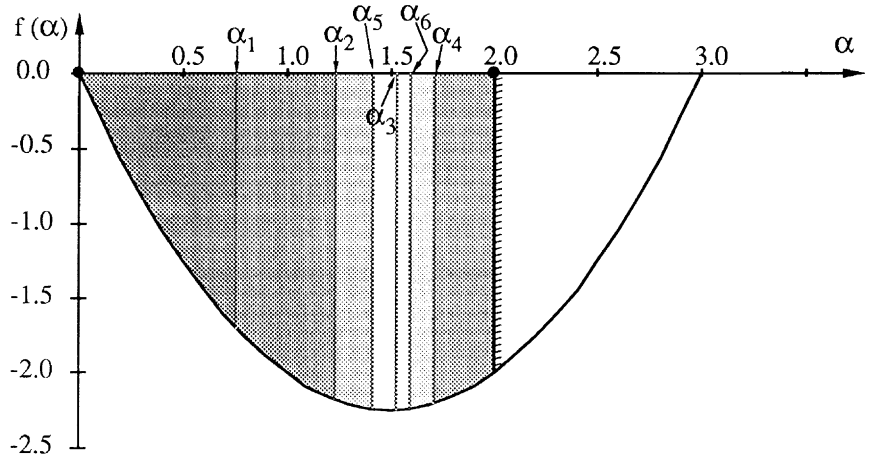


Figure 4.1.2 Iteration history for the function minimization $f(\alpha) = \alpha(\alpha - 3)$.

Since $f(\alpha_4) < f(\alpha_2) < f(2)$ we reject the interval $(\alpha_4, 2)$ and retain (α_2, α_4) as the next interval and locate the point α_5 at

$$\alpha_5 = 1.236 + 0.382(1.7082 - 1.236) = 1.4164, \qquad f(\alpha_5) = -2.243 .$$

Since $f(\alpha_5) < f(\alpha_4) < f(\alpha_2)$ we retain the interval (α_5, α_4). The next point is located at

$$\alpha_6 = 1.7082 + 0.382(1.7082 - 1.4164) = 1.5967, \qquad f(\alpha_6) = -2.241 .$$

Since $f(\alpha_6) < f(\alpha_4)$ we reject the interval (α_6, α_4) and retain the interval (α_5, α_6) of length 0.18, which is less than the interval of specified accuracy, $2\epsilon = 0.2$. The iteration history for the problem is shown in Figure 4.1.2. Hence, the minimum has been bracketed to within a resolution of ± 0.1. That is, the minimum lies between $\alpha_5 = 1.4164$ and $\alpha_6 = 1.5967$. We can take the middle of the interval, $\alpha = 1.5066 \pm 0.0902$ as the solution. The exact location of the minimum is at $\alpha = 1.5$ where the function has the value -2.25. • • •

4.1.2 *First Order Methods*

Bisection Method. Like the bracketing and the golden section search techniques which progressively reduce the interval where the minimum is known to lie, the bisection technique locates the zero of the function f' by reducing the interval of uncertainty. Beginning with the known interval (a, b) for which $f'(a)f'(b) < 0$, an approximation to the root of f' is obtained from

$$\alpha^* = \frac{a+b}{2}\,, \tag{4.1.17}$$

which is the point midway between a and b. The value of f' is then evaluated at α^*. If $f'(\alpha^*)$ agrees in sign with $f'(a)$ then the point a is replaced by α^* and the new interval of uncertainty is given by (α^*, b). If on the other hand $f'(\alpha^*)$ agrees in sign with $f'(b)$ then the point b is replaced by α^* and the new interval of uncertainty is (a, α^*). The process is then repeated using Eq. (4.1.17).

Davidon's Cubic Interpolation Method. This is a polynomial approximation method which uses both the function values and its derivatives for locating its minimum. It is especially useful in those multivariable minimization techniques which require the evaluation of the function and its gradients.

We begin by assuming the function to be minimized $f(\mathbf{x}_0 + \alpha \mathbf{s}_0)$ to be approximated by a polynomial in the form

$$p(\alpha) = a + b\alpha + c\alpha^2 + d\alpha^3\,, \tag{4.1.18}$$

with constants $a, b, c,$ and d to be determined from the values of the function, p_0 and p_1, and its derivatives, g_0 and g_1, at two points, one located at $\alpha = 0$ and the other at $\alpha = \beta$.

$$p_0 = p(0) = f(\mathbf{x}_0), \qquad p_1 = p(\beta) = f(\mathbf{x}_0 + \beta \mathbf{s})\,, \tag{4.1.19}$$

and

$$g_0 = \frac{\mathrm{d}p}{\mathrm{d}\alpha}(0) = \mathbf{s}^T \nabla f(\mathbf{x}_0), \qquad g_1 = \frac{\mathrm{d}p}{\mathrm{d}\alpha}(\beta) = \mathbf{s}^T \nabla f(\mathbf{x}_0 + \beta \mathbf{s})\,. \tag{4.1.20}$$

After substitutions, Eq. (4.1.18) takes the following form

$$p(\alpha) = p_0 + g_0\alpha - \frac{g_0 + e}{\beta}\alpha^2 + \frac{g_0 + g_1 + 2e}{3\beta^2}\alpha^3\,, \tag{4.1.21}$$

where

$$e = \frac{3}{\beta}(p_0 - p_1) + g_0 + g_1\,. \tag{4.1.22}$$

We can now locate the minimum, $\alpha = \alpha_m$, of Eq. (4.1.21) by setting its derivative with respect to α to be zero. This results in

$$\alpha_m = \beta\left(\frac{g_0 + e \pm h}{g_0 + g_1 + 2e}\right)\,, \tag{4.1.23}$$

where

$$h = (e^2 - 2g_0g_1)^{1/2} \,. \tag{4.1.24}$$

It can be easily verified, by checking $\mathrm{d}^2p/\mathrm{d}\alpha^2$, that the positive sign must be retained in Eq. (4.1.23) for α_m to be a minimum rather than a maximum. Thus, the algorithm for Davidon's cubic interpolation [5] may be summarized as follows.

1. Evaluate $p_0 = f(\mathbf{x}_0)$ and $g_0 = \mathbf{s}^T\nabla f(\mathbf{x}_0)$ and make sure that $g_0 < 0$.

2. In the absence of an estimate of the initial step length β, we may calculate it on the basis of a quadratic interpolation derived using p_0, g_0 and an estimate of $p_{\min}$. Thus,

$$\beta = \frac{2(p_{\min} - p_0)}{g_0} \,. \tag{4.1.25}$$

3. Evaluate $p_1 = f(\mathbf{x}_0 + \beta\mathbf{s})$ and $g_1 = \dfrac{df(\mathbf{x}_0 + \beta\mathbf{s})}{d\beta}$

4. If $g_1 > 0$ or if $p_1 > p_0$ go to step 6, or else go to step 5.

5. Replace β by 2β and go to step 3.

6. Calculate α_m using Eq. (4.1.23) with a positive sign.

7. Use the interval $(0, \alpha_m)$ if

$$g_{\alpha_m} = \frac{df(\mathbf{x}_0 + \alpha_m\mathbf{s})}{d\alpha_m} \geq 0 \,, \tag{4.1.26}$$

or else use the interval (α_m, β) and return to step 4.

8. If α_m corresponds to a maximum, restart the algorithm by using new points. Selection of the new points may be performed by using a strategy similar to that described for the quadratic interpolation technique.

4.1.3 *Second Order Method*

The problem of minimizing the function $f(\alpha)$ is equivalent to obtaining the root of the nonlinear equation

$$f'(\alpha) = 0 \,, \tag{4.1.27}$$

because this is the necessary condition for the extremum of f. A convenient method for solving (4.1.27) is Newton's method. This method consists of linearizing $f'(\alpha)$ about a point $\alpha = \alpha_i$ and then determining the point α_{i+1} at which the linear approximation

$$f'(\alpha_{i+1}) = f'(\alpha_i) + f''(\alpha_i)(\alpha_{i+1} - \alpha_i) \,, \tag{4.1.28}$$

vanishes. This point

$$\alpha_{i+1} = \alpha_i - \frac{f'(\alpha_i)}{f''(\alpha_i)} \,, \tag{4.1.29}$$

serves as a new approximation for a repeated application of Eq. (4.1.29) with i replaced by $i+1$. For a successful convergence to the minimum it is necessary that the second derivative of the function f be greater than zero. Even so the method may diverge depending on the starting point. Several strategies exist [6] which modify Newton's method to make it globally convergent (that is, it will converge to a minimum regardless of the starting point) for multivariable functions; some of these will be covered in the next section.

The reason this method is known as a second order method is not only because it uses second derivative information about the function f, but also because it has a rate of convergence to the minimum that is quadratic. In other words, Newton's algorithm converges to the minimum α^* such that

$$\lim_{i\to\infty} \frac{|\alpha_{i+1} - \alpha^*|}{(\alpha_i - \alpha^*)^2} = \beta\,, \tag{4.1.30}$$

where α_i and α_{i+1} are the ith and the $(i+1)$st estimates of the minimum value of the α^*, β is a non-zero constant.

4.1.4 *Safeguarded Polynomial Interpolation* [7], p. 92

Polynomial interpolations such as the *Quadratic interpolation* and the *Davidon's cubic interpolation* are sometimes found to be quite inefficient and unreliable for locating the minimum of a function along a line. If the interpolation function is not representative of the behavior of the function to be minimized within the interval of uncertainty, the minimum may fall outside the interval, or become unbounded below, or the successive iterations may be too close to one another without achieving a significant improvement in the function value. In such cases, we use what are known as safeguarded procedures. These procedures consist of combining polynomial interpolations with a simple bisection technique or the golden section search technique described earlier. At the end of the polynomial interpolation, the bisection technique would be used to find the zero of the derivative of the function f. The golden section search, on the other hand, would work with the function f itself using the known interval of uncertainty (a, b) and locate the point α^* which corresponds to the minimum of f within the interval.

4.2 Minimization of Functions of Several Variables

4.2.1 *Zeroth Order Methods*

Several methods exist for minimizing a function of several variables using only function values. However, only two of these methods may be regarded as being useful. These are the *sequential simplex* method of Spendley, Hext and Himsworth [8] and

Powell's conjugate direction method [3]. Both of these methods require that the function $f(\mathbf{x}), \mathbf{x} \in \mathbf{R}^n$, be unimodal; that is the function f has only one minimum. The sequential simplex does not require that the function f be differentiable, while the differentiability requirement on f is implicit in the exact line searches of Powell's method. It appears from tests by Nelder and Mead [9] that for most problems the performance of the sequential simplex method is comparable to if not better than Powell's method. Both of these methods are considered inefficient for $n \geq 10$; Powell's method may fail to converge for $n \geq 30$. A more recent modification of the simplex method by Chen, et al. [10] extends the applicability of this algorithm for high dimensional cases. If the function is differentiable, it is usually more efficient to use the more powerful first and second order methods with derivatives obtained explicitly or from finite difference formulae.

Sequential Simplex Method. The sequential simplex method was originally proposed by Spendley, Hext and Himsworth [8] and was subsequently improved by Nelder and Mead [9]. The method begins with a regular geometric figure called the simplex consisting of $n+1$ vertices in an n-dimensional space. These vertices may be defined by the origin and by points along each of the n coordinate directions. Such a simplex may not be geometrically regular. The following equations are suggested in Ref. 8 for the calculation of the positions of the vertices of a regular simplex of size a in the n-dimensional design space

$$\mathbf{x}_j = \mathbf{x}_0 + p\mathbf{e}_j + \sum_{\substack{k=1 \\ k \neq j}}^{n} q\mathbf{e}_k, \qquad j = 1, \ldots, n, \tag{4.2.1}$$

with

$$p = \frac{a}{n\sqrt{2}}(\sqrt{n+1} + n - 1), \qquad \text{and} \qquad q = \frac{a}{n\sqrt{2}}(\sqrt{n+1} - 1), \tag{4.2.2}$$

where $\mathbf{e}_k$ is the unit base vector along the kth coordinate direction, and $\mathbf{x}_0$ is the initial base point. For example, for a problem in two-dimensional design space Eqs. (4.2.1) and (4.2.2) lead to an equilateral triangle of side a.

Once the simplex is defined, the function f is evaluated at each of the $n+1$ vertices $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$. Let $\mathbf{x}_h$ and $\mathbf{x}_l$ denote the vertices where the function f assumes its maximum and minimum values, respectively, and $\mathbf{x}_s$ the vertex where it assumes the second highest value. The simplex method discards the vertex $\mathbf{x}_h$ and replaces it by a point where f has a lower value. This is achieved by three operations namely *reflection, contraction*, and *expansion.*

The reflection operation creates a new point $\mathbf{x}_r$ along the line joining $\mathbf{x}_h$ to the centroid $\bar{\mathbf{x}}$ of the remaining points defined as

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{\substack{i=0 \\ i \neq h}}^{n} \mathbf{x}_i, \qquad i \neq h\ . \tag{4.2.3}$$

The vertex at the end of the reflection is calculated by

$$\mathbf{x}_r = \bar{\mathbf{x}} + \alpha(\bar{\mathbf{x}} - \mathbf{x}_h), \tag{4.2.4}$$

with α being a positive constant called the reflection coefficient which is usually assumed to be unity. Any positive value of the reflection coefficient in Eq. (4.2.4) guarantees that $\mathbf{x}_r$ is on the other side of the $\bar{\mathbf{x}}$ from $\mathbf{x}_h$. If the value of the function at this new point, $f_r = f(\mathbf{x}_r)$, satisfies the condition $f_l < f_r \leq f_s$, then $\mathbf{x}_h$ is replaced by $\mathbf{x}_r$ and the process is repeated with this new simplex. If, on the other hand, the value of the function f_r at the end of the reflection is less than the lowest value of the function $f_l = f(\mathbf{x}_l)$, then there is a possibility that we can still decrease the function by going further along the same direction. We seek an improved point $\mathbf{x}_e$ by the expansion technique using the relation

$$\mathbf{x}_e = \bar{\mathbf{x}} + \beta(\mathbf{x}_r - \bar{\mathbf{x}}), \tag{4.2.5}$$

with the expansion coefficient β often being chosen to be 2. If the value of the function f_e is smaller than the value at the end of the reflection step, then we replace $\mathbf{x}_h$ by $\mathbf{x}_e$ and repeat the process with the new simplex. However, if the expansion leads to a function value equal to or larger than f_r, then we form the new simplex by replacing $\mathbf{x}_h$ by $\mathbf{x}_r$ and continue.

Finally, if the process of reflection leads to a point x_r such that, $f_r < f_h$, then we replace $\mathbf{x}_h$ by $\mathbf{x}_r$ and perform contraction. Otherwise ($f_r \geq f_h$), we perform contraction without any replacement using

$$\mathbf{x}_c = \bar{\mathbf{x}} + \gamma(\mathbf{x}_h - \bar{\mathbf{x}}), \tag{4.2.6}$$

with the contraction coefficient γ, $0 < \gamma < 1$, usually chosen to be $1/2$. If $f_c = f(\mathbf{x}_c)$ is greater than f_h, then we replace all the points by a new set of points

$$\mathbf{x}_i = \mathbf{x}_i + \frac{1}{2}(\mathbf{x}_l - \mathbf{x}_i), \qquad i = 0, 1, \ldots, n, \tag{4.2.7}$$

and restart the process with this new simplex. Otherwise, we simply replace $\mathbf{x}_h$ by $\mathbf{x}_c$ and restart the process with this simplex. The operation in Eq. (4.2.7) causes the distance between the points of the old simplex and the point with the lowest function value to be halved and is therefore referred to as the *shrinkage* operation. The flow chart of the complete method is given in Figure 4.2.1. For the convergence criterion to terminate the algorithm Nelder and Mead [9] proposed the following

$$\left\{ \frac{1}{1+n} \sum_{i=0}^{n} [f_i - f(\bar{\mathbf{x}})]^2 \right\}^{\frac{1}{2}} < \epsilon, \tag{4.2.8}$$

where ϵ is some specified accuracy requirement.

An improvement in the performance of the simplex algorithm for those cases with large number of design variables, n, is achieved by Chen, Saleem, and Grace [10]. A modified simplex search procedure proposed in Ref. [10] executes the *reflection, expansion, contraction*, and, *shrinkage* operations on more than one vertex of the simplex at a given step. This is achieved by first separating the vertices of the simplex

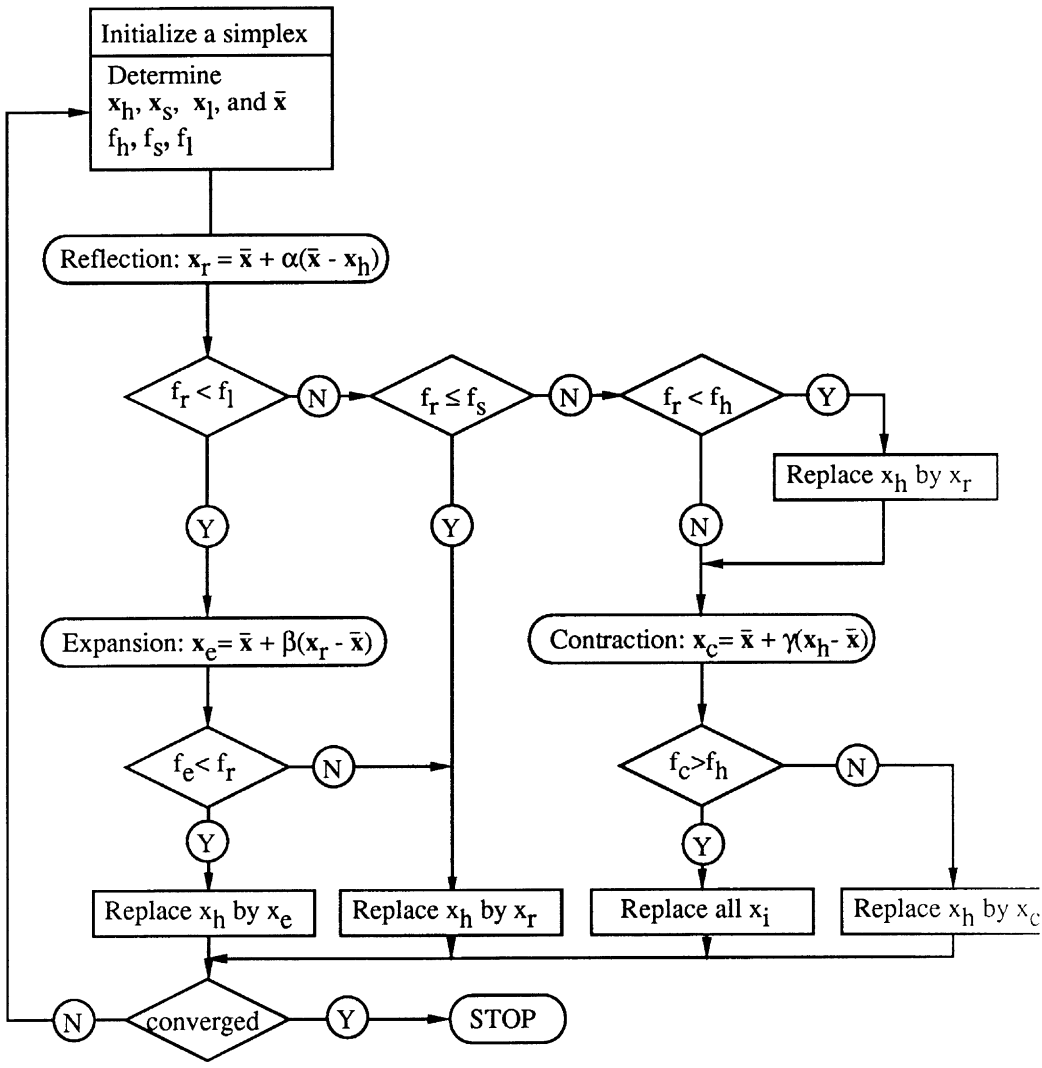


Figure 4.2.1 Flow chart of the Sequential Simplex Algorithm.

into two groups by defining a cutting value (CV) of the function, f_{cv}. The cutting value is defined by the relation

$$f_{cv} = \frac{(f_h + f_l)}{2} + \eta s, \tag{4.2.9}$$

where s is the standard deviation of the values of the function corresponding to the

vertices of the simplex,

$$s = \left[\sum_{i=0}^{n}(f_i - \bar{f})^2/(n+1)\right]^{\frac{1}{2}}, \tag{4.2.10}$$

and η is a parameter (discussed below) that controls the number of vertices to be operated on. The $\bar{f}$ value in Eq. (4.2.10) is the average of the function values over the entire current simplex.

The vertices with function values higher than the cutting value form the group to be reflected (and to be dropped). The other vertices serve as reference points. If the parameter η is sufficiently large, all the vertices of the simplex except the $\mathbf{x}_h$ stay in the group to be used as the reference points and, therefore, the algorithm is equivalent to the original form. For sufficiently small values of the parameter η, all points except the $\mathbf{x}_n$ are dropped. The selection of the parameter η depends on the difficulty of the problem as well as the number of variables. Recommended values for η are given in Table II of Ref. [10]. Among the $n+1$ vertices of the current simplex, we rearrange and number the vertices from largest to smallest function values as $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\text{cv}}, \ldots, \mathbf{x}_n$ where $i = 0, \ldots, n_{\text{cv}}$ are the elements of the group to be reflected next. The centroid of the vertices in the reference group is defined as

$$\bar{\mathbf{x}} = \frac{1}{n - n_{\text{cv}}} \sum_{i=n_{\text{cv}}+1}^{n} \mathbf{x}_i \, . \tag{4.2.11}$$

The performance of this modified simplex method has been compared [10] with the simplex method proposed by Nelder and Mead, and also with more powerful methods such as the second order Davidon-Fletcher-Powell (DFP) method which will be discussed later in this chapter. For high dimensional problems the modified simplex algorithm was found to be more efficient and robust than the DFP algorithm. Nelder and Mead [9] have also provided several illustrations of the use of their algorithm in minimizing classical test functions and compared its performance with Powell's conjugate directions method which will be discussed next.

Powell's Conjugate Directions Method and its Subsequent Modification. Although most problems have functions which are not quadratic, many unconstrained minimization algorithms are developed to minimize a quadratic function. This is because a function can be approximated well by a quadratic function near a minimum. Powell's conjugate directions algorithm is a typical example. A quadratic function in $\mathbf{R}^n$ may be written as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c} \, . \tag{4.2.12}$$

A set of directions $\mathbf{s}_i, i = 1, 2 \ldots$ are said to be Q-conjugate if

$$\mathbf{s}_i^T Q \mathbf{s}_j = 0, \qquad \text{for } i \neq j \, . \tag{4.2.13}$$

Furthermore, it can be shown that if the function f is minimized once along each direction of a set $\mathbf{s}$ of linearly independent Q-conjugate directions then the minimum

of f will be located at or before the nth step regardless of the starting point provided that no round-off errors are accumulated. This property is commonly referred to as the quadratic termination property. Powell provided a convenient method for generating such conjugate directions by a suitable combination of the simple univariate search and a pattern search technique [3]. However, in certain cases Powell's algorithm generates directions which are linearly dependent and thereby fails to converge to the minimum. Hence, Powell modified his algorithm to make it robust but at the expense of its quadratic termination property.

Powell's strategy for generating conjugate directions is based on the following property (see Ref. 3 for proof). If $\mathbf{x}_1$ and $\mathbf{x}_2$ are any two points and $\mathbf{s}$ a specified direction, and $\mathbf{x}_{1s}$ corresponds to the minimum point of a quadratic function f on a line starting at $\mathbf{x}_1$ along $\mathbf{s}$ and $\mathbf{x}_{2s}$ is the minimum point on a line starting at $\mathbf{x}_2$ along $\mathbf{s}$, then the directions $\mathbf{s}$ and $(\mathbf{x}_{2s} - \mathbf{x}_{1s})$ are Q-conjugate. The basic steps of Powell's modified method are based on a cycle of univariate minimizations. For each cycle we use the following steps.

1. Minimize f along each of the coordinate directions (univariate search) starting at $\mathbf{x}_0^k$ and generating the points $\mathbf{x}_1^k, \ldots, \mathbf{x}_n^k$ where k is the cycle number.

2. After completing the univariate cycle find the index m corresponding to the direction of the univariate search which yields the largest function decrease in going from $\mathbf{x}_{m-1}^k$ to $\mathbf{x}_m^k$.

3. Calculate the "pattern" direction $\mathbf{s}_p^k = \mathbf{x}_n^k - \mathbf{x}_0^k$ (which is the sum of all the univariate moves) and determine the value of α from $\mathbf{x}_0^k$ along $\mathbf{s}_p^k$ that minimizes f. Denote this new point by $\mathbf{x}_0^{k+1}$.

4. If

$$|\alpha| < \left[\frac{f(\mathbf{x}_0^k) - f(\mathbf{x}_0^{k+1})}{f(\mathbf{x}_{m-1}^k) - f(\mathbf{x}_m^k)}\right]^{\frac{1}{2}}, \tag{4.2.14}$$

then use the same old directions again for the next univariate cycle (that is do not discard any of the directions of the previous cycle in preference to the pattern direction $\mathbf{s}_p^k$). If Eq. (4.2.14) is not satisfied then replace the mth direction by the pattern direction $\mathbf{s}_p^k$.

5. Begin the next univariate cycle with the directions decided in step 4, and repeat the steps 2 through 4 until convergence to a specified accuracy. Convergence is assumed to be achieved when the Euclidean norm $\|\mathbf{x}^{k-1} - \mathbf{x}^k\|$ is less than a prespecified quantity ϵ.

Although Powell's original method does possess a quadratic termination property, his modified algorithm does not [3]. The modified method will now be illustrated on the following simple example from structural analysis.

Example 4.2.1

The problem of determination of the maximum deflection and tip-rotation of a cantilever beam of length l shown in Figure (4.2.2) loaded at its tip is considered. Solution of this problem is formulated as a minimization of the total potential energy of the beam which is modelled using a single cubic beam finite element. For a two-noded beam element with two degrees of freedom at each node, the displacement field is assumed to be

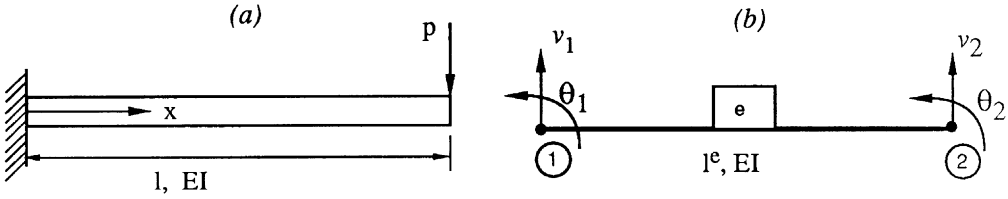


Figure 4.2.2 Tip loaded cantilever beam and its finite element model.

$$v(\xi) = \left[(1 - 3\xi^2 + 2\xi^3) \quad l(\xi - 2\xi^2 + \xi^3) \quad (3\xi^2 - 2\xi^3) \quad l(-\xi^2 + \xi^3)\right] \begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix}, \tag{4.2.15}$$

where $\xi = x/l$. The corresponding potential energy of the beam model is given by

$$\Pi = \frac{EI}{2l^3} \int_0^l \left(\frac{\mathrm{d}^2 v}{\mathrm{d}\xi^2}\right)^2 \mathrm{d}\xi + pv_2 \ . \tag{4.2.16}$$

Because of the cantilever end condition at $\xi = 0$, the first two degrees of freedom in Eq. (4.2.15) are zero. Therefore, substituting Eq. (4.2.15) into Eq. (4.2.16) we obtain

$$\Pi = \frac{EI}{2l^3}(12v_2^2 + 4\theta_2^2 l^2 - 12v_2\theta_2 l) + pv_2 \ . \tag{4.2.17}$$

Defining $f = 2\Pi l^3/EI$, $x_1 = v_2$, $x_2 = \theta_2 l$, and choosing $pl^3/EI = 1$, the problem of determining the tip deflection and rotation of the beam reduces to an unconstrained minimization of

$$f = 12x_1^2 + 4x_2^2 - 12x_1x_2 + 2x_1 \ . \tag{4.2.18}$$

Starting with an initial point of $\mathbf{x}_0^1 = (-1, \ -2)^T$ and $f(\mathbf{x}_0^1) = 2$ we will minimize f using Powell's conjugate directions method. The exact solution of this problem is at $\mathbf{x}^* = (-1/3, \ -1/2)^T$.

Since we have an explicit relation for the objective function f, the one dimensional minimizations along a given direction will be performed exactly without resorting to any of the numerical techniques discussed in the previous section. However, if these minimizations were done numerically, one of the zeroth order techniques would be sufficient. We use superscripts to denote the univariate cycle number and subscripts to denote the iteration number within a cycle.

First, we perform the univariate search along the x_1 and x_2 directions. Choosing $\mathbf{s}_1^1 = (1,\ 0)^T$ we have

$$\mathbf{x}_1^1 = \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} + \alpha \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -1+\alpha \\ -2 \end{Bmatrix}, \tag{4.2.19}$$

and

$$f(\alpha) = 12(-1+\alpha)^2 + 4(-2)^2 - 12(-1+\alpha)(-2) + 2(-1+\alpha)\ . \tag{4.2.20}$$

Taking the derivative of Eq. (4.2.20) with respect to α, we obtain the value of α which minimizes f to be $\alpha = -1/12$. Hence,

$$\mathbf{x}_1^1 = \begin{Bmatrix} \frac{-13}{12} \\ -2 \end{Bmatrix} \qquad \text{and} \quad f(\mathbf{x}_1^1) = 1.916666667\ .$$

Choosing $\mathbf{s}_2^1 = (0,\ 1)^T$, we obtain

$$\mathbf{x}_2^1 = \begin{Bmatrix} \frac{-13}{12} \\ -2 \end{Bmatrix} + \alpha \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} \frac{-13}{12} \\ -2+\alpha \end{Bmatrix}, \tag{4.2.21}$$

and

$$f(\alpha) = 12\left(\frac{-13}{12}\right)^2 + 4(-2+\alpha)^2 - 12\left(\frac{-13}{12}\right)(-2+\alpha) + 2\left(\frac{-13}{12}\right), \tag{4.2.22}$$

which is minimum at $\alpha = 3/8$. Therefore, at the end of the univariate search we have

$$\mathbf{x}_2^1 = \begin{Bmatrix} \frac{-13}{12} \\ \frac{-13}{8} \end{Bmatrix} \qquad \text{and} \quad f(\mathbf{x}_2^1) = 1.354166667\ .$$

At this point we construct a pattern direction as

$$\mathbf{s}_p^1 = \mathbf{x}_2^1 - \mathbf{x}_0^1 = \begin{Bmatrix} \frac{-13}{12} \\ \frac{-13}{8} \end{Bmatrix} - \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} = \begin{Bmatrix} \frac{-1}{12} \\ \frac{3}{8} \end{Bmatrix}, \tag{4.2.23}$$

and minimize the function along this direction by

$$\mathbf{x}_0^2 = \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} + \alpha \begin{Bmatrix} \frac{-1}{12} \\ \frac{3}{8} \end{Bmatrix} = \begin{Bmatrix} -1 - \frac{\alpha}{12} \\ -2 + \frac{3\alpha}{8} \end{Bmatrix}, \tag{4.2.24}$$

which attains its minimum value for $\alpha = 40/49$ at

$$\mathbf{x}_0^2 = \begin{Bmatrix} \frac{-157}{147} \\ \frac{-83}{49} \end{Bmatrix} \quad \text{and} \quad f(\mathbf{x}_0^2) = 1.319727891 \, .$$

The direction that corresponds to the largest decrease in the objective function f during the first cycle of the univariate search is associated with the second variable. We can now decide whether we want to replace the second ($m = 2$) univariate search direction by the pattern direction or not by checking the condition stated in step 4 of the algorithm, Eq. (4.2.24). That is, Powell's criterion

$$|\alpha| = \frac{40}{49} < \left[\frac{2 - 1.319727891}{1.916666667 - 1.354166667} \right]^{\frac{1}{2}} \, . \tag{4.2.25}$$

is satisfied, therefore, we retain the old univariate search directions for the second cycle and restart the procedure by going back to step 2 of the algorithm. The results of the second cycle are tabulated in Table 4.2.1.

Table 4.2.1. Solution of the beam problem using Powell's conjugate directions method

CycleNo.	x_1	x_2	f
0	−1.0	−2.0	2.0
1	−1.083334	−2.0	1.916667
1	−1.083334	−1.625	1.354167
2	−0.895834	−1.625	0.9322967
2	−0.895834	−1.34375	0.6158854
2	−0.33334	−0.499999	−0.333333

The effectiveness of Powell's modified method can be seen to be much more pronounced on the minimization of the following function considered by Avriel [2],

$$f = (x_1 + x_2 - x_3)^2 + (x_1 - x_2 + x_3)^2 + (-x_1 + x_2 + x_3)^2 \, ,$$

and left as an exercise for the reader (see Exercise 2). • • •

Before we proceed to the discussion of first order methods, it is worthwhile to consider when zeroth order method should be used. The sequential simplex method can be used for non differentiable functions where first order methods are not appropriate. For those unconstrained minimization problems with differentiable functions, it is preferable to calculate the exact derivatives, or generate such derivatives by using finite differences and subsequently use a first order method for minimization when these derivatives can be calculated accurately. Zeroth order methods such as Powell's conjugate directions algorithm may still have a place for problems with a highly nonlinear objective functions where the accuracy of the function evaluations may be poor. The poor accuracy in function evaluations may call for high order finite difference formulae to be used for derivative calculations, therefore, the use of a zeroth order method for minimization may be a prudent alternative.

4.2.2 *First Order Methods*

First order methods for unconstrained minimization of a function f in $\mathbf{R}^n$ use the gradient of the function as well as its value in calculating the move direction for the function minimization. These methods possess a linear or a superlinear rate of convergence. A sequence $\mathbf{x}_k, k = 0, 1, 2, \ldots$, is said to be q-superlinear convergent to $\mathbf{x}^*$ of order at least p if

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c_k \|\mathbf{x}_k - \mathbf{x}^*\|^p , \tag{4.2.26}$$

where c_k converges to zero. If c_k in Eq.(4.2.26) is a constant then the convergence is said to be a simple q-order convergence of order at least p. Thus, if $p = 1$ with c_k equal to a constant then we have a linear convergence rate, whereas if $p = 1$ and c_k is a sequence that converges to zero then the convergence is said to be superlinear (see Ref. 6 for additional definitions).

Perhaps the oldest known method for minimizing a function of n variables is the steepest descent method first proposed by Cauchy [11] for solving a system of linear equations. It can be used for function minimization as follows. The direction of move is obtained by minimizing the directional derivative of f

$$\nabla f^T \mathbf{s} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} s_i , \tag{4.2.27}$$

subject to the condition that $\mathbf{s}$ be a unit vector in $\mathbf{R}^n$ in the Euclidean sense.

$$\mathbf{s}^T \mathbf{s} = 1 . \tag{4.2.28}$$

It can easily be verified (see Exercise 6) that the steepest descent direction is given by

$$\mathbf{s} = -\frac{\nabla f}{\|\nabla f\|} , \tag{4.2.29}$$

where $\| \ \|$ denotes the Euclidean norm, and it provides the largest decrease in the function f. Starting with a point $\mathbf{x}_k$ at the kth iteration of the minimization process, we obtain the next point $\mathbf{x}_{k+1}$ as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{s} . \tag{4.2.30}$$

Here $\mathbf{s}$ is given by Eq. (4.2.29) and α is determined such that f is minimized along the chosen direction by using any one of the one-dimensional minimization techniques covered in the previous section. If the function to be minimized is quadratic in $\mathbf{R}^n$ and expressed as

$$f = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c , \tag{4.2.31}$$

the step length can be determined directly by substituting Eq. (4.2.30) into Eq. (4.2.31) for the $(k+1)$st iteration followed by a minimization of f with respect to α which yields

$$\alpha^* = -\frac{(\mathbf{x}_k{}^T \mathbf{Q} + \mathbf{b}^T)\mathbf{s}}{(\mathbf{s}^T \mathbf{Q} \mathbf{s})} . \tag{4.2.32}$$

In obtaining Eq. (4.2.32) we assume that the Hessian matrix **Q** of the quadratic form is available explicitly, and we make use of the symmetry of **Q**.

The performance of the steepest descent method depends on the condition number of the Hessian matrix **Q**. The condition number of a matrix is the ratio of the largest to the smallest eigenvalue. A large condition number implies that the contours of the function to be minimized form an elongated design space, and therefore the progress made by the steepest descent method is very slow and proceeds in a zigzag pattern known as hemstitching. This is even true for quadratic functions, and can be improved by re-scaling the variables.

Example 4.2.2

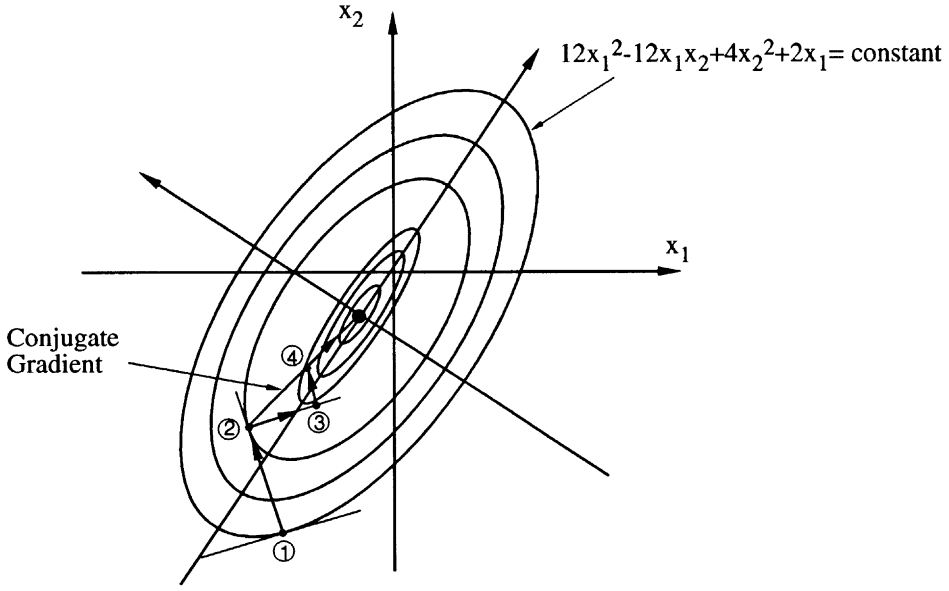


Figure 4.2.3 Contours of the cantilever beam potential energy function.

The cantilever problem discussed in the previous example illustrates this behavior most vividly. The steepest descent method when applied to this problem may exhibit the typical hemstitching phenomenon as shown in Figure 4.2.3 for certain initial starting points. However, a simple transformation of variables to improve the scaling of the variables causes the steepest descent method to converge to the minimum in a single step. For example, consider the following transformation

$$y_1 = (x_1 - \frac{1}{2}x_2), \qquad y_2 = \frac{1}{\sqrt{12}}x_2 \ . \tag{4.2.33}$$

The function f may now be expressed in terms of the new variables y_1 and y_2 as

$$f(y_1, y_2) = y_1^2 + y_2^2 + \frac{1}{6}(y_1 + \sqrt{3}y_2) \ . \tag{4.2.34}$$

As a result of the scaling and elimination of the cross-product term, the condition number of the Hessian of f is unity. Contours of the function f in the $y_1 - y_2$ plane will appear as circles. Beginning with any arbitrary starting point $\mathbf{y}_0$ and applying the steepest descent method we have

$$\mathbf{y}_1 = \mathbf{y}_0 + \alpha \begin{Bmatrix} 2y_{10} + \frac{1}{6} \\ 2y_{20} + \frac{\sqrt{3}}{6} \end{Bmatrix} . \tag{4.2.35}$$

It can be easily verified that the value of α^* that minimizes f is 0.5. Therefore,

$$\mathbf{y}_1 = \begin{Bmatrix} \frac{-1}{12} \\ \frac{-\sqrt{3}}{12} \end{Bmatrix} ,$$

at which the gradient of f is zero, implying that it is a minimum point. The corresponding values of the original variables x_1^*, and x_2^* are $-1/3$ and $-1/2$, respectively. This simple demonstration clearly shows the effectiveness of scaling in convergence of the steepest descent algorithm to the minimum of a function in $\mathbf{R}^n$. It can be shown [6] that the steepest descent method has only a linear rate of convergence in the absence of an appropriate scaling. • • •

Unfortunately, in most multivariable function minimizations it is not easy to determine the appropriate scaling transformation that leads to a one step convergence to the minimum of a general quadratic form in $\mathbf{R}^n$ using the steepest descent algorithm. This would require calculating the Hessian matrix and then performing an expensive eigenvalue analysis of the matrix. Hence, we are forced to look at other alternatives for rapid convergence to the minimum of a quadratic form. One such alternative is provided by minimizing along a set of conjugate gradient directions which guarantees a quadratic termination property. Hestenes and Stiefel [12] and later Fletcher and Reeves [13] offered such an algorithm which will be covered next.

Fletcher-Reeves' Conjugate Gradient Algorithm. This algorithm begins from an initial point $\mathbf{x}_0$ by first minimizing f along the steepest descent direction, $\mathbf{s}_0 = -\nabla f(\mathbf{x}_0) = \mathbf{g}_0$, to the new iterate $\mathbf{x}_1$. The direction for the next iteration $\mathbf{s}_1$ must be constructed so that it is Q-conjugate to $\mathbf{s}_0$ where $\mathbf{Q}$ is the Hessian of the quadratic f. The function is then minimized along $\mathbf{s}_1$ to yield the next iterate $\mathbf{x}_2$. The next direction $\mathbf{s}_2$ from $\mathbf{x}_2$ is constructed to be Q-conjugate to the previous directions $\mathbf{s}_0$ and $\mathbf{s}_1$, and the process is continued until convergence to the minimum is achieved. By virtue of Powell's theorem on conjugate directions for quadratic functions, convergence to the minimum is theoretically guaranteed at the end of the minimization of the function f along the conjugate direction $\mathbf{s}_{n-1}$. For functions which are not quadratic, conjugacy of the directions $\mathbf{s}_i, i = 1, \ldots, n$ loses its meaning since the Hessian of the functions is not a matrix of constants. However, it is a common practice to use this algorithm for non-quadratic functions. Since, for such functions, convergence to the minimum will rarely be achieved in n steps or less, the algorithm is restarted after every n steps. The basic steps of the algorithm at the $(k+1)$th iterate is as follows

1. Calculate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{k+1}\mathbf{s}_k$ where α_{k+1} is determined such that

$$\frac{\mathrm{d}f(\alpha_{k+1})}{\mathrm{d}\alpha_{k+1}} = 0 . \tag{4.2.36}$$

2. Let $\mathbf{s}_k = \mathbf{g}_k = -\nabla f(\mathbf{x}_k)$ if $k = 0$; and $\mathbf{s}_k = \mathbf{g}_k + \beta_k \mathbf{s}_{k-1}$ if $k > 0$ with

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}, \quad \text{and} \quad \mathbf{g}_k = -\nabla f(\mathbf{x}_k) \,. \tag{4.2.37}$$

3. If $\|\mathbf{g}_{k+1}\|$ or $|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)|$ is sufficiently small, then stop. Otherwise
4. If $k < n$ go to step number 1, or else restart

Example 4.2.3

We will show the effectiveness of this method on the cantilever beam problem for which we minimize

$$f = 12x_1^2 + 4x_2^2 - 12x_1x_2 + 2x_1 \,,$$

starting with the initial design point $\mathbf{x}_0^T = (-1, -2)$. The initial move direction is calculated from the gradient

$$\nabla f(\mathbf{x}_0) = \left\{ \begin{array}{c} 24x_1 - 12x_2 + 2 \\ 8x_2 - 12x_1 \end{array} \right\}_{\mathbf{x}=\mathbf{x}_0},$$

$$\mathbf{s}_0 = -\nabla f(\mathbf{x}_0) = \left\{ \begin{array}{c} -2 \\ 4 \end{array} \right\},$$

and at the end of the first step we have

$$\mathbf{x}_1 = \left\{ \begin{array}{c} -1 \\ -2 \end{array} \right\} + \alpha_1 \left\{ \begin{array}{c} -2 \\ 4 \end{array} \right\},$$

$$f(\alpha_1) = 12(-1 - 2\alpha_1)^2 + 4(-1 + 4\alpha_1)^2 - 12(-1 - 2\alpha_1)(-2 + 4\alpha_1) + 2(-1 - 2\alpha_1) \,.$$

The value of α_1 for which the function f is a minimum is obtained from the condition $\mathrm{d}f/\mathrm{d}\alpha_1 = 0$, or $\alpha_1 = 0.048077$. The new design point and the gradient at that point are

$$\mathbf{x}_1 = \left\{ \begin{array}{c} -1.0961 \\ -1.8077 \end{array} \right\}, \qquad \text{and} \quad \nabla f(\mathbf{x}_1) = \left\{ \begin{array}{c} -2.6154 \\ -1.3077 \end{array} \right\} \,.$$

Next, let $\mathbf{s}_1 = -\nabla f(\mathbf{x}_1) + \beta_1 \mathbf{s}_0$ with β_1 from Eq. (4.2.37), or

$$\beta_1 = \frac{(-2.6154)^2 + (-1.3077)^2}{(-2)^2 + (4)^2} \quad = \quad 0.4275 \,,$$

The new move direction is

$$\mathbf{s}_1 = -\left\{ \begin{array}{c} -2.6154 \\ -1.3077 \end{array} \right\} + 0.4275 \left\{ \begin{array}{c} -2 \\ 4 \end{array} \right\} = \left\{ \begin{array}{c} 1.76036 \\ 3.0178 \end{array} \right\},$$

and

$$\mathbf{x}_2 = \left\{ \begin{array}{c} -1.0961 \\ -1.8077 \end{array} \right\} + \alpha_2 \left\{ \begin{array}{c} 1.76036 \\ 3.0178 \end{array} \right\} \,.$$

Again setting $\mathrm{d}f(\alpha_2)/\mathrm{d}(\alpha_2) = 0$ we obtain $\alpha_2 = 0.4334$,

$$\mathbf{x}_2 = \begin{Bmatrix} -0.3334 \\ -0.50 \end{Bmatrix}, \qquad \text{and} \quad \nabla f(\mathbf{x}_2) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} .$$

Finally, since

$$\begin{Bmatrix} -2 \\ 4 \end{Bmatrix}^T \begin{bmatrix} 24 & -12 \\ -12 & 8 \end{bmatrix} \begin{Bmatrix} 1.76036 \\ 3.0178 \end{Bmatrix} \simeq 0 .$$

we have verified the Q-conjugacy of the two directions $\mathbf{s}_0$ and $\mathbf{s}_1$. The progress of minimization using this method is illustrated in Figure (4.2.3). • • •

Beale's Restarted Conjugate Gradient Technique. In minimizing non-quadratic functions using the conjugate gradient method, restarting the method after every n steps is not always a good strategy. Such a strategy seems to be insensitive to the nonlinear character of the function being minimized. Beale [14] and later Powell [15] have proposed restart techniques that take the nonlinearity of the function into account in deciding when to restart the algorithm. Numerical experiments with minimization of several general functions have led to the following algorithm by Powell [15].

1. Given $\mathbf{x}_0$, define $\mathbf{s}_0$ to be the steepest descent direction,

$$\mathbf{s}_0 = -\nabla f(\mathbf{x}_0) = \mathbf{g}_0 ,$$

let $k = t = 0$, and begin iterations by incrementing k.

2. For $k \geq 1$ the direction $\mathbf{s}_k$ is defined by Beale's formula [14]

$$\mathbf{s}_k = -\mathbf{g}_k + \beta_k \mathbf{s}_{k-1} + \gamma_k \mathbf{s}_t, \quad \text{and} \quad \mathbf{g}_k = -\nabla f(\mathbf{x}_k) , \tag{4.2.38}$$

where

$$\beta_k = \frac{\mathbf{g}_k^T[\mathbf{g}_k - \mathbf{g}_{k-1}]}{\mathbf{s}_{k-1}^T[\mathbf{g}_k - \mathbf{g}_{k-1}]} , \tag{4.2.39}$$

and

$$\gamma_k = \frac{\mathbf{g}_k^T[\mathbf{g}_{t+1} - \mathbf{g}_t]}{\mathbf{s}_t^T[\mathbf{g}_{t+1} - \mathbf{g}_t]}, \quad \text{if } k > t+1 , \tag{4.2.40}$$

$$\gamma_k = 0, \qquad \text{if } k = t+1 . \tag{4.2.41}$$

3. For $k \geq 1$ test the inequality

$$|\mathbf{g}_{k-1}^T \mathbf{g}_k| \geq 0.2\|\mathbf{g}_k\|^2 . \tag{4.2.42}$$

If this inequality holds, then it is taken to be an indication that enough orthogonality between $\mathbf{g}_{k-1}$and $\mathbf{g}_k$ has been lost to warrant a restart. Accordingly, t is reset $t = k - 1$ to imply restart.

4. For $k > t + 1$ the direction $\mathbf{s}_k$ is also checked to guarantee a sufficiently large gradient by testing the inequalities

$$-1.2\|\mathbf{g}_k\|^2 \leq \mathbf{s}_k^T \mathbf{g}_k \leq -0.8\|\mathbf{g}_k\|^2 . \tag{4.2.43}$$

If these inequalities are not satisfied, the algorithm is restarted by setting $t = k - 1$.

5. Finally, the algorithm is also restarted by setting $t = k - 1$, if $k - t \geq n$ as in the case of the Fletcher-Reeves method.

6. The process is terminated if $\|\mathbf{g}_{k-1}\|$ or $|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)|$ is sufficiently small. If not, k is incremented by one and the process is repeated by going to step 2.

Powell [15] has examined in great detail the effectiveness of the new restart procedure using Beale's basic algorithm on a variety of problems. These experiments clearly establish the superiority of the new procedure over the algorithms of Fletcher-Reeves and Polak-Ribiere [16]. The only disadvantage of this new algorithm appears to be its slightly increased storage requirements arising from the need for storing the vectors $\mathbf{s}_t$ and $(\mathbf{g}_{t+1} - \mathbf{g}_t)$ after a restart. More recent enhancement for the first order conjugate gradient type algorithms [17, 18] involve inclusion of certain preconditioning schemes to improve the rate of convergence.

4.2.3 *Second Order Methods*

The oldest second order method for minimizing a nonlinear multivariable function in $\mathbf{R}^n$ is Newton's method. The motivation behind Newton's method is identical to the steepest descent method. In arriving at the steepest descent direction, $\mathbf{s}$, we minimized the directional derivative, Eq. (4.2.27), subject to the condition that the Euclidean norm of $\mathbf{s}$ was unity, Eq. (4.2.28). The Euclidean norm, however, does not consider the curvature of the surface. Hence, it motivates the definition of a different norm or a metric of the surface. Thus, we pose the problem as finding the direction $\mathbf{s}$ that minimizes

$$\nabla f^T \mathbf{s} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} s_i \,, \tag{4.2.44}$$

subject to the condition that

$$\mathbf{s}^T \mathbf{Q} \mathbf{s} = 1 \,. \tag{4.2.45}$$

The solution of this problem is provided by Newton direction (see Exercise 6) to within a multiplicative constant, namely

$$\mathbf{s} = -\mathbf{Q}^{-1} \nabla f \,, \tag{4.2.46}$$

where $\mathbf{Q}$ is the Hessian of the objective function. The general form of the update equation of Newton's method for minimizing a function in $\mathbf{R}^n$ is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_{k+1} \mathbf{Q}_k^{-1} \nabla f(\mathbf{x}_k) \,, \tag{4.2.47}$$

where α_{k+1} is determined by minimizing f along the Newton direction. For $\mathbf{Q} = \mathbf{I}$, Eq. (4.2.47) yields the steepest descent solution since the norm in Eq. (4.2.45) reduces to the Euclidean norm. For quadratic functions it can be shown that the update equation reaches the optimum solution in one step with $\alpha = 1$

$$\mathbf{x}^* = \mathbf{x}_0 - [\mathbf{Q}(\mathbf{x}_0)]^{-1} [\nabla f(\mathbf{x}_0)] \,, \tag{4.2.48}$$

regardless of the initial point $\mathbf{x}_0$.

Newton's method can also shown to have a quadratic rate of convergence (see for example [4] or [8]), but the serious disadvantages of the method are the need to evaluate the Hessian $\mathbf{Q}$ and then solve the system of equations

$$\mathbf{Q}\mathbf{s} = -\nabla f\,, \tag{4.2.49}$$

to obtain the direction vector $\mathbf{s}$. For every iteration (if $\mathbf{Q}$ is non-sparse), Newton's method involves the calculation of $n(n+1)/2$ elements of the symmetric $\mathbf{Q}$ matrix, and n^3 operations for obtaining $\mathbf{s}$ from the solution of Eqs. (4.2.49). It is this feature of Newton's method that has led to the development of methods known as quasi-Newton or variable-metric methods which seek to use the gradient information to construct approximations for the Hessian matrix or its inverse.

Quasi-Newton or Variable Metric Algorithms . Consider the Taylor series expansion of the gradient of f around $\mathbf{x}_{k+1}$

$$\nabla f(\mathbf{x}_{k+1}) \simeq \nabla f(\mathbf{x}_k) + \mathbf{Q}(\mathbf{x}_{k+1} - \mathbf{x}_k)\,, \tag{4.2.50}$$

where $\mathbf{Q}$ is the actual Hessian of the function f. Assuming $\mathbf{A}_k(\mathbf{A}_k \equiv \mathbf{A}(\mathbf{x}_k))$ to be an approximation to the Hessian at the kth iteration, we may write equation (4.2.50) in a more compact form as

$$\mathbf{y}_k = \mathbf{A}_k\mathbf{p}_k\,, \tag{4.2.51}$$

where

$$\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k), \qquad \text{and} \qquad \mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k\,, \tag{4.2.52}$$

Similarly, the solution of Eq. (4.2.51) for $\mathbf{p}_k$ can be written as

$$\mathbf{B}_{k+1}\mathbf{y}_k = \mathbf{p}_k\,, \tag{4.2.53}$$

with $\mathbf{B}_{k+1}$ being an approximate inverse of the Hessian $\mathbf{Q}$. If $\mathbf{B}_{k+1}$ is to behave eventually as $\mathbf{Q}^{-1}$ then $\mathbf{B}_{k+1}\mathbf{A}_k = \mathbf{I}$. Equation (4.2.53) is known as the quasi-Newton or the secant relation. The basis for all variable-metric or quasi-Newton methods is that, the formulae which update the matrix $\mathbf{A}_k$ or its inverse $\mathbf{B}_k$ must satisfy Eq. (4.2.53) and, in addition, maintain the symmetry and positive definiteness properties. In other words, if $\mathbf{A}_k$ or $\mathbf{B}_k$ are positive definite then $\mathbf{A}_{k+1}$ or $\mathbf{B}_{k+1}$ must remain so.

A typical variable-metric algorithm with an inverse Hessian update may be stated as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_{k+1}\mathbf{s}_k\,, \tag{4.2.54}$$

where

$$\mathbf{s}_k = -\mathbf{B}_k\nabla f(\mathbf{x}_k)\,, \tag{4.2.55}$$

with $\mathbf{B}_k$ being a positive definite symmetric matrix.

Rank-One Updates. In the class of rank-one updates we have the well-known symmetric Broyden's update [19] for $\mathbf{B}_{k+1}$ given as

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{p}_k - \mathbf{B}_k\mathbf{y}_k)(\mathbf{p}_k - \mathbf{B}_k\mathbf{y}_k)^T}{(\mathbf{p}_k - \mathbf{B}_k\mathbf{y}_k)^T\mathbf{y}_k}\,. \tag{4.2.56}$$

To start the algorithm, an initial positive definite symmetric matrix $\mathbf{B}_0$ is assumed and the next point $\mathbf{x}_1$ is calculated from Eq. (4.2.54). Then, Eq. (4.2.56) is used to calculate the updated approximate inverse Hessian matrix. It is easy to verify that the columns of the second matrix on the right-hand side of Eq. (4.2.56) are multiples of each other. In other words, the update matrix has a single independent column and, hence is rank-one. Furthermore, if $\mathbf{B}_k$ is symmetric then $\mathbf{B}_{k+1}$ will also be symmetric. It is, however, not guaranteed that $\mathbf{B}_{k+1}$ will remain positive definite even if $\mathbf{B}_k$ is. This fact can lead to a breakdown of the algorithm especially when applied to general non-quadratic functions. Broyden [19] suggests choosing the step lengths α_{k+1} in Eq. (4.2.54) by either (i) an exact line search, or by (ii) $\alpha_{k+1} = 1$ for all steps, or by (iii) choosing α_{k+1} such that $\|\nabla f\|$ is minimized or reduced.

Irrespective of the type of line search used, Broyden's update guarantees a quadratic termination property. However, because of the lack of robustness in minimizing general non-quadratic functions, rank-one updates have been superseded by rank-two updates which guarantee both symmetry and positive definiteness of the updated matrices.

Rank-Two Updates. Rank-two updates for the inverse Hessian approximation may generally be written as

$$\mathbf{B}_{k+1} = \left[\mathbf{B}_k - \frac{\mathbf{B}_k\mathbf{y}_k\mathbf{y}_k^T\mathbf{B}_k}{\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k} + \theta_k\mathbf{v}_k\mathbf{v}_k^T\right]\rho_k + \frac{\mathbf{p}_k\mathbf{p}_k^T}{\mathbf{p}_k^T\mathbf{y}_k}, \tag{4.2.57}$$

where

$$\mathbf{v}_k = (\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k)^{\frac{1}{2}}\left[\frac{\mathbf{p}_k}{\mathbf{p}_k^T\mathbf{y}_k} - \frac{\mathbf{B}_k\mathbf{y}_k}{\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k}\right], \tag{4.2.58}$$

and θ_k and ρ_k are scalar parameters that are chosen appropriately. Updates given by Eqs. (4.2.57) and (4.2.58) are subsets of Huang's family of updates [20] which guarantee that $\mathbf{B}_{k+1}\mathbf{y}_k = \mathbf{p}_k$ for all choices of θ_k and ρ_k. If we set $\theta_k = 0$ and $\rho_k = 1$ for all k we obtain the Davidon-Fletcher-Powell's (DFP) update formula [21, 22] which is given as

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k\mathbf{y}_k\mathbf{y}_k^T\mathbf{B}_k}{\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k} + \frac{\mathbf{p}_k\mathbf{p}_k^T}{\mathbf{p}_k^T\mathbf{y}_k}. \tag{4.2.59}$$

The DFP update formula preserves the positive definiteness and symmetry of the matrices $\mathbf{B}_k$, and has some other interesting properties as well. When used for minimizing quadratic functions, it generates Q-conjugate directions and, therefore, at the nth iteration $\mathbf{B}_n$ becomes the exact inverse of the Hessian $\mathbf{Q}$. Thus, it has the features of the conjugate gradient as well as the Newton-type algorithms. The DFP algorithm can be used without an exact line search in determining α_{k+1} in Eq. (4.2.54). However, the step length must guarantee a reduction in the function value, and must be such that $\mathbf{p}_k^T\mathbf{y}_k > 0$ in order to maintain positive definiteness of $\mathbf{B}_k$. The performance of the algorithm, however, was shown to deteriorate as the accuracy of the line search decreases [20]. In most cases the DFP formula works quite successfully. In a few cases the algorithm has been known to break down because $\mathbf{B}_k$ became singular. This has led to the introduction of another update formula developed simultaneously

by Broyden [19], Fletcher [23], Goldfarb [24], and Shanno [25] and known known as BFGS formula. This formula can be obtained by putting $\theta_k = 1$ and $\rho_k = 1$ in Eq. (4.2.57) which reduces to

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \left[1 + \frac{\mathbf{y}_k^T \mathbf{B}_k \mathbf{y}_k}{\mathbf{p}_k^T \mathbf{y}_k}\right] \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{y}_k} - \frac{\mathbf{p}_k \mathbf{y}_k^T \mathbf{B}_k}{\mathbf{p}_k^T \mathbf{y}_k} - \frac{\mathbf{B}_k \mathbf{y}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{y}_k} . \tag{4.2.60}$$

Equation (4.2.60) can also be written in a more compact manner as

$$\mathbf{B}_{k+1} = \left[\mathbf{I} - \frac{\mathbf{p}_k \mathbf{y}_k^T}{\mathbf{p}_k^T \mathbf{y}_k}\right] \mathbf{B}_k \left[\mathbf{I} - \frac{\mathbf{y}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{y}_k}\right] + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{y}_k} . \tag{4.2.61}$$

Using $\mathbf{A}_{k+1} = \mathbf{B}_{k+1}^{-1}$ and $\mathbf{A}_k = \mathbf{B}_k^{-1}$ we can invert the above formula to arrive at an update for the Hessian approximations. It is found that this update formula reduces to

$$\mathbf{A}_{k+1} = \mathbf{A}_k - \frac{\mathbf{A}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{A}_k}{\mathbf{p}_k^T \mathbf{A}_k \mathbf{p}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{p}_k} , \tag{4.2.62}$$

which is the analog of the DFP formula (4.2.59) with $\mathbf{B}_k$ replaced by $\mathbf{A}_k$, and $\mathbf{p}_k$ and $\mathbf{y}_k$ interchanged. Conversely, if the inverse Hessian $\mathbf{B}_k$ is updated by the DFP formula then the Hessian $\mathbf{A}_k$ is updated according to an analog of the DFP formula. It is for this reason that the BFGS formula is often called the complementary DFP formula. Numerical experiments with BFGS algorithm [26] suggest that it is superior to all known variable-metric algorithms. We will illustrate its use by minimizing the potential energy function of the cantilever beam problem.

Example 4.2.4

Minimize $f(x_1, x_2) = 12x_1^2 + 4x_2^2 - 12x_1x_2 + 2x_1$ by using the BFGS update algorithm with exact line searches starting with the initial guess $\mathbf{x}_0^T = (-1, -2)$.

We initiate the algorithm with a line search along the steepest descent direction. This is associated with the assumption that $\mathbf{B}_0 = \mathbf{I}$ which is symmetric and positive definite. The resulting point is previously calculated in example 4.2.3 to be

$$\mathbf{x}_1 = \begin{Bmatrix} -1.0961 \\ -1.8077 \end{Bmatrix}, \qquad \text{and} \quad \nabla f(\mathbf{x}_1) = \begin{Bmatrix} -2.6154 \\ -1.3077 \end{Bmatrix} .$$

From Eq. (4.2.52) we calculate

$$\mathbf{p}_0 = \begin{Bmatrix} -1.0961 \\ -1.8077 \end{Bmatrix} - \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} = \begin{Bmatrix} -0.0961 \\ 0.1923 \end{Bmatrix} ,$$

$$\mathbf{y}_0 = \begin{Bmatrix} -2.6154 \\ -1.3077 \end{Bmatrix} - \begin{Bmatrix} 2 \\ -4 \end{Bmatrix} = \begin{Bmatrix} -4.6154 \\ 2.6923 \end{Bmatrix} .$$

Substituting the terms

$$\mathbf{p}_0^T \mathbf{y}_0 = (-0.0961)(-4.6154) + (0.1923)(2.6923) = 0.96127 ,$$

$$\mathbf{p}_0\mathbf{y}_0^T = \begin{Bmatrix} -0.0961 \\ 0.1923 \end{Bmatrix} [-4.6154 \quad 2.6923] = \begin{bmatrix} 0.44354 & -0.25873 \\ -0.88754 & 0.51773 \end{bmatrix},$$

into Eq. (4.2.61), we obtain

$$\mathbf{B}_1 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{0.96127} \begin{bmatrix} 0.44354 & -0.25873 \\ -0.88754 & 0.51773 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\times \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{0.96127} \begin{bmatrix} 0.44354 & -0.88754 \\ -0.25873 & 0.51773 \end{bmatrix} \right) + \frac{1}{0.96127} \begin{bmatrix} 0.00923 & -0.01848 \\ -0.01848 & 0.03698 \end{bmatrix},$$

$$= \begin{bmatrix} 0.37213 & 0.60225 \\ 0.60225 & 1.10385 \end{bmatrix}.$$

Next, we calculate the new move direction from Eq. (4.2.55)

$$\mathbf{s}_1 = - \begin{bmatrix} 0.37213 & 0.60225 \\ 0.60225 & 1.10385 \end{bmatrix} \begin{Bmatrix} -2.6154 \\ -1.3077 \end{Bmatrix} = \begin{Bmatrix} 1.7608 \\ 3.0186 \end{Bmatrix},$$

and obtain

$$\mathbf{x}_2 = \begin{Bmatrix} -1.0961 \\ -1.8077 \end{Bmatrix} + \alpha_2 \begin{Bmatrix} 1.7608 \\ 3.0186 \end{Bmatrix}.$$

Setting the derivative of $f(\mathbf{x}_2)$ with respect to α_2 to 0 yields the value $\alpha_2 = 0.4332055$, and

$$\mathbf{x}_2 = \begin{Bmatrix} -0.3333 \\ -0.5000 \end{Bmatrix}, \quad \text{with} \quad \nabla f(\mathbf{x}_2) \simeq \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}.$$

This implies convergence to the exact solution. It is left to the reader to verify that if $\mathbf{B}_1$ is updated once more we obtain

$$\mathbf{B}_2 = \begin{bmatrix} 0.1667 & 0.25 \\ 0.25 & 0.5 \end{bmatrix},$$

which is the exact inverse of the Hessian matrix

$$\mathbf{Q} = \begin{bmatrix} 24 & -12 \\ -12 & 8 \end{bmatrix}.$$

It can also be verified that, as expected, the directions $\mathbf{s}_0$ and $\mathbf{s}_1$ are Q-conjugate.
• • •

Q-conjugacy of the directions of travel has meaning only for quadratic functions, and is guaranteed for such problems in the case of variable-metric algorithms belonging to Huang's family only if the line searches are exact. In fact, Q-conjugacy of the directions is not necessary for ensuring a quadratic termination property [26]. This realization has led to the development of methods based on the DFP and BFGS formulae that abandon the computationally expensive exact line searches. The line searches must be such that they guarantee positive definiteness of the $\mathbf{A}_k$ or $\mathbf{B}_k$ matrices while reducing the function value appropriately. Positive definiteness is

guaranteed as long as $\mathbf{p}_k^T \mathbf{y}_k > 0$. To ensure a wide radius of convergence for a quasi-Newton method, it is also necessary to satisfy the following two criteria. First, a sufficiently large decrease in the function f must be achieved for the step taken and, second, the rate of decrease of f in the direction $\mathbf{s}_k$ at $\mathbf{x}_{k+1}$ must be smaller than the rate of decrease of f at $\mathbf{x}_k$ [26]. In view of this observations, most algorithms with inexact line searches require the satisfaction of the following two conditions.

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) + 10^{-4} \alpha \mathbf{s}^T \nabla f(\mathbf{x}_k) \,, \tag{4.2.63}$$

and

$$\left| \frac{\mathbf{s}^T \nabla f(\mathbf{x}_{k+1})}{\mathbf{s}^T \nabla f(\mathbf{x}_k)} \right| < 0.9 \,. \tag{4.2.64}$$

The convergence of the BFGS algorithm under these conditions has been studied by Powell [27]. Similar convergence studies with Beale's restarted conjugate gradient method under the same two conditions have been carried out by Shanno [28].

4.2.4 *Applications to Analysis*

Several of the algorithms for unconstrained minimization of functions in $\mathbf{R}^n$ can also be used for solving a system of linear or nonlinear equations. In some cases, like the problems of nonlinear structural analysis, the necessary condition for the potential energy to be stationary is that its gradient vanish. The latter can be construed as solving a system of equations of the type

$$\nabla f(\mathbf{x}) = g(\mathbf{x}) = 0 \,, \tag{4.2.65}$$

where the Hessian of f and the Jacobian of $\mathbf{g}$ are the same. In cases where the problems are posed directly as

$$\mathbf{g}(\mathbf{x}) = 0 \,, \tag{4.2.66}$$

Dennis and Schnabel [6] and others solve Eq. (4.4.2) by minimizing the nonlinear least squares function

$$f = \frac{1}{2} \mathbf{g}^T \mathbf{g} \,. \tag{4.2.67}$$

In this case, however, the Hessian of f and the Jacobian of $\mathbf{g}$ are not identical but a positive definite approximation to the Hessian of f appropriate for most minimization schemes can be easily generated from the Jacobian of $\mathbf{g}$ [6]. Minimization of f then permits the determination of not only stable but also unstable equilibrium configurations provided the minimization does not converge to a local minimum. In the case of convergence to a local minimum, certain restart [6] or deflation and tunnelling techniques [29, 30] can be invoked to force convergence to the global minimum of f at which $\|\mathbf{g}\| = 0$.

4.3 Specialized Quasi-Newton Methods

4.3.1 *Exploiting Sparsity*

The rank-one and rank-two updates that we discussed in the previous section yield updates which are symmetric but not necessarily sparse. In other words the Hessian or Hessian inverse updates lead to symmetric matrices which are fully populated. In most structural analysis problems using the finite element method it is well known that the Hessian of the potential energy (the tangent stiffness matrix) is sparse. This may be also true of many structural optimization problems. For such sparse systems the solution phase for finite element models exploits the triple $\mathbf{LDL}^{\mathrm{T}}$ factorization. Thus the Hessian or the Hessian inverse updates discussed previously are not appropriate for solving large-scale structural analysis problems which involve sparse Hessians.

In applying the BFGS method for solving large-scale nonlinear problems of structural analysis Matthies and Strang [31] have proposed an alternate implementation of the method suitable for handling large sparse problems by storing the vectors

$$\mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \,, \tag{4.3.1}$$

and

$$\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \,, \tag{4.3.2}$$

and reintroducing them to compute the new search directions. After a sequence of five to ten iterations during which the BFGS updates are used, the stiffness matrix is recomputed and the update information is deleted.

Sparse updates for solving large-scale problems were perhaps first proposed by Schubert [32], who proposed a modification of Broyden's method [33] according to which the ith row of the Hessian $\mathbf{A}_{k+1}$ is updated by using

$$\mathbf{A}_{k+1}^{(i)} = \mathbf{A}_k^{(i)} + \frac{[\mathbf{g}_i(\mathbf{x}_{k+1}) - (1-\alpha_k)\mathbf{g}(\mathbf{x}_k)]}{\alpha_k \hat{\mathbf{p}}_k^T \hat{\mathbf{p}}_k} \,, \tag{4.3.3}$$

with $\hat{\mathbf{p}}_k$ obtained from $\mathbf{p}_k$ by setting to zero those components corresponding to known zeros in $\mathbf{A}_k^{(i)}$. The method has the drawback, however, that it cannot retain symmetry of the resulting matrix even when starting with a symmetric, positive definite matrix. Not only does this result in slightly increased demands on storage, but it also requires special sparse linear equation solvers. Toint [34] and Shanno [35] have recently proposed algorithms which find updating formulae for symmetric Hessian matrices that preserve known sparsity conditions. The update is obtained by calculating the smallest correction subject to linear constraints that include the sparsity conditions. This involves the solution of a system of equations with the same sparsity pattern as the Hessian.

Curtis, Powell and Reid [36], and Powell and Toint [37] have proposed finite difference strategies for the direct evaluation of sparse Hessians of functions. In addition to using the finite difference operations, they used concepts from graph theory that minimize the number of gradient evaluations required for computing the few non-zero entries of a sparse Hessian. By using these strategies, we can exploit the sparsity not only in the computation of the Newton direction but also in the formation of Hessians [38, 39]

The Curtis-Powell-Reid (CPR) strategy exploits sparsity, but not the symmetry of the Hessian. It divides the columns of the Hessian into groups, so that in each group the row numbers of the unknown elements of the column vectors are all different. After the formation of the first group, other groups are formed successively by applying the same strategy to columns not included in the previous groups. The number of such groups for sparse or banded matrices is usually very small by comparison with n. To evaluate the Hessian of f at $\mathbf{x}_0$ we evaluate the gradient of f at $\mathbf{x}_0$. After this initial gradient evaluation, only as many more gradient evaluations as the number of groups are needed to evaluate all the non-zero elements of the Hessian using forward difference approximation. Thus

$$a_{ij} = \frac{\partial g_i}{\partial x_j} = \frac{\mathbf{g}_i(\mathbf{x}_0 + h_j \mathbf{e}_j) - \mathbf{g}_i(\mathbf{x}_0)}{h_j}\,, \tag{4.3.4}$$

where $\mathbf{e}_j$ is the jth coordinate vector and h_j is a suitable step size. Each step size may be adjusted such that the greatest ratio of the round-off to truncation error for any column of the Hessian falls within a specified range. However, such an adjustment of step sizes would require a significantly large number of gradient evaluations. Hence, to economize on the number of gradient evaluations the step sizes are not allowed to leave the range

$$[\max(\epsilon|\mathbf{x}_j|, \eta h_{uj}), h_{uj}]\,, \tag{4.3.5}$$

where ϵ is the greatest relative round-off in a single operation, η is the relative machine precision, and h_{uj} is an upper bound on h_j [36].

Powell and Toint [37] extended the CPR strategy to exploit symmetry of the Hessian. They proposed two methods, one of which is known as the substitution method. According to this, the CPR strategy is first applied to the lower triangular part, $\mathbf{L}$, of the symmetric Hessian, $\mathbf{A}$. Because, all the elements of $\mathbf{A}$ computed this way will not be correct, the incorrect elements are corrected by a back-substitution scheme. Details of this back-substitution scheme may be found in Ref. 37.

The Powell-Toint (PT) strategy of estimating sparse Hessians directly appears to be a much better alternative to Toint's sparse update algorithm [38]. One major drawback of Toint's update algorithm is that the updated Hessian approximation is not guaranteed to remain positive definite even if the initial Hessian approximation was positive definite.

4.3.2 *Coercion of Hessians for Suitability with Quasi-Newton Methods*

In minimizing a multivariable function f using a discrete Newton method or the Toint's update algorithm we must ensure that the Hessian approximation is positive

definite. If this is not so, then Newton's direction is not guaranteed to be a descent direction. There are several strategies for coercing an indefinite Hessian to a positive definite form. Prominent among these strategies is the one proposed by Gill and Murray [40]. The most impressive feature of this strategy is that the coercion of the Hessian takes place during its $\mathbf{LDL}^T$ decomposition for the computation of the Newton direction. The diagonal elements of the $\mathbf{D}$ matrix are forced to be sufficiently positive to avoid numerical difficulties while the off-diagonal terms of $\mathbf{LD}^{1/2}$ are limited by a quantity designed to guarantee positive definiteness of the resulting matrix. This is equivalent to modifying the original non-positive definite Hessian matrix by the addition of an appropriate diagonal matrix. Because this matrix modification is carried out during its $\mathbf{LDL}^T$ decomposition, the strategy for the computation of Newton's descent direction does not entail a great deal of additional computations.

4.3.3 *Making Quasi-Newton Methods Globally Convergent*

It is well known that despite a positive definite Hessian approximation, Newton's method can diverge for some starting points. Standard backtracking along the Newton direction by choosing shorter step lengths can achieve convergence to the minimum. However, backtracking along the Newton direction fails to use the n-dimensional quadratic model of the function f. Dennis and Schnabel [7] have proposed a strategy called the double-dogleg strategy which uses the full n-dimensional quadratic model to choose a new direction obtained by a linear combination of the steepest descent and the Newton direction. This new direction is a function of the radius of the trust region within which the n-dimensional quadratic model of the function approximates the true function well. The double-dogleg strategy not only makes Newton's method globally convergent (that is converge to the minimum of the function irrespective of the starting point) but also makes it significantly more efficient for certain poorly scaled problems. For details about the double-dogleg strategy readers are advised to consult Ref. 7. More recent attempts to widen the domain of convergence of the quasi-Newton method or make it globally convergent for a wide class of problems are studied in Refs. [41, 42].

4.4 Probabilistic Search Algorithms

A common disadvantage of most of the algorithms discussed so far is their inability to distinguish local and global minima. Many structural design problems have more than one local minimum, and depending on the starting point, these algorithms may converge to one of these local minima. The simplest way to check for a better local solution is to restart the optimization from randomly selected initial points to check if other solutions are possible. However, for problems with a large number of variables the possibility of missing the global minimum is large unless unpractically large number of optimization runs are performed. The topic of global optimization is an area of active research where new algorithms are emerging and old algorithms are constantly being improved [43–45].

Dealing with the problem of local minima becomes even worse if the design variables are required to take discrete values. First of all, for such problems the design space is discontinuous and disjointed, therefore derivative information is either useless or is not defined. Secondly, the use of discrete values for the design variables introduces multiple minima corresponding to various combinations of the variables, even if the objective function for the problem has a single minimum for continuous variables. A methodical way of dealing with multiple minima for discrete optimization problems is to use either random search techniques that would sample the design space for a global minimum or to employ enumerative type algorithms. In either case, the efficiency of the solution process deteriorates dramatically as the number of variables is increased.

Two algorithms, *Simulated Annealing* and *Genetic Algorithms* (see, Laarhoven [46] and Goldberg [47], respectively), have emerged more recently as tools ideally suited for optimization problems where a global minimum is sought. In addition to being able to locate near global solutions, these two algorithms are also powerful tools for problems with discrete-valued design variables. Both algorithms rely on naturally observed phenomena and their implementation calls for the use of a random selection process which is guided by probabilistic decisions. In the following sections brief descriptions of the two algorithms are presented. Application of the algorithms to structural design will be demonstrated for laminated composites in Chapter 11.

4.4.1 *Simulated Annealing*

The development of the simulated annealing algorithm was motivated by studies in statistical mechanics which deal with the equilibrium of large number of atoms in solids and liquids at a given temperature. During solidification of metals or formation of crystals, for example, a number of solid states with different internal atomic or crystalline structure that correspond to different energy levels can be achieved depending on the rate of cooling. If the system is cooled too rapidly, it is likely that the resulting solid state would have a small margin of stability because the atoms will assume relative positions in the lattice structure to reach an energy state which is only locally minimal. In order to reach a more stable, globally minimum energy state, the process of annealing is used in which the metal is reheated to a high temperature and cooled slowly, allowing the atoms enough time to find positions that minimize a steady state potential energy. It is observed in the natural annealing process that during the time spent at a given temperature it is possible to have the system jump to a higher energy state temporarily before the steady state is reached. As will be explained in the following paragraphs, it is this characteristic of the annealing process which makes it possible to achieve near global minimum energy states.

A computational algorithm that simulates the annealing process was proposed by Metropolis et al. [48], and is referred to as the *Metropolis algorithm.* At a given temperature, T, the algorithm perturbs the position of an atom randomly and computes the resulting change in the energy of the system, ΔE. If the new energy state is lower than the initial state, then the new configuration of the atoms is accepted. If, on the other hand $\Delta E \geq 0$, the perturbed state causes an increase in the energy,

the new state might be accepted or rejected based on a random probabilistic decision. The probability of acceptance, $\mathcal{P}(\Delta E)$, of a higher energy state is computed as

$$\mathcal{P}(\Delta E) = e^{\left(\frac{-\Delta E}{k_B T}\right)}, \tag{4.4.1}$$

where k_B is the Boltzmann's constant. If the temperature of the system is high, then the probability of acceptance of a higher energy state is close to one. If, on the other hand, the temperature is close to zero, then the probability of acceptance becomes very small.

The decision to accept or reject is made by randomly selecting a number in an interval $(0, 1)$ and comparing it with $\mathcal{P}(\Delta E)$. If the number is less than $\mathcal{P}(\Delta E)$, then the perturbed state is accepted, if it is greater than $\mathcal{P}(\Delta E)$, the state is rejected. At each temperature, a pool of atomic structures would be generated by randomly perturbing positions until a steady state energy level is reached (commonly referred to as thermal equilibrium). Then the temperature is reduced to start the iterations again. These steps are repeated iteratively while reducing the temperature slowly to achieve the minimal energy state.

The analogy between the simulated annealing and the optimization of functions with many variables was established recently by Kirkpatrick et al. [49], and Cerny [50]. By replacing the energy state with an objective function f, and using variables $\mathbf{x}$ for the the configurations of the particles, we can apply the Metropolis algorithm to optimization problems. The method requires only function values. The moves in the design space from one point, $\mathbf{x}^i$ to another $\mathbf{x}^j$ causes a change in the objective function, Δf^{ij}. The temperature T now becomes a control parameter that regulates the convergence of the process. Important elements that affect the performance of the algorithm are the selection of the initial value of the "temperature", T_0, and how to update it. In addition, the number of iterations (or combinations of design variables) needed to achieve "thermal equilibrium" must be decided before the T can be reduced. These parameters are collectively referred to as the "cooling schedule".

A flow chart of a typical simulated annealing algorithm is shown in Figure 4.4.1. The definition of the cooling schedule begins with the selection of the initial temperature. If a low value of T_0 is used, the algorithm would have a low probability of reaching a global minimum. The initial value of T_0 must be high enough to permit virtually all moves in the design space to be acceptable so that almost a random search is performed. Typically, T_0 is selected such that the acceptance ratio $\mathcal{X}$ (defined as the ratio of the number of accepted moves to total number of proposed moves) is approximately $\mathcal{X}_0 = 0.95$ [51]. Johnson et al. [52] determined T_0 by calculating the average increase in the objective function, $\overline{\Delta f}^{(+)}$, over a predetermined number of moves and solved

$$\mathcal{X}_0 = e^{\left(\frac{-\overline{\Delta f}^{(+)}}{T_0}\right)}, \tag{4.4.2}$$

leading to

$$T_0 = \frac{\overline{\Delta f}^{(+)}}{\ln(\mathcal{X}_0^{-1})}. \tag{4.4.3}$$

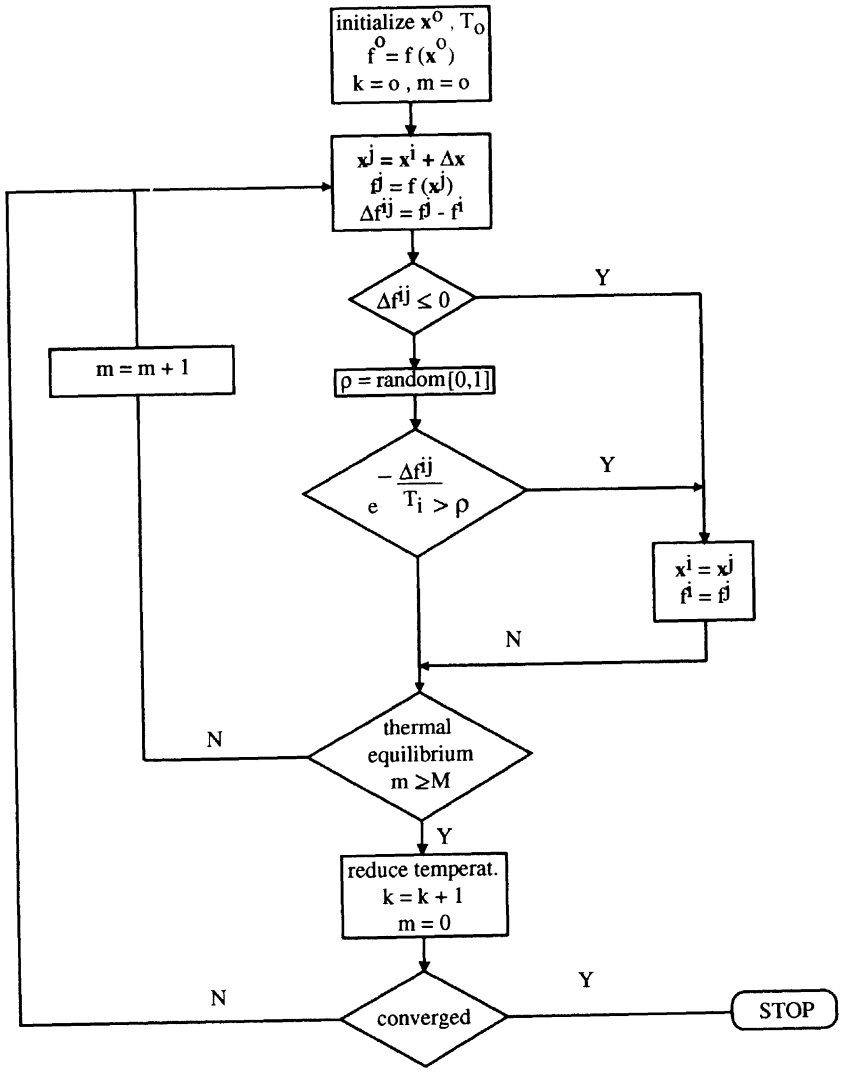


Figure 4.4.1 Flow chart of the simulated annealing algorithm.

Once the temperature is set, a number of moves in the variable space is performed by perturbing the design. The number of moves at a given temperature must be large enough to allow the solution to escape from a local minimum. One possibility is to move until the value of the objective function does not change for a specified number, M, of successive iterations. Another possibility suggested by Aarts [53] for discrete valued design variables is to make sure that every possible combinations of design variables in the neighborhood of a steady state design is visited at least once with a

probability of P. That is, if there are S neighboring designs, then

$$M = S \ln\left(\frac{1}{1-P}\right) , \tag{4.4.4}$$

where $P = 0.99$ for $S > 100$, and $P = 0.995$ for $S < 100$. For discrete valued variables there are often many options for defining the neighborhood of the design. One possibility is to define it as all the designs that can be obtained by changing one design variable to its next higher or lower value. A broader immediate neighborhood can be defined by changing more than one design variables to their next higher or lower values. For an n variable problem, the immediate neighborhood has

$$S = 3^n - 1 . \tag{4.4.5}$$

Once convergence is achieved at a given temperature, generally referred to as thermal equilibrium, the temperature is reduced and the process is repeated.

Many different schemes have been proposed for updating the temperature. A frequently used rule is a constant cooling update

$$T_{k+1} = \alpha T_k, \qquad k = 0, 1, 2, \ldots, K , \tag{4.4.6}$$

where $0.5 \leq \alpha \leq 0.95$. Nahar [54] fixes the number of decrement steps K, and suggests determination of the values of the T_k experimentally. It is also possible to divide the interval $[0, T_0]$ into a fixed K number of steps and use

$$T_K = \frac{K-k}{K} T_0, \qquad k = 1, 2, \ldots, K . \tag{4.4.7}$$

The number of intervals typically ranges from 5 to 20.

The use of simulated annealing for structural optimization has been quite recent. Elperin [55] applied the method to the design of a ten-bar truss problem where member cross-sectional dimensions were to be selected from a set of discrete values. Kincaid and Padula [56] used it for minimizing the distortion and internal forces in a truss structure. A 6-story 156 member frame structure with discrete valued variables was considered by Balling and May [57]. Optimal placement of active and passive members in a truss structure was investigated by Chen et al. [58] to maximize the finite-time energy dissipation to achieve increased damping properties.

4.4.2 *Genetic Algorithms*

Genetic algorithms use techniques derived from biology, and rely on the principle of Darwin's theory of survival of the fittest. When a population of biological creatures is allowed to evolve over generations, individual characteristics that are useful for survival tend to be passed on to the future generations, because individuals carrying them get more chances to breed. Those individual characteristics in biological populations are stored in chromosomal strings. The mechanics of natural genetics is based on operations that result in structured yet randomized exchange of genetic

information (i.e., useful traits) between the chromosomal strings of the reproducing parents, and consists of *reproduction*, *crossover*, occasional *mutation*, and *inversion* of the chromosomal strings.

Genetic algorithms, developed by Holland [59], simulate the mechanics of natural genetics for artificial systems based on operations which are the counterparts of the natural ones (even called by the same names), and are extensively used as multivariable search algorithms. As will be described in the following paragraphs, these operations involve simple, easy to program, random exchanges of location of numbers in a string, and, therefore, at the outset look like a completely random search of extremum in the parameter space based on function values only. However, genetic algorithms are experimentally proven to be robust, and the reader is referred to Goldberg [47] for further discussion of the theoretical properties of genetic algorithms. Here we discuss the genetic representation of a minimization problem, and focus on the mechanics of three commonly used genetic operations, namely; reproduction, crossover, and mutation.

Application of the operators of the genetic algorithm to a search problem first requires the representation of the possible combinations of the variables in terms of bit strings that are counterparts of the chromosomes. Naturally, the measure of goodness of a specific combination of genes is represented in an artificial system by the objective function of the search problem. For example, if we have a minimization problem

$$\textbf{minimize} \qquad f(\mathbf{x}), \qquad \mathbf{x} = \{x_1, x_2, x_3, x_4\}\,, \tag{4.4.8}$$

a binary string representation of the variable space could be of the form

$$\underbrace{0\,1\,1\,0}_{x_1}\ \underbrace{1\,0\,1}_{x_2}\ \underbrace{1\,1}_{x_3}\ \underbrace{1\,0\,1\,1}_{x_4}, \tag{4.4.9}$$

where string equivalents of the individual variables are connected head-to-tail, and, in this example, base 10 values of the variables are $x_1 = 6, x_2 = 5, x_3 = 3, x_4 = 11$, and their ranges correspond to $\{15 \geq x_1, x_4 \geq 0\}, \{7 \geq x_2 \geq 0\}$, and $\{3 \geq x_3 \geq 0\}$. Because of the bit string representation of the variables, genetic algorithms are ideally suited for problems where the variables are required to take discrete or integer variables. For problems where the design variables are continuous values within a range $x_i^L \leq x_i \leq x_i^U$, one may need to use a large number of bits to represent the variables to high accuracy. The number of bits that are needed depends on the accuracy required for the final solution. For example, if a variable is defined in a range $\{0.01 \leq x_i \leq 1.81\}$ and the accuracy needed for the final value is $x^{\text{incr}} = 0.001$, then the number of binary digits needed for an appropriate representation can be calculated from

$$2^m \geq \left((x_i^U - x_i^L)/x^{\text{incr}} + 1\right)\,, \tag{4.4.10}$$

where m is the number of digits. In this example, the smallest number of digits that satisfy the requirement would be $m = 11$, which actually produces increments of 0.00087 in the value of the variable, instead of the required value of 0.001.

Unlike the search algorithms discussed earlier that move from one point to another in the design variable space, genetic algorithms work with a population of strings

(chromosomes). This aspect of the genetic algorithms is responsible for capturing near global solutions, by keeping many solution points that may have the potential of being close to minima (local or global) in the pool during the search process rather than singling out a point early in the process and running the risk of getting stuck at a local minimum. Working on a population of designs also suggests the possibility of implementation on parallel computers. However, the concept of parallelism is even more basic to genetic algorithms in that evolutionary selection can improve in parallel many different characteristics of the design. Also, the outcome of a genetic search is a population of good designs rather than a single design. This aspect can be very useful to the designer.

Initially the size of the population is chosen and the values of the variables in each string are decided by randomly assigning 0's and 1's to the bits. The next important step in the process is *reproduction*, in which individual strings with good objective function values are copied to form a new population, an artificial version of the survival of the fittest. The bias towards strings with better performance can be achieved by increasing the probability of their selection in relation to the rest of the population. One way to achieve this is to create a biased roulette wheel where individual strings occupy areas proportional to their function values in relation to the cumulative function value of the entire population. Therefore, the population resulting from the reproduction operation would have multiple copies of the highly fit individuals.

Once the new population is generated, the members are paired off randomly for crossover. The mating of the pair also involves a random process. A random integer k between 1 and $L-1$, where L is the string length, is selected and two new strings are generated by exchanging the 0's and 1's that comes after the kth location in the first parent with the corresponding locations of the second parent. For example, the two strings of length $L = 9$

$$\begin{array}{ll} \textbf{parent 1:} & 0\,1\,1\,0\,1 \| 0\,1\,1\,1 \\ \textbf{parent 2:} & 0\,1\,0\,0\,1 \| 0\,0\,0\,1\,, \end{array} \tag{4.4.11}$$

are mated with a crossover point of $k = 5$, the offsprings will have the following composition,

$$\begin{array}{ll} \textbf{offspring 1:} & 0\,1\,1\,0\,1\,0\,0\,0\,1 \\ \textbf{offspring 2:} & 0\,1\,0\,0\,1\,0\,1\,1\,1\,. \end{array} \tag{4.4.12}$$

Multiple point crossovers in which information between the two parents are swapped among more string segments are also possible, but because of the mixing of the strings the crossover becomes a more random process and the performance of the algorithm might degrade, De Jong [60]. Exception to this is the two-point crossover. In fact, the one point crossover can be viewed as a special case of the two point crossover in which the end of the string is the second crossover point. Booker [61] showed that by choosing the end-point of the segment to be crossed randomly, the performance of the algorithm can actually be improved.

Mutation serves an important task of preventing premature loss of important genetic information by occasional introduction of random alteration of a string. As

mentioned earlier, at the end of reproduction it is possible to have populations with multiple copies of the same string. In the worst scenario, it is possible to have the entire pool to be made of the same string. In such a case, the algorithm would be unable to explore the possibility of a better solution. Mutation prevents this uniformity, and is implemented by randomly selecting a string location and changing its value from 0 to 1 or vice versa. Based on small rate of occurrence in biological systems and on numerical experiments, the role of the mutation operation on the performance of a genetic algorithm is considered to be a secondary effect. Goldberg [49] suggests a rate of mutation of one in one thousand bit operations.

Application of genetic algorithms in optimal structural design has started only recently. The first application of the algorithm to a structural design was presented by Goldberg and Samtani [62] who used the 10-bar truss weight minimization problem. More recently, Hajela [63] used genetic search for several structural design problems for which the design space is known to be either nonconvex or disjoint. Rao et al. [64] address the optimal selection of discrete actuator locations in actively controlled structures via genetic algorithms.

In closing, the basic ideas behind the simulation of a natural phenomena is finding a more mathematically sound foundation in the area of probabilistic search algorithms, especially for discrete variables. Improvements in the performance of the algorithms are constantly being made. For example, modifications in the cooling schedule proposed by Szu [65] led to the development of a new algorithm know as the fast simulated annealing. Applications and analysis of other operations that mimic the natural biological genetics (such as inversion, dominance, niches, etc.) are currently being evaluated for genetic algorithms.

4.5 Exercises

1. Solve the problem of the cantilever beam problem of example 4.2.1 by

   (a) Nelder-Mead's simplex algorithm, and

   (b) Davidon-Fletcher-Powell's algorithm.

   Begin with $\mathbf{x}_0^T = (-1, -2)$. For the simplex algorithm assume an initial simplex of size a=2.0. Assume an initial base point $\mathbf{x}_0$ with the coordinates of the other vertices to be given by Eqs. (4.2.1) and (4.2.1).

2. Find the minimum of the function

$$f = (x_1 + x_2 - x_3)^2 + (x_1 - x_2 + x_3)^2 + (-x_1 + x_2 + x_3)^2 ,$$

using Powell's conjugate directions method, starting with $\mathbf{x}_0 = (0, 0, 0)^T$.

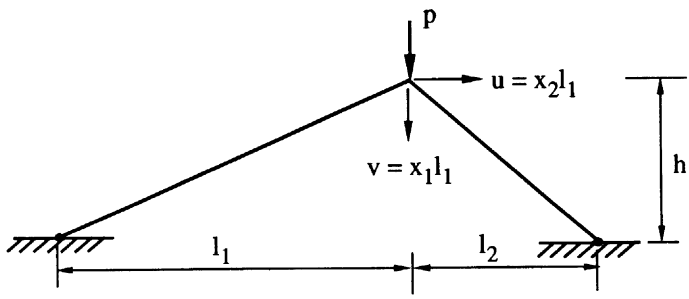


Figure 4.5.1 Two bar unsymmetric shallow truss.

3. Determine the minimum of

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 ,$$

using steepest descent method, starting with $\mathbf{x}_0 = (1.2, 1.0)^T$.

4. The stable equilibrium configuration of the two bar unsymmetric shallow truss of Figure 4.5.1 can be obtained by minimizing the potential energy function f of the non-dimensional displacement variables x_1, x_2 as

$$f(x_1, x_2) = \frac{1}{2} m\gamma(-\alpha_1 x_1 + \frac{1}{2}x_1^2 + x_2)^2 + \frac{1}{2}\left(-\alpha_1 x_1 + \frac{1}{2}x_1^2 - \frac{x_2}{\gamma}\right)\gamma^4 - \overline{p}\gamma x_1 ,$$

where $m, \gamma, \alpha_1, \overline{p}$ are nondimensional quantities defined as

$$m = \frac{A_1}{A_2}, \qquad \gamma = \frac{l_1}{l_2}, \qquad \alpha_1 = \frac{h}{l_1}, \qquad \overline{p} = \frac{p}{EA_2} ,$$

and E is the elastic modulus, A_1 and A_2 are the cross-sectional areas of the bars. Using the BFGS algorithm determine the equilibrium configuration in terms of x_1 and x_2 for $m = 5, \gamma = 4, \alpha_1 = 0.02, \overline{p} = 2 \times 10^{-5}$. Use $\mathbf{x}_0^T = (0, 0)$.

5. Continuing the analysis of the problem 4 it can be shown that the critical load p_{cr} at which the shallow truss is unstable (snap-through instability) is given by

$$p_{cr} = \frac{EA_1A_2\gamma(\gamma+1)^2}{(A_1 + A_2\gamma)} \frac{\alpha_1^3}{3\sqrt{3}} .$$

Suppose now that p_{cr} as given above is to be maximized subject to the condition that

$$A_1 l_1 + A_2 l_2 = v_0 = \text{constant} .$$

The exterior penalty formulation of Chapter 5 reduces the above problem to the unconstrained minimization of

$$p_{cr}^*(A_1, A_2, r) = \frac{EA_1A_2\gamma(\gamma+1)^2}{(A_1 + A_2\gamma)} \frac{\alpha_1^3}{3\sqrt{3}} + r(A_1 l_1 + A_2 l_2 - v_0)^2 ,$$

where r is a penalty parameter. Carry out the minimization of an appropriately nondimensionalized form of p_{cr}^* for $l_1 = 200$ in, $l_2 = 50$ in, $h = 2.50$ in, $v_0 = 200$ in^3, $E = 10^6$ psi, $r = 10^4$ to determine an approximate solution for the optimum truss configuration and the corresponding value of p_{cr}. Use the BFGS algorithm for unconstrained minimization beginning with an initial feasible guess of $A_1 = 0.952381\text{in}^2$ and $A_2 = 0.190476\text{in}^2$.

6. a) Minimize the directional derivative of f in the direction $\mathbf{s}$

$$\nabla f^T \mathbf{s} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} s_i ,$$

subject to the condition

$$\sum_{i=1}^{n} s_i^2 = 1 ,$$

to show that the steepest descent direction is given by

$$\mathbf{s} = -\frac{\nabla f}{\|\nabla f\|} . \tag{4.5.1}$$

b) Repeat the above with the constraint condition on $\mathbf{s}$ replaced by

$$\mathbf{s}^T \mathbf{Q} \mathbf{s} = 1 ,$$

to show that the Newton direction is given by

$$\mathbf{s} = -\mathbf{Q}^{-1} \nabla f ,$$

$\mathbf{Q}$ being the Hessian of the quadratic function f.

4.6 References

[1] Kamat, M.P. and Hayduk, R.J., "Recent Developments in Quasi–Newton Methods for Structural Analysis and Synthesis," AIAA J., 20 (5), 672–679, 1982.

[2] Avriel, M., Nonlinear Programming: Analysis and Methods. Prentice–Hall, Inc., 1976.

[3] Powell, M.J.D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives," Computer J., 7, pp. 155–162, 1964.

[4] Kiefer, J., "Sequential Minmax Search for a Maximum," Proceedings of the American Mathematical Society, 4, pp. 502–506, 1953.

[5] Walsh, G.R., Methods of Optimization, John Wiley, New York, 1975.

[6] Dennis, J.E. and Schnabel, R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice–Hall, 1983.

[7] Gill, P.E., Murray, W. and Wright, M.H., Practical Optimization, Academic Press, New York, p. 92, 1981.

[8] Spendley, W., Hext, G. R., and Himsworth, F. R., "Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation," Technometrics, 4 (4), pp. 441–461, 1962.

[9] Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization," Computer J., 7, pp. 308–313, 1965.

[10] Chen, D. H., Saleem, Z., and Grace, D. W., "A New Simplex Procedure for Function Minimization," Int. J. of Modelling & Simulation, 6, 3, pp. 81–85, 1986.

[11] Cauchy, A., "Methode Generale pour la Resolution des Systemes D'equations Simultanees," Comp. Rend. l'Academie des Sciences Paris, 5, pp. 536–538, 1847.

[12] Hestenes, M.R. and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems," J. Res. Nat. Bureau Stand., 49, pp. 409–436, 1952.

[13] Fletcher, R. and Reeves, C.M., "Function Minimization by Conjugate Gradients," Computer J., 7, pp. 149–154, 1964.

[14] Gill, P.E. and Murray, W.,"Conjugate-Gradient Methods for Large Scale Nonlinear Optimization," Technical Report 79-15; Systems Optimization Lab., Dept. of Operations Res., Stanford Univ., pp. 10–12, 1979.

[15] Powell, M.J.D., "Restart Procedures for the Conjugate Gradient Method," Math. Prog., 12, pp. 241–254, 1975.

[16] Polak, E., Computational Methods in Optimization: A Unified Approach, Academic Press, 1971.

[17] Axelsson, O. and Munksgaard, N., "A Class of Preconditioned Conjugate Gradient Methods for the Solution of a Mixed Finite Element Discretization of the Biharmonic Operator," Int. J. Num. Meth. Engng., 14, pp. 1001–1019, 1979.

[18] Johnson, O.G., Micchelli, C.A. and Paul, G., "Polynomial Preconditioners for Conjugate Gradient Calculations," SIAM J. Num. Anal., 20 (2), pp. 362–376, 1983.

[19] Broyden, C.G., "The Convergence of a Class of Double–Rank Minimization Algorithms 2. The New Algorithm," J. Inst. Math. Appl., 6, pp. 222–231, 1970.

[20] Oren, S.S. and Luenberger, D., "Self–scaling Variable Metric Algorithms, Part I," Manage. Sci., 20 (5), pp. 845–862, 1974.

[21] Davidon, W.C., Variable Metric Method for Minimization. Atomic Energy Commission Research and Development Report, ANL–5990 (Rev.), November 1959.

[22] Fletcher, R. and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization," Comput. J., 6, pp. 163–168, 1963.

[23] Fletcher, R., "A New Approach to Variable Metric Algorithms," Computer J., 13 (3), pp. 317–322, 1970.

[24] Goldfarb, D., "A Family of Variable-metric Methods Derived by Variational Means," Math. Comput., 24, pp. 23–26, 1970.

[25] Shanno, D.F., "Conditioning of Quasi–Newton Methods for Function Minimization," Math. Comput., 24, pp. 647–656, 1970.

[26] Dennis, J.E., Jr. and More, J.J., "Quasi-Newton Methods, Motivation and Theory," SIAM Rev., 19 (1), pp. 46–89, 1977.

[27] Powell, M.J.D., "Some Global Convergence Properties of a Variable Metric Algorithm for Minimization Without Exact Line Searches," In: Nonlinear Programming (R.W.Cottle and C.E. Lemke, eds.), American Mathematical Society, Providence, RI, pp. 53–72, 1976.

[28] Shanno, D.F., "Conjugate Gradient Methods with Inexact Searches," Math. Oper. Res., 3 (2), pp. 244–256, 1978.

[29] Kamat, M.P., Watson, L.T. and Junkins, J.L., "A Robust Efficient Hybrid Method for Finding Multiple Equilibrium Solutions," Proceedings of the Third Intl. Conf. on Numerical Methods in Engineering, Paris, France, pp. 799–807, March 1983.

[30] Kwok, H.H., Kamat, M.P. and Watson, L.T., "Location of Stable and Unstable Equilibrium Configurations using a Model Trust Region, Quasi-Newton Method and Tunnelling," Computers and Structures, 21 (6), pp. 909–916, 1985.

[31] Matthies, H. and Strang, G., "The Solution of Nonlinear Finite Element Equations," Int. J. Num. Meth. Enging., 14, pp. 1613–1626, 1979.

[32] Schubert, L.K., "Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian," Math. Comput., 24, pp. 27–30, 1970.

[33] Broyden, C.G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comput., 19, pp. 577–593, 1965.

[34] Toint, Ph.L., "On Sparse and Symmetric Matrix Updating Subject to a Linear Equation," Math. Comput., 31, pp. 954–961, 1977.

[35] Shanno, D.F., "On Variable-Metric Methods for Sparse Hessians," Math. Comput., 34, pp. 499–514, 1980.

[36] Curtis, A.R., Powell, M.J.D. and Reid, J.K., "On the Estimation of Sparse Jacobian Matrices," J. Inst. Math. Appl., 13, pp. 117–119, 1974.

[37] Powell, M.J.D. and Toint, Ph.L., "On the Estimation of Sparse Hessian Matrices," SIAM J. Num. Anal., 16 (6), pp. 1060–1074, 1979.

[38] Kamat, M.P., Watson, L.T. and VandenBrink, D.J., "An Assessment of Quasi-Newton Sparse Update Techniques for Nonlinear Structural Analysis," Comput. Meth. Appl. Mech. Enging., 26, pp. 363–375, 1981.

[39] Kamat, M.P. and VandenBrink, D.J., "A New Strategy for Stress Analysis Using the Finite Element Method," Computers and Structures 16 (5), pp. 651–656, 1983.

[40] Gill, P.E. and Murray, W., "Newton-type Methods for Linearly Constrained Optimization," In: Numerical Methods for Constrained Optimization (Gill & Murray, eds.), pp. 29–66. Academic Press, New York 1974.

[41] Griewank, A.O., Analysis and Modifications of Newton's Method at Singularities. Ph.D. Thesis, Australian National University, 1980.

[42] Decker, D.W. and Kelley, C.T., "Newton's Method at Singular Points, I and II," SIAM J. Num. Anal., 17, pp. 66–70; 465–471, 1980.

[43] Hansen, E., "Global Optimization Using Interval Analysis– The Multi Dimensional Case," Numer. Math., 34, pp. 247–270, 1980.

[44] Kao, J.-J., Brill, E. D., Jr., and Pfeffer, J. T., "Generation of Alternative Optima for Nonlinear Programming Problems," Eng. Opt., 15, pp. 233–251, 1990.

[45] Ge, R., "Finding More and More Solutions of a System of Nonlinear Equations," Appl. Math. Computation, 36, pp. 15-30, 1990.

[46] Laarhoven, P. J. M. van., and Aarts, E., Simulated Annealing: Theory and Applications, D. Reidel Publishing, Dordrecht, The Netherlands, 1987.

[47] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co. Inc., Reading, Massachusetts, 1989.

[48] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., "Equation of State Calculations by Fast Computing Machines," J. Chem. Physics, 21 (6), pp. 1087–1092, 1953.

[49] Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., "Optimization by Simulated Annealing," Science, 220 (4598), pp. 671–680, 1983.

[50] Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," J. Opt. Theory Appl., 45, pp. 41–52, 1985.

[51] Rutenbar, R. A., "Simulated Annealing Algorithms: An Overview," IEEE Circuits and Devices, January, pp. 19–26, 1989.

[52] Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C., "Optimization by Simulated Annealing: An Experimental Evaluation. Part I. Graph Partitioning," Operations Research, 37, 1990, pp. 865–893.

[53] Aarts, E., and Korst, J., Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing, John Wiley & Sons, 1989.

[54] Nahar, S., Sahni, S., and Shragowithz, E. V., in the Proceedings of 22nd Design Automation Conf., Las Vegas, June 1985, pp. 748-752.

[55] Elperin, T, "Monte Carlo Structural Optimization in Discrete Variables with Annealing ALgorithm," Int. J. Num. Meth. Eng., 26, 1988, pp. 815–821.

[56] Kincaid, R. K., and Padula, S. L., "Minimizing Distortion and Internal Forces in Truss Structures by Simulated Annealing," Proceedings of the AIAA/ASME /ASCE/AHS/ASC 31st Structures, Structural Dynamics, and Materials Conference, Long Beach, CA., 1990, Part 1, pp. 327–333.

[57] Balling, R. J., and May, S. A., "Large-Scale Discrete Structural Optimization: Simulated Annealing, Branch-and-Bound, and Other Techniques," presented at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Long Beach, CA., 1990,

[58] Chen, G.-S., Bruno, R. J., and Salama, M.,"Optimal Placement of Active/Passive Members in Structures Using Simulated Annealing," AIAA J., 29 (8), August 1991, pp. 1327–1334.

[59] Holland, J. H., Adaptation of Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, 1975.

[60] De Jong, K. A., Analysis of the Behavior of a Class of Genetic Adaptive Systems (Doctoral Dissertation, The University of Michigan; University Microfilms No. 76-9381), Dissertation Abstracts International, 36 (10), 5140B, 1975.

[61] Booker, L., "Improving Search in Genetic Algorithms," in Genetic Algorithms and Simulated Annealing, Ed. L. Davis, Morgan Kaufmann Publishers, Inc., Los Altos, CA. 1987, pp. 61–73.

[62] Goldberg, D. E., and Samtani, M. P., "Engineering Optimization via Genetic Algorithm," Proceedings of the Ninth Conference on Electronic Computation, ASCE, February 1986, pp. 471–482.

[63] Hajela, P., "Genetic Search—An Approach to the Nonconvex Optimization Problem," AIAA J., 28 (7), July 1990, pp. 1205–1210.

[64] Rao, S. S., Pan, T.-S., and Venkayya, V. B., "Optimal Placement of Actuators in Actively Controlled Structures Using Genetic Algorithms," AIAA J., 29 (6), pp. 942–943, June 1991.

[65] Szu, H., and Hartley, R.L., "Nonconvex Optimization by Fast Simulated Annealing," Proceedings of the IEEE, 75 (11), pp. 1538–1540, 1987.

Constrained Optimization 5

Most problems in structural optimization must be formulated as constrained minimization problems. In a typical structural design problem the objective function is a fairly simple function of the design variables (e.g., weight), but the design has to satisfy a host of stress, displacement, buckling, and frequency constraints. These constraints are usually complex functions of the design variables available only from an analysis of a finite element model of the structure. This chapter offers a review of methods that are commonly used to solve such constrained problems.

The methods described in this chapter are for use when the computational cost of evaluating the objective function and constraints is small or moderate. In these methods the objective function or constraints these are calculated exactly (e.g., by a finite element program) whenever they are required by the optimization algorithm. This approach can require hundreds of evaluations of objective function and constraints, and is not practical for problems where a single evaluation is computationally expensive. For these more expensive problems we go through an intermediate stage of constructing approximations for the objective function and constraints, or at least for the more expensive functions. The optimization is then performed on the approximate problem. This approximation process is described in the next chapter.

The basic problem that we consider in this chapter is the minimization of a function subject to equality and inequality constraints

$$\begin{array}{lll} \textbf{minimize} & f(\mathbf{x}) & \\ \textbf{such that} & h_i(\mathbf{x}) = 0, & i = 1, \ldots, n_e\,, \\ & g_j(\mathbf{x}) \geq 0, & j = 1, \ldots, n_g\,. \end{array} \tag{5.1}$$

The constraints divide the design space into two domains, the feasible domain where the constraints are satisfied, and the infeasible domain where at least one of the constraints is violated. In most practical problems the minimum is found on the boundary between the feasible and infeasible domains, that is at a point where $g_j(\mathbf{x}) = 0$ for at least one j. Otherwise, the inequality constraints may be removed without altering the solution. In most structural optimization problems the inequality constraints prescribe limits on sizes, stresses, displacements, etc. These limits have

great impact on the design, so that typically several of the inequality constraints are active at the minimum.

While the methods described in this section are powerful, they can often perform poorly when design variables and constraints are scaled improperly. To prevent ill-conditioning, all the design variables should have similar magnitudes, and all constraints should have similar values when they are at similar levels of criticality. A common practice is to normalize constraints such that $g(\mathbf{x}) = 0.1$ correspond to a ten percent margin in a response quantity. For example, if the constraint is an upper limit σ_a on a stress measure σ, then the constraint may be written as

$$g = 1 - \frac{\sigma}{\sigma_a} \geq 0 \ . \tag{5.2}$$

Some of the numerical techniques offered in this chapter for the solution of constrained nonlinear optimization problems are not able to handle equality constraints, but are limited to inequality constraints. In such instances it is possible to replace the equality constraint of the form $h_i(\mathbf{x}) = 0$ with two inequality constraints $h_i(\mathbf{x}) \leq 0$ and $h_i(\mathbf{x}) \geq 0$. However, it is usually undesirable to increase the number of constraints. For problems with large numbers of inequality constraints, it is possible to construct an equivalent constraint to replace them. One of the ways to replace a family of inequality constraints ($g_i(\mathbf{x}) \geq 0, i = 1 \ldots m$) by an equivalent constraint is to use the Kreisselmeier-Steinhauser function [1] (KS-function) defined as

$$KS[g_i(\mathbf{x})] = -\frac{1}{\rho} \ln[\sum_i e^{-\rho g_i(\mathbf{x})}] \ , \tag{5.3}$$

where ρ is a parameter which determines the closeness of the KS-function to the smallest inequality $min[g_i(\mathbf{x})]$. For any positive value of the ρ, the KS-function is always more negative than the most negative constraint, forming a lower bound envelope to the inequalities. As the value of ρ is increased the KS-functions conforms with the minimum value of the functions more closely. The value of the KS-function is always bounded by

$$g_{\min} \leq KS[g_i(\mathbf{x})] \leq g_{\min} - \frac{\ln(m)}{\rho} \ . \tag{5.4}$$

For an equality constraint represented by a pair of inequalities, $h_i(\mathbf{x}) \leq 0$ and $-h_i(\mathbf{x}) \leq 0$, the solution is at a point where both inequalities are active, $h_i(\mathbf{x}) = -h_i(\mathbf{x}) = 0$, Figure 5.1 . Sobieski [2] shows that for a KS-function defined by such a positive and negative pair of h_i, the gradient of the KS-function at the solution point $h_i(\mathbf{x}) = 0$ vanishes regardless of the ρ value, and its value approaches to zero as the value of ρ tends to infinity, Figure 5.1 . Indeed, from Eq. (5.4) at $\mathbf{x}$ where $h_i = 0$, the KS-function has the property

$$0 \geq KS(h, -h) \geq -\frac{\ln(2)}{\rho} \ . \tag{5.5}$$

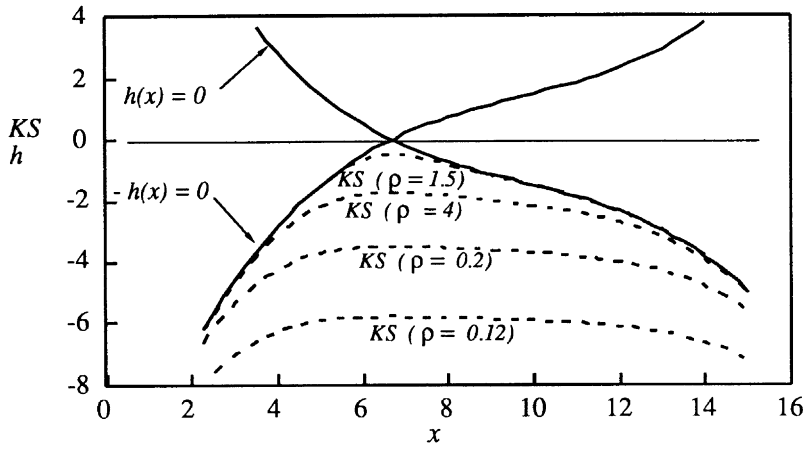


Figure 5.1 Kreisselmeier-Steinhauser function for replacing $h(\mathbf{x}) = 0$.

Consequently, an optimization problem

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad h_k(\mathbf{x}) = 0, \qquad k = 1, \ldots, n_e \,, \end{aligned} \tag{5.6}$$

may be reformulated as

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad KS(h_1, -h_1, h_2, -h_2, \ldots, h_{n_e}, -h_{n_e}) \geq -\epsilon \,. \end{aligned} \tag{5.7}$$

where ϵ is a small tolerance.

5.1 The Kuhn-Tucker Conditions

5.1.1 *General Case*

In general, problem (5.1) may have several local minima. Only under special circumstances are sure of the existence of single global minimum. The necessary conditions for a minimum of the constrained problem are obtained by using the Lagrange multiplier method. We start by considering the special case of equality constraints only. Using the Lagrange multiplier technique, we define the Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{j=1}^{n_e} \lambda_j h_j(\mathbf{x}) \,, \tag{5.1.1}$$

where λ_j are unknown Lagrange multipliers. The necessary conditions for a stationary point are

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} \lambda_j \frac{\partial h_j}{\partial x_i} = 0, \qquad i = 1, \ldots, n\,, \tag{5.1.2}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = h_j(\mathbf{x}) = 0, \qquad j = 1, \ldots, n_e\,. \tag{5.1.3}$$

These conditions, however, apply only at a *regular point*, that is at a point where the gradients of the constraints are linearly independent. If we have constraint gradients that are linearly dependent, it means that we can remove some constraints without affecting the solution. At a regular point, Eqs. (5.1.2) and (5.1.3) represent $n + n_e$ equations for the n_e Lagrange multipliers and the n coordinates of the stationary point.

The situation is somewhat more complicated when inequality constraints are present. To be able to apply the Lagrange multiplier method we first transform the inequality constraints to equality constraints by adding slack variables. That is, the inequality constraints are written as

$$g_j(\mathbf{x}) - t_j^2 = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.1.4}$$

where t_j is a slack variable which measures how far the jth constraint is from being critical. We can now form a Lagrangian function

$$\mathcal{L}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) = f - \sum_{j=1}^{n_g} \lambda_j (g_j - t_j^2)\,. \tag{5.1.5}$$

Differentiating the Lagrangian function with respect to $\mathbf{x}$, $\boldsymbol{\lambda}$ and $\mathbf{t}$ we obtain

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1, \ldots, n\,, \tag{5.1.6}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = -g_j + t_j^2 = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.1.7}$$

$$\frac{\partial \mathcal{L}}{\partial t_j} = 2\lambda_j t_j = 0, \qquad j = 1, \ldots, n_g\,. \tag{5.1.8}$$

Equations (5.1.7) and (5.1.8) imply that when an inequality constraint is not critical (so that the corresponding slack variable is non-zero) then the Lagrange multiplier associated with the constraint is zero. Equations (5.1.6) to (5.1.8) are the necessary conditions for a stationary regular point. Note that for inequality constraints a regular point is one where the gradients of the *active* constraints are linearly independent. These conditions are modified slightly to yield the necessary conditions for a minimum and are known as the Kuhn-Tucker conditions. The Kuhn-Tucker conditions may be summarized as follows:

A point x is a local minimum of an inequality constrained problem only if a set of nonnegative λ_j's may be found such that:

1. Equation (5.1.6) is satisfied
2. The corresponding λ_j is zero if a constraint is not active.

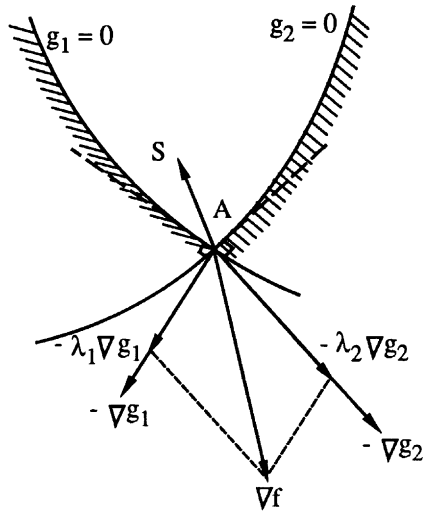


Figure 5.1.1 A geometrical interpretation of Kuhn-Tucker condition for the case of two constraints.

A geometrical interpretation of the Kuhn-Tucker conditions is illustrated in Fig. (5.1.1) for the case of two constraints. ∇g_1 and ∇g_2 denote the gradients of the two constraints which are orthogonal to the respective constraint surfaces. The vector s shows a typical feasible direction which does not lead immediately to any constraint violation. For the two-constraint case Eq. (5.1.6) may be written as

$$-\nabla f = -(\lambda_1 \nabla g_1 + \lambda_2 \nabla g_2) \; . \tag{5.1.9}$$

Assume that we want to determine whether point A is a minimum or not. To improve the design we need to proceed from point A in a direction s that is usable and feasible. For the direction to be usable, a small move along this direction should decrease the objective function. To be feasible, s should form an obtuse angle with $-\nabla g_1$ and $-\nabla g_2$. To be a direction of decreasing f it must form an acute angle with $-\nabla f$. Clearly from Figure (5.1.1), any vector which forms an acute angle with $-\nabla f$ will also form and acute angle with either $-\nabla g_1$ or $-\nabla g_2$. Thus the Kuhn-Tucker conditions mean that no feasible design with reduced objective function is to be found in the neighborhood of A. Mathematically, the condition that a direction s be feasible is written as

$$\mathbf{s}^T \nabla g_j \geq 0, \qquad j \in I_A \, , \tag{5.1.10}$$

where I_A is the set of active constraints Equality in Eq. (5.1.10) is permitted only for linear or concave constraints (see Section 5.1.2 for definition of concavity). The condition for a usable direction (one that decreases the objective function) is

$$\mathbf{s}^T \nabla f < 0 \ . \tag{5.1.11}$$

Multiplying Eq. (5.1.6) by s_i and summing over i we obtain

$$\mathbf{s}^T \nabla f = \sum_{j=1}^{n_g} \lambda_j \mathbf{s}^T \nabla g_j \ . \tag{5.1.12}$$

In view of Eqs. (5.1.10) and (5.1.11), Eq. (5.1.12) is impossible if the λ_j's are positive.

If the Kuhn-Tucker conditions are satisfied at a point it is impossible to find a direction with a negative slope for the objective function that does not violate the constraints. In some cases, though, it is possible to move in a direction which is tangent to the active constraints and perpendicular to the gradient (that is, has zero slope), that is

$$\mathbf{s}^T \nabla f = \mathbf{s}^T \nabla g_j = 0, \qquad j \in I_A \ . \tag{5.1.13}$$

The effect of such a move on the objective function and constraints can be determined only from higher derivatives. In some cases a move in this direction could reduce the objective function without violating the constraints even though the Kuhn-Tucker conditions are met. Therefore, the Kuhn-Tucker conditions are necessary but not sufficient for optimality.

The Kuhn-Tucker conditions are sufficient when the number of active constraints is equal to the number of design variables. In this case Eq. (5.1.13) cannot be satisfied with $\mathbf{s} \neq 0$ because ∇g_j includes n linearly independent directions (in n dimensional space a vector cannot be orthogonal to n linearly independent vectors).

When the number of active constraints is not equal to the number of design variables sufficient conditions for optimality require the second derivatives of the objective function and constraints. A sufficient condition for optimality is that the Hessian matrix of the Lagrangian function is positive definite in the subspace tangent to the active constraints. If we take, for example, the case of equality constraints, the Hessian matrix of the Lagrangian is

$$\nabla^2 \mathcal{L} = \nabla^2 f - \sum_{j=1}^{n_e} \lambda_j \nabla^2 h_j \ . \tag{5.1.14}$$

The sufficient condition for optimality is that

$$\mathbf{s}^T (\nabla^2 \mathcal{L}) \mathbf{s} > 0, \qquad \text{for all } \mathbf{s} \text{ for which } \mathbf{s}^T h_j = 0, \quad j = 1 \ldots, n_e \ . \tag{5.1.15}$$

When inequality constraints are present, the vector $\mathbf{s}$ also needs to be orthogonal to the active constraints with positive Lagrange multipliers. For active constraints with zero Lagrange multipliers, $\mathbf{s}$ must satisfy

$$\mathbf{s}^T \nabla g_j \geq 0, \qquad \text{when } g_j = 0 \text{ and } \lambda_j = 0 \ . \tag{5.1.16}$$

Example 5.1.1

Find the minimum of

$$f = -x_1^3 - 2x_2^2 + 10x_1 - 6 - 2x_2^3 ,$$

subject to

$$\begin{aligned} g_1 &= 10 - x_1 x_2 \geq 0, \\ g_2 &= x_1 \geq 0, \\ g_3 &= 10 - x_2 \geq 0 . \end{aligned}$$

The Kuhn-Tucker conditions are

$$\begin{aligned} -3x_1^2 + 10 + \lambda_1 x_2 - \lambda_2 &= 0, \\ -4x_2 - 6x_2^2 + \lambda_1 x_1 + \lambda_3 &= 0 . \end{aligned}$$

We have to check for all possibilities of active constraints.

The simplest case is when no constraints are active, $\lambda_1 = \lambda_2 = \lambda_3 = 0$. We get $x_1 = 1.826$, $x_2 = 0$, $f = 6.17$. The Hessian matrix of the Lagrangian,

$$\nabla^2 \mathcal{L} = \begin{bmatrix} -6x_1 & \lambda_1 \\ \lambda_1 & -4 - 12x_2 \end{bmatrix} ,$$

is clearly negative definite, so that this point is a maximum. We next assume that the first constraint is active, $x_1 x_2 = 10$, so that $x_1 \neq 0$ and g_2 is inactive and therefore $\lambda_2 = 0$. We have two possibilities for the third constraint. If it is active we get $x_1 = 1$, $x_2 = 10$, $\lambda_1 = -0.7$, and $\lambda_3 = 639.3$, so that this point is neither a minimum nor a maximum. If the third constraint is not active $\lambda_3 = 0$ and we obtain the following three equations

$$\begin{aligned} -3x_1^2 + 10 + \lambda_1 x_2 &= 0, \\ -4x_2 - 6x_2^2 + \lambda_1 x_1 &= 0, \\ x_1 x_2 &= 10 . \end{aligned}$$

The only solution for these equations that satisfies the constraints on x_1 and x_2 is

$$x_1 = 3.847, \qquad x_2 = 2.599, \qquad \lambda_1 = 13.24, \qquad f = -73.08 .$$

This point satisfies the Kuhn-Tucker conditions for a minimum. However, the Hessian of the Lagrangian at that point

$$\nabla^2 \mathcal{L} = \begin{bmatrix} -23.08 & 13.24 \\ 13.24 & -35.19 \end{bmatrix} ,$$

is negative definite, so that it cannot satisfy the sufficiency condition. In fact, an examination of the function f at neighboring points along $x_1 x_2 = 10$ reveals that the point is not a minimum.

Next we consider the possibility that g_1 is not active, so that $\lambda_1 = 0$, and

$$-3x_1^2 + 10 - \lambda_2 = 0,$$
$$-4x_2 - 6x_2^2 + \lambda_3 = 0 .$$

We have already considered the possibility of both λ's being zero, so we need to consider only three possibilities of one of these Lagrange multipliers being nonzero, or both being nonzero. The first case is $\lambda_2 \neq 0$, $\lambda_3 = 0$, then $g_2 = 0$ and we get $x_1 = 0$, $x_2 = 0$, $\lambda_2 = 10$, and $f = -6$, or $x_1 = 0$, $x_2 = -2/3$, $\lambda_2 = 10$, and $f = -6.99$. Both points satisfy the Kuhn-Tucker conditions for a minimum, but not the sufficiency condition. In fact, the vectors tangent to the active constraints ($x_1 = 0$ is the only one) have the form $\mathbf{s}^T = (0, a)$, and it is easy to check that $\mathbf{s}^T \nabla^2 \mathcal{L} \mathbf{s} < 0$. It is also easy to check that these points are indeed no minima by reducing x_2 slightly.

The next case is $\lambda_2 = 0$, $\lambda_3 \neq 0$, so that $g_3 = 0$. We get $x_1 = 1.826$, $x_2 = 10$, $\lambda_3 = 640$ and $f = -2194$. this point satisfies the Kuhn-Tucker conditions, but it is not a minimum either. It is easy to check that $\nabla^2 \mathcal{L}$ is negative definite in this case so that the sufficiency condition could not be satisfied. Finally, we consider the case $x_1 = 0$, $x_2 = 10$, $\lambda_2 = 10$, $\lambda_3 = 640$, $f = -2206$. Now the Kuhn-Tucker conditions are satisfied, and the number of active constraints is equal to the number of design variables, so that this point is a minimum.• • •

5.1.2 *Convex Problems*

There is a class of problems, namely convex problems, for which the Kuhn-Tucker conditions are not only necessary but also sufficient for a global minimum. To define convex problems we need the notions of convexity for a set of points and for a function. A set of points S is convex whenever the entire line segment connecting two points that are in S is also in S. That is

$$\text{if } \mathbf{x}_1, \mathbf{x}_2 \in S, \quad \text{then } \alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in S, \qquad 0 < \alpha < 1 . \tag{5.1.17}$$

A function is convex if

$$f[\alpha \mathbf{x}_2 + (1-\alpha)\mathbf{x}_1] \leq \alpha f(\mathbf{x}_2) + (1-\alpha) f(\mathbf{x}_1), \qquad 0 < \alpha < 1 . \tag{5.1.18}$$

This is shown pictorially for a function of a single variable in Figure (5.1.2). The straight segment connecting any two points on the curve must lie above the curve. Alternatively we note that the second derivative of f is non-negative $f''(x) \geq 0$. It can be shown that a function of n variables is convex if its matrix of second derivatives is positive semi-definite.

A convex optimization problem has a convex objective function and a convex feasible domain. It can be shown that the feasible domain is convex if all the inequality constraints g_j are *concave* (that is, $-g_j$ are convex) and the equality constraints are linear. A convex optimization problem has only one minimum, and the Kuhn-Tucker conditions are sufficient to establish it. Most optimization problems encountered in practice cannot be shown to be convex. However, the theory of convex programming is still very important in structural optimization, as we often approximate optimization problems by a series of convex approximations (see Chapter 9). The simplest such approximation is a linear approximation for the objective function and constraints– this produces a linear programming problem.

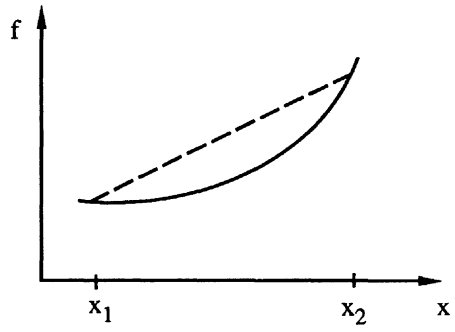


Figure 5.1.2 Convex function.

Example 5.1.2

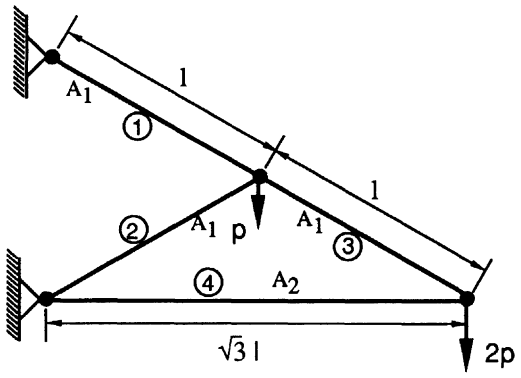


Figure 5.1.3 Four bar statically determinate truss.

Consider the minimum weight design of the four bar truss shown in Figure (5.1.3). For the sake of simplicity we assume that members 1 through 3 have the same area A_1 and member 4 has an area A_2. The constraints are limits on the stresses in the members and on the vertical displacement at the right end of the truss. Under the specified loading the member forces and the vertical displacement δ at the end are found to be

$$f_1 = 5p, \quad f_2 = -p, \quad f_3 = 4p, \quad f_4 = -2\sqrt{3}p,$$
$$\delta = \frac{6pl}{E}\left(\frac{3}{A_1} + \frac{\sqrt{3}}{A_2}\right).$$

We assume the allowable stresses in tension and compression to be $8.74 \times 10^{-4}E$ and $4.83 \times 10^{-4}E$, respectively, and limit the vertical displacement to be no greater than $3 \times 10^{-3}l$. The minimum weight design subject to stress and displacement constraints

can be formulated in terms of nondimensional design variables

$$x_1 = 10^{-3}\frac{A_1 E}{p}, \quad x_2 = 10^{-3}\frac{A_2 E}{p},$$

as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0, \\ & g_2 = x_1 - 5.73 \geq 0, \\ & g_3 = x_2 - 7.17 \geq 0 \,. \end{aligned}$$

The Kuhn-Tucker conditions are

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{3} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1, 2,$$

or

$$3 - \frac{18}{x_1^2}\lambda_1 - \lambda_2 = 0\,,$$

$$\sqrt{3} - \frac{6\sqrt{3}}{x_2^2}\lambda_1 - \lambda_3 = 0\,.$$

Consider first the possibility that $\lambda_1 = 0$. Then clearly $\lambda_2 = 3$, $\lambda_3 = \sqrt{3}$ so that $g_2 = 0$ and $g_3 = 0$, and then $x_1 = 5.73$, $x_2 = 7.17$, $g_1 = -1.59$, so that this solution is not feasible. We conclude that $\lambda_1 \neq 0$, and the first constraint must be active at the minimum. Consider now the possibility that $\lambda_2 = \lambda_3 = 0$. We have the two Kuhn-Tucker equations and the equation $g_1 = 0$ for the unknowns λ_1, x_1, x_2. The solution is

$$x_1 = x_2 = 9.464, \quad \lambda_1 = 14.93, \quad f = 44.78\,.$$

The Kuhn-Tucker conditions for a minimum are satisfied. If the problem is convex the Kuhn-Tucker conditions are sufficient to guarantee that this point is the global minimum. The objective function and the constraint functions g_2 and g_3 are linear, so that we need to check only g_1. For convexity g_1 has to be concave or $-g_1$ convex; this holds if the second derivative matrix $-\mathbf{A}_1$ of $-g_1$ is positive semi-definite

$$-\mathbf{A}_1 = \begin{bmatrix} 36/x_1^3 & 0 \\ 0 & 12\sqrt{3}x_2^3 \end{bmatrix}\,.$$

Clearly, for $x_1 > 0$ and $x_2 > 0$, $-\mathbf{A}_1$ is positive definite so that the minimum that we found is a global minimum.• • •

5.2 Quadratic Programming Problems

One of the simplest form of nonlinear constrained optimization problems is in the form of *Quadratic Programming* (QP) problem. A general QP problem has a quadratic objective function with linear equality and inequality constraints. For the sake of simplicity we consider only an inequality problem with n_g constraints stated as

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \textbf{such that} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}\,, \\ & x_i \geq 0, \qquad i = 1, \ldots, n\,. \end{aligned} \tag{5.2.1}$$

The linear constraints form a convex feasible domain. If the objective function is also convex, then we have a convex optimization problem in which, as discussed in the previous section, the Kuhn-Tucker conditions become sufficient for the optimality of the problem. Hence, having a positive semi-definite or positive definite $\mathbf{Q}$ matrix assures a global minimum for the solution of the problem, if one exists. For many optimization problems the quadratic form $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is either positive definite or positive semi-definite. Therefore, one of the methods for solving QP problems relies on solving the Kuhn-Tucker conditions.

We start by writing the Lagrange function for the Problem (5.2.1)

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{t}, \mathbf{s}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \{t_j^2\} - \mathbf{b}) - \boldsymbol{\mu}^T(\mathbf{x} - \{s_i^2\})\,, \tag{5.2.2}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the vectors of Lagrange multipliers for the inequality constraints and the nonnegativity constraints, respectively, and $\{t_j^2\}$ and $\{s_i^2\}$ are the vectors of positive slack variables for the same. The necessary conditions for a stationary point are obtained by differentiating the Lagrangian with respect to the $\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{t}$, and $\mathbf{s}$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{c} - \mathbf{Q}\mathbf{x} - \mathbf{A}^T\boldsymbol{\lambda} - \boldsymbol{\mu} = 0\,, \tag{5.2.3}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{x} - \{t_j^2\} - \mathbf{b} = 0\,, \tag{5.2.4}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \mathbf{x} - \{s_i^2\} = 0\,, \tag{5.2.5}$$

$$\frac{\partial \mathcal{L}}{\partial t_j} = 2\lambda_j t_j = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.2.6}$$

$$\frac{\partial \mathcal{L}}{\partial s_i} = 2\mu_i s_i = 0, \qquad i = 1, \ldots, n\,. \tag{5.2.7}$$

where n_g is the number of inequality constraints, and n is the number of design variables. We define a new vector $\{q_j\} = \{t_j^2\}, \quad j = 1, \ldots, n_g \quad (\mathbf{q} \geq 0)$. After multiplying Eqs. (5.2.6) and (5.2.7) by $\{t_j\}$ and $\{s_i\}$, respectively, and eliminating

$\{s_i\}$ from the last equation by using Eq. (5.2.5), we can rewrite the Kuhn-Tucker conditions

$$\mathbf{Qx} + \mathbf{A}^T\boldsymbol{\lambda} + \boldsymbol{\mu} = \mathbf{c}\,, \tag{5.2.8}$$

$$\mathbf{Ax} - \mathbf{q} = \mathbf{b}\,, \tag{5.2.9}$$

$$\lambda_j q_j = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.2.10}$$

$$\mu_i x_i = 0, \qquad i = 1, \ldots, n\,, \tag{5.2.11}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \text{and} \quad \boldsymbol{\mu} \geq \mathbf{0}\,. \tag{5.2.12}$$

Equations (5.2.8) and (5.2.9) form a set of $n + n_g$ linear equations for the solution of unknowns x_i, λ_j, μ_i, and q_j which also need to satisfy Eqs. (5.2.10) and (5.2.11). Despite the nonlinearity of the Eqs. (5.2.10) and (5.2.11), this problem can be solved as proposed by Wolfe [3] by using the procedure described in 3.6.3 for generating a basic feasible solution through the use of artificial variables. Introducing a set of artificial variables, y_i, $i = 1, \ldots, n$, we define an artificial cost function to be minimized,

$$\textbf{minimize} \quad \sum_{i=1}^{n} y_i \tag{5.2.13}$$

$$\textbf{subject to} \quad \mathbf{Qx} + \mathbf{A}^T\boldsymbol{\lambda} + \boldsymbol{\mu} + \mathbf{y} = \mathbf{c}\,, \tag{5.2.14}$$

$$\mathbf{Ax} - \mathbf{q} = \mathbf{b}\,, \tag{5.2.15}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{\mu} \geq \mathbf{0}, \quad \text{and} \quad \mathbf{y} \geq \mathbf{0}\,. \tag{5.2.16}$$

Equations (5.2.13) through (5.2.16) can be solved by using the standard simplex method with the additional requirement that (5.2.10) and (5.2.11) be satisfied. These requirements can be implemented during the simplex algorithm by simply enforcing that the variables λ_j and q_j (and μ_i and x_i) not be included in the basic solution simultaneously. That is, we restrict a non-basic variable μ_i from entering the basis if the corresponding x_i is already among the basic variables.

Other methods for solving the quadratic programming problem are also available, and the reader is referred to Gill et al. ([4], pp. 177–180) for additional details.

5.3 Computing the Lagrange Multipliers

As may be seen from example 5.1.1, trying to find the minimum directly from the Kuhn-Tucker conditions may be difficult because we need to consider many combinations of active and inactive constraints, and this would in general involve the solution of highly nonlinear equations. The Kuhn-Tucker conditions are, however, often used to check whether a candidate minimum point satisfies the necessary conditions. In such a case we need to calculate the Lagrange multipliers (also called the Kuhn-Tucker multipliers) at a given point $\mathbf{x}$. As we will see in the next section, we

may also want to calculate the Lagrange multipliers for the purpose of estimating the sensitivity of the optimum solution to small changes in the problem definition. To calculate the Lagrange multipliers we start by writing Eq. (5.1.6) in matrix notation as

$$\nabla f - \mathbf{N}\boldsymbol{\lambda} = 0\,, \tag{5.3.1}$$

where the matrix $\mathbf{N}$ is defined by

$$n_{ij} = \frac{\partial g_j}{\partial x_i}, \qquad j = 1,\ldots,r\,, \text{ and } \qquad i = 1,\ldots,n\,. \tag{5.3.2}$$

We consider only the active constraints and associated lagrange multipliers, and assume that there are r of them.

Typically, the number, r, of active constraints is less than n, so that with n equations in terms of r unknowns, Eq. (5.3.1) is an overdetermined system. We assume that the gradients of the constraints are linearly independent so that $\mathbf{N}$ has rank r. If the Kuhn-Tucker conditions are satisfied the equations are consistent and we have an exact solution. We could therefore use a subset of r equations to solve for the Lagrange multipliers. However, this approach may be susceptible to amplification of errors. Instead we can use a least-squares approach to solve the equations. We define a residual vector $\mathbf{u}$

$$\mathbf{u} = \mathbf{N}\boldsymbol{\lambda} - \nabla f\,, \tag{5.3.3}$$

A least squares solution of Eq. (5.3.1) will minimize the square of the Euclidean norm of the residual with respect to $\boldsymbol{\lambda}$

$$\|\mathbf{u}\|^2 = (\mathbf{N}\boldsymbol{\lambda} - \nabla f)^T(\mathbf{N}\boldsymbol{\lambda} - \nabla f) = \boldsymbol{\lambda}^T\mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T\mathbf{N}^T\nabla f + \nabla f^T\nabla f\,. \tag{5.3.4}$$

To minimize $\|\mathbf{u}\|^2$ we differentiate it with respect to each one of the Lagrange multipliers and get

$$2\mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} - 2\mathbf{N}^T\nabla f = 0\,, \tag{5.3.5}$$

or

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f\,. \tag{5.3.6}$$

This is the best solution in the least square sense. However, if the Kuhn-Tucker conditions are satisfied it should be the exact solution of Eq. (5.3.1). Substituting from Eq. (5.3.6) into Eq. (5.3.1) we obtain

$$\mathbf{P}\nabla f = 0\,, \tag{5.3.7}$$

where

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\,. \tag{5.3.8}$$

$\mathbf{P}$ is called the projection matrix. It will be shown in Section 5.5 that it projects a vector into the subspace tangent to the active constraints. Equation (5.3.7) implies that for the Kuhn-Tucker conditions to be satisfied the gradient of the objective function has to be orthogonal to that subspace.

In practice Eq. (5.3.6) is no longer popular for the calculation of the Lagrange multipliers. One reason is that the method is ill-conditioned and another is that it is

not efficient. An efficient and better conditioned method for least squares calculations is based on the QR factorization of the matrix $\mathbf{N}$. The QR factorization of the matrix $\mathbf{N}$ consists of an $r \times r$ upper triangular matrix $\mathbf{R}$ and an $n \times n$ orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{QN} = \begin{pmatrix} \mathbf{Q}_1\mathbf{N} \\ \mathbf{Q}_2\mathbf{N} \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} . \tag{5.3.9}$$

Here $\mathbf{Q}_1$ is a matrix consisting of the first r rows of $\mathbf{Q}$, $\mathbf{Q}_2$ includes the last $n - r$ rows of $\mathbf{Q}$, and the zero represents an $(n - r) \times r$ zero matrix (for details of the QR factorization see most texts on numerical analysis, e.g., Dahlquist and Bjorck [5]). Because $\mathbf{Q}$ is an orthogonal matrix, the Euclidean norm of $\mathbf{Qu}$ is the same as that of $\mathbf{u}$, or

$$\|\mathbf{u}\|^2 = \|\mathbf{Qu}\|^2 = \|\mathbf{QN}\boldsymbol{\lambda} - \mathbf{Q}\nabla f\|^2 = \left\| \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} \boldsymbol{\lambda} - \mathbf{Q}\nabla f \right\|^2 = \left\| \begin{pmatrix} \mathbf{R}\boldsymbol{\lambda} - \mathbf{Q}_1\nabla f \\ -\mathbf{Q}_2\nabla f \end{pmatrix} \right\|^2 . \tag{5.3.10}$$

From this form it can be seen that $\|\mathbf{u}\|^2$ is minimized by choosing $\boldsymbol{\lambda}$ so that

$$\mathbf{R}\boldsymbol{\lambda} = \mathbf{Q}_1\nabla f . \tag{5.3.11}$$

The last $n - r$ rows of the matrix $\mathbf{Q}$ denoted $\mathbf{Q}_2$ are also important in the following. They are orthogonal vectors which span the null space of $\mathbf{N}^T$. That is $\mathbf{N}^T$ times each one of these vectors is zero.

Example 5.3.1

Check whether the point $(-2, -2, 4)$ is a local minimum of the problem

$$\begin{aligned} f &= x_1 + x_2 + x_3, \\ g_1 &= 8 - x_1^2 - x_2^2 \geq 0, \\ g_2 &= x_3 - 4 \geq 0, \\ g_3 &= x_2 + 8 \geq 0 . \end{aligned}$$

Only the first two constraints are critical at $(-2, -2, 4)$

$$\frac{\partial g_1}{\partial x_1} = -2x_1 = 4, \quad \frac{\partial g_1}{\partial x_2} = -2x_2 = 4, \quad \frac{\partial g_1}{\partial x_3} = 0 ,$$

$$\frac{\partial g_2}{\partial x_1} = 0, \quad \frac{\partial g_2}{\partial x_2} = 0, \quad \frac{\partial g_2}{\partial x_3} = 1 ,$$

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial x_3} = 1 .$$

So

$$\mathbf{N} = \begin{bmatrix} 4 & 0 \\ 4 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \nabla f = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} ,$$

$$\mathbf{N}^T\mathbf{N} = \begin{bmatrix} 32 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{N}^T\nabla f = \begin{Bmatrix} 8 \\ 1 \end{Bmatrix},$$

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f = \begin{Bmatrix} 1/4 \\ 1 \end{Bmatrix},$$

also

$$\left[\mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right]\nabla f = 0\,.$$

Equation (5.3.7) is satisfied, and all the Lagrange multipliers are positive, so the Kuhn-Tucker conditions for a minimum are satisfied. • • •

5.4 Sensitivity of Optimum Solution to Problem Parameters

The Lagrange multipliers are not only useful for checking optimality, but they also provide information about the sensitivity of the optimal solution to problem parameters. In this role they are extremely valuable in practical applications. In most engineering design optimization problems we have a host of parameters such as material properties, dimensions and load levels that are fixed during the optimization. We often need the sensitivity of the optimum solution to these problem parameters, either because we do not know them accurately, or because we have some freedom to change them if we find that they have a large effect on the optimum design.

We assume now that the objective function and constraints depend on a parameter p so that the optimization problem is defined as

$$\begin{array}{ll} \textbf{minimize} & f(\mathbf{x}, p) \\ \textbf{such that} & g_j(\mathbf{x}, p) \geq 0, \qquad j = 1, \ldots, n_g\,. \end{array} \tag{5.4.1}$$

The solution of the problem is denoted $\mathbf{x}^*(p)$ and the corresponding objective function $f^*(p) = f(\mathbf{x}^*(p), p)$. We want to find the derivatives of $\mathbf{x}^*$ and f^* with respect to p. The equations that govern the optimum solution are the Kuhn-Tucker conditions, Eq. (5.3.1), and the set of active constraints

$$\mathbf{g}_a = 0. \tag{5.4.2}$$

where $\mathbf{g}_a$ denotes the vector of r active constraint functions. Equations (5.3.1) and (5.4.2) are satisfied by $\mathbf{x}^*(p)$ for all values of p that do not change the set of active constraints. Therefore, the derivatives of these equations with respect to p are zero, provided we consider the implicit dependence of $\mathbf{x}$ and $\boldsymbol{\lambda}$ on p. Differentiating Eq. (5.3.1) and (5.4.2) with respect to p we obtain

$$(\mathbf{A} - \mathbf{Z})\frac{d\mathbf{x}^*}{dp} - \mathbf{N}\frac{d\boldsymbol{\lambda}}{dp} + \frac{\partial}{\partial p}(\nabla f) - \left(\frac{\partial \mathbf{N}}{\partial p}\right)\boldsymbol{\lambda} = 0\,, \tag{5.4.3}$$

$$\mathbf{N}^T\frac{d\mathbf{x}^*}{dp} + \frac{\partial \mathbf{g}_a}{\partial p} = 0\,, \tag{5.4.4}$$

where $\mathbf{A}$ is the Hessian matrix of the objective function f, $a_{ij} = \partial^2 f / \partial x_i \partial x_j$, and $\mathbf{Z}$ is a matrix whose elements are

$$z_{kl} = \sum_j \frac{\partial^2 g_j}{\partial x_x \partial x_l} \lambda_j \ . \tag{5.4.5}$$

Equations (5.4.3) and (5.4.4) are a system of simultaneous equations for the derivatives of the design variables and of the Lagrange multipliers. Different special cases of this system are discussed by Sobieski et al. [6].

Often we do not need the derivatives of the design variables or of the Lagrange multipliers, but only the derivatives of the objective function. In this case the sensitivity analysis can be greatly simplified. We can write

$$\frac{df}{dp} = \frac{\partial f}{\partial p} + \sum_{l=1}^{n} \frac{\partial f}{\partial x_l} \frac{dx_l^*}{dp} = \frac{\partial f}{\partial p} + (\nabla f)^T \frac{d\mathbf{x}^*}{dp} \ . \tag{5.4.6}$$

Using Eq. (5.3.1) and (5.4.4) we get

$$\frac{df}{dp} = \frac{\partial f}{\partial p} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}_a}{\partial p} \ . \tag{5.4.7}$$

Equation (5.4.7) shows that the Lagrange multipliers are a measure of the effect of a change in the constraints on the objective function. Consider, for example, a constraint of the form $g_j(\mathbf{x}) = G_j(\mathbf{x}) - p \geq 0$. By increasing p we make the constraint more difficult to satisfy. Assume that many constraints are critical, but that p affects only this single constraint. We see that $\partial g_j / \partial p = -1$, and from Eq. (5.4.7) $df/dp = \lambda_j$, that is λ_j is the 'marginal price' that we pay in terms of an increase in the objective function for making g_j more difficult to satisfy.

The interpretation of Lagrange multipliers as the marginal prices of the constraints also explains why at the optimum all the Lagrange multipliers have to be non-negative. A negative Lagrange multiplier would indicate that we can reduce the objective function by making a constraint more difficult to satisfy— an absurdity.

Example 5.4.1

Consider the optimization problem

$$\begin{aligned} f &= x_1 + x_2 + x_3, \\ g_1 &= p - x_1^2 - x_2^2 \geq 0, \\ g_2 &= x_3 - 4 \geq 0, \\ g_3 &= x_2 + p \geq 0 \ . \end{aligned}$$

This problem was analyzed for $p = 8$ in Example 5.3.1, and the optimal solution was found to be $(-2, -2, 4)$. We want to find the derivative of this optimal solution with respect to p. At the optimal point we have $f = 0$ and $\boldsymbol{\lambda}^T = (0.25, 1.0)$, with the

first two constraints being critical. We can calculate the derivative of the objective function from Eq. (5.4.7)

$$\frac{\partial f}{\partial p} = 0, \qquad \frac{\partial \mathbf{g}_a}{\partial p} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix},$$

so

$$\frac{df}{dp} = -0.25 .$$

To calculate the derivatives of the design variables and constraints we need to set up Eqs. (5.4.3) and (5.4.4). We get

$$\mathbf{A} = \mathbf{0}, \qquad \frac{\partial \nabla f}{\partial p} = \mathbf{0}, \qquad \frac{\partial \mathbf{N}}{\partial p} = \mathbf{0} .$$

Only g_1 has nonzero second derivatives $\partial^2 g_1/\partial x_1^2 = \partial^2 g_1/\partial x_2^2 = -2$ so from Eq. (5.4.5)

$$z_{11} = -2\lambda_2 = -2, \qquad z_{22} = -2\lambda_2 = -2, \qquad \mathbf{Z} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

With $\mathbf{N}$ from Example 5.3.1, Eq. (5.4.3) gives us

$$\begin{aligned} 2\dot{x}_1 - 4\dot{\lambda}_1 &= 0 , \\ 2\dot{x}_2 - 4\dot{\lambda}_1 &= 0 , \\ \dot{\lambda}_2 &= 0 , \end{aligned}$$

where a dot denotes derivative with respect to p. From Eq. (5.4.4) we get

$$\begin{aligned} 4\dot{x}_1 + 4\dot{x}_2 + 1 &= 0 , \\ \dot{x}_3 &= 0 . \end{aligned}$$

The solution of these five coupled equations is

$$\dot{x}_1 = \dot{x}_2 = -0.125, \quad \dot{x}_3 = 0, \quad \dot{\lambda}_1 = -0.0625, \quad \dot{\lambda}_2 = 0 .$$

We can check the derivatives of the objective function and design variables by changing p from 8 to 9 and re-optimizing. It is easy to check that we get $x_1 = x_2 = -2.121$, $x_3 = 4$, $f = -0.242$. These values compare well with linear extrapolation based on the derivatives which gives $x_1 = x_2 = -2.125$, $x_3 = 4$, $f = -0.25$.• • •

5.5 Gradient Projecti... and Reduced Gradient Methods

Rosen's gradient projection method is based on projecting the search direction into the subspace tangent to the active constraints. Let us first examine the method for the case of linear constraints [7]. We define the constrained problem as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && g_j(\mathbf{x}) = \sum_{i=1}^{n} a_{ji}x_i - b_j \geq 0, \qquad j = 1, \ldots, n_g \ . \end{aligned} \tag{5.5.1}$$

In vector form

$$g_j = \mathbf{a}_j^T\mathbf{x} - b_j \geq 0 \ . \tag{5.5.2}$$

If we select only the r active constraints ($j \in I_A$), we may write the constraint equations as

$$\mathbf{g}_a = \mathbf{N}^T\mathbf{x} - \mathbf{b} = 0 \, , \tag{5.5.3}$$

where $\mathbf{g}_a$ is the vector of active constraints and the columns of the matrix $\mathbf{N}$ are the gradients of these constraints. The basic assumption of the gradient projection method is that $\mathbf{x}$ lies in the subspace tangent to the active constraints. If

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\mathbf{s} \, , \tag{5.5.4}$$

and both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ satisfy Eq. (5.5.3), then

$$\mathbf{N}^T\mathbf{s} = 0 \ . \tag{5.5.5}$$

If we want the steepest descent direction satisfying Eq. (5.5.5), we can pose the problem as

$$\begin{aligned} &\textbf{minimize} && \mathbf{s}^T\nabla f \\ &\textbf{such that} && \mathbf{N}^T\mathbf{s} = 0 \, , \\ &\textbf{and} && \mathbf{s}^T\mathbf{s} = 1 \ . \end{aligned} \tag{5.5.6}$$

That is, we want to find the direction with the most negative directional derivative which satisfies Eq. (5.5.5). We use Lagrange multipliers $\boldsymbol{\lambda}$ and μ to form the Lagrangian

$$\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \mu) = \mathbf{s}^T\nabla f - \mathbf{s}^T\mathbf{N}\boldsymbol{\lambda} - \mu(\mathbf{s}^T\mathbf{s} - 1) \ . \tag{5.5.7}$$

The condition for $\mathcal{L}$ to be stationary is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \nabla f - \mathbf{N}\boldsymbol{\lambda} - 2\mu\mathbf{s} = 0 \ . \tag{5.5.8}$$

Premultiplying Eq. (5.5.8) by $\mathbf{N}^T$ and using Eq. (5.5.5) we obtain

$$\mathbf{N}^T\nabla f - \mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} = 0 \, , \tag{5.5.9}$$

or

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f \ . \tag{5.5.10}$$

So that from Eq. (5.5.8)

$$\mathbf{s} = \frac{1}{2\mu}[I - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T]\nabla f = \frac{1}{2\mu}\mathbf{P}\nabla f \ . \tag{5.5.11}$$

$\mathbf{P}$ is the projection matrix defined in Eq. (5.3.8). The factor of $1/2\mu$ is not significant because $\mathbf{s}$ defines only the direction of search, so in general we use $\mathbf{s} = -\mathbf{P}\nabla f$. To show that $\mathbf{P}$ indeed has the projection property, we need to prove that if $\mathbf{w}$ is an arbitrary vector, then $\mathbf{Pw}$ is in the subspace tangent to the active constraints, that is $\mathbf{Pw}$ satisfies

$$\mathbf{N}^T\mathbf{P}\mathbf{w} = 0 \ . \tag{5.5.12}$$

We can easily verify this by using the definition of $\mathbf{P}$.

Equation (5.3.8) which defines the projection matrix $\mathbf{P}$ does not provide the most efficient way for calculating it. Instead it can be shown that

$$\mathbf{P} = \mathbf{Q}_2^T\mathbf{Q}_2 \, , \tag{5.5.13}$$

where the matrix $\mathbf{Q}_2$ consists of the last $n - r$ rows of the $\mathbf{Q}$ factor in the QR factorization of $\mathbf{N}$ (see Eq. (5.3.9)).

A version of the gradient projection method known as the *generalized reduced gradient* method was developed by Abadie and Carpentier [8]. As a first step we select r linearly independent rows of $\mathbf{N}$, denote their transpose as $\mathbf{N}_1$ and partition $\mathbf{N}^T$ as

$$\mathbf{N}^T = [\mathbf{N}_1 \quad \mathbf{N}_2] \, . \tag{5.5.14}$$

Next we consider Eq. (5.5.5) for the components s_i of the direction vector. The r equations corresponding to $\mathbf{N}_1$ are then used to eliminate r components of $\mathbf{s}$ and obtain a reduced order problem for the direction vector.

Once we have identified $\mathbf{N}_1$ we can easily obtain $\mathbf{Q}_2$ which is given as

$$\mathbf{Q}_2^T = \begin{bmatrix} -\mathbf{N}_1^{-1}\mathbf{N}_2 \\ \mathbf{I} \end{bmatrix} \ . \tag{5.5.15}$$

Equation (5.5.15) can be verified by checking that $\mathbf{N}^T\mathbf{Q}_2^T = 0$, so that $\mathbf{Q}_2\mathbf{N} = 0$, which is the requirement that $\mathbf{Q}_2$ has to satisfy (see discussion following Eq. (5.3.11)).

After obtaining $\mathbf{s}$ from Eq. (5.5.11) we can continue the search with a one dimensional minimization, Eq. (5.5.4), unless $\mathbf{s} = 0$. When $\mathbf{s} = 0$ Eq. (5.3.7) indicates that the Kuhn-Tucker conditions may be satisfied. We then calculate the Lagrange multipliers from Eq. (5.3.6) or Eq. (5.3.11). If all the components of $\boldsymbol{\lambda}$ are non-negative, the Kuhn-Tucker conditions are indeed satisfied and the optimization can be terminated. If some of the Lagrange multipliers are negative, it is an indication that while no progress is possible with the current set of active constraints, it may be possible to proceed by removing some of the constraints associated with negative Lagrange multipliers. A common strategy is to remove the constraint associated with the most negative Lagrange multiplier and repeat the calculation of $\mathbf{P}$ and $\mathbf{s}$. If $\mathbf{s}$

is now non-zero, a one-dimensional search may be started. If $\mathbf{s}$ remains zero and there are still negative Lagrange multipliers, we remove another constraint until all Lagrange multipliers become positive and we satisfy the Kuhn-Tucker conditions.

After a search direction has been determined, a one dimensional search must be carried out to determine the value of α in Eq. (5.5.4). Unlike the unconstrained case, there is an upper limit on α set by the inactive constraints. As α increases, some of them may become active and then violated. Substituting $\mathbf{x} = \mathbf{x}_i + \alpha\mathbf{s}$ into Eq. (5.5.2) we obtain

$$g_j = \mathbf{a}_j^T(\mathbf{x}_i + \alpha\mathbf{s}) - b_j \geq 0\,, \tag{5.5.16}$$

or

$$\alpha \leq -(\mathbf{a}_j^T\mathbf{x}_i - b_j)/\mathbf{a}_j^T\mathbf{s} = -g_j(\mathbf{x}_i)/\mathbf{a}_j^T\mathbf{s}\ . \tag{5.5.17}$$

Equation (5.5.17) is valid if $\mathbf{a}_j^T\mathbf{s} < 0$. Otherwise, there is no upper limit on α due to the jth constraint. From Eq. (5.5.17) we get a different α, say α_j for each constraint. The upper limit on α is the minimum

$$\bar{\alpha} = \min_{\alpha_j>0,\ j\ni I_A} \alpha_j\ . \tag{5.5.18}$$

At the end of the move, new constraints may become active, so that the set of active constraints may need to be updated before the next move is undertaken.

The version of the gradient projection method presented so far is an extension of the steepest descent method. Like the steepest descent method, it may have slow convergence. The method may be extended to correspond to Newton or quasi-Newton methods. In the unconstrained case, these methods use a search direction defined as

$$\mathbf{s} = -\mathbf{B}\nabla f\,, \tag{5.5.19}$$

where $\mathbf{B}$ is the inverse of the Hessian matrix of f or an approximation thereof. The direction that corresponds to such a method in the subspace tangent to the active constraints can be shown [4] to be

$$\mathbf{s} = -\mathbf{Q}_2^T(\mathbf{Q}_2^T\mathbf{A}_L\mathbf{Q}_2)^{-1}\mathbf{Q}_2\nabla f\,, \tag{5.5.20}$$

where $\mathbf{A}_L$ is the Hessian of the Lagrangian function or an approximation thereof.

The gradient projection method has been generalized by Rosen to nonlinear constraints [9]. The method is based on linearizing the constraints about $\mathbf{x}_i$ so that

$$\mathbf{N} = [\nabla g_1(\mathbf{x}_i),\ \nabla g_2(\mathbf{x}_i), \ldots, \nabla g_r(\mathbf{x}_i)]\ . \tag{5.5.21}$$

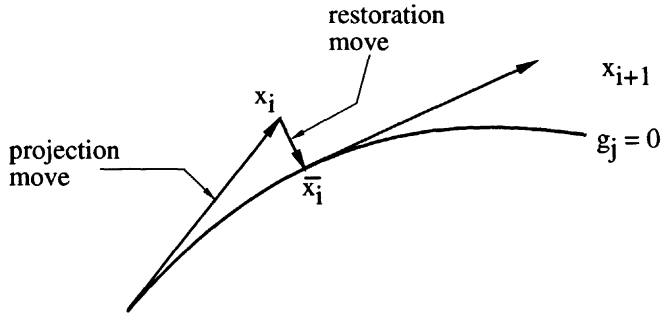


Figure 5.5.1 Projection and restoration moves.

The main difficulty caused by the nonlinearity of the constraints is that the one-dimensional search typically moves away from the constraint boundary. This is because we move in the tangent subspace which no longer follows exactly the constraint boundaries. After the one-dimensional search is over, Rosen prescribes a restoration move to bring $\mathbf{x}$ back to the constraint boundaries, see Figure 5.5.1.

To obtain the equation for the restoration move, we note that instead of Eq. (5.5.2) we now use the linear approximation

$$g_j \approx g_j(\mathbf{x}_i) + \nabla g_j^T(\bar{\mathbf{x}}_i - \mathbf{x}_i) \; . \tag{5.5.22}$$

We want to find a correction $\bar{\mathbf{x}}_i - \mathbf{x}_i$ in the tangent subspace (i.e. $\mathbf{P}(\bar{\mathbf{x}}_i - \mathbf{x}_i) = 0$) that would reduce g_j to zero. It is easy to check that

$$\bar{\mathbf{x}}_i - \mathbf{x}_i = -\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a(\mathbf{x}_i) \, , \tag{5.5.23}$$

is the desired correction, where $\mathbf{g}_a$ is the vector of active constraints. Equation (5.5.23) is based on a linear approximation, and may therefore have to be applied repeatedly until $\mathbf{g}_a$ is small enough.

In addition to the need for a restoration move, the nonlinearity of the constraints requires the re-evaluation of $\mathbf{N}$ at each point. It also complicates the choice of an upper limit for α which guarantees that we will not violate the presently inactive constraints. Haug and Arora [10] suggest a procedure which is better suited for the nonlinear case. The first advantage of their procedure is that it does not require a one-dimensional search. Instead, α in Eq. (5.5.4) is determined by specifying a desired specified reduction γ in the objective function. That is, we specify

$$f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \approx \gamma f(\mathbf{x}_i) \; . \tag{5.5.24}$$

Using a linear approximation with Eq. (5.5.4) we get

$$\alpha^* = -\frac{\gamma f(\mathbf{x}_i)}{\mathbf{s}^T \nabla f} \; . \tag{5.5.25}$$

The second feature of Haug and Arora's procedure is the combination of the projection and the restoration moves as

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha^* \mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a \, , \tag{5.5.26}$$

where Eqs. (5.5.4), (5.5.23) and (5.5.25) are used.

Example 5.5.1

Use the gradient projection method to solve the following problem

$$\begin{aligned} \textbf{minimize} \quad & f = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \textbf{subject to} \quad & g_1 = 2x_1 + x_2 + x_3 + 4x_4 - 7 \geq 0\,, \\ & g_2 = x_1 + x_2 + x_3^2 + x_4 - 5.1 \geq 0\,, \\ & x_i \geq 0, \qquad i = 1, \ldots, 4\,. \end{aligned}$$

Assume that as a result of previous moves we start at the point $\mathbf{x}_0^T = (2, 2, 1, 0)$, $f(\mathbf{x}_0) = 5.0$, where the nonlinear constraint g_2 is slightly violated. The first constraint is active as well as the constraint on x_4. We start with a combined projection and restoration move, with a target improvement of 10% in the objective function. At $\mathbf{x}_0$

$$\mathbf{N} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \\ 4 & 1 & 1 \end{bmatrix}, \qquad \mathbf{N}^T\mathbf{N} = \begin{bmatrix} 22 & 9 & 4 \\ 9 & 7 & 1 \\ 4 & 1 & 1 \end{bmatrix},$$

$$(\mathbf{N}^T\mathbf{N})^{-1} = \frac{1}{11}\begin{bmatrix} 6 & -5 & -19 \\ -5 & 6 & 14 \\ -19 & 14 & 73 \end{bmatrix},$$

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T = \frac{1}{11}\begin{bmatrix} 1 & -3 & 1 & 0 \\ -3 & 9 & -3 & 0 \\ 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \nabla f = \begin{Bmatrix} 2 \\ 4 \\ 2 \\ -3 \end{Bmatrix}.$$

The projection move direction is $\mathbf{s} = -\mathbf{P}\nabla f = [8/11, -24/11, 8/11, 0]^T$. Since the magnitude of a direction vector is unimportant we scale $\mathbf{s}$ to $\mathbf{s}^T = [1, -3, 1, 0]$. For a 10% improvement in the objective function $\gamma = 0.1$ and from Eq. (5.5.25)

$$\alpha^* = -\frac{0.1f}{\mathbf{s}^T\nabla f} = -\frac{0.1 \times 5}{-8} = 0.0625\,.$$

For the correction move we need the vector $\mathbf{g}_a$ of constraint values , $\mathbf{g}_a^T = (0, -0.1, 0)$, so the correction is

$$-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \frac{-1}{110}\begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix}.$$

Combining the projection and restoration moves, Eq. (5.5.26)

$$\mathbf{x}_1 = \begin{Bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{Bmatrix} + 0.0625\begin{Bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{Bmatrix} - \frac{1}{110}\begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 2.026 \\ 1.822 \\ 1.126 \\ 0 \end{Bmatrix},$$

we get $f(\mathbf{x}_1) = 4.64$, $g_1(\mathbf{x}_1) = 0$, $g_2(\mathbf{x}_1) = 0.016$. Note that instead of 10% reduction we got only 7% due to the nonlinearity of the objective function. However, we did satisfy the nonlinear constraint.• • •

Example 5.5.2

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0 , \\ & g_2 = x_1 - 5.73 \geq 0 , \\ & g_3 = x_2 - 7.17 \geq 0 , \end{aligned}$$

where the x_i are non-dimensional areas

$$x_i = \frac{A_i E}{1000P}, \qquad i = 1,2 .$$

The first constraint represents a limit on the vertical displacement, and the other two represent stress constraints.

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$, where $x_1 = 11.61$, $x_2 = 7.17$, and $f = 47.25$. The gradients of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix} , \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix} , \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} , \qquad \mathbf{N} = \begin{bmatrix} 0.1335 & 0 \\ 0.2021 & 1 \end{bmatrix} .$$

Because $\mathbf{N}$ is nonsingular, Eq. (5.3.8) shows that $\mathbf{P} = 0$. Also since the number of linearly independent active constraints is equal to the number of design variables the tangent subspace is a single point, so that there is no more room for progress. Using Eqs. (5.3.6) or (5.3.11) we obtain

$$\boldsymbol{\lambda} = \begin{Bmatrix} 22.47 \\ -2.798 \end{Bmatrix} .$$

The negative multiplier associated with g_3 indicates that this constraint can be dropped from the active set. Now

$$\mathbf{N} = \begin{bmatrix} 0.1335 \\ 0.2021 \end{bmatrix} .$$

The projection matrix is calculated from Eq. (5.3.8)

$$\mathbf{P} = \begin{bmatrix} 0.6962 & -0.4600 \\ -0.4600 & 0.3036 \end{bmatrix} , \qquad \mathbf{s} = -\mathbf{P}\nabla f = \begin{Bmatrix} -1.29 \\ 0.854 \end{Bmatrix} .$$

We attempt a 5% reduction in the objective function, and from Eq. (5.5.25)

$$\alpha^* = \frac{0.05 \times 47.25}{[-1.29 \; 0.854] \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}} = 0.988 .$$

Since there was no constraint violation at $\mathbf{x}_0$ we do not need a combined projection and correction step, and

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha^* \mathbf{s} = \begin{Bmatrix} 11.61 \\ 7.17 \end{Bmatrix} + 0.988 \begin{Bmatrix} -1.29 \\ 0.854 \end{Bmatrix} = \begin{Bmatrix} 10.34 \\ 8.01 \end{Bmatrix} .$$

At $\mathbf{x}_1$ we have $f(\mathbf{x}_1) = 44.89$, $g_1(\mathbf{x}_1) = -0.0382$. Obviously g_2 is not violated. If there were a danger of that we would have to limit α^* using Eq. (5.5.17). The violation of the nonlinear constraint is not surprising, and its size indicates that we should reduce the attempted reduction in f in the next move. At x_1, only g_1 is active so

$$\mathbf{N} = \nabla \mathbf{g}_1 = \begin{Bmatrix} 0.1684 \\ 0.1620 \end{Bmatrix} .$$

The projection matrix is calculated to be

$$\mathbf{P} = \begin{bmatrix} 0.4806 & -0.4996 \\ -0.4996 & 0.5194 \end{bmatrix}, \qquad \mathbf{s} = -\mathbf{P}\nabla f = \begin{Bmatrix} -0.5764 \\ 0.5991 \end{Bmatrix} .$$

Because of the violation we reduce the attempted reduction in f to 2.5%, so

$$\alpha^* = -\frac{0.025 \times 44.89}{[-0.567 \; 0.599] \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}} = 1.62 .$$

We need also a correction due to the constraint violation ($\mathbf{g}_a = -0.0382$)

$$-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \begin{Bmatrix} 0.118 \\ 0.113 \end{Bmatrix} .$$

Altogether

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha^* \mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \begin{Bmatrix} 10.34 \\ 8.01 \end{Bmatrix} - 1.62 \begin{Bmatrix} 0.576 \\ -0.599 \end{Bmatrix} + \begin{Bmatrix} 0.118 \\ 0.113 \end{Bmatrix} = \begin{Bmatrix} 9.52 \\ 9.10 \end{Bmatrix} .$$

We obtain $f(\mathbf{x}_2) = 44.32$, $g_1(\mathbf{x}_2) = -0.0328$.

The optimum design is actually $\mathbf{x}^T = (9.464, 9.464)$, $f(\mathbf{x}) = 44.78$, so after two iterations we are quite close to the optimum design.• • •

5.6 The Feasible Directions Method

The feasible directions method [11] has the opposite philosophy to that of the gradient projection method. Instead of following the constraint boundaries, we try to stay as far away as possible from them. The typical iteration of the feasible direction method starts at the boundary of the feasible domain (unconstrained minimization techniques are used to generate a direction if no constraint is active).

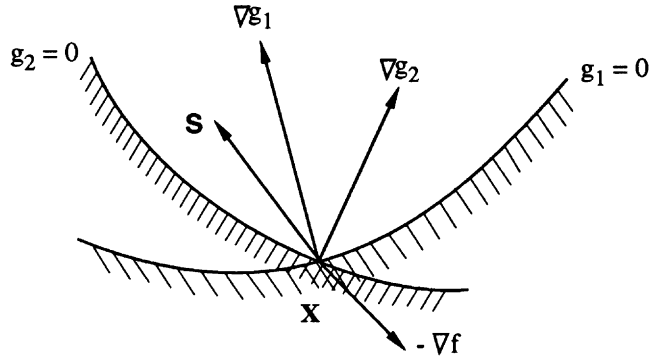


Figure 5.6.1 Selection of search direction using the feasible directions method.

Consider Figure 5.6.1. As a result of a previous move the design is at point $\mathbf{x}$ and we look for a direction $\mathbf{s}$ which keeps $\mathbf{x}$ in the feasible domain and improves the objective function. A vector $\mathbf{s}$ is defined as a feasible direction if at least a small step can be taken along it that does not immediately leave the feasible domain. If the constraints are smooth, this is satisfied if

$$\mathbf{s}^T \nabla g_j > 0, \qquad j \in I_A\,, \tag{5.6.1}$$

where I_A is the set of critical constraints at $\mathbf{x}$. The direction $\mathbf{s}$ is called a usable direction at the point $\mathbf{x}$ if in addition

$$\mathbf{s}^T \nabla f = \mathbf{s}^T \mathbf{g} < 0\,. \tag{5.6.2}$$

That is, $\mathbf{s}$ is a direction which reduces the objective function.

Among all possible choices of usable feasible directions we seek the direction which is best in some sense. We have two criteria for selecting a direction. On the one hand we want to reduce the objective function as much as possible. On the other hand we want to keep away from the constraint boundary as much as possible. A compromise is defined by the following maximization problem

$$\begin{aligned} &\textbf{maximize} && \beta \\ &\textbf{such that} && -\mathbf{s}^T \nabla g_j + \theta_j \beta \le 0, \qquad j \in I_A\,, \\ & && \mathbf{s}^T \nabla f + \beta \le 0, \qquad \theta_j \ge 0\,, \\ & && |s_i| \le 1\,. \end{aligned} \tag{5.6.3}$$

The θ_j are positive numbers called "push-off" factors because their magnitude determines how far $\mathbf{x}$ will move from the constraint boundaries. A value of $\theta_j = 0$ will result in a move tangent to the boundary of the the jth constraint, and so may be appropriate for a linear constraint. A large value of θ_j will result in a large angle between the constraint boundary and the move direction, and so is appropriate for a highly nonlinear constraint.

The optimization problem defined by Eq. (5.6.3) is linear and can be solved using the simplex algorithm. If $\beta_{max} > 0$, we have found a usable feasible direction. If we get $\beta_{max} = 0$ it can be shown that the Kuhn-Tucker conditions are satisfied.

Once a direction of search has been found, the choice of step length is typically based on a prescribed reduction in the objective function (using Eq. (5.5.25)). If at the end of the step no constraints are active, we continue in the same direction as long as $\mathbf{s}^T \nabla f$ is negative. We start the next iteration when $\mathbf{x}$ hits the constraint boundaries, or use a direction based on unconstrained technique if $\mathbf{x}$ is inside the feasible domain. Finally, if some constraints are violated after the initial step we make $\mathbf{x}$ retreat based on the value of the violated constraints. The method of feasible directions is implemented in the popular CONMIN program [12].

Example 5.6.1

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\ & g_2 = x_1 - 5.73 \geq 0\,, \\ & g_3 = x_2 - 7.17 \geq 0\,, \end{aligned}$$

where the x_i are non-dimensional areas

$$x_i = \frac{A_i E}{1000P}\,, \qquad i = 1,2\,.$$

The first constraint represents a limit on the vertical displacement, and the other two constraints represent stress constraints.

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$ where $\mathbf{x}_0^T = (11.61, 7.17)$ and $f = 47.25$. The gradient of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}\,, \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix}\,, \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}\,.$$

Selecting $\theta_1 = \theta_2 = 1$, we find that Eq. (5.6.3) becomes

$$\begin{aligned} \textbf{maximize} \quad & \beta \\ \textbf{subject to} \quad & -0.1335 s_1 - 0.2021 s_2 + \beta \leq 0\,, \\ & -s_2 + \beta \leq 0\,, \\ & 3s_1 + \sqrt{3}s_2 + \beta \leq 0\,, \\ & -1 \leq s_1 \leq 1\,, \\ & -1 \leq s_2 \leq 1\,. \end{aligned}$$

The solution of this linear program is $s_1 = -0.6172$, $s_2 = 1$, and we now need to execute the one dimensional search

$$\mathbf{x}_1 = \begin{Bmatrix} 11.61 \\ 7.17 \end{Bmatrix} + \alpha \begin{Bmatrix} -0.6172 \\ 1 \end{Bmatrix} .$$

Because the objective function is linear, this direction will remain a descent direction indefinitely, and α will be limited only by the constraints. The requirement that g_2 is not violated will lead to $\alpha = 9.527$, $x_1 = 5.73$, $x_2 = 16.7$ which violates g_1. We see that because g_1 is nonlinear, even though we start the search by moving away from it we still bump into it again (see Figure 5.6.2). It can be easily checked that for $\alpha > 5.385$ we violate g_1. So we take $\alpha = 5.385$ and obtain $x_1 = 8.29$, $x_2 = 12.56$, $f = 46.62$.

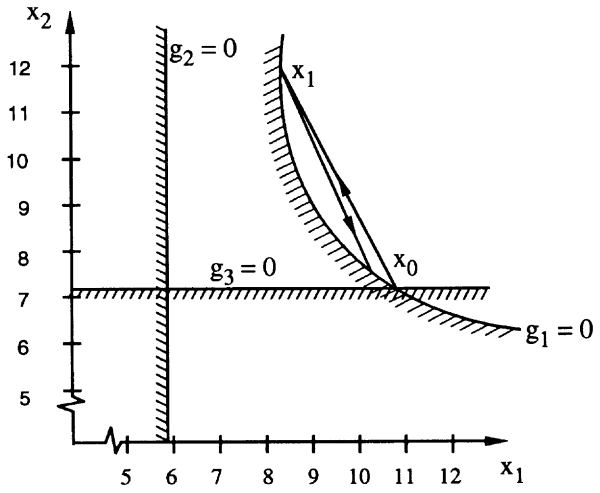


Figure 5.6.2 Feasible direction solution of 4 bar truss example.

For the next iteration we have only one active constraint

$$\nabla g_1 = \begin{Bmatrix} 0.2619 \\ 0.0659 \end{Bmatrix}, \qquad \nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix} .$$

The linear program for obtaining $\mathbf{s}$ is

$$\begin{aligned} \textbf{maximize} \quad & \beta \\ \textbf{subject to} \quad & -0.2619 s_1 - 0.0659 s_2 + \beta \leq 0 , \\ & 3 s_1 + \sqrt{3} s_2 + \beta \leq 0 , \\ & -1 \leq s_1 \leq 1 , \\ & -1 \leq s_2 \leq 1 . \end{aligned}$$

The solution of the linear program is $s_1 = 0.5512$, $s_2 = -1$, so that the one-dimensional search is

$$\mathbf{x} = \begin{Bmatrix} 8.29 \\ 12.56 \end{Bmatrix} + \alpha \begin{Bmatrix} 0.5512 \\ -1 \end{Bmatrix} .$$

Again α is limited only by the constraints. The lower limit on x_2 dictates $\alpha \leq 5.35$. However, the constraint g_1 is again more critical. It can be verified that for $\alpha > 4.957$ it is violated, so we take $\alpha = 4.957$, $x_1 = 11.02$, $x_2 = 7.60$, $f = 46.22$. The optimum design found in Example 5.1.2 is $x_1 = x_2 = 9.464$, $f = 44.78$. The design space and the two iterations are shown in Figure 5.6.2. • • •

5.7 Penalty Function Methods

When the energy crisis erupted in the middle seventies, the United States Congress passed legislation intended to reduce the fuel consumption of American cars. The target was an average fuel consumption of 27.5 miles per gallon for new cars in 1985. Rather than simply legislate this limit Congress took a gradual approach, with a different limit set each year to bring up the average from about 14 miles per gallon to the target value. Thus the limit was set at 26 for 1984, 25 for 1983, 24 for 1982, and so on. Furthermore, the limit was not absolute, but there was a fine of \$50 per 0.1 miles per gallon violation per car.

This approach to constraining the automobile companies to produce fuel efficient cars has two important aspects. First, by legislating a penalty proportional to the violation rather than an absolute limit, the government allowed the auto companies more flexibility. That meant they could follow a time schedule that approximated the government schedule without having to adhere to it rigidly. Second, the gradual approach made enforcement easier politically. Had the government simply set the ultimate limit for 1985 only, nobody would have paid attention to the law in the 1970's. Then as 1985 moved closer there would have been a rush to develop fuel efficient cars. The hurried effort could mean both non-optimal car designs and political pressure to delay the enforcement of the law.

The fuel efficiency law is an example in which constraints on behavior or economic activities are imposed via penalties whose magnitude depends on the degree of violation of the constraints. It is no wonder that this simple and appealing approach has found application in constrained optimization. Instead of applying constraints we replace them by penalties which depend on the degree of constraint violations. This approach is attractive because it replaces a constrained optimization problem by an unconstrained one.

The penalties associated with constraint violation have to be high enough so that the constraints are only slightly violated. However, just as there are political problems associated with imposing abrupt high penalties in real life, so there are numerical difficulties associated with such a practice in numerical optimization. For this reason we opt for a gradual approach where we start with small penalties and increase them gradually.

5.7.1 *Exterior Penalty Function*

The exterior penalty function associates a penalty with a violation of a constraint. The term 'exterior' refers to the fact that penalties are applied only in the exterior of the feasible domain. The most common exterior penalty function is one which associates a penalty which is proportional to the square of a violation. That is, the constrained minimization problem, Eq. (5.1)

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, n_e, \\ &&& g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, n_g, \end{aligned} \tag{5.7.1}$$

is replaced by

$$\begin{aligned} &\textbf{minimize} && \phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i=1}^{n_e} h_i^2(\mathbf{x}) + r \sum_{j=1}^{n_g} < -g_j >^2 \\ &&& r = r_1, r_2, \ldots, \qquad r_i \to \infty, \end{aligned} \tag{5.7.2}$$

where $< a >$ denote the positive part of a or $\max(a, 0)$. The inequality terms are treated differently from the equality terms because the penalty applies only for constraint violation. The positive multiplier r controls the magnitude of the penalty terms. It may seem logical to choose a very high value of r to ensure that no constraints are violated. However, as noted before, this approach leads to numerical difficulties illustrated later in an example. Instead the minimization is started with a relatively small value of r, and then r is gradually increased. A typical value for r_{i+1}/r_i is 5. A typical plot of $\phi(\mathbf{x}, r)$ as a function of r is shown in Figure 5.7.1 for a simple example.

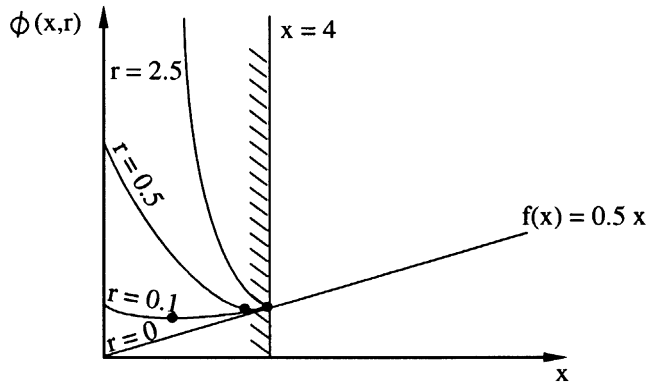


Figure 5.7.1 Exterior penalty function for $f = 0.5x$ subject to $x - 4 \geq 0$.

We see that as r is increased, the minimum of ϕ moves closer to the constraint boundary. However, the curvature of ϕ near the minimum also increases. It is

the high values of the curvature associated with large values of r which often lead to numerical difficulties. By using a sequence of values of r, we use the minima obtained for small values of r as starting points for the search with higher r values. Thus the ill-conditioning associated with the large curvature is counterbalanced by the availability of a good starting point.

Based on the type of constraint normalization given by Eq. (5.2) we can select a reasonable starting value for the penalty multiplier r. A rule of thumb is that one should start with the total penalty being about equal to the objective function for typical constraint violation of 50% of the response limits. In most optimization problems the total number of active constraints is about the same as or just slightly lower than the number of design variables. Assuming we start with one quarter of the eventual active constraints being violated by about 50% (or $g = -0.5$) then we have

$$f(\mathbf{x}_0) \approx r_0 \frac{n}{4}(0.5)^2, \qquad \text{or } r_0 = 16\frac{f(\mathbf{x}_0)}{n} . \tag{5.7.3}$$

It is also important to obtain a good starting point for restarting the optimization as r is increased. The minimum of the optimization for the previous value of r is a reasonable starting point, but one can do better. Fiacco and McCormick [13] show that the position of the minimum of $\phi(\mathbf{x}, r)$ has the asymptotic form

$$\mathbf{x}^*(r) = \mathbf{a} + \mathbf{b}/r, \qquad \text{as } r \to \infty . \tag{5.7.4}$$

Once the optimum has been found for two values of r, say r_{i-1}, and r_i, the vectors $\mathbf{a}$ and $\mathbf{b}$ may be estimated, and the value of $\mathbf{x}^*(r)$ predicted for subsequent values of r. It is easy to check that in order to satisfy Eq. (5.7.4), $\mathbf{a}$ and $\mathbf{b}$ are given as

$$\begin{aligned} \mathbf{a} &= \frac{c\mathbf{x}^*(r_{i-1}) - \mathbf{x}^*(r_i)}{c-1} , \\ \mathbf{b} &= [\mathbf{x}^*(r_{i-1}) - \mathbf{a}]\, r_{i-1} , \end{aligned} \tag{5.7.5}$$

where

$$c = r_{i-1}/r_i . \tag{5.7.6}$$

In addition to predicting a good value of the design variables for restarting the optimization for the next value of r, Eq. (5.7.4) provides us with a useful convergence criterion, namely

$$\|\mathbf{x}^* - \mathbf{a}\| \le \epsilon_1 , \tag{5.7.7}$$

where $\mathbf{a}$ is estimated from the last two values of r, and ϵ_1 is a specified tolerance chosen to be small compared to a typical value of $\|\mathbf{x}\|$.

A second convergence criterion is based on the magnitude of the penalty terms, which, as shown in Example 5.7.1, go to zero as r goes to infinity. Therefore, a reasonable convergence criterion is

$$\left|\frac{\phi - f}{f}\right| \le \epsilon_2 . \tag{5.7.8}$$

Finally, a criterion based on the change in the value of the objective function at the minimum f^* is also used

$$\left| \frac{f^*(r_i) - f^*(r_{i-1})}{f^*(r_i)} \right| \leq 0 \ . \tag{5.7.9}$$

A typical value for ϵ_2 or ϵ_3 is 0.001.

Example 5.7.1

Minimize $f = x_1^2 + 10x_2^2$ such that $x_1 + x_2 = 4$. We have

$$\phi = x_1^2 + 10x_2^2 + r(4 - x_1 - x_2)^2 \ .$$

The gradient $\nabla\phi$ is given as

$$\mathbf{g} = \begin{Bmatrix} 2x_1(1+r) + 2rx_2 - 8r \\ 2x_2(10+r) + 2rx_1 - 8r \end{Bmatrix} \ .$$

Setting the gradient to zero we obtain

$$x_1 = \frac{40r}{10 + 11r}, \qquad x_2 = \frac{4}{10 + 11r} \ .$$

The solution as a function of r is shown in Table 5.7.1.

Table 5.7.1 Minimization of ϕ for different penalty multipliers.

r	x_1	x_2	f	ϕ
1	1.905	0.1905	3.992	7.619
10	3.333	0.3333	12.220	13.333
100	3.604	0.3604	14.288	14.144
1000	3.633	0.3633	14.518	14.532

It can be seen that as r is increased the solution converges to the exact solution of $x^T = (3.636, 0.3636)$, $f = 14.54$. The convergence is indicated by the shrinking difference between the objective function and the augmented function ϕ. The Hessian of ϕ is given as

$$\mathbf{H} = \begin{bmatrix} 2 + 2r & 2r \\ 2r & 20 + 2r \end{bmatrix} \ .$$

As r increases this matrix becomes more and more ill-conditioned, as all four components become approximately $2r$. This ill-conditioning of the Hessian matrix for large values of r often occurs when the exterior penalty function is used, and can cause numerical difficulties for large problems.

We can use Table 5.7.1 to test the extrapolation procedure, Eq. (5.7.4). For example, with the values of $r = 1$ and $r = 10$, Eq. (5.7.5) gives

$$\mathbf{a} = \frac{0.1\mathbf{x}^*(1) - \mathbf{x}^*(10)}{-0.9} = \begin{Bmatrix} 3.492 \\ 0.3492 \end{Bmatrix} ,$$

$$\mathbf{b} = \mathbf{x}^*(1) - \mathbf{a} = \begin{Bmatrix} -0.159 \\ -0.0159 \end{Bmatrix} .$$

We can now use Eq. (5.7.4) to find a starting point for the optimization for $r = 100$ to get

$$\mathbf{a} + \mathbf{b}/100 = (3.490, 0.3490)^T ,$$

which is substantially closer to $\mathbf{x}^*(100) = (3.604, 0.3604)^T$ than to $\mathbf{x}^*(10) = (3.333, 0.3333)^T$. • • •

5.7.2 *Interior and Extended Interior Penalty Functions*

With the exterior penalty function, constraints contribute penalty terms only when they are violated. As a result, the design typically moves in the infeasible domain. If the minimization is terminated before r becomes very large (for example, because of shortage of computer resources) the resulting designs may be useless. When only inequality constraints are present, it is possible to define an interior penalty function that keeps the design in the feasible domain. The common form of the interior penalty method replaces the inequality constrained problem

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g , \end{aligned} \tag{5.7.10}$$

by

$$\begin{aligned} &\textbf{minimize} \quad \phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) , \\ &\qquad\qquad r = r_1, r_2, \ldots, \qquad r_i \to 0, \quad r_i > 0 . \end{aligned} \tag{5.7.11}$$

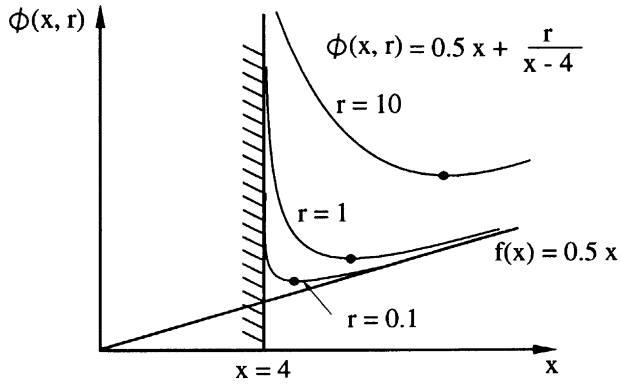


Figure 5.7.2 Interior penalty function for $f(x) = 0.5x$ subject to $x - 4 \geq 0$.

The penalty term is proportional to $1/g_j$ and becomes infinitely large at the boundary of the feasible domain creating a barrier there (interior penalty function methods are sometimes called barrier methods). It is assumed that the search is confined to the feasible domain. Otherwise, the penalty becomes negative which does not make any sense. Figure 5.7.2 shows the application of the interior penalty function to the simple example used for the exterior penalty function in Figure 5.7.1. Besides the inverse penalty function defined in Eq. (5.7.11), there has been some use of a logarithmic interior penalty function

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{j=1}^{n_g} log(g_j(\mathbf{x})) \ . \tag{5.7.12}$$

While the interior penalty function has the advantage over the exterior one in that it produces a series of feasible designs, it also requires a feasible starting point. Unfortunately, it is often difficult to find such a feasible starting design. Also, because of the use of approximation (see Chapter 6), it is quite common for the optimization process to stray occasionally into the infeasible domain. For these reasons it may be advantageous to use a combination of interior and exterior penalty functions called an extended interior penalty function. An example is the quadratic extended interior penalty function of Haftka and Starnes [14]

$$\begin{aligned} \phi(\mathbf{x}, r) &= f(\mathbf{x}) + r \sum_{j=1}^{n_g} p(g_j) \, , \\ r &= r_1, r_2, \ldots , \qquad r_i \to 0 \, , \end{aligned} \tag{5.7.13}$$

where

$$p(g_j) = \begin{cases} 1/g_j & \text{for } g_j \geq g_0 \\ 1/g_0[3 - 3(g_j/g_0) + (g_j/g_0)^2] & \text{for } g_i < g_0 \ . \end{cases} \tag{5.7.14}$$

It is easy to check that $p(g_j)$ has continuity up to second derivatives. The transition parameter g_0 which defines the boundary between the interior and exterior parts of the penalty terms must be chosen so that the penalty associated with the constraint, $rp(g_j)$, becomes infinite for negative g_j as r tends to zero. This results in the requirement that

$$r/g_0^3 \to \infty, \quad \text{as } r \to 0 \ . \tag{5.7.15}$$

This can be achieved by selecting g_0 as

$$g_0 = cr^{1/2} \, , \tag{5.7.16}$$

where c is a constant.

It is also possible to include equality constraints with interior and extended interior penalty functions. For example, the interior penalty function Eq. (5.7.11) is augmented as

$$\begin{aligned} \phi(\mathbf{x}, r) &= f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) + r^{-1/2} \sum_{i=1}^{n_e} h_i^2(\mathbf{x}) \, , \\ r &= r_1, r_2, \ldots , \qquad r_i \to 0 \ . \end{aligned} \tag{5.7.17}$$

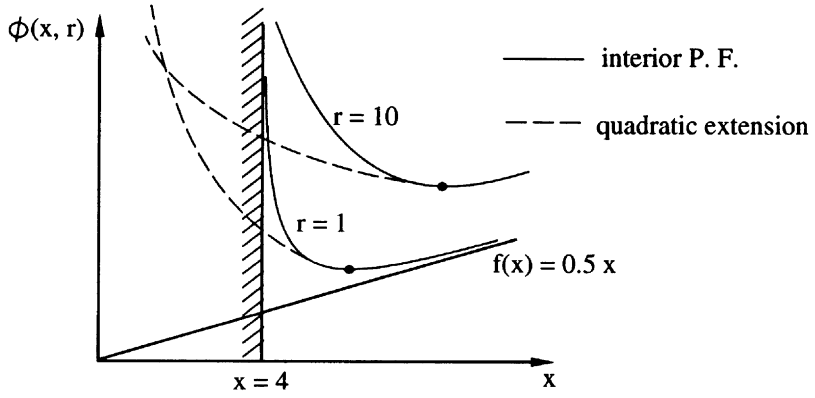


Figure 5.7.3 Extended interior penalty function for $f(x) = 0.5x$ *subject to* $g(x) = x - 4 \geq 0$.

The considerations for the choice of an initial value of r are similar to those for the exterior penalty function. A reasonable choice for the interior penalty function would require that $n/4$ active constraints at $g = 0.5$ (that is 50% margin for properly normalized constraints) would result in a total penalty equal to the objective function. Using Eq. (5.7.3) we obtain

$$f(\mathbf{x}) = \frac{n}{4}\frac{r}{0.5}, \qquad \text{or} \quad r = 2f(\mathbf{x})/n \ .$$

For the extended interior penalty function it is more reasonable to assume that the $n/4$ constraints are critical ($g = 0$), so that from Eq. (5.7.13)

$$f(\mathbf{x}) = r\frac{n}{4}\frac{3}{g_0}, \qquad \text{or} \quad r = \frac{4}{3}g_0 f(\mathbf{x})/n \ .$$

A reasonable starting value for g_0 is 0.1. As for the exterior penalty function, it is possible to obtain an expression for the asymptotic (as $r \to 0$) coordinates of the minimum of ϕ as [10]

$$\mathbf{x}^*(r) = \mathbf{a} + \mathbf{b}r^{1/2}, \quad \text{as} \quad r \to 0\,, \tag{5.7.18}$$

and

$$f^*(r) = a + br^{1/2}, \quad \text{as} \quad r \to 0 \ .$$

$\mathbf{a}$, $\mathbf{b}$, a and b may be estimated once the minimization has been carried out for two values of r. For example, the estimates for $\mathbf{a}$ and $\mathbf{b}$ are

$$\begin{aligned} \mathbf{a} &= \frac{c^{1/2}\mathbf{x}^*(r_{i-1}) - \mathbf{x}^*(r_i)}{c^{1/2} - 1}\,, \\ \mathbf{b} &= \frac{\mathbf{x}^*(r_{i-1}) - \mathbf{a}}{r_{i-1}^{1/2}}\,, \end{aligned} \tag{5.7.19}$$

where $c = r_i/r_{i-1}$. As in the case of exterior penalty function, these expressions may be used for convergence tests and extrapolation.

5.7.3 *Unconstrained Minimization with Penalty Functions*

Penalty functions convert a constrained minimization problem into an unconstrained one. It may seem that we should now use the best available methods for unconstrained minimization, such as quasi-Newton methods. This may not necessarily be the case. The penalty terms cause the function ϕ to have large curvatures near the constraint boundary even if the curvatures of the objective function and constraints are small. This effect permits an inexpensive approximate calculation of the Hessian matrix, so that we can use Newton's method without incurring the high cost of calculating second derivatives of constraints. This may be more attractive than using quasi-Newton methods (where the Hessian is also approximated on the basis of first derivatives) because a good approximation is obtained with a single analysis rather than with the n moves typically required for a quasi-Newton method. Consider, for example, an exterior penalty function applied to equality constraints

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i=1}^{n_e} h_i^2(\mathbf{x}) \; . \tag{5.7.20}$$

The second derivatives of ϕ are given as

$$\frac{\partial^2 \phi}{\partial x_k \partial x_l} = \frac{\partial^2 f}{\partial x_k \partial x_l} + r \sum_{i=1}^{n_e} 2 \left(\frac{\partial h_i}{\partial x_k} \frac{\partial h_i}{\partial x_l} + h_i \frac{\partial^2 h_i}{\partial x_k \partial x_l} \right) \; . \tag{5.7.21}$$

Because of the equality constraint, h_i is close to zero, especially for the later stages of the optimization (large r), and we can neglect the last term in Eq. (5.7.21). For large values of r we can also neglect the first term, so that we can calculate second derivatives of ϕ based on first derivatives of the constraints. The availability of inexpensive second derivatives permits the use of Newton's method where the number of iterations is typically independent of the number of design variables. Quasi-Newton and conjugate gradient methods, on the other hand, require a number of iterations proportional to the number of design variables. Thus the use of Newton's method becomes attractive when the number of design variables is large. The application of Newton's method with the above approximation of second derivatives is known as the Gauss-Newton method.

For the interior penalty function we have a similar situation. The augmented objective function ϕ is given as

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) \, , \tag{5.7.22}$$

and the second derivatives are

$$\frac{\partial^2 \phi}{\partial x_k \partial x_l} = \frac{\partial^2 f}{\partial x_k \partial x_l} + r \sum_{j=1}^{n_g} \frac{1}{g_j^3} \left(2 \frac{\partial g_j}{\partial x_k} \frac{\partial g_j}{\partial x_l} - g_j \frac{\partial^2 g_j}{\partial x_k \partial x_l} \right) \; . \tag{5.7.23}$$

Now the argument for neglecting the first and last terms in Eq. (5.7.23) is somewhat lengthier. First we observe that because of the $1/g_j^3$ term, the second derivatives are dominated by the critical constraints (g_j small). For these constraints the last term in Eq. (5.7.23) is negligible compared to the first-derivative term because g_j is small. Finally, from Eq. (5.7.18) it can be shown that r/g_j^3 goes to infinity for active constraints as r goes to zero, so that the first term in Eq. (5.7.23) can be neglected compared to the second. The same argument can also be used for extended interior penalty functions [14].

The power of the Gauss-Newton method is shown in [14] for a high- aspect-ratio wing made of composite materials (see Figure 5.7.4) designed subject to stress and displacement constraints.

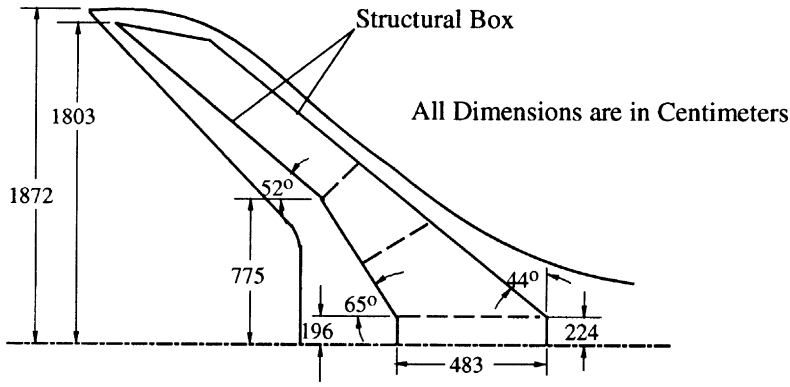


Figure 5.7.4 Aerodynamic planform and structural box for high-aspect ratio wing, from [14].

Table 5.7.2 Results of high-aspect-ratio wing study

Number of design variables	CDC 6600 CPU time sec	Total number of unconstrained minimizations	Total number of analyses	Final mass, kg
13	142	4	21	887.3
25	217	4	19	869.1
32	293	5	22	661.7
50	460	5	25	658.2
74	777	5	28	648.6
146	1708	5	26	513.0

The structural box of the wing was modeled with a finite element model with 67 nodes and 290 finite elements. The number of design variables controlling the thickness of the various elements was varied from 13 to 146. The effect of the number of design variables on the number of iterations (analyses) is shown in Table 5.7.2.

It is seen that the number of iterations per unconstrained minimization is almost constant (about five). With a quasi-Newton method that number may be expected to be similar to the number of design variables.

Because of the sharp curvature of ϕ near the constraint boundary, it may also be appropriate to use specialized line searches with penalty functions [15].

5.7.4 *Integer Programming with Penalty Functions*

An extension of the penalty function approach has been implemented by Shin et al. [16] for problems with discrete-valued design variables. The extension is based on introduction of additional penalty terms into the augmented-objective function $\phi(\mathbf{x}, r)$ to reflect the requirement that the design variables take discrete values,

$$x_i \in X_i = \{d_{i1}, d_{i2}, \ldots, d_{il}\}, \qquad i \in I_d, \tag{5.7.24}$$

where I_d is the set of design variables that can take only discrete values, and X_i is the set of allowable discrete values. Note that several variables may have the same allowable set of discrete values. In this case the augmented objective function which includes the penalty terms due to constraints and the non-discrete values of the design variables is defined as

$$\phi(\mathbf{x}, r, s) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} p(g_j) + s \sum_{i \in I_d} \psi_d(x_i) \ , \tag{5.7.25}$$

where s is a penalty multiplier for non-discrete values of the design variables, and $\psi_d(x_i)$ the penalty term for non-discrete values of the ith design variable. Different forms for the discrete penalty function are possible. The penalty terms $\psi_d(x_i)$ are assumed to take the following sine-function form in Ref. [16],

$$\psi_d(x_i) = \frac{1}{2}\left(\sin \frac{2\pi[x_i - \frac{1}{4}(d_{i(j+1)} + 3d_{ij})]}{d_{ij+1} - d_{ij}} + 1\right) , \qquad d_{ij} \le x_i \le d_{i(j+1)} \ . \tag{5.7.26}$$

While penalizing the non-discrete valued design variables, the functions $\psi_d(x_i)$ assure the continuity of the first derivatives of the augmented function at the discrete values of the design variables. The response surfaces generated by Eq. (5.7.25) are determined according to the values of the penalty multipliers r and s. In contrast to the multiplier r, which initially has a large value and decreases as we move from one iteration to another, the value of the multiplier s is initially zero and increases gradually.

One of the important factors in the application of the proposed method is to determine when to activate s, and how fast to increase it to obtain discrete optimum design. Clearly, if the initial value of s is too big and introduced too early in the design process, the design variables will be trapped away from the global minimum, resulting in a sub-optimal solution. To avoid this problem, the multiplier s has to be activated after optimization of several response surfaces which include only constraint penalty terms. In fact, since sometimes the optimum design with discrete values is

in the neighborhood of the continuous optimum, it may be desirable not to activate the penalty for the non-discrete design variables until reasonable convergence to the continuous solution is achieved. This is especially true for problems in which the intervals between discrete values are very small.

A criterion for the activation of the non-discrete penalty multiplier s is the same as the convergence criterion of Eq. (5.7.6), that is

$$\left|\frac{\phi - f}{f}\right| \leq \epsilon_c \ . \tag{5.7.27}$$

A typical value for ϵ_c is 0.01. The magnitude of the non-discrete penalty multiplier, s, at the first discrete iteration is calculated such that the penalty associated with the discrete-valued design variables that are not at their allowed values is of the order of 10 percent of the constraint penalty.

$$s \approx 0.1 r p(g) \ . \tag{5.7.28}$$

As the iteration for discrete optimization proceeds, the non-discrete penalty multiplier for the new iteration is increased by a factor of the order of 10. It is also important to decide how to control the penalty multiplier for the constraints, r, during the discrete optimization process. If r is decreased for each discrete optimization iteration as in the continuous optimization process, the design can be stalled due to high penalties for constraint violation. Thus, it is suggested that the penalty multiplier r be frozen at the end of the continuous optimization process. However, the nearest discrete solution at this response surface may not be a feasible design, in which case the design must move away from the continuous optimum by moving back to the previous response surface. This can be achieved by increasing the penalty multiplier, r, by a factor of 10.

The solution process for the discrete optimization is terminated if the design variables are sufficiently close to the prescribed discrete values. The convergence criterion for discrete optimization is

$$\max_{i \in \mathbf{I}_i} \left\{ \mathbf{min} \left\{ \frac{|x_i - d_{ij}|}{d_{i(j+1)} - d_{ij}}, \frac{|x_i - d_{i(j+1)}|}{d_{i(j+1)} - d_{ij}} \right\} \right\} \leq \epsilon_d \, , \tag{5.7.29}$$

where a typical value of the convergence tolerance ϵ_d is 0.001.

Example 5.7.2

Cross-sectional areas of members of a two-bar truss shown in the Figure 5.7.5 are to be selected from a discrete set of values, $A_i \in \{1.0, 1.5, 2.0\}$, $i = 1, 2$. Determine the minimum weight structure using the modified penalty function approach such that the horizontal displacement u at the point of application of the force does not exceed $2/3(Fl/E)$. Use a tolerance $\epsilon_c = 0.1$ for the activation of the penalty terms for non-discrete valued design variables, and a convergence tolerance for the design variables $\epsilon_d = 0.001$.

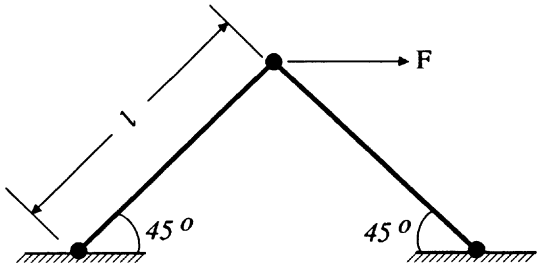


Figure 5.7.5 Two-bar truss.

Upon normalization, the design problem is posed as

$$\begin{aligned} \textbf{minimize} \quad & f = \frac{W}{\rho l} = x_1 + x_2 \\ \textbf{subject to} \quad & g = \frac{uE}{Fl} = 1.5 - 1/x_1 - 1/x_2 \geq 0\,, \\ & x_i = A_i \in \{1.0, 1.5, 2.0\}, \qquad i = 1, \ldots, 2\,. \end{aligned}$$

Using an initial design of $x_1 = x_2 = 5$ and transition parameter $g_0 = 0.1$, we have $g = 1.1 > g_0$, therefore, from Eq. (5.7.14) the penalty terms for the constraints are in the form of $p(g) = 1/g$. The augmented function for the extended interior penalty function approach is

$$\phi = x_1 + x_2 + \frac{r}{1.5 - 1/x_1 - 1/x_2}\,.$$

Setting the gradient to zero, we can show that the minimum of the augmented function as a function of the penalty multiplier r is

$$x_1 = x_2 = \frac{24 + \sqrt{576 - 36\,(16 - 4r)}}{18}\,.$$

The initial value of the penalty multiplier r is chosen so that the penalty introduced for the constraint is equal to the objective function value,

$$r\frac{1}{g(x_0)} = f(x_0), \qquad r = 11\,.$$

The minima of the augmented function as functions of the penalty multiplier r are shown in Table 5.7.3 . After four iterations the constraint penalty $(\phi - f)$ is within the desired range of the objective function to activate the penalty terms for the non-discrete values of the design variables.

From Eq. (5.7.25) the augmented function for the modified penalty function approach has the form

$$\begin{aligned} \phi = & x_1 + x_2 + \frac{r}{1.5 - 1/x_1 - 1/x_2} + \frac{s\,\{1 + \sin[4\pi\,(x_1 - 1.125)]\}}{2} \\ & + (s/2)\,\{1 + \sin[4\pi\,(x_2 - 1.125)]\}\,. \end{aligned}$$

Table 5.7.3 Minimization of ϕ without the discrete penalty

r	x_1	x_2	f	g	ϕ
-	5.000	5.000	10.00	1.100	-
11	3.544	3.544	7.089	0.9357	18.844
1.1	2.033	2.033	4.065	0.5160	6.197
0.11	1.554	1.554	3.109	0.2134	3.624
0.011	1.403	1.403	2.807	0.0747	2.954

The minimum of the augmented function can again be obtained by setting the gradient to zero

$$1 - \frac{r}{(1.5 - 2/x_1)^2 {x_1}^2} + 2\pi s \cos[4\pi\,(x_1 - 1.125)] \;= 0\,,$$

which can be solved numerically. The initial value of the penalty multiplier s is calculated from Eq. (5.7.28)

$$s = 0.1\ (0.011)\ \frac{1}{0.0747} = 0.0147\ .$$

The minima of the augmented function (which includes the penalty for the non-discrete valued variables) are shown in Table 5.7.4 as a function of s.

Table 5.7.4 Minimization of ϕ with the discrete penalty

r	s	x_1	x_2	f	ϕ
0.011	0.0147	1.406	1.406	2.813	2.963
	0.1472	1.432	1.432	2.864	3.021
	1.472	1.493	1.493	2.986	3.060
	14.72	1.499	1.499	2.999	3.065
	147.2	1.500	1.500	3.000	3.066

After four discrete iterations we obtain a minimum at $x_1 = x_2 = 3/2$. There are two more minima, $\mathbf{x} = (2,1)$ and $\mathbf{x} = (1,2)$, with the same value of the objective function of $f = 3.0$. • • •

5.8 Multiplier Methods

Multiplier methods combine the use of Lagrange multipliers with penalty functions. When only Lagrange multipliers are employed the optimum is a stationary point rather than a minimum of the Lagrangian function. When only penalty functions are employed we have a minimum but also ill-conditioning. By using both we may hope to get an unconstrained problem where the function to be minimized does not suffer from ill-conditioning. A good survey of multiplier methods was conducted by

Bertsekas [17]. We study first the use of multiplier methods for equality constrained problems.

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad h_j(\mathbf{x}) = 0, \qquad j = 1, \ldots, n_e \, . \end{aligned} \tag{5.8.1}$$

We define the augmented Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) - \sum_{j=1}^{n_e} \lambda_j h_j(\mathbf{x}) + r \sum_{j=1}^{n_e} h_j^2(\mathbf{x}) \, . \tag{5.8.2}$$

If all the Lagrange multipliers are set to zero, we get the usual exterior penalty function. On the other hand, if we use the correct values of the Lagrange multipliers, λ_j^*, it can be shown that we get the correct minimum of problem (5.8.1) for any positive value of r. Then there is no need to use the large value of r required for the exterior penalty function. Of course, we do not know what are the correct values of the Lagrange multipliers.

Multiplier methods are based on estimating the Lagrange multipliers. When the estimates are good, it is possible to approach the optimum without using large r values. The value of r needs to be only large enough so that $\mathcal{L}$ has a minimum rather than a stationary point at the optimum. To obtain an estimate for the Lagrange multipliers we compare the stationarity conditions for $\mathcal{L}$,

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} (\lambda_j - 2rh_j) \frac{\partial h_j}{\partial x_i} = 0 \, , \tag{5.8.3}$$

with the exact conditions for the Lagrange multipliers

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} \lambda_j^* \frac{\partial h_j}{\partial x_i} = 0 \, . \tag{5.8.4}$$

Comparing Eqs. (5.8.3) and (5.8.4) we expect that

$$\lambda_j - 2rh_j \to \lambda_j^* \, , \tag{5.8.5}$$

as the minimum is approached. Based on this relation, Hestenes [18] suggested using Eq. (5.8.5) as an estimate for λ_j^*. That is

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} - 2r^{(k)} h_j^{(k)} \, , \tag{5.8.6}$$

where k is an iteration number.

Example 5.8.1

We repeat Example 5.7.1 using Hestenes' multiplier method.

$$f(\mathbf{x}) = x_1^2 + 10x_2^2 ,$$
$$h(\mathbf{x}) = x_1 + x_2 - 4 = 0 .$$

The augmented Lagrangian is

$$\mathcal{L} = x_1^2 + 10x_2^2 - \lambda(x_1 + x_2 - 4) + r(x_1 + x_2 - 4)^2 .$$

To find the stationary points of the augmented Lagrangian we differentiate with respect to x_1 and x_2 to get

$$2x_1 - \lambda + 2r(x_1 + x_2 - 4) = 0 ,$$
$$20x_2 - \lambda + 2r(x_1 + x_2 - 4) = 0 ,$$

which yield

$$x_1 = 10x_2 = \frac{5\lambda + 40r}{10 + 11r} .$$

We want to compare the results with those of Example 5.7.1, so we start with the same initial r value $r_0 = 1$, the initial estimate of $\lambda = 0$ and get

$$\mathbf{x}_1 = (1.905, 0.1905)^T, \qquad h = -1.905 .$$

So, using Eq. (5.8.6) we estimate $\lambda^{(1)}$ as

$$\lambda^{(1)} = -2 \times 1 \times (-1.905) = 3.81 .$$

We next repeat the optimization with $r^{(1)} = 10$, $\lambda^{(1)} = 3.81$ and get

$$\mathbf{x}_2 = (3.492, 0.3492)^T, \qquad h = -0.1587 .$$

For the same value of r, we obtained in Example 5.7.1 $\mathbf{x}_2 = (3.333, 0.3333)^T$, so that we are now closer to the exact solution of $\mathbf{x} = (3.636, 0, 3636)^T$. Now we estimate a new λ from Eq. (5.8.6)

$$\lambda^{(2)} = 3.81 - 2 \times 10 \times (-0.1587) = 6.984 .$$

For the next iteration we may, for example, fix the value of r at 10 and change only λ. For $\lambda = 6.984$ we obtain

$$\mathbf{x}_3 = (3.624, 0.3624), \qquad h = -0.0136 ,$$

which shows that good convergence can be obtained without increasing r.• • •

There are several ways to extend the multiplier method to deal with inequality constraints. The formulation below is based on Fletcher's work [19]. The constrained problem that we examine is

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g \, . \end{aligned} \tag{5.8.7}$$

The augmented Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} \langle \frac{\lambda_j}{2r} - g_j \rangle^2 \, , \tag{5.8.8}$$

where $< a >= max(a, 0)$. The condition of stationarity of $\mathcal{L}$ is

$$\frac{\partial f}{\partial x_i} - 2r \sum_{j=1}^{n_g} \langle \frac{\lambda_j}{2r} - g_j \rangle \frac{\partial g_j}{\partial x_i} = 0 \, . \tag{5.8.9}$$

The exact stationarity condition is

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j^* \frac{\partial g_j}{\partial x_i} = 0 \, , \tag{5.8.10}$$

where it is also required that $\lambda_j^* g_j = 0$. Comparing Eqs (5.8.9) and (5.8.10) we expect an estimate for λ_j^* of the form

$$\lambda_j^* = max(\lambda_j - 2r g_j, 0) \, . \tag{5.8.11}$$

5.9 Projected Lagrangian Methods (Sequential Quadratic Programming)

The addition of penalty terms to the Lagrangian function by multiplier methods converts the optimum from a stationary point of the Lagrangian function to a minimum point of the augmented Lagrangian. Projected Lagrangian methods achieve the same result by a different method. They are based on a theorem that states that the optimum is a minimum of the Lagrangian function in the subspace of vectors orthogonal to the gradients of the active constraints (the tangent subspace). Projected Lagrangian methods employ a quadratic approximation to the Lagrangian in this subspace. The direction seeking algorithm is more complex than for the methods considered so far. It requires the solution of a quadratic programming problem, that is an optimization problem with a quadratic objective function and linear constraints. Projected Lagrangian methods are part of a class of methods known as sequential quadratic programming (SQP)methods. The extra work associated with the solution of the quadratic programming direction seeking problem is often rewarded by faster convergence.

The present discussion is a simplified version of Powell's projected Lagrangian method [20]. In particular we consider only the case of inequality constraints

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g \ . \end{aligned} \tag{5.9.1}$$

Assume that at the ith iteration the design is at $\mathbf{x}_i$, and we seek a move direction $\mathbf{s}$. The direction $\mathbf{s}$ is the solution of the following quadratic programming problem

$$\begin{aligned} &\textbf{minimize} \quad \phi(\mathbf{s}) = f(\mathbf{x}_i) + \mathbf{s}^T \mathbf{g}(\mathbf{x}_i) + \frac{1}{2}\mathbf{s}^T \mathbf{A}(\mathbf{x}_i, \boldsymbol{\lambda}_i)\mathbf{s} \\ &\textbf{such that} \quad g_j(\mathbf{x}_i) + \mathbf{s}^T \nabla g_j(\mathbf{x}_i) \geq 0, \qquad j = 1, \ldots, n_g \, , \end{aligned} \tag{5.9.2}$$

where $\mathbf{g}$ is the gradient of f, and $\mathbf{A}$ is a positive definite approximation to the Hessian of the Lagrangian function discussed below. This quadratic programming problem can be solved by a variety of methods which take advantage of its special nature. The solution of the quadratic programming problem yields $\mathbf{s}$ and $\boldsymbol{\lambda}_{i+1}$. We then have

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{s} \, , \tag{5.9.3}$$

where α is found by minimizing the function

$$\psi(\alpha) = f(\mathbf{x}) + \sum_{j=1}^{n_g} \mu_j |min(0, g_j(\mathbf{x}))| \, , \tag{5.9.4}$$

and the μ_j are equal to the absolute values of the Lagrange multipliers for the first iteration, i.e.

$$\mu_j = max[|\lambda_j^{(i)}, \frac{1}{2}(\mu_j^{(i-1)} + |\lambda_j^{(i-1)}|)] \, , \tag{5.9.5}$$

with the superscript i denoting iteration number. The matrix $\mathbf{A}$ is initialized to some positive definite matrix (e.g the identity matrix) and then updated using a BFGS type equation (see Chapter 4).

$$\mathbf{A}_{new} = \mathbf{A} - \frac{\mathbf{A}\Delta\mathbf{x}\Delta\mathbf{x}^T\mathbf{A}}{\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}} + \frac{\Delta\mathbf{l}\Delta\mathbf{l}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} \, , \tag{5.9.6}$$

where

$$\Delta\mathbf{x} = \mathbf{x}_{i+1} - \mathbf{x}_i \ , \qquad \Delta\mathbf{l} = \nabla_x \mathcal{L}(\mathbf{x}_{i+1}, \boldsymbol{\lambda}_i) - \nabla_x \mathcal{L}(\mathbf{x}_i, \boldsymbol{\lambda}_i) \, , \tag{5.9.7}$$

where $\mathcal{L}$ is the Lagrangian function and ∇_x denotes the gradient of the Lagrangian function with respect to $\mathbf{x}$. To guarantee the positive definiteness of $\mathbf{A}$, $\Delta\mathbf{l}$ is modified if $\Delta\mathbf{x}^T\Delta\mathbf{l} \leq 0.2\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}$ and replaced by

$$\Delta\mathbf{l}' = \theta\Delta\mathbf{l} + (1-\theta)\mathbf{A}\Delta\mathbf{x} \, , \tag{5.9.8}$$

where

$$\theta = \frac{0.8\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}}{\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{x}^T\Delta\mathbf{l}} \ . \tag{5.9.9}$$

Example 5.9.1

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\ & g_2 = x_1 - 5.73 \geq 0\,, \\ & g_3 = x_2 - 7.17 \geq 0\,. \end{aligned}$$

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$ where $x_1 = 11.61$, $x_2 = 7.17$ and $f = 47.25$. The gradient of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}\,, \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix}\,, \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}\,, \qquad \mathbf{N} = \begin{bmatrix} 0.1335 & 0 \\ 0.2021 & 1 \end{bmatrix}\,.$$

We start with $\mathbf{A}$ set to the unit matrix so that

$$\phi(\mathbf{s}) = 47.25 + 3s_1 + \sqrt{3}s_2 + 0.5s_1^2 + 0.5s_2^2\,,$$

and the linearized constraints are

$$\begin{aligned} g_1(\mathbf{s}) &= 0.1335s_1 + 0.2021s_2 \geq 0\,, \\ g_2(\mathbf{s}) &= 5.88 + s_1 \geq 0\,, \\ g_3(\mathbf{s}) &= s_2 \geq 0\,. \end{aligned}$$

We solve this quadratic programming problem directly with the use of the Kuhn-Tucker conditions

$$\begin{aligned} 3 + s_1 - 0.1335\lambda_1 - \lambda_2 &= 0\,, \\ \sqrt{3} + s_2 - 0.2021\lambda_1 - \lambda_3 &= 0\,. \end{aligned}$$

A consideration of all possibilities for active constraints shows that the optimum is obtained when only g_1 is active, so that $\lambda_2 = \lambda_3 = 0$ and $\lambda_1 = 12.8$, $s_1 = -1.29$, $s_2 = 0.855$. The next design is

$$\mathbf{x}_1 = \begin{Bmatrix} 11.61 \\ 7.17 \end{Bmatrix} + \alpha \begin{Bmatrix} -1.29 \\ 0.855 \end{Bmatrix}\,,$$

where α is found by minimizing $\psi(\alpha)$ of Eq. (5.9.4). For the first iteration $\mu_j = |\lambda_j|$ so

$$\psi = 3(11.61 - 1.29\alpha) + \sqrt{3}(7.17 + 0.855\alpha) + 12.8 \left| 3 - \frac{18}{11.61 - 1.29\alpha} - \frac{6\sqrt{3}}{7.17 + 0.855\alpha} \right|\,.$$

By changing α systematically we find that ψ is a minimum near $\alpha = 2.2$, so that

$$\mathbf{x}_1 = (8.77, 9.05)^T, \quad f(\mathbf{x}_1) = 41.98, \quad g_1(\mathbf{x}_1) = -0.201 .$$

To update $\mathbf{A}$ we need $\Delta\mathbf{x}$ and $\Delta\mathbf{l}$. We have

$$\mathcal{L} = 3x_1 + \sqrt{3}x_2 - 12.8(3 - 18/x_1 + 6\sqrt{3}/x_2) ,$$

so that

$$\nabla_x \mathcal{L} = (3 - 230.4/x_1^2,\ \sqrt{3} - 133.0/x_2^2)^T ,$$

and

$$\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0 = \begin{Bmatrix} -2.84 \\ 1.88 \end{Bmatrix}, \quad \Delta\mathbf{l} = \nabla_x \mathcal{L}(\mathbf{x}_1) - \nabla_x \mathcal{L}(\mathbf{x}_0) = \begin{Bmatrix} -1.31 \\ 0.963 \end{Bmatrix} .$$

With $\mathbf{A}$ being the identity matrix we have $\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x} = 11.6$, $\Delta\mathbf{x}^T\Delta\mathbf{l} = 5.53$. Because $\Delta\mathbf{x}^T\Delta\mathbf{l} > 0.2\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}$ we can use Eq. (5.9.5) to update $\mathbf{A}$

$$A_{new} = I - \frac{\Delta\mathbf{x}\Delta\mathbf{x}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} + \frac{\Delta\mathbf{l}\Delta\mathbf{l}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} = \begin{bmatrix} 0.453 & 0.352 \\ 0.352 & 0.775 \end{bmatrix} .$$

For the second iteration

$$\begin{aligned} \phi(\mathbf{s}) &= 41.98 + 3s_1 + \sqrt{3}s_2 + 0.5(0.453s_1^2 + 0.775s_2^2 + 0.704s_1s_2) , \\ g_1(\mathbf{s}) &= -0.201 + 0.234s_1 + 0.127s_2 \geq 0 , \\ g_2(\mathbf{s}) &= 3.04 + s_1 \geq 0 , \\ g_3(\mathbf{s}) &= 1.88 + s_2 \geq 0 . \end{aligned}$$

We can again solve the quadratic programming directly with the use of the Kuhn-Tucker conditions

$$\begin{aligned} 3 + 0.453s_1 + 0.352s_2 - 0.234\lambda_1 - \lambda_2 &= 0 , \\ \sqrt{3} + 0.352s_1 + 0.775s_2 - 0.127\lambda_1 - \lambda_3 &= 0 . \end{aligned}$$

The solution is

$$\lambda_1 = 14.31, \quad \lambda_2 = \lambda_3 = 0, \quad s_1 = 1.059, \quad s_2 = -0.376 .$$

The one dimensional search seeks to minimize

$$\psi(\alpha) = f(\alpha) + \mu_1 g_1(\alpha) ,$$

where

$$\mu_1 = max(\lambda_1, \frac{1}{2}(|\lambda_1| + \mu_1^{old})) = 14.31 .$$

The one-dimensional search yields approximately $\alpha = 0.5$, so that

$$\mathbf{x}_2 = (9.30, 8.86)^T, \quad f(\mathbf{x}_2) = 43.25, \quad g_1(\mathbf{x}_2) = -0.108 ,$$

so that we have made good progress towards the optimum $\mathbf{x}^* = (9.46, 9.46)^T$. • • •

5.10 Exercises

1. Check the nature of the stationary points of the constrained problem

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = x_1^2 + 4x_2^2 + 9x_3^2 \\ \textbf{such that} \quad & x_1 + 2x_2 + 3x_3 \geq 30\,, \\ & x_2 x_3 \geq 2\,, \\ & x_3 \geq 4\,, \\ & x_1 x_2 \geq 0\,. \end{aligned}$$

2. For the problem

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = 3x_1^2 - 2x_1 - 5x_2^2 + 30x_2 \\ \textbf{such that} \quad & 2x_1 + 3x_2 \geq 8\,, \\ & 3x_1 + 2x_2 \leq 15\,, \\ & x_2 \leq 5\,. \end{aligned}$$

Check for a minimum at the following points: (a) (5/3, 5.00) (b) (1/3, 5.00) (c) (3.97,1.55).

3. Calculate the derivative of the solution of Example 5.1.2 with respect to a change in the allowable displacement. First use the Lagrange multiplier to obtain the derivative of the objective function, and then calculate the derivatives of the design variables and Lagrange multipliers and verify the derivative of the objective function. Finally, estimate from the derivatives of the solution how much we can change the allowable displacement without changing the set of active constraints.

4. Solve for the minimum of problem 1 using the gradient projection method from the point (17, 1/2, 4).

5. Complete two additional moves in Example 5.5.2.

6. Find a feasible usable direction for problem 1 at the point (17, 1/2, 4).

7. Use an exterior penalty function to solve Example 5.1.2.

8. Use an interior penalty function to solve Example 5.1.2.

9. Consider the design of a box of maximum volume such that the surface area is equal to S and there is one face with an area of $S/4$. Use the method of multipliers to solve this problem, employing three design variables.

10. Complete two more iterations in Example 5.9.1.

5.11 References

[1] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," Proceedings of IFAC Symposium on Computer Aided Design of Control Systems, Zurich, Switzerland, pp. 113-117, 1979.

[2] Sobieszczanski-Sobieski, J., "A Technique for Locating Function Roots and for Satisfying Equality Constraints in Optimization," NASA TM-104037, NASA LaRC, 1991.

[3] Wolfe, P.. "The Simplex Method for Quadratic Programming," Econometrica, 27 (3), pp. 382–398, 1959.

[4] Gill, P.E., Murray, W., and Wright, M.H., Practical Optimization, Academic Press, 1981.

[5] Dahlquist, G., and Bjorck, A., Numerical Methods, Prentice Hall, 1974.

[6] Sobieszczanski-Sobieski, J., Barthelemy, J.F., and Riley, K.M., "Sensitivity of Optimum Solutions of Problem Parameters", AIAA Journal, 20 (9), pp. 1291–1299, 1982.

[7] Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming—Part I: Linear Constraints", The Society for Industrial and Appl. Mech. Journal, 8 (1), pp. 181– 217, 1960.

[8] Abadie, J., and Carpentier, J., "Generalization of the Wolfe Reduced Gradient Method for Nonlinear Constraints", in: Optimization (R. Fletcher, ed.), pp. 37–49, Academic Press, 1969.

[9] Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming—Part II: Nonlinear Constraints", The Society for Industrial and Appl. Mech. Journal, 9 (4), pp. 514–532, 1961.

[10] Haug, E.J., and Arora, J.S., Applied Optimal Design: Mechanical and Structural Systems, John Wiley, New York, 1979.

[11] Zoutendijk, G., Methods of Feasible Directions, Elsevier, Amsterdam, 1960.

[12] Vanderplaats, G.N., "CONMIN—A Fortran Program for Constrained Function Minimization", NASA TM X-62282, 1973.

[13] Fiacco, V., and McCormick, G.P., Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley, New York, 1968.

[14] Haftka, R.T., and Starnes, J.H., Jr., "Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization", AIAA Journal, 14 (6), pp.718–724, 1976.

[15] Moe, J., "Penalty Function Methods in Optimum Structural Design—Theory and Applications", in: Optimum Structural Design (Gallagher and Zienkiewicz, eds.), pp. 143–177, John Wiley, 1973.

[16] Shin, D.K, Gürdal, Z., and Griffin, O. H. Jr., "A Penalty Approach for Nonlinear Optimization with Discrete Design Variables," Engineering Optimization, 16, pp. 29–42, 1990.

[17] Bertsekas, D.P., "Multiplier Methods: A Survey," Automatica, 12, pp. 133–145, 1976.

[18] Hestenes, M.R., "Multiplier and Gradient Methods," Journal of Optimization Theory and Applications, 4 (5), pp. 303–320, 1969.

[19] Fletcher, R., "An Ideal Penalty Function for Constrained Optimization," Journal of the Institute of Mathematics and its Applications, 15, pp.319–342, 1975.

[20] Powell, M.J.D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", Proceedings of the 1977 Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics, Vol. 630, pp. 144–157, Springer-Verlag, Berlin, 1978.

Aspects of The Optimization Process in Practice 6

Occasionally, a structural analyst will write a design program that includes the calculation of structural response as well as an implementation of a constrained optimization algorithm, such as those discussed in Chapter 5. More often, however, the analyst will have a structural analysis package, such as a finite-element program, as well as an optimization software package available to him. The task of the analyst is to combine the two so as to bring them to bear on the structural design problem that he wishes to solve.

Two major difficulties are associated with the process of interfacing a structural analysis package with an optimization program. The first is a programming difficulty. Optimization packages typically expect subroutines that evaluate the objective function and constraints. When the structural analysis program is large, or if the analyst does not have access to the source code of the program (a common situation), it is very difficult to transform the analysis package into a subroutine called by the optimization program.

The second serious problem is the high computational cost required for many applications. For many structural optimization problems the evaluation of objective function and constraints requires the execution of costly finite element analyses for displacements, stresses or other structural response quantities. The optimization process may require evaluating objective function and constraints hundreds or thousands of times. The cost of repeating the finite element analysis so many times is usually prohibitive.

Fortunately, there is an approach to interfacing an optimization program with an analysis program that solves both problems. This increasingly popular approach, called *sequential approximate optimization*, was suggested by Schmit and Farshi [1]. The computational cost problem is addressed by the use of approximate analyses during portions of the optimization process. The structural analysis package is first used to analyze an initial design, and then to generate information that allows the construction of constraint approximations. For example, when the number of design variables is small it is practical to analyze the structure at a number of points in the design space, and use the response at those points to construct a polynomial approximation to the response at other points. The optimization package is then

applied to the approximate problem represented by the polynomial approximation. Since the polynomial approximation is typically easy to program, it is straight-forward to interface it to the optimization package.

The simple approximations generated by repeated use of the analysis package are often referred to as low-cost *explicit approximations*, in contrast to the *implicit* dependence of the response on the structural design variables via a finite element solution. The polynomial approximation obtained by analyzing the structure at a number of design points is a *global approximation.* Obtaining such a global approximation can be quite expensive for a large number of design variables. For example, if we want to fit the structural response by a quadratic polynomial, we need to analyze the structure for at least $n(n+1)/2$ design points (typically many more to ensure a robust approximation), where n is the number of design variables. This will result in thousands of analyses when the number of design variables is larger than, say 40. Therefore, it is more common to use *local approximations* based on derivatives of the objective function and constraints with respect to the design variables. The simplest approach is to replace the objective function and constraints with *linear approximations* based on these derivatives. However, these approximations are useful only in a neighborhood of the design space. Therefore, it is necessary to impose limits, called *move limits*, on the magnitudes of changes in the design that are permitted while the approximate analysis is used.

Following an optimization based on approximate analysis and move limits, an exact analysis is performed at the design point obtained by the approximate optimization, and new derivatives are calculated so that a new approximation for objective function and constraints can be constructed. The process is repeated until convergence is achieved, typically measured by the magnitude of changes in the objective function or the degree of satisfaction of the optimality conditions (e.g., the Kuhn-Tucker conditions). Because each approximate optimization is only one cycle in the overall optimization process, it is usually possible to employ lax convergence criteria for these approximate problems, except for the last one. To distinguish them from the iterations inside approximate optimizations, each such optimization is referred to as a *cycle* rather than as an iteration.

When linear approximations are used, and the move limits are posed as linear inequalities, this process is called *sequential linear programming* (SLP), and was known for many years before Schmit and Farshi proposed the use of approximations for structural optimization. However, there is no need to limit the process to linear approximations, as long as the approximations are substantially cheaper to calculate than the exact analyses. For example, Schmit and Farshi demonstrated the use of inexpensive nonlinear approximations by using the reciprocal approximation, discussed in Section 6.1.

The use of sequential approximate optimization in the design process is the key step in interfacing a structural analysis program with an optimization program, and so it is the major topic discussed in this chapter. However, there are other aspects of the practical use of the optimization process in design that deserve consideration. For shape optimization problems, it is important to be able to modify the discretization of the structure (e.g., the finite-element model) as the design is changed. This requires

sophisticated mesh generators, and is discussed in Section 6.5. Other topics discussed in this chapter include optimization packages, and test problems that are often used to check on the performance of these packages. One important topic which is not discussed in this chapter is the calculation of the derivatives of the response of the structure needed for constructing the approximation. This topic requires a more detailed study and is the subject of Chapters 7 and 8.

The use of sequential approximate optimization is by no means universally accepted as the only way to deal with the optimization of complex structures. Many analysts prefer to use their judgement so as to produce a *design model* of the problem which employs a much coarser discretization than they would accept for the final analysis of the structure. They hope that the design trends revealed by optimizing the coarse model will hold for the more refined model. While this approach is quite legitimate, it will not be discussed here, because it requires a great deal of experience on the part of the analyst, and is highly problem dependent. As such it is very difficult to codify in a textbook.

6.1 Generic Approximations

The most commonly used approximations to objective functions and constraints are based on the value of the function and its derivatives at one or several points. Most of these approximations are applicable to any function, regardless of whether it describes structural response or not. For this reason we refer to such approximations as generic. Approximations that are specific to the form of analysis that is used to generate the function are dealt with in the next section. Generic approximations can be divided into local approximations, that are sufficiently accurate only in a limited region of the design space, and global approximations that attempt to approximate the function in the entire design space. Midrange approximations offer a compromise between the two.

6.1.1 *Local Approximations*

The simplest local approximation is the linear approximation based on the Taylor series. Given a function $g(\mathbf{x})$, the linear approximation $g_L(\mathbf{x})$ is

$$g_L(\mathbf{x}) = g(\mathbf{x}_0) + \sum_{i=1}^{n} (x_i - x_{0i}) \left(\frac{\partial g}{\partial x_i} \right)_{\mathbf{x}_0} . \tag{6.1.1}$$

For many applications the linear approximation is inaccurate even for design points $\mathbf{x}$ that are close to $\mathbf{x}_0$. Accuracy can be increased by retaining additional terms in the Taylor series expansion. This, however, requires the costly calculation of higher-order derivatives. A more attractive alternative is to find intervening variables that would make the approximated function behave more linearly. That is, define

$$y_i = y_i(\mathbf{x}) \qquad i = 1, \ldots, m , \tag{6.1.2}$$

where y_i are m functions of the design variables called intervening variables. The linear approximation, g_I, in terms of the intervening variables is

$$g_I(\mathbf{y}) = g(\mathbf{y}_0) + \sum_{i=1}^{m} (y_i - y_{0i}) \left(\frac{\partial g}{\partial y_i} \right)_{\mathbf{y}_0} , \tag{6.1.3}$$

where $y_{0i} = y_i(\mathbf{x}_0)$, and the derivatives of g with respect to the y_i's can be calculated from the derivatives with respect to the x_i's.

Example 6.1.1

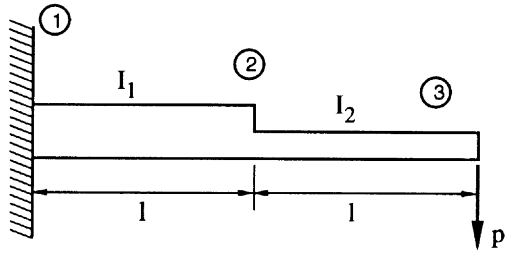


Figure 6.1.1 Beam example.

The beam shown in Fig. (6.1.1) has a rectangular cross section of width b_i and height h_i, $i = 1, 2$. The tip displacement is constrained not to exceed w_{all}; with elementary beam theory this constraint can be written as

$$g = w_{\text{all}} - \left(\frac{23}{6} \right) \frac{pl^3}{EI_1} - \left(\frac{5}{6} \right) \frac{pl^3}{EI_2} .$$

If the design variables are the width and height of each section, we can express g in terms of these design variables as

$$g = w_{\text{all}} - \frac{46pl^3}{Eb_1h_1^3} - \frac{10pl^3}{Eb_2h_2^3} .$$

This expression is a highly non-linear function of the design variables, but it can be linearized by using the intervening variables

$$y_1 = \frac{1}{I_1} = \frac{12}{b_1h_1^3} , \qquad \text{and} \qquad y_2 = \frac{1}{I_2} = \frac{12}{b_2h_2^3} .$$

The constraint function can then be written as a linear function

$$g = w_{\text{all}} - \left(\frac{23}{6} \right) \frac{pl^3}{E} y_1 - \left(\frac{5}{6} \right) \frac{pl^3}{E} y_2 .$$

• • •

The cases where intervening variables can exactly linearize the constraint are rather rare. Example (6.1.1) is typical of statically determinate structures where such linearization is often possible. However, as shown by Mills-Curran et al. [2], even in the case of statically indeterminate beam and frame structures, the reciprocals of moments of inertia are good intervening variables for displacement constraints.

In many applications the intervening variables are functions of a single design variable, that is

$$y_i = y_i(x_i) \qquad i = 1, \ldots, n \ . \tag{6.1.4}$$

In this case it is often convenient to write g_I, Eq. (6.1.3), in terms of the original variables

$$g_I(\mathbf{x}) = g(\mathbf{x}_0) + \sum_{i=1}^{n} \Big(y_i(x_i) - y_i(x_{0i})\Big) \left(\frac{\partial g}{\partial x_i} \Big/ \frac{dy_i}{dx_i}\right)_{\mathbf{x}_0} \ . \tag{6.1.5}$$

Note that while g_I is a linear function of $\mathbf{y}$ it is, in general, a nonlinear function of $\mathbf{x}$.

One of the more popular intervening variables is the reciprocal of x_i

$$y_i = \frac{1}{x_i} \ . \tag{6.1.6}$$

This popularity reflects the fact that many of the early structural optimization studies were performed on structures consisting of truss or plane-stress elements. The design variables in these studies were usually the cross-sectional areas of the truss elements and the thicknesses of the plane-stress elements. For statically determinate structures stress and displacements constraints are linear functions of the reciprocals of these design variables. For statically indeterminate structures, using the reciprocals of the design variables still proved to be a useful device in making the constraints more linear (see, for example, Storaasli and Sobieszczanski [3], and Noor and Lowder [4]). For the *reciprocal approximation* Eq. (6.1.5) becomes

$$g_R(\mathbf{x}) = g(\mathbf{x}_0) + \sum_{i=1}^{n} (x_i - x_{0i}) \frac{x_{0i}}{x_i} \left(\frac{\partial g}{\partial x_i}\right)_{\mathbf{x}_0} \ . \tag{6.1.7}$$

One of the attractive features of the reciprocal approximation, even for statically indeterminate structures, is that it preserves the property of scaling. That is, when the stiffness matrix is a homogeneous function of order h in the components of $\mathbf{x}$, the displacements are homogeneous functions of order $-h$ in the components of $\mathbf{x}$. For truss and membrane elements, $h = 1$ so that the displacements are homogeneous functions of the reciprocals of the design variables. If all the design variables are scaled by a factor, the displacement vector is scaled by the reciprocal of that factor. Therefore the reciprocal approximation is exact for scaling the design. Fuchs [5] has investigated the importance of the homogeneity property, and Fuchs and Haj Ali [6] have proposed a family of approximations that generalizes the reciprocal approximation to any order of homogeneity.

Another approximation, called the *conservative approximation* [7], is a hybrid form of the linear and reciprocal approximations which is more conservative than

either. It is particularly suitable for interior and extended interior penalty function methods (see Section 5.7) which do not tolerate well constraint violations. To obtain the conservative approximation we start by subtracting the reciprocal approximation from the linear approximation

$$g_L(\mathbf{x}) - g_R(\mathbf{x}) = \sum_{i=1}^{n} \frac{(x_i - x_{0i})^2}{x_i} \left(\frac{\partial g}{\partial x_i}\right)_{\mathbf{x}_0} . \tag{6.1.8}$$

The sign of each term in the sum is determined by the sign of the ratio $(\partial g/\partial x_i)/x_i$ which is also the sign of the product $x_i(\partial g/\partial x_i)$. Contributions from design variables for which this product is negative make the reciprocal approximation larger (more positive) than the linear approximation, and vice versa. Since the constraint is expressed as $g(\mathbf{x}) \geq 0$, a more positive approximation is less conservative. The conservative approximation, g_C, is, therefore, created by selecting for each design variable the smaller (less positive) contribution

$$g_C(\mathbf{x}) = g(\mathbf{x}_0) + \sum_{i=1}^{n} G_i(x_i - x_{0i}) \left(\frac{\partial g}{\partial x_i}\right)_{\mathbf{x}_0} , \tag{6.1.9}$$

where

$$G_i = \begin{cases} 1 & \text{if } x_{0i}(\partial g/\partial x_i) \leq 0, \\ x_{0i}/x_i & \text{otherwise.} \end{cases} \tag{6.1.10}$$

Note that $G_i = 1$ corresponds to a linear approximation, and $G_i = x_{0i}/x_i$ corresponds to a reciprocal approximation in x_i.

The conservative approximation is not the only hybrid linear-reciprocal approximation possible. Sometimes physical considerations may dictate the use of linear approximation for some variables and the reciprocal for others, (see Haftka and Shore [8], and Prasad [9]). The conservative approximation, however, has the advantage of being concave (Exercise 1). If all the constraints are approximated by the conservative approximation, the feasible domain of the approximate optimization problem is convex (see Section 5.1.2). If we also approximate the objective function by a convex function, the approximate optimization problem is convex. Convex problems are guaranteed to have only a single optimum, and they are amenable to treatment by dual methods (see Section 9.2.2). In fact, a convex approximation $f_C(\mathbf{x})$ to the objective function, $f(\mathbf{x})$, is obtained by reversing the process for obtaining the conservative concave approximation. That is (Exercise 1),

$$f_C(\mathbf{x}) = f(\mathbf{x}_0) + \sum_{i=1}^{n} F_i(x_i - x_{0i}) \left(\frac{\partial f}{\partial x_i}\right)_{\mathbf{x}_0} , \tag{6.1.11}$$

where

$$F_i = \begin{cases} x_{0i}/x_i & \text{if } x_{0i}(\partial f/\partial x_i) \leq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{6.1.12}$$

This process of using the conservative approximation for the constraints and the convex approximation for the objective function has been introduced by Braibant and

Fleury [10], and is known as convex linearization. In many papers and textbooks, the constraints are posed as $g(\mathbf{x}) \leq 0$ rather than $g(\mathbf{x}) \geq 0$. In this case, the conservative approximation is convex rather than concave (that is we use the form of Eqs. (6.1.11) and (6.1.12) also for the constraints). There are other conservative approximations (for example, see Prasad [11] or Woo [12]), but it is important to note that the one presented here, as well as the others, are not guaranteed to be conservative in an absolute sense (that is, we do not know that the approximation is more conservative than the exact constraint, $g_C(\mathbf{x}) \leq g(\mathbf{x})$). The approximation presented here is only more conservative than either the linear and reciprocal approximations.

Higher order approximations are also used occasionally. For example, the quadratic approximation, g_Q is obtained by including the quadratic terms in the Taylor series expansion

$$g_Q(\mathbf{x}) = g(\mathbf{x}_0) + \sum_{i=1}^{n}(x_i - x_{0i})\left(\frac{\partial g}{\partial x_i}\right)_{\mathbf{x}_0} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_{0i})(x_j - x_{0j})\left(\frac{\partial^2 g}{\partial x_i \partial x_j}\right)_{\mathbf{x}_0} . \tag{6.1.13}$$

The reciprocal quadratic approximation g_{QR} is obtained by using the quadratic approximation in terms of the reciprocal design variables (Exercise 2),

$$\begin{aligned} g_{QR}(\mathbf{x}_0) = g(\mathbf{x}_0) &+ \sum_{i=1}^{n}\left(\frac{x_{0i}}{x_i}\right)\left(2 - \frac{x_{0i}}{x_i}\right)(x_i - x_{0i})\left(\frac{\partial g}{\partial x_i}\right)_{\mathbf{x}_0} \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{x_{0i}}{x_i}\right)\left(\frac{x_{0j}}{x_j}\right)(x_i - x_{0i})(x_j - x_{0j})\left(\frac{\partial^2 g}{\partial x_i \partial x_j}\right)_{\mathbf{x}_0} . \end{aligned} \tag{6.1.14}$$

Example 6.1.2

Comparison of various approximations is demonstrated through the use of a simple

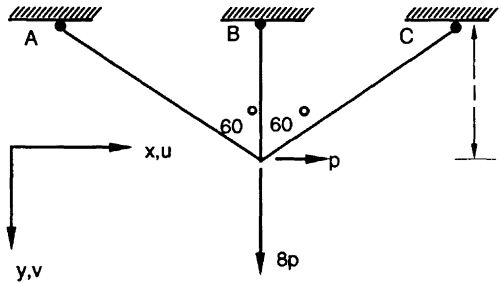


Figure 6.1.2 Three bar truss.

three bar truss shown in Figure 6.1.2. The horizontal force p can act either to the right (as shown) or to the left. The truss is designed subject to stress and displacement

constraints with the design variables being the cross-sectional areas A_A, A_B, and A_C. Because of the symmetry of the truss and the arbitrary direction of the horizontal load we must have $A_A = A_C$. We examine the approximations to the constraint on the stress in member C, which requires that stress to be less than σ_0 both in tension and compression.

The stresses in the three members can be expressed in terms of the displacement components at the tip of the truss as

$$\sigma_A = E(v + \sqrt{3}u)/4l, \qquad \sigma_B = Ev/l, \qquad \text{and} \qquad \sigma_C = E(v - \sqrt{3}u)/4l \ .$$

From the horizontal equation of equilibrium

$$\frac{\sqrt{3}}{2} A_A(\sigma_A - \sigma_C) = p, \qquad \text{or} \quad \frac{3EA_A}{4l} u = p \ .$$

Similarly, from the vertical equation of equilibrium

$$\frac{1}{2} A_A(\sigma_A + \sigma_C) + A_B\sigma_B = 8p, \qquad \text{or} \quad \frac{Ev}{l}\left(A_B + \frac{A_A}{4}\right) = 8p \ ,$$

so that

$$u = 4pl/3EA_A, \qquad v = 8pl/E(A_B + 0.25A_A) \ ,$$

and

$$\sigma_C = p\left(-\frac{\sqrt{3}}{3A_A} + \frac{2}{A_B + 0.25A_A}\right) \ .$$

Assuming that member C is in tension, we may write the constraint function as

$$g = 1 - \frac{\sigma_C}{\sigma_0} = 1 - \frac{p}{\sigma_0}\left(-\frac{\sqrt{3}}{3A_A} + \frac{2}{A_B + 0.25A_A}\right) \ .$$

We now define normalized design variables

$$x_1 = A_A\sigma_0/p, \qquad x_2 = A_B\sigma_0/p \ ,$$

so that

$$g = 1 + \frac{\sqrt{3}}{3x_1} - \frac{2}{x_2 + 0.25x_1} \ .$$

We approximate g about the point $\mathbf{x}_0^T = (1, 1)$. The first derivatives are

$$\frac{\partial g}{\partial x_1} = \left(-\frac{\sqrt{3}}{3x_1^2} + \frac{0.5}{(x_2 + 0.25x_1)^2}\right)_{\mathbf{x}_0} = -0.2574 \ ,$$

$$\frac{\partial g}{\partial x_2} = \frac{2}{(x_2 + 0.25x_1)^2}\Big|_{\mathbf{x}_0} = 1.28 \ .$$

and the second derivatives are

$$\frac{\partial^2 g}{\partial x_1^2} = \left(\frac{2\sqrt{3}}{3x_1^3} - \frac{0.25}{(x_2 + 0.25x_1)^3} \right)_{\mathbf{x}_0} = 1.0267 ,$$

$$\frac{\partial^2 g}{\partial x_1 x_2} = -\frac{1}{(x_2 + 0.25x_1)^3}\Big|_{\mathbf{x}_0} = -0.512 ,$$

$$\frac{\partial^2 g}{\partial x_2^2} = -\frac{4}{(x_2 + 0.25x_1)^3}\Big|_{\mathbf{x}_0} = -2.048 .$$

Using these derivatives and $g(\mathbf{x}_0) = -0.0227$ we can construct the following approximations

$$g_L = -0.0227 - 0.2574(x_1 - 1) + 1.28(x_2 - 1) ,$$

$$g_R = -0.0227 - 0.2574\left(1 - \frac{1}{x_1}\right) + 1.28\left(1 - \frac{1}{x_2}\right) = 1 + .2574/x_1 - 1.28/x_2 ,$$

$$g_C = -0.0227 - 0.2574(x_1 - 1) + 1.28\left(1 - \frac{1}{x_2}\right) ,$$

$$g_Q = g_L + 0.5134(x_1 - 1)^2 - 0.512(x_1 - 1)(x_2 - 1) - 1.024(x_2 - 1)^2 ,$$

$$g_{QR} = -0.0227 - 0.2574\left(2 - \frac{1}{x_1}\right)\left(1 - \frac{1}{x_1}\right) + 1.28\left(2 - \frac{1}{x_2}\right)\left(1 - \frac{1}{x_2}\right)$$
$$+ 0.5134\left(1 - \frac{1}{x_1}\right)^2 - 0.512\left(1 - \frac{1}{x_1}\right)\left(1 - \frac{1}{x_2}\right) - 1.024\left(1 - \frac{1}{x_2}\right)^2 .$$

All of these approximations have the correct value and correct derivatives at $\mathbf{x}_0 = (1,1)^T$. The two quadratic approximations also have the correct second derivatives at that point. The reciprocal approximations tend to one as the design variables tend to infinity. This corresponds to the stress in member C tending to zero as the cross-sectional areas tend to infinity. This correct physical behavior is not shared by the other approximations. Table 6.1.1 compares the predictions of the five approximations to the exact values when x_1 and x_2 vary between 0.75 and 1.25.

Table 6.1.1

x_1	x_2	g	g_L	g_R	g_C	g_Q	g_{QR}
0.75	0.75	-0.3635	-0.2783	-0.3635	-0.3850	-0.3422	-0.3635
1.00	0.75	-0.4227	-0.3426	-0.4493	-0.4493	-0.4066	-0.4209
1.25	0.75	-0.4205	-0.4070	-0.5008	-0.5137	-0.4070	-0.4280
0.75	1.00	0.0856	0.0417	0.0631	0.0417	0.0738	0.0915
1.25	1.00	-0.0619	-0.0870	-0.0741	-0.0871	-0.0549	-0.0639
0.75	1.25	0.3786	0.3617	0.3191	0.2977	0.3617	0.3919
1.00	1.25	0.2440	0.2974	0.2334	0.2334	0.2334	0.2435
1.25	1.25	0.1819	0.2330	0.1819	0.1690	0.1691	0.1819

The Table shows that the approximations based on reciprocal variables are more accurate than the approximations based on the actual variables, and in particular,

they are exact when the two variables are scaled by the same factor (that is $\mathbf{x}$ is replaced by $\alpha\mathbf{x}$ where α is a scalar). The quadratic approximations are substantially more accurate than the three first-order approximations. The conservative approximation is not guaranteed to be more conservative than the second-order approximations, but usually, as in this example, it is. We see, however, that the price of this extra conservativeness is that it is the least accurate approximation.

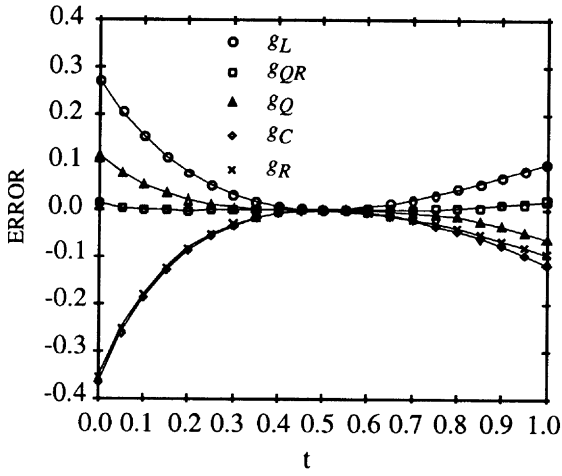


Figure 6.1.3 Comparison of constraint approximation errors.

The constraint approximations can also be used to check for errors in the derivatives used to construct them. This is done by calculating the exact constraint along a line in design space and plotting the error in the approximation along that line. A first order approximation must have a zero slope for the error curve at the nominal design, while a second-order approximation must also have zero curvature there. For example, let us compare the various approximations along the line

$$x_1 = 1.25 - 0.5t, \quad x_2 = 0.5 + 1.0t, \qquad 0 \le t \le 1 \ ,$$

where $t = 0.5$ represents the nominal design. Figure 6.1.3 shows the error as a function of t. It is seen that the first-order approximations indeed have zero slope at $t = 0.5$, while the second-order approximations also have zero curvature there. For this example, the reciprocal approximation is quite conservative, so that the conservative approximation is almost identical to it.• • •

The approximations covered so far are obtained by algebraically manipulating the constraint functions. In an effort to improve the quality of the approximations recent research efforts have concentrated on the extension of the concept of intermediate design variables to the concept of intermediate response quantities. The concept was introduced by Schmit and Miura [13] in 1976, but it was not applied until about ten

years later (e.g., [14]). The approach seeks intermediate response quantities that are well approximated linearly. If the response quantities appearing in the constraint can be calculated inexpensively from the intermediate response, than we can have a nonlinear inexpensive and accurate approximation.

One of the most successful intermediate response approximation was proposed for stress constraints in structural design by Vanderplaats and coworkers (e.g., [15–17]). Vanderplaats argued that an approximation for member forces will be more accurate than the corresponding approximation for member stresses. This is expected because member forces change more slowly than member stresses when cross-sectional areas are changed. In particular, for a statically determinate truss, force in each of the members is constant, while member stresses are inversely proportional to member areas. This motivates the use of the member forces as intermediate response quantities.

Consider, for example, a typical stress constraint for a truss member of the form

$$g_i = 1 - \frac{\sigma_i}{\sigma_{\text{all}}} \geq 0 \; . \tag{6.1.15}$$

A common approximation for member stresses uses the reciprocal design variables, $x_i = 1/A_i$, where A_i is the cross-sectional area of the ith member. Using a linear approximation for the member forces, and then dividing by the cross-sectional area to obtain an approximation to the stress, as suggested by Vanderplaats, we obtain a constraint of the form

$$g_{LF_i} = A_i - \frac{[F_i(\mathbf{A}_0) + \nabla^T F_i(\mathbf{A}_0)(\mathbf{A} - \mathbf{A}_0)]}{\sigma_{\text{all}}} \geq 0 \; . \tag{6.1.16}$$

This is linear in the cross-sectional area design variables. Note that for a statically determinate truss, where the gradient of the member forces with respect to the cross-sectional areas is zero, the approximation of Eq. (6.1.16) is a constant. Equation (6.1.16) has the dimension of area, and it should be nondimensionalized by dividing it by a reference area. A comparison of the performance of this linear force approximation with other approximations is given in Section 6.4.

6.1.2 *Global and Midrange Approximations*

The most common global approximation is the response surface approach. With this approach the function is sampled at a number of points, and then an analytical expression called the response surface (typically a polynomial) is fitted to the data. Construction of response surface often relies heavily on the theory of experiments [18] and is an iterative process that begins with the assumption of the analytical form of the response surface, for example, a quadratic polynomial. The approximation contains a number of unknown parameters (such as polynomial coefficients) that must be adjusted to match the function to be approximated. To do so, analyses are performed at a number of carefully selected design points, and a least square solution is typically used to extract the parameter values from the analysis results. Then the approximate model (the response surface) is used to predict the function at a number of selected test points, and statistical measures are used to assess the goodness-of-fit,

or the accuracy of the response surface. If the fit is not satisfactory, the process is restarted, and further experiments are made, or the postulated model is improved by removing and/or adding terms.

Response surface techniques have not been used extensively in structural optimization (see Barthelemy and Haftka [19] for applications). This may be due to the fact that the technique is practical only for problems with a small number of design variables (less than 20). The number of analyses required to construct the response surface increases dramatically with the number of design variables.

Example 6.1.3

To demonstrate the use of response surfaces we fit a linear response surface to the stress constraint of Example 6.1.2

$$g = 1 + \frac{\sqrt{3}}{3x_1} - \frac{2}{(x_2 + 0.25x_1)}.$$

The response surface is assumed to be a linear polynomial

$$g_{rs} = a + bx_1 + cx_2. \qquad (a)$$

We assume that the design space is

$$0 \leq x_1,\ x_2 \leq 2.$$

To find a, b, and c we need to evaluate g at 3 or more points. For robustness we use more points, so we select the following 4 points:

$$\mathbf{x}_1^T = (0.5, 0.5), \quad \mathbf{x}_2^T = (1.5, 0.5), \quad \mathbf{x}_3^T = (0.5, 1.5), \quad \mathbf{x}_4^T = (1.5, 1.5).$$

Substituting each of these points into Eq. (a) we get 4 equations

$$\begin{bmatrix} 1 & 0.5 & 0.5 \\ 1 & 1.5 & 0.5 \\ 1 & 0.5 & 1.5 \\ 1 & 1.5 & 1.5 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{Bmatrix} -1.0453 \\ -0.9008 \\ 0.9239 \\ 0.3182 \end{Bmatrix}.$$

To get a least-square solution of these 4 equations in 3 unknowns, we multiply both sides by the transpose of the coefficient matrix and solve the resulting 3×3 system. We obtain $a = -1.5395$, $b = -0.2306$, $c = 1.5941$, or

$$g_{rs} = -1.5395 - 0.2306x_1 + 1.5941x_2.$$

We compare this with the linear approximation about (1,1) that we found in Example 6.1.2

$$g_L = -0.0227 - 0.2574(x_1 - 1) + 1.28(x_2 - 1).$$

As expected, g_L is more accurate near (1,1), and g_{rs} further away. For example at (0.75,0.75) we get $g = -0.3635$, $g_L = -0.2783$, $g_{rs} = -0.5169$. While at (0.5,0.5) we get $g = -1.0453$, $g_L = -0.5340$, $g_{rs} = -0.8578$.• • •

In response surface techniques the design space is sampled ahead of the optimization process. However, because the optimization process requires the calculation of constraints and their derivatives at more than one point, it makes sense to use the information from previous calculations to construct wide ranging approximations rather than approximations based on information at a single point. This leads to the concept of multipoint approximations that qualify for the label *midrange approximations*. Haftka et al. [20] examined approximations based on two and three points. Their experience was that the approximation worked well when it represented interpolation (for example, at points inside the triangle formed by three data points in a three-point approximation), but gave only marginal improvement in accuracy for extrapolation.

A two-point approximation that shows more promise was proposed by Fadel et al. [21]. The approximation is a linear approximation in the variables $y_i = x_i^{p_i}$, where the exponentials are selected to match the data. We start by constructing a linear approximation in y_i at the first point $\mathbf{x}_0$. The approximation may be written in terms of the original variables as

$$g_{tp} = g(\mathbf{x}_0) + \sum_{i=1}^{n} \left[\left(\frac{x_i}{x_{0i}} \right)^{p_i} - 1 \right] \left(\frac{x_{0i}}{p_i} \right) \left(\frac{\partial g}{\partial x_i} \right)_{\mathbf{x}_0} . \tag{6.1.17}$$

Then the exponentials p_i are found from the condition that the derivatives of g match those of g_{tp} at a second point, $\mathbf{x}_1$. It is easy to show that this leads to

$$p_i = 1 + \frac{log \left\{ \left(\frac{\partial g}{\partial x_i} \right)_{\mathbf{x}_1} \Big/ \left(\frac{\partial g}{\partial x_i} \right)_{\mathbf{x}_0} \right\}}{log(x_{1i}/x_{0i})} . \tag{6.1.18}$$

When p_i is larger in magnitude than 1 it is set to $sign(p_i)$ so as to avoid large exponents. Special provisions need to be made when the ratios in the numerator or denominator in Eq. (6.1.18) are negative or if p_i is zero. In the first case p_i is taken to be 1, while in the second case it can be shown by the use of Taylor series expansion that

$$\lim_{p_i \to 0} \frac{\left[\left(\frac{x_i}{x_{0i}} \right)^{p_i} - 1 \right]}{p_i} = log \left(\frac{x_i}{x_{0i}} \right) . \tag{6.1.19}$$

Another midrange approximation is the scaling or local-global approximation [22]. It is intended to improve a global approximation, available from a response surface approach or from a simpler model of the problem, by injecting some local information into it. The simplest approach for doing that is to use a scale factor based on the value of the function at a point $\mathbf{x}_0$. That is, the scale factor s_c is given as

$$s_c(\mathbf{x}) = g(\mathbf{x})/g_G(\mathbf{x}) , \tag{6.1.20}$$

where g_G is the global approximation. Then the scaled global approximation, g_{s0}, is given as

$$g_{s0}(\mathbf{x}) = s_c(\mathbf{x}_0) g_G(\mathbf{x}) . \tag{6.1.21}$$

An improvement on this scale factor can be obtained by using the derivative of g to construct a linear scale factor s_{c1} given as

$$s_{c1} = s_{c0} + \sum_{i=1}^{n} (x_i - x_{0i}) \left(\frac{\partial s_c}{\partial x_i} \right)_{\mathbf{x}_0} . \tag{6.1.22}$$

where the derivative of the scale factor is

$$\left(\frac{\partial s_c}{\partial x_i} \right) = s_c \left(\frac{1}{g} \frac{\partial g}{\partial x_i} - \frac{1}{g_G} \frac{\partial g_G}{\partial x_i} \right) . \tag{6.1.23}$$

The local-global approximation was applied by Chang et al. [23] for approximating displacements, stresses and frequencies of a supersonic wing structure obtained by a finite element model. The global approximation used was a plate model of the wing.

6.2 Fast Reanalysis Techniques

Fast reanalysis techniques take advantage of the computations performed at one design point to reduce the computational cost of the analysis at another design point. They are often approximate in nature, working well when the latter design point is close to the former. In this section we assume that the exact structural response is available at a design point $\mathbf{x}_0$, and that we want to calculate the effect of a small to moderate perturbation $\Delta\mathbf{x}$ on the response. We will denote the structural properties and response at $\mathbf{x}_0$ by a subscript zero, and the perturbations in properties and response by Δ. For example, $\mathbf{u}_0 = \mathbf{u}(\mathbf{x}_0)$ denotes the displacement field for the nominal design, and $\mathbf{u}_0 + \Delta\mathbf{u} = \mathbf{u}(x_0 + \Delta\mathbf{x})$ denotes the displacement field for the perturbed design.

6.2.1 *Linear Static Response*

The discrete equations of equilibrium for linear static response (obtained, for example, from a finite element analysis) at a design point $\mathbf{x}_0$ are

$$\mathbf{K}_0\mathbf{u}_0 = \mathbf{f}_0 , \tag{6.2.1}$$

where $\mathbf{K}_0$, $\mathbf{u}_0$ and $\mathbf{f}_0$ are the stiffness matrix, the displacement vector and the load vector at $\mathbf{x}_0$, respectively. Consider now a change $\Delta\mathbf{x}$ in the design which results in a change $\Delta\mathbf{K}$ in the stiffness matrix, and $\Delta\mathbf{f}$ in the load vector. The equations of equilibrium at $\mathbf{x}_0 + \Delta\mathbf{x}$ are

$$(\mathbf{K}_0 + \Delta\mathbf{K})(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{f}_0 + \Delta\mathbf{f} . \tag{6.2.2}$$

Subtracting Eq. (6.2.1) from Eq. (6.2.2) we obtain

$$(\mathbf{K}_0 + \Delta\mathbf{K})\Delta\mathbf{u} = \Delta\mathbf{f} - \Delta\mathbf{K}\mathbf{u}_0 , \tag{6.2.3}$$

and we can obtain a first approximation $\Delta\mathbf{u}_1$ to $\Delta\mathbf{u}$ by neglecting the $\Delta\mathbf{K}\Delta\mathbf{u}$ term

$$\mathbf{K}_0\Delta\mathbf{u}_1 = \Delta\mathbf{f} - \Delta\mathbf{K}\mathbf{u}_0 \,. \tag{6.2.4}$$

This approximation will be quite good when $\Delta\mathbf{x}$ is small in magnitude. Furthermore, usually we have $\mathbf{K}_0$ factored or inverted in the solution for $\mathbf{u}_0$. Therefore, it is relatively inexpensive to solve Eq. (6.2.4). When $\Delta\mathbf{K}$ is a linear function of $\mathbf{x}$ the approximation is in fact identical with the linear approximation of $\mathbf{u}$ based on the Taylor series. We can further improve the approximation by repeating the same process to obtain higher-order approximations to $\Delta\mathbf{u}$. Subtracting Eq. (6.2.4) from Eq. (6.2.3) we get

$$(\mathbf{K}_0 + \Delta\mathbf{K})(\Delta\mathbf{u} - \Delta\mathbf{u}_1) = -\Delta\mathbf{K}\Delta\mathbf{u}_1 \,, \tag{6.2.5}$$

and again we can neglect the $\Delta\mathbf{K}(\Delta\mathbf{u} - \Delta\mathbf{u}_1)$ on the left hand side of the equation to get and approximation $\Delta\mathbf{u}_2$ to $\Delta\mathbf{u} - \Delta\mathbf{u}_1$ by solving

$$\mathbf{K}_0\Delta\mathbf{u}_2 = -\Delta\mathbf{K}\Delta\mathbf{u}_1 \,. \tag{6.2.6}$$

The process can be continued indefinitely to obtain

$$\Delta\mathbf{u} = \sum_{i=1}^{\infty} \Delta\mathbf{u}_i \,, \tag{6.2.7}$$

where the terms $\Delta\mathbf{u}_i$ in the series are obtained through the iterative process of solving

$$\mathbf{K}_0\Delta\mathbf{u}_i = -\Delta\mathbf{K}\Delta\mathbf{u}_{i-1} \,. \tag{6.2.8}$$

Of course, the series is not guaranteed to converge, especially when $\Delta\mathbf{x}$ is not small.

Another approach for improving on $\Delta\mathbf{u}_1$ was suggested by Kirsch and Taye [24]. Their idea is that changes in the structure can be divided into overall scaling and redistribution of material. That is, we write the perturbed stiffness matrix as

$$\mathbf{K}_0 + \Delta\mathbf{K} = s\mathbf{K}_0 + \Delta\mathbf{K}_s \,, \tag{6.2.9}$$

where s is a scaling factor. Overall scaling can be dealt with in a simple manner, so that we need to analyse only the redistribution part. We choose s so as to minimize $\Delta\mathbf{K}_s$. That is, s is chosen so that $s\mathbf{K}_0$ is as close as possible to $\mathbf{K} + \Delta\mathbf{K}$. Kirsch and Taye suggested minimizing the sum of the squares of the elements of $\Delta\mathbf{K}_s$. Then it can be shown (Exercise 7), that s is given as

$$s = 1 + \frac{\sum_{i,j} k_{0ij}\Delta k_{ij}}{\sum_{i,j} k_{0ij}^2} \,. \tag{6.2.10}$$

Now we consider our nominal design to be the one with the matrix $s\mathbf{K}_0$ instead of $\mathbf{K}_0$. For this design the displacement field is

$$\mathbf{u}_s = (1/s)\mathbf{u}_0 \,. \tag{6.2.11}$$

We consider only the case where there is no change in the force, $\Delta\mathbf{f} = 0$. Then Eq. (6.2.4) for this scaled design is

$$s\mathbf{K}_0\Delta\mathbf{u}_{s1} = -\Delta\mathbf{K}_s\mathbf{u}_s = -[\Delta\mathbf{K} - (s-1)\mathbf{K}_0]\mathbf{u}_0/s\,, \tag{6.2.12}$$

where we used Eq. (6.2.9). Comparing this Equation to Eq. (6.2.4) we get

$$\Delta\mathbf{u}_{s1} = \frac{1}{s^2}\Delta\mathbf{u}_1 + \frac{s-1}{s^2}\mathbf{u}_0\,.$$

The total change in $\mathbf{u}$ predicted by this approach, $\Delta\mathbf{u}$, is

$$\Delta\mathbf{u}_s = \mathbf{u}_s - \mathbf{u}_0 + \Delta\mathbf{u}_{s1} = (\frac{1}{s} - 1)\mathbf{u}_0 + \frac{1}{s^2}\Delta\mathbf{u}_1 + \frac{s-1}{s^2}\mathbf{u}_0 = \frac{1}{s^2}\Delta\mathbf{u}_1 - \frac{(1-s)^2}{s^2}\mathbf{u}_0\,. \tag{6.2.13}$$

Example 6.2.1

Apply a first term correction, without and with scaling, to approximate the stress constraint in member C of Example (6.1.2) when the area of member B is increased by 25 percent ($x_1 = 1$, and x_2 is increased from 1 to 1.25, in terms of the nondimensional areas defined in Example 6.1.2).

The stiffness matrix for the three-bar truss is easily verified to be

$$\mathbf{K} = \frac{E}{l}\begin{bmatrix} 0.75A_A & 0 \\ 0 & A_B + 0.25A_B \end{bmatrix} = \frac{Ep}{l\sigma_0}\begin{bmatrix} 0.75x_1 & 0 \\ 0 & (x_2 + 0.25x_1) \end{bmatrix}\,,$$

so that

$$\mathbf{K}_0 = \frac{Ep}{l\sigma_0}\begin{bmatrix} 0.75 & 0 \\ 0 & 1.25 \end{bmatrix}\,, \qquad \Delta\mathbf{K}_0 = \frac{Ep}{l\sigma_0}\begin{bmatrix} 0 & 0 \\ 0 & 0.25 \end{bmatrix}\,.$$

Also, from Example (6.1.2) we have

$$\mathbf{u}_0 = \frac{pl}{E}\begin{Bmatrix} 4/3A_A \\ 8/(A_B + 0.25A_A) \end{Bmatrix} = \frac{l\sigma_0}{E}\begin{Bmatrix} 1.333 \\ 6.400 \end{Bmatrix}\,.$$

With $\Delta\mathbf{f} = 0$, Eq.(6.2.4) yields

$$\Delta\mathbf{u}_1 = -\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{u}_0 = \frac{l\sigma_0}{E}\begin{Bmatrix} 0 \\ -1.28 \end{Bmatrix}\,.$$

From Example (6.1.2) we also had

$$\sigma_C = E(v - \sqrt{3}u)/4l, \qquad \text{and} \quad g = 1 - \sigma_C/\sigma_0\,,$$

so that

$$\Delta g = -\frac{E}{4l\sigma_0}(\Delta v - \sqrt{3}\Delta u)\,. \tag{a}$$

Substituting the components of $\Delta\mathbf{u}_1$ we get

$$\Delta g = -\frac{E}{4l\sigma_0}\frac{l\sigma_0}{E}(-1.28) = 0.32\,.$$

Since for the nominal design $g = -0.0227$, for the perturbed design g is predicted to be

$$g \approx g_0 + \Delta g = -0.0227 + 0.32 = 0.2973\,,$$

which, as expected, is the same as the linear approximation (see Table 6.1.1). For the scaled approximation we use Eq. (6.2.10), to get

$$s = 1 + \frac{0.25 \times 1.25}{0.75^2 + 1.25^2} = 1.147\,.$$

Equation (6.2.13) becomes

$$\Delta\mathbf{u}_s = \frac{l\sigma_0}{1.147^2 E}\begin{Bmatrix} 0 \\ -1.28 \end{Bmatrix} - \frac{0.147^2 l\sigma_0}{1.147^2 E}\begin{Bmatrix} 1.333 \\ 6.400 \end{Bmatrix} = \frac{l\sigma_0}{E}\begin{Bmatrix} 0.0218 \\ -1.0780 \end{Bmatrix}.$$

Substituting into Eq. (a) we get

$$\Delta g_s = -0.25(-1.078 - \sqrt{3} \times 0.0218) = 0.2789\,,$$

and

$$g_s = g_0 + \Delta g_s = -0.0227 + 0.2789 = 0.2562\,.$$

This approximation is substantially closer to the exact result (see Table 6.1.1) of $g = 0.2440$.• • •

It is well known (e.g., Haley [25]) that when the matrix of a system of equations is modified by adding a matrix of low rank it is relatively inexpensive to find the effect on the solution of the system. The computational effort is roughly equal to finding r solutions to the original system, where r is the rank of the modification matrix. When r is small, and the order of the system of equations is large, finding r solutions of the original system is much cheaper than a new factorization of the modified system.

This situation often occurs when we modify a small part of a structure. For example, when the stiffness of a single truss member is modified, the modification matrix is of rank one, and the solution can be found by a single solution of the original problem. Furthermore, it can be shown [25] that once this single solution was found, the exact solution is available for an arbitrary magnitude of the modification. Fuchs and Steinberg [26] showed that this single solution is the same needed for obtaining the derivative of the displacement with respect to the change in stiffness. Thus they were able to derive an approximation to the displacement field which is exact if a single truss member is modified. Similarly, Holnicki-Szulc [27] has developed a method, based on virtual distortions, which permits arbitrary modifications of r members of a truss at the cost of r displacement solutions for the original truss. These approaches are particularly useful for optimization, because once the displacement solutions have been obtained, the truss elements can be modified again and again with very little additional computational cost. For finite elements with higher-rank stiffness matrices, the same approach is still applicable, but the advantages tend to be diminished.

6.2.2 *Eigenvalue Problems*

Vibration or buckling response is typically modeled as a symmetric eigenvalue problem. At the nominal design the vibration eigenproblem may be written as

$$\mathbf{K}_0\mathbf{u}_0 - \mu_0\mathbf{M}_0\mathbf{u}_0 = 0 \ , \tag{6.2.14}$$

where $\mathbf{K}_0$ and $\mathbf{M}_0$ are the stiffness and mass matrices, respectively, and μ_0 and $\mathbf{u}_0$ are the eigenvalue (square of frequency) and eigenvector (vibration mode), respectively, all evaluated at a nominal design point $\mathbf{x}_0$. When μ_0 is a nonrepeated eigenvalue, the effect of perturbing the design can be easily estimated. Rewriting the eigenvalue problem at $\mathbf{x}_0 + \Delta\mathbf{x}$ we have

$$(\mathbf{K}_0 + \Delta\mathbf{K})(\mathbf{u}_0 + \Delta\mathbf{u}) - (\mu_0 + \Delta\mu)(\mathbf{M}_0 + \Delta\mathbf{M})(\mathbf{u}_0 + \Delta\mathbf{u}) = 0 \ . \tag{6.2.15}$$

We subtract Eq. (6.2.14) from Eq. (6.2.15) and neglect quadratic and cubic terms in the perturbation such as $\Delta\mathbf{K}\Delta\mathbf{u}$ to get

$$(\mathbf{K}_0 - \mu_0\mathbf{M}_0)\Delta\mathbf{u} + (\Delta\mathbf{K} - \mu_0\Delta\mathbf{M})\mathbf{u}_0 - \Delta\mu\mathbf{M}_0\mathbf{u}_0 \approx 0 \ . \tag{6.2.16}$$

Premultiplying by $\mathbf{u}_0^T$ and using Eq. (6.2.14) and the symmetry of $\mathbf{K}_0$ and of $\mathbf{M}_0$ we get

$$\Delta\mu \approx \frac{\mathbf{u}_0^T(\Delta\mathbf{K} - \mu_0\Delta\mathbf{M})\mathbf{u}_0}{\mathbf{u}_0^T\mathbf{M}_0\mathbf{u}_0} \ . \tag{6.2.17}$$

Alternatively, we can premultiply Eq. (6.2.15) by $(\mathbf{u}_0+\Delta\mathbf{u})^T$ and neglect some higher order terms in the perturbation to get

$$\mu_0 + \Delta\mu \approx \frac{\mathbf{u}_0^T(\mathbf{K}_0 + \Delta\mathbf{K})\mathbf{u}_0}{\mathbf{u}_0^T(\mathbf{M}_0 + \Delta\mathbf{M})\mathbf{u}_0} \ . \tag{6.2.18}$$

Equations (6.2.17) and Eq. (6.2.18) have been obtained by neglecting quadratic and cubic terms, and it can be shown that their errors (which are not the same) are proportional to the square of the perturbation in the design $\Delta\mathbf{x}$, or that they are first order approximations.

Another first-order approximation was suggested by Pritchard and Adelman [28]. It is based on integrating the derivative of the eigenvalue μ with respect to a design variable x. Equation (7.3.5) for the eigenvalue derivative may be written as

$$\frac{d\mu}{dx} = a - \mu b \ , \tag{6.2.19}$$

where

$$a = \frac{\mathbf{u}^T\dfrac{d\mathbf{K}}{dx}\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}} \ , \qquad \text{and} \qquad b = \frac{\mathbf{u}^T\dfrac{d\mathbf{M}}{dx}\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}} \ . \tag{6.2.20}$$

Assuming that a and b do not change and $a \neq 0$, we obtain the solution of the differential equation as a function of the design variable x as

$$\mu = (\mu_0 - \frac{a}{b})e^{-b(x-x_0)} + \frac{a}{b} . \tag{6.2.21}$$

As $a \to 0$ Eq. (6.2.21) tends to the standard linear approximation. When several variables are changed simultaneously, x can be taken to be the distance along the path from $\mathbf{x}_0$ to $\mathbf{x}_0 + \Delta\mathbf{x}$ (see [28]). This approximation is called the DEB (Differential Equation Based) approximation in [28].

The first order approximation of Equation (6.2.18) is the Rayleigh quotient approximation to the perturbed eigenvalue based on the nominal eigenvector $\mathbf{u}_0$. If we can calculate a linear approximation $\mathbf{u}_L$ to the eigenvector (e.g., using first derivative information), then we can use Rayleigh's quotient to get a superior approximation to the perturbed eigenvalue, namely

$$\mu_0 + \Delta\mu \approx \frac{\mathbf{u}_L^T(\mathbf{K}_0 + \Delta\mathbf{K})\mathbf{u}_L}{\mathbf{u}_L^T(\mathbf{M}_0 + \Delta\mathbf{M})\mathbf{u}_L} . \tag{6.2.22}$$

This time the error in Eq. (6.2.22) is proportional to $\|\Delta\mathbf{x}\|^4$, see Murthy and Haftka [29], so that Eq. (6.2.22) is a third-order approximation.

Example 6.2.2

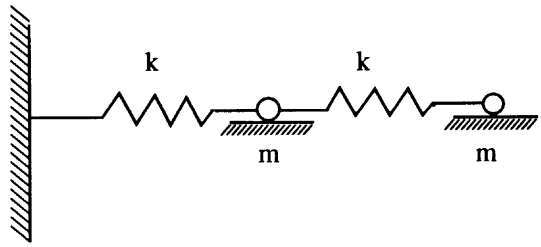


Figure 6.2.1 Mass-spring system.

Consider the two-degrees-of-freedom system shown in Fig. (6.2.1). Estimate the effect on the lowest frequency caused by doubling the left mass. The stiffness and mass matrices for this system are

$$\mathbf{K} = k\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} , \qquad \mathbf{M} = m\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

The lowest eigenvalue and corresponding eigenvector are

$$\mu_0 = 0.382k/m , \qquad \mathbf{u}_0^T = (1,\ 1.618) .$$

For the perturbed system there is no change in the stiffness matrix, and

$$\mathbf{M} + \Delta\mathbf{M} = m\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \text{or} \qquad \Delta\mathbf{M} = \begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}.$$

From Eq. (6.2.20) we get

$$\Delta\mu \approx \frac{[1 \quad 1.618](-0.382k/m)\begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}}{[1 \quad 1.618]\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}} = -0.106k/m ,$$

or

$$\mu_0 + \Delta\mu \approx 0.276k/m .$$

Similarly, from Eq. (6.2.21) we get

$$\mu_0 + \Delta\mu \approx \frac{[1 \quad 1.618]\begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}}{[1 \quad 1.618]\begin{bmatrix} 2m & 0 \\ 0 & m \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}} = 0.299k/m .$$

We now consider the DEB approximation of Eq. (6.2.21) with x being the change in left mass

$$b = \frac{[1 \quad 1.618]\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}}{[1 \quad 1.618]\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}\begin{Bmatrix} 1 \\ 1.618 \end{Bmatrix}} = 0.276/m , \quad \text{and} \quad a = 0 .$$

For the nominal design $x = 0$, and for the perturbed design $x = m$ so that

$$\mu = 0.382(k/m)e^{-0.276} = 0.290(k/m) .$$

The exact result is

$$\mu_0 + \Delta\mu = 0.293k/m .$$

We see that the errors associated with the three first-order approximations, 5.8%, 2.0%, and 1.0%, are small compared to the 30.4% difference between the nominal $(0.382k/m)$ and perturbed $(0.293k/m)$ eigenvalues.• • •

6.3 Sequential Linear Programming

The constraint approximations and approximate analysis procedures described in the previous sections are particularly useful when the computational cost of a single evaluation of the objective function, the constraints, and their derivatives is very large

compared to the computational cost associated with the optimization operations, such as the calculation of search directions. This is a typical situation when we employ a finite element model with thousands of degrees of freedom to analyze a structural design which is defined in terms of a handful of design variables. It then pays to reduce the number of exact structural analyses required for the design process by applying optimization algorithms to a model of the structure based on approximations.

The simplest and most popular approximation approach is that of sequential linear programming (SLP). Consider an optimization problem of the form

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}), \\ \textbf{subject to} \quad & g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, n_g \ . \end{aligned} \tag{6.3.1}$$

The SLP approach starts with a trial design $\mathbf{x}_0$, and replaces the objective function and constraints by linear approximations obtained from a Taylor series expansion about $\mathbf{x}_0$

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}_0) + \sum_{i=1}^{n} (x_i - x_{0i}) \left(\frac{\partial f}{\partial x_i} \right)_{\mathbf{x}_0} \ , \\ \textbf{subject to} \quad & g_j(\mathbf{x}_0) + \sum_{i=1}^{n} (x_i - x_{0i}) \left(\frac{\partial g_j}{\partial x_i} \right)_{\mathbf{x}_0} \geq 0 \qquad j = 1, \ldots, n_g \ , \\ \textbf{and} \quad & a_{li} \leq x_i - x_{0i} \leq a_{ui} \ . \end{aligned} \tag{6.3.2}$$

The last set of constraints are called *move limits*, with a_{li} and a_{ui} being the lower and upper bounds, respectively, on the allowed change in x_i.

Because of the approximation involved, and the move limits, it is rare that the final design of the linearized problem, $\mathbf{x}_L$, is acceptably close to the optimum design. However, if the move limits are small enough to guarantee a good approximation within these move limits, $\mathbf{x}_L$ will be closer to the optimum than $\mathbf{x}_0$. We can, therefore, replace $\mathbf{x}_0$ by $\mathbf{x}_L$, and repeat the linear optimization with Eq. (6.3.1) linearized about the new starting point. This process is repeated, so that we replace the original optimization problem by a sequence of linear programming (LP) problems (hence the name SLP). Each linear optimization is called an optimization cycle. The nature of the linearization of a nonlinear problem and the application of move limits are demonstrated in the following example.

Example 6.3.1

Consider the problem

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = -2x_1 - x_2 \ , \\ \textbf{subject to} \quad & g_1 = 25 - x_1^2 - x_2^2 \geq 0 \ , \\ & g_2 = 7 - x_1^2 + x_2^2 \geq 0 \ , \\ & x_1, \quad x_2 \geq 0 \ . \end{aligned}$$

Linearize the constraint functions about the starting point of $\mathbf{x}_0^T = (1.0, 1.0)$, and use move limits of $1.0x_{0i}$.

Evaluating the constraint functions and derivatives at the initial point we have

$$g_1(\mathbf{x}_0) = 25 - 1 - 1 = 23, \qquad g_2(\mathbf{x}_0) = 7 - 1 + 1 = 7,$$

$$\nabla g_1 = \begin{Bmatrix} -2x_1 \\ -2x_2 \end{Bmatrix}, \qquad (\nabla g_1)_{\mathbf{x}_0} = \begin{Bmatrix} -2 \\ -2 \end{Bmatrix},$$

$$\nabla g_2 = \begin{Bmatrix} -2x_1 \\ 2x_2 \end{Bmatrix}, \qquad (\nabla g_2)_{\mathbf{x}_0} = \begin{Bmatrix} -2 \\ 2 \end{Bmatrix}.$$

Therefore, the linear approximations take the form

$$g_{1L}(\mathbf{x}) = 23 + [-2 \;\; -2] \begin{Bmatrix} x_1 - 1 \\ x_2 - 1 \end{Bmatrix} = 27 - 2x_1 - 2x_2 \geq 0,$$

$$g_{2L}(\mathbf{x}) = 7 + [-2 \;\; 2] \begin{Bmatrix} x_1 - 1 \\ x_2 - 1 \end{Bmatrix} = 7 - 2x_1 + 2x_2 \geq 0 .$$

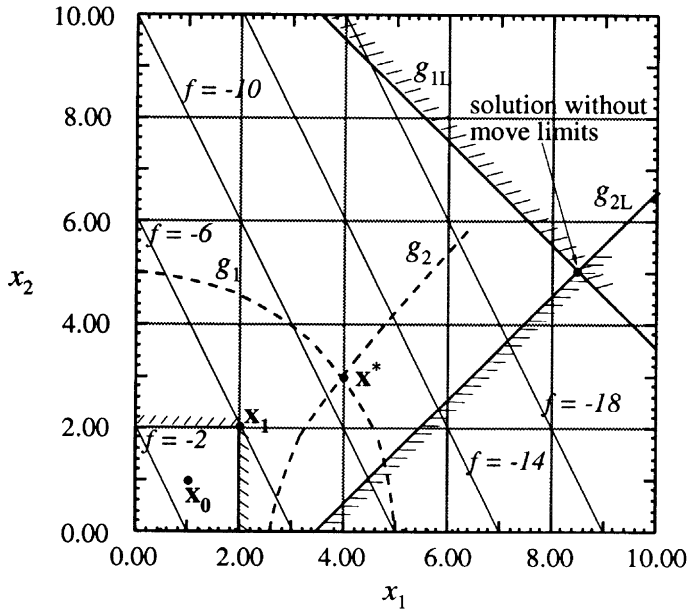


Figure 6.3.1 Constraint linearization and move limits.

These linear approximations are shown in Figure (6.3.1) together with the original constraints represented by the dashed lines. Also shown in the figure are the move limits which form a rectangular boundary around the initial design point.

The solution of this new linear programming problem is $\mathbf{x}_1^T = (2.0\ 2.0)$ with an objective function of $f = -6$ which corresponds to a 100% improvement in the objective function. If there were no move limits, the solution of the problem would have been at $\mathbf{x}_1^T = (8.5\ 5.0)$ and the resulting value of the objective function would be $f = -22$ (see Figure 6.3.1).

Although without move limits we achieve a much larger gain in the objective function, the exact constraints are violated substantially, as shown in Figure (6.3.1). A procedure for evaluating the acceptability of constraint violations is discussed later in this section. • • •

SLP is attractive because reliable LP packages are readily available to most computer users through system library packages, while reliable nonlinear programming packages are not so readily available. However, the SLP strategy has several problems associated with it. First, it greatly increases the computational cost associated with optimization operations, because the optimization process is repeated several times (typically five to forty times). Thus, this strategy is reasonable only when the cost of these optimization computations is small compared to the cost of analysis plus the cost of sensitivity derivatives. The efficiency of the LP package used for the SLP approach can, therefore, become an important consideration.

Second, without a proper choice of move limits, the process may never converge. In general, move limits should be gradually shrunk as the design approaches the optimum. Part of the reason for the need to shrink the move limits is that the accuracy of the approximation is required to be higher when we get close to the optimum. When we are far from the optimum design, the gains that are made during each cycle are large, and we can tolerate significant errors and still make progress towards the optimum. When we get close to the optimum, the gains are small and can be swamped by approximation errors. However, reduction of the move limits early in the process may unnecessarily slow down the convergence too, especially if the initial design is far from the actual optimum. The need to reduce move limits is indicated when the final design of a cycle proves, upon exact analysis, to be inferior to the initial design of that cycle (which is the final design of the previous cycle), or provides no gain in the function f. The move limits are typically shrunk by ten to fifty percent of their previous values until the improvement in the objective function for a given set of move limits becomes smaller than a given tolerance. Popular choices for starting values of the move limits are in the range of ten to thirty percent of the design variables. However, this choice is reasonable only if a design variable is not exceedingly small because it may be on its way to changing its sign. In such a case, it may be reasonable for the move limits to be ten to thirty percent of a typical value (as opposed to the instantaneous value) of that design variable.

A third difficulty associated with SLP arises occasionally when the starting design is infeasible. The combined effects of approximation and move limits can then result in a situation where the linearized optimization problem does not have a feasible solution. That is, if the initial point of a problem is infeasible with respect to the normalized constraints and the move limits are small, the region formed by the move limits may remain entirely inside the infeasible linearized design space leading to an infeasible problem. In this case it is advisable to relax the constraints during the first

few cycles. This can be done, for example, by replacing the optimization problem Eq. (6.3.2) by

$$\begin{aligned}
\textbf{minimize} \quad & f(\mathbf{x}_0) + \sum_{i=1}^{n}(x_i - x_{0i})\left(\frac{\partial f}{\partial x_i}\right)_{\mathbf{x}_0} + k\beta \,, \\
\textbf{subject to} \quad & g_j(\mathbf{x}_0) + \sum_{i=1}^{n}(x_i - x_{0i})\left(\frac{\partial g_j}{\partial x_i}\right)_{\mathbf{x}_0} + \beta \geq 0 \,, \qquad j = 1, \ldots, n_g \,, \\
& a_{li} \leq x_i - x_{0i} \leq a_{ui} \,, \\
\textbf{and} \quad & \beta \geq 0 \,,
\end{aligned} \tag{6.3.3}$$

where β is an additional design variable, representing the allowed margin of original constraint violation, and k is a number chosen to make the contribution of β to the objective function large enough, so that the optimization cycle will emphasize reducing β over reducing f.

Finally, if the solution of the original problem is not at a vertex of the constraint set it is possible that the iterations can cycle between two points. For example, if the actual optimum is at the boundary of a nonlinear constraint, solution of the linearized problem may take the design back to the initial point of the previous linear problem. An appropriate move limits reduction strategy can resolve this difficulty easily.

The following example demonstrates some of the considerations in the choice of move limits.

Example 6.3.2

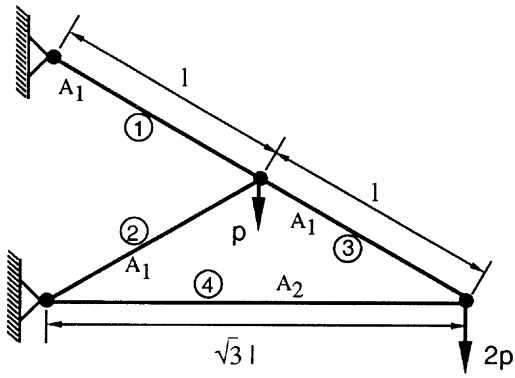


Figure 6.3.2 Four bar statically determinate truss.

We consider the minimum weight design of a four bar statically determinate truss shown in Figure (6.3.2). In the interest of simplicity we assume members 1 through 3 to have the same area A_1 and member 4 an area A_2. Under the specified loading

the member forces and the vertical displacement at joint 2 can be easily verified to be

$$f_1 = 5p, \quad f_2 = -p, \quad f_3 = 4p, \quad f_4 = -2\sqrt{3}p \ ,$$
$$\delta_2 = \frac{6pl}{E}\left(\frac{3}{A_1} + \frac{\sqrt{3}}{A_2}\right) \ ,$$

where a negative sign denotes compression. We assume the allowable stresses in tension and compression to be $7.73 \times 10^{-4}E$ and $4.833 \times 10^{-4}E$, respectively, and limit the vertical displacement to be no larger than $3 \times 10^{-3}l$. The problem of the minimum weight design subject to stress and displacement constraints can be formulated in terms of nondimensional variables

$$x_1 = 10^3\left(\frac{p}{A_1E}\right), \quad x_2 = 10^3\left(\frac{p}{A_2E}\right) \ ,$$

as

$$\begin{aligned} \textbf{minimize} \quad & f(x_1, x_2) = \frac{3}{x_1} + \frac{\sqrt{3}}{x_2} \ , \\ \textbf{subject to} \quad & 18x_1 + 6\sqrt{3}x_2 \leq 3 \ , \\ & 0.05 \leq x_1 \leq 0.1546 \ , \\ & 0.05 \leq x_2 \leq 0.1395 \ , \end{aligned}$$

where lower bound limits on x_1 and x_2 have been assumed to be 0.05.

We start the first cycle with an initial guess of $\mathbf{x}_0^T = (0.1, 0.1)$ which satisfies the constraints and gives $f = 47.32$. The LP problem is started with ten percent move limits, $a_{ui} = a_{li} = 0.01$, $i = 1, 2$. Only the objective function is nonlinear, and its derivatives at $\mathbf{x}_0$ are

$$\frac{\partial f}{\partial x_1} = -300, \qquad \frac{\partial f}{\partial x_2} = -173.2,$$

so that the first LP is

$$\begin{aligned} \textbf{minimize} \quad & f_L = 47.32 - 300(x_1 - 0.1) - 173.2(x_2 - 0.1) \ , \\ \textbf{subject to} \quad & 18x_1 + 6\sqrt{3}x_2 \leq 3, \\ & 0.09 \leq x_1 \leq 0.11, \\ & 0.09 \leq x_2 \leq 0.11 \ . \end{aligned}$$

This problem is solved to yield $x_1 = 0.10316$, $x_2 = 0.11$, and $f_L = 44.6410$. However, $f(0.10316, 0.11) = 44.8274$, so that the linear approximation exaggerated the improvement in f. We next linearize f about $(0.10316, 0.11)^T$, and keeping same-size move limits, we get for the second cycle the following LP:

$$\begin{aligned} \textbf{minimize} \quad & f_L = 44.8274 - 281.9(x_1 - 0.10316) - 143.1(x_2 - 0.11), \\ \textbf{subject to} \quad & 18x_1 + 6\sqrt{3}x_2 \leq 3, \\ & 0.09316 \leq x_1 \leq 0.11316, \\ & 0.1 \leq x_2 \leq 0.12 \ . \end{aligned}$$

The solution for this problem is $x_1 = 0.10893$, $x_2 = 0.1$, $f_L = 44.63126$, $f = 44.86069$. That is, this move resulted in apparent gain (in terms of f_L), but actual loss (in terms of f). This is an indication that we need to reduce the move limits.

We reduce the move limits to 0.005 and perform two additional cycles starting from the best design so far, $\mathbf{x}_0^T = (0.10316, 0.11)$. The first cycle yields $x_1 = 0.10604$, $x_2 = 0.105$, $f_L = 44.72937$, $f = 44.78560$. With the second cycle we get back $x_1 = 0.10316$, $x_2 = 0.11$, and this oscillation again indicates the need for reducing move limits for further improvements. However, with the last set of move limits we reduced f from 44.8274 to 44.78560 which is by less than 0.1 percent. Thus, it may be reasonable to quit. Indeed, for each one of the LP's the nonlinear displacement constraint was active, so that we can find the exact solution by setting

$$18x_1 + 6\sqrt{3}x_2 = 3, \qquad \text{or} \qquad x_2 = \frac{3 - 18x_1}{6\sqrt{3}} ,$$

and substituting into f to get

$$f = \frac{3}{x_1} + \frac{6}{3 - 18x_1} .$$

It is easy to check that the minimum of f is at $x_1 = x_2 = 0.105662$, $f = 44.7846$.

The design space for this problem is shown in Figure 6.3.3• • •

It is possible that the optimum design obtained for a linearized problem at any cycle of the iterative SLP process may violate the constraints of the original problem. We have seen in Example 6.3.1 that if the move limits in that example were not imposed or they were large enough, the solution of the linear problem would have caused a significant violation of the original constraint set. Such constraint violations are generally associated with objective function improvements. It is also possible that, from the solution of one linear problem to the next, the objective function may deteriorate and the constraint violations be reduced. These events can be prevented by altering the imposed move limits. However, neither of the two events is necessarily objectionable for the overall convergence of the SLP. Following is a discussion of how to judge whether a new design obtained by the LP is an improvement when a better objective function is accompanied by constraint violation, or a better satisfaction of the constraint set is accompanied by an increase in the objective function.

Suppose the optimum of the LP during the ith cycle, $\mathbf{x}_{iL}^*$, leads to a set of active or violated constraints $g_j(\mathbf{x}_{iL}^*)$, $j \in \mathbf{J}$, where $\mathbf{J}$ is the set of active constraints. We can view the solution of the linearized problem as the exact solution of the following modified nonlinear problem

$$\begin{aligned} \textbf{minimize}: \quad & f(\mathbf{x}), \\ \textbf{subject to}: \quad & g_j(\mathbf{x}) \geq p g_j(\mathbf{x}_{iL}^*), \end{aligned} \tag{6.3.4}$$

for $p = 1$. The actual problem we want to solve is for $p = 0$. Using Eq. (5.4.7) we can estimate that the optimum value of the objective function for the unmodified problem is

$$\mathcal{L} = f(\mathbf{x}_{iL}^*) - \sum_{j=1}^{r} \lambda_j g_j(\mathbf{x}_{iL}^*) , \tag{6.3.5}$$

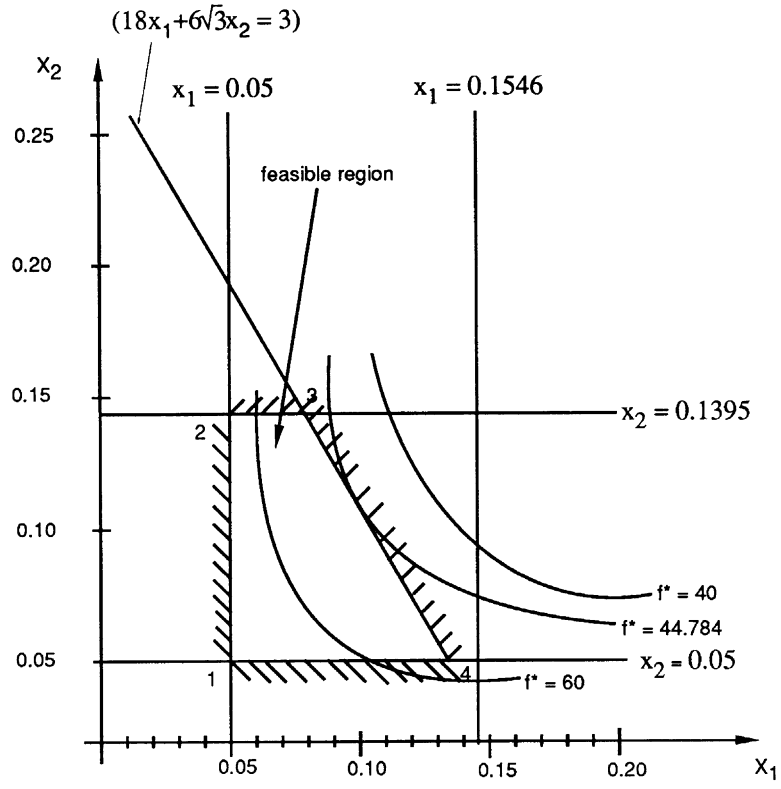


Figure 6.3.3 Design space for four-bar truss problem.

where $\mathcal{L}$ is the Lagrangian function. This suggests the following procedure: If the objective function and the most critical constraints both improve, always accept the new design. If the objective function improves and the constraints deteriorate or vice versa, compare the values of the Lagrangians. If the Lagrangian at the end of a cycle is smaller than its value at the beginning of the cycle, then accept the new design. If, on the other hand, the Lagrangian increases, modify the move limits. We recommend using only critical and violated constraints in the Lagrangians.

Example 6.3.3

Consider example 6.3.2 with variables $y_i = 1/x_i$ (proportional to the cross-sectional areas). The problem takes the following form

$$\textbf{minimize}: \quad f(\mathbf{y}) = 3y_1 + \sqrt{3}y_2 \ ,$$

$$\textbf{subject to}: \quad g = 3 - \frac{18}{y_1} - \frac{6\sqrt{3}}{y_2} \geq 0,$$

$$8.0 \leq y_1 \leq 20,$$
$$8.0 \leq y_2 \leq 20 \ ,$$

where lower bound for the variables are increased to 8.0 for convenience. An initial guess of $y_1 = 12$, and $y_2 = 8$ results in $f = 49.856$.

Linearizing the problem with 30% move limits leads us to the problem

$$\begin{aligned} \textbf{minimize}: \quad & 3y_1 + \sqrt{3}y_2 \ , \\ \textbf{subject to}: \quad & 0.125y_1 + 0.1624y_2 \geq 2.598, \\ & 8.4 \leq y_1 \leq 15.6, \\ & 8.0 \leq y_2 \leq 10.4 \ . \end{aligned}$$

Solution of this LP yields $y_1 = 8.4$, $y_2 = 9.534$, and $f = 41.713$. Reanalysis reveals $g = -0.2329$. Also from the solution of the LP problem we obtain $\lambda_1 = 10.667$ (corresponding to g) and $\lambda_2 = 1.667$ (corresponding to move limit). Therefore the Lagrangian is

$$\mathcal{L} = 41.713 - 10.667(-0.2329) = 44.197 \ ,$$

which, compared to the initial objective function of $f = 49.856$, is a smaller improvement than the solution of the LP, 41.713, but still acceptable.

Linearizing the constraint function about the last design point and formulating the LP problem with 30% move limits we find the problem

$$\begin{aligned} \textbf{minimize}: \quad & 3y_1 + \sqrt{3}y_2 \ , \\ \textbf{subject to}: \quad & 0.2551y_1 + 0.1143y_2 \geq 3.4658, \\ & 8.0 \leq y_1 \leq 10.92, \\ & 8.0 \leq y_2 \leq 12.3938 \ , \end{aligned}$$

which has a solution of $y_1 = 10.000$, $y_2 = 8.0$, and $f = 43.858$, with constraint function multiplier of $\lambda_1 = 11.76$ and a lower bound multiplier of $\lambda_2 = 0.38$. Although the objective function increased roughly by 5%, evaluation of the actual constraint shows a smaller constraint violation, $g = -0.09896$ compared to the initial design of this LP problem. Therefore, we must calculate the Lagrangian in order to accept or reject this design. At the end of this step the Lagrangian is

$$\mathcal{L} = 43.858 - 11.76(-0.09896) = 45.022 \ ,$$

which is larger than the value of the Lagrangian calculated at the end of the previous LP problem. We, therefore, reject the design and reconstruct the LP problem with smaller move limits.• • •

6.4 Sequential Nonlinear Approximate Optimization

We can generalize SLP by using nonlinear approximations for some of the constraints and objective function. For the application of SLP we need to linearize even

simple nonlinear functions. With the more general procedure we approximate only expensive-to-calculate functions using either linear or nonlinear (such as quadratic) approximations. Inexpensive constraints need not be approximated at all. We start by identifying those constraints (and possibly the objective function) which require large computational resources for evaluation. These constraints are singled out for approximation, while the cheaper constraints are evaluated exactly. Given a trial solution $\mathbf{x}_0$ to the structural design problem, we construct approximations to the expensive constraints about $\mathbf{x}_0$. As in the case of SLP, we need to augment the approximate problem with move limits to guard against large changes in design variables that can result in poor approximations.

The solution of the approximate problem with the move limits, obtained by any optimization procedure is denoted as $\mathbf{x}_1$. We perform a new exact structural analysis at $\mathbf{x}_1$, use it to construct new approximations to the expensive constraints, and perform a new optimization of the approximate problem. That is, the original optimization problem Eq. (6.3.1) is replaced by

$$\begin{aligned} \textbf{minimize} \quad & f_a(\mathbf{x}, \mathbf{x}_0^{(i)}), \\ \textbf{subject to} \quad & g_{aj}(\mathbf{x}, \mathbf{x}_0^{(i)}) \geq 0 \quad j = 1, \ldots, n_g \ , \\ \textbf{and} \quad & \|\mathbf{x} - \mathbf{x}_0^{(i)}\| \leq a_i \ , \\ \text{for} \quad & i = 0, 1, 2, \ldots \ , \end{aligned} \tag{6.4.1}$$

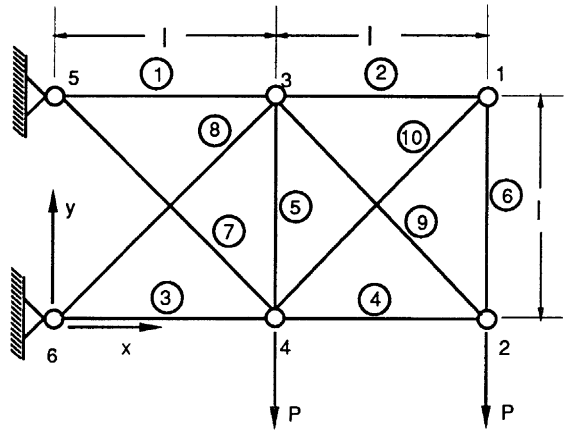
where f_a and g_{aj} denote the approximate objective function and constraints, respectively, $\mathbf{x}_0^{(i)}$ is the solution of the ith minimization, and a_i is a suitably chosen move limit.

Because most of the cost of the optimization is associated with the exact analysis and sensitivity calculations, it is often not important what optimization procedure is used for obtaining the optimum of approximate problems. In general, it is more important to emphasize reliability and robustness in the choice of the optimization procedure rather than computational efficiency.

The following example demonstrates the use of sequential nonlinear approximate optimization with the standard approximations discussed in section 6.1 as well as one which was tailored more to the problem at hand.

Example 6.4.1

The ten-bar truss shown in Figure (6.4.1) is a standard example used by many authors. The minimum weight design obtained by changing the cross-sectional areas of the truss members is sought subject to stress constraints and minimum gage constraints of $0.1in^2$. The maximum allowable stress in each member is the same in tension and compression. This allowable is set to 25 ksi for all members except member 9. For member 9 the stress allowable is 75 ksi. The density of the truss material is $0.1lb/in^3$.



l = 360 ", P = 100 Kips

Figure 6.4.1 10-bar truss.

Table 6.4.1 Ten-bar truss designs

Member	Initial area (in^2)	Optimum area (in^2)	Stress (ksi)
1	5.0	7.90	25.0
2	5.0	0.10	25.0
3	5.0	8.10	-25.0
4	5.0	3.90	-25.0
5	5.0	0.10	-0.07
6	5.0	0.10	25.0
7	5.0	5.80	25.0
8	5.0	5.51	-25.0
9	5.0	3.68	37.5
10	5.0	0.14	-25.0

The five generic local approximations described in section 6.1 were used here, together with the linear force approximation proposed by Vanderplaats and coworkers [e.g., 15]. Table 6.4.1 shows the initial and optimum designs and the stresses in the optimum truss members.

Table 6.4.2 compares the convergence history of twelve cycles of approximate optimization using the six approximations. To compare the performance of the various approximations in Table 6.4.2 a useful measure of performance is the number of cycles required to get to within one percent of the optimum weight (that is to 1514 *lb*). The linear, reciprocal-quadratic, and linear force approximations required six cycles, the quadratic approximation seven, the reciprocal approximation ten, and the conservative approximation never made it. The difference between the linear and reciprocal approximations turns out to be an idiosyncrasy of this problem. For many truss

Table 6.4.2 Convergence of optimum weight (lb) using different approximations

Cycle	Linear	Reciprocal	Conservative	Quadratic	Recip-quadratic	Linear force
1	1845	1774	2361	2002	1931	1891
2	1637	1673	1960	1741	1684	1688
3	1601	1593	1722	1650	1595	1589
4	1558	1566	1641	1586	1548	1549
5	1531	1548	1587	1547	1522	1526
6	1514	1537	1566	1525	1509	1511
7	1507	1528	1555	1514	1506	1504
8	1502	1522	1546	1507	1502	1501
9	1500	1518	1540	1503	1500	1500
10	1500	1511	1538	1501	1500	1499
11	1500	1511	1535	1500	1499	1499
12	1499	1508	1532	1499	1499	1499

problems the reciprocal approximation does better than the linear one. As a group, the second order approximations are slightly better than the first order ones, but the difference does not appear to be significant enough to justify the cost of computing second derivatives (see Section 7.2.2 for discussion of the cost of calculating second derivatives).

The dismal performance of the conservative approximation is explained by the fact that it is typically less accurate than either the linear or reciprocal approximation. It is useful in situations where we need the conservativeness (such as when it is employed with interior penalty function algorithms), or the convexity (such as with dual algorithms, see Chapter 9). However, for sequential approximate optimization it is of little use. Finally, the linear force approximation due to Vanderplaats is comparable in performance to the second-order approximations even though it employs only first derivatives. This is due to the fact that it approximates a "more linear" quantity than the stress. In using this approximation we approximate an intermediate quantity—the member force, and compute the stress exactly from the force. Similar physical insight leading to identification of quantities that are approximately linear can afford comparable gains in other problems. • • •

6.5 Special Problems Associated with Shape Optimization

The term shape optimization is employed here in a very broad sense. In terms of a finite element model we consider as shape optimization any problem where we need to change the position of the nodes of the finite-element model or the element connectivity (e.g remove elements). Shape optimization problems are contrasted with sizing optimization problems where we change only element stiffness properties, such as bar cross-sectional areas or plate thicknesses. The term shape optimization is often used in a narrow sense referring only to the optimal design of the shape of the boundary of two- and three-dimensional structural components. The broad usage includes also *geometrical* optimization of skeletal structures, and *topological*

optimization which decides the connectivity of the structure (for example, which nodes are connected by elements).

Shape optimization problems are typically more difficult to tackle than sizing optimization problems. Consider first the optimization of the boundary shape of two- and three-dimensional bodies. The calculation of sensitivity derivatives for these shape optimization problems is associated with accuracy problems discussed in Chapters 7 and 8. Another serious problems is mesh deformation. As the shape of the structure changes we need to change the finite-element mesh. Simple remeshing rules that translate node positions as the boundary changes, usually lead to highly deformed finite elements and concomitant loss of accuracy. This problem can be addressed by manually remeshing during the optimization process (which is time consuming), or employing sophisticated mesh generators. Work in shape optimization has indeed spurred the development and usage of such mesh generators (e.g., [30,31]).

Another problem associated with boundary shape optimization is that of the existence or creation of internal boundaries or holes. In many problems the optimal design will have internal cavities. It is impossible to generate these cavities with a standard optimization approach without prior knowledge of their existence. That is, an optimization procedure can easily find for us the optimum shape of a cavity once we assume there is one, but it cannot tell us that there should be one, two, or three cavities. One approach for dealing with this problem is to assume that the material is not homogeneous, but instead has an underlying microstructure. This underlying microstructure can be of fibers and matrix composite material. However, typically the assumed microstructure is more general than that of the fiber and matrix components of a laminated plate, and includes microcavities in the material. This type of microstructure was devised so as to probe the theoretical limits of strength and stiffness that can be attained by a structure (see, e.g., Kohn and Strang [32], or Rozvany et al. [33]). Bendsøe and Kikuchi [34] showed that it can be used to determine the need for introducing cavities into the structure. Figure 6.5.1 shows the type of structure obtained by Bendsøe and Kikuchi by permitting microcavities. The structure under consideration is a bar in tension where the cross sections at the two ends are given (solid areas in figure), and the cross section on the left is larger than the that on the right. The objective is to maximize the stiffness of the bar for a given volume. The result shown in the figure, while not practical in itself, permits us to identify regions where cavities exist. Standard optimization techniques can then be used to find the optimal shape of these cavities.

An example of the application of this technique was reported by Rasmussen [35] for the design of a floor beam design in a civil transport. Figure 6.5.2 shows the topology that was assumed by the designers and the topology identified by the homogenization approach which led to a substantially lighter design.

The problem of finding the cavities in two- and there-dimensional bodies belongs to the realm of topological optimization. Topological optimization is a difficult problem which has received more attention in applications to skeletal structures such as trusses and frames. There the optimum topology is typically defined by decisions as to which joints are connected to each other by members. The basic approach followed by most researchers is to create a *ground structure* where every joint is connected to

Figure 6.5.1 Optimal shapes for a fillet problem using microstructure.

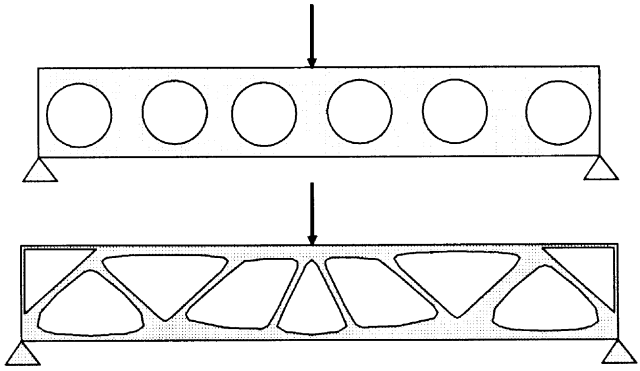


Figure 6.5.2 Shape design of floor beam for a civil transport aircraft: initial and final geometries.

every other joint. If the design problem is minimum weight with constraints on the plastic collapse load, then as shown in Chapter 3, the optimization problem is linear, and the simplex method may be used to find the optimum design. The algorithm also automatically removes all unnecessary members. This approach was first taken by Dorn and co-workers [36].

When the structure is designed subject to stress and displacement constraints rather than plastic collapse, it may be impossible to start with a ground structure and rely on a standard optimization algorithm to remove unnecessary members. One problem is that the members that need to be removed may experience large strains as their areas are reduced, so that the optimization algorithm will tend to reinforce them rather than eliminate them. Because this problem is related to the compatibility conditions, it is possible to relax these during part of the optimization process for the purpose of eliminating members (e.g., Sheu and Schmit [37], or Reinschmidt and Russel [38]). Another problem is that the stiffness matrix may become singular due

to the removal of members. This problem may be overcome by using simultaneous-analysis-and-design techniques which do not require the inversion or factorization of the stiffness matrix (see Section 10.6). The reader is referred to two survey papers by Topping [39], and Kirsch [40] for additional information on topological optimization.

Geometrical optimization of skeletal structures refers to the search for the optimum locations of the joints of the structures. The problem can be solved by standard techniques, but there are often numerical advantages to treating the geometry variables differently from the sizing variables and employing a two-level optimization approach. This topic is discussed in Chapter 10 in Section 10.5.

6.6 Optimization Packages

During the first few years of the development of structural optimization most analysts developed special purpose finite-element programs with built-in optimization procedures for their own use. When these programs were used by other analysts they found them to be insufficiently documented and difficult to modify. In recent years it has become more common to employ general purpose constrained optimization packages and interface them with general purpose structural analysis codes. Additionally, the growing popularity of structural optimization as a tool for industrial applications is generating demand for the introduction of optimization capabilities into general-purpose analysis packages. The purpose of this section is to provide the reader with brief description of some of the more popular packages.

First consider integrated packages which combine structural analysis and optimization procedures. One of the more popular programs of this class was the TSO program (originally called WASP [41,42]) developed for the preliminary design of aircraft wing and tail structures subject to aeroelastic constraints. The program models the wing or tail structure as an orthotropic plate and employs simplified plate analysis rather than a finite element model. Design variables are coefficients of polynomials that describe the thickness distribution and ply orientations over the surface. The optimization procedure is based on an interior penalty function formulation (see Chapter 5). The program has been used extensively for design studies and for some actual aircraft design problems (see [43]).

Many integrated structural optimization packages are based on special purpose finite element programs. One of the better known is the ACCESS program developed by Schmit and co-workers [44,45]. Other programs of this type include FASTOP [46], OPSTAT [47], OPTCOMP [48], OPTIMUM [49], ASOP [50], STARS [51] and DESAP [52].

Because of the lack of generality associated with special purpose finite-element programs, there has been interest in structural optimization packages built around a general purpose finite element program. Two early examples of this type are PARS[53] and PROSSS [54]. These programs are based on the SPAR finite-element package and its commercial derivative EAL. However, because the optimization software was not supported by the developer of the finite-element package, the use of

PARS and PROSSS has been limited. The EAL program, however, lends itself to interfacing with other programs, and has been used with optimization software; Walsh [55] reports on the use of EAL together with the CONMIN [56] program.

Other finite-element programs have also been recently used to form structural optimization packages. The OPTSYS package [57] is based on the ASKA and ABAQUS finite-element programs, the ASTROS system [58] evolved from the public domain version of NASTRAN, and the NISAOPT package (including the programs SHAPE [59] and STROPT [60]) is based on NISA II.

The demand for structural optimization is pushing finite-element software developers to include optimization capabilities in their programs. The NASTRAN program [61] and the I-DEAS program [62] now have sensitivity and optimization capabilities, and ANSYS has a built in rudimentary optimizer. A recently developed program, GENESIS [63], has gone one step further in that it is a general finite element program developed together with sensitivity, approximation and optimization capabilities. In the not too distant future we can expect that most commercial structural analysis packages will offer built-in optimization capabilities.

Until that day, and probably even after, there will be a continuing demand for general purpose optimization software that can be coupled to structural analysis programs. Most finite-element packages lend themselves to the calculation of sensitivities via finite-differences (see Section 7.1), so that the analyst can construct constraint approximations based on these derivatives (see Section 6.1) and use the optimization package on this approximation in a sequential-approximate-optimization mode. The most commonly available general-purpose optimization packages are linear programming (LP) solvers. These are usually available at most computer centers as part of IMSL or similar subroutine libraries. While in some cases there are advantages to using more general optimization algorithms, LP packages seem to work well in the majority of applications.

At the other extreme of generality we find the ADS [64], DOT [65] and DOC [66] packages from VMA Engineering which allow the user a wide menu of optimization algorithms and strategies. These programs evolved from the very popular CONMIN [56] package which was used extensively for structural optimization. DOT (Design Optimization Tool) is a collection of fortran subroutines for optimization, and DOC (Design Optimization Control) is a control program that simplifies the use of optimization (calling DOT subroutines). Another general-purpose optimization package, commonly used in structural optimization, is NEWSUMT [67] developed by Miura and Schmit which is based on a penalty function procedure (see Chapter 5), and an updated version of the program NEWSUMT-A which incorporates constraint approximations [68]. Other packages of this type include OPT based on the reduced gradient algorithm (see Chapter 5), and IDESIGN [69] based on sequential quadratic programming (see Chapter 5). There are also several packages available from mathematical programming specialists. However, these programs do not enjoy as much popularity in structural optimization applications as the aforementioned programs which were developed by engineers.

6.7 Test Problems

Standard test problems are useful for the purpose of checking optimization algorithms and software. The three test problems given in this section have been widely used for this purpose.

6.7.1 *Ten-bar Truss*

The ten bar truss shown in Figure 6.4.1 is a classical example used to show the difference between a fully stressed design (FSD) and an optimum design. The material properties and the minimum area are given in Table 6.7.1. When the truss is designed subject to stress constraints only, the optimum and FSD designs are identical. However, when the stress allowable for member 9 is increased above 37,500 psi the optimum design and the FSD design are different. The three designs are given in Table 6.7.2. The truss has also been optimized with displacement constraints (Table 6.7.3) and the final design is given in Table 6.7.4. For additional information, see Ref. [70].

Table 6.7.1 Data for ten bar truss

Material:	aluminum	Specific mass:	0.1 lbm/in^3
Young's modulus:	10^7 psi	Allowable stress:	±25 000 psi
Minimum area:	0.1 in^2		

Table 6.7.2 Final designs for ten bar truss with stress constraints only

Member	FSD and optimum areas(in^2)	Increased allowable, member 9 FSD areas(in^2)	Increased allowable, member 9 optimum design areas(in^2)
1	7.94	4.11	7.90
2	0.10	3.89	0.10
3	8.06	11.89	8.10
4	3.94	0.11	3.90
5	0.10	0.10	0.10
6	0.10	3.89	0.10
7	5.74	11.16	5.80
8	5.57	0.15	5.51
9	5.57	0.10	3.68
10	0.10	5.51	0.14
Mass (lbm)	1593.2	1725.2	1497.6

Table 6.7.3 Displacement allowables for ten bar truss

Case	Node	Direction	Displacement limits lower	upper
A	1	Y	-2.0	-2.0
	3	Y	-1.0	-2.0
B	1–4	Y	-2.0	+2.0

Table 6.7.4 Optimum designs for ten bar truss with displacement constraints

Member	Cross-sectional areas (in^2) Case A	Case B	Member	Case A	Case B
1	22.66	30.52	6	0.10	0.55
2	1.40	0.10	7	12.69	7.46
3	21.58	23.20	8	14.54	21.04
4	8.43	15.22	9	11.93	21.53
5	0.10	0.10	10	1.98	0.10
Mass(lbm)	4048.96	5060.85			

6.7.2 *Twenty-five-bar Truss*

The twenty-five-bar truss is shown in Figure 6.7.1. The loading, material properties and allowables are shown in Tables 6.7.5, 6.7.6, 6.7.7, and 6.7.8, and the final design is shown in Table 6.7.9. For additional details see Ref. [70].

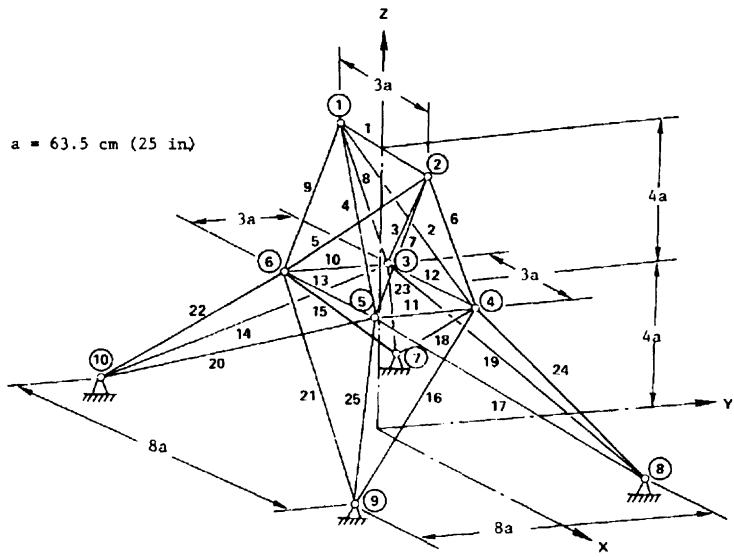


Figure 6.7.1 Twenty-five-bar truss.

Table 6.7.5 Data for twenty-five-bar truss

Material:	aluminum
Young's modulus:	10^7 psi
Specific mass:	0.1 lbm/in^3
Minimum area:	0.01 in^2

Table 6.7.6 Allowable stresses for twenty-five-bar truss (psi)

Members	Tension	Compression	Members	Tension	Compression
1	40000	-35092	12,13	40000	-35092
2–5	40000	-11590	14–17	40000	-6759
6–9	40000	-17305	18–21	40000	-6959
10,11	40000	-35092	22–25	40000	-11082

Table 6.7.7 Nodal load components (lbf) for twenty-five-bar truss

Load case	Node	x	y	z
1	1	1000	10000	-5000
	2	0	10000	-5000
	3	500	0	0
	6	500	0	0
2	5	0	20000	-5000
	6	0	-20000	-5000

Table 6.7.8 Displacement allowables for twenty-five-bar truss

	Displacement limits (in)		
Node	x	y	z
1	±0.35	±0.35	±0.35
2	±0.35	±0.35	±0.35

Table 6.7.9 Optimum design for twenty-five-bar truss

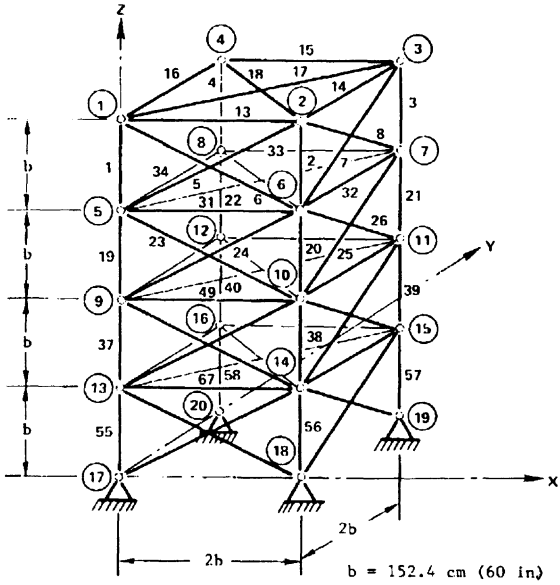
Design variable	Members	Areas (in^2)
1	1	0.010
2	2–5	1.987
3	6–9	2.991
4	10,11	0.010
5	12,13	0.012
6	14–17	0.683
7	18–21	1.679
8	22–25	2.664
Mass(lbm)		545.22

6.7.3 *Seventy-two-bar Truss*

The seventy-two-bar truss is shown in Figure 6.7.2. The loadings, material properties and allowables are shown in Tables 6.7.10, 6.7.11, and 6.7.12, and the optimum design is shown in Table 6.7.13. For additional details see Ref. [70].

Table 6.7.10 Data for seventy-two-bar truss

Material:	aluminum
Young's modulus:	10^7 psi
Specific mass:	0.1 lbm/in^3
Allowable stress:	±25 000 psi
Minimum area:	0.01 in^2



Note: For the sake of clarity, not all elements are drawn in this figure.

Figure 6.7.2 Seventy-two-bar truss.

Table 6.7.11 Nodal load components (lbf) for seventy-two-bar truss

Load case	Node	x	y	z
1	1	5000	5000	-5000
2	1	0	0	-5000
	2	0	0	-5000
	3	0	0	-5000
	4	0	0	-5000
2	5	0	20000	-5000
	6	0	-20000	-5000

Table 6.7.12 Displacement allowables for seventy-two-bar truss

	Displacement limits (in)		
Node	x	y	z
1	±0.25	±0.25	-
2	±0.25	±0.25	-
3	±0.25	±0.25	-
4	±0.25	±0.25	-

Table 6.7.13 Optimum design for seventy-two-bar truss

Design variable	Members	Areas (in^2)
1	1–4	0.1571
2	5–12	0.5356
3	13–16	0.4099
4	17,18	0.5690
5	19–22	0.5067
6	23–30	0.5200
7	31–34	0.1
8	35,36	0.1
9	37–40	1.280
10	41–48	0.5148
11	49–52	0.1
12	53,54	0.1
13	55–58	1.897
14	59–66	0.5158
15	67–70	0.1
16	71,72	0.1
Mass(lbm)		379.66

6.8 Exercises

1. Show that the conservative approximation, Eq. (6.1.9) is concave, and the approximation of Eq. (6.1.11) is convex as long as the design variables do not change their sign.

2. Derive Eq. (6.1.14).

3. Add to Table 6.1.1 a column representing an approximation to the constraint based on a linear approximation of the force in member C (This linear-force approximation is due to Vanderplaats and coworkers [15–17]).

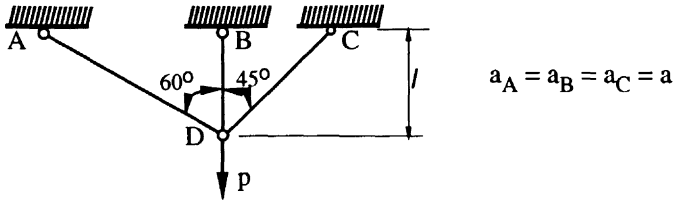


Figure 6.8.1 Asymmetric three-bar truss.

4. The three-bar truss in Figure 6.8.1 has members with equal cross-sectional areas. Calculate the five approximations discussed in Section 6.1 as well as the Linear-force approximation discussed in the previous problem for the stress in member A. Compare the accuracy and conservativeness of the approximations for changes of ±25% in the area of member C.

5. Obtain a good approximation to the stress in member A in the previous problem in terms of the two angles of the truss.

6. The beam in Figure 6.1.1 has a mass density ρ, and cross-sectional area proportional to the square root of the moment of inertia $A = \alpha\sqrt{I}$. Use the global-local approximation to obtain the lowest vibration frequency as I_2/I_1 is varied from 1 to 2. Use a two-element model as the exact solution, and a 1 element model as a global approximation. Note that this requires you to derive the stiffness matrix of a beam with a variable cross section.

7. Prove Eq. (6.2.10).

8. Repeat Example 6.2.1 doubling the left spring constant instead of the mass.

8. Use sequential linear programming to design the three-bar truss of Figure 6.1.2 subject to a yield stress constraint of σ_0 and a minimum gage constraint on all members of $0.1p/\sigma_0$.

9. Repeat the previous problem with the reciprocal approximation.

6.9 References

[1] Schmit, L.A. Jr., and Farshi, B., "Some Approximation Concepts for Structural Synthesis," AIAA Journal, 12, 5, 692–699, 1974.

[2] Mills-Curran, W.C., Lust, R.V., and Schmit, L.A. Jr., "Approximation Methods for Space Frame Synthesis," AIAA Journal, 21 (11), 1571–1580, 1983.

[3] Storaasli, O.O., and Sobieszczanski, J., "On the Accuracy of the Taylor Approximation for Structure Resizing," AIAA Journal, 12 (2), 231–233, 1974.

[4] Noor, A.K., and Lowder, H.E., "Structural Reanalysis via a Mixed Method," Computers and Structures, 5, 9–12,1975.

[5] Fuchs, M.B., "Linearized Homogeneous Constraints in Structural Design," Int. J. Mech. Sci., 22, pp. 33–40, 1980.

[6] Fuchs, M.B., and Haj Ali, R.M., "A Family of Homogeneous Analysis Models for the Design of Scalable Structures," Structural Optimization, 2, pp. 143–152, 1990.

[7] Starnes, J.H. Jr., and Haftka, R.T., "Preliminary Design of Composite Wings for Buckling, Stress and Displacement Constraints," Journal of Aircraft, 16, 564–570, 1979.

[8] Haftka, R.T., and Shore, C.P., "Approximate Methods for Combined Thermal-Structural Analysis," NASA TP-1428, 1979.

[9] Prasad, B., "Explicit Constraint Approximation Forms in Structural Optimization—Part 1:Analyses and Projections," Computer Methods in Applied Mechanics and Engineering, 40 (1), 1–26, 1983.

[10] Braibant, V., and Fleury, C., "An Approximation Concept Approach to Shape Optimal Design," Computer Methods in Applied Mechanics and Engineering, 53, pp. 119–148, 1985.

[11] Prasad, B., "Novel Concepts for Constraint Treatments and Approximations in Efficient Structural Synthesis," AIAA J., 22, 7, pp. 957–966, 1984.

[12] Woo, T.H., "Space Frame Optimization Subject to Frequency Constraints," AIAA J. 25, 10, pp. 1396–1404, 1987.

[13] Schmit, L.A., Jr., and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA CR-2552, 1976.

[14] Lust, R.V., and Schmit, L.A., Jr., "Alternative Approximation Concepts for Space Frame Synthesis," AIAA J., 24, 10, pp. 1676–1684, 1986.

[15] Salajeghah, E., and Vanderplaats G.N.,"An Efficient Approximation Method for Structural Synthesis with Reference to Space Structures," Space Struct. J., 2, pp. 165–175, 1986/7.

[16] Kodiyalam, S., and Vanderplaats G.N., "Shape Optimization of 3D Continuum Structures Via Force Approximation Technique," AIAA J., 27 (9), pp. 1256–1263, 1989.

[17] Hansen, S. R., and Vanderplaats G.N., "Approximation Method for Configuration Optimization of Trusses," AIAA J., 28 (1), pp. 161–168, 1990.

[18] Box, G.E.P., and Draper, N.R., Empirical Model-Building and Response Surface, Wiley, New York, 1987.

[19] Barthelemy, J.-F., and Haftka, R.T., "Recent Advances in Approximation Concepts for Optimum Structural Design," NASA TM 104032, 1991.

[20] Haftka, R.T., Nachlas, J.A., Watson, L.T., Rizzo, T., and Desai, R., "Two-Point Constraint Approximation in Structural Optimization," Computer Methods in Applied Mechanics and Engineering, 60, pp. 289–301, 1989.

[21] Fadel, G.M., Riley, M.F., and Barthelemy, J.-F.M., "Two Point Exponential Approximation Method for Structural Optimization," Structural Optimization, 2, pp. 117–124, 1990.

[22] Haftka, R.T., "Combining Local and Global Approximations," AIAA Journal, Vol. 29 (9), pp. 1523-1525, 1991.

[23] Chang, K.-J., Haftka, R.T., Giles, G.L., and Kao, P.-J., "Sensitivity Based Scaling for Correlating Structural Response from Different Analytical Models," AIAA Paper 91-0925, Proceedings of AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference, Baltimore, MD, April 8–10, 1991.

[24] Kirsch, U., and Taye, S., "High Quality Approximations of Forces for Optimum Structural Design," Computers and Structures, 30, 3, pp. 519–527, 1988.

[25] Haley, S.B., "Solution of Modified Matrix Equations," SIAM J. Numer. Anal., 24 (4), pp. 946–951, 1987.

[26] Fuchs, M.B., and Steinberg, Y., "An Efficient Approximate Analysis Method Based on an Exact Univariate Model for the Element Loads", Structural Optimization, 3 (1), 1991.

[27] Holnicki-Szulc, J., Virtual Distortion Method, Springer Verlag, Berlin, pp. 30–40, 1991.

[28] Pritchard, J.I., and Adelman, H.M., "Differential Equation Based Method for Accurate Approximation in Optimization," AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 2-4, Part I, pp. 414–424, 1990.

[29] Murthy, D.V., and Haftka, R.T., "Approximations to Eigenvalues of Modified General Matrices," Computers and Structures, 29, pp. 903–917, 1988.

[30] Shephard, M.S., and Yerry, M.A., "Automatic Finite Element Modeling for Use with Three-Dimensional Shape Optimization," in The Optimum Shape (Bennett, J.A., and Botkin M.E., eds.), Plenum Press, N.Y. 1986, pp. 113–135.

[31] Yang, R.J., and Botkin, M.E., "A Modular Approach for Three-Dimensional Shape Optimization of Structures," AIAA J., 25 (3), pp. 492–497, 1987.

[32] Kohn, R.V., and Strang, G., "Optimal Design and Relaxation of Variational Problems," Comm. Pure Appl. Math., 39, pp. 113–137 (Part I), pp. 139–182 (Part II), and pp. 353–377 (Part III), 1986.

[33] Rozvany, G.I.N., Ong, T.G., Szeto, W.T., Olhoff, N., and Bendsøe, M.P., "Least-Weight Design of Perforated Plates," Int. J. Solids Struct., 23, pp. 521–536 (Part I), and pp. 537–550 (Part II), 1987.

[34] Bendsøe, M.P., and Kikuchi, N., "Generating Optimal Topologies in Structural Design using a Homogeneization Method," Comp. Meth. Appl. Mech. Engng., 71, pp.197–224, 1988.

[35] Rasmussen, J., "Shape Optimization and CAD," SARA, 1, 33–45, 1991.

[36] Dorn, W.S., Gomory, R.E., and Greenberg, H.J., "Automatic Design of Optimal Structures," J. Mécanique, 3, pp. 25–52, 1964.

[37] Sheu, C.Y., and Schmit, L.A., "Minimum Weight Design of Elastic Redundant Trusses under Multiple Static Loading Conditions," AIAA, J., 10 (2), pp. 155–162, 1972.

[38] Reinschmidt, K.F., and Russel, A.D., "Applications of Linear Programming in Structural Layout and Optimization," Comput. Struct., 4, pp. 855–869, 1974.

[39] Topping, B.H.V., "Shape Optimization of Skeletal Structures—a Review," ASCE J. Struct. Enging., 109 (8), pp. 1933–1951, 1983.

[40] Kirsch, U., "Optimal Topologies of Structures," Appl. Mech. Rev., 42 (8), pp. 223–239, 1989.

[41] McCullers, L.A., and Lynch, R.W., "Composite Wing Design for Aeroelastic Tailoring Requirements," Air Force Conference on Fibrous Composites in Flight Vehicle Design, Dayton, Ohio, September, 1972.

[42] McCullers, L.A., and Lynch, R.W., "Dynamic Characteristics of Advanced Filamentary Composites Structures," AFFDL-TR-73-111, Vol. II, 1974.

[43] Haftka, R.T., "Structural Optimization with Aeroelastic Constraints—A Survey of US Applications," Int. J. Vehicle Design, 7, pp. 381–392, 1986.

[44] Schmit, L.A., and Miura, H., "A New Structural Analysis / Synthesis Capability — Access I, AIAA J., 14 (5), pp. 661–671,1976.

[45] Fleury, C., and Schmit, L.A., "ACCESS 3—Approximation Concepts Code for Efficient Structural Synthesis—User's Guide," NASA CR-159260, September 1980.

[46] Wilkinson, K., et al., "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles, Vol. I—Theory, Vol. II—Program User's Manual," AFFDL-TR-75-137, 1975.

[47] Venkayya, V.B., and Tischler, V.A., "OPSTAT–A Computer Program for Optimal Design of Structures Subjected to Static Loads," AFFDL-TR-79-67,1979.

[48] Khot, N.S., "Computer Program (OPTCOMP) for Optimization of Composite Structures for Minimum Weight Design," AFFDL-TR-76-149, 1977.

[49] Gellatly, R.A., Dupree, D.M., and Berke, L., "OPTIMUM II: A MAGIC Compatible Large Scale Automated Minimum Weight Design Program," AFFDL-TR-74-97, Vols. I and II, 1974.

[50] Isakson, G., and Pardo, H., "ASOP-3: A Program for the Minimum Weight Design of Structures Subjected to Strength and Deflection Constraints," AFFDL-TR-76-157, 1976.

[51] Bartholomew, P., and Wellen, H.K., "Computer Aided Optimization of Aircraft Structures," J. Aircraft, 27 (12), pp. 1079–1086, 1990.

[52] Kiusalaas, J., and Reddy, G.B., "DESAP 2—A Structural Design Program with Stress and Buckling Constraints," NASA CR-2797 to 2799, 1977.

[53] Haftka, R.T., and Prasad, B., "Programs for Analysis and Resizing of Complex Structures," Comput. Struct., 10, pp. 323–330, 1979.

[54] Sobieszczanski-Sobieski, J., and Rogers, J.L., Jr., "A Programming System for Research and Applications in Structural Optimization," Int. Symposium on Optimum Structural Design, Tucson, Arizona, pp. 11-9–11-21, 1981.

[55] Walsh, J.L., "Application of Mathematical Optimization Procedures to a Structural Model of a Large Finite-Element Wing," NASA TM-87597, 1986.

[56] Vanderplaats, G.N., "CONMIN– A Fortran Program for Constrained Function Minimization: User's manual," NASA TM X-62282, 1973.

[57] Bråmå, T., "Applications of Structural Optimization Software in the Design Process," in Computer Aided Optimum Design of Structures: Applications, (Eds, C. A. Brebbia and S. Hernandez), Computational Mechanics Publications, Springer-Verlag, 1989, pp. 13–21.

[58] Neill, D.J., Johnson, E.H., and Canfield, R., "ASTROS—A Multidisciplinary Automated Structural Design Tool," J. Aircraft, 27, 12, pp. 1021–1027, 1990.

[59] Atrek, E., "SHAPE: A Program for Shape Optimization of Continuum Structures," in Computer Aided Optimum Design of Structures: Applications, (Eds, C. A. Brebbia and S. Hernandez), Computational Mechanics Publications, Springer-Verlag, 1989, pp. 135–144.

[60] Hariran, M., Paeng, J.K., and Belsare, S., "STROPT—the Structural Optimization System," Proceedings of the 7th International Conference on Vehicle Structural Mechanics, Detroit, MI, April 11–13, 1988, SAE, pp. 27–38.

[61] Vanderplaats, G.N., Miura, H., Nagendra, G., and Wallerstein, D., " Optimization of Large Scale Structures using MSC/NASTRAN," in Computer Aided Optimum Design of Structures: Applications, (Eds, C. A. Brebbia and S. Hernandez), Computational Mechanics Publications, Springer-Verlag, 1989, pp. 51–68.

[62] Ward, P. and Cobb, W.G.C., "Application of I-DEAS Optimization for the Static and Dynamic Optimization of Engineering Structures," in Computer Aided Optimum Design of Structures: Applications, (Eds, C. A. Brebbia and S. Hernandez), Computational Mechanics Publications, Springer-Verlag, 1989, pp. 33–50.

63] GENESIS User's Manual (version 1.00), VMA Engineering, Goleta, California, September, 1991.

[64] Vanderplaats, G.N., "ADS: A FORTRAN Program for Automated Design Synthesis", VMA Engineering, Inc. Goleta, California, May 1985.

[65] DOT User's Manual (version 2.0B), VMA Engineering, Inc. Goleta, California, Sept. 1990.

[66] DOC User's manual (version 1.00), VMA Engineering, Inc. Goleta, California, March 1991.

[67] Miura, H., and Schmit, L.A., Jr., "NEWSUMT—A Fortran Program for Inequality Constrained Function Minimization—User's Guide," NASA CR-159070, June, 1979.

[68] Grandhi, R.V., Thareja, R., and Haftka, R.T., "NEWSUMT-A: A General Purpose Program for Constrained Optimization Using Constraint Approximations," ASME Journal of Mechanisms, Transmissions and Automation in Design, 107, pp. 94–99, 1985.

[69] Arora, J.S. and Tseng, C.H., "User Manual for IDESIGN: Version 3.5, Optimal Design Laboratory, College of Engineering, The University of Iowa, Iowa City, 1987.

[70] Fleury, C., and Schmit, L.A. Jr., "Dual Methods and Approximation Concepts in Structural Synthesis," NASA CR-3226, December, 1980.

Sensitivity of Discrete Systems 7

The first step in the analysis of a complex structure is spatial discretization of the continuum equations into a finite element, finite difference or a similar model. The analysis problem then requires the solution of algebraic equations (static response), algebraic eigenvalue problems (buckling or vibration) or ordinary differential equations (transient response). The sensitivity calculation is then equivalent to the mathematical problem of obtaining the derivatives of the solutions of those equations with respect to their coefficients. This is the main subject of the present chapter.

In some cases it is advantageous to differentiate the continuum equations governing the structure with respect to design variables before the process of discretization. One advantage is that the resulting sensitivity equations are equally applicable to various analysis techniques, whether finite element, Ritz solution, collocation, etc. This approach is discussed in the next chapter.

As noted in chapter 6, the calculation of the sensitivity of structural response to changes in design variables is often the major computational cost of the optimization process. Therefore, it is important to have efficient algorithms for evaluating these sensitivity derivatives.

The sensitivity of structural response to problem parameters also has other applications. For example, it is usually impossible to know all the parameters of a structural model, such as material properties, loads and dimensions exactly. The sensitivity of the response to small variations in these parameters is essential for calculating the statistical variation in the response of the structure.

The simplest technique for calculating derivatives of response with respect to a design variable is the finite-difference approximation. This technique is often computationally expensive, but is easy to implement and very popular. The efficiency of the analytical methods discussed in the present chapter is measured by comparison to the finite-difference alternative. Unfortunately, finite-difference approximations often have accuracy problems. We begin this chapter with a discussion of these approximations to sensitivity derivatives.

7.1 Finite Difference Approximations

The simplest finite difference approximation is the first-order forward-difference approximation. Given a function $u(x)$ of a design variable x, the forward-difference approximation $\Delta u/\Delta x$ to the derivative du/dx is given as

$$\frac{\Delta u}{\Delta x} = \frac{u(x+\Delta x) - u(x)}{\Delta x} . \tag{7.1.1}$$

Another commonly used finite-difference approximation is the second-order central-difference approximation

$$\frac{\Delta u}{\Delta x} = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} . \tag{7.1.2}$$

It is also possible to employ higher-order finite-difference approximations, but they are rarely used in structural optimization applications because of the associated high computational cost. If we need to find the derivatives of the structural response with respect to n design variables the forward-difference approximation requires n additional analyses, the central-difference approximation $2n$ additional analyses, and higher order approximations are even more expensive.

The key to the selection of the approximation and the step size Δx is an estimate of the required accuracy. This topic is discussed in [1] and [2], and is summarized in the following section.

7.1.1 *Accuracy and Step Size Selection*

Whenever finite-difference formulae are used to approximate derivatives, there are two sources of error: truncation and condition errors. The *truncation error* $e_T(\Delta x)$ is a result of the neglected terms in the Taylor series expansion of the perturbed function. For example, the Taylor series expansion of $u(x+\Delta x)$ can be written as

$$u(x+\Delta x) = u(x) + \Delta x \frac{du}{dx}(x) + \frac{(\Delta x)^2}{2}\frac{d^2u}{dx^2}(x+\zeta\Delta x), \qquad 0 \le \zeta \le 1 . \tag{7.1.3}$$

From Eq. (7.1.3) it follows that the truncation error for the forward-difference approximation is

$$e_T(\Delta x) = \frac{\Delta x}{2}\frac{d^2u}{dx^2}(x+\zeta\Delta x) , \qquad 0 \le \zeta \le 1 . \tag{7.1.4}$$

Similarly, by including one more term in the Taylor series expansion we find that the truncation error for the central difference approximation is

$$e_T(\Delta x) = \frac{\Delta x^2}{6}\frac{d^3u}{dx^3}(x+\zeta\Delta x) , \qquad -1 \le \zeta \le 1 . \tag{7.1.5}$$

The *condition error* is the difference between the numerical evaluation of the function and its exact value. One contribution to the condition error is round-off error in

calculating du/dx from the original and perturbed values of u. This contribution is comparatively small for most computers unless Δx is extremely small. However if $u(x)$ is computed by a lengthy or ill-conditioned numerical process, the round-off contribution to the condition error can be substantial. Additional condition errors may occur if $u(x)$ is calculated by an iterative process which is terminated early. If we have a bound ϵ_u on the absolute error in the computed function u, we can estimate the condition error. For example, for the forward-difference approximation the condition error $e_C(\Delta x)$ is (very!) conservatively estimated from Eq. (7.1.1) as

$$e_C(\Delta x) = \frac{2}{\Delta x}\epsilon_u \,. \tag{7.1.6}$$

Equations (7.1.4) and (7.1.6) present us with the so called "step-size dilemma." If we select the step size to be small, so as to reduce the truncation error, we may have an excessive condition error. In some cases there may not be any step size which yields an acceptable error!

Example 7.1.1

Suppose the function $u(x)$ is defined as the solution of the following two equations

$$\begin{aligned} 101u + xv &= 10\,, \\ xu + 100v &= 10\,, \end{aligned}$$

and let us consider the derivative du/dx evaluated at $x = 100$.

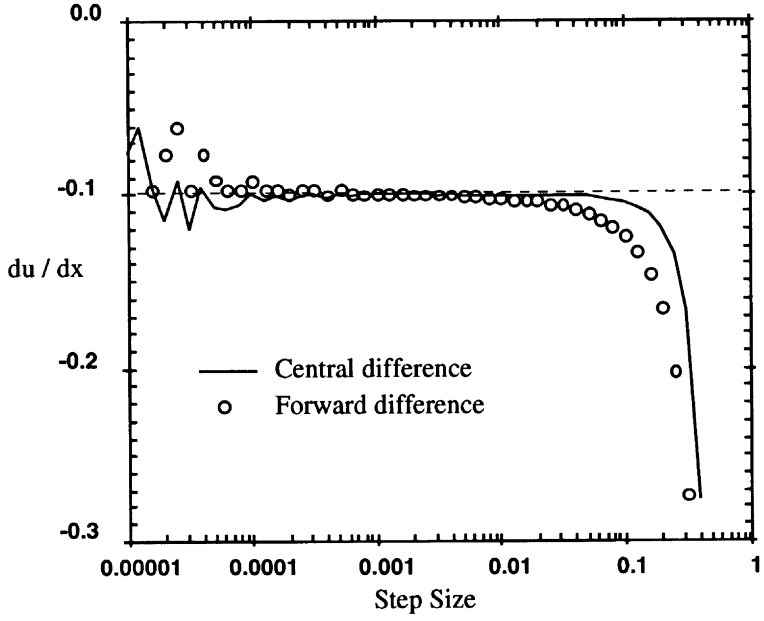


Figure 7.1.1 Effect of step size on derivative.

The solution for u is

$$u = \frac{-10x + 1000}{10100 - x^2},$$

and the exact value of du/dx at $x = 100$ is -0.10. The forward-difference and central-difference derivatives are plotted in Figure 7.1.1 for a range of step sizes. Note that for the very small step sizes the error oscillates because the condition error is not a continuous function. For the higher step sizes the total error is dominated by the truncation error which is a smooth function of the step size. We can change the problem slightly to make it more ill-conditioned, and increase the condition error as follows

$$10001u + xv = 1000,$$
$$xu + 10000v = 1000.$$

The values of the forward- and central-difference approximations at $x = 10000$ are shown in Figure 7.1.2. Now the range of acceptable step sizes is narrowed and we have to use the central-difference approximation if we want to have a reasonable range.•••

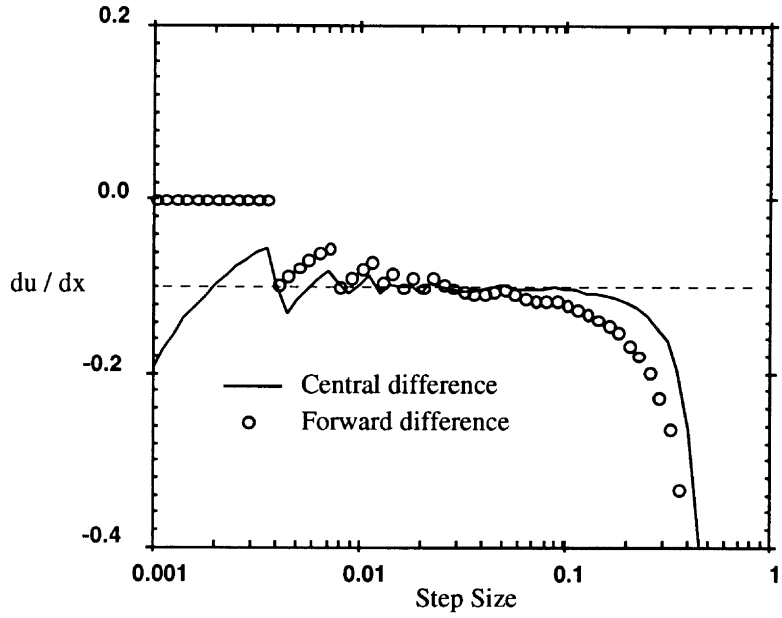


Figure 7.1.2 Effect of step size on derivative.

A bound e on the total error— the sum of the truncation and condition errors— for the forward-difference approximation is obtained from Eqs. (7.1.4) and (7.1.6) as

$$e = \frac{\Delta x}{2}|s_b| + \frac{2}{\Delta x}\epsilon_u, \tag{7.1.7}$$

where s_b is a bound on the second derivative in the interval $[x, x+\Delta x]$. When ϵ_u and s_b are available it is possible to calculate an optimum step-size that minimizes e as

$$\Delta x_{opt} = 2\sqrt{\frac{\epsilon_u}{|s_b|}}. \tag{7.1.8}$$

Procedures for estimating s_b and ϵ_u are given in [1] and [2].

7.1.2 *Iterative Methods*

Condition errors can become important when iterative methods are used for performing some of the calculations. Consider a simple example of a single displacement component u which is obtained by solving a nonlinear algebraic equation which depends on one design variable x

$$f(x,u) = 0. \tag{7.1.9}$$

The solution of Eq. (7.1.9) is obtained by an iterative process which starts with some initial guess of u and terminates when the iterate $\tilde{u}$ is estimated to be within some tolerance ϵ of the exact u (Note that ϵ is a bound on the condition error in u). To calculate the derivative du/dx, assume that we use the forward-difference approximation. That is, we perturb x by Δx and solve Eq. (7.1.9) for u_Δ

$$f(x+\Delta x, u_\Delta) = 0. \tag{7.1.10}$$

The iterative solution of Eq.(7.1.10) yields an approximation $\tilde{u}_\Delta$, and then du/dx is approximated as

$$\frac{du}{dx} \approx \frac{\tilde{u}_\Delta - \tilde{u}}{\Delta x}. \tag{7.1.11}$$

To start the iterative process for obtaining u_Δ, we can use either of two initial guesses. The first is the same initial guess that was used to solve for u. If the convergence of the iterative process is monotonic there is a good chance that when we use Eq. (7.1.11) the errors in $\tilde{u}$ and $\tilde{u}_\Delta$ will almost cancel out, and we will get a very small condition error. The other logical initial guess for u_Δ is $\tilde{u}$. This initial guess is good if Δx is small, and so we may get fast convergence. Unfortunately, this time we cannot expect the condition errors to cancel. As we iterate on $\tilde{u}_\Delta$, the original error (the difference between u and $\tilde{u}$) will be reduced at the same time that the change due to Δx is taking effect. (Consider, for example, what happens if Δx is set to zero, or an extremely small number).

Reference [3] suggests a strategy which allows us to start the iteration for u_Δ from $\tilde{u}$ without worrying about excessive condition errors. The approach is to pretend that $\tilde{u}$ is the exact rather than approximate solution by changing the problem that we want to solve. Indeed, $\tilde{u}$ is the exact solution of

$$f(x,u) - f(x,\tilde{u}) = 0, \tag{7.1.12}$$

which is only slightly different from our original problem (because $f(x,\tilde{u})$ is almost zero). We now find the derivative du/dx from Eq.(7.1.12), by obtaining u_Δ as the solution of

$$f(x+\Delta x, u_\Delta) - f(x,\tilde{u}) = 0. \tag{7.1.13}$$

Because $\tilde{u}$ is the exact solution of this equation for $\Delta x = 0$ the iterative process will only reflect the effect of Δx.

Example 7.1.2

Consider the nonlinear equation

$$f(u, x) = u^2 - x = 0\,,$$

and the iterative solution process

$$u_m = 0.5(u_{m-1} + x/u_{m-1})\,,$$

which is an application of Newton's method to the square-root problem and therefore has quadratic convergence properties.

Table 7.1.1 Iteration history starting with $u = x$

	$x = 1000$		$x + \Delta x = 1000.1$			$x + \Delta x = 1100$		
Iter.	$\tilde{u}$	f	$\tilde{u}_\Delta$	f	$\Delta u/\Delta x$	$\tilde{u}_\Delta$	f	$\Delta u/\Delta x$
0	1000.00	999,000	1000.10	999,000	0.99850	1100.00	1,208,000	1.00000
1	500.500	250,000	500.550	250,000	0.49800	550.500	302,000	0.50000
2	251.249	62,100	251.274	62,100	0.24900	276.249	75,200	0.25000
3	127.615	15,300	127.627	15,300	0.12450	140.115	18,500	0.12500
4	67.7253	3,590	67.7315	3,590	0.06225	73.9380	4,370	0.06258
5	41.2454	701.2	41.2486	701.3	0.03174	44.4256	873.6	0.03180
6	32.7453	72.25	32.7471	72.27	0.01862	34.5930	96.68	0.01848
7	31.6420	1.216	31.6436	1.217	0.01587	33.1957	1.954	0.01553
8	31.6228	-0.005	31.6244	0.000	0.01587	33.1663	0.0007	0.01543

Exact values $u(x = 1000) = 31.6228$; $du/dx = 0.01581$

Table 7.1.1 shows the convergence of u for $x = 1000$, $x = 1000.1$ and $x = 1100$, and the estimate of the derivative du/dx at $x = 1000$. The first guess for u is taken to be x in all three cases. Note that far from the solution the convergence is slow with the error being halved at each iteration. As the error gets smaller the convergence rate increases. It is seen that the convergence of the derivative is slightly slower than that of u. Also, we do not see that the small Δx leads to any large condition errors as compared to the large Δx. This is due to the monotonic convergence and the resulting cancellation of condition errors.

Now we switch the first guess of the perturbed solution to an iterate of the nominal one. Starting the perturbed solution from a good approximation to the nominal solution we obtain fast convergence; usually we need only one or two iterations. Therefore, the value of the finite-difference derivative remains virtually constant after the first two iterations. Table 7.1.2 shows the second iterate u_2 obtained when the perturbed solution is started from each of the last four iterates of the nominal solution given in Table 7.1.1.

Inspection of Table 7.1.2 shows that, because the perturbed solution is more accurate than the nominal one, the derivative obtained by finite differences is erroneous,

Table 7.1.2 Effect of starting u_Δ from u_0

	$x + \Delta x = 1000.1$		$x + \Delta x = 1100$	
$u_0^\dagger$	u_2	$\Delta u/\Delta x$	u_2	$\Delta u/\Delta x$
41.2454	31.6436	-96.0181	33.1755	-0.08070
32.7453	31.6244	-11.2093	33.1662	0.00421
31.6420	31.6243	- 0.1772	33.1663	0.01524
31.6228	31.6243	0.01572	33.1663	0.01543

$^\dagger u_0$ are iterates from Table 7.1.1.

except at very high accuracies (low ϵ). The effect of the finite difference increment Δx is also evident. The errors for the small Δx are larger than for the larger Δx, except when u_0 has fully converged (so that there is no condition error).

We now use the approach of 7.1.13, replacing the original equation by

$$u^2 - x - \bar{f} = 0\,,$$

where $\bar{f}$ is the residual of the last iterate of the nominal solution. That is, for the perturbed solution we try to calculate the root of $x + \bar{f}$ instead of x. The results of the modified calculation are shown in Table 7.1.3. We can now get a reasonable approximation to the derivative in two iterations.• • •

Table 7.1.3 Modified derivative calculation

	$x + \Delta x = 1100$		$x + \Delta x = 1000.1$	
u_0	u_2	$\Delta u/\Delta x$	u_2	$\Delta u/\Delta x$
41.2454	42.4404	0.01195	41.2466	0.01205
32.7453	34.2382	0.01493	32.7468	0.01511
31.6420	33.1846	0.01543	31.6436	0.01572
31.6228	33.1663	0.01543	31.6243	0.01572

Cost and accuracy considerations often dictate that we avoid the use of finite-difference derivatives. For static displacement and stress constraints analytical derivatives are fairly easy to get, as discussed in the next section.

7.1.3 *Effect of Derivative Magnitude on Accuracy*

It is well known that small displacements and stresses are not calculated as accurately as large stresses and displacements. The same applies to derivatives. When both the function u and the variable x are positive, the relative magnitude of the derivative can be estimated from the logarithmic derivative

$$\frac{d_l u}{dx} = \frac{d(\log u)}{d(\log x)} = \frac{du/u}{dx/x}\,. \tag{7.1.14}$$

The logarithmic derivative gives the percentage change in u due to a percent change in x. Therefore, when the logarithmic derivative is larger than unity the relative change

in u is larger than the relative change in x and the derivative can be considered to be large. When the logarithmic derivative is much smaller than unity, the relative change in u is much smaller than the relative change in x. In this case the derivative is considered to be small, and in general, it would be difficult to evaluate it accurately using finite-difference differentiation (or any other procedure subject to condition or truncation errors). Fortunately, when the logarithmic derivative is small it is usually not important to evaluate it accurately, because its influence on the optimization process is small.

The logarithmic derivative can be misleading when a variable is about to change sign so that it is very small in magnitude. In that case we recommend using typical values of u and x instead of local values. That is, we define a modified logarithmic derivative $d_{lm}u/dx$ as

$$\frac{d_{lm}u}{dx} = \frac{du/u_t}{dx/x_t}, \tag{7.1.15}$$

where x_t and u_t are representative values of the variable and the function, respectively.

Example 7.1.3

The increased error associated with small derivatives is demonstrated in the following simple design problem. We consider the design of a submerged beam of rectangular cross section so as to minimize the perimeter of the cross section (so as to reduce corrosion damage). The beam is subject to a bending moment M and we require the maximum bending stress to be less than the allowable stress σ_0. The design variables are the width b and height h of the rectangular cross-section. The problem can be formulated as

$$\begin{aligned} &\textbf{minimize} && 2(b+h)\,, \\ &\textbf{such that} && \frac{6M}{bh^2} \le \sigma_0\,. \end{aligned}$$

We nondimensionalize the problem by defining a characteristic length l and using it to define new design variables x_1 and x_2 as

$$l = (6M/\sigma_0)^{1/3}, \qquad x_1 = b/l, \qquad x_2 = h/l\,.$$

In terms of the new variables the problem can be reformulated as

$$\begin{aligned} &\textbf{minimize} && u = x_1 + x_2\,, \\ &\textbf{such that} && \frac{1}{x_1 x_2^2} = 1\,, \end{aligned}$$

where the inequality has been replaced by an equality because it is clear that the stress constraint will be active (otherwise the solution is $b = h = 0$). The equality can be used to eliminate x_1, so that the objective function can be written as

$$u = 1/x_2^2 + x_2\,.$$

We now consider the calculation of the derivative by finite differences at two points; at an initial design where $x_2 = 1$, and near the optimum, at $x_2 = 1.29$. In both cases we use forward differences with $\Delta x_2 = 0.01$. At $x_2 = 1$ we get

$$\frac{\Delta u}{\Delta x_2} = \frac{1/1.01^2 + 1.01 - 2}{0.01} = -0.970\,,$$

which is 3 percent off the exact value of the derivative $du/dx_2 = -1.0$. However, at $x_2 = 1.29$ we get

$$\frac{\Delta u}{\Delta x_2} = \frac{1/1.30^2 + 1.30 - (1/1.29^2 + 1.29)}{0.01} = 0.0791\,,$$

which is 16 percent off the exact value of 0.0683. The logarithmic derivative can warn us that we should expect the large relative error in the second case. Indeed, for $x_2 = 1$, we have $u = 2.0$, and the logarithmic derivative is estimated from the finite difference derivative to be

$$\frac{d_l u}{dx_2} \approx \frac{\Delta_l u}{\Delta x_2} = \frac{\Delta u}{\Delta x_2}\frac{x_2}{u} = -0.97 \times 1/2 = -0.485\,.$$

At $x_2 = 1.29$ we have $u = 1.891$ and

$$\frac{d_l u}{dx_2} \approx \frac{\Delta_l u}{\Delta x_2} = \frac{\Delta u}{\Delta x_2}\frac{x_2}{u} = 0.0791 \times 1.29/1.891 = 0.054\,,$$

so that the logarithmic derivative is indeed quite small.• • •

7.2 Sensitivity Derivatives of Static Displacement and Stress Constraints

7.2.1 *Analytical First Derivatives*

The equations of equilibrium in terms of the nodal displacement vector $\mathbf{u}$ are generated from a finite element model in the form

$$\mathbf{K}\mathbf{u} = \mathbf{f}\,, \tag{7.2.1}$$

where $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}$ is a load vector. A typical constraint, involving a limit on a displacement or a stress component, may be written as

$$g(\mathbf{u}, x) \geq 0\,, \tag{7.2.2}$$

where, for the sake of simplified notation, it is assumed that g depends on only a single design variable x. Using the chain rule of differentiation, we obtain

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \mathbf{z}^T\frac{d\mathbf{u}}{dx}\,, \tag{7.2.3}$$

where $\mathbf{z}$ is a vector with components

$$z_i = \frac{\partial g}{\partial u_i}. \tag{7.2.4}$$

Note that we use the notation dg/dx to denote the total derivative of g with respect to x. This total derivative includes the explicit part $\partial g/\partial x$ plus the implicit part through the dependence on $\mathbf{u}$. The explicit part of the derivative is usually zero or easy to obtain, so we discuss only the computation of the implicit part. Differentiating Eq. (7.2.1) with respect to x we obtain

$$\mathbf{K}\frac{d\mathbf{u}}{dx} = \frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u}. \tag{7.2.5}$$

Premultiplying Eq. (7.2.5) by $\mathbf{z}^T\mathbf{K}^{-1}$ obtain

$$\mathbf{z}^T\frac{d\mathbf{u}}{dx} = \mathbf{z}^T\mathbf{K}^{-1}(\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u}). \tag{7.2.6}$$

Numerically, the calculation of $\mathbf{z}^T d\mathbf{u}/dx$ may be performed in two ways. The first, called the *direct method*, consists of solving Eq. (7.2.5) for $d\mathbf{u}/dx$ and then taking the scalar product with $\mathbf{z}$. The second approach, called the *adjoint method*, defines an adjoint vector $\boldsymbol{\lambda}$ which is the solution of the system

$$\mathbf{K}\boldsymbol{\lambda} = \mathbf{z}, \tag{7.2.7}$$

and then we write Eq. (7.2.3)as

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \boldsymbol{\lambda}^T(\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u}), \tag{7.2.8}$$

where we have used the symmetry of $\mathbf{K}$.

The solution of Eq. (7.2.7) for $\boldsymbol{\lambda}$ is similar to a solution for displacement under a load vector $\mathbf{z}$. The adjoint method is also known as the *dummy-load method* because $\mathbf{z}$ is often described as a dummy load. When g in Eq. (7.2.2) is an upper limit on a single displacement component, the dummy load also has a single nonzero component corresponding to the constrained displacement component. Similarly, when g is an upper limit on the stress in a truss member, the dummy load is composed of a pair of equal and opposite forces acting on the two ends of the member.

For this case of static response the derivation of the adjoint technique is very simple. However the technique will be used in many other cases where we will want to calculate the derivative of a constraint without having to calculate first the derivative of the response $\mathbf{u}$. We repeat the derivation of the adjoint method in a procedure that is applicable to the general case. This procedure consists of adding the derivative of the equations of equilibrium multiplied by a Lagrange multiplier to the derivative of the constraint. The Lagrange multiplier, which is equal to the adjoint vector, is then

selected to satisfy equations that lead to elimination of the derivative of the response. For the present case we rewrite Eq. (7.2.3) as

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \mathbf{z}^T \frac{d\mathbf{u}}{dx} + \boldsymbol{\lambda}^T \left(\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u} - \mathbf{K}\frac{d\mathbf{u}}{dx} \right), \tag{7.2.9}$$

where the additional term is the adjoint vector times the derivative of the equations of equilibrium. Rearranging the terms in Eq. (7.2.9) we have

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + (\mathbf{z}^T - \boldsymbol{\lambda}^T\mathbf{K})\frac{d\mathbf{u}}{dx} + \boldsymbol{\lambda}^T \left(\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u} \right). \tag{7.2.10}$$

If we want to eliminate $d\mathbf{u}/dx$ from this expression we need to select $\boldsymbol{\lambda}$ so as to eliminate its coefficient, which gives us Eq. (7.2.7) for $\boldsymbol{\lambda}$. The remaining terms are the same as Eq. (7.2.8) for the derivative of the constraint.

Example 7.2.1

In this example, we calculate the sensitivity derivative of a constraint on the tip displacement of a stepped cantilever beam with respect to the moment of inertia I_1 and the length l_1.

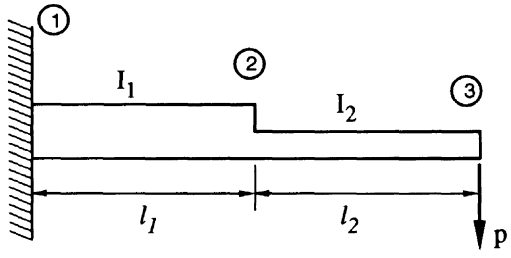


Figure 7.2.1 Beam example for derivatives of static response.

The constraint on the tip displacement is posed as

$$g = c - w_{tip} \geq 0\,.$$

The problem is simple and has an analytical solution based on elementary beam theory, namely

$$w_{tip} = \frac{p}{3EI_1}(l_1^3 + 3l_1^2 l_2 + 3l_1 l_2^2) + \frac{pl_2^3}{3EI_2}\,,$$

so that

$$\frac{\partial g}{\partial I_1} = \frac{p}{3EI_1^2}(l_1^3 + 3l_1^2 l_2 + 3l_1 l_2^2)\,,$$

$$\frac{\partial g}{\partial l_1} = -\frac{p}{3EI_1}(3l_1^2 + 6l_1 l_2 + 3l_2^2) = -\frac{p}{EI_1}(l_1 + l_2)^2\,.$$

The finite element solution is based on a standard cubic beam element, with one element used for each section. We denote the displacement and rotation at the ith node by w_i and θ_i, respectively. The element stiffness matrix is

$$\mathbf{K}^e = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix},$$

so that the global stiffness matrix, corresponding to degrees of freedom w_2, θ_2, w_3, θ_3, is

$$\mathbf{K} = E\begin{bmatrix} 12(I_1/l_1^3 + I_2/l_2^3) & -6(I_1/l_1^2 - I_2/l_2^2) & -12I_2/l_2^3 & 6I_2/l_2^2 \\ & 4(I_1/l_1 + I_2/l_2) & -6I_2/l_2^2 & 2I_2/l_2 \\ & & 12I_2/l_2^3 & -6I_2/l_2^2 \\ \multicolumn{2}{c}{sym} & & 4I_2/l_2 \end{bmatrix}.$$

The load vector $\mathbf{f} = [0, 0, p, 0]^T$, and the solution for the displacement vector is

$$\mathbf{u} = \begin{Bmatrix} w_2 \\ \theta_2 \\ w_3 \\ \theta_3 \end{Bmatrix} = \left(\frac{p}{E}\right)\begin{Bmatrix} l_1^3/3I_1 + l_1^2 l_2/2I_1 \\ l_1^2/2I_1 + l_1 l_2/I_1 \\ (l_1^3 + 3l_1^2 l_2 + 3l_1 l_2^2)/3I_1 + l_2^3/3I_2 \\ l_1^2/2I_1 + l_1 l_2/I_1 + l_2^2/I_2 \end{Bmatrix}.$$

We first use analytical methods for the derivative calculation, so that we need $(\partial\mathbf{K}/\partial I_1)\mathbf{u}$ and $(\partial\mathbf{K}/\partial l_1)\mathbf{u}$

$$\frac{\partial\mathbf{K}}{\partial I_1}\mathbf{u} = \left(\frac{E}{l_1^3}\right)\begin{bmatrix} 12 & -6l_1 & 0 & 0 \\ -6l_1 & 4l_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} w_2 \\ \theta_2 \\ w_3 \\ \theta_3 \end{Bmatrix} = \left(\frac{E}{l_1^3}\right)\begin{Bmatrix} 12w_2 - 6l_1\theta_2 \\ -6l_1 w_2 + 4l_1^2\theta_2 \\ 0 \\ 0 \end{Bmatrix}$$

$$= \left(\frac{p}{I_1}\right)\begin{Bmatrix} 1 \\ l_2 \\ 0 \\ 0 \end{Bmatrix},$$

where the solution for w_2 and θ_2 was used. Similarly,

$$\frac{\partial\mathbf{K}}{\partial l_1}\mathbf{u} = \left(\frac{EI_1}{l_1^4}\right)\begin{bmatrix} -36 & 12l_1 & 0 & 0 \\ 12l_1 & -4l_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} w_2 \\ \theta_2 \\ w_3 \\ \theta_3 \end{Bmatrix} = \left(\frac{4EI_1}{l_1^4}\right)\begin{Bmatrix} -9w_2 + 3l_1\theta_2 \\ 3l_1 w_2 - l_1^2\theta_2 \\ 0 \\ 0 \end{Bmatrix}$$

$$= \left(\frac{p}{l_1}\right)\begin{Bmatrix} -6(1 + l_2/l_1) \\ 2(l_1 + l_2) \\ 0 \\ 0 \end{Bmatrix}.$$

In the direct method

$$\frac{\partial\mathbf{u}}{\partial I_1} = \mathbf{K}^{-1}\left[\frac{\partial\mathbf{f}}{\partial I_1} - \frac{\partial\mathbf{K}}{\partial I_1}\mathbf{u}\right],$$

or

$$\frac{\partial}{\partial I_1}\begin{Bmatrix} w_2 \\ \theta_2 \\ w_3 \\ \theta_3 \end{Bmatrix} = \mathbf{K}^{-1}\begin{Bmatrix} p/I_1 \\ pl_2/I_1 \\ 0 \\ 0 \end{Bmatrix} = -\frac{p}{EI_1^2}\begin{Bmatrix} l_1^2 l_2/2 + l_1^3/3 \\ l_1 l_2 + l_1^2/2 \\ l_1^2 l_2 + l_1 l_2^2 + l_1^3/3 \\ l_1 l_2 + l_1^2/2 \end{Bmatrix},$$

so that $\partial g/\partial I_1 = -\partial w_3/\partial I_1$, which agrees with the beam-theory result.

Similarly

$$\frac{\partial \mathbf{u}}{\partial l_1} = \mathbf{K}^{-1}[\frac{\partial \mathbf{f}}{\partial l_1} - \frac{\partial \mathbf{K}}{\partial l_1}\mathbf{u}],$$

or

$$\frac{\partial}{\partial l_1}\begin{Bmatrix} w_2 \\ \theta_2 \\ w_3 \\ \theta_3 \end{Bmatrix} = -\mathbf{K}^{-1}\begin{Bmatrix} -(6p/l_1)(1 + l_2/l_1) \\ (2p/l_1)(l_1 + l_2) \\ 0 \\ 0 \end{Bmatrix} = (\frac{p}{EI_1})\begin{Bmatrix} l_1^2 + l_1 l_2 \\ l_1 + l_2 \\ (l_1 + l_2)^2 \\ l_1 + l_2 \end{Bmatrix},$$

so that $\partial g/\partial l_1 = -\partial w_3/\partial l_1$, again agreeing with the beam-theory result.

In the adjoint method, $\mathbf{z}^T = -\partial w_{tip}/\partial \mathbf{u} = [0, 0, -1, 0]$, and we can solve for the adjoint vector

$$\boldsymbol{\lambda} = \mathbf{K}^{-1}\mathbf{z} = \mathbf{K}^{-1}\begin{Bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{Bmatrix} = (-\frac{1}{E})\begin{Bmatrix} l_1^3/3I_1 + l_1^2 l_2/2I_1 \\ l_1^2/2I_1 + l_1 l_2/I_1 \\ (l_1^3 + 3l_1^2 l_2 + 3l_1 l_2^2)/3I_1 + l_2^3/3I_2 \\ l_1^2/2I_1 + l_1 l_2/I_1 + l_2^2/I_2 \end{Bmatrix},$$

so that from Eq. (7.2.8)

$$\frac{\partial g}{\partial I_1} = -\boldsymbol{\lambda}^T\frac{\partial \mathbf{K}}{\partial I_1}\mathbf{u} = \frac{p}{EI_1}(\frac{l_1^3}{3I_1} + \frac{l_1^2 l_2}{2I_1} + \frac{l_1^2 l_2}{2I_1} + \frac{l_1 l_2^2}{I_1}) = \frac{p}{EI_1^2}(l_1^2 l_2 + l_1 l_2^2 + l_1^3/3),$$

and

$$\frac{\partial g}{\partial l_1} = -\boldsymbol{\lambda}^T\frac{\partial \mathbf{K}}{\partial l_1}\mathbf{u} = \frac{p}{El_1}(l_1 + l_2)(-\frac{2l_1^2}{I_1} - \frac{3l_1}{I_1} + \frac{l_1^2}{I_1} + \frac{2l_1 l_2}{I_1}) = \frac{p}{EI_1}(l_1 + l_2)^2.$$

• • •

The difference between the computational effort associated with the direct method and with the adjoint method depends on the relative number of constraints and design variables. The direct method requires the solution of Eq. (7.2.5) once for each design variable, while the adjoint method requires the solution of Eq. (7.2.7) once for each constraint. Thus the direct method is the more efficient when the number of design variables is smaller than the number of displacement and stress constraints that need to be differentiated. The adjoint method is more efficient when the number of design variables is larger than the number of these constraints.

In practical design situations we usually have to consider several load cases. The effort associated with the direct method is approximately proportional to the number of load cases. The number of critical constraints at the optimum design, on the other

hand, is usually less than the number of design variables. Therefore, in a multiple-load-case situation the adjoint method becomes more attractive.

Both the direct and adjoint methods require the solution of a system of equations as the major part of the computational effort. However, the factored form of the matrix $\mathbf{K}$ of the equations is usually available from the solution of Eq. (7.2.1) for the displacements. The solution for $d\mathbf{u}/dx$ or $\boldsymbol{\lambda}$ is therefore much cheaper than the original solution of Eq. (7.2.1). This provides the major computational advantage of these two analytical methods over the finite-difference calculation of the derivatives. For example, the forward difference approximation to $d\mathbf{u}/dx$

$$\frac{d\mathbf{u}}{dx} \approx \frac{\mathbf{u}(x+\Delta x) - \mathbf{u}(x)}{\Delta x} \tag{7.2.11}$$

requires the evaluation of $\mathbf{u}(x+\Delta x)$ by re-assembling the stiffness matrix and load vector at the perturbed design and solving

$$\mathbf{K}(x+\Delta x)\mathbf{u}(x+\Delta x) = \mathbf{f}(x+\Delta x)\,. \tag{7.2.12}$$

The required factorization of $\mathbf{K}(x+\Delta x)$ is typically much more expensive than a solution for another right hand side with the already factored $\mathbf{K}(x)$ in Eqs. (7.2.5) and (7.2.7). The advantage of the analytical methods over the finite-difference approximation becomes very pronounced for a large number of design variables.

7.2.2 *Second Derivatives*

In some applications (e.g., calculation of sensitivity of optimum solutions, see Section 5.4) we also need second derivatives of constraint functions with respect to the design variables. In the following we obtain expressions for evaluating $d^2g/dxdy$ where x and y are design variables. For the sake of simplicity we assume that the constraint function g is not an explicit function of the design variables, so that $\partial g/\partial x$ and $\partial g/\partial y$ are zero. More general expressions are to be found in [4].

As in the case of first derivatives we have a direct method and an adjoint method for obtaining second derivatives. The direct method starts by differentiating Eq. (7.2.3) with respect to y

$$\frac{d^2g}{dxdy} = \mathbf{z}^T\frac{d^2\mathbf{u}}{dxdy} + (\frac{d\mathbf{u}}{dx})^T\mathbf{R}\frac{d\mathbf{u}}{dy}\,, \tag{7.2.13}$$

where $\mathbf{R}$ is the matrix of second derivatives of g with respect to $\mathbf{u}$, that is

$$r_{ij} = \frac{\partial^2 g}{\partial u_i \partial u_j}\,. \tag{7.2.14}$$

We obtain the second derivative of the displacement field by differentiating Eq. (7.2.5)

$$\mathbf{K}\frac{d^2\mathbf{u}}{dxdy} = \frac{d^2\mathbf{f}}{dxdy} - \frac{d^2\mathbf{K}}{dxdy}\mathbf{u} - \frac{d\mathbf{K}}{dx}\frac{d\mathbf{u}}{dy} - \frac{d\mathbf{K}}{dy}\frac{d\mathbf{u}}{dx}\,. \tag{7.2.15}$$

Solving Eq. (7.2.5) for $d\mathbf{u}/dx$, a similar equation for $d\mathbf{u}/dy$, and Eq. (7.2.15) for $d^2\mathbf{u}/dxdy$ we finally substitute into Eq. (7.2.13).

The adjoint method starts by differentiating Eq. (7.2.8) with respect to y

$$\frac{d^2g}{dxdy} = (\frac{d\boldsymbol{\lambda}}{dy})^T(\frac{\partial \mathbf{f}}{\partial x} - \frac{d\mathbf{K}}{dx}\mathbf{u}) + \boldsymbol{\lambda}^T(\frac{d^2\mathbf{f}}{dxdy} - \frac{d^2\mathbf{K}}{dxdy}\mathbf{u} - \frac{d\mathbf{K}}{dx}\frac{d\mathbf{u}}{dy}). \tag{7.2.16}$$

To evaluate the first term we differentiate Eq. (7.2.7) with respect to y

$$\mathbf{K}\frac{d\boldsymbol{\lambda}}{dy} = \mathbf{R}\frac{d\mathbf{u}}{dy} - \frac{d\mathbf{K}}{dy}\boldsymbol{\lambda}. \tag{7.2.17}$$

Using Eqs. (7.2.5) and (7.2.17), Eq. (7.2.13) becomes

$$\frac{d^2g}{dxdy} = (\frac{d\mathbf{u}}{dy})^T\mathbf{R}\frac{d\mathbf{u}}{dx} - \boldsymbol{\lambda}^T(\frac{d\mathbf{K}}{dy}\frac{d\mathbf{u}}{dx} + \frac{d\mathbf{K}}{dx}\frac{d\mathbf{u}}{dy} - \frac{d^2\mathbf{f}}{dxdy} + \frac{d^2\mathbf{K}}{dxdy}\mathbf{u}). \tag{7.2.18}$$

In this case the adjoint method is always more efficient than the direct method. Assume that we have n design variables and m constraint functions. The direct method requires as its major computational effort the solution of Eq. (7.2.5) n times, and the solution of Eq. (7.2.15) $n(n+1)/2$ times. The adjoint method, on the other hand, requires the solution of Eq. (7.2.5) n times for the first derivatives, and the solution of Eq. (7.2.7) m times for the adjoint vectors.

7.2.3 The Semi-Analytical Method

Both the direct and adjoint methods require the derivatives of the stiffness matrix and load vectors with respect to design variables. These derivatives are often difficult to calculate analytically, especially for shape design variables which change element geometry. For this reason a semi-analytical approach, where the derivatives of the stiffness matrix and load vector are approximated by finite differences, is popular. Typically, these derivatives are calculated by the first-order forward difference approximation, so that $d\mathbf{K}/dx$ is approximated as

$$\frac{d\mathbf{K}}{dx} \approx \frac{\mathbf{K}(x+\Delta x) - \mathbf{K}(x)}{\Delta x}. \tag{7.2.19}$$

However, while the semi-analytical method is as efficient as the analytical direct or adjoint methods, it is based on finite-difference approximations, and may have accuracy problems. Such accuracy problems can be particularly serious for derivatives of beam and plate structures response with respect to geometrical parameters.

The accuracy problem was observed first in Ref. [5] for the car model shown in Fig. (7.2.2) made of beam elements. The semi-analytical method was used successfully for all section size and most geometrical design variables. However, for some of the derivatives with respect to the overall length dimensions of the car, there were serious accuracy problems.

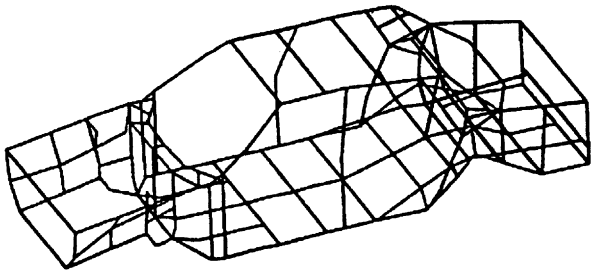


Figure 7.2.2 Stick model of a Car.

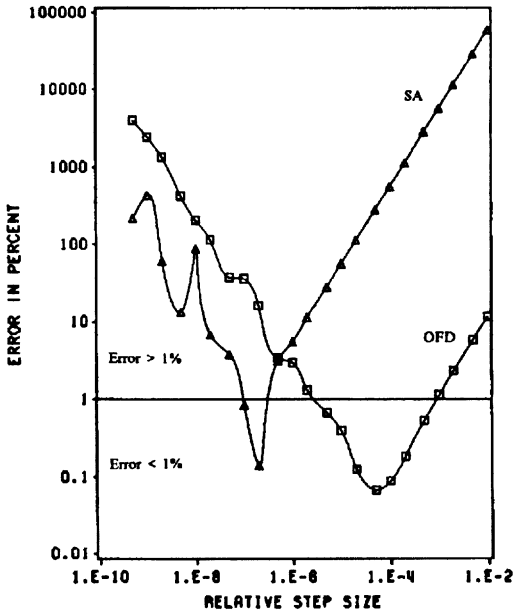


Figure 7.2.3 Errors in the derivative of the strain energy with respect to a length variable of the stick model for overall-finite-differences (OFD) and semi-analytical (SA) methods.

Figure (7.2.3) shows the dependence of the relative error of the derivative of the strain energy of the model with respect to one length variable in the semi-analytical (SA) method and the overall finite difference (OFD) approach. For large step sizes, the OFD method has smaller error (mostly truncation error) than the SA method. The step-size range for which the approximate derivative has an error less than 1%

is much larger for the OFD than for the SA approximation. For small step sizes the OFD method has a larger error (mostly condition error) than the SA method. Figure (7.2.3) shows that, for a relative step size of 10^{-7}, the SA method approximates well the derivative. For some variables, however, there was no step size giving accurate derivatives! To solve the accuracy problem the central difference approximation to the derivative of the stiffness matrix had to be used, which increased substantially the computational cost.

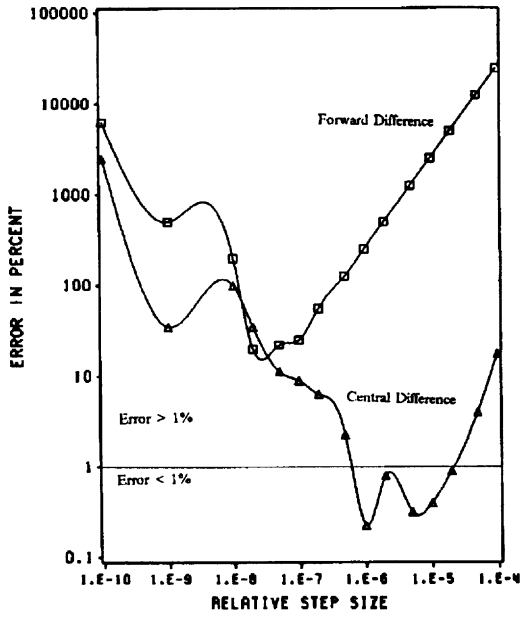


Figure 7.2.4 Forward- and central-difference SA approximation of the derivative of the strain energy with respect to a second length variable of car stick model.

Figure (7.2.4) compares the forward- and central-difference approximations of the derivative with respect to a second length variable. We can clarify the cause of the high truncation errors associated with the semi-analytical method by considering Eq. (7.2.5) carefully. The right hand side of the equation, sometimes referred to as the pseudo load, is the 'load' that has to be applied to the structure to produce a displacement field $d\mathbf{u}/dx$. For beam and plate structures the derivative of the displacement field with respect to geometrical variables is usually not a legitimate displacement field (for example, it may grossly violate the Kirchhoff assumption). The finite element approximation to this illegitimate field is a valid, though highly unusual, displacement field, which requires large self-cancelling components in the pseudo load. As the finite-element mesh is refined, the pseudo load required to generate $d\mathbf{u}/dx$ acquires ever larger self-cancelling components. Thus the errors in the pseudo load due to the finite difference derivative of the stiffness matrix can be greatly magnified.

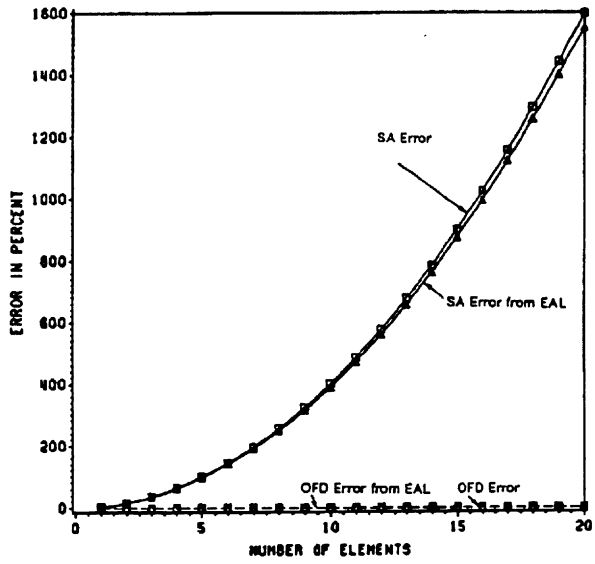


Figure 7.2.5 Errors in the semi-analytical (SA) and overall-finite-difference (OFD) approximations to the derivative of tip displacement with respect to cantilever beam length (one percent step size).

This phenomenon is demonstrated in Fig. (7.2.5) which shows that the error in the derivative of the tip displacement of a cantilever beam with respect to the length of the beam greatly increases as the finite-element mesh is refined.

When a beam or a plate structure is modeled by more general elements, such as three dimensional elements, mesh refinement is no problem. However, as the beam becomes more slender or the plate thinner, the displacement-derivative field becomes more and more incompatible with the geometry, and the same accuracy problems ensue. Reference [6] reports very large errors for beams modeled by truss, plane-stress and solid elements for slenderness ratios larger than ten.

Example 7.2.2

We repeat the calculation of derivatives in Example 7.2.1 to compare the errors associated with the finite-difference and semi-analytical methods. Using forward differences we find

$$\frac{\partial g}{\partial I_1} \approx -\frac{w_{tip}(I_1 + \Delta I_1) - w_{tip}(I_1)}{\Delta I_1} ,$$

the truncation error, e_T, given by Eq. (7.1.4) is approximately

$$e_T = -\frac{\partial^2 w_{tip}}{\partial I_1^2}\frac{\Delta I_1}{2} = -\frac{p}{3EI_1^3}(l_1^3 + 3l_1^2 l_2 + 3l_1 l_2^2)\Delta I_1 ,$$

and the relative truncation error is

$$\frac{e_T}{\dfrac{\partial g}{\partial I_1}} = -\frac{\Delta I_1}{I_1} ,$$

Therefore, it is enough to take $\Delta I_1/I_1 = 10^{-3}$ to get a negligible truncation error. Similarly, the truncation error for the derivative with respect to l_1 is approximately

$$e_T = -\frac{\partial^2 w_{tip}}{\partial l_1^2}\frac{\Delta l_1}{2} = -\frac{p}{EI_1}(l_1 + l_2)\Delta l_1 ,$$

$$\frac{e_T}{\dfrac{\partial g}{\partial l_1}} = \frac{\Delta l_1}{l_1 + l_2} ,$$

and it is enough to take a perturbation in l_1 to be $0.001 l_1$. The error analysis for the semi-analytical method is more complicated. The derivative with respect to the moment of inertia is approximated as

$$\frac{\partial g}{\partial I_1} \approx \boldsymbol{\lambda}^T \frac{\mathbf{K}(I_1 + \Delta I_1) - \mathbf{K}(I_1)}{\Delta I_1}\mathbf{u} ,$$

and the truncation error vanishes

$$e_T = \frac{\Delta I_1}{2}\boldsymbol{\lambda}^T \frac{\partial^2 \mathbf{K}}{\partial I_1^2}\mathbf{u} = 0 ,$$

because $\mathbf{K}$ is a linear function of I_1. The situation is not as good for the truncation error $\partial g/\partial l_1$ which is approximately

$$e_T = \frac{\Delta l_1}{2}\boldsymbol{\lambda}^T \frac{\partial^2 \mathbf{K}}{\partial l_1^2}\mathbf{u} = \frac{p\Delta l_1}{EI_1 l_1}(3l_1^2 + 7l_1 l_2 + 4l_2^2) ,$$

so that the relative error is

$$\frac{e_T}{\dfrac{\partial g}{\partial l_1}} = -\frac{3l_1^2 + 7l_1 l_2 + 4l_2^2}{(l_1 + l_2)^2}\frac{\Delta l_1}{l_1} .$$

Comparing the semi-analytical error to the one obtained by the finite difference approach, we note that it is seven times larger when $l_1 = l_2$. As shown in Ref. [7], this larger error for the semi-analytical method increases as the mesh is refined.• • •

7.2.4 *Nonlinear Analysis*

For nonlinear analysis, the equations of equilibrium may be written as

$$\mathbf{f}(\mathbf{u}, x) = \mu \mathbf{p}(x) , \tag{7.2.20}$$

where $\mathbf{f}$ is the internal force generated by the deformation of the structure, and $\mu\mathbf{p}$ is the external applied load. The load scaling factor μ is used in nonlinear analysis procedures for tracking the evolution of the solution as the load is increased. This is useful because the equations of equilibrium may have several solutions for the same applied loads. By increasing μ gradually we make sure that we obtain the solution that corresponds to the structure being loaded from zero.

Differentiating Eq. (7.2.20) with respect to the design variable x we obtain

$$\mathbf{J}\frac{d\mathbf{u}}{dx} = \mu\frac{d\mathbf{p}}{dx} - \frac{\partial\mathbf{f}}{\partial x}\,, \tag{7.2.21}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$ at $\mathbf{u}$,

$$J_{kl} = \frac{\partial f_k}{\partial u_l}\,, \tag{7.2.22}$$

often called the tangential stiffness matrix.

The direct method for obtaining dg/dx is to solve Eq. (7.2.21) for $d\mathbf{u}/dx$ and substitute into Eq. (7.2.3). The matrix $\mathbf{J}$ is often available from the solution of the equations of equilibrium when these are solved by using Newton's method. Newton's method is based on a linear approximation of the equations of equilibrium about a trial solution $\tilde{\mathbf{u}}$

$$\mathbf{f}(\tilde{\mathbf{u}}, x) + \mathbf{J}(\tilde{\mathbf{u}}, x)(\mathbf{u} - \tilde{\mathbf{u}}) \approx \mu\mathbf{p}(x)\,. \tag{7.2.23}$$

Equation (7.2.23), solved for $\mathbf{u}$, typically provides a better approximation to $\mathbf{u}$ than $\tilde{\mathbf{u}}$. This new approximation replaces $\tilde{\mathbf{u}}$ in Eq. (7.2.23) for the next iteration, either with an updated value of $\mathbf{J}$ (Newton's method) or with the old value (modified Newton's method). The iteration continues until convergence to a desired accuracy is achieved. If the last iterate $\tilde{\mathbf{u}}$, for which $\mathbf{J}$ was calculated, is close enough to $\mathbf{u}$, then that $\mathbf{J}$ can be used for calculating the derivative of $\mathbf{u}$.

The adjoint approach is very similar to that used in the linear case. The adjoint vector $\boldsymbol{\lambda}$ is the solution of the equation

$$\mathbf{J}^T\boldsymbol{\lambda} = \mathbf{z}\,, \tag{7.2.24}$$

where again $\mathbf{z}$ is the vector of derivatives of the constraint with respect to the displacement components, $z_i = \partial g/\partial u_i$. It is easy to check that we obtain

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \boldsymbol{\lambda}^T(\mu\frac{d\mathbf{p}}{dx} - \frac{\partial\mathbf{f}}{\partial x})\,. \tag{7.2.25}$$

7.2.5 *Sensitivity of Limit Loads*

At a critical point with the load value denoted as μ^*, the tangential stiffness matrix $\mathbf{J}$ becomes singular, and we can have either a bifurcation point or a limit load. We can distinguish between the two by differentiating Eq. (7.2.20) with respect to a loading

parameter that increases monotonically throughout the loading history. The load parameter μ is not a good choice, because at a limit point it reaches a maximum and is not monotonic. Instead we often use a displacement component, known to increase monotonically, or the arc length in the $(\mathbf{u}, \mu)$ space. We denote such a monotonic load parameter by α, and denote a derivative with respect to α by a prime. Differentiating Eq. (7.2.20) with respect to α we get

$$\mathbf{J}\mathbf{u}' = \mu'\mathbf{p}\,. \tag{7.2.26}$$

At a critical point, $\mathbf{J}$ is singular, and we denote the left eigenvector associated with the zero eigenvalue of $\mathbf{J}$ by $\mathbf{v}$, that is

$$\mathbf{v}^T\mathbf{J}^* = 0\,, \tag{7.2.27}$$

where the asterisk denotes quantities evaluated at the critical point. Premultiplying Eq. (7.2.26) by $\mathbf{v}^T$, we get

$$\mu'\mathbf{v}^T\mathbf{p} = 0\,. \tag{7.2.28}$$

At a limit point this equation is satisfied because the load reaches a maximum, and then $\mu' = 0$. In that case, Eq. (7.2.26) indicates that the buckling mode, which is the right eigenvector of the tangential stiffness matrix $\mathbf{J}$, is equal to the derivative of $\mathbf{u}$ with respect to the loading parameter. At a bifurcation point $\mu' \neq 0$, and instead

$$\mathbf{v}^T\mathbf{p} = 0\,. \tag{7.2.29}$$

For a symmetric tangential stiffness matrix $\mathbf{v}$ is also the buckling mode, and Eq. (7.2.29) indicates that the buckling mode is orthogonal to the load vector.

To calculate sensitivity of limit loads we need to consider a more general response path parameter ν which can be a load parameter, a design variable, or a combination of both—a parameter that controls both structural design and loading simultaneously. We denote differentiation with respect to ν by a dot and differentiate Eq. (7.2.20) with respect to ν to get

$$\mathbf{J}\dot{\mathbf{u}} + \frac{\partial \mathbf{f}}{\partial x}\dot{x} = \dot{\mu}\mathbf{p} + \mu\frac{d\mathbf{p}}{dx}\dot{x}\,. \tag{7.2.30}$$

We now want a parameter ν that controls the design variable x and the load parameter μ so that we remain at a limit load, $\mu = \mu^*$. We select $\nu = x$, and then Eq. (7.2.30) becomes

$$\mathbf{J}^*\dot{\mathbf{u}} + (\frac{\partial \mathbf{f}}{\partial x})^* = \frac{d\mu^*}{dx}\mathbf{p} + \mu^*\frac{d\mathbf{p}}{dx}\,, \tag{7.2.31}$$

where we used the fact that for our choice of parameter $\dot{x} = 1$. Premultiplying Eq. (7.2.31) by the left eigenvector, $\mathbf{v}^T$, and rearranging we get

$$\frac{d\mu^*}{dx} = \frac{\mathbf{v}^T\left[(\frac{\partial \mathbf{f}}{\partial x})^* - \mu^*\frac{d\mathbf{p}}{dx}\right]}{\mathbf{v}^T\mathbf{p}}\,. \tag{7.2.32}$$

The quantity in brackets in the numerator of Eq. (7.2.32) is the derivative of the residual of the equations of equilibrium at the limit point. Thus we can use the semi-analytical method to evaluate the limit load sensitivity as follows: We perturb the design variable, calculate the change in the residual (for fixed displacements) and take the dot product with the buckling mode to get the numerator. The denominator is the dot product of the buckling mode with the load vector.

7.3 Sensitivity Calculations for Eigenvalue Problems

Eigenvalue problems are commonly encountered in structural stability and vibration analysis. When forces are conservative, and no damping is considered, these problems lead to real eigenvalues which represent buckling loads or vibration frequencies. In the more general case the eigenvalues are complex. Our discussion starts with the simpler case of real eigenvalues.

7.3.1 *Sensitivity Derivatives of Vibration and Buckling Constraints*

Undamped vibration and linear buckling analysis lead to eigenvalue problems of the type

$$\mathbf{K}\mathbf{u} - \mu\mathbf{M}\mathbf{u} = 0\,, \tag{7.3.1}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{M}$ is the mass matrix (vibration) or the geometric stiffness matrix (buckling) and $\mathbf{u}$ is the mode shape. For vibration problems μ is the square of the frequency of free vibration, and for buckling problems it is the buckling load factor. Both $\mathbf{K}$ and $\mathbf{M}$ are symmetric, and $\mathbf{K}$ is positive semidefinite. The mode shape is often normalized with a symmetric positive definite matrix $\mathbf{W}$ such that

$$\mathbf{u}^T\mathbf{W}\mathbf{u} = 1\,, \tag{7.3.2}$$

where, for vibration problems, $\mathbf{W}$ is usually the mass matrix $\mathbf{M}$. Equations (7.3.1) and (7.3.2) hold for all eigenpairs $(\mu_k, \mathbf{u}^k)$. Differentiating these equations with respect to a design variable x we obtain

$$(\mathbf{K} - \mu\mathbf{M})\frac{d\mathbf{u}}{dx} - \frac{d\mu}{dx}\mathbf{M}\mathbf{u} = -(\frac{d\mathbf{K}}{dx} - \mu\frac{d\mathbf{M}}{dx})\mathbf{u}\,, \tag{7.3.3}$$

and

$$\mathbf{u}^T\mathbf{W}\frac{d\mathbf{u}}{dx} = -\frac{1}{2}\mathbf{u}^T\frac{d\mathbf{W}}{dx}\mathbf{u}\,, \tag{7.3.4}$$

where we have used of the symmetry of $\mathbf{W}$. Equations (7.3.3) and (7.3.4) are valid only for the case of distinct eigenvalues (repeated eigenvalues are, in general, not differentiable, and only directional derivatives may be obtained, see Haug et al. [8]). In most applications we are interested only in the derivatives of the eigenvalues. These derivatives may be obtained by premultiplying Eq. (7.3.3) by $\mathbf{u}^T$ to obtain

$$\frac{d\mu}{dx} = \frac{\mathbf{u}^T(\frac{d\mathbf{K}}{dx} - \mu\frac{d\mathbf{M}}{dx})\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}}\,. \tag{7.3.5}$$

In some applications the derivatives of the eigenvectors are also required. For example, in automobile design we often require that critical vibration modes have low amplitudes at the front seats. For this design problem we need derivatives of the

mode shape. To obtain eigenvector derivatives we can use the direct approach and combine Eqs. (7.3.3) and (7.3.4) as

$$\begin{bmatrix} \mathbf{K}-\mu\mathbf{M} & -\mathbf{M}\mathbf{u} \\ -\mathbf{u}^T\mathbf{W} & 0 \end{bmatrix} \begin{Bmatrix} \frac{d\mathbf{u}}{dx} \\ \frac{d\mu}{dx} \end{Bmatrix} = \begin{Bmatrix} -(\frac{d\mathbf{K}}{dx}-\mu\frac{d\mathbf{M}}{dx})\mathbf{u} \\ \frac{1}{2}\mathbf{u}^T\frac{d\mathbf{W}}{dx}\mathbf{u} \end{Bmatrix} . \tag{7.3.6}$$

The system (7.3.6) may be solved for the derivatives of the eigenvalue and the eigenvector. However, care must be taken in the solution process because the principal minor $\mathbf{K}-\mu\mathbf{M}$ is singular. Cardani and Mantegazza [9] and Murthy and Haftka [10] discuss several solution strategies which address this problem.

One of the more popular solution techniques is due to Nelson[11]. Nelson's method temporarily replaces the normalization condition, Eq. (7.3.2), by the requirement that the largest component of the eigenvector be equal to one. Denoting this re-normalized vector $\bar{\mathbf{u}}$, and assuming that its largest component is the mth one, we replace Eq. (7.3.2) by

$$\bar{u}_m = 1 , \tag{7.3.7}$$

and Eq.(7.3.4) by

$$\frac{d\bar{u}_m}{dx} = 0 . \tag{7.3.8}$$

Equation (7.3.3) is valid with $\mathbf{u}$ replaced by $\bar{\mathbf{u}}$, but Eq. (7.3.8) is used to reduce its order by deleting the mth row and the mth column. When the eigenvalue μ is distinct, the reduced system is not singular, and may be solved by standard techniques.

To retrieve the derivative of the eigenvector with the original normalization of Eq. (7.3.2) we note that $\mathbf{u} = u_m\bar{\mathbf{u}}$, so that

$$\frac{d\mathbf{u}}{dx} = \frac{du_m}{dx}\bar{\mathbf{u}} + u_m\frac{d\bar{\mathbf{u}}}{dx} , \tag{7.3.9}$$

and du_m/dx may be obtained by substituting Eq. (7.3.9) into Eq. (7.3.4) to obtain

$$\frac{du_m}{dx} = -u_m^2\mathbf{u}^T\mathbf{W}\frac{d\bar{\mathbf{u}}}{dx} - \frac{u_m}{2}\mathbf{u}^T\frac{d\mathbf{W}}{dx}\mathbf{u} . \tag{7.3.10}$$

We can also use an adjoint or modal technique for calculating the derivatives of the eigenvector by expanding that derivative as a linear combination of eigenvectors. That is, denoting the ith eigenpair of Eq. (7.3.1) by $(\mu_i, \mathbf{u}^i)$ we assume

$$\frac{d\mathbf{u}^k}{dx} = \sum_{j=1}^{l} c_{kj}\mathbf{u}^j , \tag{7.3.11}$$

and the coefficients c_{kj} can be shown to be (see, for example, Rogers [12])

$$c_{kj} = \frac{\mathbf{u}^{jT}(\frac{d\mathbf{K}}{dx}-\mu_k\frac{d\mathbf{M}}{dx})\mathbf{u}^k}{(\mu_k-\mu_j)\mathbf{u}^{jT}\mathbf{M}\mathbf{u}^j} , \qquad k \neq j . \tag{7.3.12}$$

Using the normalization condition of Eq. (7.3.7) we find

$$c_{kk} = -\sum_{j \neq k} c_{kj} u_m^j \,. \tag{7.3.13}$$

On the other hand, if we use the normalization condition of Eq. (7.3.2) with $\mathbf{W} = \mathbf{M}$, we get

$$c_{kk} = -\frac{1}{2}(\mathbf{u}^k)^T \frac{d\mathbf{M}}{dx}\mathbf{u}^k \,. \tag{7.3.14}$$

If all the eigenvectors are included in the sum, Eq. (7.3.11) is exact. For most problems it is not practical to calculate all the eigenvectors, so that only a few of the eigenvectors associated with the lowest eigenvalues are included. Wang [13] developed a modified modal method that accelerates the convergence. Instead of Eq. (7.3.11) we use

$$\frac{d\mathbf{u}^k}{dx} = \mathbf{u}_s^k + \sum_{j=1}^{l} d_{kj}\mathbf{u}^j \,, \tag{7.3.15}$$

where

$$\mathbf{u}_s^k = \mathbf{K}^{-1}\left[\frac{d\mu}{dx}\mathbf{M} - \frac{d\mathbf{K}}{dx} + \mu\frac{d\mathbf{M}}{dx}\right]\mathbf{u}^k \tag{7.3.16}$$

is a static correction term, and

$$d_{kj} = \mu_k \frac{\mathbf{u}^{jT}(\dfrac{d\mathbf{K}}{dx} - \mu_k\dfrac{d\mathbf{M}}{dx})\mathbf{u}^k}{\mu_j(\mu_k - \mu_j)\mathbf{u}^{jT}\mathbf{M}\mathbf{u}^j} \,, \qquad k \neq j \,. \tag{7.3.17}$$

The coefficient d_{kk} is still given by Eq. (7.3.14) for the normalization condition of $\mathbf{u}^T\mathbf{M}\mathbf{u} = 1$. For the normalization condition of (7.3.7)

$$d_{kk} = -u_{sm}^k - \sum_{j \neq k} d_{kj} u_m^j \,. \tag{7.3.18}$$

Sutter et al. [14] present a study of the convergence of the derivative with increasing number of modes using both the modal method and the modified modal method and demonstrate the improved convergence of the modified modal method.

Example 7.3.1

The spring-mass-dashpot system shown in Fig. (7.3.1) is analysed here for the case that the dashpot is inactivated, that is $c = 0$. Initially the two masses and the three springs have values of 1, and we want to calculate the derivatives of the lowest vibration frequency and the lowest vibration mode with respect to k for two possible normalization conditions: one of the form Eq. (7.3.2) with $\mathbf{W} = \mathbf{M}$, and one of the form Eq. (7.3.7) with the second component of the mode set to 1.

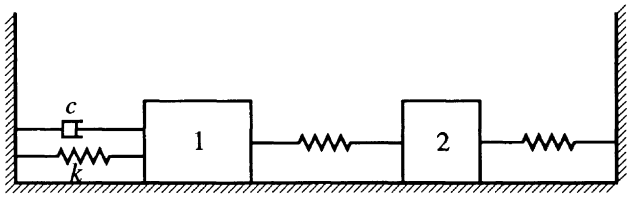


Figure 7.3.1 Spring-mass-dashpot example for eigenvalue derivatives.

Denoting the motions of the two masses as u_1 and u_2, we find the elastic energy, E, and the kinetic energy, T, to be

$$E = 0.5\left[ku_1^2 + (u_2 - u_1)^2 + u_2^2\right], \qquad T = 0.5(\dot{u}_1^2 + \dot{u}_2^2)\,.$$

This gives us the stiffness and mass matrices as

$$\mathbf{K} = \begin{bmatrix} 1+k & -1 \\ -1 & 2 \end{bmatrix}, \qquad \mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

For $k = 1$, the eigenvalue problem, Eq. (7.3.1) becomes

$$\begin{bmatrix} 2-\omega^2 & -1 \\ -1 & 2-\omega^2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = 0\,. \tag{a}$$

Setting the determinant of the system to zero we get the two frequencies, $\omega_1 = 1$, and $\omega_2 = \sqrt{3}$. Substituting back the lowest frequency into Eq. (a) we get for the first vibration mode

$$u_1 - u_2 = 0\,,$$
$$-u_1 + u_2 = 0\,.$$

As expected, the system is singular at a natural frequency, so that we need the normalization condition to determine the eigenvector. For the normalization condition (7.3.2) the additional equation is

$$\mathbf{u}^T\mathbf{M}\mathbf{u} = u_1^2 + u_2^2 = 1\,.$$

For the normalization condition Eq. (7.3.7), the condition is

$$\bar{u}_2 = 1\,,$$

where we use the bar to denote the vibration mode with the second normalization condition. The solutions with the normalization conditions are

$$\mathbf{u} = \frac{\sqrt{2}}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix}, \qquad \bar{\mathbf{u}} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}.$$

Next we calculate the derivative of the lowest frequency from Eq. (7.3.5) using primes to denote derivatives with respect to k. For our example

$$\mathbf{K}' = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{M}' = \mathbf{0}.$$

We use the mode normalized by the mass matrix in Eq. (7.3.5), so that the denominator is equal to 1, and then

$$\mu' = (\omega^2)' = \mathbf{u}^T\mathbf{K}'\mathbf{u} = 0.5.$$

We can also get the derivative of the frequency and the mode together by using Eq. (7.3.6). We note that

$$\mathbf{K} - \mu\mathbf{M} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{M}\mathbf{u} = \mathbf{W}\mathbf{u} = \mathbf{u} = \frac{\sqrt{2}}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix},$$

$$-(\mathbf{K}' - \mu\mathbf{M}')\mathbf{u} = -\mathbf{K}'\mathbf{u} = \frac{\sqrt{2}}{2}\begin{Bmatrix} -1 \\ 0 \end{Bmatrix}, \quad \frac{1}{2}\mathbf{u}^T\mathbf{W}'\mathbf{u} = 0.$$

Equation (7.3.6) is then

$$\begin{bmatrix} 1 & -1 & -\sqrt{2}/2 \\ -1 & 1 & -\sqrt{2}/2 \\ -\sqrt{2}/2 & -\sqrt{2}/2 & 0 \end{bmatrix} \begin{Bmatrix} u_1' \\ u_2' \\ \mu' \end{Bmatrix} = \begin{Bmatrix} -\sqrt{2}/2 \\ 0 \\ 0 \end{Bmatrix}.$$

We solve this equation to get

$$u_1' = -\sqrt{2}/8, \quad u_2' = \sqrt{2}/8, \quad \mu' = 1/2.$$

In order to solve for $\bar{u}'$ from Eq. (7.3.3), with the additional condition $\bar{u}_2' = 0$, we need to evaluate the expressions:

$$\mu'\mathbf{M}\bar{\mathbf{u}} = 0.5\bar{\mathbf{u}} = \begin{Bmatrix} 0.5 \\ 0.5 \end{Bmatrix}, \quad -(\mathbf{K}' - \mu\mathbf{M}')\bar{\mathbf{u}} = -\mathbf{K}'\bar{\mathbf{u}} = \begin{Bmatrix} -1 \\ 0 \end{Bmatrix}.$$

Then Eq. (7.3.3), with $\bar{\mathbf{u}}$ replacing $\mathbf{u}$, and the additional condition yield

$$\begin{aligned} \bar{u}_1' - \bar{u}_2' &= -0.5, \\ -\bar{u}_1' + \bar{u}_2' &= 0.5, \\ \bar{u}_2' &= 0. \end{aligned}$$

The solution is

$$\bar{u}_1' = -0.5, \qquad \bar{u}_2' = 0.$$

We can show that $\mathbf{u}$ can indeed be retrieved from $\bar{\mathbf{u}}'$ by using Eqs. (7.3.9) and (7.3.10). Equation (7.3.10) becomes

$$u_2' = -u_2^2\mathbf{u}^T\bar{\mathbf{u}}' = -0.5(\sqrt{2}/2)[\,1 \quad 1\,]\begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix} = \sqrt{2}/8,$$

which agrees with our previous result. Equation (7.3.9) becomes

$$\mathbf{u}' = (\sqrt{2}/8)\bar{\mathbf{u}} + (\sqrt{2}/2)\bar{\mathbf{u}}' = \frac{\sqrt{2}}{8}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \frac{\sqrt{2}}{2}\begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix} = \frac{\sqrt{2}}{8}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix},$$

which also agrees with our previous result.• • •

When the eigenvalue μ is repeated with a multiplicity of m, there are m linearly independent eigenvectors associated with it. Furthermore, any linear combination of these eigenvectors is also an eigenvector, so that the choice of eigenvectors is not unique. In this case the eigenvectors that are obtained from a structural analysis program will be determined by the idiosyncrasies of the computational procedure used for the solution of the eigenproblem. Assuming that $\mathbf{u}^1, \ldots, \mathbf{u}^m$ is a set of linearly independent eigenvectors associated with μ, we may write any eigenvector associated with μ as

$$\mathbf{u} = \sum_{i=1}^{m} q_i \mathbf{u}^i = \mathbf{U}\mathbf{q}, \tag{7.3.19}$$

where $\mathbf{q}$ is a vector of coefficients and $\mathbf{U}$ a matrix with columns equal to $\mathbf{u}^i$, $i = 1, \ldots, m$. As the design variable x is changed, the eigenvalues usually separate, and the eigenvectors become unique again. We obtain these eigenvectors by substituting Eq. (7.3.19) into Eq. (7.3.3) and premultiplying by $\mathbf{U}^T$ to obtain

$$(\mathbf{A} - \frac{d\mu}{dx}\mathbf{B})\mathbf{q} = 0, \tag{7.3.20}$$

where

$$\mathbf{A} = \mathbf{U}^T(\frac{d\mathbf{K}}{dx} - \mu\frac{d\mathbf{M}}{dx})\mathbf{U}, \tag{7.3.21}$$

and

$$\mathbf{B} = \mathbf{U}^T\mathbf{M}\mathbf{U}. \tag{7.3.22}$$

Equation (7.3.20) is an $m \times m$ eigenvalue problem for $d\mu/dx$. The m solutions correspond to the derivatives of the m eigenvalues derived from μ as x is changed, and the eigenvectors $\mathbf{q}$ give us, through Eq. (7.3.19), the eigenvectors associated with the perturbed eigenvalues. A generalization of Nelson's method to obtain derivatives of the eigenvectors was suggested by Ojalvo [15] and amended by Mills-Curran [16] and Dailey [17]. Their procedure seems to contradict the earlier assertion that repeated eigenvalues are not differentiable. However, while we can find derivatives with respect to any individual variable, these are only good as directional derivatives, in that derivatives with respect to x and y cannot be combined in a linear fashion. That is

$$d\mu = \frac{\partial \mu}{\partial x}dx + \frac{\partial \mu}{\partial y}dy \tag{7.3.23}$$

will not hold in general. This is demonstrated in the following example.

Example 7.3.2

Let us consider a simple, two variable system

$$\mathbf{K} = \begin{bmatrix} 2+y & x \\ x & 2 \end{bmatrix}, \qquad \mathbf{W} = \mathbf{M} = \mathbf{I}.$$

The two eigenvalues are

$$\mu_{1,2} = 2 + y/2 \pm \sqrt{x^2 + y^2/4}. \tag{a}$$

The two eigenvalues are identical for $x = y = 0$, and we will first demonstrate that the eigenvectors are discontinuous at the origin. In fact for $x = 0$ the two eigenvectors are

$$\mathbf{u}^1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}, \quad \mathbf{u}^2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix},$$

and for $y = 0$

$$\mathbf{u}^1 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}, \quad \mathbf{u}^2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}.$$

Obviously, we can get either set of eigenvectors as close to the origin as we wish by approaching it either along the x axis or along the y axis.

Next we calculate the derivatives of the two eigenvalues with respect to x and y at the origin. At (0,0) any vector is an eigenvector, and we select the two coordinate unit vectors as a basis, that is

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We first calculate derivatives with respect to x, and using Eqs. (7.3.21) and (7.3.22) we get

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The solution of the eigenvalue problem, Eq. (7.3.20) is

$$\left(\frac{\partial \mu}{\partial x}\right)_1 = 1, \quad \left(\frac{\partial \mu}{\partial x}\right)_2 = -1,$$

and the corresponding eigenvectors are

$$\mathbf{q}^1 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix},$$

$$\mathbf{q}^2 = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix},$$

and because $\mathbf{U}$ is the unit matrix, from Eq. (7.3.19) $\mathbf{u}^i = \mathbf{q}^i$. It is easy to check that these are indeed the eigenvectors along the y axis $(x, 0)$. Similarly, for derivatives with respect to y we have

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and the two eigenvalues of Eq. (7.3.20) are

$$(\frac{\partial \mu}{\partial y})_1 = 1, \quad (\frac{\partial \mu}{\partial y})_2 = 0.$$

The corresponding eigenvectors are

$$\mathbf{q}^1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix},$$

$$\mathbf{q}^2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}.$$

To see that the above derivatives cannot be used to calculate the change in μ due to a simultaneous change in x and y, consider an infinitesimal change $dy = 2dx = 2dt$. From the solution for the two eigenvalues, Eq. (a), we have

$$d\mu = dt \pm \sqrt{2}dt.$$

On the other hand, Eq. (7.3.23) yields four values depending on which of two values we use for the x and y derivatives. These are $3dt$, dt, dt, and $-dt$.• • •

The implications of the failure of calculating a derivative in an arbitrary direction from derivatives in the coordinate directions are quite serious. Most optimization algorithms rely on these calculations to choose move directions or to estimate objective function and constraints. Therefore, these algorithms could experience serious difficulties for problems with repeated eigenvalues. On the bright side, computational experience shows that even minute differences between eigenvalues are often sufficient to prevent such difficulties. Furthermore, the coalescence of eigenvalues often has an adverse effect on structural performance. In buckling problems it is associated with imperfection sensitivity, and for structural control problems coalescence of vibration frequencies can lead to control difficulties. Therefore, constraints are often used to separate the eigenvalues in design problems.

7.3.2 Sensitivity Derivatives for Non-Hermitian Eigenvalue Problems

When structural damping is important or when damping is supplied by aerodynamic forces or active control systems, the damped motion $\bar{\mathbf{u}}$ is governed by

$$\mathbf{M}\ddot{\bar{\mathbf{u}}} + \mathbf{C}\dot{\bar{\mathbf{u}}} + \mathbf{K}\bar{\mathbf{u}} = 0, \tag{7.3.24}$$

where $\mathbf{C}$ is the damping matrix, assumed to be symmetric, and a dot denotes differentiation with respect to time. Setting

$$\tilde{\mathbf{u}} = \mathbf{u}e^{\mu t}\,, \tag{7.3.25}$$

we get

$$[\mu^2\mathbf{M} + \mu\mathbf{C} + \mathbf{K}]\mathbf{u} = 0\,. \tag{7.3.26}$$

Note that we have not defined the eigenvalue μ in the way we did for the undamped vibration problem. There μ was the square of the frequency, while here, when $\mathbf{C} = 0$, we get $\mu = i\omega$ where ω is the vibration frequency. The derivative of the eigenvalue μ with respect to a design variable x is obtained by differentiating Eq. (7.3.26) with respect to x and premultiplying by $\mathbf{u}^T$

$$\frac{d\mu}{dx} = -\frac{\mu^2\mathbf{u}^T\dfrac{d\mathbf{M}}{dx}\mathbf{u} + \mu\mathbf{u}^T\dfrac{d\mathbf{C}}{dx}\mathbf{u} + \mathbf{u}^T\dfrac{d\mathbf{K}}{dx}\mathbf{u}}{2\mu\mathbf{u}^T\mathbf{M}\mathbf{u} + \mathbf{u}^T\mathbf{C}\mathbf{u}}\,. \tag{7.3.27}$$

This equation can be used for estimating the effect of adding a small amount of damping to an undamped system. For the undamped system $\mathbf{C} = 0$, the eigenvalue is $\mu = i\omega$, and the eigenvector is the vibration mode that we will denote here as $\boldsymbol{\phi}$ to distinguish it from the damped mode $\mathbf{u}$. Then Eq. (7.3.27) becomes

$$\frac{d\mu}{dx} = -\frac{\boldsymbol{\phi}^T\dfrac{d\mathbf{C}}{dx}\boldsymbol{\phi}}{2\boldsymbol{\phi}^T\mathbf{M}\boldsymbol{\phi}}\,. \tag{7.3.28}$$

Example 7.3.3

Use linear extrapolation to estimate the effect of the dashpot in Figure (7.3.1) on the first vibration mode, and then compare with the exact effect for $c = 0.2$, and $c = 1.0$.

For this example we take $x = c$ and then (using $\mathbf{K}$, and $\mathbf{M}$ from Example 7.3.1)

$$\mathbf{C} = \begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix}\,, \qquad \frac{d\mathbf{M}}{dx} = \frac{d\mathbf{K}}{dx} = \mathbf{0}, \qquad \frac{d\mathbf{C}}{dx} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\,.$$

Using the first vibration mode from Example (7.3.1) which is normalized so that the denominator of Eq. (7.3.28) is 1, $(\boldsymbol{\phi}^1)^T = (\sqrt{2}/2)[1\,,1]$, we get

$$\frac{d\mu}{dc} \equiv \frac{d\mu}{dx} = -0.5\boldsymbol{\phi}^T\frac{d\mathbf{C}}{dx}\boldsymbol{\phi} = -0.25\,.$$

From Example (7.3.1), the frequency of the first natural mode is $\omega_1 = 1$ (which corresponds to $\mu = i$ in the notation of this section). Then using linear extrapolation to calculate an approximate eigenvalue μ_a we get

$$\mu_a = \mu\Big|_{c=0} + \frac{d\mu}{dc}c = -0.25c + i\,.$$

For the two given values of $c = 0.2$, and $c = 1.0$, the approximate eigenvalues are $-0.05 + i$, and $-0.25 + i$, respectively. We compare this approximation to the exact result obtained by solving Eq. (7.3.26); this yields

$$\begin{bmatrix} \mu^2 + c\mu + 2 & -1 \\ -1 & \mu^2 + 2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = 0 . \qquad (a)$$

The eigenvalue μ is obtained by setting the determinant of this equation to zero. For the two values of c we get

$$c = 0.2 : \quad \mu = -0.05025 + 1.0013i .$$

$$c = 1.0 : \quad \mu = -0.29178 + 1.0326i .$$

We see that the prediction that c changes only the damping and not the frequency is quite good, and that linear extrapolation worked quite well for predicting the damping. • • •

The order of the damped eigenproblem is commonly reduced by approximating the damped mode as a linear combination of a small number of natural vibration modes $\mathbf{u}^i$, $i = 1 \ldots, m$. This may be written as

$$\mathbf{u} = \mathbf{U}\mathbf{q} , \qquad (7.3.29)$$

where $\mathbf{U}$ is a matrix with $\mathbf{u}^i$ as columns, and $\mathbf{q}$ is a vector of modal amplitudes. Substituting Eq. (7.3.29) into Eq. (7.3.26) and premultiplying by $\mathbf{U}^T$ we get

$$[\mu^2 \mathbf{M}_R + \mu \mathbf{C}_R + \mathbf{K}_R]\mathbf{q} = 0 , \qquad (7.3.30)$$

where

$$\mathbf{M}_R = \mathbf{U}^T \mathbf{M} \mathbf{U}, \quad \mathbf{C}_R = \mathbf{U}^T \mathbf{C} \mathbf{U}, \quad \mathbf{K}_R = \mathbf{U}^T \mathbf{K} \mathbf{U} . \qquad (7.3.31)$$

After we solve for the reduced eigenvector $\mathbf{q}$ from Eq. (7.3.30), we can calculate the derivative of the eigenvalue using two approaches. The first approach, called the fixed-mode approach, employs Eq. (7.3.27) with μ calculated from Eq. (7.3.30) and $\mathbf{u}$ given by Eq. (7.3.29). The second approach, called the updated-mode approach, uses Eq. (7.3.27) for the reduced problem, that is

$$\frac{d\mu}{dx} = -\frac{\mu^2 \mathbf{q}^T \dfrac{d\mathbf{M}_R}{dx}\mathbf{q} + \mu \mathbf{q}^T \dfrac{d\mathbf{C}_R}{dx}\mathbf{q} + \mathbf{q}^T \dfrac{d\mathbf{K}_R}{dx}\mathbf{q}}{2\mu \mathbf{q}^T \mathbf{M}_R \mathbf{q} + \mathbf{q}^T \mathbf{C}_R \mathbf{q}} . \qquad (7.3.32)$$

The derivative of K_R is given as

$$\frac{d\mathbf{K}_R}{dx} = \mathbf{U}^T \frac{d\mathbf{K}}{dx}\mathbf{U} + \frac{d\mathbf{U}^T}{dx}\mathbf{K}\mathbf{U} + \mathbf{U}^T \mathbf{K} \frac{d\mathbf{U}}{dx} \qquad (7.3.33)$$

with similar expressions for the derivatives of $\mathbf{M}_R$ and $\mathbf{C}_R$. The names of the two approaches are associated with the fact that the corresponding derivatives will agree with a finite-difference derivative calculations with the modes being fixed or updated, respectively. Also, it can be shown that if we omit the terms with $d\mathbf{U}/dx$ from the updated-mode expression we will recover the fixed-mode result. The calculation of derivatives of vibration modes is expensive, and for this reason the fixed-mode approach is more appealing. However, as the following example demonstrates, the updated-mode approach can, occasionally, be substantially more accurate.

Example 7.3.4

For the spring-mass-dashpot example shown in Fig. (7.3.1) construct a reduced model based only on the first vibration mode. Calculate the fixed-mode and updated-mode derivatives of the eigenvalue associated with the lowest frequency with respect to the constant k of the leftmost spring. Compare with the exact derivatives for $c = 0.2$ and $c = 1.0$.

Full-model analysis:

The eigenvalue problem for this example is given by Eq. (a) of Example (7.3.3), and the exact eigenvalue is solved in that example for the two required values of c. For the eigenvector we use a normalization condition that the second component, u_2, is equal to 1, and employ the second equation of the eigenproblem to obtain

$$\mathbf{u} = \begin{Bmatrix} \mu^2 + 1 \\ 1 \end{Bmatrix}.$$

To calculate the derivative of μ with respect to the stiffness k of the leftmost spring we use Eq. (7.3.27) with matrices calculated in Examples 7.3.1 and 7.3.3

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} c & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{K} = \begin{bmatrix} k+1 & -1 \\ -1 & 2 \end{bmatrix},$$
$$\mathbf{M}' = 0, \qquad \mathbf{C}' = 0, \qquad \mathbf{K}' = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

where a prime is used to denote a derivative with respect to k. Then from Eq. (7.3.27) we get

$$\mu' = -\frac{\mathbf{u}^T \mathbf{K}' \mathbf{u}}{\mathbf{u}^T (\mathbf{C} + 2\mu \mathbf{M}) \mathbf{u}} = \frac{-(\mu^2 + 2)^2}{c(\mu^2 + 2)^2 + 2\mu\left[(\mu^2 + 2)^2 + 1\right]}.$$

For the two values of c we get (see Example 7.3.3 for values of μ)

$$\text{For } c = 0.2: \quad \mu = -0.05025 + 1.0013i, \quad \mu' = 0.02525 + 0.2522i$$

$$\text{For } c = 1.0: \quad \mu = -0.29178 + 1.0326i, \quad \mu' = 0.1544 + 0.3460i$$

Reduced-basis analysis:

The vibration frequencies and first vibration mode were calculated in Example (7.3.1). Since the normalization condition for the full-model eigenvector was that the second component be equal to 1, we take the vibration mode with the same normalization. This mode was denoted with an overbar in Example (7.3.1), but we drop this overbar since it is the only mode used here

$$\mathbf{u} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}.$$

Since we use only one mode for the reduced basis, $\mathbf{U} = \mathbf{u}$, and using Eq. (7.3.31) with $k = 1$ we get

$$\mathbf{M}_R = 2, \quad \mathbf{C}_R = c, \quad \mathbf{K}_R = 2\,.$$

Equation (7.3.30) for the reduced system becomes

$$(2\mu^2 + c\mu + 2)q = 0\,,$$

so that

$$\mu_R = -0.25c + i\sqrt{1 - 0.0625c^2}\,,$$

where the subscript R is used to denote the fact that this is the eigenvalue obtained from the reduced system. The eigenvector, which has only one component, we select as $q = 1$. For the two values of c we get

$$\text{c=0.2:} \quad \mu_R = -0.05 + 0.9987i\,,$$

$$\text{c=1.0:} \quad \mu_R = -0.25 + 0.9682i\,.$$

It appears that the reduced model gives excellent results for the low-damping case, and moderate errors for the high damping case.

Fixed mode derivative:

For the fixed-mode derivative we still use Eq. (7.3.27), but with μ replaced by μ_R and $\mathbf{u}$ replaced by its approximation in term of the vibration modes. Since the eigenvector $q = 1$, this approximation is equal to the first vibration mode, so

$$\mu' = -\frac{\mathbf{u}^T\mathbf{K}'\mathbf{u}}{\mathbf{u}^T(\mathbf{C} + 2\mu_R\mathbf{M})\mathbf{u}} = \frac{-1}{c + 4\mu_R}\,,$$

For the two values of c we get

$$\text{c=0.2:} \quad \mu'_{Rf} = 0.2503i\,,$$

$$\text{c=1.0:} \quad \mu'_{Rf} = 0.2582i\,,$$

where the subscript f was used to denote derivatives calculated with the fixed-mode approach. We note that the derivative of the imaginary part (frequency) is good only in the low-damping case, and that the fixed-mode derivative misses out altogether the effect on the real part (damping). Large errors of this type can happen when the derivative is small. Recall that the size of a derivative is best estimated by the logarithmic derivative. However, here the logarithmic derivative of the real part, say for the low damping case is

$$\frac{d\mu^r/\mu^r}{dk/k} = 0.02525/(-0.05025) = -0.5025\,,$$

so that it is quite substantial.

Updated-mode derivative:

In this case we need the derivative of the vibration mode with respect to k. This was calculated in Example (7.3.1) as (remember that we use $\bar{\mathbf{u}}$ from that example)

$$\mathbf{u}' = \begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix}.$$

Then from Eq. (7.3.33)

$$\mathbf{K}'_R = \mathbf{u}^T[\mathbf{K}'\mathbf{u} + 2\mathbf{K}\mathbf{u}'] = [1 \quad 1]\left[\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + 2\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix}\right] = 0.$$

Similarly

$$\mathbf{M}'_R = 2\mathbf{u}^T\mathbf{M}\mathbf{u}' = 2[1 \quad 1]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix} = -1,$$

$$\mathbf{C}'_R = 2\mathbf{u}^T\mathbf{C}\mathbf{u}' = 2[1 \quad 1]\begin{bmatrix} c & 0 \\ 0 & 0 \end{bmatrix}\begin{Bmatrix} -0.5 \\ 0 \end{Bmatrix} = -c.$$

Finally, from Eq. (7.3.32)

$$\mu'_{Ru} = -\frac{-c\mu_R - \mu_R^2}{4\mu_R + c}.$$

For the two values of c we get

$$\text{c=0.2:} \quad \mu'_{Ru} = 0.025 + 0.2513i,$$

$$\text{c=1.0:} \quad \mu'_{Ru} = 0.125 + 0.2843i,$$

which is a much better approximation to the exact derivative than μ'_{Rf}.• • •

In many applications the damping matrix is not symmetric, and then it is convenient to transform the equations of motion Eq. (7.3.24) to a first order system

$$\mathbf{B}\dot{\bar{\mathbf{w}}} + \mathbf{A}\bar{\mathbf{w}} = 0, \tag{7.3.34}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{C} & \mathbf{K} \\ -\mathbf{I} & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \quad \bar{\mathbf{w}} = \begin{Bmatrix} \dot{\bar{\mathbf{u}}} \\ \bar{\mathbf{u}} \end{Bmatrix}. \tag{7.3.35}$$

Setting

$$\bar{\mathbf{w}} = \mathbf{w}e^{\mu t}, \tag{7.3.36}$$

we get a first-order eigenvalue problem

$$\mathbf{A}\mathbf{w} + \mu\mathbf{B}\mathbf{w} = 0. \tag{7.3.37}$$

For calculating the derivatives of the eigenvalues it is convenient to use the left eigenvector $\mathbf{v}$ which is the solution of the associated eigenproblem

$$\mathbf{v}^T\mathbf{A} + \mu\mathbf{v}^T\mathbf{B} = 0. \tag{7.3.38}$$

The two eigenproblems defined in Eqs. (7.3.38) and (7.3.37) are easily shown to have the same eigenvalues (e.g., [18]). Differentiating (7.3.37) with respect to a design variable x

$$(\mathbf{A}+\mu\mathbf{B})\frac{d\mathbf{w}}{dx}+(\frac{d\mathbf{A}}{dx}+\mu\frac{d\mathbf{B}}{dx})\mathbf{w}+\frac{d\mu}{dx}\mathbf{B}\mathbf{w}=0\,, \tag{7.3.39}$$

and premultiplying by $\mathbf{v}^T$ we get

$$\frac{d\mu}{dx}=-\frac{\mathbf{v}^T(\frac{d\mathbf{A}}{dx}+\mu\frac{d\mathbf{B}}{dx})\mathbf{w}}{\mathbf{v}^T\mathbf{B}\mathbf{w}}\,. \tag{7.3.40}$$

To obtain derivatives of the eigenvector we need a normalization condition. A quadratic condition such as Eq. (7.3.2) is inappropriate because the eigenvector is complex and $\mathbf{w}^T\mathbf{W}\mathbf{w}$ can be zero. Even if we eliminate this possibility by replacing the transpose with the hermitian transpose, the condition

$$\mathbf{w}^H\mathbf{W}\mathbf{w}=1 \tag{7.3.41}$$

does not define the eigenvector uniquely because we can still multiply the eigenvector by any complex number of modulus one without changing the product in Eq. (7.3.41). Therefore, it is more reasonable to normalize the eigenvector by requiring that

$$\mathbf{v}^T\mathbf{B}\mathbf{w}=1,\qquad w_m=v_m=1\,, \tag{7.3.42}$$

where m is chosen so that both w_m and v_m are not small compared to other components of $\mathbf{w}$ and $\mathbf{v}$. The derivative of the normalization condition gives us

$$\frac{dw_m}{dx}=0,\quad \frac{dv_m}{dx}=0\,, \tag{7.3.43}$$

and together with Eq. (7.3.39) we can solve for the derivative of the eigenvector. This is the direct method for calculating the eigenvector derivatives. As in the symmetric case, the adjoint method for calculating the same derivatives is based on expressing the derivative of the eigenvector in terms of all the eigenvectors of the problem. Denoting the ith eigenvalue as μ_i and the corresponding eigenvectors as $\mathbf{w}^i$ and $\mathbf{v}^i$ we assume

$$\frac{d\mathbf{w}^k}{dx}=\sum_{j=1}^{l}c_{kj}\mathbf{w}^j\,, \tag{7.3.44}$$

and the coefficients c_{kj} are

$$c_{kj}=\frac{\mathbf{v}^{jT}(\frac{d\mathbf{A}}{dx}+\mu\frac{d\mathbf{B}}{dx})\mathbf{w}^k}{(\mu_k-\mu_j)\mathbf{v}^{jT}\mathbf{B}\mathbf{w}^j}\,,\qquad k\neq j\,, \tag{7.3.45}$$

and

$$c_{kk}=-\sum_{j\neq k}c_{kj}w_m^j\,. \tag{7.3.46}$$

The upper limit in the sum, l, is the order of the matrices $\mathbf{A}$ and $\mathbf{B}$. As in the symmetric case, it is possible to truncate the series without taking all the eigenvectors for the purpose of reducing the cost of the derivative calculation. This introduces an error which, in general, is problem dependent. Additional information on the various options for derivative calculation can be found in [10].

7.3.3 *Sensitivity Derivatives for Nonlinear Eigenvalue Problems*

In flutter and nonlinear vibration problems, we encounter eigenvalue problems where the dependence on the eigenvalue is not linear. For example, Bindolino and Mantegazza [19] consider an aeroelastic response problem which produces a transcendental eigenvalue problem of the form

$$\mathbf{A}(\mu, x)\mathbf{u} = 0 \tag{7.3.47}$$

Differentiating Eq. (7.3.47) we get

$$\mathbf{A}\frac{d\mathbf{u}}{dx} + \frac{d\mu}{dx}\frac{\partial \mathbf{A}}{\partial \mu} = -\frac{\partial \mathbf{A}}{\partial x}\mathbf{u} \tag{7.3.48}$$

Using the normalizing condition $u_m = 1$ we can solve Eq. (7.3.48) for $d\mathbf{u}/dx$ and $d\mu/dx$. Instead, it is also possible to use the adjoint method, employing the left eigenvector $\mathbf{v}$ satisfying

$$\mathbf{v}^T\mathbf{A} = 0, \qquad v_m = 1 \tag{7.3.49}$$

to obtain

$$\frac{d\mu}{dx} = -\frac{\mathbf{v}^T\frac{d\mathbf{A}}{dx}\mathbf{u}}{\mathbf{v}^T\frac{d\mathbf{A}}{d\mu}\mathbf{u}} \tag{7.3.50}$$

A common treatment of flutter problems is to have two real parameters representing the frequency and speed as an eigenpair instead of one complex eigenvalue. For example Murthy [20] replaces Eq. (7.3.47) by

$$\mathbf{A}(M, \omega)\mathbf{u} = 0\,, \tag{7.3.51}$$

where the Mach number, M, and the frequency, ω, are real parameters. Using this approach, differentiate Eq. (7.3.51), premultiply by $\mathbf{v}^T$, and use Eq. (7.3.49) to get

$$f_M \frac{dM}{dx} + f_\omega \frac{d\omega}{dx} = -f_x\,, \tag{7.3.52}$$

where

$$f_M = \mathbf{v}^T\frac{\partial \mathbf{A}}{\partial M}\mathbf{u}, \qquad f_\omega = \mathbf{v}^T\frac{\partial \mathbf{A}}{\partial \omega}\mathbf{u}, \qquad f_x = \mathbf{v}^T\frac{\partial \mathbf{A}}{\partial x}\mathbf{u}\,. \tag{7.3.53}$$

Multiplying Eq. (7.3.52) by $\bar{f}_\omega$ (the complex conjugate of f_ω) we get

$$f_M\bar{f}_\omega \frac{dM}{dx} + |\, f_\omega\,|^2 \frac{d\omega}{dx} = -\bar{f}_\omega f_x \tag{7.3.54}$$

The second term in Eq. (7.3.54) as well as dM/dx are real, so by taking the imaginary part of Eq. (7.3.54) we get

$$\frac{dM}{dx} = -\frac{Im(\bar{f}_\omega f_x)}{Im(f_M\bar{f}_\omega)} = -\frac{Im\left[\left(\mathbf{v}^T\frac{\partial \mathbf{A}}{\partial x}\mathbf{u}\right)\left(\bar{\mathbf{v}}^T\frac{\partial \bar{\mathbf{A}}}{\partial \omega}\bar{\mathbf{u}}\right)\right]}{Im\left[\left(\mathbf{v}^T\frac{\partial \mathbf{A}}{\partial M}\mathbf{u}\right)\left(\bar{\mathbf{v}}^T\frac{\partial \bar{\mathbf{A}}}{\partial \omega}\bar{\mathbf{u}}\right)\right]}\,. \tag{7.3.55}$$

Next, multiplying Eq. (7.3.52) by $\bar{f}_M$ and following a similar procedure we find

$$\frac{d\omega}{dx} = -\frac{Im\left[\left(\mathbf{v}^T\frac{\partial \mathbf{A}}{\partial x}\mathbf{u}\right)\left(\bar{\mathbf{v}}^T\frac{\partial \bar{\mathbf{A}}}{\partial M}\bar{\mathbf{u}}\right)\right]}{Im\left[\left(\mathbf{v}^T\frac{\partial \mathbf{A}}{\partial M}\mathbf{u}\right)\left(\bar{\mathbf{v}}^T\frac{\partial \bar{\mathbf{A}}}{\partial \omega}\bar{\mathbf{u}}\right)\right]}\,. \tag{7.3.56}$$

7.4 Sensitivity of Constraints on Transient Response

Compared to constraints on steady-state response, constraints on transient response depend on one additional parameter—time. That is, a typical constraint may be written as

$$g(\mathbf{u}, x, t) \geq 0\,, \qquad 0 \leq t \leq t_f\,, \tag{7.4.1}$$

where for simplicity we assume that the constraint must be satisfied from $t = 0$ to some final time t_f. For actual computation the constraint must be discretized at a series of n_t time points as

$$g_i = g(\mathbf{u}, x, t_i) \geq 0\,, \qquad i = 1, \ldots, n_t\,. \tag{7.4.2}$$

The distribution of time points has to be dense enough to preclude the possibility of significant constraint violation between time points. This type of constraint discretization can greatly increase the number of constraints, and thereby the cost of the optimization. Therefore it is desirable to find ways to remove the time dependence without substantially increasing the number of constraints.

7.4.1 *Equivalent Constraints*

One way of removing the time dependence of the constraint is to replace it with an equivalent integrated constraint which averages the severity of the constraint over the time interval. An example is the equivalent exterior constraint

$$\bar{g}(\mathbf{u}, x) = \left[\frac{1}{t_f}\int_0^{t_f} < -g(\mathbf{u}, x, t) >^2 dt\right]^{1/2}\,, \tag{7.4.3}$$

where $< a >$ denotes $max(a, 0)$. The equivalent constraint $\bar{g}$ is violated if the original constraint is violated for any finite period of time. If, however, $g(\mathbf{u}, x, t)$ is not violated anywhere, $\bar{g}(\mathbf{u}, x)$ is zero. The equivalent exterior constraint is identically zero in the feasible domain, and so no indication is provided when the constraint is almost critical. An equivalent constraint which is nonzero when the constraint is satisfied is based on the Kresselmeier-Steinhauser function, [21, 22], and Eq. (7.4.2)

$$\bar{g}(\mathbf{u}, x) = \frac{-1}{\rho} ln \left[\sum_{i=1}^{n_t} e^{-\rho g_i} dt\right]\,, \tag{7.4.4}$$

where ρ is a parameter which determines the relation between $\bar{g}$ and the most critical value of g, $g_{\min}$. Indeed, we can write Eq. (7.4.4) as

$$\bar{g} = g_{\min} - \frac{1}{\rho} ln \left[\sum_{i=1}^{n_t} e^{-\rho(g_i - g_{\min})} dt\right]\,. \tag{7.4.5}$$

And from Eq. (7.4.5) we get

$$g_{\min} \geq \bar{g} \geq g_{\min} - \frac{ln(n_t)}{\rho}\,, \tag{7.4.6}$$

so that $\bar{g}$ is an envelope constraint in that it is always more critical than g. The parameter ρ determines how much more critical $\bar{g}$ is. However, if ρ is made too large for the purpose of reducing the difference between $\bar{g}$ and g_{min}, the problem can become ill conditioned.

The savings obtained by replacing the discretized constraint, Eq. (7.4.2), by an equivalent one may seem illusory because the integral in Eq. (7.4.3) or the sum in Eq. (7.4.4) usually require the evaluation of $g(\mathbf{u}, x, t)$ at many time points. The savings are realized in the optimization effort and in the computation of constraint derivatives discussed later.

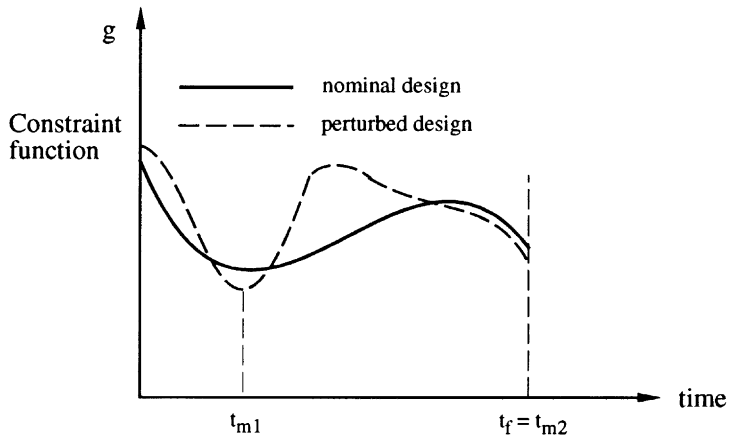


Figure 7.4.1 Critical points.

The disadvantage of equivalent constraints is that they may tend to blur design trends. Consider, for example a change in design which moves the constraint g from the solid to the dashed line in Fig. (7.4.1). An equivalent constraint $\bar{g}$ may become more positive, indicating a beneficial effect, while the situation has become more critical because we have moved closer to the constraint boundary ($g = 0$), at least at some time point t_{m1}. To avoid this blurring effect we use the critical point constraint replacing the original constraint by

$$g(\mathbf{u}, x, t_{mi}) \geq 0\,, \qquad i = 1, 2 \ldots\,, \tag{7.4.7}$$

where t_{mi} are time points where the constraint has a local minimum. Figure (7.4.1) shows a typical situation where the constraint function has two local minima: an interior one at t_{m1}, and a boundary minimum at t_{m2}. The local minima are critical points in the sense that they represent time points likely to be involved first in constraint violations.

One attractive feature of the critical point constraint is that, for the purpose of obtaining first derivatives, the location of the critical point may be assumed to be

fixed in time. This is shown by differentiating Eq. (7.4.7) with respect to the design variable x

$$\frac{dg(t_{mi})}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g}{\partial \mathbf{u}}\frac{d\mathbf{u}}{dx} + \frac{\partial g}{\partial t}\frac{dt_{mi}}{dx}. \tag{7.4.8}$$

The last term in Eq. (7.4.8) is always zero. At an interior minimum such as t_{m1} in Fig. (7.4.1) $\partial g/\partial t$ is zero. We get a boundary minimum when $\partial g/\partial t$ is positive at the left boundary or negative at the right boundary. This boundary minimum cannot move away from the boundary unless the slope, $\partial g/\partial t$ becomes zero. This means that as long as $\partial g/\partial t$ is nonzero at a boundary minimum, the minimum cannot move, so thatdt_{mi}/dx is zero.

7.4.2 *Derivatives of Constraints*

For the purpose of calculating derivatives of constraints we assume that the constraint is of the form

$$\bar{g}(\mathbf{u}, x) = \int_0^{t_f} p(\mathbf{u}, x, t)dt \geq 0. \tag{7.4.9}$$

This form represents most equivalent constraints, as well as the critical-point constraint, which can be obtained by defining

$$p(\mathbf{u}, x, t) = g(\mathbf{u}, x, t)\delta(t - t_{mi}). \tag{7.4.10}$$

The derivative of the constraint with respect to a design variable x is

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} (\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \mathbf{u}}\frac{d\mathbf{u}}{dx})dt. \tag{7.4.11}$$

To evaluate the integral we need to differentiate the equations of motion with respect to x. These equations are written in a general first-order form

$$\mathbf{A}\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u}, x, t), \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{7.4.12}$$

where $\mathbf{u}$ is a vector of generalized degrees of freedom, and $\mathbf{f}$ is a vector which includes contributions of external and internal loads.

We now discuss several methods for calculating the constraint derivative starting with the simplest—the direct method. As in the steady-state case, the direct method proceeds by differentiating Eq. (7.4.12) to obtain an equation for $d\mathbf{u}/dx$

$$\mathbf{A}\frac{d\dot{\mathbf{u}}}{dx} = \mathbf{J}\frac{d\mathbf{u}}{dx} - \frac{d\mathbf{A}}{dx}\dot{\mathbf{u}} + \frac{\partial \mathbf{f}}{\partial x}, \qquad \frac{d\mathbf{u}}{dx}(0) = 0, \tag{7.4.13}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$

$$J_{ij} = \frac{\partial f_i}{\partial u_j}. \tag{7.4.14}$$

The direct method consists of solving for $d\mathbf{u}/dx$ from Eq. (7.4.13), and then substituting into Eq. (7.4.11). The disadvantage of this method is that each design variable

requires the solution of a system of differential equations, Eq.(7.4.13). When we have many design variables and few constraint functions we can, as in the static case, use a vector of adjoint variables which depends only on the constraint functions and not on the design variables. To obtain the adjoint method, we pursue the standard procedure of multiplying the derivatives of the response equations, Eq. (7.4.13), by an adjoint vector and adding them to the derivatives of the constraint

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} (\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \mathbf{u}}\frac{d\mathbf{u}}{dx})dt + \int_0^{t_f} \boldsymbol{\lambda}^T(\mathbf{A}\frac{d\dot{\mathbf{u}}}{dx} - \mathbf{J}\frac{d\mathbf{u}}{dx} - \frac{\partial \mathbf{f}}{\partial x} + \frac{d\mathbf{A}}{dx}\dot{\mathbf{u}})dt \qquad (7.4.15)$$

We want to group together all the terms involving $d\mathbf{u}/dx$ and define the adjoint variable so that the coefficient of $d\mathbf{u}/dx$ will vanish. To do that, we need to integrate the term involving $d\dot{\mathbf{u}}/dx$. Integrating by parts and rearranging we obtain

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} \left\{ \frac{\partial p}{\partial x} - \boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{f}}{\partial x} - \frac{d\mathbf{A}}{dx}\dot{\mathbf{u}} \right) + \left[\frac{\partial p}{\partial \mathbf{u}} - \boldsymbol{\lambda}^T(\dot{\mathbf{A}} + \mathbf{J}) - (\dot{\boldsymbol{\lambda}})^T\mathbf{A} \right] \frac{d\mathbf{u}}{dx} \right\} dt + \boldsymbol{\lambda}^T\mathbf{A}\frac{d\mathbf{u}}{dx}\Big|_0^{t_f} . \qquad (7.4.16)$$

Equation (7.4.16) indicates that the adjoint variable should satisfy

$$\mathbf{A}^T\dot{\boldsymbol{\lambda}} + (\mathbf{J}^T + \dot{\mathbf{A}}^T)\boldsymbol{\lambda} = (\frac{\partial p}{\partial \mathbf{u}})^T, \qquad \boldsymbol{\lambda}(t_f) = 0 . \qquad (7.4.17)$$

Then from Eq. (7.4.16) we get

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} \left[\frac{\partial p}{\partial x} - \boldsymbol{\lambda}^T(\frac{\partial \mathbf{f}}{\partial x} - \frac{d\mathbf{A}}{dx}\dot{\mathbf{u}}) \right] dt , \qquad (7.4.18)$$

where we used the fact that $d\mathbf{u}/dx$ is zero at $t = 0$. Equation (7.4.17) is a system of ordinary differential equations for $\boldsymbol{\lambda}$ which are integrated backwards (from t_f to 0). This system has to be solved once for each constraint rather than once for each design variable. As in the static case, the direct method is preferable when the number of design variable is smaller than the number of constraints, and the adjoint method is preferable otherwise. Equation (7.4.17) takes a simpler form for the critical-point constraint

$$\mathbf{A}^T\dot{\boldsymbol{\lambda}} + (\mathbf{J}^T + \dot{\mathbf{A}}^T)\boldsymbol{\lambda} = (\frac{\partial g}{\partial \mathbf{u}})^T\delta(t - t_{mi}), \qquad \boldsymbol{\lambda}(t_f) = 0 . \qquad (7.4.19)$$

By integrating Eq. (7.4.19) from $t_{mi} - \epsilon$ to $t_{mi} + \epsilon$ for an infinitesimal ϵ, we can easily show that Eq. (7.4.19) is equivalent to

$$\mathbf{A}^T\dot{\boldsymbol{\lambda}} + (\mathbf{J}^T + \dot{\mathbf{A}}^T)\boldsymbol{\lambda} = 0, \qquad \boldsymbol{\lambda}^T(t_{mi}) = -\frac{\partial g}{\partial \mathbf{u}}(t_{mi})\mathbf{A}^{-1} . \qquad (7.4.20)$$

A third method available for derivative calculation is the Green's function approach [23]. This method is useful when the number of degrees of freedom in Eq.

(7.4.12) is smaller than either the number of design variables or the number of constraints. This can happen when the order of Eq. (7.4.12) has been reduced by employing modal analysis. The Green's function method will be discussed for the case of $\mathbf{A} = \mathbf{I}$ in Eq. (7.4.12) so that Eq. (7.4.13) becomes

$$\frac{d\dot{\mathbf{u}}}{dx} = \mathbf{J}\frac{d\mathbf{u}}{dx} + \frac{\partial \mathbf{f}}{\partial x}, \qquad \frac{d\mathbf{u}}{dx}(0) = 0\,. \tag{7.4.21}$$

The solution of Eq. (7.4.21) may be written [23] in terms of Green's function $\mathbf{K}(t,\tau)$ as

$$\frac{d\mathbf{u}}{dx} = \int_0^{t_f} \mathbf{K}(t,\tau)\frac{\partial \mathbf{f}}{\partial x}(\tau)d\tau\,, \tag{7.4.22}$$

where $\mathbf{K}(t,\tau)$ satisfies

$$\begin{aligned} &\dot{\mathbf{K}}(t,\tau) - \mathbf{J}(t)\mathbf{K}(t,\tau) = \delta(t-\tau)\mathbf{I}\,, \\ &\mathbf{K}(0,\tau) = 0\,, \end{aligned} \tag{7.4.23}$$

and where $\delta(t-\tau)$ is the Dirac delta function. It is easy to check, by direct substitution, that $d\mathbf{u}/dx$ defined by Eq. (7.4.22), indeed satisfies Eq. (7.4.21).

If the elements of $\mathbf{J}$ are bounded then it can be shown that Eq. (7.4.23) is equivalent to

$$\begin{aligned} &\mathbf{K}(t,\tau) = \mathbf{0}\,, && t < \tau\,, \\ &\mathbf{K}(\tau,\tau) = \mathbf{I}\,, && \\ &\dot{\mathbf{K}}(t,\tau) - \mathbf{J}(t)\mathbf{K}(t,\tau) = 0, && t > \tau\,. \end{aligned} \tag{7.4.24}$$

Therefore, the integration of Eq. (7.4.22) needs to be carried out only up to $\tau = t$. To see how $d\mathbf{u}/dx$ is evaluated with the aid of Eq. (7.4.24), assume that we divide the interval $0 \le t \le t_f$ into n subintervals with end points at $\tau_0 = 0 < t_1 < \ldots < t_n = t_f$. The end points τ_i are dense enough to evaluate Eq. (7.4.22) by numerical integration and to interpolate $d\mathbf{u}/dx$ to other time points of interest with sufficient accuracy. We now define the initial value problem

$$\begin{aligned} &\dot{\mathbf{K}}(t,\tau_k) - \mathbf{J}(t)\mathbf{K}(t,\tau_k) = 0\,, \\ &\mathbf{K}(\tau_k,\tau_k) = \mathbf{I}\,, \qquad\qquad k = 0,1,\ldots,n-1\,. \end{aligned} \tag{7.4.25}$$

Each of the equations in (7.4.25) is integrated from τ_k to τ_{k+1} to yield $\mathbf{K}(\tau_{k+1},\tau_k)$. The value of $\mathbf{K}$ for any other pair of points is given by (see [23] for proof)

$$\mathbf{K}(\tau_j,\tau_k) = \mathbf{K}(\tau_j,\tau_{j-1})\mathbf{K}(\tau_{j-1},\tau_{j-2})\ldots\mathbf{K}(\tau_{k+1},\tau_k)\,, \qquad j > k\,. \tag{7.4.26}$$

The solution for $\mathbf{K}$ is equivalent to solving n_m systems of the type of Eq. (7.4.13) or (7.4.20) where n_m is the order of the vector $\mathbf{u}$. Therefore, the Green's function method should be considered for cases where the number of design variables and constraints both exceed n_m. This is likely to happen when the order of the system has been reduced by using some type of modal or reduced-basis approximation.

Example 7.4.1

We consider a single degree-of-freedom system governed by the differential equation

$$a\dot{u} = (u - b)^2, \qquad u(0) = 0\,,$$

and a constraint on the response u in the form

$$g(u) = c - u(t) \geq 0, \qquad 0 \leq t \leq t_f\,.$$

The response has been calculated and found to be monotonically increasing, so that the critical-point constraint takes the form

$$\bar{g}(u) = g[(u(t_f)] = c - u(t_f)\,.$$

We want to use the direct, adjoint, and Green's function methods to calculate the derivative of $\bar{g}$ with respect to a and b.

The problem may be integrated directly to yield

$$u = \frac{b^2 t}{bt + a}\,.$$

In our notation

$$\mathbf{A} = a, \qquad \mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 2(u - b)\,.$$

Direct Method. The direct method requires us to write Eq. (7.4.13) for $x = a$ and $x = b$. For $x = a$ we obtain

$$a\frac{d\dot{u}}{da} = 2(u - b)\frac{du}{da} - \dot{u}, \qquad \frac{du}{da}(0) = 0\,.$$

In general the values for u and $\dot{u}$ would be available only numerically, so that the equation for du/da will also be integrated numerically. Here, however, we have the closed-form solution for u, so that we can substitute it into the derivative equation

$$a\frac{d\dot{u}}{da} = \frac{2ab}{bt + a}\frac{du}{da} - \frac{ab^2}{(bt + a)^2}, \qquad \frac{du}{da}(0) = 0\,,$$

and solve analytically to obtain

$$\frac{du}{da} = -\frac{b^2 t}{(bt + a)^2}\,.$$

Then

$$\frac{d\bar{g}}{da} = -\frac{du}{da}(t_f) = \frac{b^2 t_f}{(bt_f + a)^2}\,.$$

We now repeat the process for $x = b$. Equation (7.4.13) becomes

$$a\frac{d\dot{u}}{db} = 2(u-b)\frac{du}{db} - 2(u-b), \qquad \frac{du}{db}(0) = 0\,.$$

Solving for du/db we obtain

$$\frac{du}{db} = \frac{b^2t^2 + 2abt}{(bt+a)^2}\,,$$

and then

$$\frac{d\bar{g}}{db} = -\frac{du}{db}(t_f) = -\frac{b^2t_f^2 + 2abt_f}{(bt_f+a)^2}\,.$$

Adjoint Method. The adjoint method requires the solution of Eq. (7.4.20) which becomes

$$a\dot{\lambda} + 2(u-b)\lambda = 0, \qquad \lambda(t_f) = -\frac{1}{a}\frac{\partial g}{\partial u}(t_f) = \frac{1}{a}\,,$$

or

$$a\dot{\lambda} - \frac{2ab}{bt+a}\lambda = 0, \qquad \lambda(t_f) = \frac{1}{a}\,,$$

which can be integrated to yield

$$\lambda = \frac{1}{a}\left(\frac{bt+a}{bt_f+a}\right)^2.$$

Then $d\bar{g}/da$ is obtained from Eq. (7.4.18) which becomes

$$\frac{d\bar{g}}{da} = \int_0^{t_f} \lambda\dot{u}dt = \int_0^{t_f} \frac{1}{a}\left(\frac{bt+a}{bt_f+a}\right)^2\frac{ab^2}{(bt+a)^2}dt = \frac{b^2t_f}{(bt_f+a)^2}\,.$$

Similarly, $d\bar{g}/db$ is

$$\frac{d\bar{g}}{db} = \int_0^{t_f} 2\lambda(u-b)dt = -\frac{2}{a}\int_0^{t_f}\left(\frac{bt+a}{bt_f+a}\right)^2\frac{ab}{bt+a}dt = -\frac{b^2t_f^2 + 2abt_f}{(bt_f+a)^2}\,.$$

Green's Function Method. We recast the problem as

$$\dot{u} = (u-b)^2/a\,,$$

so that the Jacobian J is

$$J = 2(u-b)/a\,.$$

Equation (7.4.24) becomes

$$\dot{k}(t,\tau) - [2(u-b)/a]k(t,\tau) = 0, \qquad k(\tau,\tau) = 1\,,$$

or

$$\dot{k}(t,\tau) + \frac{2b}{bt+a}k(t,\tau) = 0\,.$$

The solution for k is

$$k = \left(\frac{b\tau + a}{bt + a}\right)^2, \qquad t \geq \tau,$$

so that from Eq. (7.4.22)

$$\frac{du}{da} = \int_0^{t_f} \frac{\partial f}{\partial a} k d\tau = -\int_0^{t_f} \left(\frac{b\tau + a}{bt + a}\right) \frac{(u-b)^2}{a^2} d\tau = -\frac{b^2 t}{(bt+a)^2}.$$

Similarly

$$\frac{du}{db} = \int_0^{t_f} \frac{\partial F}{\partial b} k d\tau = -\int_0^{t_f} 2\left(\frac{b\tau + a}{bt + a}\right)^2 \left(\frac{u-b}{a}\right) d\tau = -\frac{b^2t^2 + 2abt}{(bt+a)^2}.$$

• • •

7.4.3 *Linear Structural Dynamics*

For the case of linear structural dynamics it may be advantageous to retain the second-order equations of motion rather than reduce them to a set of first-order equations. It is also common to use modal reduction for this case. In this section we discuss the application of the direct and adjoint methods to this special case. The equations of motion are written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t). \tag{7.4.27}$$

Most often the problem is reduced in size by expressing $\mathbf{u}$ in terms of m basis functions $\mathbf{u}^i$, $i = 1, \ldots m$, where m is usually much less than the number of degrees of freedom of the original system, Eq.(7.4.27)

$$\mathbf{u} = \mathbf{U}\mathbf{q}, \tag{7.4.28}$$

where $\mathbf{U}$ is a matrix with $\mathbf{u}^i$ as columns. Then a reduced set of equations can be written as

$$\mathbf{M}_R\ddot{\mathbf{q}} + \mathbf{C}_R\dot{\mathbf{q}} + \mathbf{K}_R\mathbf{q} = \mathbf{f}_R, \tag{7.4.29}$$

where

$$\mathbf{M}_R = \mathbf{U}^T\mathbf{M}\mathbf{U}, \quad \mathbf{C}_R = \mathbf{U}^T\mathbf{C}\mathbf{U}, \quad \mathbf{K}_R = \mathbf{U}^T\mathbf{K}\mathbf{U}, \quad \mathbf{f}_R = \mathbf{U}^T\mathbf{f}. \tag{7.4.30}$$

When the basis functions are the first m natural vibration modes of the structure scaled to unit modal masses, $\mathbf{U}$ satisfies the equation

$$\mathbf{K}\mathbf{U} - \mathbf{M}\mathbf{U}\boldsymbol{\theta}^2 = 0, \tag{7.4.31}$$

where $\boldsymbol{\theta}$ is a diagonal matrix with the ith natural frequency ω_i in the ith row. In that case $\mathbf{K}_R = \boldsymbol{\theta}^2$ and $\mathbf{M}_R = \mathbf{I}$ are diagonal matrices. For special forms of damping, the damping matrix $\mathbf{C}_R$ is also diagonal so that the system Eq. (7.4.29) is uncoupled. After $\mathbf{q}$ is calculated from Eq. (7.4.29) we can use Eq. (7.4.28) to calculate $\mathbf{u}$. This modal reduction method is known as the mode-displacement method.

When the load $\mathbf{f}$ has spatial discontinuities the convergence of the modal approximation, Eq. (7.4.29) can be very slow [24, 25]. The convergence can be dramatically accelerated by using the mode acceleration method, originally proposed by Williams [26]. The mode acceleration method can be derived by rewriting Eq. (7.4.27) as

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{f} - \mathbf{K}^{-1}\mathbf{C}\dot{\mathbf{u}} - \mathbf{K}^{-1}\mathbf{M}\ddot{\mathbf{u}}\,. \tag{7.4.32}$$

The first term in Eq. (7.4.32) is called the quasi-static solution because it represents the response of the structure if the loads are applied very slowly. The second term and third terms are approximated in terms of the modal solution. It can be shown (e.g., Greene [27]) that $\mathbf{K}^{-1}$ can be approximated as

$$\mathbf{K}^{-1} = \mathbf{U}\boldsymbol{\theta}^{-2}\mathbf{U}^T \tag{7.4.33}.$$

Using this approximation for the second and third terms of Eq. (7.4.32) we get

$$\mathbf{u} \approx \mathbf{K}^{-1}\mathbf{f} - \mathbf{U}\boldsymbol{\theta}^{-2}\mathbf{C}_R\dot{\mathbf{q}} - \mathbf{U}\boldsymbol{\theta}^{-2}\ddot{\mathbf{q}}\,. \tag{7.4.34}$$

This approximation is exact when $\mathbf{U}$ contains the full set of vibration modes. Note that $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ in Eq. (7.4.34) are obtained from the mode-displacement solution, Eq. (7.4.29). Therefore, there is no difference in velocities and accelerations between the mode-displacement and the mode acceleration methods.

In considering the calculation of sensitivities we treat first the mode-displacement method. The direct method of calculating the response sensitivity is obtained by differentiating Eq. (7.4.29) to obtain

$$\mathbf{M}_R\frac{d\ddot{\mathbf{q}}}{dx} + \mathbf{C}_R\frac{d\dot{\mathbf{q}}}{dx} + \mathbf{K}_R\frac{d\mathbf{q}}{dx} = \mathbf{r}\,, \tag{7.4.35}$$

where

$$\mathbf{r} = \frac{d\mathbf{f}_R}{dx} - \frac{d\mathbf{M}_R}{dx}\ddot{\mathbf{q}} - \frac{d\mathbf{M}_R}{dx}\dot{\mathbf{q}} - \frac{d\mathbf{K}_R}{dx}\mathbf{q}\,. \tag{7.4.36}$$

The derivative of $\mathbf{K}_R$ with respect to x is given by Eq. (7.3.33), and similar expressions are used for the derivatives of $\mathbf{M}_R$, $\mathbf{C}_R$, and $\mathbf{f}_R$. The calculation is simplified considerably by using a fixed set of basis functions $\mathbf{U}$ or neglecting the effect of the change in the modes. In some cases (e.g., [28]) the error associated with neglecting the effect of changing modes is small. When this error is unacceptable we have to face the costly calculation of the derivatives of the modes needed for calculating the derivatives of the reduced matrices, such as Eq. (7.3.33). Fortunately it was found by Greene [27] that the cost of calculating the derivatives of the modes can be substantially reduced by using the modified modal method Eq. (7.3.15) keeping only the first term in this equation. This approximation to the derivatives of the modes may not always be accurate, but it appears to be sufficient for calculating the sensitivity of the dynamic response.

For the adjoint method we consider a constraint in the form of Eq. (7.4.9)

$$\bar{g}(\mathbf{q}, x) = \int_0^{t_f} p(\mathbf{q}, x, t)dt \geq 0\,, \tag{7.4.37}$$

so that

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} (\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dx}) dt \,. \tag{7.4.38}$$

To avoid the calculation of $d\mathbf{q}/dx$ we multiply the response derivative equation, Eq. (7.4.35), by an adjoint vector, $\boldsymbol{\lambda}$, and add to the derivative of the constraint

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} (\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dx}) dt + \int_0^{t_f} \boldsymbol{\lambda}^T (-\mathbf{M}_R \frac{d\ddot{\mathbf{q}}}{dx} - \mathbf{C}_R \frac{d\dot{\mathbf{q}}}{dx} - \mathbf{K}_R \frac{d\mathbf{q}}{dx} + \mathbf{r}) dt \,. \tag{7.4.39}$$

We want to get rid of the response derivative terms by selecting $\boldsymbol{\lambda}$ appropriately. We use integration by parts to get rid of time derivatives in the response derivative terms. We obtain

$$\begin{aligned} \frac{d\bar{g}}{dx} = \int_0^{t_f} \left\{ \frac{\partial p}{\partial x} - \boldsymbol{\lambda}^T \mathbf{r} + \left[\frac{\partial p}{\partial \mathbf{q}} - \ddot{\boldsymbol{\lambda}}^T \mathbf{M}_R + \dot{\boldsymbol{\lambda}}^T \mathbf{C}_R - \boldsymbol{\lambda}^T \mathbf{K}_R \right] \frac{d\mathbf{q}}{dx} \right\} dt \\ - \boldsymbol{\lambda}^T \mathbf{M}_R \frac{d\dot{\mathbf{q}}}{dx}\Big|_0^{t_f} + \dot{\boldsymbol{\lambda}}^T \mathbf{M}_R \frac{d\mathbf{q}}{dx}\Big|_0^{t_f} - \boldsymbol{\lambda}^T \mathbf{C}_R \frac{d\mathbf{q}}{dx}\Big|_0^{t_f} \,. \end{aligned} \tag{7.4.40}$$

If the initial conditions do not depend on the design variable x, Eq. (7.4.40) suggests the following definition for $\boldsymbol{\lambda}$

$$\mathbf{M}_R \ddot{\boldsymbol{\lambda}} - \mathbf{C}_R \dot{\boldsymbol{\lambda}} + \mathbf{K}_R \boldsymbol{\lambda} = (\frac{\partial p}{\partial \mathbf{q}})^T, \qquad \boldsymbol{\lambda}(t_f) = \dot{\boldsymbol{\lambda}}(t_f) = 0 \,, \tag{7.4.41}$$

and then Eq. (7.4.40) becomes

$$\frac{d\bar{g}}{dx} = \int_0^{t_f} (\frac{\partial p}{\partial x} - \boldsymbol{\lambda}^T \mathbf{r}) dt \,. \tag{7.4.42}$$

For the mode-acceleration method we consider only the direct method. We start by differentiating Eq. (7.4.27) and rearranging it as

$$\frac{d\mathbf{u}}{dx} = \mathbf{K}^{-1} \left[\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{u} - \mathbf{C}\frac{d\dot{\mathbf{u}}}{dx} - \frac{d\mathbf{C}}{dx}\dot{\mathbf{u}} - \mathbf{M}\frac{d\ddot{\mathbf{u}}}{dx} - \frac{d\mathbf{M}}{dx}\ddot{\mathbf{u}} \right] . \tag{7.4.43}$$

Next we use Eq. (7.4.34) to approximate the second term, and the modal expansion Eq. (7.4.28) to approximate the other terms to get

$$\begin{aligned} \frac{d\mathbf{u}}{dx} \approx \mathbf{K}^{-1} \Bigg[\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx} [\mathbf{K}^{-1}\mathbf{f} - \mathbf{U}\boldsymbol{\theta}^{-2}\mathbf{C}_R\dot{\mathbf{q}} - \mathbf{U}\boldsymbol{\theta}^{-2}\ddot{\mathbf{q}}] - \\ \mathbf{C}\mathbf{U}\frac{d\dot{\mathbf{q}}}{dx} - \frac{d\mathbf{C}}{dx}\mathbf{U}\dot{\mathbf{q}} - \mathbf{M}\mathbf{U}\frac{d\ddot{\mathbf{q}}}{dx} - \frac{d\mathbf{M}}{dx}\mathbf{U}\ddot{\mathbf{q}} \Bigg] . \end{aligned} \tag{7.4.44}$$

Finally we use the modal approximation to $\mathbf{K}^{-1}$, Eq. (7.4.33) to obtain

$$\begin{aligned} \frac{d\mathbf{u}}{dx} \approx \mathbf{K}^{-1} \left[\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{K}^{-1}\mathbf{f} \right] + \\ \mathbf{U}\boldsymbol{\theta}^{-2}\mathbf{U}^T \left[\frac{d\mathbf{K}}{dx}\mathbf{U}\boldsymbol{\theta}^{-2}\mathbf{C}_R\dot{\mathbf{q}} - \frac{d\mathbf{C}}{dx}\mathbf{U}\dot{\mathbf{q}} - \mathbf{C}\mathbf{U}\frac{d\dot{\mathbf{q}}}{dx} \right] + \\ \mathbf{K}^{-1} \left[\frac{d\mathbf{K}}{dx}\mathbf{U}\boldsymbol{\theta}^{-2} - \frac{d\mathbf{M}}{dx}\mathbf{U} \right] \ddot{\mathbf{q}} - \mathbf{U}\boldsymbol{\theta}^{-2}\frac{d\ddot{\mathbf{q}}}{dx} \,. \end{aligned} \tag{7.4.45}$$

Note that the calculation involves the solution of Eqs. (7.4.29) and (7.4.35) for $\mathbf{q}$ and $d\mathbf{q}/dx$, followed by Eq. (7.4.45) for retrieving the $d\mathbf{u}/dx$. Additional details can be found in [27].

7.5 Exercises

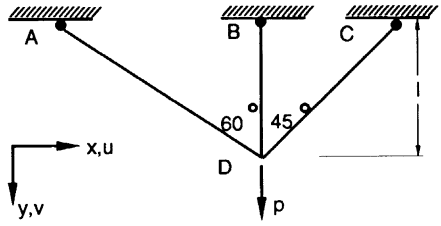


Figure 7.5.1 Three-bar truss.

1. Write a program using the finite-element method to calculate the displacements and stresses in the three-bar truss shown in Fig. (7.5.1). Also calculate the derivative of the stress in member A with respect to A_A by the forward- and central-difference techniques. Consider the case $A_A = A_B = kA_C$. (a)Take $k = 10^{-m}$ where m is the number of decimal digits you use in the computation minus two. Find the optimum step size. (b)Find the smallest value of k that allows an error of less than 10 percent.

2. Calculate the derivatives of the stress in member A of the three bar truss of Fig. (7.5.1) at a design point where all three cross-sectional areas have the same value A. First calculate the derivative with respect to the cross-sectional area of A using the direct and adjoint method. Next calculate the derivative with respect to the cross-sectional areas of members B and C using one method only.

3. Calculate all the second derivatives of the stress in member A of problem 2 with respect to the three cross-sectional areas.

4. Obtain a method for calculating third derivatives of constraints on displacement and stresses (static case).

5. Obtain a finite-element approximation to the first vibration frequency of the truss of problem 1 in terms of A, l, Young's modulus E and the mass density ρ. Assume that there is no bending. Then calculate the derivative of the frequency with respect to the cross-sectional area of the three members.

6. Calculate the derivative of the lowest (in absolute magnitude) eigenvalue of problem 5 with respect to the strength c of a horizontal dashpot at joint D: (i) when $c = 0$; (ii) when c is selected (by linear extrapolation on the basis of part (i)) to make the damping ratio (negative of real part over the absolute value of the eigenvalue) be 0.05.

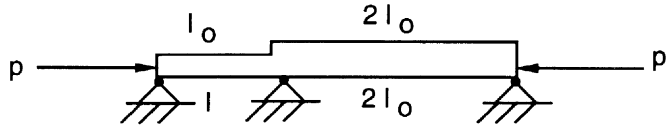


Figure 7.5.2 Two-span beam.

7. The beam shown in Fig. (7.5.2) needs to be stiffened to increase its buckling load. Calculate the derivative of the buckling load with respect to the moment of inertia of the left and right segments, and decide what is the most economical way of stiffening the beam. Assume that the cost is proportional to the mass, and the cross-sectional area is proportional to the square root of the moment of inertia.

8. Obtain an expression for the second derivatives of the buckling load with respect to structural parameters.

9. Repeat Example 7.3.4 for the derivative with respect to c instead of k.

10. Consider the equation of motion for a mass-spring-damper system

$$m\ddot{w} + c\dot{w} + kw = f(t)$$

where $f(t) = f_0 H(t)$ is a step function, and $w(0) = \dot{w}(0) = 0$. Calculate the derivative of the maximum displacement with respect to c for the case $k/m = 4.$, $c/m = 0.05$, $f_0/m = 2.$ using the direct method.

11. Obtain the derivatives of the maximum displacement in Problem 10 with respect to c, m, f_0 and k using the adjoint method.

12. Solve problem 10 using Green's function method.

13. Solve problem 10 using the mode-displacement method and mode-acceleration methods with a single mode.

7.6 References

[1] Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H., "Computing Forward-Difference Intervals for Numerical Optimization", SIAM J. Sci. and Stat. Comp., Vol. 4, No. 2, pp. 310-321, June 1983.

[2] Iott, J., Haftka, R.T., and Adelman, H.M., "Selecting Step Sizes in Sensitivity Analysis by Finite Differences," NASA TM- 86382, 1985.

[3] Haftka, R.T., "Sensitivity Calculations for Iteratively Solved Problems," International Journal for Numerical Methods in Engineering, Vol. 21, pp.1535–1546, 1985.

[4] Haftka, R.T., "Second-Order Sensitivity Derivatives in structural Analysis", AIAA Journal, Vol. 20, pp.1765-1766, 1982.

[5] Barthelemy, B., Chon, C.T., and Haftka, R.T., " Sensitivity Approximation of Static Structural Response", paper presented at the First World Congress on Computational Mechanics, Austin Texas, Sept. 1986.

[6] Barthelemy, B., and Haftka, R.T., "Accuracy Analysis of the Semi-analytical Method for Shape Sensitivity Calculations," Mechanics of Structures and Machines, 18, 3, pp. 407–432, 1990.

[7] Barthelemy, B., Chon, C.T., and Haftka, R.T., "Accuracy Problems Associated with Semi-Analytical Derivatives of Static Response," Finite Elements in Analysis and Design, 4, pp. 249–265, 1988.

[8] Haug, E.J., Choi, K.K., and Komkov, V., Design Sensitivity Analysis of Structural Systems, Academic Press, 1986.

[9] Cardani, C. and Mantegazza, P., "Calculation of Eigenvalue and Eigenvector Derivatives for Algebraic Flutter and Divergence Eigenproblems," AIAA Journal, Vol. 17, pp.408–412, 1979.

[10] Murthy, D.V., and Haftka, R.T., "Derivatives of Eigenvalues and Eigenvectors of General Complex Matrix", International Journal for Numerical Methods in Engineering, 26, pp. 293–311,1988.

[11] Nelson, R.B., "Simplified Calculation of Eigenvector Derivatives," AIAA Journal, Vol. 14, pp. 1201–1205,1976.

[12] Rogers, L.C., "Derivatives of Eigenvalues and Eigenvectors", AIAA Journal, Vol. 8, No. 5, pp. 943-944, 1970.

[13] Wang, B.P., Improved Approximate Methods for Computing Eigenvector Derivatives in Structural Dynamics," AIAA Journal, 29 (6), pp. 1018–1020, 1991.

[14] Sutter, T.R., Camarda, C.J., Walsh, J.L., and Adelman, H.M., "Comparison of Several Methods for the Calculation of Vibration Mode Shape Derivatives", AIAA Journal, 26 (12), pp. 1506–1511, 1988.

[15] Ojalvo, I.U., "Efficient Computation of Mode-Shape Derivatives for Large Dynamic Systems" AIAA Journal, 25, 10, pp. 1386–1390, 1987.

[16] Mills-Curran, W.C., "Calculation of Eigenvector Derivatives for Structures with Repeated Eigenvalues", AIAA Journal, 26 (7), pp. 867–871, 1988.

[17] Dailey, R.L., "Eigenvector Derivatives with Repeated Eigenvalues", AIAA Journal, 27 (4), pp. 486–491, 1989.

[18] Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

[19] Bindolino, G., and Mantegazza, P., "Aeroelastic Derivatives as a Sensitivity Analysis of Nonlinear Equations," AIAA Journal, 25 (8), pp. 1145–1146, 1987.

[20] Murthy, D.V., "Solution and Sensitivity of a Complex Transcendental Eigenproblem with Pairs of Real Eigenvalues," Proceedings of the 12th Biennial ASME Conference on Mechanical Vibration and Noise (DE-Vol. 18-4), Montreal Canada, September 17–20, 1989, pp. 229–234 (in press Int. J. Num. Meth. Eng. 1991).

[21] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index", Proceedings of IFAC Symposium on Computer Aided Design of Control Systems, Zurich, Switzerland, 1979, pp.113-117.

[22] Barthelemy, J-F. M., and Riley, M. F., "Improved Multilevel Optimization Approach for the Design of Complex Engineering Systems", AIAA Journal, 26 (3), pp. 353–360, 1988.

[23] Kramer, M.A., Calo, J.M., and Rabitz, H., "An Improved Computational Method for Sensitivity Analysis: Green's Function Method with AIM," Appl. Math. Modeling, Vol. 5, pp.432–441, 1981.

[24] Sandridge, C.A. and Haftka, R.T., "Accuracy of Derivatives of Control Performance Using a Reduced Structural Model," Paper presented at the AIAA Dynamics Specialists Meeting, Monterey California, April, 1987.

[25] Tadikonda, S.S.K. and Baruh, H., "Gibbs Phenomenon in Structural Mechanics," AIAA Journal, 29 (9), pp. 1488–1497, 1991.

[26] Williams, D., "Dynamic Loads in Aeroplanes Under Given Impulsive Loads with Particular Reference to Landing and Gust Loads on a Large Flying Boat," Great Britain Royal Aircraft Establishment Reports SME 3309 and 3316, 1945.

[27] Greene, W.H., Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis, Ph.D dissertation, Virginia Polytechnic Institute and State University, August 1989.

[28] Greene, W.H., and Haftka, R.T., "Computational Aspects of Sensitivity Calculations in Transient Structural Analysis" , Computers and Structures, 32, pp. 433–443, 1989.

Introduction to Variational Sensitivity Analysis 8

The methods for discrete sensitivity analysis discussed in the previous chapter are very general in that they may be applied to a variety of nonstructural sensitivity analyses involving systems of linear equations, eigenvalue problems, etc. However, for structural applications they have two disadvantages. First, not all methods of structural analysis lead to the type of discretized equations that are discussed in Chapter 7. For example, shell-of-revolution codes such as FASOR [1] directly integrate the equations of equilibrium without first converting them to systems of algebraic equations. Second, operating on the discretized equations often requires access to the source code of the structural analysis program which implements these equations. Unfortunately, many of the popular structural analysis programs do not provide such access to most users. It is desirable, therefore, to have sensitivity analysis methods that are more generally applicable and can be implemented without extensive access to and knowledge of the insides of structural analysis programs. Variational methods of sensitivity analysis achieve this goal by differentiating the equations governing the structure before they are discretized. The resulting sensitivity equations can then be solved with the aid of a structural analysis program. It is not even essential that the same program be used for the analysis and the sensitivity calculations.

As an example of this approach consider the Euler-Bernoulli plane beam governed by the differential equation

$$\frac{d^2}{dx^2}\left[EI(x)\frac{d^2w}{dx^2}\right] = q(x)\,, \tag{8.1}$$

where w denotes the transverse displacement, EI is the flexural rigidity and q is the load. Equation (8.1) is supplemented by appropriate boundary conditions. Imagine that we have to design a class of structures that are modeled well by this beam equation with complex loading and boundary conditions corresponding to intermediate supports. We have an old computer program, written to solve this problem, for which we do not have any programming documentation. We now want to use this program to calculate the sensitivity of the response to changes in the stiffness properties of the beam. Finite difference sensitivity calculations are, of course, the first choice in this type of situation. However, difficulties in finding good step-sizes for accurate derivatives (see Section 7.1) force us to consider the calculation of analytical derivatives.

We start by differentiating Eq. (8.1) with respect to a parameter p (since x is used in this chapter to denote a coordinate variable, we use p for the generic design variable) which affects the moment of inertia of the beam over part of the span

$$\frac{d^2}{dx^2}\left[EI(x)\frac{d^2w_{,p}}{dx^2}\right] = -\frac{d^2}{dx^2}\left[EI_{,p}(x)\frac{d^2w}{dx^2}\right], \tag{8.2}$$

where a comma subscript followed by p denotes differentiation with respect to p. Comparing Eqs. (8.1) and (8.2) we note that both have the same left-hand side in the unknown functions, w and $w_{,p}$, respectively. If we treat the right-hand side of Eq. (8.2) as a load, the similarity is complete. As in Chapter 7, we call this right-hand side a pseudo load. If that pseudo load is applied to the beam, the response to the pseudo load will be the derivative of the original response with respect to p. We now have a prescription for using our beam analysis program for calculating sensitivity. We need to write a postprocessor that will take the solution w and the derivative of the moment of inertia $I_{,p}$, construct the pseudo load, and output it in a form required by our program for the load definition.

There are many approaches to sensitivity calculations using variational methods. Reference [2] provides an excellent exposition of the methods, as well as a sound mathematical basis. The present chapter has the more modest aim of introducing some of the basic methods with a few examples. The discussion is based on the principle of virtual work which provides a good foundation for both discrete and continuum based sensitivity analysis. Most of the material in this chapter is limited to the calculation of sensitivity with respect to stiffness (sizing) parameters, with the last section introducing sensitivity with respect to shape.

The results obtained in this chapter depend often on the differentiability of the structural response with respect to the sizing or shape parameter. Throughout the chapter it is assumed that the structural response is differentiable with respect to the parameter in question, and that the sensitivity field has the same differentiability properties with respect to space coordinates as the original response.

Finally, the material in this chapter is rather abstract, and many readers may want to skip the derivations and focus only on the implementation of the final results of the derivations. It is suggested that the introductory part of Section 8.1 be read to understand the notation, and then the implementation notes at the end of each section be read to obtain information on how to implement sensitivity calculations using structural analysis programs without the need for access to the source code of the program.

8.1 Linear Static Analysis

The equations governing static structural response include the strain-displacement relation, the constitutive equations, and the equations of equilibrium. These equations take different forms depending on whether we consider the full three-dimensional problem or special cases such as plane-stress, plate bending analysis or beam analysis. For the sake of obtaining results that are generally applicable, we adopt Budiansky's

operator notation for these equations. The notation is compact and allows for ease in algebraic manipulations. However, it is abstract, and it is not always easy to grasp. The reader who has trouble with the notation may want to translate the abstract equations for a specific case such as plane-stress or beam analysis. For linear analysis, the strain displacement relation is written as

$$\boldsymbol{\varepsilon} = \mathbf{L}_1(\mathbf{u}), \tag{8.1.1}$$

where $\boldsymbol{\varepsilon}$ is the generalized strain tensor, and $\mathbf{u}$ the displacement vector, and $\mathbf{L}_1$ is a linear differential operator. For example, for Euler-Bernoulli beam analysis the generalized strain tensor has one component, the curvature κ, and Eq. (8.1.1) translates into

$$\kappa = w_{,xx}\,. \tag{8.1.2}$$

The strain is obtained from the generalized strain κ as $\epsilon = -y\kappa$ where y is the distance from the neutral axis of the beam. However, in using the principle of virtual work it is convenient to use the generalized strain and stress tensors rather than actual strains and stresses.

For plane-stress analysis $\boldsymbol{\varepsilon}$ has actual strain components ϵ_x, ϵ_y, and γ_{xy}, while $\mathbf{L}_1$ is given as

$$\mathbf{L}_1(\mathbf{u}) = \begin{Bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{Bmatrix}. \tag{8.1.3}$$

However, the constitutive equations are written in terms of generalized stresses which are the stress resultants.

The linear constitutive equations are the appropriate version of Hooke's law, and may be written as

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i), \tag{8.1.4}$$

where $\boldsymbol{\sigma}$ is the generalized stress tensor, $\mathbf{D}$ is the material stiffness matrix, and $\boldsymbol{\varepsilon}^i$ is the initial strain (e.g. due to an applied temperature field). For example, for the plane-stress problem, $\boldsymbol{\sigma}$ includes the stress resultant components N_x, N_y, and N_{xy}, while for a beam bending the stress resultant is the section bending moment M, and the constitutive equation is

$$M = EI(\kappa - \kappa^i), \tag{8.1.5}$$

where E is Young's modulus and I is the section moment of inertia.

The equations of equilibrium are written via the principle of virtual work as

$$\boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon} = \mathbf{f} \bullet \delta\mathbf{u}, \tag{8.1.6}$$

where $\mathbf{f}$ is the applied load field, and a bullet denotes a scalar product followed by integration over the structural domain. For example, for the plane stress case

$$\boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon} = \int \boldsymbol{\sigma} \cdot \delta\boldsymbol{\varepsilon}\, dA = \int (N_x \delta\epsilon_x + N_y \delta\epsilon_y + 2N_{xy}\delta\gamma_{xy}) dA\,,$$

and

$$\mathbf{f} \bullet \delta\mathbf{u} = \int \mathbf{f} \cdot \delta\mathbf{u} dA = \int (f_x \delta u + f_y \delta v) dA + \int (T_x \delta u + T_y \delta v) d\Gamma_T \,, \tag{8.1.7}$$

where f_x and f_y are body forces per unit area, and T_x, T_y are tractions on the loaded boundary Γ_T

The virtual displacement field $\delta\mathbf{u}$ must be differentiable and satisfy the kinematic boundary conditions, but is otherwise arbitrary. The virtual strain field $\delta\boldsymbol{\varepsilon}$ is obtained from the virtual displacement field via Eq. (8.1.1) as

$$\delta\boldsymbol{\varepsilon} = \mathbf{L}_1(\delta\mathbf{u}) \,. \tag{8.1.8}$$

This operator notation for the equations is quite general, in that it is equally applicable to continuum problems as well as to discrete formulations. It is also very convenient for sensitivity calculations. In this section we consider only sensitivities with respect to a stiffness parameter appearing in the material stiffness matrix $\mathbf{D}$. For one or two dimensional problems the parameter can include sizing variables, such as rod cross-sectional areas or plate thicknesses, since these variables are incorporated in $\mathbf{D}$ (as is the beam section moment of inertia in Eq. (8.1.5)).

8.1.1 *The Direct Method*

The direct method for sensitivity calculation is obtained by differentiating the equations defining the response of the structure with respect to p. We then obtain a set of equations for the response sensitivity $\mathbf{u}_{,p}$, $\boldsymbol{\varepsilon}_{,p}$, and $\boldsymbol{\sigma}_{,p}$. The governing equations for the sensitivity fields are shown to be the same as the equations for the response itself, albeit with a different loading terms, that are called pseudo loads. The implication is that if we replace the loading in the original problem by the pseudo loads our structural analysis package will compute the response sensitivities instead of the response. We start by differentiating the strain-displacement relation

$$\boldsymbol{\varepsilon}_{,p} = \mathbf{L}_1(\mathbf{u}_{,p}) \,. \tag{8.1.9}$$

Similarly, differentiating the constitutive equations we obtain

$$\boldsymbol{\sigma}_{,p} = \mathbf{D}\boldsymbol{\varepsilon}_{,p} + \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) \,, \tag{8.1.10}$$

and the differentiated equations of equilibrium are

$$\boldsymbol{\sigma}_{,p} \bullet \delta\boldsymbol{\varepsilon} = 0 \,, \tag{8.1.11}$$

where $\delta\boldsymbol{\varepsilon}$, given by Eq. (8.1.8), is not a function of p because $\delta\mathbf{u}$ is an arbitrary field. Note that all the sensitivity fields have units of the original fields divided by units of p. For example, if p represents the cross-sectional area of a truss member then $\boldsymbol{\sigma}_{,p}$ has units of stress divided by area.

We now compare the differentiated equations, Eq. (8.1.9), (8.1.10), and (8.1.11) to the original governing equations, Eq. (8.1.1), (8.1.4), and (8.1.6). We see that the

sensitivity field $\mathbf{u}_{,p}$, $\boldsymbol{\varepsilon}_{,p}$, $\boldsymbol{\sigma}_{,p}$ can be viewed as the solution of the original structure under a different set of loads called the pseudo loads. These loads do not include any mechanical components, but just an initial strain field $\boldsymbol{\varepsilon}^p$. This initial strain field is obtained by rearranging Eq. (8.1.10)as

$$\boldsymbol{\sigma}_{,p} = \mathbf{D}(\boldsymbol{\varepsilon}_{,p} - \boldsymbol{\varepsilon}^p), \qquad \boldsymbol{\varepsilon}^p = -\mathbf{D}^{-1}\mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i)\,. \tag{8.1.12}$$

For example, for truss members the relation between the generalized stress (member force N) and the strain is

$$N = EA(\epsilon - \epsilon^i)\,. \tag{8.1.13}$$

Differentiating this equation with respect to A we get

$$N_{,A} = EA[\epsilon_{,A} + (\epsilon - \epsilon^i)/A]\,, \tag{8.1.14}$$

so that to implement the direct method we need to apply an initial strain of magnitude $-(\epsilon - \epsilon^i)/A$ instead of the actual loads.

As another example consider the isotropic plane-stress case, where the constitutive equations are

$$\left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \end{array}\right\} = \frac{Eh}{1-\nu^2}\left[\begin{array}{ccc} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{array}\right]\left\{\begin{array}{c} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{array}\right\}. \tag{8.1.15}$$

By differentiating Eq. (8.1.15) with respect to the thickness h we can show that to find the sensitivity with respect to change in thickness we need to apply a pseudo initial strain of $\boldsymbol{\varepsilon}^p = -\boldsymbol{\varepsilon}/h$. To obtain sensitivity with respect to Poisson's ratio ν we note that

$$\mathbf{D}^{-1} = \frac{1}{Eh}\left[\begin{array}{ccc} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{array}\right], \quad \mathbf{D}_{,\nu} = \frac{Eh}{(1-\nu^2)^2}\left[\begin{array}{ccc} 2\nu & 1+\nu^2 & 0 \\ 1+\nu^2 & 2\nu & 0 \\ 0 & 0 & -\frac{(1-\nu)^2}{2} \end{array}\right]. \tag{8.1.16}$$

so that we need to apply a pseudo initial strain of

$$\boldsymbol{\varepsilon}^p = \frac{1}{1-\nu^2}\left\{\begin{array}{c} -\nu\epsilon_x - \epsilon_y \\ -\nu\epsilon_y - \epsilon_x \\ (1-\nu)\gamma_{xy} \end{array}\right\}. \tag{8.1.17}$$

When we analyze the structure using a finite-element model, the pseudo initial strain is converted to a pseudo nodal force $\mathbf{f}^p$ such that

$$\mathbf{D}\boldsymbol{\varepsilon}^p \bullet \delta\boldsymbol{\varepsilon} = \mathbf{f}^p \bullet \delta\mathbf{u}\,. \tag{8.1.18}$$

With other solution techniques the pseudo load is obtained from the initial strain in a different manner. For example, in a three dimensional continuum formulation the pseudo initial strain, $\boldsymbol{\varepsilon}^p$, can be replaced by pseudo body forces with components $f_i = (\mathbf{D}\boldsymbol{\varepsilon}^p)_{ij,j}$ and surface tractions with components $T_i = (\mathbf{D}\boldsymbol{\varepsilon}^p)_{ij}n_j$. Where n_j are the components of the vector normal to the boundary S, and a comma followed by an index j denotes derivative with respect to the coordinate x_j.

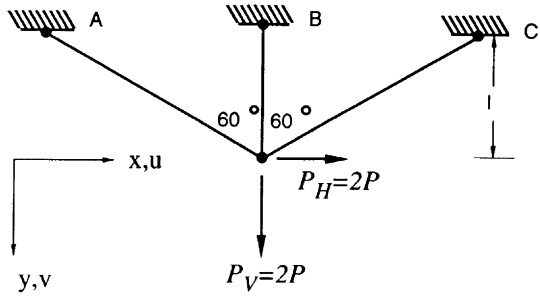


Figure 8.1.1 Three bar truss.

Example 8.1.1

Calculate the derivative of the stress in members A, B and C of the truss in Fig. (8.1.1) with respect to the area of member B. At the nominal configuration all three members have the same area A.

We assume that the areas of members A and C remain the same, A_A, and denote the area of member B as A_B. Due to symmetry, the vertical force contributes only to the vertical displacement, and the horizontal force only to the horizontal displacement. Furthermore, member B does not influence the horizontal displacements. It is easy then to check that the two displacements at the point of load application are given as

$$u = 4P_H l/3EA_A, \qquad v = P_V l/[(A_B + 0.25A_A)E]. \tag{a}$$

The forces in members A, B and C are then calculated to be

$$\begin{aligned} N_A &= 0.57735P_H + \frac{0.25P_V A_A}{A_B + 0.25A_A} = 0.97735P, \\ N_B &= \frac{P_V A_B}{A_B + 0.25A_A} = 1.6P, \\ N_C &= -0.57735P_H + \frac{0.25P_V A_A}{A_B + 0.25A_A} = -0.17735P. \end{aligned} \tag{b}$$

We can calculate the derivatives of these forces with respect to A_B analytically for the purpose of comparing it later with the derivatives we obtain using the direct method.

$$\frac{dN_C}{dA_B} = \frac{dN_A}{dA_B} = \frac{-0.25P_V A_A}{(A_B + 0.25A_A)^2} = -0.32P/A.$$

$$\frac{dN_B}{dA_B} = \frac{0.25P_V A_A}{(A_B + 0.25A_A)^2} = 0.32P/A.$$

For our problem, we need to apply to member B a pseudo initial strain

$$\epsilon^p = -\epsilon_B/A_B = -N_B/EA_B^2 = -1.6P/EA^2,$$

while for other members the pseudo initial strain is zero. Note that as with all sensitivity fields the units of the pseudo initial strain are units of strain divided by units of p (area here). The displacement field generated by this initial strain is obtained by applying to member B a pair of opposite forces, with the force at the bottom joint (having units of force over area) being (see Eq. (8.1.18))

$$f^p \delta v = \int_0^l EA_B \epsilon^p \delta\epsilon dy = EA_B \epsilon^p (\delta v/l) l = -1.6P/A\delta v.$$

We can get the corresponding displacements by setting the horizontal force P_H to zero and replacing the vertical force P_V in Eq. (a) with f^p. These displacements are the derivatives of the original displacements. Thus

$$\frac{du}{dA_B} = 0, \qquad \frac{dv}{dA_B} = \frac{(-1.6P/A)l}{(A_B + 0.25A_A)E} = \frac{-1.28Pl}{EA^2}.$$

The derivative of N_A and N_C can be similarly obtained from Eq. (b)

$$\frac{dN_C}{dA_B} = \frac{dN_A}{dA_B} = \frac{0.25(-1.6P/A)A_A}{A_B + 0.25A_A} = -0.32\frac{P}{A}.$$

However, the internal load in member B due to the pseudo initial strain, which corresponds to the derivative of N_B cannot be obtained in a similar way from Eq. (b) because member B has now initial strain. Instead we use Eq. (8.1.14), which requires the derivative of ϵ_B

$$\frac{d\epsilon_B}{dA_B} = \frac{1}{l}\frac{dv}{dA_B} = \frac{-1.28P}{EA^2},$$

and then

$$\frac{dN_B}{dA_B} = EA_B\left(\frac{d\epsilon_B}{dA_B} - \epsilon^p\right) = \frac{-1.28P}{A} + \frac{1.6P}{A} = 0.32\frac{P}{A}.$$

We note that both derivatives agree with the expressions we obtained by explicit differentiation.

To calculate the derivatives of the stresses from the derivatives of the loads we note that

$$\sigma_A = \frac{N_A}{A_A}, \qquad \sigma_B = \frac{N_B}{A_B}, \qquad \sigma_C = \frac{N_C}{A_A},$$

and therefore

$$\frac{d\sigma_A}{dA_B} = \frac{1}{A_A}\frac{dN_A}{dA_B} = \frac{-0.32P}{A^2},$$
$$\frac{d\sigma_C}{dA_B} = \frac{1}{A_A}\frac{dN_C}{dA_B} = \frac{-0.32P}{A^2},$$

and

$$\frac{d\sigma_B}{dA_B} = \frac{1}{A_B}\frac{dN_B}{dA_B} - \frac{N_B}{A_B^2} = \frac{0.32P}{A^2} - \frac{1.6P}{A^2} = -1.28\frac{P}{A^2}.$$

• • •

8.1.2 *The Adjoint Method*

Often we do not need the derivatives of the entire displacement or stress fields, but only a few quantities such as the derivative of the vertical displacement at a point, or the Von Mises stress at another point. In such cases it may be more economical to use the adjoint method to calculate these derivatives. We therefore consider the adjoint method for calculating derivatives of displacement and stress functionals. Consider first a displacement functional defined by an integral over the structural domain V

$$H = \int h(\mathbf{u}, p) dV \,. \tag{8.1.19}$$

This could also be used to represent the value of a displacement component at a point by employing the Dirac delta function as part of h. The derivative of H with respect to a design parameter p is

$$H_{,p} = \int h_{,p} dV + \mathbf{h}_{,u} \bullet \mathbf{u}_{,p} \,, \tag{8.1.20}$$

where $\mathbf{h}_{,u}$ is a load-like vector field (recall that a bullet denotes scalar product followed by integration over the structure). For example, in a plane-stress case if $h = u^2 + v^2$, then

$$\mathbf{h}_{,u} = \begin{Bmatrix} 2u \\ 2v \end{Bmatrix} . \tag{8.1.21}$$

The calculation of $h_{,p}$ and $\mathbf{h}_{,u}$ is typically easy, and the main difficulty is to obtain the derivative of the displacement field, $\mathbf{u}_{,p}$. We can use the direct method to calculate $\mathbf{u}_{,p}$, but instead, as shown below, we can define an adjoint problem with $\mathbf{h}_{,u}$ as the load, and use it to eliminate $\mathbf{u}_{,p}$. Since we want the derivative of H with the requirement that Eqs. (8.1.1), (8.1.4) and (8.1.6) are satisfied, we multiply these equations by some appropriate Lagrange multipliers (called adjoint variables) and add them to H. The Lagrange multipliers for Eqs. (8.1.1) and (8.1.4) are an adjoint stress field and an adjoint strain field, respectively. Equation (8.1.6) represents the equations of equilibrium written as the work done on a virtual displacement field $\delta\mathbf{u}$, and the corresponding virtual strain field $\delta\boldsymbol{\varepsilon} = \mathbf{L}_1(\delta\mathbf{u})$. Multiplying the equations of equilibrium by a Lagrange multiplier is equivalent to calculating the work done when this Lagrange multiplier is treated as a virtual displacement field. So we replace the $\delta\mathbf{u}$ by the adjoint displacement field. Denoting the adjoint fields by a superscript a we get

$$H^* = H + \boldsymbol{\sigma}^a \bullet \Big(\boldsymbol{\varepsilon} - \mathbf{L}_1(\mathbf{u})\Big) + \boldsymbol{\varepsilon}^a \bullet \Big(\boldsymbol{\sigma} - \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i)\Big) + \mathbf{f} \bullet \mathbf{u}^a - \boldsymbol{\sigma} \bullet \mathbf{L}_1(\mathbf{u}^a) \,. \tag{8.1.22}$$

Because Eqs. (8.1.1), (8.1.4) and (8.1.6) have to be satisfied for all values of p, we have $H^* = H$, and $H^*_{,p} = H_{,p}$. We will now differentiate Eq. (8.1.21), and then define the adjoint fields so as to get rid of the terms involving the (expensive) derivatives of the response. The derivative of Eq. (8.1.21) with respect to p is

$$\begin{aligned} H_{,p} = \int h_{,p} dV + \mathbf{h}_{,u} \bullet \mathbf{u}_{,p} + (\boldsymbol{\sigma}^a - \mathbf{D}\boldsymbol{\varepsilon}^a) \bullet \boldsymbol{\varepsilon}_{,p} - \boldsymbol{\sigma}^a \bullet \mathbf{L}_1(\mathbf{u}_{,p}) \\ -\boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) + \Big(\boldsymbol{\varepsilon}^a - \mathbf{L}_1(\mathbf{u}^a)\Big) \bullet \boldsymbol{\sigma}_{,p} \,. \end{aligned} \tag{8.1.23}$$

We can get rid of the terms involving $\boldsymbol{\sigma}_{,p}$ and $\boldsymbol{\varepsilon}_{,p}$ by requiring the adjoint fields to satisfy the linear strain displacement relationship and Hooke's law

$$\boldsymbol{\varepsilon}^a = \mathbf{L}_1(\mathbf{u}_a)\,, \tag{8.1.24}$$

$$\boldsymbol{\sigma}^a = \mathbf{D}\boldsymbol{\varepsilon}^a\,. \tag{8.1.25}$$

The terms involving $\mathbf{u}_{,p}$ can be removed by requiring the adjoint field to satisfy the equilibrium equations with a body force equal to $\mathbf{h}_{,u}$, so that from the principle of virtual work

$$\boldsymbol{\sigma}^a \bullet \delta\boldsymbol{\varepsilon} = \mathbf{h}_{,u} \bullet \delta\mathbf{u}\,. \tag{8.1.26}$$

Indeed, if we choose $\delta\mathbf{u} = \mathbf{u}_{,p}$ in Eq. (8.1.26) we get the desired elimination of the $\mathbf{u}_{,p}$ terms. Altogether we get

$$H_{,p} = \int h_{,p}dV - \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) \bullet \boldsymbol{\varepsilon}^a\,. \tag{8.1.27}$$

Using Eqs. (8.1.12), and (8.1.25) we can write this as

$$H_{,p} = \int h_{,p}dV + \boldsymbol{\varepsilon}^p \bullet \boldsymbol{\sigma}^a\,. \tag{8.1.28}$$

When we use the finite element method for the analysis we can transform the second term further. To this end we set $\delta\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^a$, $\delta\mathbf{u} = \mathbf{u}^a$ in Eq. (8.1.18) to obtain

$$\mathbf{f}^p \bullet \mathbf{u}^a = \mathbf{D}\boldsymbol{\varepsilon}^p \bullet \boldsymbol{\varepsilon}^a = \boldsymbol{\varepsilon}^p \bullet \boldsymbol{\sigma}^a\,, \tag{8.1.29}$$

so that Eq. (8.1.28) becomes

$$H_{,p} = \int h_{,p}dV + \mathbf{f}^p \bullet \mathbf{u}^a\,. \tag{8.1.30}$$

The treatment of a generalized stress functional is similar. We limit the treatment to the case where there is no initial strain in the structure (that is, mechanical loads are allowed, but no temperature loading, dislocations, etc.) and consider the stress functional

$$G = \int g(\boldsymbol{\sigma}, p)dV \tag{8.1.31}$$

and its derivative

$$G_{,p} = \int g_{,p}dV + \mathbf{g}_{,\sigma} \bullet \boldsymbol{\sigma}_{,p}\,, \tag{8.1.32}$$

where $\mathbf{g}_{,\sigma}$ is a strain-like tensor. Again, to get rid of the expensive derivative of the response, $\boldsymbol{\sigma}_{,p}$, we add the adjoint terms as Lagrange multipliers on Eqs. (8.1.1), (8.1.4) and (8.1.6)

$$G^* = G + \boldsymbol{\sigma}^a \bullet \Big(\boldsymbol{\varepsilon} - \mathbf{L}_1(\mathbf{u})\Big) + \boldsymbol{\varepsilon}^a \bullet (\boldsymbol{\sigma} - \mathbf{D}\boldsymbol{\varepsilon}) + \mathbf{f} \bullet \mathbf{u}^a - \boldsymbol{\sigma} \bullet \mathbf{L}_1(\mathbf{u}^a)\,. \tag{8.1.33}$$

We differentiate Eq. (8.1.33) with respect to p to obtain

$$G_{,p} = G^*_{,p} = \int g_{,p} dV + \mathbf{g}_{,\sigma} \bullet \boldsymbol{\sigma}_{,p} + (\boldsymbol{\sigma}^a - \mathbf{D}\boldsymbol{\varepsilon}^a) \bullet \boldsymbol{\varepsilon}_{,p} - \boldsymbol{\sigma}^a \bullet \mathbf{L}_1(\mathbf{u}_{,p}) \\ + \boldsymbol{\varepsilon}^a \bullet \boldsymbol{\sigma}_{,p} - \boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{,p} \bullet \mathbf{L}_1(\mathbf{u}^a) . \tag{8.1.34}$$

We use Eq. (8.1.10) and rearrange terms to get

$$G_{,p} = \int g_{,p} dV + (\boldsymbol{\sigma}^a + \mathbf{D}\mathbf{g}_{,\sigma} - \mathbf{D}\boldsymbol{\varepsilon}^a) \bullet \boldsymbol{\varepsilon}_{,p} - \boldsymbol{\sigma}^a \bullet \mathbf{L}_1(\mathbf{u}_{,p}) \\ + (\mathbf{g}_{,\sigma} - \boldsymbol{\varepsilon}^a) \bullet \mathbf{D}_{,p}\boldsymbol{\varepsilon} + \left(\boldsymbol{\varepsilon}^a - \mathbf{L}_1(\mathbf{u}^a)\right) \bullet \boldsymbol{\sigma}_{,p} . \tag{8.1.35}$$

From Eq. (8.1.35) we can see that we can eliminate the terms including derivatives of the response by using an adjoint strain-displacement relation in the form of Eq. (8.1.24), and setting Hooke's law for the adjoint field as

$$\boldsymbol{\sigma}^a = \mathbf{D}(\boldsymbol{\varepsilon}^a - \mathbf{g}_{,\sigma}) , \tag{8.1.36}$$

and equilibrium as

$$\boldsymbol{\sigma}^a \bullet \delta\boldsymbol{\varepsilon} = 0 . \tag{8.1.37}$$

That is, in this case the adjoint loading is an initial strain $\mathbf{g}_{,\sigma}$ with no mechanical load. Then

$$G_{,p} = \int g_{,p} dV - \mathbf{D}_{,p}\boldsymbol{\varepsilon} \bullet (\boldsymbol{\varepsilon}^a - \mathbf{g}_{,\sigma}) . \tag{8.1.38}$$

While Eq. (8.1.38) gives us $G_{,p}$ without the need to calculate first the design sensitivity field, its second term involves calculations of stiffness matrix derivatives at the element level, and may require some knowledge of the details of the finite-element analysis. To overcome this problem we note that by using Eq. (8.1.36) we can transform the second term of Eq. (8.1.38) into

$$\mathbf{D}_{,p}\boldsymbol{\varepsilon} \bullet (\boldsymbol{\varepsilon}^a - \mathbf{g}_{,\sigma}) = \mathbf{D}_{,p}\boldsymbol{\varepsilon} \bullet \mathbf{D}^{-1}\boldsymbol{\sigma}^a , \tag{8.1.39}$$

so that using Eq. (8.1.12), which with $\boldsymbol{\varepsilon}^i = 0$ reduces to $\boldsymbol{\varepsilon}^p = -\mathbf{D}^{-1}\mathbf{D}_{,p}$, we can also write $G_{,p}$ as

$$G_{,p} = \int g_{,p} dV + \boldsymbol{\varepsilon}^p \bullet \boldsymbol{\sigma}^a . \tag{8.1.40}$$

In obtaining Eq. (8.1.40) we used the fact that if $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are two stress tensors, then $\boldsymbol{\sigma}_1 \bullet \mathbf{D}^{-1}\boldsymbol{\sigma}_2 = \mathbf{D}^{-1}\boldsymbol{\sigma}_1 \bullet \boldsymbol{\sigma}_2$. As with the displacement functional we can also write $G_{,p}$ in terms of the pseudo load. We use Eq. (8.1.18) with $(\delta\mathbf{u}, \delta\boldsymbol{\varepsilon})$ set to $(\mathbf{u}^a, \boldsymbol{\varepsilon}^a)$ and Eq. (8.1.12) to obtain

$$\mathbf{f}^p \bullet \mathbf{u}^a = \mathbf{D}\boldsymbol{\varepsilon}^p \bullet \boldsymbol{\varepsilon}^a = -\mathbf{D}_{,p}\boldsymbol{\varepsilon} \bullet \boldsymbol{\varepsilon}^a , \tag{8.1.41}$$

and then Eq. (8.1.38) becomes

$$G_{,p} = \int g_{,p} dV + \mathbf{f}^p \bullet \mathbf{u}^a + \mathbf{D}_{,p}\boldsymbol{\varepsilon} \bullet \mathbf{g}_{,\sigma} . \tag{8.1.42}$$

The last term in Eq. (8.1.42) still involves computations with displacements and strains which may not be easy to implement in a general structural analysis code. However, when G is simply the average stress (not generalized stress!) in an element, the first and last terms often cancel. Consider, for example, the average stress in the ith element of a truss. In a truss element the generalized stress is the member force N, so

$$G = \frac{1}{l_i}\int \frac{N}{A} dl_i \qquad \text{and} \qquad \mathbf{g}_{,\sigma} = \frac{1}{Al_i}\,. \tag{8.1.43}$$

When we need the derivative of G with respect to a design variable which does not affect the ith element both the first and third terms in Eq. (8.1.42) are zero. For the derivative of G with respect to the area of the ith element we have from Eq. (8.1.42) (using $\mathbf{D}_{,p} = E$ and $\epsilon = N/AE$)

$$G_{,p} = \frac{1}{l_i}\int (-N/A^2) dl_i + \mathbf{f}^p \bullet \mathbf{u}^a + \int (N/A)(1/Al_i) dl_i = \mathbf{f}^p \bullet \mathbf{u}^a\,. \tag{8.1.44}$$

Note that, as in the discrete case, both the direct and adjoint methods use the pseudo load $\mathbf{f}^p$. In the direct method $\mathbf{f}^p$ is applied to the structure to stand for the pseudo initial strain $\boldsymbol{\varepsilon}^p$ of Eq. (8.1.12), and the response to that load is $\mathbf{u}_{,p}$. In the adjoint method $\mathbf{f}^p$ is used to form a scalar product with the adjoint displacement field $\mathbf{u}^a$, as in Eq. (8.1.42).

Example 8.1.2

We solve Example (8.1.1) again using the adjoint method to obtain the derivatives of the stresses in members A and B with respect to the cross-sectional areas of both members.

Consider first the stress in member B written in terms of the generalized stress (member force) N_B

$$G = \sigma_B = \frac{1}{l_B}\int \frac{N_B}{A_B} dl_B\,.$$

The adjoint load is an initial strain $\mathbf{g}_{,\sigma}$ which is denoted here as $g_{,N}$ because the member force N is the only component of $\boldsymbol{\sigma}$.

$$g_{,N} = \frac{1}{l_B A_B} = \frac{1}{lA}\,.$$

Note that adjoint initial strain is measured in units of 1/(volume) in contrast to the dimensionless physical strains. As a result, all the units of the adjoint field will be the original ones divided by volume. As in Example (8.1.1), the effect of this initial strain is obtained by applying a pair of opposite forces to member B, with the force at the bottom being $EA_B g_{,N} = E/l$. Using Eq. (a) of Example (8.1.1) we get

$$v^a = (E/l)l/[(A_B + 0.25A_A)E] = 0.8/A\,.$$

Following Eq. (8.1.44) we multiply this by the pseudo load f^p of -1.6P/A obtained in Example (8.1.1) to get

$$\frac{d\sigma_B}{dA_B} = G_{,A_B} = -1.28\frac{P}{A^2},$$

which agrees with the result obtained in Example (8.1.1).

Next we calculate the derivative of σ_B with respect to A_A. As in Example (8.1.1) we need to calculate the pseudo load due to a change in A_A. This change affects both members A and C, leading to pseudo initial strains of $-\epsilon_A/A_A$ and $-\epsilon_C/A_A$, respectively. This in turn leads to pseudo loads of $-N_A/A_A$ in the direction of member A and $-N_C/A_A$ in the direction of member C. The components of the pseudo load are

$$P_H^p = -(\frac{N_A}{A_A} - \frac{N_C}{A_A})sin60° = -\frac{P}{A}, \qquad P_V^p = -(\frac{N_A}{A_A} + \frac{N_C}{A_A})cos60° = -0.4\frac{P}{A},$$

where the values of N_A and N_C are substituted from Eq. (b) of Example (8.1.1). Multiplying the adjoint displacement by the pseudo load we obtain

$$\frac{d\sigma_B}{dA_A} = G_{,A_A} = v^a P_V^p = -0.32\frac{P}{A^2},$$

which can be easily checked directly.

Next we calculate the derivatives of σ_A by considering the functional

$$G = \frac{1}{l_A}\int \frac{N_A}{A_A} dl_A .$$

We need to impose an adjoint initial strain of

$$g_{,N} = \frac{1}{l_A A_A} = \frac{1}{2lA}.$$

This is implemented by applying a pair of opposite forces at the two nodes of member A of magnitude $EA_A g_{,N} = E/2l$ collinear with member A. The horizontal and vertical component of the adjoint force at the bottom node are

$$P_H^a = \frac{0.433E}{l}, \qquad P_V^a = \frac{0.25E}{l}.$$

Using Eq.(a) of Example 8.1.1 we get

$$v^a = \frac{(0.25E/l)l}{(A_B + 0.25A_A)E} = \frac{0.2}{A}, \qquad u^a = \frac{4(0.433E/l)l}{3EA_A} = \frac{0.57735}{A}.$$

To get the derivative of the stress with respect to A_B we multiply the adjoint displacements by the pseudo load associated with A_B to obtain

$$\frac{d\sigma_A}{dA_B} = \frac{-1.6P}{A}\frac{0.2}{A} = -0.32\frac{P}{A^2}.$$

Similarly, to obtain the derivative with respect to A_A we multiply the adjoint displacements by the pseudo loads associated with A_A

$$\frac{d\sigma_A}{dA_A} = \frac{-P}{A}\frac{0.57735}{A} - \frac{0.4P}{A}\frac{0.2}{A} = -0.65735\frac{P}{A^2}.$$

This last result can be checked directly by using the expression for N_A in Eq. (b) of Example (8.1.1)• • •

8.1.3 *Implementation Notes*

In general, the direct method is easier to implement than the adjoint method, particularly if the implementation is outside the structural analysis program. The direct method will require a postprocessor that calculates the value of the pseudo initial strain from the values of the actual strains based on Eq. (8.1.12). The derivative of the material stiffness matrix $\mathbf{D}_{,p}$ which needs to be evaluated in this postprocessor requires knowledge of the form of Hooke's law used in the analysis program, but not of any finite element implementation. Then the values of the pseudo strain can be used as initial strain input to the same structural analysis package. The output of the package will be the sensitivity field, instead of the response. If the structural analysis package does not have the capability of accepting initial strain input it is often possible to use a combination of a temperature field and anisotropic coefficients of thermal expansion to get the required initial strains.

The implementation of the adjoint method with the displacement functional H of Eq. (8.1.19) is very simple. We need to perform the structural analysis with the actual loading replaced by the adjoint load $\mathbf{h}_{,u}$. Note that the adjoint load $\mathbf{h}_{,u}$ is similar to the adjoint load used in the discrete case; that was the derivative of the constraint with respect to the displacement vector (see Section 7.2.1). Its units are the units of h divided by the units of $\mathbf{u}$, and in general these will not be the units of force per unit volume. As a result, the units of the adjoint field will not be the normal units associated with displacement, strain and stress fields. We also need to add a postprocessor that will perform the calculations indicated in Eqs.(8.1.28) or Eqs. (8.1.30). The former involves the pseudo initial strain of Eq. (8.1.12) and the latter the pseudo load associated with this strain, Eq. (8.1.18). Equation (8.1.30) is typically easier to implement than Eq. (8.1.28) because it requires only a scalar product of the pseudo-load with the adjoint displacement field. It is completely analogous to Eq. (7.2.8), except that the pseudo-load is not obtained in terms of derivatives of stiffness matrices, and so does not require intimate knowledge of the finite-element package.

The implementation of the adjoint method for a stress functional, Eq. (8.1.31) is more complicated. First we need to implement the calculation of an initial strain field $\mathbf{g}_{,\sigma}$, which is usually fairly simple. We then need to implement Eq. (8.1.42), which requires calculations at the element level for a finite element program. The discussion following Eq. (8.1.42) shows that when the stress functional is just the stress itself this difficulty can be in many cases bypassed.

8.2 Nonlinear Static Analysis and Limit Loads

8.2.1 *Static Analysis*

In this section we generalize the results of the previous section to the case of geometric nonlinearity. We consider only the case where the nonlinearity is adequately represented by replacing Eq. (8.1.1) by

$$\boldsymbol{\varepsilon} = \mathbf{L}_1(\mathbf{u}) + \frac{1}{2}\mathbf{L}_2(\mathbf{u}), \tag{8.2.1}$$

where $\mathbf{L}_2$ is a second order homogeneous operator. For example, for the nonlinear deformation of a beam under lateral and axial loads, the generalized strain has one component of axial strain ϵ_x and one component of curvature κ, and Eq.(8.2.1) is written as

$$\begin{Bmatrix} \epsilon_x \\ \kappa \end{Bmatrix} = \begin{Bmatrix} u_{,x} \\ w_{,xx} \end{Bmatrix} + \frac{1}{2}\begin{Bmatrix} w_{,x}^2 \\ 0 \end{Bmatrix}. \tag{8.2.2}$$

The variation of the strain is specified in terms of displacement variation as

$$\delta\boldsymbol{\varepsilon} = \mathbf{L}_1(\delta\mathbf{u}) + \mathbf{L}_{11}(\mathbf{u}, \delta\mathbf{u}), \tag{8.2.3}$$

where $\mathbf{L}_{11}$ is a symmetric bilinear operator, i.e. $\mathbf{L}_{11}(\mathbf{u}, \mathbf{v}) = \mathbf{L}_{11}(\mathbf{v}, \mathbf{u})$, defined by

$$\mathbf{L}_2(\mathbf{u} + \mathbf{v}) = \mathbf{L}_2(\mathbf{u}) + \mathbf{L}_2(\mathbf{v}) + 2\mathbf{L}_{11}(\mathbf{u}, \mathbf{v}). \tag{8.2.4}$$

In particular Eq. (8.2.4) yields

$$\mathbf{L}_{11}(\mathbf{u}, \mathbf{u}) = \mathbf{L}_2(\mathbf{u}). \tag{8.2.5}$$

In solving nonlinear analysis problems it is customary to increase the load gradually from zero to its final value. To accommodate this practice we assume that the load $\mathbf{f}$ and the initial strain $\boldsymbol{\varepsilon}^i$ depend on a load amplitude parameter μ, that is

$$\mathbf{f} = \mathbf{f}(\mu), \quad \text{and} \quad \boldsymbol{\varepsilon}^i = \boldsymbol{\varepsilon}^i(\mu). \tag{8.2.6}$$

The structural response can then be obtained by solving Eqs. (8.2.1), (8.1.4) and (8.1.6) as a function of the load parameter μ.

Unfortunately, in the nonlinear regime the response is not always a single-valued function of the load parameter μ. Figure (8.2.1) shows a typical load displacement curve for two values of the stiffness parameter p. At load levels near the maximum (limit load), there are two solutions for each value of μ. Structural analysis packages that solve for nonlinear response often use more general parameters for tracing the response curve. A typical parameter is the arc length in the $(\mathbf{u}, \mu)$ space. We call any parameter that is used to trace an equilibrium path (that is, a path of solutions to Eqs. (8.2.1), (8.1.4) and (8.6)) a path parameter.

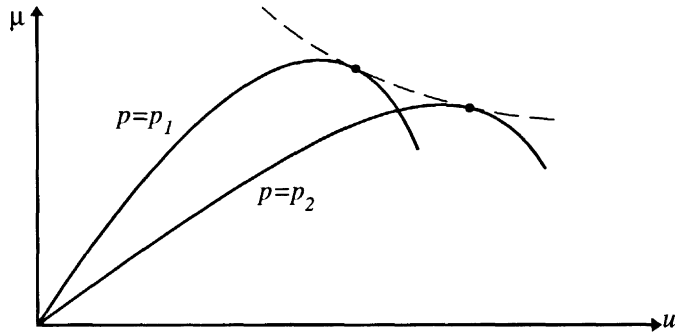


Figure 8.2.1 Load displacement diagram.

In considering changes in the structure we want to have the freedom to change both the load parameter and the stiffness simultaneously. Such simultaneous changes will be needed in the calculation of derivatives of limit loads. Figure (8.2.1) shows one example of the curve traced by such a more general path parameter. The dashed curve in the Figure connects all the limit points for configurations with different stiffnesses. We denote derivatives with respect to general path parameters by a dot. Differentiating Eqs. (8.2.1), (8.1.4) and (8.1.6) with respect to such a parameter we get

$$\dot{\varepsilon} = \mathbf{L}_1(\dot{\mathbf{u}}) + \mathbf{L}_{11}(\mathbf{u}, \dot{\mathbf{u}}), \tag{8.2.7}$$

$$\dot{\sigma} = \dot{\mathbf{D}}(\varepsilon - \varepsilon^i) + \mathbf{D}(\dot{\varepsilon} - \dot{\mu}\varepsilon^{i'}), \tag{8.2.8}$$

$$\dot{\sigma} \bullet \delta\varepsilon + \sigma \bullet \mathbf{L}_{11}(\dot{\mathbf{u}}, \delta\mathbf{u}) = \dot{\mu}\mathbf{f}' \bullet \delta\mathbf{u}, \tag{8.2.9}$$

where a prime denotes differentiation with respect to μ. The second term in Eq. (8.2.9) is due to the dependence of $\delta\varepsilon$ on $\mathbf{u}$, Eq. (8.2.3).

Most solution algorithms used for nonlinear analysis are based on gradually incrementing the load parameter μ. Quite often the solution requires the calculation of the sensitivity of the response with respect to μ. Specializing Eqs. (8.2.7)–(8.2.9) to this case, and denoting load sensitivities by primes, we obtain

$$\varepsilon' = \mathbf{L}_1(\mathbf{u}') + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}'), \tag{8.2.10}$$

$$\sigma' = \mathbf{D}(\varepsilon' - \varepsilon^{i'}), \tag{8.2.11}$$

$$\sigma' \bullet \delta\varepsilon + \sigma \bullet \mathbf{L}_{11}(\mathbf{u}', \delta\mathbf{u}) = \mathbf{f}' \bullet \delta\mathbf{u}, \tag{8.2.12}$$

where $\mathbf{D}$ is assumed to be independent of the load. We will refer to the calculations required for solving Eqs. (8.2.10)-(8.2.12) as the load sensitivity module. These equations for the derivatives with respect to the load parameter are quite similar to the equations governing the sensitivity to a stiffness parameter obtained by specializing Eqs. (8.2.7)–(8.2.9) to the case of a stiffness parameter p:

$$\varepsilon_{,p} = \mathbf{L}_1(\mathbf{u}_{,p}) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_{,p}), \tag{8.2.13}$$

$$\boldsymbol{\sigma}_{,p} = \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) + \mathbf{D}\boldsymbol{\varepsilon}_{,p} = \mathbf{D}(\boldsymbol{\varepsilon}_{,p} - \boldsymbol{\varepsilon}^p)\,, \tag{8.2.14}$$

$$\boldsymbol{\sigma}_{,p} \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \delta\mathbf{u}) = 0\,. \tag{8.2.15}$$

Comparing the two sets of equations we note that the load term in Eq. (8.2.12) is missing from Eq. (8.2.15), and that the constitutive relation Eq. (8.2.14) includes a different initial strain which is equal to $\boldsymbol{\varepsilon}^p$ defined by Eq. (8.1.12). Consequently, in terms of implementing the calculation of design sensitivity in a structural analysis package, we use the load sensitivity module with the actual load and initial strain replaced by the pseudo initial strain $\boldsymbol{\varepsilon}^p$ with zero mechanical load.

In terms of a finite element analysis, the load sensitivity equations are governed by the tangent stiffness matrix. So the only difference between the linear and nonlinear sensitivity calculation is that the pseudo initial strain is applied to the "tangent" structure instead of the original structure. Finally, let us note that both the load sensitivity equations and the design sensitivity equations are linear, even though the analysis problem is nonlinear. This is a general property of sensitivity analysis of nonlinear problems.

It can be shown that the effect of nonlinearity on the adjoint method is similar to its effect on the direct method. That is, in the case of a displacement functional H of Eq. (8.1.19) the adjoint structure satisfies

$$\boldsymbol{\varepsilon}^a = \mathbf{L}_1(\mathbf{u}^a) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}^a)\,, \tag{8.2.16}$$

$$\boldsymbol{\sigma}^a = \mathbf{D}\boldsymbol{\varepsilon}^a\,, \tag{8.2.17}$$

$$\boldsymbol{\sigma}^a \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}^a, \delta\mathbf{u}) = \mathbf{h}_{,u} \bullet \delta\mathbf{u}\,. \tag{8.2.18}$$

The adjoint structure is therefore the tangent structure with $\mathbf{h}_{,u}$ as the applied load [see also [3]). To implement the calculation of the adjoint field in a structural analysis package, we need only to replace the actual load by $\mathbf{h}_{,u}$ in the load sensitivity module. It can be shown (Exercise 5) that Eq. (8.1.28) is still applicable. Similarly, in the case of a stress functional G of Eq. (8.1.31) we apply an initial strain $\mathbf{g}_{,\sigma}$ to the tangent structure, and we can still obtain Eq. (8.1.42).

Example 8.2.1

The beam shown in Figure (8.2.2) has a cross sectional area A_0, a moment of inertia $I = 0.001A_0L^2$, and is subject to a constant applied temperature, T (measured from the stress-free temperature), and a variable transverse load, μP. The applied temperature T is selected so that the resulting axial load is close to the buckling load limit, that is, $EA_0\epsilon^i = EA_0\alpha T = 7.5EI_0/L^2$, where α is the coefficient of thermal expansion, and the applied load is $P = 1.2 \times 10^{-4}EA_0$. We want to calculate the derivative of the displacement under the load, w_m, with respect to the cross sectional area A (assuming that P and I remain constant).

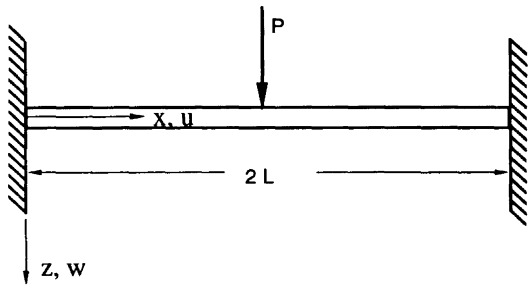


Figure 8.2.2 Beam subject to initial strain and normal load.

For a beam under combined axial and bending actions the generalized strain tensor has two components ϵ_x and κ, and the generalized stress tensor includes the axial load N and the bending moment M. The nonlinear strain-displacement relation for the beam is given by Eq. (8.2.2), and Hooke's law is

$$N = EA(\epsilon_x - \epsilon^i), \qquad M = EI\kappa\,,$$

where $\epsilon^i = \alpha T$. The virtual work equation is

$$\int_0^{2L} (N\delta\epsilon_x + M\delta\kappa)dx = \mu P\delta w_m\,,$$

where

$$\delta\epsilon_x = \delta u_{,x} + w_{,x}\delta w_{,x}, \qquad \delta\kappa = \delta w_{,xx}\,.$$

First we solve the analysis problem in closed form based on a simple finite-element model. Because of the symmetry we need analyze only the left half of the beam, using half of the force, and symmetry conditions of $u = 0$, and $w_{,x} = 0$ in the middle. We approximate the left half of the beam by a single beam finite element with linear variation of u and cubic variation of w. Using the boundary conditions, and the finite-element shape functions we have

$$u = 0, \qquad (w/L) = \bar{w}(3\bar{x}^2 - 2\bar{x}^3), \qquad \text{where } \bar{x} = x/L, \qquad \bar{w} = w_m/L\,.$$

The expressions for the strains and generalized stresses are

$$\epsilon_x = 18\bar{w}^2(\bar{x} - \bar{x}^2)^2, \qquad \kappa = (6\bar{w}/L)(1 - 2\bar{x})\,,$$

$$\delta\epsilon_x = 36\bar{w}\delta\bar{w}(\bar{x} - \bar{x}^2)^2, \qquad \delta\kappa = (6\delta\bar{w}/L)(1 - 2\bar{x})\,,$$

$$N = 18EA\bar{w}^2(\bar{x} - \bar{x}^2)^2 - EA\epsilon^i, \qquad M = (6EI\bar{w}/L)(1 - 2\bar{x})\,.$$

Integrating the virtual work equation over the element (with a load $0.P$ at the end) we obtain

$$1.02857EA\bar{w}^3 + 12(EI/L^2)\bar{w} - 1.2EA\epsilon^i\bar{w} = 0.5\mu P\,. \qquad (a)$$

Dividing Eq. (a) by EA and using the relations between the values of I, ϵ^i and P for $A = A_0$ we get

$$1.02857\bar{w}^3 + 0.003\bar{w} = 6 \times 10^{-5}\mu\,.$$

For $\mu = 1$ we get $\bar{w} = 0.01800$.

Before applying the direct and adjoint methods to calculate $\bar{w}_{,A}$, we consider the tangent state of derivatives with respect to μ. Equations (8.2.10) - (8.2.12) become

$$\epsilon'_x = u'_{,x} + w_{,x}w'_{,x} = 36\bar{w}\bar{w}'(\bar{x} - \bar{x}^2)^2, \qquad \kappa' = w'_{,xx} = 6\bar{w}'(1 - 2\bar{x})/L\,,$$

$$N' = EA\epsilon'_x, \qquad M' = EI\kappa'\,,$$

$$\int_0^L (N'\delta\epsilon_x + M'\delta\kappa + Nw'_{,x}\delta w_{,x})dx = 0.5P\delta w_m\,.$$

This last equation can be integrated to yield

$$3.08571EA\bar{w}^2\bar{w}' + 12(EI/L^2)\bar{w}' - 1.2EA\epsilon^i\bar{w}' = 0.5P\,. \qquad (b)$$

This equation can be verified by differentiating Eq. (a) with respect to μ.

Direct method: Equations (8.2.13)- (8.2.15) become (remember that I is constant)

$$\epsilon_{x,p} = u_{,xp} + w_{,x}w_{,xp}, \qquad \kappa_{,p} = w_{,xxp}\,,$$

$$N_{,p} = EA[\epsilon_{x,p} + (\epsilon_x - \epsilon^i)/A], \qquad M_{,p} = EI\kappa_{,p}\,,$$

$$\int_0^L (N_{,p}\delta\epsilon_x + M_{,p}\delta\kappa + Nw_{,xp}\delta w_{,x})dx = 0\,.$$

We note that the sensitivity equations are identical to the tangent state equations except that instead of the load P we have the pseudo initial strain $\epsilon^p = -(\epsilon_x - \epsilon^i)/A$. Using Eq. (8.1.18), we find that the initial strain gives rise to a pseudo load defined by

$$\begin{aligned} P^p\delta w_m &= -\int_0^L E(\epsilon_x - \epsilon^i)\delta\epsilon_x dx = -E\int_0^L [18\bar{w}^2(\bar{x} - \bar{x}^2)^2 - \epsilon^i]36\bar{w}\delta\bar{w}(\bar{x} - \bar{x}^2)^2 dx \\ &= -1.02857E\bar{w}^3\delta w_m + 1.2E\epsilon^i\bar{w}\delta w_m\,. \end{aligned} \qquad (c)$$

The design sensitivity equation is obtained from the load sensitivity equation, Eq. (b), by replacing the actual load $(0.5P)$ with the pseudo load, P^p and replacing $\bar{w}'$ with $\bar{w}_{,A}$, so that the equation for $\bar{w}_{,A}$ is

$$3.08571EA\bar{w}^2\bar{w}_{,A} + 12(EI/L^2)\bar{w}_{,A} - 1.2EA\epsilon^i\bar{w}_{,A} = -1.02857E\bar{w}^3 + 1.2E\epsilon^i\bar{w}\,. \qquad (d)$$

This result can be verified by differentiating Eq. (a) with respect to A. Solving for $\bar{w}_{,A}$ we obtain

$$\bar{w}_{,A} = \frac{-1.02857\bar{w}^3 + 0.096\bar{w}}{(3.08571\bar{w}^2 + 0.024)A} = \frac{0.6888}{A}\,.$$

Adjoint method: To use the adjoint method we define $\bar{w}$ as

$$\bar{w} = H = \int_0^{2L} (w/L)\delta(x - L)dx\,,$$

where $\delta(x - L)$ is the Dirac delta function. For this case $\mathbf{h}_{,u}$ is a vertical unit concentrated load of magnitude $1/L$ in the middle of the beam. Since the adjoint field is obtained from the load sensitivity module by replacing the actual load with $\mathbf{h}_{,u}$, the equation for the adjoint state w^a can be obtained by replacing $\bar{w}'$ by $\bar{w}^a$ and $0.5P$ by $1/L$ in Eq. (b)

$$\bar{w}^a = [3.08571EA\bar{w}^2 + 12(EI/L^2) - 1.2EA\epsilon^i]^{-1}/L\,.$$

Then we obtain $\bar{w}_{,A}$ from Eq.(8.1.30) as

$$\bar{w}_{,A} = H_{,A} = P^p w(l/2) = LP^p \bar{w}^a$$

which is identical to the result obtained from the direct method. • • •

8.2.2 *Limit Loads*

Next we consider the calculation of a limit load; here the load sensitivity equations, Eqs. (8.2.10)–(8.2.12), become singular. To circumvent the problem associated with this singularity it is customary to define the response path in term of a parameter other than the load (e.g., a displacement component or an arc length parameter). We specialize Eqs. (8.2.7)–(8.2.9) to that case, where the parameter controls the response and μ, but not the stiffness (that is $\dot{\mathbf{D}} = 0$). At the limit point, $\dot{\mu} = 0$, and we denote the derivative of the response with respect to the path parameter by a subscript 1. That is Eqs. (8.2.7)–(8.2.9) become

$$\boldsymbol{\varepsilon}_1 = \mathbf{L}_1(\mathbf{u}_1) + \mathbf{L}_{11}(\mathbf{u}^*, \mathbf{u}_1)\,, \tag{8.2.19}$$

$$\boldsymbol{\sigma}_1 = \mathbf{D}\boldsymbol{\varepsilon}_1\,, \tag{8.2.20}$$

$$\boldsymbol{\sigma}_1 \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma}^* \bullet \mathbf{L}_{11}(\mathbf{u}_1, \delta\mathbf{u}) = 0\,, \tag{8.2.21}$$

where an asterisk denotes the response at the limit point. Note that Eqs. (8.2.19)–(8.2.21) are similar to the homogenous part of equations for the load sensitivities, Eqs. (8.2.10)–(8.2.12). The fact that the homogenous equations have a nontrivial solution indicates that the load sensitivity equations are singular (as expected at a limit point). The singularity can occur not only at a limit point, but also at a bifurcation point; here the solution of Eqs. (8.2.1)–(8.2.12) is not unique. At a bifurcation point we have Eqs. (8.2.19)–(8.2.21) even though $\dot{\mu} \neq 0$. Whether a limit load or a bifurcation buckling, we call $\mathbf{u}_1$ the buckling mode.

To calculate the derivative of the limit load μ^* with respect to a stiffness parameter p we need to specialize Eqs. (8.2.7)–(8.2.9) so that we can change the stiffness and the load simultaneously, but have the load remain at its limit value as we change

the stiffness. This path is denoted by ν and shown as a dashed line in Fig. (8.2.1). Along that path we have

$$p = \nu, \qquad \mu = \mu^*(p), \qquad \mathbf{u} = \mathbf{u}^*(p)\,. \tag{8.2.22}$$

To denote the simultaneous change of p and the load we use both the p subscript for derivative and the asterisk for critical load, so that Eqs. (8.2.7)–(8.2.9) become

$$\boldsymbol{\varepsilon}^*_{,p} = \mathbf{L}_1(\mathbf{u}^*_{,p}) + \mathbf{L}_{11}(\mathbf{u}^*, \mathbf{u}^*_{,p})\,, \tag{8.2.23}$$

$$\boldsymbol{\sigma}^*_{,p} = \mathbf{D}_{,p}(\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon}^i) + \mathbf{D}(\boldsymbol{\varepsilon}^*_{,p} - \mu^*_{,p}\boldsymbol{\varepsilon}^{i'})\,, \tag{8.2.24}$$

$$\boldsymbol{\sigma}^*_{,p} \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma}^* \bullet \mathbf{L}_{11}(\mathbf{u}^*_{,p}, \delta\mathbf{u}) = \mu^*_{,p}\mathbf{f}' \bullet \delta\mathbf{u}\,. \tag{8.2.25}$$

We can now get an expression for the derivative of the limit load by substituting $\delta\mathbf{u} = \mathbf{u}_1$ into Eq. (8.2.25)

$$\mu^*_{,p} = \frac{\boldsymbol{\sigma}^*_{,p} \bullet \boldsymbol{\varepsilon}_1 + \boldsymbol{\sigma}^* \bullet \mathbf{L}_{11}(\mathbf{u}^*_{,p}, \mathbf{u}_1)}{\mathbf{f}' \bullet \mathbf{u}_1}\,. \tag{8.2.26}$$

This equation requires the derivatives of the prebuckling response. We can eliminate these derivatives without using an adjoint field by noting the similarity of the numerator to what we get by substituting $\delta\mathbf{u} = \mathbf{u}^*_{,p}$ into Eq. (8.2.21)

$$\boldsymbol{\sigma}_1 \bullet \boldsymbol{\varepsilon}^*_{,p} + \boldsymbol{\sigma}^* \bullet \mathbf{L}_{11}(\mathbf{u}_1, \mathbf{u}^*_{,p}) = 0\,. \tag{8.2.27}$$

To make Eqs. (8.2.27) more similar to the numerator of Eq. (8.2.26) we use Eqs. (8.2.21) and (8.2.24) to rewrite Eq. (8.2.27) as

$$\boldsymbol{\sigma}^*_{,p} \bullet \boldsymbol{\varepsilon}_1 - \mathbf{D}_{,p}(\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon}^i) \bullet \boldsymbol{\varepsilon}_1 + \mu^*_{,p}\mathbf{D}\boldsymbol{\varepsilon}^{i'} \bullet \boldsymbol{\varepsilon}_1 + \boldsymbol{\sigma}^* \bullet \mathbf{L}_{11}(\mathbf{u}_1, \mathbf{u}^*_{,p}) = 0\,. \tag{8.2.28}$$

Finally, combining Eqs. (8.2.26) and (8.2.28) we get a form of the derivative of the limit load with respect to a stiffness parameter

$$\mu^*_{,p} = \frac{\mathbf{D}_{,p}(\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon}^i) \bullet \boldsymbol{\varepsilon}_1}{\mathbf{f}' \bullet \mathbf{u}_1 + \mathbf{D}\boldsymbol{\varepsilon}^{i'} \bullet \boldsymbol{\varepsilon}_1} \tag{8.2.29}$$

that does not require derivatives of prebuckling response. This expression can be simplified further for the case of finite-element calculation. Using Eqs. (8.1.12) and (8.1.18) we get

$$\mu^*_{,p} = \frac{-\mathbf{f}^{p*} \bullet \mathbf{u}_1}{(\mathbf{f}' + \mathbf{f}^{i'}) \bullet \mathbf{u}_1}\,, \tag{8.2.30}$$

where $\mathbf{f}^{p*}$ is the pseudo load of Eq. (8.1.18) evaluated at the limit point, and $\mathbf{f}^{i'}$ is the equivalent nodal load due to the initial strain $\boldsymbol{\varepsilon}^{i'}$.

The above calculation appears to be applicable also to bifurcation buckling. However, for bifurcation buckling $\dot{\mu}$ in Eq. (8.2.9) is not zero. The consistency condition for this equation is for the right-hand side to be orthogonal to the nonzero solution of the homogeneous problem $\mathbf{u}_1$. That is, for the bifurcation problem $(\mathbf{f}' + \mathbf{f}^{i'}) \bullet \mathbf{u}_1 = 0$, and we cannot use Eqs. (8.2.29) and (8.2.30). The sensitivity of bifurcation buckling loads is discussed in the next section.

Example 8.2.2

The two-bar truss shown in Figure 8.2.3 is subject to a constant load P and variable negative applied temperature $-\mu T$. As the truss is cooled the displacement h under the load will increase until a limit point is reached and the truss collapses. We want to calculate the derivative of the limit load factor μ^* with respect to the cross-sectional area A for $A = A_0$. The other parameters of the problem are $P/EA_0 = 0.001$, $\alpha T = 0.01$, and $\theta = 10°$, where E is Young's modulus, and α is the coefficient of thermal expansion.

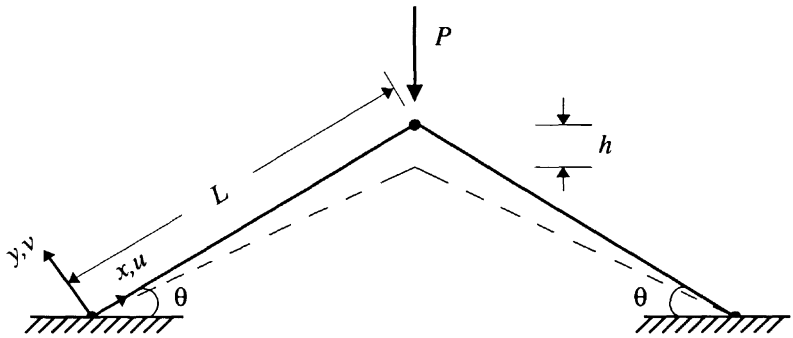


Figure 8.2.3 Two-bar truss under combined mechanical and thermal loading

Because of symmetry we need analyze only one half of the truss, applying to it one half of the mechanical load. We select a coordinate x that runs along the truss member. The strain-displacement relation, Hooke's law, and the virtual work equation are given as

$$\epsilon = u_{,x} + 0.5(u_{,x}^2 + v_{,x}^2), \qquad N = EA(\epsilon + \mu\alpha T), \qquad \int_0^L N\delta\epsilon dx = 0.5P\delta h\,,$$

where

$$\delta\epsilon = \delta u_{,x} + u_{,x}\delta u_{,x} + v_{,x}\delta v_{,x}$$

Since we are dealing with a truss, we can assume linear variation of u and v as a function of x and get

$$u_{,x} = -\bar{h}sin\theta, \quad v_{,x} = -\bar{h}cos\theta, \quad \text{where} \quad \bar{h} = h/L\,,$$

so that

$$\epsilon = -\bar{h}sin\theta + 0.5\bar{h}^2, \qquad \delta\epsilon = -\delta\bar{h}sin\theta + \bar{h}\delta\bar{h}\,,$$

and

$$N = EA(-\bar{h}sin\theta + 0.5\bar{h}^2 + \mu\alpha T)\,.$$

Substituting into the virtual work equation we obtain

$$EA(-\bar{h}sin\theta + 0.5\bar{h}^2 + \mu\alpha T)(-sin\theta + \bar{h})\delta\bar{h} = 0.5P\delta\bar{h}\,.$$

Dividing by $0.5EA$, rearranging the equation, and using the data for the problem gives us

$$\bar{h}^3 - 0.5209\bar{h}^2 + 0.02(3.015 + \mu)\bar{h} = 0.001A_0/A + 0.003473\mu\,. \qquad (a)$$

Equation (a) can be used to trace the response of the truss as the temperature is increased. For a given load parameter μ this requires the solution of a cubic equation. However, it is possible instead to gradually increase $\bar{h}$ and calculate the resultant μ. Tracing the curve we find that the limit load factor is $\mu^* = 0.56274$ corresponding to a displacement $\bar{h} = 0.09424$. Since this problem has only one degree of freedom, the buckling mode has only one component, $\bar{h}$, and we can take it to have a unit value.

To calculate the sensitivity of μ^* using Eq. (8.2.30) we also need $\epsilon - \epsilon^i$ at the limit point. Using the expression for the strain in terms of $\bar{h}$ we get

$$\epsilon^* - \epsilon^{i*} = -\bar{h}^* sin\theta + 0.5(\bar{h}^*)^2 + \mu^*\alpha T = -0.006297\,.$$

The pseudo initial strain for a truss element is $-(\epsilon - \epsilon^i)/A$ (see Eq. (8.1.14)), so that the magnitude of the pseudo load is

$$f^{p*} = -E(\epsilon^* - \epsilon^{i*}) = 0.006297E\,.$$

The pseudo load consists of two forces collinear with the truss element and acting at its two ends. We also need to calculate the pseudo load associated with $\epsilon^{i'} = -\alpha T$. The magnitude of this force is

$$f^{i'} = -EA\alpha T = -0.01EA\,.$$

This force is also collinear with the element. We now use Eq. (8.2.30), noting that for our case $\mathbf{f}' = 0$

$$\mu^*_{,A} = \frac{-0.006297Ecos(90^0 + \theta)}{-0.01EAcos(90^0 + \theta)} = \frac{0.6297}{A_0}\,.$$

We check this result by finite differences. We increase the area by one percent, and substitute $A = 1.01A_0$ into Eq. (a). Solving again we get $\mu^* = 0.56899$ so that the finite-difference approximation to the derivative is

$$\mu^*_{,A} \approx \frac{0.56899 - 0.56274}{0.01A_0} = \frac{0.625}{A_0}\,,$$

which is in reasonable agreement with the analytical derivative.• • •

8.2.3 *Implementation Notes*

A structural analysis package for nonlinear analysis will typically have facilities for generating the derivatives of the applied loads with respect to the loading parameter μ, and for solving the tangent equations of equilibrium at any value of that load. For the sensitivity of static response using the direct method only the second is needed. The procedure is identical to that used in the linear case (see Section 8.1.3). The actual load is replaced by the initial strains associated with the stiffness change (Eq. (8.1.12), and the tangent equations of equilibrium are solved by the structural analysis package. The output of the package will then be the sensitivity to the stiffness variable.

The adjoint method is similar to that used in the linear case. The same adjoint load is used, but it is applied to the tangent system. Equations. (8.1.28) and (8.1.42) are still applicable. However, for nonlinear analysis there is even less of a reason to use the adjoint method than in the linear case. In nonlinear analysis the cost of the analysis is much larger than the cost of sensitivity calculations (which are always linear). Therefore, even when the number of response functionals to be differentiated is much smaller than the number of design variables, the direct method is still a reasonable choice.

For sensitivity of limit loads, Eq. (8.2.30) is easy to implement. It requires calculation of the pseudo load associated with the stiffness change, and the computation of two scalar products: of the pseudo load and the actual load (including both mechanical and initial strain components) with the buckling mode.

8.3 Vibration and Buckling

We first consider small free harmonic vibrations with frequency ω superimposed on the nonlinear equilibrium state $(\mathbf{u}(\mu), \boldsymbol{\varepsilon}(\mu), \boldsymbol{\sigma}(\mu))$ associated with load parameter μ. We denote the vibration amplitude fields by $\mathbf{u}_1$, $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\sigma}_1$. These vibration amplitude fields can be viewed as small perturbations off the nonlinear equilibrium state. Therefore, the equations satisfied by these fields are obtained by adding a small perturbation to the nonlinear field equations, Eqs. (8.2.1), (8.1.4) and (8.1.6) and replacing the body force $\mathbf{f}$ with a D'Alembert inertia force. Assuming that there is no initial strain we obtain

$$\boldsymbol{\varepsilon}_1 = \mathbf{L}_1(\mathbf{u}_1) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_1)\,, \tag{8.3.1}$$

$$\boldsymbol{\sigma}_1 = \mathbf{D}\boldsymbol{\varepsilon}_1\,, \tag{8.3.2}$$

$$\boldsymbol{\sigma}_1 \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_1, \delta\mathbf{u}) = \omega^2 \mathbf{M}\mathbf{u}_1 \bullet \delta\mathbf{u}\,, \tag{8.3.3}$$

where $\mathbf{M}$ denotes the mass tensor and $\delta\boldsymbol{\varepsilon}$ is given by Eq. (8.2.3). Note that these equations are identical to the load sensitivity equations, Eqs. (8.2.10)–(8.2.12) except that $\mathbf{f}'$ is replaced by the inertia load. Setting $\delta\mathbf{u} = \mathbf{u}_1$ in Eq. (8.3.3) we obtain the Rayleigh quotient for the vibration frequency

$$\omega^2 = \frac{\boldsymbol{\sigma}_1 \bullet \boldsymbol{\varepsilon}_1 + \boldsymbol{\sigma} \bullet \mathbf{L}_2(\mathbf{u}_1)}{\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1}\,. \tag{8.3.4}$$

Under static loading the structure buckles at a load μ^* corresponding to a pre-buckling state $\mathbf{u}^* = \mathbf{u}(\mu^*)$, $\boldsymbol{\varepsilon}^* = \boldsymbol{\varepsilon}(\mu^*)$, $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}(\mu^*)$. The buckling load corresponds to a zero vibration frequency. Therefore the bucking mode $\mathbf{u}_1$, $\boldsymbol{\varepsilon}_1$, $\boldsymbol{\sigma}_1$ satisfies Eqs. (8.3.1), (8.3.2), and (8.3.3) with $\omega = 0$ and $\mathbf{u} = \mathbf{u}^*$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}^*$.

8.3.1 *The Direct Method*

To calculate the derivative of the frequency with respect to a stiffness parameter p we start by differentiating Eqs. (8.3.1), (8.3.2), and (8.3.3) with respect to p, then set $\delta\mathbf{u}$ equal to the mode shape $\mathbf{u}_1$, and use Eq. (8.2.5) to obtain

$$\boldsymbol{\varepsilon}_{1,p} = \mathbf{L}_1(\mathbf{u}_{1,p}) + \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_{1,p}), \tag{8.3.5}$$

$$\boldsymbol{\sigma}_{1,p} = \mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 + \mathbf{D}\boldsymbol{\varepsilon}_{1,p}, \tag{8.3.6}$$

$$\begin{aligned} &\boldsymbol{\sigma}_{1,p} \bullet \boldsymbol{\varepsilon}_1 + \boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) + \boldsymbol{\sigma}_{,p} \bullet \mathbf{L}_2(\mathbf{u}_1) + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_{1,p}, \mathbf{u}_1) \\ &= (\omega^2)_{,p}\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1 + \omega^2\mathbf{M}_{,p}\mathbf{u}_1 \bullet \mathbf{u}_1 + \omega^2\mathbf{M}\mathbf{u}_{1,p} \bullet \mathbf{u}_1 . \end{aligned} \tag{8.3.7}$$

The derivatives of the vibration mode $\mathbf{u}_{1,p}$, $\boldsymbol{\sigma}_{1,p}$ can be eliminated from Eq. (8.3.7) by first setting $\delta\mathbf{u} = \mathbf{u}_{1,p}$ in Eq. (8.3.3), and using Eq. (8.2.3); this gives

$$\boldsymbol{\sigma}_1 \bullet [\mathbf{L}_1(\mathbf{u}_{1,p}) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_{1,p})] + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_1, \mathbf{u}_{1,p}) = \omega^2\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_{1,p} . \tag{8.3.8}$$

Then subtracting Eq. (8.3.8) from Eq. (8.3.7) and using Eqs. (8.3.5) and (8.3.6) we can get (Exercise 7)

$$(\omega^2)_{,p} = \frac{\mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 + 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) + \boldsymbol{\sigma}_{,p} \bullet \mathbf{L}_2(\mathbf{u}_1) - \omega^2\mathbf{M}_{,p}\mathbf{u}_1 \bullet \mathbf{u}_1}{\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1} . \tag{8.3.9}$$

The first and last terms in the numerator of Eq. (8.3.9) correspond to the derivatives of the stiffness matrix and mass matrix, respectively, in Eq. (7.3.5). When we calculate derivatives of natural frequencies the other terms in the numerator vanish. However, for the vibration frequencies of a loaded structure we need the other term which contain derivatives of the static field $\mathbf{u}$, $\boldsymbol{\sigma}$ with respect to p. These derivatives need to be calculated by solving Eqs. (8.2.13) - (8.2.15).

The derivative of the buckling load is obtained from the condition that $\omega^2 = 0$ at buckling. As p changes, μ^* must change with it so that ω^2 remains zero, that is $d(\omega^2) = 0$. Thus

$$d(\omega^2) = (\omega^2)_{,p}dp + (\omega^2)'d\mu^* = 0 \tag{8.3.10}$$

where a prime denotes derivative with respect to μ. The first term in Eq. (8.3.10) is the change in ω^2 at a fixed load level, and the second is the change in ω^2 due to a change in load level. These two changes add up to zero, so that the frequency remains zero at the buckling load. Equation (8.3.10) gives

$$\mu^*_{,p} = -\frac{(\omega^2)_{,p}}{(\omega^2)'} . \tag{8.3.11}$$

To calculate the derivative of the frequency with respect to the load parameter μ we start by differentiating Eqs. (8.3.1) - (8.3.3) with respect to μ and then set $\delta\mathbf{u} = \mathbf{u}_1$

$$\boldsymbol{\varepsilon}_1' = \mathbf{L}_1(\mathbf{u}_1') + \mathbf{L}_{11}(\mathbf{u}', \mathbf{u}_1) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_1'), \tag{8.3.12}$$

$$\boldsymbol{\sigma}_1' = \mathbf{D}\boldsymbol{\varepsilon}_1', \tag{8.3.13}$$

$$\boldsymbol{\sigma}_1' \bullet \boldsymbol{\varepsilon}_1 + \boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}', \mathbf{u}_1) + \boldsymbol{\sigma}' \bullet \mathbf{L}_2(\mathbf{u}_1) + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_1', \mathbf{u}_1) = (\omega^2)'\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1 + \omega^2 \mathbf{M}\mathbf{u}_1' \bullet \mathbf{u}_1 . \tag{8.3.14}$$

Next, we eliminate the derivatives of the vibration field with respect to μ by setting $\delta\mathbf{u} = \mathbf{u}_1'$ in Eq. (8.3.3) and using Eq. (8.2.3)

$$\boldsymbol{\sigma}_1 \bullet [\mathbf{L}_1(\mathbf{u}_1') + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_1')] + \boldsymbol{\sigma}\mathbf{b}|\mathbf{L}_{11}(\mathbf{u}_1, \mathbf{u}_1') = \omega^2 \mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1', \tag{8.3.15}$$

and then subtracting Eq. (8.3.15) from Eq.(8.3.14) and using Eqs. (8.3.2), (8.3.12), and (8.3.13) to get

$$(\omega^2)' = \frac{2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}', \mathbf{u}_1) + \boldsymbol{\sigma}' \bullet \mathbf{L}_2(\mathbf{u}_1)}{\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1} . \tag{8.3.16}$$

Finally, substituting Eqs. (8.3.9) and (8.3.16) evaluated at the buckling load into Eq. (8.3.11) gives

$$\mu_{,p}^* = -\frac{\mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 + 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}^*, \mathbf{u}_1) + \boldsymbol{\sigma}_{,p}^* \bullet \mathbf{L}_2(\mathbf{u}_1)}{2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}'^*, \mathbf{u}_1) + \boldsymbol{\sigma}'^* \bullet \mathbf{L}_2(\mathbf{u}_1)} , \tag{8.3.17}$$

where the asterisk denotes prebuckling quantities evaluated at the buckling load. Note that the field $\mathbf{u}_1$, $\boldsymbol{\sigma}_1$ now denotes the zero-frequency or buckling mode.

Example 8.3.1

The beam in Example (8.2.1) has a mass density ρ. Calculate the derivative of the lowest frequency of lateral vibration with respect to the cross-sectional area A with the applied load parameter $\mu = 1$ (assuming again that I and P do not change).

We use the same single finite-element approximation for half the beam that we used in Example (8.2.1). Assuming a symmetric mode shape, we find the vibration mode

$$u_1 = 0, \qquad w_1/L = 3\bar{x}^2 - 2\bar{x}^3 .$$

To calculate the vibration frequency we use the Rayleigh quotient Eq. (8.3.4). The first term in the numerator is

$$\boldsymbol{\sigma}_1 \bullet \boldsymbol{\varepsilon}_1 = \int_0^L (N_1\epsilon_{x1} + M_1\kappa_1)dx .$$

Using Eqs (8.3.1) and (8.3.2) and expressions from Example (8.2.1) we have

$$\epsilon_{x1} = w_{,x}w_{1,x} = 36\bar{w}(\bar{x} - \bar{x}^2)^2, \qquad N_1 = EA\epsilon_{x1} ,$$

$$\kappa_1 = w_{1,xx} = 6(1-2\bar{x})/L, \qquad M_1 = EI\kappa_1 .$$

So

$$\begin{aligned}\boldsymbol{\sigma}_1 \bullet \boldsymbol{\varepsilon}_1 &= \int_0^L [1296EA\bar{w}(\bar{x}-\bar{x}^2)^4 + 36EI(1-2\bar{x})^2/L^2]dx \\ &= 2.05714EA\bar{w}L + 12EI/L .\end{aligned}$$

The other terms in the Rayleigh quotient are

$$\boldsymbol{\sigma} \bullet L_2(\mathbf{u}_1) = \int_0^L N w_{1,x}^2 dx = 1.02857EA\bar{w}^2 L - 1.2EAL\epsilon^i ,$$

$$\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1 = \int_0^L A\rho w_1^2 dx = 0.3714\rho AL^3 ,$$

so that

$$\omega^2 = \frac{3.08571EA\bar{w}^2 + 12EI/L^2 - 1.2EA\epsilon^i}{0.3714\rho AL^2} = 0.01077\frac{E}{\rho L^2} .$$

Note that for the unloaded beam, $\bar{w} = 0$, $\epsilon^i = 0$ we get

$$\omega = 5.68\sqrt{\frac{EI}{\rho AL^4}} ,$$

which is about 1.5% above the exact answer. We can differentiate ω^2 with respect to A, for an analytical derivative that we can use later for comparison

$$(\omega^2)_{,A} = \frac{3.08571E(\bar{w}^2 + 2A\bar{w}\bar{w}_{,A}) - 1.2E\epsilon^i}{0.3714\rho AL^2} - \frac{\omega^2}{A} .$$

For the direct method to calculate the same derivative we use Eq. (8.3.9). The individual terms in this equation are calculated as follows:

$$\mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 = \int_0^L E\epsilon_{x1}^2 dx = 2.05714E\bar{w}^2 L ,$$

$$2\boldsymbol{\sigma}_1 \bullet L_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) = 2\int_0^L N_1 w_{,xA} w_{1,x} dx = 4.11428EA\bar{w}\bar{w}_{,A} L ,$$

$$\boldsymbol{\sigma}_{,p} \bullet L_2(\mathbf{u}_1) = \int_0^L N_{,A} w_{1,x}^2 dx ,$$

where

$$N_{,A} = E(\epsilon_x - \epsilon^i) + EA\epsilon_{x,A} = E(\epsilon_x - \epsilon^i) + EAw_{,x}w_{,xA} .$$

So

$$\boldsymbol{\sigma}_{,p} \bullet L_2(\mathbf{u}_1) = 1.02857E\bar{w}^2 L - 1.2E\epsilon^i L + 2.05714EA\bar{w}\bar{w}_{,A} L ,$$

and

$$\omega^2 \mathbf{M}_{,A}\mathbf{u}_1 \bullet \mathbf{u}_1 = \omega^2 \int_0^L \rho w_1^2 dx = 0.3714\omega^2\rho L^3 .$$

Altogether

$$(\omega^2)_{,A} = \frac{3.08571E\bar{w}^2 + 6.17142EA\bar{w}\bar{w}_{,A} - 1.2E\epsilon^i}{0.3714\rho AL^2} - \frac{\omega^2}{A},$$

which agrees with the analytical result. Using the values for $\bar{w}$ and $\bar{w}_{,A}$ from Example (8.2.1) we get

$$(\omega^2)_{,A} = 0.1788\frac{E}{\rho AL^2}.$$

• • •

8.3.2 *The Adjoint Method*

The direct sensitivity approach requires the calculation of sensitivities of the static field (prebuckling state), Eqs.(8.2.13) - (8.2.15). This calculation can become expensive when we need sensitivities with respect to a large number of structural parameters. In that case an adjoint technique that eliminates the need for static sensitivities is appropriate. As usual, we multiply the equations that govern static equilibrium by Lagrange multipliers (that we call the adjoint fields) and add them to to ω^2; thus

$$\begin{aligned}(\omega^2)^* =& m_0\omega^2 + \boldsymbol{\sigma}^a \bullet [\boldsymbol{\varepsilon} - \mathbf{L}_1(\mathbf{u}) - \frac{1}{2}\mathbf{L}_2(\mathbf{u})] + \boldsymbol{\varepsilon}^a \bullet [\boldsymbol{\sigma} - \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i)] + \mathbf{f} \bullet \mathbf{u}^a \\ & -\boldsymbol{\sigma} \bullet [\mathbf{L}_1(\mathbf{u}^a) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}^a)],\end{aligned} \tag{8.3.18}$$

where m_0 is the value of $\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1$ for the nominal value of p (that is m_0 does not change with p). The constant m_0 is included to simplify the final expressions for the adjoint field. We differentiate Eq. (8.3.18) and use Eq. (8.3.9) to get

$$\begin{aligned}m_0(\omega^2)_{,p} = (\omega^2)^*_{,p} =& \mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 + 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) + \boldsymbol{\sigma}_{,p} \bullet \mathbf{L}_2(\mathbf{u}_1) - \omega^2\mathbf{M}_{,p}\mathbf{u}_1 \bullet \mathbf{u}_1 \\ & + \boldsymbol{\sigma}^a \bullet \Big[\boldsymbol{\varepsilon}_{,p} - \mathbf{L}_1(\mathbf{u}_{,p}) - \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_{,p})\Big] + \boldsymbol{\varepsilon}^a \bullet [\boldsymbol{\sigma}_{,p} - \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) - \mathbf{D}\boldsymbol{\varepsilon}_{,p}] \\ & - \boldsymbol{\sigma}_{,p} \bullet \Big[\mathbf{L}_1(\mathbf{u}^a) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}^a)\Big] - \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}^a).\end{aligned} \tag{8.3.19}$$

Grouping together terms that involve displacement derivatives, strain derivatives and stress derivatives we get

$$\begin{aligned}m_0(\omega^2)_{,p} =& \mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 - \omega^2\mathbf{M}_{,p}\mathbf{u}_1 \bullet \mathbf{u}_1 - \boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) \\ & - \boldsymbol{\sigma}^a \bullet [\mathbf{L}_1(\mathbf{u}_{,p}) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}_{,p})] - \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}^a) + 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) \\ & + \boldsymbol{\varepsilon}_{,p} \bullet [\boldsymbol{\sigma}^a - \mathbf{D}\boldsymbol{\varepsilon}^a] + \boldsymbol{\sigma}_{,p} \bullet [\boldsymbol{\varepsilon}^a - \mathbf{L}_1(\mathbf{u}^a) - \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}^a) + \mathbf{L}_2(\mathbf{u}_1)].\end{aligned} \tag{8.3.20}$$

From Eq. (8.3.20) it is clear that in order to eliminate of derivatives of the static equilibrium state the adjoint state should satisfy the following equations

$$\boldsymbol{\varepsilon}^a = \mathbf{L}_1(\mathbf{u}^a) + \mathbf{L}_{11}(\mathbf{u}, \mathbf{u}^a) - \mathbf{L}_2(\mathbf{u}_1), \tag{8.3.21}$$

$$\boldsymbol{\sigma}^a = \mathbf{D}\boldsymbol{\varepsilon}^a, \tag{8.3.22}$$

$$\boldsymbol{\sigma}^a \bullet [\mathbf{L}_1(\delta\mathbf{u}) + \mathbf{L}_{11}(\mathbf{u}, \delta\mathbf{u})] + \boldsymbol{\sigma} \bullet \mathbf{L}_{11}(\mathbf{u}^a, \delta\mathbf{u}) - 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_1, \delta\mathbf{u}) = 0\,. \qquad (8.3.23)$$

Then the derivative of the frequency is given as

$$(\omega^2)_{,p} = \frac{\mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 - \omega^2 \mathbf{M}_{,p}\mathbf{u}_1 \bullet \mathbf{u}_1 - \boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i)}{\mathbf{M}\mathbf{u}_1 \bullet \mathbf{u}_1}\,. \qquad (8.3.24)$$

The adjoint equations, which have a homogeneous part identical to that of Eqs. (8.3.1) - (8.3.3) for $\omega = 0$, may be considered to be the field equations of an adjoint structure for which the term $\mathbf{L}_2(\mathbf{u}_1)$ in Eq. (8.3.21) is an initial strain term, and the last term in Eq. (8.3.23) corresponds to body-force loading. In a buckling problem ($\omega = 0$) the homogeneous part is singular, the adjoint fields are not unique, and any multiple of the buckling mode $\mathbf{u}_1$ can be added. Any convenient orthogonality relation can be used to make the adjoint fields unique.

The derivative of the buckling eigenvalue is similarly given as

$$\mu^*_{,p} = -\frac{\mathbf{D}_{,p}\boldsymbol{\varepsilon}_1 \bullet \boldsymbol{\varepsilon}_1 - \mathbf{D}_{,p}\boldsymbol{\varepsilon}^* \bullet \boldsymbol{\varepsilon}^a}{2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}'^*, \mathbf{u}_1) + \boldsymbol{\sigma}'^* \bullet \mathbf{L}_2(\mathbf{u}_1)}\,. \qquad (8.3.25)$$

Equation (8.3.25) is based on the buckling mode and the prebuckling state calculated at $\mu = \mu^*$. The usual practice, however, is to estimate the buckling load by solving a linearized eigenvalue problem based on a load $\mu < \mu^*$. It is shown in [4] that the error introduced in the derivative μ^*_p due to this approximation is of the order of $(\mu^* - \mu)^2$.

Example 8.3.2

We repeat Example (8.3.1) using the adjoint approach. We need to recalculate the two terms that depend on the derivative of the static solution. From that example these are

$$A = 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_{,p}, \mathbf{u}_1) + \boldsymbol{\sigma}_{,p} \bullet \mathbf{L}_2(\mathbf{u}_1) = 1.02857EL\bar{w}^2 - 1.2EL\epsilon^i + 6.17142AL\bar{w}\bar{w}_{,A}\,. \quad (a)$$

Using the adjoint method these two terms are replaced by the term

$$A = -\boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i)$$

in Eq. (8.3.24).

The adjoint state, defined by Eqs. (8.3.21)-(8.3.23), has an initial strain and a body force. The initial strain is $\mathbf{L}_2(\mathbf{u}_1) = w^2_{1,x}$. The corresponding equivalent nodal force, f^a_1, is

$$f^a_1 L\delta\bar{\mathbf{w}} = \int_0^L w^2_{1,x} EA\delta\epsilon_x dx\,.$$

Using expressions from Examples (8.2.1) and (8.3.1) for $\delta\epsilon_x$ and w_1 we get

$$f^a_1 = 1296\frac{EA\bar{w}}{L}\int_0^L (\bar{x} - \bar{x}^2)^4 dx = 2.05714EA\bar{w}\,.$$

The body force is

$$f_2^a L\delta\bar{w} = 2\boldsymbol{\sigma}_1 \bullet \mathbf{L}_{11}(\mathbf{u}_1, \delta\mathbf{u}) = 2\int_0^L N_{x1} w_{1,x} \delta w_{,x} dx = 2EA \int_0^L w_{1,x}^2 w_{,x} \delta w_{,x} dx$$
$$= 2592 EA\bar{w}\delta\bar{w} \int_0^L (\bar{x} - \bar{x}^2)^4 dx = 4.11428 EAL\bar{w}\delta\bar{w}\,.$$

Altogether, the total nodal force is

$$f^a = f_1^a + f_2^a = 6.17142 EA\bar{w}\,.$$

This force has to be applied to the tangent structure. This means that if we use it to replace the right-hand side of the tangent state equation, Eq. (b) of Example (8.2.1) then we must use $\bar{w}^a$ to replace w' on the left side. That is

$$3.08571 EA\bar{w}^2\bar{w}^a + 12(EI/L^2)\bar{w}^a - 1.2EA\epsilon^i\bar{w}^a = 6.17142 EA\bar{w}\,. \qquad (b)$$

For later use we compare Eq. (b) to Eq. (d) of Example (8.2.1) and note that

$$\bar{w}^a = \frac{6.17142 A\bar{w}_{,A}}{-1.02857\bar{w}^2 + 1.2\epsilon^i}\,. \qquad (c)$$

Once we have $\bar{w}^a$ from Eq. (a) we can calculate A as

$$A = -\boldsymbol{\varepsilon}^a \bullet \mathbf{D}_{,p}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^i) = -\int_0^L E\epsilon_x^a(\epsilon - \epsilon^i) dx\,.$$

From Eq. (8.3.21)

$$\epsilon_x^a = w_{,x}^a w_{,x} - w_{1,x}^2 = 36(\bar{x} - \bar{x}^2)^2(\bar{w}^a\bar{w} - 1)\,,$$

so that

$$\begin{aligned} A =& 36(1 - \bar{w}^a\bar{w}) \int_0^L (\bar{x} - \bar{x}^2)^2 \left[18(\bar{x} - \bar{x}^2)^2 - \epsilon^i\right] dx \\ =& (1 - \bar{w}^a\bar{w}) EL(1.02857\bar{w}^2 - 1.2\epsilon^i)\,. \end{aligned} \qquad (d)$$

We can now calculate w^a from Eq. (b) and A from Eq. (d) to get the derivative of ω^2 without calculating $w_{,A}$. To check that Eq. (d) gives the same result as Eq.(a) we use Eq. (c) to obtain

$$(1 - \bar{w}^a\bar{w}) = \frac{6.17142 A\bar{w}_{,A} + 1.02857\bar{w}^2 - 1.2\epsilon^i}{1.02857\bar{w}^2 - 1.2\epsilon^i}\,.$$

Substituting this expression into Eq. (d) we find Eq. (a).• • •

8.4 Static Shape Sensi ity

The calculation of sensitivity with respect to shape variation is more complicated than that for stiffness variation. The present section is limited to shape sensitivity of static response in the linear elastic range, and is based on Refs. [5-11]. Furthermore, the discussion is limited to formulations which do not have curvature (such as arch and shell formulations). The reader is also referred to Ref. [2] for proofs of many of the results presented here.

Two general approaches have been used for variational shape sensitivity. The first and more popular is the material derivative approach, and the second one is the domain parametrization approach, also known as the control volume approach. While both methods are very general, the domain parametrization approach is simpler, and is particularly powerful for finite element analysis with isoparametric elements. We start this section with a discussion of these two methods, and then see how they can be applied with the direct and adjoint methods.

8.4.1 *The Material Derivative*

Consider a shape variation field ϕ such that the a material particle located at $\mathbf{x}$ is moved to $\mathbf{x}_\phi$

$$\mathbf{x}_\phi = \mathbf{x} + \phi(\mathbf{x}, p)\,, \tag{8.4.1}$$

where p is a shape design variable. The coordinate $\mathbf{x}$ is typically referred to as the material or Lagrangian coordinate in that it is associated with a material particle.

The variation changes the domain V and the boundary S of the structure as shown in Figure 8.4.1.

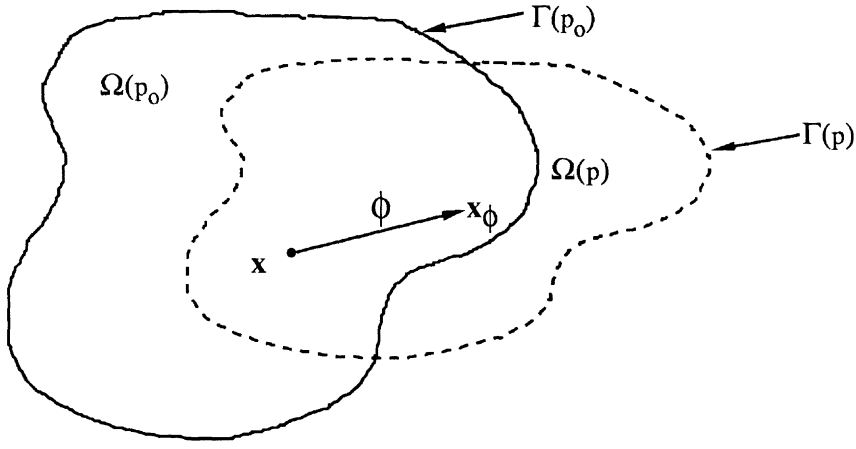


Figure 8.4.1 Shape variation of structural domain.

Consider a function $f(\mathbf{x}, p)$ defined on the changing structural domain V. We denote the partial derivative $\partial f/\partial p$ of f with respect to p by $f_{,p}$. This derivative measures the change in f at a fixed position in the structure, and is often referred to as the local derivative. The derivative that measures the change in f at a fixed material point needs to take into account also the change in $\mathbf{x}$ as p changes. This derivative is called the material derivative or the total derivative of f, and is denoted here by f_p

$$f_p = f_{,p} + \nabla f^T \mathbf{x}_{\phi,p} = f_{,p} + \nabla f^T \mathbf{v}\,, \tag{8.4.2}$$

where ∇f denotes the gradient of f in space, and

$$\mathbf{v} = \mathbf{x}_{\phi,p} = \boldsymbol{\phi}_{,p} \tag{8.4.3}$$

is often referred to as the shape velocity field. This terminology is based on viewing p as a time-like variable, so that $\mathbf{x}_{\phi,p}$ is a velocity field. The components of $\mathbf{v}$ are denoted by v_k where k runs from 1 to the dimension of the problem, and $v_1 = v_x$, $v_2 = v_y$, $v_3 = v_z$.

Consider now a vector function such as the displacement field $\mathbf{u}$. For each component u_i of $\mathbf{u}$ we can use Eq. (8.4.1) to obtain the material derivative as

$$u_{ip} = u_{i,p} + (\nabla u_i)^T \mathbf{v}\,. \tag{8.4.4}$$

This equation is abbreviated as

$$\mathbf{u}_p = \mathbf{u}_{,p} + (\nabla \mathbf{u})\mathbf{v}\,, \tag{8.4.5}$$

where $\nabla \mathbf{u}$ is a matrix called the deformation gradient with components given by

$$(\nabla \mathbf{u})_{ij} = u_{i,j} \equiv \frac{\partial u_i}{\partial x_j}\,. \tag{8.4.6}$$

Note that a comma followed by an index subscript j denotes differentiation with respect to x_j. From this definition we get that

$$(\nabla \mathbf{u})\mathbf{v} = \mathbf{u}_{,j} v_j\,, \tag{8.4.7}$$

with repeated indices summed over the dimension of the problem, so that for the two-dimensional case

$$\mathbf{u}_{,i} v_i = \mathbf{u}_{,1} v_1 + \mathbf{u}_{,2} v_2 = \mathbf{u}_{,x} v_x + \mathbf{u}_{,y} v_y\,. \tag{8.4.8}$$

Similarly for a tensor such as the strain tensor $\boldsymbol{\varepsilon}$ the material derivative is given by

$$\boldsymbol{\varepsilon}_p = \boldsymbol{\varepsilon}_{,p} + (\nabla \boldsymbol{\varepsilon})\mathbf{v} = \boldsymbol{\varepsilon}_{,p} + \boldsymbol{\varepsilon}_{,i} v_i\,. \tag{8.4.9}$$

Typically the material derivative is more physically interesting than the local derivative. For example, if we change the shape of a hole boundary to relieve stress concentration at that boundary, we would want the derivative of the stress at the boundary rather than at a point with fixed coordinates. Mathematically, the material

derivative is more complicated to handle than the local derivative. For example, the local derivative commutes with differentiation with respect to coordinates while the material derivative does not. Consider, for example, the strain field associated with a displacement field $\mathbf{u}$, and denote it as $\boldsymbol{\varepsilon}(\mathbf{u})$. The strain is obtained from the displacements by differentiation, and since we can change the order of differentiation for local derivatives

$$\boldsymbol{\varepsilon}_{,p}(\mathbf{u}) = \boldsymbol{\varepsilon}(\mathbf{u}_{,p}) , \tag{8.4.10}$$

while we cannot write a similar equation for the material derivative $\boldsymbol{\varepsilon}_p$.

In order to differentiate the virtual work equation with respect to p we need to calculate derivatives of integrals over the volume and over the surface of the structure. Let I_V denote an integral over the domain of the structure

$$I_V = \int_V f(\mathbf{x}, p) dV . \tag{8.4.11}$$

The derivative of I_V with respect to p is

$$I_{Vp} = \int_V f_p dV + \int_V f(dV)_p = \int_V (f_p + \bar{V}_p f) dV , \tag{8.4.12}$$

where $\bar{V}_p$ is the relative change in volume. It can be shown (e.g., [2]) that

$$(dV)_p = \bar{V}_p dV = v_{k,k} dV . \tag{8.4.13}$$

Recall that repeated indices are summed over the dimensionality of the problem, so that in the three-dimensional case

$$v_{k,k} = v_{1,1} + v_{2,2} + v_{3,3} = v_{x,x} + v_{y,y} + v_{z,z} = div\ \mathbf{v} . \tag{8.4.14}$$

The derivative of the surface integral

$$I_S = \int_S f(\mathbf{x}, p) dS \tag{8.4.15}$$

is handled in a similar manner; thus

$$I_{Sp} = \int_S f_p dS + \int_S f(dS)_p . \tag{8.4.16}$$

The derivative of the surface element is given as

$$(dS)_p = \bar{S}_p dS = -H \mathbf{n}^T \mathbf{v} dS , \tag{8.4.17}$$

where $\mathbf{n}$ is the vector normal to the boundary S, and H is the curvature of S in two dimensions and twice the mean curvature in three dimensions.

8.4.2 *Domain Parametrization*

The discussion of the domain parametrization is based on the work of Haber and coworkers, and in particular [11]. With this approach the material coordinate vector $\mathbf{x}$ is given in terms of some reference domain as

$$\mathbf{x} = \mathbf{x}(\mathbf{r}, p)\,, \tag{8.4.18}$$

where $\mathbf{r}$ is a coordinate vector in the reference domain Ω with boundary Γ, and p is a shape parameter (see Figure 8.4.2). When isoparametric elements are used, it is convenient to use the parent element as the reference domain for the actual element. Specifically, for isoparametric elements the coordinate vector $\mathbf{x}$ in the element is written as

$$\mathbf{x} = \sum_{i=1}^{\#nodes} h_i(\mathbf{r})\mathbf{d}_i(p)\,, \tag{8.4.19}$$

where h_i are shape functions for the element, $\mathbf{r}$ is a vector of intrinsic coordinates, and $\mathbf{d}_i$ are vectors of nodal coordinates. Variations in geometry are represented by variations of the nodal coordinates, with the shape functions held fixed.

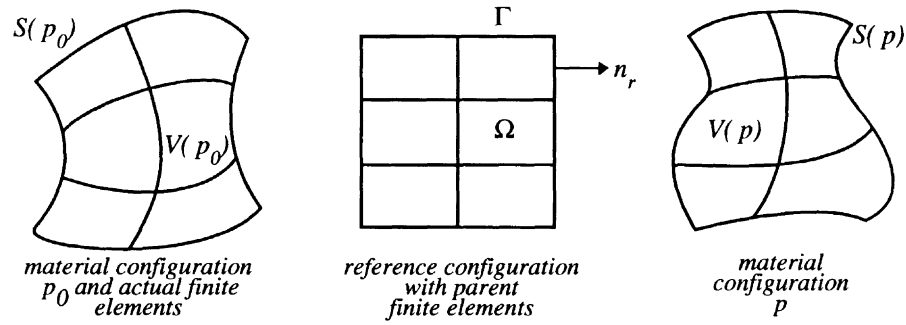


Figure 8.4.2 Domain parametrization approach

The transformation between the reference domain and material domain is characterized by the Jacobian of the transformation $\mathbf{J}^E$, known as the Eulerian Jacobian, and its inverse $\mathbf{J}^{-E}$

$$J_{ij} = \frac{\partial r_i}{\partial x_j} = r_{i,j}, \qquad \text{and} J_{ij}^{-E} = \frac{\partial x_i}{\partial r_j} = x_{i,j}\,. \tag{8.4.20}$$

Note that a comma followed by an index subscript (such as i or j) denotes differentiation with respect to a material coordinate, while a dot followed by an index subscript denotes differentiation with respect to a reference coordinate. The differential volume and area in the material configuration are expressed in terms of the reference configuration using the determinant of the Eulerian Jacobian

$$dV = det(\mathbf{J}^{-E})d\Omega, \qquad dS = det(\mathbf{K}^{-E})d\Gamma\,, \tag{8.4.21}$$

where $\mathbf{K}^{-E}$ is a Jacobian of the transformation between the surface coordinates in the reference and material configurations. Its determinant is given as

$$det(\mathbf{K}^{-E}) = (J^E_{ji} J^E_{ki} n^r_j n^r_k)^{1/2} det(\mathbf{J}^{-E}), \tag{8.4.22}$$

where n^r_i are the components of the unit outward normal to the surface area Γ of the reference domain, and repeated indices are summed. The derivative of $\mathbf{J}^{-E}$ with respect to p is obtained from its definition

$$J^{-E}_{ij,p} = (x_{i,j})_{,p}, \tag{8.4.23}$$

while the derivative of $\mathbf{J}^E$ requires using the formula for the derivative of an inverse

$$\mathbf{J}^E_{,p} = -\mathbf{J}^E \mathbf{J}^{-E}_{,p} \mathbf{J}^E, \tag{8.4.24}$$

to get

$$J^E_{ij,p} = -J^E_{ik} J^E_{lj} (x_{k,l})_{,p}. \tag{8.4.25}$$

With the domain parametrization approach the displacements, strains and stresses are considered to be functions of the reference coordinates $\mathbf{r}$. Therefore, when we evaluate their derivatives with respect to p we get derivatives for a constant position $\mathbf{r}$ which are essentially the material derivatives of these quantities. A function $f(\mathbf{x}, p)$ is first rewritten in terms of the reference coordinates as $\bar{f}(\mathbf{r}, p)$ and then

$$f_p = \frac{\partial \bar{f}}{\partial p}. \tag{8.4.26}$$

Derivatives with respect to material coordinates have to be transformed to derivatives with respect to reference coordinates using the chain rule of differentiation. Thus, the linear strain displacement relationship becomes

$$\epsilon_{ij} = \tfrac{1}{2}(u_{i,j} + u_{j,i}) = \tfrac{1}{2}(u_{i,k} J^E_{kj} + u_{j,k} J^E_{ki}). \tag{8.4.27}$$

This produces an explicit dependence of the strain on the shape parameter, and to reflect that we rewrite Eq. (8.1.1) as

$$\boldsymbol{\varepsilon} = \mathbf{L}_1(\mathbf{u}, p). \tag{8.4.28}$$

Derivatives of integrals are handled in a similar way to the material derivative approach. A volume integral is written in terms of the reference coordinates

$$I_V = \int_V f(\mathbf{x}, p) dV = \int_\Omega \bar{f}(\mathbf{r}, p) det(J^{-E}) d\Omega, \tag{8.4.29}$$

where $\bar{f}$ is the new form of the function when it is written in terms of the reference coordinates. Then

$$I_{Vp} = \int_\Omega (f_p + \bar{V}_p f) det(\mathbf{J}^{-E}) \mathbf{d}\Omega, \tag{8.4.30}$$

where

$$\bar{V}_p = \Big(det(\mathbf{J}^{-E})\Big)_p / det(\mathbf{J}^{-E})\,. \tag{8.4.31}$$

Similarly, for a surface integral I_S, Eq. (8.4.15), we get

$$I_{Sp} = \int_\Gamma (f_p + \bar{S}_p f) det(\mathbf{K}^{-E}) d\Gamma\,, \tag{8.4.32}$$

where

$$\bar{S}_p = \Big(det(\mathbf{K}^{-E})\Big)_p / det(\mathbf{K}^{-E})\,. \tag{8.4.33}$$

8.4.3 *The Direct Method*

To apply the direct method to shape sensitivity calculation we need to differentiate the strain displacement relation, Eq. (8.1.1), Hooke's law, Eq. (8.1.4) and the equilibrium equations, Eq. (8.1.6) with respect to p. We start with the strain displacement relation and with the material derivative approach. Using Eqs. (8.4.9) and (8.4.10) the differentiated strain-displacement relation is

$$\boldsymbol{\varepsilon}_p = \boldsymbol{\varepsilon}_{,p} + (\nabla\boldsymbol{\varepsilon})\mathbf{v} = \mathbf{L}_1(\mathbf{u}_{,p}) + (\nabla\boldsymbol{\varepsilon})\mathbf{v}\,. \tag{8.4.34}$$

Using Eq. (8.4.5) we can write this as

$$\boldsymbol{\varepsilon}_p = \mathbf{L}_1(\mathbf{u}_p) - \bar{\boldsymbol{\varepsilon}}\,, \tag{8.4.35}$$

where

$$\bar{\boldsymbol{\varepsilon}} = \mathbf{L}_1\Big[(\nabla\mathbf{u})\mathbf{v}\Big] - (\nabla\boldsymbol{\varepsilon})\mathbf{v} \tag{8.4.36}$$

is an initial-strain associated with the sensitivity field. Even though Eq. (8.4.36) appears to include strain gradients, these gradients cancel out and $\bar{\boldsymbol{\varepsilon}}$ includes only first derivatives of the displacement and shape velocity fields. For example, for the three-dimensional case we obtain

$$\bar{\epsilon}_{ij} = \tfrac{1}{2}(u_{i,k}v_{k,j} + u_{j,k}v_{k,i})\,. \tag{8.4.37}$$

We can get another form of $\bar{\boldsymbol{\varepsilon}}$ by using the domain parametrization approach. Differentiating Eq. (8.4.27) we get

$$(\epsilon_{ij})_p = \tfrac{1}{2}(u_{pi.k}J^E_{kj} + u_{pj.k}J^E_{ki}) + \tfrac{1}{2}(u_{i.k}J^E_{kj,p} + u_{j.k}J^E_{ki,p}) = [\mathbf{L}_1(\mathbf{u}_p)]_{ij} - \bar{\epsilon}_{ij}\,, \tag{8.4.38}$$

where

$$\bar{\epsilon}_{ij} = -\tfrac{1}{2}(u_{i.k}J^E_{kj,p} + u_{j.k}J^E_{ki,p})\,. \tag{8.4.39}$$

We assume that the elastic coefficients do not change with shape change, and that there is no initial strain. Then the derivative of Hooke's law is

$$\boldsymbol{\sigma}_p = \mathbf{D}\boldsymbol{\varepsilon}_p\,. \tag{8.4.40}$$

The derivative of the equations of equilibrium is

$$(\boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon})_p = (\mathbf{f} \bullet \delta\mathbf{u})_p \,. \tag{8.4.41}$$

The term on the left side of Eq. (8.4.41) is a volume integral which according to Eqs. (8.4.12) and (8.4.30) is the volume integral of the derivative of the integrand plus a term which accounts for the change in the volume element. This translates to

$$(\boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon})_p = \boldsymbol{\sigma}_p \bullet \delta\boldsymbol{\varepsilon} + \boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon}_p + \boldsymbol{\sigma} \bullet (\bar{V}_p \delta\boldsymbol{\varepsilon}) \,. \tag{8.4.42}$$

The derivative of the virtual strain $\delta\boldsymbol{\varepsilon}_p$ is obtained in a similar manner to Eq. (8.4.35) as

$$\delta\boldsymbol{\varepsilon}_p = \mathbf{L}_1(\delta\mathbf{u}_p) - \delta\bar{\boldsymbol{\varepsilon}} \,, \tag{8.4.43}$$

where with the material derivative approach

$$\delta\bar{\boldsymbol{\varepsilon}} = \mathbf{L}_1[\nabla(\delta\mathbf{u})\mathbf{v}] - (\nabla\delta\boldsymbol{\varepsilon})\mathbf{v} \,, \tag{8.4.44}$$

while for the domain parametrization approach

$$\delta\bar{\varepsilon}_{ij} = -\tfrac{1}{2}(\delta u_{i,k} J^E_{kj,p} + \delta u_{j,k} J^E_{ki,p}) \,. \tag{8.4.45}$$

The derivative of the virtual work of the applied loads is more complicated because this work is composed of volume and surface integrals

$$\mathbf{f} \bullet \delta\mathbf{u} = \mathbf{f}_b \bullet \delta\mathbf{u} + \mathbf{T} \bullet \delta\mathbf{u} \,, \tag{8.4.46}$$

where $\mathbf{f}_b$ denotes the body load vector, and $\mathbf{T}$ denotes the vector of applied tractions. The first term on the right hand side of Eq. (8.4.46) is a volume integral, while the second term is a surface integral. Differentiating the body force integral is straight forward. However, the traction term can be a problem if there are corners on the boundary or if the loaded boundary is changing. We will assume that there are no corners or changes in loaded boundary. Then we can differentiate Eq. (8.4.46) to get

$$(\mathbf{f} \bullet \delta\mathbf{u})_p = \mathbf{f}_{bp} \bullet \delta\mathbf{u} + \mathbf{f}_b \bullet \delta\mathbf{u}_p + \mathbf{f}_b \bullet (\bar{V}_p \delta\mathbf{u}) + \mathbf{T}_p \bullet \delta\mathbf{u} + \mathbf{T} \bullet \delta\mathbf{u}_p + \mathbf{T} \bullet (\bar{S}_p \delta\mathbf{u}) \,. \tag{8.4.47}$$

The virtual displacement $\delta\mathbf{u}$ is arbitrary except that it needs to satisfy the kinematic boundary conditions, which are assumed to be independent of p. We make sure that $\delta\mathbf{u}$ satisfies these boundary conditions as the shape changes by requiring that

$$\delta\mathbf{u}_p = 0 \,. \tag{8.4.48}$$

Using Eq. (8.4.48), Eq. (8.4.43) becomes

$$\delta\boldsymbol{\varepsilon}_p = -\delta\bar{\boldsymbol{\varepsilon}} \,. \tag{8.4.49}$$

Finally, using Eqs. (8.4.41), (8.4.42), (8.4.47) and (8.4.49) we get

$$\boldsymbol{\sigma}_p \bullet \delta\boldsymbol{\varepsilon} = \mathbf{f}_{bp} \bullet \delta\mathbf{u} + \mathbf{f}_b \bullet (\bar{V}_p \delta\mathbf{u}) + \mathbf{T}_p \bullet \delta\mathbf{u} + \mathbf{T} \bullet (\bar{S}_p \delta\mathbf{u}) + \boldsymbol{\sigma} \bullet \delta\bar{\boldsymbol{\varepsilon}} - \boldsymbol{\sigma}\bar{V}_p \bullet \delta\boldsymbol{\varepsilon} \,. \tag{8.4.50}$$

The right hand side of Eq. (8.4.50) represents the body forces that need to be applied to the structure (along with the initial strain $\bar{\boldsymbol{\varepsilon}}$) in order for the solution to be the sensitivity field. The pseudo load $\mathbf{f}^p$ that needs to be applied to the original structure to produce the sensitivity field includes the terms on the right hand side of Eq. (8.4.50) as well as a pseudo force due to the initial strain $\bar{\boldsymbol{\varepsilon}}$

$$\mathbf{f}^p \bullet \delta\mathbf{u} = \mathbf{f}_{bp} \bullet \delta\mathbf{u} + \mathbf{f}_b \bar{V}_p \bullet \delta\mathbf{u} + \mathbf{T}_p \bullet \delta\mathbf{u} + \mathbf{T}\bar{S}_p \bullet \delta\mathbf{u} + \boldsymbol{\sigma} \bullet \delta\bar{\boldsymbol{\varepsilon}} - \boldsymbol{\sigma}\bar{V}_p \bullet \delta\boldsymbol{\varepsilon} + \mathbf{D}\bar{\boldsymbol{\varepsilon}} \bullet \delta\boldsymbol{\varepsilon} \,. \tag{8.4.51}$$

When the curve separating the loaded and unloaded boundaries is changing, and when the boundary has corners, there are additional terms (see [6]). By using Eq. (8.4.51), we may write Eq. (8.4.50) as

$$\boldsymbol{\sigma}_p \bullet \delta\boldsymbol{\varepsilon} = \mathbf{f}^p \bullet \delta\mathbf{u} - \mathbf{D}\bar{\boldsymbol{\varepsilon}} \bullet \delta\boldsymbol{\varepsilon} \tag{8.4.52}$$

Example 8.4.1

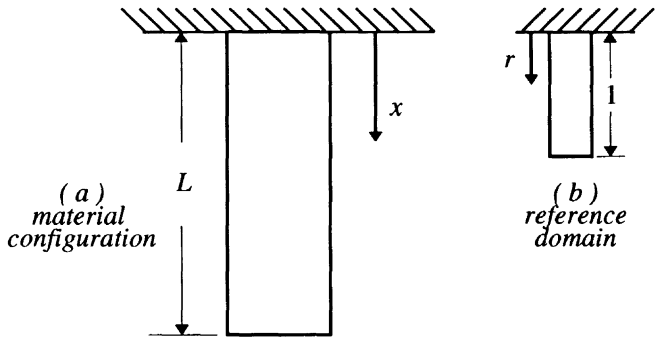


Figure 8.4.3 Bar under self-weight loading.

The bar shown in Figure (8.4.3) is loaded under its own weight. Calculate the sensitivity of the solution to changes in the length of the bar (approximated by a single finite element) using the direct method.

The loading in this case is a body force of constant magnitude $f = \rho Ag$. The exact solution for the displacement u and the member force N is given in terms of the density ρ, the area A and the acceleration due to gravity g as

$$N = \rho Ag(L - x), \quad u = (\rho g/E)(Lx - x^2/2)\,.$$

Using a single linear finite element we concentrate half of the body force at each node, so that each node is loaded by $\rho AgL/2$. The finite-element solution is

$$u_2 = \rho gL^2/2E, \quad \epsilon = \rho gL/2E, \quad N = \rho AgL/2\,,$$

so that the maximum displacement is correct, but the maximum member force is off by a factor of 2. The derivatives of these two quantities with respect to L are

$$u_{2L} = \rho gL/E, \qquad N_L = \rho Ag/2\,. \tag{a}$$

To calculate the sensitivity field with the material derivative approach we need to assume a shape variation field ϕ. We assume that as the length of the bar changes, all points in the bar are moved proportionately. Denoting the new length of the bar by p we find

$$x_\phi = x(p/L), \qquad \text{or} \quad \phi = x(p/L - 1)\,,$$

and the shape velocity field

$$v = \phi_{,p} = x/L\,.$$

We now have

$$\bar{V}_p = v_{k,k} = v_{,x} = 1/L, \quad \bar{\epsilon} = u_{,x}v_{,x} = \epsilon/L, \quad \bar{\delta\epsilon} = \delta u_{,x}v_{,x} = \delta\epsilon/L\,.$$

For the domain parametrization method we use a parent element of length one, so that

$$x = x_1(1-r) + x_2 r, \qquad u = u_1(1-r) + u_2 r\,,$$

where in our case $x_1 = 0$, $x_2 = p$. The Jacobian of the transformation is a single number

$$J^{-E} = \frac{\partial x}{\partial r} = -x_1 + x_2 = p\,.$$

Then $J^E = 1/p$, so that from Eq. (8.4.39)

$$\bar{\epsilon} = -u_{1,1} J^E_{,p} = -\frac{\partial u}{\partial r}\frac{-1}{p^2} = \frac{u_2}{p^2}\,,$$

which is the same as $\bar{\epsilon}$ obtained from the material derivative approach. The relative volume change, $\bar{V}_p$, is

$$\bar{V}_p = \Big(det(J^{-E})\Big)_{,p} / det(J^{-E} = 1/p\,,$$

which also agrees with the material derivative result. The first term in the pseudo load expression Eq. (8.4.52) is zero, because the body load is constant. The second term introduces a body force of $f/L = \rho A g/L$ which accounts for the effect of change in volume element on the resultant of the original body force. This is equivalent to an end load of $\rho A g/2$. The two terms associated with the tractions vanish because we have no applied surface tractions. The next term two terms are evaluated using the fact that for the finite-element model the member force N and the strain ϵ are constant in the element

$$\boldsymbol{\sigma} \bullet \delta\bar{\boldsymbol{\varepsilon}} - \bar{V}_p \boldsymbol{\sigma} \bullet \delta\boldsymbol{\varepsilon} = \int_0^L (N\delta\epsilon/L)dx - \int_0^L (1/L)N\delta\epsilon = 0\,.$$

The last term is

$$\mathbf{D}\bar{\boldsymbol{\varepsilon}} \bullet \delta\boldsymbol{\varepsilon} = \int_0^L (EA\epsilon\delta\epsilon/L)dx = N\delta\epsilon = (\rho A g/2)\delta u_2\,.$$

Altogether

$$\mathbf{f}^p \bullet \delta\mathbf{u} = \rho A g \delta u_2$$

which indicates that $\mathbf{f}^p$ is equal to a force of $\rho A g$. Under this force we get

$$u_{2p} = \rho g L/E$$

which agrees with the results in Eq. (a) above. To calculate the derivative of the member force, N_p we first calculate ϵ_p from Eq. (8.4.19).

$$\epsilon_p = L_1(u_p) - \bar{\epsilon} = u_{p,x} - \epsilon/L = u_{2p}/L - \epsilon/L$$

so that

$$\epsilon_p = \rho g/2E, \quad N_p = EA\epsilon_p = \rho A g/2$$

which agrees with the result in Eq. (a) above.• • •

8.4.4 *The Adjoint Method*

Consider now the sensitivity of a displacement functional H given by Eq. (8.1.19). Using Eq. (8.4.12) to obtain the derivative with respect to the first argument, we differentiate H to obtain

$$H_p = \int (h_p + h\bar{V}_p) dV + \mathbf{h}_{,u} \bullet \mathbf{u}_p \tag{8.4.53}$$

To eliminate of the displacement derivative term, we multiply the derivatives of the governing equations, Eqs (8.4.35), (8.4.40), and (8.4.53) by adjoint fields as Lagrange multipliers and add them to H_p to obtain

$$\begin{aligned} H_p &= \int (h_p + h\bar{V}_p) dV + \mathbf{h}_{,u} \bullet \mathbf{u}_p + \boldsymbol{\sigma}^a \bullet \left[\boldsymbol{\varepsilon}_p - \mathbf{L}_1(\mathbf{u}_p) + \bar{\boldsymbol{\varepsilon}} \right] \\ &\quad + \boldsymbol{\varepsilon}^a \bullet (\boldsymbol{\sigma}_p - \mathbf{D}\boldsymbol{\varepsilon}_p) - \boldsymbol{\sigma}_p \bullet \mathbf{L}_1(\mathbf{u}^a) + \mathbf{f}^p \bullet \mathbf{u}^a - \mathbf{D}\bar{\boldsymbol{\varepsilon}} \bullet \mathbf{L}_1(\mathbf{u}^a) \\ &= \int (h_p + h\bar{V}_p) dV + \mathbf{h}_{,u} \bullet \mathbf{u}_p - \boldsymbol{\sigma}^a \bullet \mathbf{L}_1(\mathbf{u}_p) + \boldsymbol{\varepsilon}_p \bullet (\boldsymbol{\sigma}^a - \mathbf{D}\boldsymbol{\varepsilon}^a) \\ &\quad + \boldsymbol{\sigma}_p \bullet \left[\boldsymbol{\varepsilon}^a - \mathbf{L}_1(\mathbf{u}^a) \right] + \left[\boldsymbol{\sigma}^a - |D\mathbf{L}_1(\mathbf{u}^a) \right] \bullet \bar{\boldsymbol{\varepsilon}} + \mathbf{f}^p . \end{aligned} \tag{8.4.54}$$

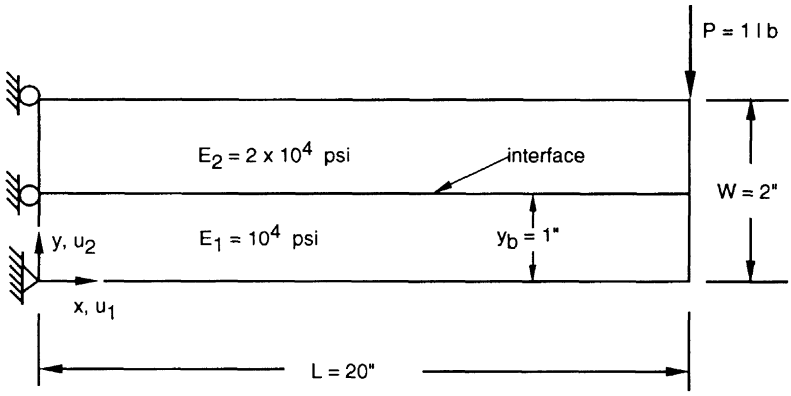
From Eq. (8.4.54) we see that we can eliminate of the response sensitivity terms by defining the adjoint as we did in the stiffness variable case, Eqs. (8.1.24)–(8.1.26). Then we get

$$H_p = \int (h_p + h\bar{V}_p) dV + \mathbf{f}^p \bullet \mathbf{u}^a . \tag{8.4.55}$$

Equation (8.4.55) requires the evaluation of $\mathbf{f}^p$ from the volume integrals in Eq. (8.4.51). It is possible to transform $\mathbf{f}^p \bullet \mathbf{u}^a$ to surface integrals (e.g., [6], [7]). However, there has been unfavorable computational experience with the surface version of the adjoint method (e.g., [12]). Unfortunately, it is not always possible to tell which method gives more accurate results as demonstrated in the following Example.

Example 8.4.2

The cantilever beam shown in Figure (8.4.4) is modeled with rectangular plane stress elements. The beam is composed of two materials with different Young's modulus, and the position of the interface between the two is the design parameter p. The sensitivity of the tip displacement with respect to the position of the interface was calculated using six methods: (i)Overall finite-differences (OFD); (ii) the semi-analytical method (SA); (iii)the discrete direct method (DD); (iv) the direct variational method (DV); (v) the adjoint variational domain method (AVD); and (vi) the adjoint variational surface method (AVS). The first three methods are discussed in Chapter 7, the next two in this chapter, and the last method in [6].



nx = number of elements along x

nys = number of elements along y under interface

nyh = number of elements along y above intetrface

Figure 8.4.4 Geometry, loading and mesh definition of cantilever beam modeled by plane stress elements (from [13].)

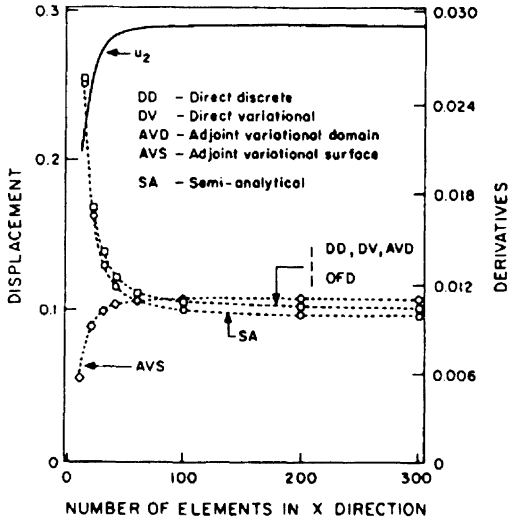


Figure 8.4.5 Convergence of the tip displacement and its derivative for $ny = 8$.

The convergence of the the displacement and its derivative as the number of elements along the axis of the beam is increased, is shown in Figure (8.4.5). It is clear that the derivatives converge more slowly than the displacement, which is to be expected. It is also seen, that though several methods including the direct methods and the adjoint variational domain method agree very well with the overall finite difference method, they are not more accurate than the adjoint variational surface method. They just converge to the correct value from a different direction.• • •

8.5 Exercises

1. The three-bar truss of Example 8.1.1 is loaded by heating member A by ΔT degrees instead of by mechanical loads. Use the direct method to calculate the derivatives of the stresses in the three members with respect to the cross-sectional area of member A in terms of A, l, E, ΔT and the coefficient of thermal expansion α. 2. Derive the expression for the adjoint loading in the case of linear structural analysis for a functional $g(T_i)$, where T_i are the components of the traction on the boundary.

3. Derive the expression for $G_{,p}$ of section 8.1.2 for the case of nonzero initial strain.

4. Using the results obtained in the previous problem calculate the derivatives of the stress in member A of the three-bar truss of Problem 1 with respect to the two cross-sectional areas using the adjoint method.

5. Show that Eqs. (8.1.27) and (8.1.40) are applicable also for the nonlinear case.

6. Calculate the derivative of the axial strain σ_x in Example 8.2.1 with respect to A using the direct and adjoint methods.

7. Derive Eq. (8.3.9)

8. Repeat Example 8.4.1 using the adjoint method.

8.6 References

[1] Cohen, G.A., "FASOR—A program for Stress, Buckling, and Vibration of Shells of Revolution", Advances in Engineering Software, 3 (4), pp.155–162, 1981.

[2] Haug, E.J., Choi, K.K., and Komkov, V., Design Sensitivity Analysis of Structural Systems, Academic Press, 1986.

[3] Mróz, Z., Kamat, M.P., and Plaut, R.H., "Sensitivity Analysis and Optimal Design of Nonlinear Beams and Plates," J. Struct. Mech., 13 (3/4), pp. 245–266, 1985.

[4] Cohen, G.A., "Effect of Nonlinear Prebuckling State on the Postbuckling Behavior and Imperfection Sensitivity of Elastic Structures", AIAA Journal, 6 (8), pp. 1616-1619, 1968.

[5] Mróz, Z., "Sensitivity Analysis and Optimal Design with Account for Varying Shape and Support Conditions", In Computer Aided Optimal Design: Structural and Mechanical Systems (C.A. Mota Soares, Editor), Springer-Verlag, 1987, pp. 407–438.

[6] Choi, K.K., "Shape Design Sensitivity Analysis and Optimal Design of Structural Systems", In Computer Aided Optimal Design: Structural and Mechanical Systems (C.A. Mota Soares, Editor), Springer-Verlag, 1987, pp. 439–492.

[7] Yang, R.-J., "A Three Dimensional Shape Optimization System—SHOP3D", Computers and Structures, 31(8), pp. 881–890, 1989.

[8] Dems, K., and Haftka, R.T., "Two Approaches to Sensitivity Analysis for Shape Variation of Structures," Mechanics of Structures and Machines, Vol. 16, No. 4, pp. 501–522, 1988/89.

[9] Dems, K., and Mróz, Z., "Variational Approach by Means of Adjoint Systems to Structural Optimization and Sensitivity Analysis, Part I: Variation of Material Parameters within Fixed Domain," Int. J. Solids Struct., 19 (8), pp. 677–692, 1983, "Part II: Structure Shape Variation," 20, pp. 527–552, 1984.

[10] Phelan, D.G., and Haber, R.B., "Sensitivity Analysis of Linear Elastic Systems Using Domain Parametrization and a Mixed Mutual Energy Principle," Computer Methods in Applied Mechanics and Engineering, Vol. 77, pp. 31–59, 1989.

[11] Arora, J.S., and Cardoso, J.B., "A Variational Principle for Shape Design Sensitivity Analysis," AIAA Paper 91-1213-CP, Proceedings AIAA/ASME/ASCE-/AHS/ASC 32nd Structures, Structural Dynamics and Material Conference, Baltimore, MD, April 8–10, 1991, Part 1, pp. 664–674.

[12] Choi, K.K. and Seong, H.G., "A Domain Method for Shape Design Sensitivity of Built-Up Structures", Computer Methods in Applied Mechanics and Engineering, Vol. 57, pp. 1–15, 1986.

[13] Haftka, R.T., and Barthelemy, B., "On the Accuracy of Shape Sensitivity Derivatives", In: Eschenauer, H.A., and Thierauf, G. (eds), Discretization Methods and Structural Optimization—Procedures and Applications, pp. 136–144, Springer-Verlag, Berlin, 1989.

Dual and Optimality Criteria Methods 9

In most of the analytically solved examples in Chapter 2, the key to the solution is the use of an algebraic or a differential equation which forms the optimality condition. For an unconstrained algebraic problem the simple optimality condition is the requirement that the first derivatives of the objective function vanish. When the objective function is a functional the optimality conditions are the Euler-Lagrange equations (e.g., Eq. (2.2.13)). On the other hand, the numerical solution methods discussed in chapters 4 and 5 (known as *direct search methods*) do not use the optimality conditions to arrive at the optimum design. The reader may have wondered why we do not have numerical methods that mimic the solution process for the problems described in Chapter 2. In fact, such numerical methods do exist, and they are known as optimality criteria methods. One reason that the treatment of these methods is delayed until this chapter is their limited acceptance in the optimization community. While the direct search methods discussed in Chapters 4 and 5 are widely used in many fields of engineering, science and management science, optimality criteria method have been used mostly for structural optimization, and even in this field there are many practitioners that dispute their usefulness.

The usefulness of optimality criteria methods, however, becomes apparent when we realize their close relationship with duality and dual solution methods (see Section 3.7). This relationship, established by Fleury, helps us understand that these methods can be very efficient when the number of constraints is small compared to the number of design variables. This chapter attempts to demonstrate the power of optimality criteria methods and dual methods for the case where the number of constraints is small. In particular, when there is only one constraint (plus possibly lower and upper limits on the variables) there is little doubt that dual methods and optimality criteria methods are the best solution approaches. The chapter begins with the discussion of intuitive optimality criteria methods. These have motivated the development of the more rigorous methods in use today. Then we discuss dual methods, and finally we show that optimality criteria methods are closely related to dual methods.

9.1 Intuitive Optimality Criteria Methods

Optimality criteria methods consist of two complementary ingredients. The first is the stipulation of the optimality criteria, which can be rigorous mathematical statements such as the Kuhn-Tucker conditions, or an intuitive one such as the stipulation that the strain energy density in the structure is uniform. The second ingredient is the algorithm used to resize the structure for the purpose of satisfying the optimality criterion. Again, a rigorous mathematical method may be used to achieve the satisfaction of the optimality criterion, or one may devise an ad-hoc method which sometimes works and sometimes does not. The division into intuitive and rigorous methods is usually made on the basis of the chosen optimality criterion rather than of the resizing algorithm. This convention will be employed in the following sections.

9.1.1 *Fully Stressed Design*

The Fully Stressed Design (FSD) technique is probably the most successful optimality criteria method, and has motivated much of the initial interest in these methods. The FSD technique is applicable to structures that are subject only to stress and minimum gage constraints. The FSD optimality criterion can be stated as follows:

For the optimum design each member of the structure that is not at its minimum gage is fully stressed under at least one of the design load conditions.

This optimality criterion implies that we should remove material from members that are not fully stressed unless prevented by minimum gage constraints. This appears reasonable, but it is based on an implicit assumption that the primary effect of adding or removing material from a structural member is to change the stresses in that member. If this assumption is violated, that is if adding material to one part of the structure can have large effects on the stresses in other parts of the structure, we may want to have members that are not fully stressed because they help to relieve stresses in other members.

For statically determinate structures the assumption that adding material to a member influences primarily the stresses in that member is correct. In fact, without inertia or thermal loads there is no effect at all on stresses in other members. Therefore, we can expect that the FSD criterion holds at the minimum weight design for such structures, and this has been proven [1,2]. However, for statically indeterminate structures the minimum weight design may not be fully stressed [3-6]. In most cases of a structure made of a single material, there is a fully stressed design near the optimum design, and so the method has been extensively used for metal structures, especially in the aerospace industry (see for example, [7,8]). As illustrated by the following example, the FSD method may not do as well when several materials are employed.

Example 9.1.1

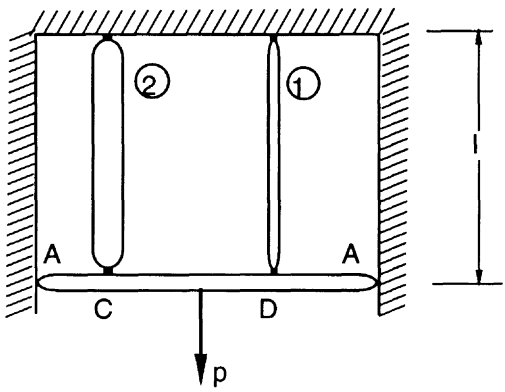


Figure 9.1.1 Two-bar structure.

A load p is transmitted by a rigid platform A–A to two axial members as shown in Figure 9.1.1. The platform is kept perfectly horizontal so that the vertical displacements of point C and of point D are identical. This may be accomplished by moving the force p horizontally as the cross sections of members 1 and 2 are changed. The two members are made from different steel alloys having the same Young's modulus E but different densities ρ_1 and ρ_2, and different yield stresses σ_{01} and σ_{02}, respectively. The alloy with higher yield stress is also more brittle, and for this reason we cannot use it in both members. We want to select the two cross-sectional areas A_1 and A_2 so as to obtain the minimum-mass design without exceeding the yield stress in either member. Additionally, the cross-sectional areas are required to be larger than a minimum gage value of A_0.

The mass, which is the objective function to be minimized, is

$$m = l(\rho_1 A_1 + \rho_2 A_2)\,.$$

The stresses in the two members (based on the assumption that the platform remains horizontal) are easily shown to be

$$\sigma_1 = \sigma_2 = \frac{p}{A_1 + A_2}\,.$$

Now assume that member one is made of a high-strength low-density alloy such that $\sigma_{01} = 2\sigma_{02}$ and $\rho_1 = 0.9\rho_2$. In this case the critical constraint is

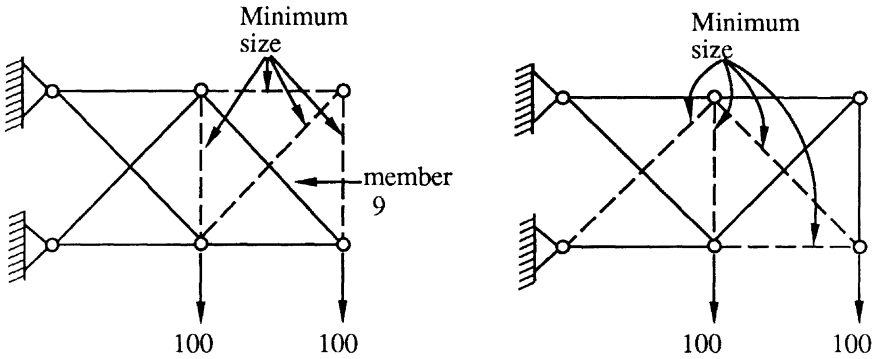
$$\sigma_2 = \sigma_{02} = \frac{p}{A_1 + A_2}\,,$$

so that $A_1 + A_2 = p/\sigma_{02}$. The minimum mass design obviously will make maximum use of the superior alloy by reducing the area of the second member to its minimum gage value $A_2 = A_0$, so that $A_1 = p/\sigma_{02} - A_0$, provided that p/σ_{02} is larger than $2A_0$. This optimum design is not fully stressed as the stress in member 1 is only half of the allowable and member 1 is not at minimum gage. The fully stressed design (obtained by the stress-ratio technique which is described below) is: $A_1 = A_0$ and $A_2 = p/\sigma_{02} - A_1$. In this design, member 2 is fully stressed and member 1 is at minimum gage. This is, of course, an absurd design because we make minimal use of the superior alloy and maximum use of the inferior one. For an illustration of the effect on mass assume that

$$p/\sigma_{02} = 20A_0 .$$

For this case the optimal design has $A_1 = 19A_0$, $A_2 = A_0$ and $m = 18.1\rho_2 A_0 l$. The fully stressed design, on the other hand, has $A_1 = A_0$, $A_2 = 19A_0$ and $m = 19.9\rho_2 A_0 l$.
• • •

Beside the use of two materials, another essential feature of Example 9.1.1 is a structure which is highly redundant, so that changing the area of one member has a large effect on the stress in the other member. This example is simple enough so that the optimum and fully-stressed designs can be found by inspection.



(a) Optimum design (b) Fully stresses design

Figure 9.1.2 Ten-bar truss.

A more complex classical example (developed by Berke and Khot [9]) often used to demonstrate the weakness of the FSD is the ten-bar truss shown in Figure 9.1.2. The truss is made of aluminum (Young's modulus $E = 10^7$psi and density $\rho = 0.1$ lb/in^3), with all members having a minimum gage of 0.1 in^2. The yield stress is ± 25000 psi for all members except member 9. Berke and Khot have shown that for $\sigma_{09} \leq 37500$ psi the optimum and FSD designs are identical, but for $\sigma_{09} \geq 37500$ psi the optimum design weighs 1497.6 lb and member 9 is neither fully stressed nor at minimum gage. The FSD design weighs 1725.2 lb, 15% heavier than the optimum, with member 9 at minimum gage. The two designs are shown in Figure 9.1.2.

The FSD technique is usually complemented by a resizing algorithm based on the assumption that the load distribution in the structure is independent of member sizes. That is, the stress in each member is calculated, and then the member is resized to bring the stresses to their allowable values assuming that the loads carried by members remained constant (this is logical since the FSD criterion is based on a similar assumption). For example, for truss structures, where the design variables are often cross-sectional areas, the force in any member is σA where σ is the axial stress and A the cross-sectional area. Assuming that σA is constant leads to the stress ratio resizing technique

$$A_{new} = A_{old}\frac{\sigma}{\sigma_0}, \tag{9.1.1}$$

which gives the resized area A_{new} in terms of the current area A_{old}, the current stress σ, and the allowable stress σ_0. For a statically determinate truss, the assumption that member forces are constant is exact, and Eq. (9.1.1) will bring the stress in each member to its allowable value. If the structure is not statically determinate Eq. (9.1.1) has to be applied repeatedly until convergence to any desired tolerance is achieved. Also, if A_{new} obtained by Eq. (9.1.1) is smaller than the minimum gage, the minimum gage is selected rather than the value given by Eq. (9.1.1). This so called *stress-ratio technique* is illustrated by the following example.

Example 9.1.2

For the structure of Example 9.1.1 we use the stress ratio formula and follow the iteration history. We assume that the initial design has $A_1 = A_2 = A_0$, and that the applied load is $p = 20A_0\sigma_{02}$. The iteration history is given in Table 9.1.1.

Table 9.1.1

Iteration	A_1/A_0	A_2/A_0	σ_1/σ_{01}	σ_2/σ_{02}
1	1.00	1.00	5.00	10.00
2	5.00	10.00	0.67	1.33
3	3.33	13.33	0.60	1.2
4	2.00	16.00	0.56	1.11
5	1.11	17.78	0.56	1.059
6	1.00	18.82	0.504	1.009
7	1.00	18.99	0.500	1.0005

Convergence is fast, and if material 2 were lighter this would be the optimum design. • • •

As can be seen from Example 9.1.2 the convergence of the stress ratio technique can be quite rapid, and this is a major attraction of the method. A more attractive feature is that it does not require derivatives of stresses with respect to design variables. When we have a structure with hundreds or thousands of members which need to be individually sized, the cost of obtaining derivatives of all critical stresses with respect to all design variables could be prohibitive. Practically all mathematical programming algorithms require such derivatives, while the stress ratio technique

does not. The FSD method is, therefore, very efficient for designing truss structures subject only to stress constraints.

For other types of structures the stress ratio technique can be generalized by pursuing the assumption that member forces are independent of member sizes. For example, in thin wall construction, where only membrane stresses are important, we would assume that $\sigma_{ij}t$ are constant, where t is the local thickness and σ_{ij} are the membrane stress components. In such situations the stress constraint is often expressed in terms of an equivalent stress σ_e as

$$\sigma_e = f(\sigma_{ij}) \leq \sigma_0 \,. \tag{9.1.2}$$

For example, in a plane-stress problem the Von-Mises stress constraint for an isotropic material is

$$\sigma_e^2 = \sigma_x^2 + \sigma_y^2 - \sigma_x\sigma_y + 3\tau_{xy}^2 \leq \sigma_0^2 \,. \tag{9.1.3}$$

In such a case the stress ratio technique becomes

$$t_{new} = t_{old}\frac{\sigma_e}{\sigma_0} \,. \tag{9.1.4}$$

In the presence of bending stresses the resizing equation is more complicated. This is the subject of Exercise 3.

When the assumption that member forces remain constant is unwarranted the stress ratio technique may converge slowly, and the FSD design may not be optimal. This may happen when the structure is highly redundant (see Adelman et al. [10], for example), or when loads depend on sizes (e.g., thermal loads or inertia loads). The method can be generalized to deal with size-dependent loads (see, for example, Adelman and Narayanaswami [11] for treatment of thermal loads), but not much can be done to resolve the problems associated with redundancy. The combination of FSD with the stress ratio technique is particularly inappropriate for designing structures made of composite materials. Because composite materials are not isotropic the FSD design may be far from optimum, and because of the redundancy inherent in the use of composite materials, convergence can be very slow.

The success of FSD prompted extensions to optimization under displacement constraints which became the basis of modern optimality criteria methods. Venkayya [12] proposed a rigorous optimality criterion based on the strain energy density in the structure. The criterion states that at the optimum design the strain energy of each element bears a constant ratio to the strain energy capacity of the element. This was the beginning of the more general optimality criteria methods discussed later.

The strain energy density criterion is rigorous under some conditions, but it has also been applied to problems where it is not the exact optimality criterion. For example, Siegel [13] used it for design subject to flutter constraints. Siegel proposed that the strain energy density associated with the flutter mode should be constant over the structure. In both [12] and [13] the optimality criterion was accompanied by a simple resizing rule similar to the stress ratio technique.

9.1.2 *Other Intuitive Methods*

The simultaneous failure mode approach was an early design technique similar to FSD in that it assumed that the lightest design is obtained when two or more modes of failure occur simultaneously. It is also assumed that the failure modes that are active at the optimum (lightest) design are known in advance.

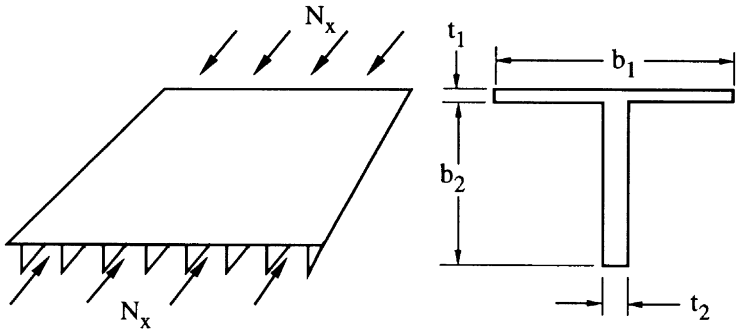


Figure 9.1.3 Metal blade-stiffened panel with four design variables.

Consider, for example (from Stroud [14]) how this procedure is used to design a metal blade-stiffened panel having the cross section shown in Figure 9.1.3. There are four design variables b_1, b_2, t_1, t_2. Rules of thumb based on considerable experience are first used to establish proportions, such as plate width-to-thickness ratios. The establishment of these proportions eliminates two of the design variables. The remaining two variables are then calculated by setting the overall buckling load and local buckling load equal to the applied load. This approach results in two equations for the two unknown design variables. The success of the method hinges on the experience and insight of the engineer who sets the proportions and identifies the resulting failure modes. For metal structures having conventional configurations, insight has been gained through many tests. Limiting the proportions accomplishes two goals: it reduces the number of design variables, and it prevents failure modes that are difficult to analyze. This simplified design approach is, therefore, compatible with simplified analysis capability.

9.2 Dual Methods

As noted in the introduction to this chapter, dual methods have been used to examine the theoretical basis of some of the popular optimality criteria methods. Historically optimality criteria methods preceded dual methods in their application to optimum structural design. However, because of their theoretical significance we will reverse the historical order and discuss dual methods first.

9.2.1 *General Formulation*

The Lagrange multipliers are often called the dual variables of the constrained optimization problem. For linear problems the primal and dual formulations have been presented in Chapter 3, and the role of dual variables as Lagrange multipliers is not difficult to establish (See Exercise 1). If the primal problem is written as

$$\begin{aligned} \textbf{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \textbf{subject to} \quad & \mathbf{A}\mathbf{x} - \mathbf{b} \geq 0\,, \\ & \mathbf{x} \geq 0\,. \end{aligned}$$

Then the dual formulation in terms of the Lagrange multipliers is

$$\begin{aligned} \textbf{maximize} \quad & \boldsymbol{\lambda}^T\mathbf{b} \\ \textbf{subject to} \quad & \boldsymbol{\lambda}^T\mathbf{A} - \mathbf{c} \leq 0\,, \\ & \boldsymbol{\lambda} \geq 0\,. \end{aligned}$$

There are several ways of generalizing the linear dual formulation to nonlinear problems. In applications to structural optimization, the most successful has been one due to Falk [15] as specialized to separable problems by Fleury [16].

The original optimization problem is called the primal problem and is of the form

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) \\ \textbf{subject to} \quad & g_j(\mathbf{x}) \geq 0\,, \qquad j = 1, \ldots, n_g\,. \end{aligned} \tag{9.2.1}$$

The necessary conditions for a local minimum of problem (9.2.1) at a point $\mathbf{x}^*$ is that there exist a vector $\boldsymbol{\lambda}^*$ (with components $\lambda_1^*, \cdots, \lambda_{n_g}^*$) such that

$$g_j(\mathbf{x}^*) \geq 0\,, \qquad j = 1, \ldots, n_g\,, \tag{9.2.2}$$

$$\lambda_j^* g_j(\mathbf{x}^*) = 0\,, \qquad j = 1, \ldots, n_g\,, \tag{9.2.3}$$

$$\lambda_j^* \geq 0\,, \qquad j = 1, \ldots, n_g\,, \tag{9.2.4}$$

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j^* \frac{\partial g_j}{\partial x_i} = 0\,, \qquad i = 1, \ldots, n\,, \tag{9.2.5}$$

Equations (9.2.3)–(9.2.5) are the Kuhn-Tucker conditions (see Chapter 5). They naturally motivate the definition of a function $\mathcal{L}$ called the Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{j=1}^{n_g} \lambda_j g_j(\mathbf{x})\,. \tag{9.2.6}$$

Equations (9.2.5) can then be viewed as stationarity conditions with respect to $\mathbf{x}$ for the Lagrangian function. Falk's dual formulation is

$$\begin{aligned} \text{maximize} \quad & \mathcal{L}_m(\boldsymbol{\lambda}) \\ \text{such that} \quad & \lambda_j \geq 0\,, \qquad j = 1, \ldots, n_g\,, \end{aligned} \tag{9.2.7}$$

where

$$\mathcal{L}_m(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in C} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}), \tag{9.2.8}$$

and where C is some closed convex set introduced to insure the well conditioning of the problem. For example, if we know that the solution is bounded, we may select C to be

$$C = \{\mathbf{x} : \ -r \leq \ x_i \leq r, \qquad i = 1, \ldots, n\}, \tag{9.2.9}$$

where r is a suitably large number. Under some restrictive conditions the solution of (9.2.7) is identical to the solution of the original problem (9.2.1), and the optimum value of $\mathcal{L}_m$ is identical to the optimum value of f. One set of conditions is for the optimization problem to be convex (that is, $f(\mathbf{x})$ bounded and convex, and $g_j(\mathbf{x})$ concave), f and g_j to be twice continuously differentiable, and the matrix of second derivatives of $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ with respect to $\mathbf{x}$ to be nonsingular at $\mathbf{x}^*$.

Under these conditions the convexity requirement also guarantees that we have only one minimum. For the linear case the Falk dual leads to the dual formulation discussed in Section 3.7 (Exercise 1).

In general, it does not make sense to solve (9.2.7), which is a nested optimization problem, instead of (9.2.1) which is a single optimization problem. However, both the maximization of (9.2.7) and the minimization of (9.2.8) are virtually unconstrained. Under some circumstances these optimizations become very simple to execute. This is the case when the objective function and the constraints are separable functions.

9.2.2 *Application to Separable Problems*

The optimization problem is called separable when both the objective function and constraints are separable, that is

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i), \tag{9.2.10}$$

$$g_j(\mathbf{x}) = \sum_{i=1}^{n} g_{ji}(x_i), \qquad j = 1, \ldots, n_g. \tag{9.2.11}$$

The primal formulation does not benefit much from the separability. However, the dual formulation does because $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ is also a separable function and can, therefore, be minimized by a series of one-dimensional minimizations, and $\mathcal{L}_m(\boldsymbol{\lambda})$ is therefore easy to calculate.

Example 9.2.1

Find the minimum of $f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2$ subject to the two constraints

$$g_1(\mathbf{x}) = x_1 + x_2 - 10 \geq 0,$$
$$g_2(\mathbf{x}) = x_2 + 2x_3 - 8 \geq 0.$$

Solution via dual method:

$$\begin{aligned}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) &= x_1^2 + x_2^2 + x_3^2 - \lambda_1(x_1 + x_2 - 10) - \lambda_2(x_2 + 2x_3 - 8) \\ &= L_0 + L_1(x_1) + L_2(x_2) + L_3(x_3)\,,\end{aligned} \tag{a}$$

where

$$\begin{aligned}L_1(x_1) &= x_1^2 - \lambda_1 x_1\,, \\ L_2(x_2) &= x_2^2 - (\lambda_1 + \lambda_2)x_2\,, \\ L_3(x_3) &= x_3^2 - 2\lambda_2 x_3\,, \\ L_0 &= 10\lambda_1 + 8\lambda_2\,.\end{aligned} \tag{b}$$

Each one of the functions L_1, L_2, L_3 can be minimized separately to get the minimum of $\mathcal{L}(\mathbf{x},\boldsymbol{\lambda})$ with respect to $\mathbf{x}$. The minimum of L_1 is found by setting its first derivative to zero

$$2x_1 - \lambda_1 = 0\,,$$

so that $x_1 = \lambda_1/2$. Similarly we obtain $x_2 = (\lambda_1 + \lambda_2)/2$ and $x_3 = \lambda_2$. Substituting back into $\mathcal{L}(\mathbf{x},\boldsymbol{\lambda})$ we get

$$\mathcal{L}_m(\boldsymbol{\lambda}) = -0.5\lambda_1^2 - 1.25\lambda_2^2 - 0.5\lambda_1\lambda_2 + 10\lambda_1 + 8\lambda_2\,.$$

We now need to find the maximum of $\mathcal{L}_m(\boldsymbol{\lambda})$ subject to the constraints $\lambda_1 \geq 0$, $\lambda_2 \geq 0$. Differentiating $\mathcal{L}_m(\boldsymbol{\lambda})$ we obtain

$$\frac{\partial \mathcal{L}_m}{\partial \lambda_1} = -\lambda_1 - 0.5\lambda_2 + 10 = 0\,,$$

$$\frac{\partial \mathcal{L}_m}{\partial \lambda_2} = -2.5\lambda_2 - 0.5\lambda_1 + 8 = 0\,,$$

or

$$\lambda_1 = 9\frac{1}{3}\,, \quad \lambda_2 = 1\frac{1}{3}\,, \qquad \mathcal{L}_m(\boldsymbol{\lambda}) = 52\,.$$

We also have to check for a maximum along the boundary $\lambda_1 = 0$ or $\lambda_2 = 0$. If $\lambda_1 = 0$

$$\mathcal{L}_m(\boldsymbol{\lambda}) = -1.25\lambda_2^2 + 8\lambda_2\,,$$

and this function attains its maximum for $\lambda_2 = 3.2$, $\mathcal{L}_m(\boldsymbol{\lambda}) = 12.8$. For $\lambda_2 = 0$ we get

$$\mathcal{L}_m(\boldsymbol{\lambda}) = -0.5\lambda_1^2 + 10\lambda_1\,,$$

with the maximum attained at $\lambda_1 = 10$, $\mathcal{L}_m = 50$. We conclude that the maximum is inside the domain. From the expressions for x_1, x_2, x_3 above we obtain

$$x_1 = 4\frac{2}{3}\,, \quad x_2 = 5\frac{1}{3}\,, \quad x_3 = 1\frac{1}{3}\,, \qquad f(\mathbf{x}) = 52\,.$$

The equality of the maximum of $\mathcal{L}_m(\boldsymbol{\lambda})$ and the minimum of $f(\mathbf{x})$ is a useful check that we obtained the correct solution. • • •

9.2.3 *Discrete Design Variables*

Because the dual method requires only one-dimensional minimization in the design variable space, it has been used for cases where some of the design variables are discrete (see, Schmit and Fleury [17]). To demonstrate this approach we suppose all the design variables are discrete. That is, the optimization problem may be written as

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i) \\ &\textbf{such that} \quad g_j(\mathbf{x}) = \sum_{i=1}^{n} g_{ji}(x_i) \geq 0\,, \qquad j = 1, \ldots, n_g\,, \\ &\textbf{and} \quad x_i \in X_i\,, \qquad i = 1, \ldots, n\,. \end{aligned} \tag{9.2.12}$$

The set $X_i = \{d_{i1}, d_{i2}, \ldots, \}$ is a set of discrete values that the ith design variable can take. The Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{n} L_i(x_i, \boldsymbol{\lambda})\,, \tag{9.2.13}$$

where

$$L_i(x_i, \boldsymbol{\lambda}) = f_i(x_i) - \sum_{j=1}^{n_g} \lambda_j g_{ji}(x_i), \qquad i = 1, \ldots, n\,. \tag{9.2.14}$$

For a given $\boldsymbol{\lambda}$ we obtain $\mathcal{L}_m(\boldsymbol{\lambda})$ by minimizing each L_i with respect to x_i by running through all the discrete values in the set X_i

$$\mathcal{L}_m(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \min_{x_i \in X_i} L_i(x_i)\,. \tag{9.2.15}$$

Note that for a given x_i, L_i is a linear function of $\boldsymbol{\lambda}$. The minimum over X_i of L_i is a piecewise linear function, with the pieces joined along lines where L_i has the same value for two different values of x_i. If the set X_i is ordered monotonically, and the discrete values of x_i are close, we can expect that these lines will be at intersections where

$$L_i(d_{ij}) = L_i(d_{i(j+1)})\,. \tag{9.2.16}$$

Equation (9.2.16) defines boundaries in $\boldsymbol{\lambda}$-space, which divide this space into regions where $\mathbf{x}$ is fixed to a particular choice of the discrete values. The use of these boundaries in the solution of the dual problem is demonstrated in the following example from Ref. [18].

Example 9.2.2

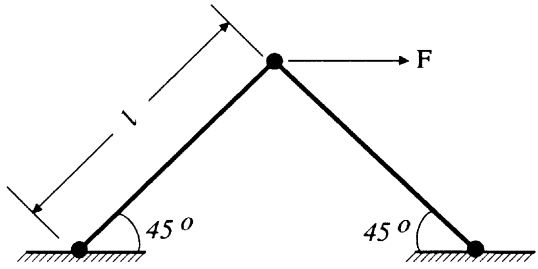


Figure 9.2.1 Two-bar truss

For the two bar truss shown in Figure 9.2.1, it is required to find the minimum weight structure by selecting each of the cross-sectional areas A_i, $i = 1, 2$, from the discrete set of areas

$$A = \{1, 1.5, 2\},$$

while at the same time satisfying the displacement constraints

$$u \leq 0.75(FL/E), \qquad v \leq 0.25(FL/E).$$

The truss is statically determinate, and the displacements are found to be

$$u = \frac{FL}{2E}\left(\frac{1}{A_1} + \frac{1}{A_2}\right), \qquad v = \frac{FL}{2E}\left(\frac{1}{A_1} - \frac{1}{A_2}\right).$$

It is convenient to use $y_i = 1/A_i$ as design variables. Denoting the weight by W, and the weight density by ρ, we can formulate the optimization problem as

$$\begin{aligned} \textbf{minimize} \quad & \frac{W}{\rho L} = \frac{1}{y_1} + \frac{1}{y_2} \\ \textbf{such that} \quad & 1.5 - y_1 - y_2 \geq 0, \\ & 0.5 - y_1 + y_2 \geq 0, \\ \textbf{and} \quad & y_1, y_2 \in \left\{\frac{1}{2}, \frac{2}{3}, 1\right\}. \end{aligned}$$

The Lagrangian function is

$$\mathcal{L}(\mathbf{y}, \boldsymbol{\lambda}) = 1/y_1 + 1/y_2 - \lambda_1(1.5 - y_2 - y_2) - \lambda_2(0.5 - y_1 + y_2),$$

and

$$\begin{aligned} L_1(y_1, \boldsymbol{\lambda}) &= -1.5\lambda_1 - 0.5\lambda_2 + 1/y_1 + (\lambda_1 + \lambda_2)y_1, \\ L_2(y_2, \boldsymbol{\lambda}) &= 1/y_2 + (\lambda_1 - \lambda_2)y_2. \end{aligned}$$

The boundaries for changes in values of y_1, from Eq. (9.2.16) are

$$\frac{1}{1/2} + \frac{1}{2}(\lambda_1 + \lambda_2) = \frac{1}{2/3} + \frac{2}{3}(\lambda_1 + \lambda_2)\,,$$
$$\frac{1}{2/3} + \frac{2}{3}(\lambda_1 + \lambda_2) = \frac{1}{1} + \lambda_1 + \lambda_2\,.$$

This yields

$$\lambda_1 + \lambda_2 = 3\,, \qquad \text{and} \quad \lambda_1 + \lambda_2 = 1.5\,.$$

Similarly, we obtain the boundaries for changes in y_2 as

$$\lambda_1 - \lambda_2 = 3\,, \qquad \text{and} \quad \lambda_1 - \lambda_2 = 1.5\,.$$

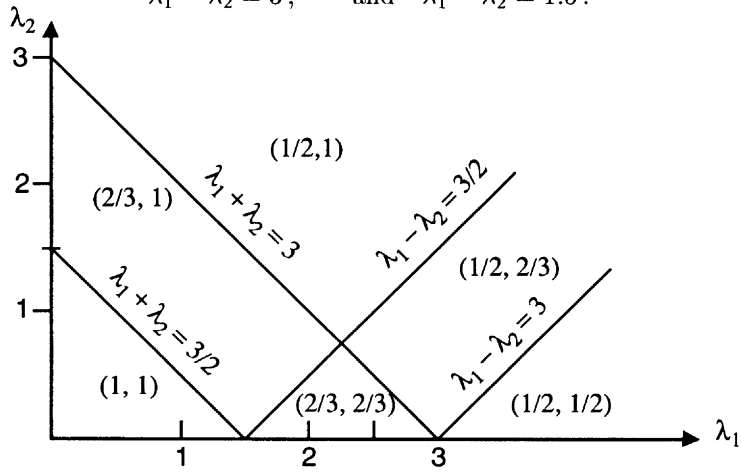


Figure 9.2.2 Regions in (λ_1, λ_2) space for two-bar truss problem

These lines divide the positive quadrant of the (λ_1, λ_2) plane into 6 regions with the values of y_1 and y_2 in each region shown in Figure (9.2.2). We start the search for the maximum of $\mathcal{L}_m$ at the origin, $\boldsymbol{\lambda} = (0, 0)$. At this point $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = 1/y_1 + 1/y_2$, so that $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ is minimized by selecting the discrete values $y_1 = y_2 = 1$, as also indicated by the figure. For the region where these values are fixed

$$\mathcal{L}_m = 2 + 0.5\lambda_1 - 0.5\lambda_2\,.$$

Obviously, to maximize $\mathcal{L}_m$ we increase λ_1 (we cannot reduce λ_2) until we get to the boundary of the region at $(1.5, 0)$ with $\mathcal{L}_m = 2.75$. We can now move into the region where the values of (y_1, y_2) are $(2/3, 1)$, or into the region where these values are $(2/3, 2/3)$. In the former region

$$\mathcal{L}_m = 2.5 + \lambda_1/6 - 5\lambda_2/6\,,$$

and in the latter region

$$\mathcal{L}_m = 3 - \lambda_1/6 - \lambda_2/2\,.$$

In either region we cannot increase $\mathcal{L}_m$. Because the maximum of $\mathcal{L}_m$ is reached at a point $(1.5, 0)$ that belongs to three regions, we have three possibilities for the values of the y's. However, only one selection $\mathbf{y}^T = (2/3, 2/3)$ does not violate the constraints. For this selection the value of the objective function is 3, which is different from the maximum of $\mathcal{L}_m$, which was 2.75. This difference, which is called the duality gap, is a reminder that the discrete problem is not convex, so that we are not sure that we obtained a minimum of the objective function. For the present example we did find one of the minima, the other being (1/2,1).• • •

To demonstrate that the procedure can occasionally provide a solution which is not the minimum we repeat Example (9.2.1) with the condition that the variables must be integers.

Example 9.2.3

The problem formulation is now

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 \\ \textbf{subject to} \quad & g_1(\mathbf{x}) = x_1 + x_2 - 10 \geq 0\,, \\ & g_2(\mathbf{x}) = x_2 + 2x_3 - 8 \geq 0\,. \\ \textbf{and} \quad & x_i \text{ are integers}, \qquad i = 1, 2, 3\,. \end{aligned}$$

The Lagrangian is given by Eqs. (a) and (b) of Problem (9.2.1), and the continuous solution was obtained as

$$\begin{aligned} & x_1 = 4\frac{2}{3}\,, \quad x_2 = 5\frac{1}{3}\,, \quad x_3 = 1\frac{1}{3}\,, \qquad f(\mathbf{x}) = 52\,. \\ & \lambda_1 = 9\frac{1}{3}\,, \quad \lambda_2 = 1\frac{1}{3}\,, \qquad \mathcal{L}_m(\boldsymbol{\lambda}) = 52\,. \end{aligned}$$

We will look for the integer solution in the neighborhood of the continuous solution. Therefore, we need the boundaries given by Eq. (9.2.16) for integer values of x_i near the continuous optimum. For x_1 we consider transition between 3 and 4, and between 4 and 5. For these transitions Eq. (9.2.16), applied to L_1, yields

$$9 - 3\lambda_1 = 16 - 4\lambda_1\,, \qquad \text{and} \quad 16 - 4\lambda_1 = 25 - 5\lambda_1\,,$$

or

$$\lambda_1 = 7\,, \quad \lambda_1 = 9\,.$$

For x_2 we consider transitions between 4 and 5 and between 5 and 6. Equation(9.2.16), applied to L_2 yields

$$\lambda_1 + \lambda_2 = 9\,, \qquad \text{and} \quad \lambda_1 + \lambda_2 = 11\,.$$

Similarly, for x_3, Eq. (9.2.16) applied to L_3 for transitions between 0 and 1 and between 1 and 2, gives

$$\lambda_2 = 0.5\,, \qquad \text{and} \quad \lambda_2 = 1.5\,.$$

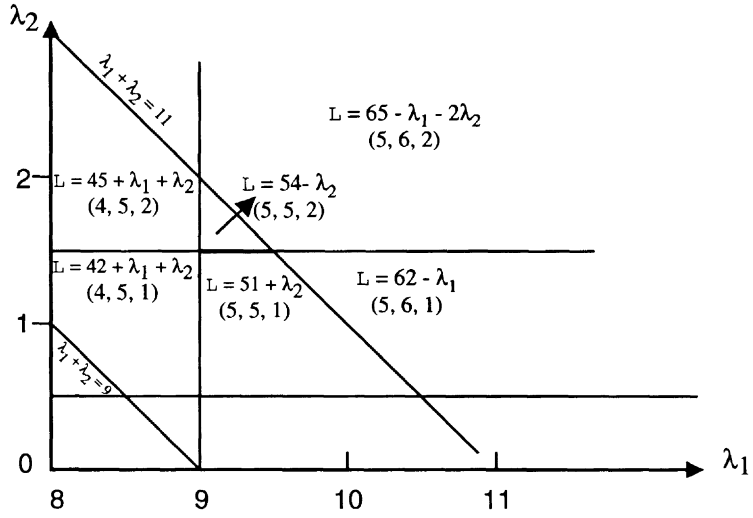


Figure 9.2.3 λ-plane for Example 9.2.3.

These boundaries, and the values of $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ in some of the regions near the continuous optimum are shown in Figure 9.2.3. We start the search for the optimum at the continuous optimum values of $\lambda_1 = 9\frac{1}{3}$, $\lambda_2 = 1\frac{1}{3}$. For this region the values of the x_i's that maximize $\mathcal{L}_m$ are (5,5,1), and $\mathcal{L}_m = 51 + \lambda_2$. This indicates that λ_2 should be increased. For $\lambda_2 = 1.5$ we reach the boundary of the region and $\mathcal{L}_m = 52.5$. That value is attained for the entire boundary of the region marked in heavy line in Figure 9.2.3. There are six adjacent regions to that boundary, and using the expressions for $\mathcal{L}_m$ given in the figure we can check that $\mathcal{L}_m = 52.5$ is the maximum. We now have six possible choices for the values of the x_i's, as indicated by the six regions that touch on the segment where $\mathcal{L}_m$ is maximal. The two leftmost regions violate the first constraint, and the three bottom regions violate the second constraint. Of the two regions that correspond to feasible designs (5,5,2) has lower objective function $f = 54$. The optimum, however, is at (4,6,1) with $f = 53$.• • •

While this example demonstrates that the method is not guaranteed to converge to the optimum, it has been found useful in many applications. In particular, the method has been applied extensively by Grierson and coworkers to the design of steel frameworks using standard sections [19–21]. The reader is directed to Ref. [18] for additional information on the implementation of automatic searches in λ-space for the maximum, and for the case of mixed discrete and continuous variables.

9.2.4 *Application with First Order Approximations*

Many of the first order approximations discussed in chapter 6 are separable. The linear and the conservative approximations are also concave, and the reciprocal approximation is concave in some cases. Therefore, if the objective function is convex and separable the dual approach is attractive for the optimization of the approximate problem. Assume, for example, that the reciprocal approximation is employed

for the constraints, and the objective function is approximated linearly. That is, the approximate optimization problem is

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = f_0 + \sum_{i=1}^{n} f_i x_i \\ \textbf{subject to} \quad & g_j(\mathbf{x}) = c_{0j} - \sum_{i=1}^{n} c_{ij}/x_i \geq 0\,, \qquad j = 1, \ldots, n_g\,, \end{aligned} \tag{9.2.17}$$

where the constants c_{ij} in (9.2.17) are calculated from the values of f and g_j and their derivatives at a point $\mathbf{x}_0$. That is

$$f_0 = f(\mathbf{x}_0) - \sum_{i=1}^{n} x_{0i} \frac{\partial f}{\partial x_i}(\mathbf{x}_0)\,, \qquad f_i = \frac{\partial f}{\partial x_i}(\mathbf{x}_0)\,, \tag{9.2.18}$$

and from Eq. (6.1.7)

$$\begin{aligned} c_{0j} &= g_j(\mathbf{x}_0) + \sum_{i=1}^{n} x_{0i} \frac{\partial g_j}{\partial x_i}(\mathbf{x}_0)\,, \\ c_{ij} &= x_{0i}^2 \frac{\partial g_j}{\partial x_i}(\mathbf{x}_0)\,. \end{aligned} \tag{9.2.19}$$

This approximate problem is convex if all the c_{ij}'s are positive. Alternatively, the problem is convex in terms of the reciprocals of the design variables if all the f_i's are positive. In either case we have a unique optimum. The Lagrangian function is now

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f_0 + \sum_{i=1}^{n} f_i x_i - \sum_{j=1}^{n_g} \lambda_j \left(c_{0j} - \sum_{i=1}^{n} c_{ij}/x_i \right)\,. \tag{9.2.20}$$

The first step in the dual method is to find $\mathcal{L}_m(\boldsymbol{\lambda})$ by minimizing $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ over $\mathbf{x}$. Differentiating $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ with respect to $\mathbf{x}$, we obtain

$$f_i - \sum_{j=1}^{n_g} \lambda_j c_{ij}/x_i^2 = 0\,, \tag{9.2.21}$$

so that

$$x_i = \left(\frac{1}{f_i} \sum_{j=1}^{n_g} \lambda_j c_{ij} \right)^{1/2}\,. \tag{9.2.22}$$

Substituting Eq. (9.2.22) into Eq. (9.2.20) we obtain

$$\mathcal{L}_m(\boldsymbol{\lambda}) = f_0 + \sum_{i=1}^{n} f_i x_i(\boldsymbol{\lambda}) - \sum_{j=1}^{n_g} \lambda_j \left(c_{0j} - \sum_{i=1}^{n} c_{ij}/x_i(\boldsymbol{\lambda}) \right)\,, \tag{9.2.23}$$

where $x_i(\boldsymbol{\lambda})$ is given by Eq. (9.2.22).

The maximization of $\mathcal{L}_m(\boldsymbol{\lambda})$ may be performed numerically, and then we need the derivatives of $\mathcal{L}_m(\boldsymbol{\lambda})$. Using Eq. (9.2.21) we find

$$\frac{\partial \mathcal{L}_m}{\partial \lambda_j} = -c_{0j} + \sum_{i=1}^{n} c_{ij}/x_i(\boldsymbol{\lambda}), \tag{9.2.24}$$

and

$$\frac{\partial^2 \mathcal{L}_m}{\partial \lambda_j \partial \lambda_k} = -\sum_{i=1}^{n} (c_{ij}/x_i^2)\frac{\partial x_i}{\partial \lambda_k}, \tag{9.2.25}$$

or, using Eq.(9.2.22)

$$\frac{\partial^2 \mathcal{L}_m}{\partial \lambda_j \partial \lambda_k} = -\frac{1}{2}\sum_{i=1}^{n} \frac{c_{ij}c_{ik}}{f_i x_i^3(\boldsymbol{\lambda})}. \tag{9.2.26}$$

With second derivatives so readily available it is reasonable to use Newton's method for the maximization.

In general, when some of the c_{ij}'s are negative, Eq. (9.2.22) may yield imaginary values. It is, therefore, safer to employ the conservative-convex approximation. This is, indeed, the more common practice when the dual method is used [22].

Example 9.2.4

The three-bar truss in Figure 9.2.4 is subjected to a combination of a horizontal and a vertical load. For this example we assume that the vertical load is twice the horizontal load, $p_H = p$, $p_V = 2p$. The horizontal load could be acting either to the right or to the left, and for this reason we are seeking a symmetric design. The truss is to be designed subject to a constraint that the displacement at the point of load application does not exceed a value d in either the horizontal or vertical directions. The design variables are the cross-sectional areas A_A, A_B and A_C of the truss members. Because of symmetry we assume that $A_A = A_C$. The objective h selected for this example is somewhat artificial, and is given as

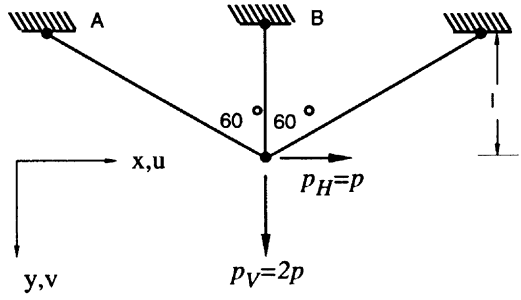


Figure 9.2.4 Three-bar truss.

$$h = A_A + 2A_B\,,$$

which is based on the assumption that the cost of member B is high. The constraints are

$$g_1 = 1 - u/d \geq 0\,,$$

$$g_2 = 1 - v/d \geq 0\,,$$

where u and v are the horizontal and vertical displacements, respectively. Assuming that all three members have the same Young's modulus E, we may check that

$$u = \frac{4pl}{3EA_A}\,,$$

$$v = \frac{2pl}{E(A_B + 0.25A_A)}\,.$$

We define a reference area $A_0 = 4pl/3Ed$ and normalized design variables $x_1 = A_A/A_0$, $x_2 = A_B/A_0$, and we may now write the optimization problem as

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = x_1 + 2x_2 \\ \textbf{subject to} \quad & g_1(\mathbf{x}) = 1 - 1/x_1 \geq 0\,, \\ & g_2(\mathbf{x}) = 1 - 1.5/(x_2 + 0.25x_1) \geq 0\,. \end{aligned}$$

We now use the reciprocal approximation for $g_2(\mathbf{x})$ about an initial design point $\mathbf{x}_0^T = (1, 1)$

$$\begin{aligned} g_2(\mathbf{x}_0) &= -0.2\,, \\ \frac{\partial g_2}{\partial x_1}(\mathbf{x}_0) &= \frac{0.375}{(x_2 + 0.25x_1)^2}\Big|_{\mathbf{x}_0} = 0.24\,, \\ \frac{\partial g_2}{\partial x_2}(\mathbf{x}_0) &= \frac{1.5}{(x_2 + 0.25x_1)^2}\Big|_{\mathbf{x}_0} = 0.96\,. \end{aligned}$$

so the reciprocal approximation g_{2R} is

$$\begin{aligned} g_{2R}(\mathbf{x}) &= -0.2 + 0.24(x_1 - 1)/x_1 + 0.96(x_2 - 1)/x_2 \\ &= 1 - 0.24/x_1 - 0.96/x_2\,. \end{aligned}$$

The approximate problem is

$$\begin{aligned} \textbf{minimize} \quad & f = x_1 + 2x_2 \\ \textbf{subject to} \quad & g_1(\mathbf{x}) = 1 - 1/x_1 \geq 0\,, \\ & g_{2R}(\mathbf{x}) = 1 - 0.24/x_1 - 0.96/x_2 \geq 0\,. \end{aligned}$$

In the notation of Eq. (9.2.17)

$$f_1 = 1, \quad f_2 = 2, \quad c_{11} = 1, \quad c_{21} = 0, \quad c_{12} = 0.24, \quad c_{22} = 0.96, \quad c_{01} = c_{02} = 1\,.$$

$\mathcal{L}_m(\boldsymbol{\lambda})$ is maximized here using Newton's method, starting with an initial guess of $\boldsymbol{\lambda}_0^T = (1,1)$. Then from Eq.(9.2.22)

$$x_1 = (1.24)^{1/2} = 1.113, \qquad x_2 = (0.48)^{1/2} = 0.693\,.$$

From Eq.(9.2.24)

$$\frac{\partial \mathcal{L}_m}{\partial \lambda_1} = -1 + \frac{1}{1.113} = -0.1015\,,$$
$$\frac{\partial \mathcal{L}_m}{\partial \lambda_2} = -1 + \frac{0.24}{1.113} + \frac{0.96}{0.693} = 0.60\,,$$

and from Eq.(9.2.26)

$$\frac{\partial^2 \mathcal{L}_m}{\partial \lambda_1^2} = -\frac{1}{2}\left(\frac{1}{1.113}\right)^3 = -0.3626\,,$$
$$\frac{\partial^2 \mathcal{L}_m}{\partial \lambda_1 \partial \lambda_2} = -\frac{1}{2}\left(\frac{0.24}{1.113^3}\right) = -0.0870\,,$$
$$\frac{\partial^2 \mathcal{L}_m}{\partial \lambda_2^2} = -\frac{1}{2}\left(\frac{0.24^2}{1.113^3} + \frac{0.96^2}{2 \times 0.693^3}\right) = -0.7132\,.$$

Using Newton's method for maximizing $\mathcal{L}_m$, we have

$$\boldsymbol{\lambda}_1 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} - \begin{bmatrix} -0.3626 & -0.0870 \\ -0.0870 & -0.7132 \end{bmatrix}^{-1} \begin{Bmatrix} -0.1015 \\ 0.60 \end{Bmatrix} = \begin{Bmatrix} 0.503 \\ 1.903 \end{Bmatrix},$$

so that

$$x_1 = (0.503 + 0.24 \times 1.903)^{1/2} = 0.980, \quad x_2 = (0.48 \times 1.903)^{1/2} = 0.956\,.$$

One additional iteration of Newton's method yields $\lambda_1 = 0.356$, $\lambda_2 = 2.05$, $x_1 = 1.02$, $x_2 = 1.17$. We can check on the convergence by noting that the two Lagrange multipliers are positive, so that we expect both constraints to be critical. Setting $g_1(\mathbf{x}) = 0$ and $g_{2R}(\mathbf{x}) = 0$, we obtain $x_1 = 1, x_2 = 1.263, f = 3.526$ as the optimum design for the approximate problem. Newton's method appears to converge quite rapidly. The optimum of the original problem can be found by setting $g_1(\mathbf{x}) = 0$ and $g_2(\mathbf{x}) = 0$ to obtain $x_1 = 1$, $x_2 = 1.25$, $f = 3.5$. • • •

Because dual methods operate in the space of Lagrange multipliers they are particularly powerful when the number of constraints is small compared to the number of design variables. The same is true for optimality criteria methods which are discussed next. These methods are indeed exceptional when we have only a single critical constraint.

9.3 Optimality Criteria Methods for a Single Constraint

Optimality criteria methods originated with the work of Prager and his co-workers (e.g., [23]) for distributed parameter systems, and in the work of Venkayya, Khot,

and Reddy ([24] for discr ,stems. They formulated optimality criteria such as the uniform energy distributi riterion discussed earlier. Later the discrete optimality criteria were generalized by Berke, Venkayya, Khot, and others (e.g., [25]–[27]) to deal with general displacement constraints. The discussion here is limited to discrete optimality criteria, and it is based to large extent on Refs [28] and [29]. The reader interested in distributed optimality criteria is referred to a textbook by Rozvany [30] who has contributed extensively to this field.

Optimality criteria methods are typically based on a rigorous optimality criterion derived from the Kuhn-Tucker conditions, and a resizing rule which is heuristic. Usually the resizing rule can be shown to be based on an assumption that the internal loads in the structure are insensitive to the resizing process. This is the same assumption that underlies the FSD approach and the accompanying stress-ratio resizing rule. This assumption turns out to be equivalent in many cases to the assumption that the reciprocal approximation is a good approximation for displacement constraints. This connection between optimality criteria methods and the reciprocal approximation is useful for a better understanding of the relationship between optimality criteria methods and mathematical programming methods, and is discussed in the next section.

9.3.1 *The Reciprocal Approximation for a Displacement Constraint*

We start by showing that for some structural design problems the assumption of constant internal loads is equivalent to the use of the reciprocal approximation for the displacements. The equations of equilibrium of the structure are written as

$$\mathbf{K}\mathbf{u} = \mathbf{f}\,, \tag{9.3.1}$$

where $\mathbf{K}$ is the stiffness matrix of the structure, $\mathbf{u}$ is the displacement vector, and $\mathbf{f}$ is the load vector.

Because the reciprocal approximation is used extensively in the following, we introduce a vector $\mathbf{y}$ of reciprocal variables, $y_i = 1/x_i$, $i = 1, \ldots, n$. The displacement constraint is written in terms of the reciprocal design variables as

$$g(\mathbf{u}, \mathbf{y}) = \bar{z} - \mathbf{z}^T\mathbf{u} \geq 0\,, \tag{9.3.2}$$

where $\bar{z}$ is a displacement allowable, and $\mathbf{z}^T\mathbf{u}$ is a linear combination of the displacement components. The reciprocal approximation is particularly appropriate for a special class of structures defined by a stiffness matrix which is a linear homogeneous function of the design variables x_i (e.g., truss structures with cross-sectional areas being design variables)

$$\mathbf{K} = \sum_{i=1}^{n} \mathbf{K}_i x_i = \sum_{i=1}^{n} \mathbf{K}_i / y_i\,. \tag{9.3.3}$$

We also assume that the load is independent of the design variables. Under the above conditions we will show that

$$g(\mathbf{u}) = \bar{z} + \sum_{i=1}^{n} y_i \frac{\partial g}{\partial y_i}\,. \tag{9.3.4}$$

That is, what appears to be a first order approximation of g is actually exact. Equation (9.3.4) does not imply that the constraint is a linear function of the design variables because $\partial g/\partial y_i$ depends on the design variables. To prove Eq. (9.3.4) we use Eq. (7.2.8) for the derivative of a constraint, replacing x by y_i. As the load vector is independent of the design variables, Eq. (7.2.8) yields

$$\frac{\partial g}{\partial y_i} = -\boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{K}}{\partial y_i} \right) \mathbf{u}\,, \tag{9.3.5}$$

where $\boldsymbol{\lambda}$ is obtained by solving Eq. (7.2.7) (Note, however, that in Eq. (7.2.7) $\mathbf{z}$ is a vector with components equal to $\partial g/\partial u_i$, while here, $\mathbf{z}$ is the negative of this vector, see Eq. (9.3.2), so we have to replace $\mathbf{z}$ by $-\mathbf{z}$ in the solution for $\boldsymbol{\lambda}$). Also from Eq. (9.3.3)

$$\frac{\partial \mathbf{K}}{\partial y_i} = -\frac{\mathbf{K}_i}{y_i^2}\,. \tag{9.3.6}$$

Using Eqs. (7.2.7), (9.3.4), (9.3.5) and (9.3.6) and the symmetry of $\mathbf{K}$ we get

$$\sum_{i=1}^{n} \frac{\partial g}{\partial y_i} y_i = \sum_{i=1}^{n} \boldsymbol{\lambda}^T \frac{\mathbf{K}_i}{y_i^2} \mathbf{u} y_i = \boldsymbol{\lambda}^T \left(\sum_{i=1}^{n} \mathbf{K}_i / y_i \right) \mathbf{u} = \boldsymbol{\lambda}^T \mathbf{K} \mathbf{u} = -\mathbf{z}^T \mathbf{u}\,. \tag{9.3.7}$$

From Eqs. (9.3.7) and (9.3.2) we can see that Eq. (9.3.4) is indeed correct.

Equation (9.3.4) motivates the use of the reciprocal approximation for displacement constraints. For statically determinate structures, under the assumptions used to prove Eq. (9.3.4), we have even a stronger result that the derivatives $\partial g/\partial y_i$ are constant, so that the reciprocal approximation is exact. We prove this assertion by showing that if internal loads in the structure are independent of the design variables then $\partial g/\partial y_i$ are constant. The internal loads in a statically determinate structure are, of course, independent of design variables which control stiffness but not geometry or loads.

We consider $\mathbf{K}_i/y_i$ in Eq.(9.3.3) to be the contribution of the part of the structure controlled by the ith design variable to the total stiffness matrix $\mathbf{K}$. The forces acting on that part of the structure are $\mathbf{f}_i$

$$\mathbf{f}_i = \frac{\mathbf{K}_i}{y_i} \mathbf{u}\,. \tag{9.3.8}$$

If the ith part of the structure is constrained against rigid body motion, the same forces will be obtained from a reduced stiffness matrix $\mathbf{K}_i'$ and a reduced displacement vector $\mathbf{u}_i'$

$$\mathbf{f}_i = \frac{\mathbf{K}_i'}{y_i} \mathbf{u}_i'\,, \tag{9.3.9}$$

where $\mathbf{K}_i'$ is obtained from $\mathbf{K}_i$ by enforcing rigid body motion constraints, and $\mathbf{u}_i'$ is obtained from $\mathbf{u}$ by removing the components of rigid body motion from the part of $\mathbf{u}$ which pertains to the ith part of the structure. Under these conditions $\mathbf{K}_i'$ is invertible, so that

$$\mathbf{u}_i' = y_i \mathbf{u}_i\,, \tag{9.3.10}$$

where

$$\mathbf{u}_i = (\mathbf{K}_i')^{-1}\mathbf{f}_i\,. \tag{9.3.11}$$

Using Eqs. (9.3.6), (9.3.8) and (9.3.9), we now write Eq. (9.3.5) as

$$\frac{\partial g}{\partial y_i} = \boldsymbol{\lambda}^T\frac{\mathbf{K}_i}{y_i^2}\mathbf{u} = \boldsymbol{\lambda}^T\mathbf{f}_i/y_i = \boldsymbol{\lambda}^T\frac{\mathbf{K}_i'}{y_i}\mathbf{u}_i\,. \tag{9.3.12}$$

The vector $\boldsymbol{\lambda}^T\mathbf{K}_i'/y_i$ is the internal force vector due to the (dummy) load $\mathbf{z}$ (see Eq. (7.2.7)), and is constant if we assume that internal forces are independent of design variables. Also $\mathbf{f}_i$ in Eq. (9.3.9) is constant, and so is $\mathbf{u}_i$ from Eq. (9.3.11). Therefore, finally, from Eq. (9.3.12) $\partial g/\partial y_i$ is constant.

We will now consider the use of optimality criteria methods for a single displacement constraint, based on the reciprocal approximation.

9.3.2 *A Single Displacement Constraint*

Because of the special properties of the reciprocal approximation for displacement constraints, we pose the optimization problem in terms of reciprocal variables as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{y}) \\ &\textbf{subject to} && g(\mathbf{y}) \geq 0\,. \end{aligned} \tag{9.3.13}$$

For this problem, the Kuhn-Tucker condition is

$$\frac{\partial f}{\partial y_i} - \lambda\frac{\partial g}{\partial y_i} = 0\,, \qquad i = 1,\ldots,n\,. \tag{9.3.14}$$

In many cases the objective function is linear or almost linear in terms of the original design variables x_i, and since $y_i = 1/x_i$, Eq. (9.3.14) is rewritten as

$$x_i^2\frac{\partial f}{\partial x_i} + \lambda\frac{\partial g}{\partial y_i} = 0\,, \tag{9.3.15}$$

so that

$$x_i = \left(-\lambda\frac{\partial g/\partial y_i}{\partial f/\partial x_i}\right)^{1/2}\,, \qquad i = 1,\ldots,n\,. \tag{9.3.16}$$

The Lagrange multiplier λ is obtained from the requirement that the constraint remains active (with a single inequality displacement constraint we can usually assume that it is active). Setting the reciprocal approximation of the constraint to zero we have

$$g_R = g(\mathbf{y}_0) + \sum_{i=1}^{n}\frac{\partial g}{\partial y_i}(y_i - y_{0i}) = c_0 + \sum_{i=1}^{n}\frac{\partial g}{\partial y_i}\frac{1}{x_i} = 0\,, \tag{9.3.17}$$

where

$$c_0 = g(\mathbf{y}_0) - \sum_{i=1}^{n} y_{0i}\frac{\partial g}{\partial y_i} = g(\mathbf{x}_0) + \sum_{i=1}^{n} x_{0i}\frac{\partial g}{\partial x_i}\,. \tag{9.3.18}$$

Substituting from Eq.(9.3.16) into Eq. (9.3.17) we obtain

$$\lambda = \left[\frac{1}{c_0}\sum_{i=1}^{n}\left(-\frac{\partial f}{\partial x_i}\frac{\partial g}{\partial y_i}\right)^{1/2}\right]^2 . \tag{9.3.19}$$

Equations (9.3.19) and (9.3.16) can now be used as an iterative resizing algorithm for the structure.

The process starts with the calculation of the displacement constraint and its derivatives, then λ is calculated from Eq. (9.3.19) and the new sizes from Eq. (9.3.16). The iteration is repeated, and if $\partial f/\partial x_i$ and $\partial g/\partial y_i$ are not too volatile the process converges.

In most practical design situations we also have lower and upper limits on design variables besides the displacement constraint, and the resizing algorithm must be modified slightly. First, Eq. (9.3.16) is supplemented by the lower and upper bounds, so that if it violates these limits the offending design variable is set at its limit. Second, the set of design variables which are at their lower or upper limits is called the passive set and denoted I_p, while the set including the rest of the variables is called the active set and denoted I_a. Equation (9.3.17) is now written as

$$c_0^* + \sum_{i\in I_a}\frac{\partial g}{\partial y_i}\frac{1}{x_i} = 0 , \tag{9.3.20}$$

where

$$c_0^* = c_0 + \sum_{i\in I_p}\frac{\partial g}{\partial y_i}\frac{1}{x_i} . \tag{9.3.21}$$

Equation (9.3.19) for λ is similarly modified to

$$\lambda = \left[\frac{1}{c_0^*}\sum_{i\in I_a}\left(-\frac{\partial f}{\partial x_i}\frac{\partial g}{\partial y_i}\right)^{1/2}\right]^2 . \tag{9.3.22}$$

The resizing process described in this section does not have step-size control. That is, Eq. (9.3.16) could possibly result in a very large change in the design variables from the initial values used to calculate the derivatives of f and g. The process can be modified to have control on the amount of change in the design variables, as discussed in the following sections. Including such control, Khot [28] showed that the optimality criterion method discussed here can be made completely equivalent to the gradient projection method applied together with the reciprocal approximation.

Example 9.3.1

We repeat example 9.2.2 with only a single displacement constraint on the vertical displacement. Using the normalized design variables, we pose the mathematical formulation of the problem as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) = x_1 + 2x_2 \\ &\textbf{subject to} && g(\mathbf{x}) = 1 - 1.5/(x_2 + 0.25x_1) \geq 0 . \end{aligned}$$

We also add minimum gage requirements that $x_1 \geq 0.5$ and $x_2 \geq 0.5$.

The derivatives required for the resizing process are

$$\frac{\partial f}{\partial x_1} = 1, \quad \frac{\partial f}{\partial x_2} = 2, \quad \frac{\partial g}{\partial y_1} = -x_1^2 \frac{\partial g}{\partial x_1} = -\frac{0.375 x_1^2}{(x_2 + 0.25 x_1)^2},$$

$$\frac{\partial g}{\partial y_2} = -x_2^2 \frac{\partial g}{\partial x_2} = -\frac{1.5 x_2^2}{(x_2 + 0.25 x_1)^2};$$

$$c_0 = g(\mathbf{y}) + \frac{\partial g}{\partial y_1} y_1 + \frac{\partial g}{\partial y_2} y_2$$

$$= 1 - \frac{1.5}{(x_2 + 0.25 x_1)} + \frac{0.375 x_1}{(x_2 + 0.25 x_1)^2} + \frac{1.5 x_2}{(x_2 + 0.25 x_1)^2} = 1.$$

We start with an initial design $\mathbf{x}_0 = (1, 1)^T$, and the iterative process is summarized in Table 9.3.1.

Table 9.3.1

						Eq.(9.3.16)	
x_1	x_2	$\partial g/\partial y_1$	$\partial g/\partial y_2$	c_0^*	λ	x_1	x_2
1.0	1.0	-0.24	-0.96	1.0	3.518	0.92	1.30
0.92	1.30	-0.136	-1.083	1.0	3.387	0.68	1.35
0.68	1.35	-0.0751	-1.183	1.0	3.284	0.496	1.39
0.50	1.39	-0.0408	-1.263	0.918	2.997	0.350	1.376
0.50	1.376	-0.0416	-1.261	0.917	2.999	0.353	1.375

The design converged fast to $x_1 = 0.5$ (lower bound) and $x_2 = 1.375$ even though the derivative of the constraint with respect to y_1 is far from constant. The large variation in the derivative with respect to y_1 is due to the fact that the three bar truss is highly redundant. This statement that one extra member constitutes high redundancy may seem curious, but what we have here is a structure with 50% more members than needed. • • •

As can be seen from the example, the optimality criteria approach to the single constraint problem works beautifully. Indeed, it is difficult to find a more suitable method for dealing with this class of problems.

9.3.3 *Generalization for Other Constraints*

The optimality criteria approach discussed in the previous section is very similar to the dual method. In particular, Eq. (9.2.22) is a special case of Eq. (9.3.16). While the derivations in the previous section were motivated by the match between displacement constraints and the reciprocal approximation, they are clearly suitable for any constraint that is reasonably approximated by the reciprocal approximation. In the present section we generalize the approach of the previous section, and demonstrate its application to more general constraints.

The optimality criterion for a single constraint

$$\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} = 0\,, \qquad i = 1, \ldots, n\,, \tag{9.3.23}$$

may be written as

$$\lambda = \frac{\partial f}{\partial x_i} \Big/ \frac{\partial g}{\partial x_i}\,, \qquad i = 1, \ldots, n\,. \tag{9.3.24}$$

The right-hand-side of Eq. (9.3.24) is a measure of the cost effectiveness of the ith design variable in affecting the constraint. The denominator measures the effect of x_i on the constraint, and the numerator measures the cost associated with it. Equation (9.3.24) tells us that at the optimum all design variables are equally cost effective in changing the constraint. Away from the optimum some design variables may be more effective than others. A reasonable resizing technique is to increase the utilization of the more effective variables and decrease that of the less effective ones. For example, in the simple case where x_i, $\partial f/\partial x_i$ and $\partial g/\partial x_i$ are all positive, a possible resizing rule is

$$x_i^{new} = x_i^{old}(\lambda e_i)^{1/\eta}\,, \tag{9.3.25}$$

where

$$e_i = (\partial g/\partial x_i)/(\partial f/\partial x_i)\,, \tag{9.3.26}$$

is the effectiveness of the ith variable and η is a step size parameter. A large value of η results in small changes to the design variables, which is appropriate for problems where derivatives change fast. A small value of η can accelerate convergence when derivatives are almost constant, but can cause divergence otherwise. To estimate the Lagrange multiplier we can require the constraint to be critical at the resized design. Using the reciprocal approximation, Eq. (9.3.17), and substituting into it x_i from Eq. (9.3.25), we get

$$\lambda = \left[\frac{1}{c_0}\sum_{i=1}^{n} x_i \frac{\partial g}{\partial x_i} e_i^{-\frac{1}{\eta}}\right]^{\eta}\,, \tag{9.3.27}$$

with c_0 obtained from Eq. (9.3.18). A resizing rule of this type is used in the FASTOP program [31] for the design of wing structures subject to a flutter constraint.

Example 9.3.2

A container with an open top needs to have a minimum volume of 125m^3. The cost of the sides of the container is $\$10/\text{m}^2$, while the ends and the bottom cost $\$15/\text{m}^2$. Find the optimum dimensions of the container.

We denote the width, length and height of the container as x_1, x_2, and x_3, respectively. The design problem can be formulated then as

$$\begin{aligned} &\textbf{minimize} \quad f = 20x_2x_3 + 30x_1x_3 + 15x_1x_2 \\ &\textbf{such that} \quad g = x_1x_2x_3 - 125 \geq 0\,. \end{aligned}$$

The e_i's for the three variables are given as

$$e_1 = x_2x_3/(30x_3 + 15x_2)\,,$$
$$e_2 = x_1x_3/(20x_3 + 15x_1)\,,$$
$$e_3 = x_1x_2/(20x_2 + 30x_1)\,.$$

We start with an initial design of a cube $x_1 = x_2 = x_3 = 5$m, $f = \$1625$, and obtain $\partial g/\partial x_1 = \partial g/\partial x_2 = \partial g/\partial x_3 = 25$, $c_0 = 375$ and $e_1 = 1/9$, $e_2 = 1/7$, $e_3 = 1/10$. Selecting $\eta = 2$ we obtain from Eq. (9.3.27) $\lambda = 8.62$, and using Eq. (9.3.25) we get

$$x_1 = 5(8.62/9)^{1/2} = 4.893\,,$$
$$x_2 = 5(8.62/7)^{1/2} = 5.549\,,$$
$$x_3 = 5(8.62/10)^{1/2} = 4.642\,.$$

For the new values of the design variables we obtain $f = 1604$, $g = 1.04$, $e_1 = 0.1158$, $e_2 = 0.1366$, $e_3 = 0.1053$, and $\lambda = 8.413$. The next iteration is then

$$x_1 = 4.893(8.413 \times 0.1158)^{1/2} = 4.829\,,$$
$$x_2 = 5.549(8.413 \times 0.1366)^{1/2} = 5.949\,,$$
$$x_3 = 4.642(8.413 \times 0.1053)^{1/2} = 4.370\,.$$

Finally, for these values of the design variables the effectivenesses are $e_1 = 0.1180$, $e_2 = 0.1320$ and $e_3 = 0.1089$, $g = 0.54$, and $f = 1584$. We see that the maximum difference between the e_i's which started at 43 percent is now 21 percent. By continuing the iterative process we find that the optimum design is $x_1 = 4.8075$, $x_2 = 7.2112$, $x_3 = 3.6056$ and $f = 1560$. At the optimum $e_1 = e_2 = e_3 = 0.120$. Even though the design variables change much from the values we obtained after two iterations, the objective function changed by less than two percent, which was expected in view of the close initial values of the e_i's. • • •

9.3.4 *Scaling-based Resizing*

As noted in the previous section, Eq. (9.3.24) indicates that at the optimum all design variables (which are not at their lower or upper bounds) are equally cost effective, and that their cost effectiveness is equal to $1/\lambda$. It is possible, therefore, to estimate λ as an average of the reciprocal of the cost effectivenesses. Venkayya [29] proposed to estimate λ as

$$\lambda = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i e_i}\,, \tag{9.3.28}$$

where the a_i represent some suitable weights (such as $\partial f/\partial x_i$). Equation (9.3.28) can then be used in conjunction with a resizing rule, such as Eq. (9.3.25).

Unfortunately, the combination of Eq. (9.3.28) with a resizing rule does not contain any mechanism for keeping the constraint active, and so the iterative process will tend to drift either into the feasible or infeasible domains. Therefore, an estimate

of λ from Eq. (9.3.28) must be accompanied by an additional step to insure that the design remains at the constraint boundary. One simple mechanism, used extensively with optimality criteria formulations is that of design variable scaling. One reason for the popularity of scaling is that for the simple case represented by Eq. (9.3.3) it is very easy to accomplish. It is easy to check from Eqs. (9.3.1) and (9.3.3) that scaling the design variable vector by a scalar α to $\alpha\mathbf{x}$ scales the displacement vector to $(1/\alpha)\mathbf{u}$. Venkayya [29] proposed the following procedure for the more general case. Consider a constraint g of the form

$$g(\mathbf{x}) = \bar{z} - z(\mathbf{x}) \geq 0\,, \tag{9.3.29}$$

where $z(\mathbf{x})$ represents some response quantity such as a displacement component, and $\bar{z}$ is an upper bound for z. If at the current design $g \neq 0$, we would like to find α so that

$$z(\alpha\mathbf{x}) = \bar{z}\,. \tag{9.3.30}$$

Approximating $z(\alpha\mathbf{x})$ linearly about x we get

$$z(\alpha\mathbf{x}) \approx z(\mathbf{x}) + \sum_{i=1}^{n} \frac{\partial z_i}{\partial x_i}(\alpha - 1)x_i = \bar{z}\,, \tag{9.3.31}$$

or

$$\alpha = 1 + \frac{\bar{z} - z}{\sum_{i=1}^{n} \frac{\partial z}{\partial x_i} x_i} = 1 - \frac{g}{\sum_{i=1}^{n} \frac{\partial g}{\partial x_i} x_i}\,. \tag{9.3.32}$$

If we use the reciprocal approximation in Eq. (9.3.31) we get instead

$$\alpha = \frac{\sum_{i=1}^{n} \frac{\partial g}{\partial x_i} x_i}{g + \sum_{i=1}^{n} \frac{\partial g}{\partial x_i} x_i}\,. \tag{9.3.33}$$

For the simple case represented by Eq. (9.3.3) and if the response quantity z is a stress or displacement component, the reciprocal scaling is exact. Furthermore, $z(\alpha\mathbf{x}) = (1/\alpha)z(\mathbf{x})$, so that Eq. (9.3.33) can be replaced by

$$\alpha = z/\bar{z} = 1 - g/\bar{z}\,. \tag{9.3.34}$$

Venkayya suggests that Eq. (9.3.32) be used when

$$\frac{1}{z}\sum_{i=1}^{n} \frac{\partial z}{\partial x_i} x_i \geq 0\,, \tag{9.3.35}$$

otherwise Eq. (9.3.33) is to be used. It can be readily checked that the scaling equations for α in terms of g are valid also for lower bound constraints of the form $z - \bar{z} \geq 0$.

In combining the resizing step, Eq. (9.3.25), with the scaling step we must consider whether we calculate new derivatives for each of these two operations. If we do, then the number of derivative calculation will increase to two per iteration. In most cases this is unnecessary. Unless the scaling step results in large changes in the design variables we can calculate the Lagrange multiplier using derivatives obtained before scaling.

Example 9.3.3

Consider again the container problem of Example (9.3.2). We will solve it again using Eq. (9.3.28) for estimating λ, and also employ scaling.

We start with the same initial design as in Example (9.3.2) of $x_1 = x_2 = x_3 = 5m$. For this design $g = 0$, so that we do not need any scaling. We have $e_1 = 1/9$, $e_2 = 1/7$, $e_3 = 1/10$, so that Eq. (9.3.28) with all the weights set to one gives us

$$\lambda = \frac{3}{1/9 + 1/7 + 1/10} = 8.475\,,$$

Then, using Eq. (9.3.25) with $\eta = 2$, we have

$$\begin{aligned} x_1 &= 5(8.475/9)^{1/2} = 4.852\,, \\ x_2 &= 5(8.475/7)^{1/2} = 5.502\,, \\ x_3 &= 5(8.475/10)^{1/2} = 4.603\,. \end{aligned}$$

For the new values of the design variables $g = -2.12$, $f = 1577$, $\partial g/\partial x_1 = 25.325$, $\partial g/\partial x_2 = 22.334$, $\partial g/\partial x_3 = 26.695$, $e_1 = 0.1148$, $e_2 = 0.1355$, $e_3 = 0.1044$. In our case $z = x_1 x_2 x_3$ and it is easy to check that Eq. (9.3.35) is satisfied, so that we use Eq. (9.3.32) for scaling.

$$\alpha = 1 - \frac{-2.12}{25.325 \times 4.852 + 22.334 \times 5.502 + 26.695 \times 4.603} = 1.00576\,.$$

Scaling the design variables we get $x_1 = 4.880$, $x_2 = 5.533$, and $x_3 = 4.630$. For these scaled variables $g = 0.015$, indicating that the scaling worked. For this scaled design $f = 1595$ which is a truer measure of improvement than the $f = 1577$ of the unscaled design, because the constraint is not violated. We next obtain

$$\lambda = \frac{3}{0.1148 + 0.1355 + 0.1044} = 8.457\,,$$

and resize to obtain

$$\begin{aligned} x_1 &= 4.880(8.457 \times 0.1148)^{1/2} = 4.808\,, \\ x_2 &= 5.533(8.457 \times 0.1355)^{1/2} = 5.923\,, \\ x_3 &= 4.628(8.457 \times 0.1044)^{1/2} = 4.351\,. \end{aligned}$$

For this design $g = -1.08$, $f = 1570$, $\partial g/\partial x_1 = 25.772$, $\partial g/\partial x_2 = 20.921$, $\partial g/\partial x_3 = 28.481$, $e_1 = 0.1175$, $e_2 = 0.1315$, $e_3 = 0.1084$. The scaling factor α is

$$\alpha = 1 - \frac{-1.08}{25.772 \times 4.808 + 20.921 \times 5.923 + 28.481 \times 4.351} = 1.0029\,.$$

Scaling the design variables we get $x_1 = 4.822$, $x_2 = 5.941$, and $x_3 = 4.364$. For these values $g = 0.018$ and $f = 1579$. Note that convergence is faster than in Example (9.3.2).

9.4 Several Constraints

9.4.1 *Reciprocal-Approximation Based Approach*

We start again by posing the optimization problem in terms of the reciprocal variables

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{y}) \\ &\textbf{subject to} && g_j(\mathbf{y}) \geq 0\,, \qquad j = 1, \ldots, n_g\,, \end{aligned} \tag{9.4.1}$$

so that the Kuhn-Tucker conditions are

$$\frac{\partial f}{\partial y_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial y_i} = 0\,, \qquad i = 1, \ldots, n\,. \tag{9.4.2}$$

As in the case of a single constraint we assume that f is approximately linear in x, so we replace the derivative with respect to y by a derivative with respect to x to get

$$x_k^2 f_k - \sum_{j=1}^{n_g} c_{kj} \lambda_j = 0\,, \tag{9.4.3}$$

where

$$f_k = \frac{\partial f}{\partial x_k}, \quad c_{kj} = -\frac{\partial g_j}{\partial y_k}, \qquad k = 1, \ldots, n\,. \tag{9.4.4}$$

This equation can be used to obtain x_k as

$$x_k = \left(\frac{1}{f_k} \sum_{j=1}^{n_g} \lambda_j c_{kj} \right)^{1/2}, \qquad k = 1, \ldots, n\,. \tag{9.4.5}$$

However, several other possibilities for using Eq. (9.4.3) have been proposed and used. One resizing rule, called the exponential rule, is based on rewriting Eq. (9.4.3) as

$$x_k^{new} = x_k \left(\frac{1}{x_k^2 f_k} \sum_{j=1}^{n_g} \lambda_j c_{kj} \right)^{1/\eta}, \qquad k = 1, \ldots, n\,, \tag{9.4.6}$$

where the old value of x_k is used on the right-hand side to produce a new estimate for x_k. A linearized form of Eq. (9.4.6) can be obtained by using the binomial expansion as

$$x_k^{new} = x_k + \Delta x_k\,, \qquad k = 1, \ldots, n\,, \tag{9.4.7}$$

where

$$\Delta x_k = \frac{1}{\eta} \left(\frac{1}{x_k^2 f_k} \sum_{j=1}^{n_g} \lambda_j c_{kj} - 1 \right) x_k\,, \qquad k = 1, \ldots, n\,. \tag{9.4.8}$$

It is clear from the form of the last two equations that η is a damping or step-size parameter. A high value of η reduces the correction to the present design, prevents oscillations, but can slow down progress towards the final design. A value of $\eta = 2$ corresponds to Eq.(9.4.5).

The main difficulty in the case of multiple constraints is the calculation of the Lagrange multipliers. It is possible to use the dual method and calculate the Lagrange multipliers using Newton's method. A second approach is to calculate them from the condition that the critical constraints remain critical, similar to Eq. (9.3.17). Assume, for example, that the n_g constraints are all critical. Then the Lagrange multipliers are found from the condition that

$$g_l(\mathbf{x}^{new}) = 0, \qquad l = 1, \ldots, n_g, \tag{9.4.9}$$

or

$$g_l(\mathbf{x}) + \sum_{k=1}^{n} \frac{\partial g_l}{\partial x_k} \Delta x_k = g_l(\mathbf{x}) + \sum_{k=1}^{n} \frac{c_{kl}}{x_k^2} \Delta x_k = 0, \quad l = 1, \ldots, n_g. \tag{9.4.10}$$

Using Eq.(9.4.8) for Δx_k we have

$$\sum_{j=1}^{n_g} \sum_{k=1}^{n} \frac{c_{kl} c_{kj}}{x_k^3 f_k} \lambda_j = \sum_{k=1}^{n} \frac{c_{kl}}{x_k} - \eta g_l(\mathbf{x}), \qquad l = 1, \ldots, n_g. \tag{9.4.11}$$

Equation (9.4.11) is a system of linear equations for $\boldsymbol{\lambda}$. Often the solution will yield negative values for some of the Lagrange multipliers which may indicate that the corresponding constraints should not be considered active. Several iterations with revised sets of active constraints may be needed before a set of positive Lagrange multipliers is found. Equation (9.4.11) may also be used to find starting values for a solution with the dual approach.

Stress constraints can be dealt with using the above approach. However, in many optimality criteria procedures they are handled instead by using the stress ratio technique. Member sizes obtained by the stress ratio technique are then used as minimum gages for the next optimality criteria iteration. The two approaches are compared in the following example.

Example 9.4.1

Find the minimum-mass design of the truss in Figure 9.4.1 subject to a limit of $d = 0.001l$ on the vertical displacement and a limit of σ_0 on the stresses. The design variables are the cross-sectional areas of the members, A_A, A_B and A_C, and because of symmetry it is required that $A_A = A_C$. All members are made from the same material having Young's modulus E, density ρ and yield stress $\sigma_0 = 0.002E$. After finding the optimum design we also want to estimate the effect of increasing the displacement allowable to $1.25d$.

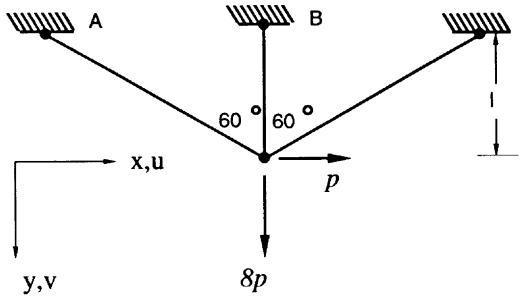


Figure 9.4.1 Three-bar truss.

The truss was analyzed in example 6.1.2, and the vertical displacement and stresses in the members were found to be

$$
\begin{aligned}
v &= \frac{8pl}{E(A_B + 0.25A_A)}, \\
\sigma_A &= p\left(\frac{\sqrt{3}}{3A_A} + \frac{2}{A_B + 0.25A_A}\right), \\
\sigma_B &= \frac{8p}{A_B + 0.25A_A}, \\
\sigma_C &= p\left(-\frac{\sqrt{3}}{3A_A} + \frac{2}{A_B + 0.25A_A}\right).
\end{aligned}
$$

The design problem may be written as

$$
\begin{aligned}
\textbf{minimize} \quad & m = \rho l(A_B + 4A_A) \\
\textbf{subject to} \quad & g_1 = 1 - \frac{v}{0.001l} \geq 0, \\
& g_2 = 1 - \frac{\sigma_A}{\sigma_0} \geq 0, \qquad g_3 = 1 - \frac{\sigma_B}{\sigma_0} \geq 0, \\
& g_4 = 1 - \frac{\sigma_C}{\sigma_0} \geq 0, \qquad g_5 = 1 + \frac{\sigma_C}{\sigma_0} \geq 0.
\end{aligned}
$$

where the second constraint on σ_C is needed because σ_C could be negative. Defining nondimensional design variables

$$x_1 = A_A\sigma_0/p, \quad \text{and} \quad x_2 = A_B\sigma_0/p,$$

we may rewrite the problem as

$$\begin{aligned}
\textbf{minimize} \quad & f(\mathbf{x}) = 4x_1 + x_2 \\
\textbf{such that} \quad & g_1(\mathbf{x}) = 1 - \frac{16}{(x_2 + 0.25x_1)} \geq 0\,, \\
& g_2(\mathbf{x}) = 1 - \frac{\sqrt{3}}{3x_1} - \frac{2}{(x_2 + 0.25x_1)} \geq 0\,, \\
& g_3(\mathbf{x}) = 1 - \frac{8}{(x_2 + 0.25x_1)} \geq 0\,, \\
& g_4(\mathbf{x}) = 1 + \frac{\sqrt{3}}{3x_1} - \frac{2}{(x_2 + 0.25x_1)} \geq 0\,, \\
& g_5(\mathbf{x}) = 1 - \frac{\sqrt{3}}{3x_1} + \frac{2}{(x_2 + 0.25x_1)} \geq 0\,.
\end{aligned}$$

Obviously, g_1 is always more critical than g_3, and g_2 is always more critical than either g_4 or g_5, so that we need consider only g_1 and g_2. We solve the problem first by using the stress ratio technique coupled with the optimality criterion for the displacement constraint.

Using the stress-ratio technique we resize the areas as

$$(A_A)_{new} = \left(\frac{\sigma_A}{\sigma_0}\right)(A_A)_{old}\,,$$

$$(A_B)_{new} = \left(\frac{\sigma_B}{\sigma_0}\right)(A_B)_{old}\,,$$

or in terms of the nondimensional variables

$$(x_1)_{new} = [1 - g_2(\mathbf{x})]x_1\,,$$

$$(x_2)_{new} = [1 - g_3(\mathbf{x})]x_2\,.$$

These values are now employed as minimum gage values for the optimality criteria method applied to g_1 only, using Eqs. (9.3.19) and (9.3.16). For the calculations we need the following derivatives

$$\begin{aligned}
\frac{\partial g_1}{\partial y_1} &= -x_1^2\frac{\partial g_1}{\partial x_1} = -\frac{4x_1^2}{(x_2 + 0.25x_1)^2}\,, \\
\frac{\partial g_1}{\partial y_2} &= -x_2^2\frac{\partial g_1}{\partial x_2} = -\frac{16x_2^2}{(x_2 + 0.25x_1)^2}\,, \\
\frac{\partial f}{\partial x_1} &= 4, \quad \frac{\partial f}{\partial x_2} = 1, \\
c_0 &= g(\mathbf{y}) - \frac{\partial g}{\partial y_1}y_1 - \frac{\partial g}{\partial y_2}y_2 \\
&= 1 - \frac{16x_2^2}{(x_2 + 0.25x_1)} + \frac{4x_1}{(x_2 + 0.25x_1)^2} + \frac{16x_2^2}{(x_2 + 0.25x_1)} = 1\,.
\end{aligned}$$

We start at $\mathbf{x}_0^T = (1, 10)$ and obtain

$$g_2 = 0.2275, \quad g_3 = 0.2195, \quad \frac{\partial g_1}{\partial y_1} = -0.03807, \quad \frac{\partial g_1}{\partial y_2} = -15.23\,.$$

Applying the stress ratio technique we get $(x_1)_{new} = 0.7725$, $(x_2)_{new} = 7.805$. Because of the large difference in the derivatives of g_1 with respect to y_1 and y_2 we expect the optimality criteria approach to try and reduce x_1 further, so that the value obtained from the stress ratio technique will end up as a minimum gage constraint. Therefore, we consider x_1 to be a passive design variable (i.e $x_1 \in I_p$). Then from Eqs.(9.3.20) and (9.3.21) we have

$$c_0^* = 1 - 0.03807 = 0.9619, \quad \lambda = \left(\frac{\sqrt{15.23}}{0.9619}\right)^2 = 16.46\,.$$

Finally from Eq. (9.3.16) we obtain $x_1 = 0.356$, $x_2 = 15.83$, confirming the assumption that x_1 is controlled by the stress constraints. The iteration is continued in Table 9.4.1.

Table 9.4.1

Iteration	x_1	x_2	$(x_1)_{new}$	$(x_2)_{new}$	c_0^*	λ	x_1	x_2
1	1.	10.	0.7725	7.805	0.9619	16.46	0.356	15.83
2	0.7725	15.83	0.6738	7.904	0.9880	16.00	0.193	15.81
3	0.6738	15.81	0.6617	7.916	0.9894	16.00	0.169	15.83

We next solve the same problem using the optimality criteria technique for both constraints. We use Eq. (9.4.11) for calculating the Lagrange multipliers, and Eq. (9.4.8) with $\eta = 2$ for updating the design variables. The iteration history is given in Table 9.4.2.

Table 9.4.2

Iteration	x_1	x_2	g_1	g_2	λ_1	λ_2	Δx_1	Δx_2
1	1.	10.	-0.5610	0.2275	11.70	0.	-0.4443	3.906
2	0.5557	13.906	-0.1392	-0.1814	15.00	2.648	0.0897	1.694
3	0.6434	15.600	-0.0152	-0.0243	15.63	2.826	0.0160	0.231

Note that Tables 9.4.1 and 9.4.2 indicate convergence to the same design, with λ in Table 9.4.1 and λ_1 in Table 9.4.2 converging to 16.00. this value is the 'price' of g_1. At the optimum design $g_1 = 0$ or $v = d$. If we increase that allowable displacement to $1.25d$, then $g_1 = 0.2$, and the expected decrease in the objective function is approximately $0.2 \times 16 = 3.2$. • • •

9.4.2 *Scaling-based Approach*

The Kuhn-Tucker conditions, Eqs. (9.2.5), can be written as

$$\sum_{j=1}^{n_g} \lambda_j e_{ij} = 1\,, \qquad i = 1, \ldots, n\,, \tag{9.4.12}$$

where

$$e_{ij} = \frac{\partial g_j}{\partial x_i} \Big/ \frac{\partial f}{\partial x_i} \qquad i = 1, \ldots, n\,, \quad j = 1, \ldots, n_g\,, \tag{9.4.13}$$

is the effectiveness parameter of the ith design variable with respect to the jth constraint. Equation (9.4.12) indicates that at the optimum the effectivenesses of all design variables, weighted by the Lagrange multipliers, are the same. This form of weighting makes sense, since the Lagrange multipliers measure the importance of the constraints in terms of their effect on the optimum value of the objective function. Venkayya [29] suggests the generalization of Eq. (9.3.25) as

$$x_i^{new} = x_i^{old} \left(\sum_{j=1}^{n_g} \lambda_j e_{ij} \right)^{1/\eta}\,, \qquad i = 1, \ldots, n\,, \tag{9.4.14}$$

for resizing the design variables. For the Lagrange multiplier evaluation he proposes using estimates based on a single constraint, that is Eq. (9.3.28), which gives

$$\lambda_j = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i e_{ij}}\,, \qquad j = 1, \ldots, n_g\,. \tag{9.4.15}$$

However, Lagrange multipliers are calculated only for the most critical constraints, and are set to zero for the other constraints. Finally, scaling is used, based on the most critical design constraint. This approach is demonstrated by repeating the previous example.

Example 9.4.2

The minimization problem that we consider is

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = 4x_1 + x_2 \\ \textbf{such that} \quad & g_1(\mathbf{x}) = 1 - \frac{16}{(x_2 + 0.25x_1)} \geq 0\,, \\ & g_2(\mathbf{x}) = 1 - \frac{\sqrt{3}}{3x_1} - \frac{2}{(x_2 + 0.25x_1)} \geq 0\,. \end{aligned}$$

We solve this problem assuming that a constraint is critical if after scaling its value is less than 0.15. Starting with $x_1 = 1$, $x_2 = 10$, we get $g_1 = -0.5610$, $g_2 = 0.2275$, so that we need to scale based on the first constraint. For this constraint we have

$$\frac{\partial g_1}{\partial x_1} = \frac{4}{(x_2 + 0.25x_1)^2} = 0.03807, \quad \frac{\partial g_1}{\partial x_2} = \frac{16}{(x_2 + 0.25x_1)^2} = 0.1523\,,$$

$$e_{11} = \frac{\partial g_1/\partial x_1}{\partial f/\partial x_1} = 0.009518, \quad e_{21} = \frac{\partial g_1/\partial x_2}{\partial f/\partial x_2} = 0.1523\,.$$

For this case $z = 1 - g$ so that the scaling test, Eq. (9.3.35) yields

$$\frac{1}{z}\sum_{i=1}^{n}\frac{\partial z}{\partial x_i}x_i = \frac{-1}{1-g}\sum_{i=1}^{n}\frac{\partial g}{\partial x_i}x_i = \frac{-1}{1.561}(0.03807 \times 1 + 0.1523 \times 10) \le 0\,.$$

Therefore we use the reciprocal scaling , Eq. (9.3.33)

$$\alpha = \frac{0.03807 \times 1 + 0.1523 \times 10}{-0.561 + 0.03807 \times 1 + 0.1523 \times 10} = 1.561\,.$$

The scaled variables are $x_1 = 1.561$, $x_2 = 15.61$. If we check the constraints we find that $g_1 = 0.$, $g_2 = 0.5051$ so that the scaling is exact. This is because the structure satisfies Eq. (9.3.3) so that we can use Eq. (9.3.34) which simplifies here to $\alpha = 1 - g$. We now estimate λ from Eq. (9.3.28) using $a_1 = a_2 = 1$ to get

$$\lambda = \frac{2}{0.009518 + 0.1523} = 12.36\,.$$

Next we resize the design variables using Eq. (9.3.24) with $\eta = 2$ to get

$$x_1 = 1.561(12.36 \times 0.009518)^{1/2} = 0.5354, \quad x_2 = 15.61(12.36 \times 0.1523)^{1/2} = 21.42\,.$$

The large change in the design variables indicates that the value of $\eta = 2$ that we used is to low, so we increase it to 4 and repeat the resizing

$$x_1 = 1.561(12.36 \times 0.009518)^{1/4} = 0.9142, \quad x_2 = 15.61(12.36 \times 0.1523)^{1/4} = 18.28\,.$$

For these new values of the design variables we have $g_1 = 0.1357$, $g_2 = 0.2604$. We expect that after scaling g_2 will be under 0.15, so that both constraints will be considered critical. Therefore we calculate derivatives for both constraints.

$$\frac{\partial g_1}{\partial x_1} = 0.01167, \quad \frac{\partial g_1}{\partial x_2} = 0.04669, \quad e_{11} = 0.00292, \quad e_{21} = 0.04669\,,$$

and

$$\frac{\partial g_2}{\partial x_1} = \frac{\sqrt{3}}{3x_1^2} + \frac{0.5}{(x_2 + 0.25x_1)^2} = 0.6923, \quad \frac{\partial g_2}{\partial x_2} = \frac{2}{(x_2 + 0.25x_1)^2} = 0.00584\,,$$
$$e_{12} = 0.1731, \quad e_{22} = 0.00584\,.$$

We first resize to obtain

$$\alpha = 1 - g_1 = 0.8643, \quad x_1 = 0.7901, \quad x_2 = 15.80\,.$$

We then calculate the Lagrange multipliers

$$\lambda_1 = 2/(0.00292 + 0.04669) = 40.32, \quad \lambda_2 = 2/(0.1731 + 0.00584) = 11.18\,,$$

and resize using Eq. (9.4.14) with $\eta = 4$ (based on the experience of the previous iteration)

$$x_1 = 0.7901(0.00292 \times 40.32 + 0.1731 \times 11.18)^{1/4} = 0.9457\,,$$
$$x_2 = 15.80(0.04669 \times 40.32 + 0.00584 \times 11.18)^{1/4} = 18.67\,.$$

The first few iterations are summarized in Table 9.4.3. The solution oscillates more than in Example 9.4.1, and seems to drift away once it gets close to the optimum of $x_1 = 0.6598$, $x_2 = 15.83$. The Lagrange multipliers are not converging to their correct values because they are based on a single-constraint approximation.• • •

Table 9.4.3

Scaled						Resized	
x_1	x_2	g_1	g_2	λ_1	λ_2	x_1	x_2
1.5610	15.61	0	0.5051	12.36	0	0.9142	18.28
0.7901	15.80	0	0.1443	40.32	11.18	0.9457	18.67
0.8004	15.80	0	0.1537	42.04	0	0.4688	18.51
0.6277	24.78	0.3584	0	0	3.017	0.7448	9.000
1.2974	15.68	0	0.4300	9.927	0	0.7598	18.36
0.6593	15.93	0.006	0	40.49	7.807	0.7910	18.77
0.6672	15.83	0	0.0096	42.34	8.453	0.8003	18.66
0.6789	15.83	0	0.0246	41.85	8.646	0.8143	18.66

9.4.3 *Other Formulations*

There are several other formulations of optimality criteria methods. These are often tailored to treat specific constraints. An example is the treatment of stability constraints by Khot in [32]. The stability eigenvalue problem is typically written as

$$[\mathbf{K} - \mu_k \mathbf{K}_G]\mathbf{u}_k = 0\,, \tag{9.4.16}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{K}_G$ is the geometric stiffness matrix, μ_k is the buckling eigenvalue, and $\mathbf{u}_k$ is the corresponding eigenvector or buckling mode. We assume that the modes are normalized so that

$$\mathbf{u}_k^T \mathbf{K}_G \mathbf{u}_k = 1\,, \tag{9.4.17}$$

and then the eigenvalue μ_k is given by

$$\mu_k = \mathbf{u}_k^T \mathbf{K} \mathbf{u}_k\,. \tag{9.4.18}$$

The constraints on eigenvalues considered in [32] are of the form

$$g_j = \mu_j - \bar{\mu}_j \geq 0\,, \qquad j = 1, \ldots, n_g\,. \tag{9.4.19}$$

The derivative of g_j with respect to a design variable x_i is obtained from Eq. (7.3.5) as

$$\frac{\partial g_j}{\partial x_i} = \frac{\partial \mu_j}{\partial x_i} = \mathbf{u}_j^T \left[\frac{\partial \mathbf{K}}{\partial x_i} - \mu_j \frac{\partial \mathbf{K}_G}{\partial x_i} \right] \mathbf{u}_j\,. \tag{9.4.20}$$

The second term of the right-hand side of Eq. (9.4.20) is zero if the prebuckling internal loads, and therefore $\mathbf{K}_G$, do not depend on the design variables. Even when the second term is not zero, there are many situations where it can be neglected. Khot defines

$$b_{ij} = x_i^2 \frac{\partial \mu_j}{\partial x_i} = x_i^2 \mathbf{u}_j^T \frac{\partial \mathbf{K}}{\partial x_i} \mathbf{u}_j\,. \tag{9.4.21}$$

If the stiffness matrix is a linear combination of the design variables

$$\mathbf{K} = \sum_{i=1}^{n} \frac{\partial \mathbf{K}}{\partial x_i} x_i\,. \tag{9.4.22}$$

then from Eqs. (9.4.18) and (9.4.21)

$$\mu_j = \sum_{i=1}^{n} \frac{b_{ij}}{x_i}, \tag{9.4.23}$$

and from Eq. (9.4.21)

$$\frac{\partial g_j}{\partial x_i} = \frac{\partial \mu_j}{\partial x_i} = \frac{b_{ij}}{x_i^2}. \tag{9.4.24}$$

Equations (9.4.23) and (9.4.24) together indicate that b_{ij} could not be approximately constant (if it were we should have a minus sign in one of these equations). However, we can still proceed in the same manner as for displacement constraints, with the optimality conditions written as

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j \frac{b_{ij}}{x_i^2} = 0, \tag{9.4.25}$$

so that

$$x_i = \left(\frac{1}{f_i} \sum_{j=1}^{n_g} \lambda_j b_{ij} \right)^{1/2}, \tag{9.4.26}$$

where $f_i = \partial f / \partial x_i$. We can use the more general form corresponding to Eq. (9.4.6)

$$x_i^{new} = x_i \left(\frac{1}{x_i^2 f_i} \sum_{j=1}^{n_g} \lambda_j b_{ij} \right)^{1/\eta}. \tag{9.4.27}$$

The calculation of the Lagrange multipliers then follows one of the methods suggested in this section. In [32] the method leading to Eq. (9.4.11) was employed. The method converged well for the truss examples in [32] even though the coefficients b_{ij} can be expected to change substantially with changes in the design.

To conclude this chapter we should note that it emphasized the relationship between optimality criteria methods, dual methods and approximation concepts. There are other treatments of optimality criteria both for specific and for general constraints. The reader is directed to Refs. [33–34] for survey of other works on optimality criteria methods.

9.5 Exercises

1. Show that for the linear case the Falk dual leads to the dual formulation discussed in Chapter 3.

2. The truss of Figure 9.2.4 is to be designed subject to stress and Euler buckling constraints for two load conditions: a horizontal load of magnitude p; and a vertical load of magnitude $2p$. The yield stress is $\sigma_0 = \alpha E$ where E is Young's modulus and

α a proportionality constant. Assume that the moment of inertia of each member is $I = \beta A^2$ where β is a constant and A the cross-sectional area. Write a program to obtain a fully-stressed design of the truss, assuming that member A and member C are identical, for various α, β, p, E, and l. What is the design for $\alpha = 10^{-3}$, $\beta = 1.0$ and $\sigma_0 l^2/p = 10^5$.

3. Obtain the FSD resizing rule for a panel of thickness t subject to in-plane loads n_x, n_y, n_{xy} and bending moments m_x, m_y, m_{xy} (all per unit length) using the Tresca (maximum shear stress) yield criterion.

4. Using the dual method find the minimum of $f = x_1 + x_2 x_3 + x_4^2$ subject to the constraint $10 - 1/x_1^2 - 2x_2 x_3 - 1/x_4 \geq 0$ and $x_i \geq 0$, $i = 1, \ldots, 4$.

5. Write a computer program to solve Example 9.2.4. Perform enough iterations to obtain the optimum design to three significant digits.

6. Repeat Example 9.2.3 when x_1 and x_2 can take only even integer values, and x_3 can vary continuously.

7. Write a program to repeat Example 9.3.1 when the design is not symmetric, so that we have three design variables. Member C is not subject to minimum gage constraints, but members A and B are.

8. Find how small we can make η in Example 9.3.2 without causing divergence of the solution.

9. Solve Example 9.4.1 with the additional constraint that the horizontal displacement does not exceed $d = 0.0005l$.

10. Complete Tables 9.4.1 and 9.4.2 for Example 9.4.1.

11. Use an optimality criteria method to design the truss of Figure 9.2.4 so that the fundamental frequency is about 1 Hertz, and the second frequency above 3 Hertz. Assume that all members have the same material properties.

9.6 References

[1] Mitchell, A.G.M., "The Limits of Economy of Material in Framed Structures," Phil. Mag., 6, pp. 589–597, 1904.

[2] Cilly, F.H., "The Exact Design of Statically Determinate Frameworks, and Exposition of its Possibility, but Futility," Trans. ASCE, 43, pp. 353–407, 1900.

[3] Schmit, L.A., "Structural Design by Systematic Synthesis," Proceedings 2nd ASCE Conference on Electronic Computation, New York, pp. 105–132, 1960.

[4] Reinschmidt, K., Cornell, C.A., and Brotchie, J.F., "Iterative Design and Structural Optimization," J. Strct. Div. ASCE, 92, ST6, pp. 281–318, 1966.

[5] Razani, R., "Behavior of Fully Stressed Design of Structures and its Relationship to Minimum Weight Design," AIAA J., 3 (12), pp. 2262–2268, 1965.

[6] Dayaratnam, P. and Patnaik, S., "Feasibility of Full Stress Design," AIAA J., 7 (4), pp. 773–774, 1969.

[7] Lansing, W., Dwyer, W., Emerton, R. and Ranalli, E., "Application of Fully-Stressed Design Procedures to Wing and Empennage Structures," J. Aircraft, 8 (9), pp. 683–688, 1971.

[8] Giles, G.L., Blackburn, C.L. and Dixon, S.C., "Automated Procedures for Sizing Aerospace Vehicle Structures (SAVES)," AIAA Paper 72-332, presented at the AIAA/ASME/SAE 13th Structures, Structural Dynamics and Materials Conference, 1972.

[9] Berke, L. and Khot, N.S., "Use of Optimality Criteria for Large Scale Systems," AGARD Lecture Series No. 170 on Structural Optimization, AGARD-LS-70, 1974.

[10] Adelman, H.M., Haftka, R.T. and Tsach, U., "Application of Fully Stressed Design Procedures to Redundant and Non-isotropic Structures," NASA TM-81842, July 1980.

[11] Adelman, H.M. and Narayanaswami, R., "Resizing procedure for structures under combined mechanical and thermal loading," AIAA J., 14 (10), pp. 1484–1486, 1976.

[12] Venkayya, V.B., "Design of Optimum Structures," Comput. Struct., 1, pp. 265–309, 1971.

[13] Siegel, S., "A Flutter Optimization Program for Aircraft Structural Design," Proc. AIAA 4th Aircraft Design, Flight Test and Operations Meeting, Los Angeles, California, 1972.

[14] Stroud, W.J., "Optimization of Composite Structures," NASA TM-84544, August 1982.

[15] Falk, J.E., "Lagrange Multipliers and Nonlinear Programming," J. Math. Anal. Appl., 19, pp. 141–159, 1967.

[16] Fleury, C., "Structural Weight Optimization by Dual Methods of Convex Programming," Int. J. Num. Meth. Engng., 14 (12), pp. 1761–1783, 1979.

[17] Schmit, L.A., and Fleury, C., "Discrete-Continuous Variable Structural Synthesis using Dual Methods," AIAA J., 18 (12), pp. 1515–1524, 1980.

[18] Schmit, L.A., and Fleury, C., "Discrete-Continuous Variable Structural Synthesis using Dual Methods," Paper 79-0721, Proceedings of the AIAA/ASME/AHS 20th Structures, Structural Dynamics and Materials Conference, St. Louis, MO, April 4–6, 1979.

[19] Grierson, D.E., and Lee, W.H., "Optimal Synthesis of Steel Frameworks Using Standard Sections," J. Struct. Mech., 12(3), pp. 335–370, 1984.

[20] Grierson, D.E., and Lee, W.H., "Optimal Synthesis of Frameworks under Elastic and Plastic Performance Constraints Using Discrete Sections," J. Struct. Mech., 14(4), pp. 401–420, 1986.

[21] Grierson, D.E., and Cameron, G.E., "Microcomputer-Based Optimization of Steel Structures in Professional Practice," Microcomputers in Civil Engineering, 4 (4), pp. 289–296, 1989.

[22] Fleury C., and Braibant, V., "Structural Optimization: A New Dual Method Using Mixed Variables," Int. J. Num. Meth. Eng., 23, pp. 409-428, 1986.

[23] Prager, W., "Optimality Criteria in Structural Design," Proc. Nat. Acad. Sci. USA, 61 (3), pp. 794–796, 1968.

[24] Venkayya, V.B, Khot, N.S., and Reddy, V.S., "Energy Distribution in an Optimum Structural Design," AFFDL-TR-68-156, 1968.

[25] Berke, L., "An Efficient Approach to the Minimum Weight Design of Deflection Limited Structures," AFFDL-TM-70-4-FDTR, 1970.

[26] Venkayya, V.B., Khot, N.S., and Berke, L., "Application of Optimality Criteria Approaches to Automated design of Large Practical Structures," Second Symposium on Structural Optimization, AGARD-CP-123, pp. 3-1 to 3-19, 1973.

[27] Gellatly, R.A, and Berke, L., "Optimality Criteria Based Algorithm," Optimum Structural Design, R.H. Gallagher and O.C., Zienkiewicz, eds., pp. 33–49, John Wiley, 1972.

[28] Khot, N.S., "Algorithms Based on Optimality Criteria to Design Minimum Weight Structures," Eng. Optim., 5, pp. 73–90, 1981.

[29] Venkayya, V.B., "Optimality Criteria: A Basis for Multidisciplinary Optimization," Computational Mechanics, Vol. 5, pp. 1–21, 1989.

[30] Rozvany, G.I.N., Structural Design via Optimality Criteria: The Prager Approach to Structural Optimization, Kluwer Academic Publishers, Dordrecht, Holland, 1989.

[31] Wilkinson, K. et al. "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles," AFFDL-TR-75-137, December 1975.

[32] Khot, N.S., "Optimal Design of a Structure for System Stability for a Specified Eigenvalue Distribution," in New Directions in Optimum Structural Design (E. Atrek, R.H., Gallagher, K.M., Ragsdell and O.C. Zienkiewicz, editors), pp. 75–87, John Wiley, 1984.

[33] Venkayya, V.B., "Structural Optimization Using Optimality Criteria: A Review and Some Recommendations," Int. J. Num. Meth. Engng., 13, pp. 203–228, 1978.

[34] Berke, L., and Khot, N.S., "Structural Optimization Using Optimality Criteria," Computer Aided Structural Design: Structural and Mechanical Systems (C.A. Mota Soares, Editor), Springer Verlag, 1987.

Decomposition and Multilevel Optimization 10

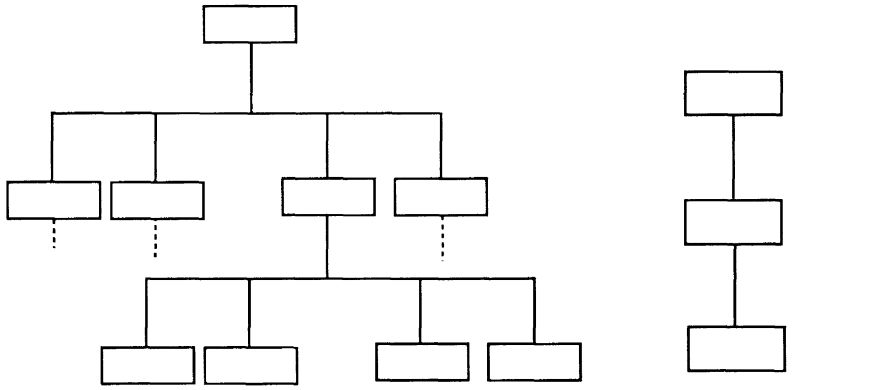
10.1 The Relation between Decomposition and Multilevel Formulation

The resources required for the solution of an optimization problem typically increase with the dimensionality of the problem at a rate which is more than linear. That is, if we double the number of design variables in a problem, the cost of solution will typically more than double. Large problems may also require excessive computer memory allocations. For these reasons we often seek ways of breaking a large optimization problem into a series of smaller problems.

One of the more popular methods for achieving such a break-up is decomposition. The process of decomposition consists of identifying relationships between design variables and constraints that permit us to separate them into groups that are only weakly interconnected. Once we have accomplished the process of decomposition we need to identify an optimization method that would take advantage of the grouping and replace the overall design with a series of optimizations of the individual groups, coordinated so as to optimize the entire system.

The coordination process is often achieved by an optimization algorithm, and then the overall optimization becomes a two-level optimization process. The coordination level is usually referred to as the top level, and the small optimization problems are called the subordinate level. Of course, it may be possible to break each one of the groups in the subordinate level to further subgroups, so that we obtain a three-level optimization, and so on. The multilevel structure generated through the process of decomposition is usually characterized by a large number of daughter subproblems in successive levels. When the decomposition process is depicted schematically (see Figure 10.1.1a), the diagram has a wide-tree (or multiple branching) structure.

2.65 Fig10.1.1E Multilevel-problem structures Multilevel optimization is not only generated through decomposition. Some problems have natural multilevel structure with only one or few daughter sublevels, that is they have a narrow-tree structure (see



(a) wide-tree structure (b) narrow-tree structure

Figure 10.1.1 Multilevel-problem structures

cases it is possible to formulate the structural analysis as an optimization process by minimizing the total potential energy of the structure. In this case the design problem can be viewed as a two-level optimization problem, analysis being a single daughter sublevel. Another example, is optimization with different types of design variables, such as sizing and shape variables, where it may be advantageous to deal with them at different levels. Finally, in multidisciplinary optimization we may have cases where it is advantageous to have sublevels corresponding to individual disciplinary optimizations coordinated at an upper level.

Because multilevel optimization techniques also have some drawbacks (discussed below), we may seek to transform some multilevel problems (especially narrow-tree problems) to a single-level structure. For example, for design problems where the analysis is performed as a second-level optimization, it may be advantageous to use a single level formulation. This single-level formulation is called simultaneous analysis and design, and is discussed along with other narrow-tree multilevel problems in Section 10.5.

10.2 Decomposition

The process of decomposition begins by the identification of groups of design variables, so that variables in each group interact closely, but interact weakly with the rest of the design variables (the strength of interaction between variables will be defined shortly). Assuming that there are s such groups, the design variable vector $\mathbf{x}$ is written as

$$\mathbf{x}^T = (\mathbf{x}_1, \ldots, \mathbf{x}_s)^T . \tag{10.2.1}$$

The groups of design variables do not interact at all when the objective function is separable in terms of the groups, that is

$$f(\mathbf{x}) = \sum_{i=1}^{s} f(\mathbf{x}_i) , \tag{10.2.2}$$

and each constraint depends only on variables from a single group. That is, if we denote the vector of constraints associated with $\mathbf{x}_i$ as $\mathbf{g}_i$, the constraints may be written as

$$\mathbf{g}_i(\mathbf{x}_i) \geq 0\,, \qquad i = 1, \ldots, s\,. \tag{10.2.3}$$

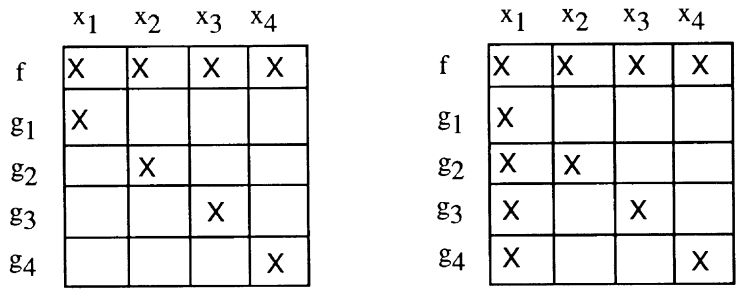


Figure 10.2.1 Block-diagonal and block-angular structures

This simple problem structure is diagrammed in Figure 10.2.1a. The rows in the diagram represent the objective function and constraints, and the columns represent the design variables. An 'x' in a block indicates that the objective function or the constraint corresponding to the row of the block depends on the vector of design variables associated with the column of that block. For a block-diagonal problem the solution naturally breaks down to a series of problems

$$\begin{aligned} &\textbf{minimize} && f_i(\mathbf{x}_i) \\ &\textbf{such that} && \mathbf{g}_i(\mathbf{x}_i) \geq 0\,, \end{aligned} \tag{10.2.4}$$

which can be solved independently for $i = 1, \ldots, s$ (that is, the problem is separable, see Section 9.2.2). This is an ideal situation because we replace the solution of the large problem with a series of smaller problems without the need for any coordination between subproblems. This is also the simplest example of problem decomposition.

It is extremely rare to encounter problems that have a simple block-diagonal structure, but in many cases we have optimization problems where the coupling between groups of variables is very weak. The coupling between groups of variables means that some of the blank off-diagonal squares in Fig. (10.2.1) fill up. A weak coupling means that the derivatives in these off-diagonal squares are small compared with the derivatives in the diagonal squares. In cases of weak coupling it may be possible to proceed as if the problem form were block diagonal. However, instead of optimizing each group of variables only once, we have to repeat the process several times to account for the weak coupling between groups. For example, consider the design of truss structures subject to stress and local buckling constraints. We can design the cross-sectional parameter of each member of the truss separately to satisfy the stress and local buckling constraints, assuming that member forces remain

constant. Of course, in a statically indeterminate truss, member forces will change, so that we will need to iterate the process. This approach is a generalization of the stress-ratio sizing technique to fully stressed design discussed in Chapter 9; it can be applied to individual members as well as to substructures (see Giles [1] and Sobieszczanski and Loendorf[2]). Furthermore, as for the stress-ratio technique, it is possible for the process to converge to a non-optimal (though usually near-optimal) design.

A more common situation is where the subproblems are interconnected through a small number of design variables. We denote the coupling design variable vector, involved in the interaction between groups, as $\mathbf{y}$. Then the minimization problem is written as

$$\begin{aligned} \textbf{minimize} \quad & f_0(\mathbf{y}) + \sum_{i=1}^{s} f_i(\mathbf{x}_i, \mathbf{y}) \\ \textbf{such that} \quad & \mathbf{g}_0(\mathbf{y}) \geq 0\,, \\ \textbf{and} \quad & \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}) \geq 0\,, \qquad i = 1, \ldots, s\,, \end{aligned} \tag{10.2.5}$$

where $\mathbf{g}_0$ is a vector of global constraints. The connectivity matrix is diagrammed in Figure 10.2.1b, and is said to have a block-angular form. The subsystem variables, $\mathbf{x}_i$, are often called local variables, while the coupling variables, $\mathbf{y}$ are called global variables. Beside the block-diagonal and block-angular problem structures there are other cases that are suited to decomposition. The reader is referred to Barthelemy [3] for a more complete discussion of problem structures which favor decomposition.

One case where the block-angular problem structure is obtained naturally is in the limit design of structures subject to several load cases (see Section 3.1). Consider a truss with r members made from a single material and subject to s load cases, given in terms of the nodal load vectors $\mathbf{p}^i$, $i = 1, \ldots, s$. The equations of equilibrium under these loads may be written as

$$\mathbf{E}\mathbf{n}^i = \mathbf{p}^i\,, \qquad i = 1, \ldots, s\,, \tag{10.2.6}$$

where $\mathbf{n}^i$ denotes the member force vector for the ith load case, and $\mathbf{E}$ is a matrix of direction cosines. For the limit design problem of the truss we need to enforce the yield constraints under each load case as

$$A_j\sigma_C \leq n_j^i \leq A_j\sigma_T\,, \qquad j = 1, \ldots, r\,, \quad i = 1, \ldots, s\,, \tag{10.2.7}$$

where σ_T, and σ_C denote the yield stress in tension and compression, respectively, A_j is the cross sectional area of the jth member, and n_j^i denotes the force in member j under the ith load case. The limit design problem for minimum weight design of the truss can then be formulated as

$$\begin{aligned} \textbf{minimize} \quad & m = \sum_{j=1}^{r} \rho A_j L_j \\ \textbf{subject to} \quad & \mathbf{E}\mathbf{n}^i = \mathbf{p}^i\,, \\ \textbf{and} \quad & A_j\sigma_C \leq n_j^i \leq A_j\sigma_T\,, \end{aligned} \tag{10.2.8}$$

where ρ and L_j denote the density and length of the jth member, respectively. In this problem the member forces and cross-sectional areas are the design variables. In this case the member forces for the ith load case, $\mathbf{n}^i$ play the role of the local variable vectors $\mathbf{x}_i$ since $\mathbf{n}^i$ appears only in the constraints associated with the ith load case. The cross-sectional areas play the role of the coupling vector $\mathbf{y}$ since they appear in the objective function and in the constraints for all load conditions.

Example 10.2.1

The three-bar truss in Figure 10.2.2 is to be designed for minimum mass so as not to collapse, under two load systems: a vertical load of magnitude $8p$ and a horizontal load of magnitude p. We assume that the truss can collapse not only due to yield, but also due to Euler buckling of the compression members. The post-buckling behavior is assumed to be flat (that is constant load with increasing deformation), so that the buckling stress can be substituted for the yield stress in Eq. (10.2.7) for members in compression. The design variables are the cross-sectional areas and moments of inertia of the members (assumed to be independent).

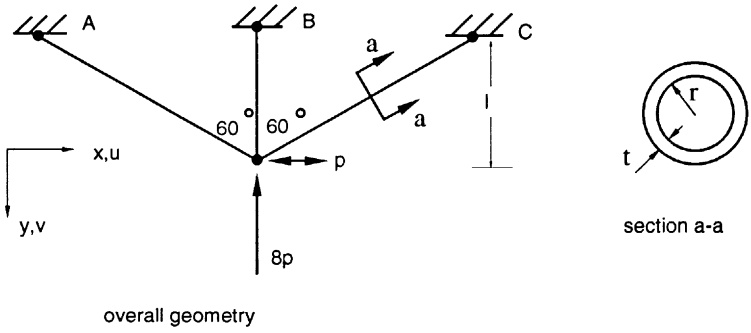


Figure 10.2.2 Three-bar tubular truss in compression

The horizontal load can act either to the right or to the left, and so we require a symmetric design, $A_A = A_C$ and $I_A = I_C$. We assume that the material properties of the members are identical, and that under the horizontal load member B will not be critical in tension. Denoting the two load cases by superscripts H and V, we

formulate the limit design problem as

$$
\begin{aligned}
\textbf{minimize} \quad & m = \rho l(4A_A + A_B) \\
\textbf{such that} \quad & 0.866(n_A^H - n_C^H) = p\,, \\
& n_B^H + 0.5(n_A^H + n_C^H) = 0\,, \\
& n_A^H \le \sigma_T A_A\,, \quad -n_B^H \le \frac{\pi^2 E I_B}{l^2}\,, \quad -n_C^H \le \frac{\pi^2 E I_A}{4l^2}\,, \\
\textbf{and} \quad & 0.866(n_A^V - n_C^V) = 0\,, \\
& n_B^V + 0.5(n_A^V + n_C^V) = -8p\,, \\
& -n_A^V \le \frac{\pi^2 E I_A}{4l^2}\,, \quad -n_B^V \le \frac{\pi^2 E I_B}{l^2}\,, \quad -n_C^V \le \frac{\pi^2 E I_A}{4l^2}\,.
\end{aligned}
$$

The block diagram for the problem is shown in Figure 10.2.3 , with a detailed variable-by-variable diagram in (a), and a variable-group diagram in (b). The diagram shows that the optimization problem has a block angular form, with the cross-sectional properties being the coupling variables, and the member forces for each load case being the local variables.• • •

A block angular form can be used in various ways, discussed later, to replace the overall optimization problem by a series of smaller problems. Aside from its value in decomposition, a block angular form also has other computational benefits. The main advantage is that derivative calculation is inexpensive because constraints depend only on a limited number of design variables. Therefore, it is worthwhile to try and induce such a block angular structure by proper choice of design variables, even if we use a standard optimization algorithm to solve the problem. This is illustrated in the following example.

Example 10.2.2

The three-bar truss in Figure 10.2.2 is now to be designed for minimum weight in the elastic range by varying the radius and the thickness of the members. The two loads are now assumed to act simultaneously, so that we consider only a single load case. Because of symmetry we assume that members A and C are identical so that the design variables are r_A, t_A, r_B, and t_B. We assume that the thicknesses of the tubes are small compared to the radii, so that the cross-sectional areas are approximated as

$$A_A = A_C = 2\pi r_A t_A, \qquad A_B = 2\pi r_B t_B\,.$$

Displacement, stress and buckling constraints are applied. The vertical displacement v is restricted to be less than $0.001l$. The stress in each member should be less than $\sigma_0 = 0.002E$, where E is Young's modulus, and σ_0 is the yield stress in tension and compression, $\sigma_0 = 10^5 p/l^2$. Additionally, the members should not buckle. This means that the stress in each member is limited to be below the shell-buckling stress of $0.605Et/r$ where r is the radius of the member and t its thickness, and the stress

(a)

		A_A	I_A	A_B	I_B	n_A^H	n_B^H	n_C^H	n_A^V	n_B^V	n_C^V
	mass	x		x							
Horizontal Load	horizontal eql.					x		x			
	vertical eql.					x	x	x			
	yielding A	x				x					
	buckling B			x	x		x				
	buckling C		x					x			
Vertical Load	horizontal eql.								x		x
	vertical eql.								x	x	x
	buckling A		x						x		
	buckling B			x	x					x	
	buckling C		x								x

(b)

	Cross Sectional Variables	n_A^H n_B^H n_C^H	n_A^V n_B^V n_C^V
mass	X		
Horizontal load constraints	X	X	
Vertical load constraints	X		X

Figure 10.2.3 Block diagram for Example 10.2.1

must also be below the Euler buckling stress of $\pi^2 Er^2/2L^2$ where L is the length of the member.

The truss was analyzed in Example 6.1.2 for a vertical tensile force, and it is easy to change the sign of that force and obtain

$$
\begin{aligned}
v &= -\frac{8pl}{E(A_B + 0.25A_A)}, \\
\sigma_A &= p\left(\frac{\sqrt{3}}{3A_A} - \frac{2}{A_B + 0.25A_A}\right), \\
\sigma_B &= -\frac{8p}{A_B + 0.25A_A}, \\
\sigma_C &= -p\left(\frac{\sqrt{3}}{3A_A} + \frac{2}{A_B + 0.25A_A}\right).
\end{aligned}
$$

We assume that the yield stress is the same in compression and in tension, and then member C will always be more critical than member A, so that the design problem may be written as

$$
\begin{aligned}
\textbf{minimize} \quad & m = \rho l(A_B + 4A_A) \\
\textbf{such that} \quad & 1 + \frac{v}{0.001l} \geq 0, \qquad 1 + \frac{\sigma_B}{\sigma_0} \geq 0, \\
& \frac{0.605Et_B}{r_B\sigma_0} + \frac{\sigma_B}{\sigma_0} \geq 0, \qquad \frac{\pi^2 E r_B^2}{2l^2\sigma_0} + \frac{\sigma_B}{\sigma_0}, \geq 0, \\
& 1 + \frac{\sigma_C}{\sigma_0} \geq 0, \qquad \frac{0.605Et_A}{r_A\sigma_0} + \frac{\sigma_C}{\sigma_0} \geq 0, \\
& \frac{\pi^2 E r_A^2}{8l^2\sigma_0} + \frac{\sigma_C}{\sigma_0} \geq 0.
\end{aligned}
$$

As posed the problem is fully coupled in that each constraint depends on all four design variables (note that the stresses in each member depend on the area and hence on the thickness and radius of the other member). However, it is simple to decouple the members and construct a block angular problem structure by changing design variables. We select the cross-sectional areas as the coupling variables ($\mathbf{y}$), and then either the radii or the thicknesses of the members can be the local or subsystem variables. In this example, let us use the two radii as the local variables. The thicknesses may then be obtained from the radius and cross-sectional areas. We define nondimensional area variables as

$$
y_1 = A_A\sigma_0/p, \quad \text{and} \quad y_2 = A_B\sigma_0/p,
$$

and then the mass, the displacement, and the stresses may be written in terms of y_1 and y_2 only. The buckling constraints also require the radii. Defining the nondimensional radii as

$$
x_1 = r_A/l, \qquad x_2 = r_B/l,
$$

we can write the buckling stress limits for member B as

$$
\frac{0.605Et_B}{r_B} = \frac{0.605EA_B}{2\pi r_B^2} = \frac{0.605}{2\pi}\frac{E}{\sigma_0}\frac{p}{\sigma_0 l^2}\frac{y_2}{x_2^2}\sigma_0 = 4.814 \times 10^{-4}\frac{y_2}{x_2^2}\sigma_0,
$$

and

$$\frac{\pi^2 E r_B^2}{2l^2} = \frac{\pi^2}{2}\frac{E}{\sigma_0}\sigma_0 x_2^2 = 2467\sigma_0 x_2^2 .$$

Using similar expressions for member C, we can now write the design problem as

$$\begin{aligned}
\textbf{minimize} \quad & m = (\rho l p/\sigma_0)(4y_1 + y_2) \\
\textbf{such that} \quad & g_1(\mathbf{y}) = 1 - \frac{16}{y_2 + 0.25y_1} \geq 0 , \qquad \text{(displacement)} \\
& g_2(\mathbf{y}) = 1 - \frac{8}{y_2 + 0.25y_1} \geq 0 , \qquad \text{(stress in B)} \\
& g_3(\mathbf{y}) = 1 - \frac{\sqrt{3}}{3y_1} - \frac{2}{y_2 + 0.25y_1} \geq 0 , \qquad \text{(stress in C)} \\
\textbf{and} \quad & g_{11}(x_1, \mathbf{y}) = 4.814 \times 10^{-4}\frac{y_1}{x_1^2} - \frac{\sqrt{3}}{3y_1} - \frac{2}{y_2 + 0.25y_1} \geq 0 , \text{ (shell buc. C)} \\
& g_{12}(x_1, \mathbf{y}) = 616.9x_1^2 - \frac{\sqrt{3}}{3y_1} - \frac{2}{y_2 + 0.25y_1} \geq 0 , \qquad \text{(Euler buckling C)} \\
& g_{21}(x_2, \mathbf{y}) = 4.814 \times 10^{-4}\frac{y_2}{x_2^2} - \frac{8}{y_2 + 0.25y_1} \geq 0 , \quad \text{(shell buckling B)} \\
& g_{22}(x_2, \mathbf{y}) = 2467x_2^2 - \frac{8}{y_2 + 0.25y_1} \geq 0 . \qquad \text{(Euler buckling B)}
\end{aligned}$$

The problem now has the requisite block angular structure.• • •

Now consider the case of a more complex truss structure composed of s tubular members, designed for minimum mass and subject to stress, displacement, and local buckling constraints. The stresses will be calculated from a finite element model. For optimization we will need the derivatives of the stresses with respect to design variables, and this derivative calculation can be the major cost in the optimization process, especially if derivatives are calculated by finite differences. If the radii and thicknesses of the members are used as design variables, then the problem is fully coupled, in that a change in each design variable may affect the stresses in all members. We will need to calculate derivatives of the stresses in the members with respect to $2s$ design variables. If, on the other hand, we use the decomposition approach employed for the three-bar truss, the cross-sectional areas and the radii are the design variables. The partial derivatives of the stresses with respect to the member radii are taken for fixed values of the corresponding areas (this is, of course, possible because the thicknesses are not specified). So these derivatives of stresses with respect to radii are zero, and we need to calculate only the s partial derivatives of stresses with respect to areas.

A similar approach may be used for frame type structures.The portal frame shown in Figure 10.2.4 , for example, was introduced by Sobieski et al. [4] for demonstrating multilevel optimization concepts. Each one of the three beams has an I cross-section defined by 6 design variables. Constraints are imposed on stresses and displacements under the loads shown in the figure. If the detail (local) design variables are used,

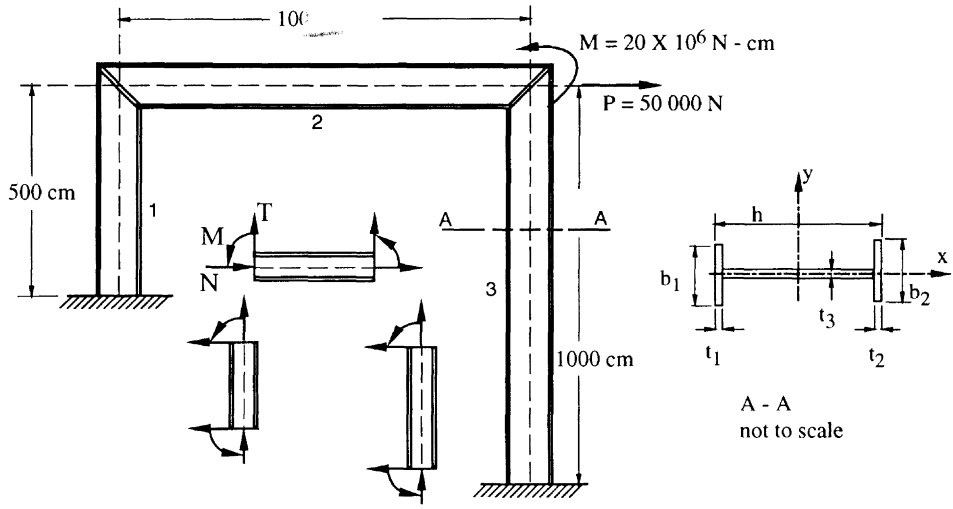


Figure 10.2.4 Decomposition of portal frame

the stress and displacement constraints are fully coupled, in that they are affected by each one of the 18 design variables. However, if we choose the cross-sectional area A and the moment of inertia I of each beam as design variables, we can eliminate 2 of the local design variables for each beam. Now all the constraints depend on the areas and moments of inertia, but the other four variables for each beam influence only the stresses in that same beam. It is possible to apply the same approach to a planar frame with s members, and have $2s$ coupling ($\mathbf{y}$) design variables, and s subsystems.

For both truss and frame problems decomposition is achieved by recognizing that the effect of one member on the rest of the structure can be expressed in terms of a small number of parameters (areas for a truss, areas and moments of inertia for a planar frame). These parameters become "global" or coupling variables, and are used to eliminate an equal number of local variables.

Thareja and Haftka [5] employed a similar approach for composite panels, using panel membrane stiffnesses as global variables. However, for more general structures, it may not be easy to select global variables that decompose the design problem.

Another difficulty associated with the decomposition is the elimination of the local variables in terms of global variables. For panel problems, as well as for complex truss and frame cross-sectional forms, it is impossible to find analytical expressions for eliminating local variables and replacing them with global variables. It is possible to keep both local and global variables, and supplement the problem with equality constraints that guarantee the consistency of the global variables with the local variables. However, this approach often tends to make the optimization problem more ill-conditioned as well as increase the number of design variables (e.g., [6]). In many cases it is possible, instead, to eliminate local design variables in terms of global ones even if analytical expressions for the elimination are not available.

Consider, for example, a generalization of the truss and frame cases where each subsystem has a set of global variables that are used to eliminate a number of the subsystem variables. For the sake of simplicity we will consider a single subsystem, and omit the subscript associated with it. That is, let $\mathbf{x}$ be the vector of subsystem variables (such as the radius and thickness for the truss tube member), and let $\mathbf{y}$ be the part of the global variable vector associated with that subsystem (such as the cross-sectional area for that truss member).

We assume that we can identify a subset of $\mathbf{x}$ that can be eliminated in terms of $\mathbf{y}$ and denote it as $\mathbf{x}_E$, and denote the rest of the local variables (to be retained) as $\mathbf{x}_R$. The relationship between $\mathbf{y}$, $\mathbf{x}_E$, and $\mathbf{x}_R$ is given as

$$\mathbf{h}(\mathbf{y}, \mathbf{x}_E, \mathbf{x}_R) = 0\,. \tag{10.2.9}$$

This relationship cannot always be solved analytically to yield an expression for $\mathbf{x}_E$ in terms of $\mathbf{y}$ and $\mathbf{x}_R$, but it can be solved numerically (e.g., Newton's method). The numerical solution for $\mathbf{x}_E$ is usually inexpensive, because Eq. (10.2.9) is a small system of algebraic equations. It is important, however, to choose $\mathbf{x}_E$ such that the system has a solution, that is the Jacobian $\partial\mathbf{h}/\partial\mathbf{x}_E$ must be nonsingular.

If we replace $\mathbf{x}$ by $\mathbf{y}$ and $\mathbf{x}_R$ as design variables without having an analytical expression for the eliminated variables, our main difficulty will be in calculating derivatives of objective function and constraints with respect to the new set of design variables. Consider, for example, a constraint function

$$g(\mathbf{x}) = g(\mathbf{x}_R, \mathbf{x}_E) = \bar{g}(\mathbf{x}_R, \mathbf{y})\,. \tag{10.2.10}$$

We need to calculate the derivatives of $\bar{g}$ without having an explicit expression for it. This is easily accommodated using implicit differentiation. Differentiating Eq. (10.2.10) we get

$$\begin{aligned} \frac{\partial \bar{g}}{\partial \mathbf{x}_R} &= \frac{\partial g}{\partial \mathbf{x}_R} + \frac{\partial g}{\partial \mathbf{x}_E}\frac{\partial \mathbf{x}_E}{\partial \mathbf{x}_R}\,, \\ \frac{\partial \bar{g}}{\partial \mathbf{y}} &= \frac{\partial g}{\partial \mathbf{x}_E}\frac{\partial \mathbf{x}_E}{\partial \mathbf{y}}\,. \end{aligned} \tag{10.2.11}$$

Note the difference between $\partial g/\partial \mathbf{x}_R$ and $\partial \bar{g}/\partial \mathbf{x}_R$. The first is a derivative of the constraint with $\mathbf{x}_E$ held constant, while the second is a derivative of the constraint with $\mathbf{y}$ held constant.

To be able to evaluate the derivatives from Eq. (10.2.11) we need the derivatives $\partial \mathbf{x}_E/\partial \mathbf{x}_R$ and $\partial \mathbf{x}_E/\partial \mathbf{y}$. These are obtained by differentiating Eq. (10.2.9) as

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{y}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}_E}\frac{\partial \mathbf{x}_E}{\partial \mathbf{y}} &= 0\,, \\ \frac{\partial \mathbf{h}}{\partial \mathbf{x}_R} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}_E}\frac{\partial \mathbf{x}_E}{\partial \mathbf{x}_R} &= 0\,, \end{aligned} \tag{10.2.12}$$

which can be solved to yield

$$\begin{aligned} \frac{\partial \mathbf{x}_E}{\partial \mathbf{y}} &= -\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}_E}\right]^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{y}}\,, \\ \frac{\partial \mathbf{x}_E}{\partial \mathbf{x}_R} &= -\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}_E}\right]^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{x}_R}\,. \end{aligned} \tag{10.2.13}$$

This process is illustrated in the following example.

Example 10.2.3

Consider again the portal frame of Figure 10.2.4. The natural global variables are the cross-sectional areas and moments of inertia. Denoting the area and moment of inertia of a typical member by A and I, respectively, and assuming that the thicknesses are much smaller than the other dimensions we have for Eq. (10.2.9)

$$\begin{aligned} h_1 &= b_1 t_1 + b_2 t_2 + H t_3 - A = 0\,, \\ h_2 &= t_3 H^3/12 + (b_1 t_1 + b_2 t_2) H^2/4 - (b_1 t_1 - b_2 t_2)^2 H^2/4A - I = 0\,. \end{aligned} \tag{a}$$

Assume that we have a local constraint which requires (say, to avoid unreasonable geometries) that the web accounts for at least 20 percent of the total area, that is

$$g = H t_3 - 0.2A = 0.8 H t_3 - 0.2(b_1 t_1 + b_2 t_2) \geq 0\,. \tag{b}$$

Assume further that we use the area and moment of inertia to eliminate the variables t_1 and t_3. That is, here t_1 and t_3 are the components of $\mathbf{x}_E$ and b_1, b_2, t_2 and H are the components of $\mathbf{x}_R$. After the elimination of the two local variables the constraint may be written as

$$\bar{g}(A, I, b_1, b_2, t_2, H) \geq 0\,.$$

We want to demonstrate that we do not need to have an explicit form for $\bar{g}$ to be able to evaluate it and its derivatives. To evaluate $\bar{g}$ for a given set of its arguments we first solve Eqs. (a) for t_1 and t_3, and then we evaluate g from (b) and note that

$$\bar{g}(A, I, b_1, b_2, t_2, H) = g(b_1, t_1, b_2, t_2, H, t_3)\,.$$

Consider now, for example, the derivative of $\bar{g}$ with respect to the area A.

$$\frac{\partial \bar{g}}{\partial A} = \frac{\partial g}{\partial t_1}\frac{\partial t_1}{\partial A} + \frac{\partial g}{\partial t_3}\frac{\partial t_3}{\partial A} = 0.8H\frac{\partial t_3}{\partial A} - 0.2 b_1 \frac{\partial t_1}{\partial A}\,.$$

We obtain $\partial t_1/\partial A$ and $\partial t_3/\partial A$ by differentiating Eqs. (a) with respect to A

$$\begin{aligned} & b_1 \frac{\partial t_1}{\partial A} + H\frac{\partial t_3}{\partial A} - 1 = 0\,, \\ & \left(\frac{b_1 H^2 - 2(b_1 t_1 - b_2 t_2)H^2/A}{4}\right)\frac{\partial t_1}{\partial A} + \frac{H^3}{12}\frac{\partial t_3}{\partial A} + \frac{(b_1 t_1 - b_2 t_2)^2 H^2}{4A^2} = 0\,. \end{aligned} \tag{c}$$

For example, consider a nominal design with $t_1 = t_2 = t_3 = t$ and $b_1 = b_2 = H$. For this initial design $A = 3Ht$, $I = 7tH^3/12$, and $g = 0.4Ht$. We start by solving Eqs. (c) for $\partial t_1/\partial A$ and $\partial t_3/\partial A$ to obtain

$$\frac{\partial t_1}{\partial A} = \frac{-1}{2H}\,, \qquad \frac{\partial t_3}{\partial A} = \frac{3}{2H}\,,$$

and then

$$\frac{\partial \bar{g}}{\partial A} = 1.3 .$$

As a check we can change the area by a small amount ΔA without changing the other arguments of $\bar{g}$. This can be accomplished by changing t_1 by $(\partial t_1/\partial A)\Delta A = -0.5\Delta A/H$ and changing t_3 by $(\partial t_3/\partial A)\Delta A = 1.5\Delta A/H$. We then check that the moment of inertia I does not change (to first order in ΔA), and that g changes by approximately $1.3\Delta A$.• • •

When it is difficult to eliminate local variables by using the global variables, we may want to use both types of variables. As noted before, the use of equality constraints to enforce compatibility between local and global variables may lead to ill-conditioning. Instead Schmit and co-workers (e.g., [7]) used the objective function of the lower-level problems as a means of enforcing compatibility. That objective function was made to be a measure of the discrepancy between the lower level and upper level variables. This approach (as well as the use of equality constraints) transfers the problem of the compatibility between lower-level and upper-level variables from the formulation or decomposition stage to the solution stage. The solution of a decomposed problem is discussed in the following sections.

10.3 Coordination and Multilevel Optimization

Once a problem has been transformed to have a block-angular form, we realize important savings in the cost of calculating sensitivity derivatives. However, it may be possible to gain additional savings by employing an optimization method which capitalizes on the special form of the problem (in particular on the smaller size of the subproblems).

A natural approach to the problem is to use a nested or two-level optimization procedure where the optimization of the subsystem variables, $\mathbf{x}_i$, is nested inside an upper-level optimization of the global variables, $\mathbf{y}$. In some cases the two levels of optimization can be uncoordinated, with the optimization process simply shuttling back and forth between the upper-level and the lower-level optimization. If changes in the global variables affect local constraints only weakly, this process can converge fast (but not necessarily to the optimum). For example, Kirsch [8] developed a three-level optimization procedure for reinforced concrete structures which relies on such an iterative procedure.

In many cases, however, the optimization process at the two levels has to be coordinated. For linear problems Dantzig and Wolfe ([9] and [10]) and Rosen [11] and [12] developed two-levels algorithms for the block-angular problem, Eq. (10.2.5). For nonlinear problems, one possible approach is known as the model-coordination method. Here we describe a version based on derivatives of the optima of subsystems with respect to upper-level variables. Consider the block-angular problem, Eq.

(10.2.5). We start by replacing it by the following two level-problem

$$\begin{aligned} &\textbf{minimize} \quad f_0(\mathbf{y}) + \sum_{i=1}^{s} f_i^*(\mathbf{y}) \\ &\textbf{such that} \quad \mathbf{g}_0(\mathbf{y}) \geq 0\,, \\ &\textbf{where} \\ &f_i^*(\mathbf{y}) = \min_{\mathbf{x}_i} \quad f_i(\mathbf{x}_i, \mathbf{y}) \\ &\textbf{such that} \quad \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}) \geq 0\,. \end{aligned} \tag{10.3.1}$$

This problem can be solved in two stages. First, an initial guess for $\mathbf{y}$ is selected, and each of the s sublevels is optimized for the corresponding $\mathbf{x}_i$. Then, the sensitivities of the optima of each sublevel with respect to changes in $\mathbf{y}$ are calculated (as described in section 5.4). Finally, these sensitivities are used to change the coupling or top-level variables ($\mathbf{y}$), in one or more iterations.

One of the difficulties associated with such a two-level approach is that for some values of $\mathbf{y}$ there may be no feasible solution to some of the sublevel problems. For linear programming, Rosen's algorithm [12] starts by finding a feasible solution. For nonlinear problems it is difficult to ensure that for a given value of the vector $\mathbf{y}$ all subproblems have feasible solutions, even though it is possible to add constraints to the upper-level problem that help prevent lower-level infeasibility (see Kirsch, [13]). Additionally, the use of sensitivities of the subsystems to changes in the top-level variables has one serious drawback: These sensitivities may not be continuous (see Barthelemy and Sobieski [14]). This is demonstrated in the following example.

Example 10.3.1

Consider again the three-bar truss of Example 10.2.1. As shown in that example, the problem has a block-angular form, with the areas and moments of inertia being the global design variables, and member forces the local variables. The upper level optimization in a two-level approach for this problem can be formulated as follows:

$$\begin{aligned} &\textbf{minimize} \quad m = \rho l(4A_A + A_B) \\ &\textbf{such that} \quad p_c^H - p \geq 0\,, \\ &\qquad\qquad\; p_c^V - p \geq 0\,, \end{aligned}$$

where p_c^H and p_c^V denote the collapse values of p for the horizontal and vertical load cases, respectively. These collapse values are obtained from the solution of two sublevel optimization problems. For the horizontal load we solve

$$\begin{aligned} &\textbf{maximize} \quad p_c^H \\ &\textbf{such that} \quad 0.866(n_A^H - n_C^H) = p_c^H\,, \\ &\qquad\qquad\; n_B^H + 0.5(n_A^H + n_C^H) = 0\,, \\ &\qquad\qquad\; n_A^H \leq \sigma_T A_A\,, \quad -n_B^H \leq \frac{\pi^2 E I_B}{l^2}\,, \quad -n_C^H \leq \frac{\pi^2 E I_A}{4l^2}\,. \end{aligned}$$

Similarly, for the vertical load we solve

$$\begin{aligned}
\textbf{maximize} \quad & p_c^V \\
\textbf{such that} \quad & 0.866(n_A^V - n_C^V) = 0\,, \\
& n_B^V + 0.5(n_A^V + n_C^V) = -8p_c^V\,, \\
& -n_A^V \le \frac{\pi^2 E I_A}{4l^2}\,, \quad -n_B^V \le \frac{\pi^2 E I_B}{l^2}\,, \quad -n_C^V \le \frac{\pi^2 E I_A}{4l^2}\,.
\end{aligned}$$

To optimize the upper-level problem we will need derivatives of the two collapse loads with respect to the cross-sectional areas and moments of inertia. We will consider only the derivatives of the horizontal collapse load p_c^H. The problem is simple enough, so that the solution for the collapse load can be found by inspection. If I_B is large enough, so that member B is not critical, then collapse will be reached when members A and C reach their maximum (yield or buckling) loads, and from the horizontal equation of equilibrium we get

$$p_c^H = 0.866\left(\sigma_T A_A + \frac{\pi^2 E I_A}{4l^2}\right)\,.$$

From the vertical equation of equilibrium we can then check that at this load member B will be indeed below its failure load if

$$I_B > I_{B0} = \frac{\sigma_T A_A l^2}{2\pi^2 E} - \frac{I_A}{8}\,.$$

If, on the other hand, $I_B < I_{B0}$ then members C and B will reach their maximum load first, and using the two equations of equilibrium we find that

$$p_c^H = \frac{0.866\pi^2 E}{l^2}(2I_B + 0.5I_A)\,.$$

It is easy to check that when $I_B = I_{B0}$ both expressions for the collapse load give identical results, so that p_c^H is a continuous function of I_B. The derivative of p_c^H with respect to I_B, on the other hand, is not continuous. When $I_B < I_{B0}$ this derivative is zero, as the collapse load is independent of the properties of member B when that member is not critical. For $I_B > I_{B0}$ we get

$$\frac{\partial p_c^H}{\partial I_B} = \frac{1.732\pi^2}{l^2}\,.$$

This discontinuity in the derivative can pose difficulties to most optimization algorithms, especially if the optimum design is in the vicinity of $I_B = I_{B0}$.• • •

10.4 Penalty and Envelope Function Approaches

One way of avoiding the difficulties of the two-level approach discussed above is to use an exterior or extended interior penalty-function method (see Section 5.7)

for the objective function at the lower levels. The penalty function approach allows us to accept upper-level ($\mathbf{y}$ variables) that do not have lower-level ($\mathbf{x}_i$ variables) feasible solutions. Indeed, the penalty associated with constraint violation at the lower levels will eventually drive the upper-level design variables away from regions with no lower-level feasible solutions. Also, the extended penalty function smoothens the discontinuities associated with the derivatives of the lower-level optima, especially when the lower-level optimization is not performed with extreme values of the penalty parameter. Finally, the use of a penalty function solves the difficulty that occurs when the lower-level variables do not contribute to the objective function.

Consider the block-angular problem described by Eq. (10.2.5). Using a penalty function approach we replace the constrained problem with

$$\textbf{minimize} \quad \Phi(\mathbf{y},\mathbf{x},r) = f_0(\mathbf{y}) + p_v[\mathbf{g}_0(\mathbf{y}),r] + \sum_{i=1}^{s} \left(f_i(\mathbf{x}_i,\mathbf{y}) + p_v[\mathbf{g}_i(\mathbf{x}_i,\mathbf{y}),r]\right) , \tag{10.4.1}$$

where p_v is the penalty associated with a vector of constraints. For example, if $\mathbf{g}$ is a constraint vector with m components we will often use a cumulative penalty function

$$p_v(\mathbf{g},r) = \sum_{j=1}^{m} p(g_j,r) , \tag{10.4.2}$$

where p denotes some penalty function such as the convenient extended interior penalty function (see Section 5.7)

$$p(g_j,r) = \begin{cases} 1/g_j & \text{for } g_j \geq g_0 , \\ 1/g_0[3 - 3(g_j/g_0) + (g_j/g_0)^2] & \text{for } g_j < g_0 . \end{cases} \tag{10.4.3}$$

The transition parameter g_0 depends on r as

$$g_0 = g_{00} r^{1/2} , \tag{10.4.4}$$

where g_{00} is a constant. The problem described by Eq. (10.4.1) is solved for a series of values of r such that $r \to 0$. A multilevel version of this formulation is

$$\begin{aligned} \textbf{minimize} \quad & f_0(\mathbf{y}) + p_v[\mathbf{g}_0(\mathbf{y},r)] + \sum_{i=1}^{s} \phi_i(\mathbf{y},r) \\ \textbf{where} \quad & \phi_i(\mathbf{y},r) = \min_{\mathbf{x}_i, r_i} \{ f_i(\mathbf{x}_i,\mathbf{y}) + p_v[\mathbf{g}_i(\mathbf{x}_i,\mathbf{y}), r_i] \} . \end{aligned} \tag{10.4.5}$$

The series of values for the subsystem penalty parameters r_i tend to zero together with the global penalty parameter r.

The method of varying the penalty parameters of the subsystems defines the particular multilevel algorithm. One attractive approach is to perform each sublevel optimization for only a single value of r_i, arguing that there is no point in striving for an exact sublevel optimum before the upper-level variables have settled close to their final values. That single value of the penalty parameter for each subsystem

can then be gradually reduced towards zero as the optimization proceeds. Reference [15] shows that when all subsystems use the same penalty parameter, the multilevel optimization is completely equivalent to the single-level approach. This means that the same series of intermediate designs are obtained on the way to the final optimum, and the calculations performed could be made to be identical. The process can be viewed as a two-level optimization, or a single-level optimization where the block-angular form is utilized to reduce the amount of computation and permit parallel operations.

Note that even when other techniques are used to solve multilevel optimization problems it is common practice to use approximate or partially converged solutions of the sub-level optimizations.

Example 10.4.1

Consider the two-level formulation of the elastic design of the three-bar truss in Example 10.2.2. For this simple example it is convenient to use a vector penalty function p_v which is equal to the penalty associated with the most critical constraint.

$$p_v(\mathbf{g}, r) = p[\min_i(g_i), r] .$$

In general this penalty approach may create discontinuity problems when the most critical constraint changes identity. For our problem, though, this does not happen. The penalty function formulation is then

$$\begin{aligned} \textbf{minimize} \quad & \phi = m(\mathbf{y}) + p_v[g_1(\mathbf{y}), g_2(\mathbf{y}), g_3(\mathbf{y}), r] + \phi_1(\mathbf{y}, r) + \phi_2(\mathbf{y}, r) \\ \textbf{where} \quad & \phi_1(\mathbf{y}, r) = \min_{x_1} p_v[g_{11}(x_1, \mathbf{y}), g_{12}(x_1, \mathbf{y}), r] , \\ & \phi_2(\mathbf{y}, r) = \min_{x_2} p_v[g_{21}(x_2, \mathbf{y}), g_{22}(x_2, \mathbf{y}), r] , \end{aligned} \tag{10.4.6}$$

where the mass and constraint functions are given in Example 10.2.2. Note that the local variables x_1 and x_2 do not contribute to the mass, so that the formulation of Eq. (10.3.1) would not have any objective function at the lower level, and the lower level problems would only require finding a feasible solution.

With this penalty function formulation, the lower-level objectives ϕ_1 and ϕ_2 each contain the contributions of two constraints. Because the penalty is based on the most critical constraint the lower-level optimum occurs when these two constraints are equally critical. For the first subsystem we get $g_{11} = g_{12}$ which yields

$$x_1 = 0.02972 y_1^{1/4} .$$

For the second subsystem we get $g_{21} = g_{22}$ or

$$x_2 = 0.02102 y_2^{1/4} .$$

With these relationships we can now solve the upper level problem as a single-level optimization problem.• • •

Instead of a penalty function it is possible to use an envelope function which replaces a vector of constraints with a single envelope constraint. Sobieski and coworkers have made extensive use of the Kresselmeier-Steinhauser (KS) envelope constraint (see Chapter 5) for multilevel formulations (e.g., Sobieski et al. [16]). The KS envelope constraint replaces the constraint vector $\mathbf{g}$ by $KS(\mathbf{g})$ where

$$KS(\mathbf{g}) = -g_{min} - (1/\rho)log\left[\sum_i exp[\rho(g_{min} - g_i)]\right],$$

where g_i are the components of $\mathbf{g}$, ρ is a factor that plays the same role as the penalty parameter and g_{min} is the most critical constraint. It is easy to show that

$$g_{min} \geq KS(\mathbf{g}) \geq g_{min} - (1/\rho)log(m)\,, \tag{10.4.7}$$

where m is the number of constraints (components of $\mathbf{g}$). As ρ is increased $KS(\mathbf{g})$ approaches the value of g_{min}. The negative of KS can be used instead of p_v in the penalty formulation.

10.5 Narrow-Tree Multilevel Problems

While in many cases the objective of decomposition is to produce a problem with many daughter sublevels (Figure 10.1.1a), there are many cases where we have a narrow-tree structure with one or few daughter sublevels (Figure 10.1.1b). In some cases there is an advantage to pursue the solution using multilevel optimization. However, in other cases it may be better to convert the multilevel problem into a single-level one.

10.5.1 *Simultaneous Analysis and Design*

Interest in converting a two-level optimization problem into a single-level problem has been particularly evident in the area of simultaneous structural analysis and design. The simultaneous analysis and design (SAND) approach seeks to change the nested approach typical of traditional structural optimization. In the nested approach the structure is analyzed for a trial design, the sensitivity of the response with respect to structural sizes is then calculated, and the sizes are modified based on these sensitivities to obtain the next trial design. The structural analysis is nested inside the optimization procedure, repeated again and again for a sequence of trial designs. The SAND approach seeks to perform the analysis and design as a single problem with response variables added to structural sizes as unknowns to be treated all in a similar way.

The two-level form of the traditional nested approach is evident in problems where the structural analysis can be formulated as an optimization problem. For example, limit design of structures can be formulated as weight minimization subject to constraints on the collapse loads. These collapse loads are the solution of a maximization problem. In Example 10.3.1 we saw the two-level format of a limit design problem

that was formulated as a single-level problem in Example 10.2.1. The single-level formulation had cross-sectional areas (structural sizes) and member forces (structural response) as design variables.

In the case of limit design the single-level formulation, that is the SAND approach, is the method of choice in engineering practice. However, in the elastic range the nested approach is the rule. The problem of minimum-weight design subject to displacement and stress constraints in the elastic range can be formulated as

$$\begin{aligned} &\textbf{minimize} \quad W(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{u},\mathbf{x}) \geq 0\,, \quad j = 1,\ldots,m\,, \end{aligned} \tag{10.5.1}$$

where the displacement field $\mathbf{u}$ can be obtained as the solution to the minimization of the potential energy U given in term of the stiffness matrix $\mathbf{K}$ and the load vector $\mathbf{f}$

$$\textbf{minimize} \quad U = (1/2)\mathbf{u}^T\mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{u}^T\mathbf{f}\,. \tag{10.5.2}$$

The common approach is to solve this problem as a two-level optimization since the solution to the energy minimization problem is obtained simply by solving the equations of equilibrium $\mathbf{Ku} = \mathbf{f}(\mathbf{x})$.

The SAND approach of using the equations of equilibrium as equality constraints and treating both structural sizes and displacements as design variables was attempted in the 1960's by Fox and Schmit [17] using a conjugate gradient (CG) technique for the optimization. However, the CG method could not deal effectively with the equality constraints associated with the equations of equilibrium because the stiffness matrix generated by a finite-element model is typically ill-conditioned. Gaussian elimination techniques lose accuracy when applied to ill-conditioned equations, but this can be tolerated if the number of digits used in the computer arithmetic is high enough (most finite-element computations are done in double precision). The effect of ill-conditioning on iterative methods such as the CG method is to slow down convergence.

Recent advances in optimization methods such as preconditioned CG methods, however, improve the efficiency of the SAND approach, and make it competitive for three-dimensional problems that result in a poorly-banded stiffness matrices. As a result there has been a revival of interest in SAND approaches (see Haftka [18], Smaoui and Schmit [19], Ringertz [20], and Haftka and Kamat [21]). Overall, the SAND method eliminates the need for continually reanalysing the structure, at the expense of solving a larger optimization problem (including displacements as design variables). It is, therefore, most appropriate to use SAND in problems with a very large number of structural design variables, where the addition of displacement variables has a small effect on the total number of design variables.

The SAND method is not the method of choice when there are many load cases because in that case the number of displacement design variables becomes very large. However, Chibani [22] employed SAND in this case using a two-level approach and geometric programming to alleviate the computational burden. The method is also very useful in topology optimization where the traditional nested approach runs into

trouble when the elimination of parts of the structure can render the stiffness matrix singular (see Bendsøe et al. [23])

It is not always possible to transform a two-level problem into a single-level one. Consider, for example, the problem of maximizing the lowest frequency ω_1 of a structure subject to the constraint that its weight W does not exceed a limit W_u. A two-level formulation of the problem is

$$\begin{aligned} &\textbf{maximize} \quad \omega_1(\mathbf{x}) \\ &\textbf{such that} \quad W_u - W(\mathbf{x}) \geq 0\,, \end{aligned} \tag{10.5.3}$$

where ω_1 is the solution of the lower-level minimization of the Rayleigh quotient

$$\omega_1^2 = \min_{\mathbf{u}} \frac{\mathbf{u}^T\mathbf{K}\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}}\,, \tag{10.5.4}$$

with $\mathbf{M}$ being the mass matrix and $\mathbf{u}$ the eigenvector corresponding to ω_1. It is not possible to replace this two-level problem by the single-level problem

$$\begin{aligned} \textbf{find x and u to maximize} \quad & \omega_1^2 = \frac{\mathbf{u}^T\mathbf{K}\mathbf{u}}{\mathbf{u}^T\mathbf{M}\mathbf{u}} \\ \textbf{such that} \quad & W_u - W(\mathbf{x}) \geq 0\,, \end{aligned} \tag{10.5.5}$$

because in the above formulation the optimization will choose the eigenvector corresponding to the highest rather than the lowest frequency. It is still possible to convert this frequency maximization problem to an SAND single-level approach [24] by using the Kuhn-Tucker conditions of the problem, but the process is more complex and more computationally costly than the nested approach of Eqs. (10.5.3) and (10.5.4).

10.5.2 *Other Applications*

One of the common applications of multilevel approach to a problem with a narrow-tree form is in combined sizing and geometry optimization. Typically, the geometrical design variables are selected to be the upper level variables, and the sizing variables to be the lower-level variables. The motivation for this approach has been the disparate nature of the two types of variables that can lead to numerical difficulties when they are treated together as a single group of design variables. Typical applications have been to truss (e.g., [25–28]) and frame design (e.g., [29]–31]) problems.

10.6 Decomposition in Response and Sensitivity Calculations

Systems that have block-angular structures in term of the design optimization problem will usually have a similar structure in the analysis problem. That is, if we denote the response of the s subsystems by $\mathbf{u}_i$, $i = 1, \cdots, s$, we can often find a set of global response variables $\mathbf{w}$ which decouples the response computations (that is the

analysis) of the individual subsystems. That is, the equations governing the response of the system can be written as

$$\begin{aligned} \mathbf{r}_0(\mathbf{u}_1, \cdots, \mathbf{u}_s, \mathbf{w}) &= 0\,, \\ r_i(\mathbf{u}_i, \mathbf{w}) &= 0\,, \qquad i = 1, \cdots, s\,. \end{aligned} \tag{10.6.1}$$

We can take advantage of this block angular structure in the solution procedure. For example, consider the use of Newton's method for solving the system. Given an initial estimate for the solution we compute a correction to that estimate from a first order Taylor series expansion

$$\begin{aligned} \mathbf{r}_0 + r_{0,1}\Delta\mathbf{u}_1 + \cdots + \mathbf{r}_{0,s}\Delta\mathbf{u}_s + \mathbf{r}_{0,0}\Delta\mathbf{w} &= 0\,, \\ \mathbf{r}_i + \mathbf{r}_{i,i}\Delta\mathbf{u}_i + \mathbf{r}_{i,0}\Delta\mathbf{w} &= 0\,, \qquad i = 1, \cdots, s\,. \end{aligned} \tag{10.6.2}$$

where a comma followed by i indicates a derivative with respect to $\mathbf{u}_i$ and a comma followed by a zero indicates a derivative with respect to $\mathbf{w}$. For example, $\mathbf{r}_{0,i}$ indicates a matrix with its jth row consisting of the derivatives of the jth component of $\mathbf{r}_0$ with respect to the components of $\mathbf{u}_i$. All quantities are evaluated at the initial estimate. An examination of Eq. (10.6.2) shows that we can first express $\Delta\mathbf{u}_i$ in terms of $\Delta\mathbf{w}$ as

$$\Delta\mathbf{u}_i = -\mathbf{r}_{i,i}^{-1}(\mathbf{r}_i + \mathbf{r}_{i,0}\Delta\mathbf{w})\,, \tag{10.6.3}$$

and then substitute into Eq. (10.6.2) to obtain

$$\left(\mathbf{r}_{0,0} - \sum_{i=1}^{s} \mathbf{r}_{i,i}^{-1}\mathbf{r}_{i,0}\right)\Delta\mathbf{w} = -\mathbf{r}_0 + \sum_{i=1}^{s} \mathbf{r}_{i,i}^{-1}\mathbf{r}_i\,. \tag{10.6.4}$$

That is, the problem can be reduced to a solution of a system, Eq. (10.6.4), of the order of $\mathbf{w}$, and then the individual subsystem responses, $\mathbf{u}_i$, can be calculated, as needed, from Eq. (10.6.3).

The same procedure can be used to calculate the sensitivity of the response with respect to design variables. Assume now that the system depends also on a design parameter x. That is, we have

$$\begin{aligned} \mathbf{r}_0(\mathbf{u}_1, \cdots, \mathbf{u}_s, \mathbf{w}, x) &= 0\,, \\ \mathbf{r}_i(\mathbf{u}_i, \mathbf{w}, x) &= 0\,, \qquad i = 1, \cdots, s\,. \end{aligned} \tag{10.6.5}$$

Differentiating the system with respect to x we get

$$\begin{aligned} \frac{\partial \mathbf{r}_0}{\partial x} + r_{0,1}\frac{\partial \mathbf{u}_1}{\partial x} + \cdots + \mathbf{r}_{0,s}\frac{\partial \mathbf{u}_s}{\partial x} + \mathbf{r}_{0,0}\frac{\partial \mathbf{w}}{\partial x} &= 0\,, \\ \mathbf{r}_i + \mathbf{r}_{i,i}\frac{\partial \mathbf{u}_i}{\partial x} + \mathbf{r}_{i,0}\frac{\partial \mathbf{w}}{\partial x} &= 0\,, \qquad i = 1, \cdots, s\,. \end{aligned} \tag{10.6.6}$$

We can now express $\partial\mathbf{u}_i/\partial x$ in terms of $\partial\mathbf{w}/\partial x$ and reduce the problem to a system of the same order as that of $\mathbf{w}$.

The typical example of the above approach is substructuring. For displacement-based finite element formulation $\mathbf{w}$ is the vector of boundary degrees of freedom, and $\mathbf{u}_i$ is the vector of interior degrees of freedom of the ith substructure. However, another important example is from the area of multidisciplinary design. There each subsystem may represent a different disciplinary analysis of the same system. The $\mathbf{u}_i$ are disciplinary responses that do not influence the other disciplinary analyses while the $\mathbf{w}$ vector includes all the response quantities that affect more than one discipline. In this case, however, the components of $\mathbf{w}$ can typically be identified with one discipline or another, so that it is convenient to divide $\mathbf{w}$ into subvectors $\mathbf{w}_i$, $i = 1, \cdots, s$, where $\mathbf{w}_i$ consists of the response variables of the ith discipline which affect the response calculations in one or more other disciplines.

Sobieski [32] developed the following procedure for calculating the sensitivity of a multidisciplinary system with respect to design variables and called it the global sensitivity equation (GSE). In describing the GSE procedure we assume that the response calculations in each discipline are performed by some analytical or software blocks (or 'black boxes') or even experimental tools that can be described as

$$\mathbf{w}_i = \mathbf{r}_i(\mathbf{w}_1, \cdots, \mathbf{w}_{i-1}, \mathbf{w}_{i+1}, \cdots, \mathbf{w}_s, \mathbf{x}), \qquad \mathbf{u}_i = \mathbf{t}_i(\mathbf{w}_1, \cdots, \mathbf{w}_s, \mathbf{x}). \tag{10.6.7}$$

That is, $\mathbf{r}_i$ is a procedure for calculating $\mathbf{w}_i$ given the response of the other disciplines and a vector $\mathbf{x}$ of design variables. Similarly $\mathbf{t}_i$ represents a procedure for calculating the response $\mathbf{u}_i$. Equation (10.6.7) represents a system of coupled nonlinear equations in the $\mathbf{w}_i$, $i = 1, \cdots, s$. The solution of this system can proceed, for example, by the use of Newton's method, so that given an initial estimate $\mathbf{w}_i^0$ for the $\mathbf{w}_i$'s we can find a correction $\Delta\mathbf{w}_i$by solving

$$\mathbf{J}\Delta\mathbf{w} = \Delta\mathbf{r}, \tag{10.6.8}$$

where

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & -\mathbf{r}_{1,2} & \cdot & \cdot & \cdot & -\mathbf{r}_{1,s} \\ -\mathbf{r}_{2,1} & \mathbf{I} & \cdot & \cdot & \cdot & -\mathbf{r}_{2,s} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -\mathbf{r}_{s,1} & -\mathbf{r}_{s,2} & \cdot & \cdot & \cdot & \mathbf{I} \end{bmatrix}, \qquad \mathbf{w} = \begin{Bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{w}_s \end{Bmatrix}, \qquad \Delta\mathbf{r} = \begin{Bmatrix} \Delta\mathbf{r}_1 \\ \Delta\mathbf{r}_2 \\ \cdot \\ \cdot \\ \cdot \\ \Delta\mathbf{r}_s \end{Bmatrix}, \tag{10.6.9}$$

and where

$$\Delta\mathbf{r}_i = \mathbf{w}_i^0 - \mathbf{r}_i(\mathbf{w}_1^0, \cdots, \mathbf{w}_{i-1}^0, \mathbf{w}_{i+1}^0, \cdots, \mathbf{w}_s^0, \mathbf{x}). \tag{10.6.10}$$

After we converge to the solution for $\mathbf{w}$ we can then find the $\mathbf{u}_i$ from Eq. (10.6.7). The calculation of sensitivity with respect to a design parameter proceeds in a similar manner. Differentiating Eq. (10.6.7) with respect to a component of $\mathbf{x}$ we get

$$\mathbf{J}\frac{\partial\mathbf{w}}{\partial x_i} = \frac{\partial\mathbf{r}}{\partial x_i}. \tag{10.6.11}$$

The special structure of the Jacobian $\mathbf{J}$ permits us to reduce the order of the equations by eliminating one of the $\mathbf{w}_i$'s as illustrated in the example below.

The GSE approach requires the derivatives of the individual disciplinary responses with respect to the input of all the other disciplines. The cost of these calculation can be very large when the front of interaction between disciplines is large. In comparing the cost of the GSE approach to that of finite-difference calculation of the derivatives, a key parameter is the number of design variables. For a large number of design variables, the GSE method tends to be more efficient than the finite-difference method, while for a small number of design variables finite-differences are less expensive. For a more detailed discussion of the cost issues, as well as the pathological cases when the GSE matrix may be singular, the reader is referred to [32].

As noted before, the major difficulty associated with using multilevel techniques is in finding a way to decompose the problem so that it would have the requisite hierarchical structure. Successful decomposition breaks the problem into elements that have only narrow fronts of interaction. For multidisciplinary analysis and sensitivity we seek ways to narrow the front of interaction between disciplines. The following example of integrated aerodynamic-structural wing analysis and sensitivity calculations illustrates the use of a reduced-basis technique for achieving this goal.

Example 10.6.1

Consider the aeroelastic analysis of an aircraft wing. The flow field around the wing is calculated based on the shape of the wing. Then pressures and loads are calculated from flow velocities, and these are used to calculate structural displacements which in turn change the shape of the wing. The solution for this coupled problem is often performed iteratively, starting with the flow field around a rigid wing, continuing with the loads and displacements associated with this flow field, updating the shape of the wing based on these displacements, and so on. This approach, called *fixed-point iteration*, may be preferable to Newton's method if the calculation of the Jacobian is expensive. However, if we need also the sensitivity of the aeroelastic response to design parameters, it may be advantageous to use Newton's method instead of the fixed-point iteration. The feasibility of using Newton's method depends on the width of the front of interaction. To focus on the question of the front of interaction we start without consideration of design variables and examine the solution of the aeroelastic interaction.

We assume that we have an aerodynamic 'black box' that solves for the flow field represented, say, by the velocity vector, $\mathbf{v}$, given the shape of the wing which is represented by a shape vector, $\mathbf{s}$,

$$\mathbf{v} = \mathbf{b}_a(\mathbf{s})\,, \qquad (a)$$

where $\mathbf{b}_a$ denotes the application of the aerodynamic black box. Next we have a force black box which translates the flow velocities into aerodynamic loads $\mathbf{f}_a$ that can be used in the structural analysis

$$\mathbf{f}_a = \mathbf{b}_f(\mathbf{v})\,. \qquad (b)$$

The next black box is the structural analysis package which combines the aerodynamic loads with inertia loads and calculates the displacement vector $\mathbf{u}$

$$\mathbf{u} = \mathbf{b}_s(\mathbf{f}_a)\,. \qquad (c)$$

Finally, we have an interpolation black-box which updates the shape of the wing based on the displacement field

$$\mathbf{s} = \mathbf{b}_I(\mathbf{u}) . \tag{d}$$

At first glance, the system described by Eqs. (a)–(d) appears to be fully coupled. Solving this system by Newton's method appears to be impractical because of the huge size of the Jacobian. The flow field vector $\mathbf{v}$ and the displacement vector $\mathbf{u}$ usually have thousands or tens of thousands of components. However, the vectors $\mathbf{f}_a$ and $\mathbf{s}$ can have a fairly small number of components, and we can reduce the problem size enormously by combining the first two and last two black boxes. The first combination gives us the aerodynamic forces in terms of the shape of the wing

$$\mathbf{f}_a = \mathbf{r}_1(\mathbf{s}) = \mathbf{b}_f[\mathbf{b}_a(\mathbf{s})] , \tag{e}$$

and the second combination gives us the shape of the wing as a function of the aerodynamic forces

$$\mathbf{s} = \mathbf{r}_2(\mathbf{f}_a) = \mathbf{b}_I[\mathbf{b}_s(\mathbf{f}_a)] . \tag{f}$$

We note that the variables $\mathbf{f}_a$ and $\mathbf{s}$ play the role of $\mathbf{w}_1$ and $\mathbf{w}_2$ in Eq. (10.6.7), while $\mathbf{v}$ and $\mathbf{u}$ play the role of $\mathbf{u}_1$ and $\mathbf{u}_2$.

The above approach of using only $\mathbf{f}_a$ and $\mathbf{s}$ as interaction variables leads to a great reduction in the number of cross-derivatives that need to be calculated. However, the number of components of $\mathbf{f}_a$ and $\mathbf{s}$ is often several dozens, and calculating the Jacobian can still be prohibitively expensive. Further reduction in the number of required derivatives is achieved by using a reduced basis technique to represent the displacements for the purpose of describing the aeroelastic interaction. The displacement vectors are assumed to be adequately represented by a linear combination of mode shapes (often vibration modes) as

$$\mathbf{u} = \mathbf{U}\mathbf{q} , \tag{g}$$

where $\mathbf{U}$ is a matrix of modes and $\mathbf{q}$ a vector of modal amplitudes. The order of the vector $\mathbf{q}$ is typically much smaller than that of $\mathbf{u}$ or even $\mathbf{f}_a$. Furthermore, for the reduced basis structural analysis we now do not need $\mathbf{f}_a$ but instead the generalized load vector $\mathbf{f}_a^*$ given as (see Eq. (7.4.30))

$$\mathbf{f}_a^* = \mathbf{U}^T\mathbf{f}_a . \tag{h}$$

The reduced-basis (or modal) structural analysis black box is now described schematically as

$$\mathbf{q} = \mathbf{b}_s^*(\mathbf{f}_a^*) . \tag{i}$$

It is now most efficient to group our four black boxes in a slightly different order to make $\mathbf{f}_a^*$ and $\mathbf{q}$ the interaction variables. That is, the generalized aerodynamic forces are given in terms of the modal amplitudes as

$$\mathbf{f}_a^* = \mathbf{r}_1^*(\mathbf{q}) = \mathbf{U}^T\mathbf{b}_f\{\mathbf{b}_a[\mathbf{b}_I(\mathbf{U}\mathbf{q})]\} , \tag{j}$$

and $\mathbf{r}_2^*$ is simply equal to $\mathbf{b}_s^*$.

For the Newton iteration, Eq. (10.6.8), we need to calculate $\mathbf{J}_{12} = \partial \mathbf{r}_1^*/\partial \mathbf{q}$ and $\mathbf{J}_{21} = \partial \mathbf{r}_2^*/\partial \mathbf{f}_a^*$. These are cross derivatives, in that they are derivatives of the aerodynamic forces with respect to the shape changes due to structural displacements, and derivatives of shape change due to structural displacements with respect to the aerodynamic loads. $\mathbf{J}_{12}$ and $\mathbf{J}_{21}$ are matrices, and it is convenient to label them as $\mathbf{A}$ and $\mathbf{S}$. The component a_{ij} of the matrix $\mathbf{A}$ is the derivative of the ith component of $\mathbf{f}_a^*$, f_{ai}^*, with respect to the jth component of $\mathbf{q}$, q_j. Similarly the component s_{ij} of the matrix $\mathbf{S}$ is the derivative of q_i with respect to f_{aj}^*. These derivatives are often calculated by finite differences. For example, if we perturb q_j and recalculate $\mathbf{f}_a^*$ from Eq. (j) we can estimate the jth column of the matrix $\mathbf{A}$ as the difference in $\mathbf{f}_a^*$ divided by the perturbation in q_j.

Equation (10.6.8) can now be written as

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A} \\ -\mathbf{S} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{f}_a^* \\ \Delta \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} \Delta \mathbf{r}_1^* \\ \Delta \mathbf{r}_2^* \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_a^{*0} - \mathbf{r}_1^*(\mathbf{q}^0) \\ \mathbf{q}^0 - \mathbf{r}_2^*(\mathbf{f}_a^{*0}) \end{Bmatrix} . \tag{k}$$

Because of the special structure of Eq. (k) we can eliminate either $\Delta \mathbf{q}$ or $\Delta \mathbf{f}_a^*$. For example, if $\Delta \mathbf{f}_a^*$ has more components than $\Delta \mathbf{q}$, it may be advantageous to eliminate $\Delta \mathbf{f}_a^*$ by using the first row of Eq. (k)

$$\Delta \mathbf{f}_a^* = \mathbf{A} \Delta \mathbf{q} + \Delta \mathbf{r}_1^* , \tag{l}$$

and substituting it into the second row to get

$$(\mathbf{I} - \mathbf{S}\mathbf{A}) \Delta \mathbf{q} = \Delta \mathbf{r}_2^* + \mathbf{S} \Delta \mathbf{r}_1^* . \tag{m}$$

The solution of the acroelastic interaction via Newton's method will consist then of solving Eq. (m) for $\Delta \mathbf{q}$, followed by the calculation of $\Delta \mathbf{f}_a$ from Eq. (l), and then updating $\mathbf{q}$ and $\mathbf{f}_a^*$ and calculating new $\mathbf{A}$ and $\mathbf{S}$ to repeat the iteration.

The calculation of sensitivity with respect to a design parameter x will proceed along the same line. Equation (10.6.11) will become

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A} \\ -\mathbf{S} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \mathbf{f}_a^*}{\partial x} \\ \dfrac{\partial \mathbf{q}}{\partial x} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial \mathbf{r}_1^*}{\partial x} \\ \dfrac{\partial \mathbf{r}_2^*}{\partial x} \end{Bmatrix} . \tag{n}$$

While the reduced basis technique approximates the aeroelastic interaction, it does not require that we also approximate the calculation in each individual discipline. After we find $\mathbf{f}_a^*$ and $\mathbf{q}$ from the coupled analysis, we do not need to use Eq. (g) to calculate the displacements. Instead we can calculate the actual aerodynamic forces $\mathbf{f}_a$ corresponding to displacements $\mathbf{U}\mathbf{q}$, and then calculate the displacement from the full structural analysis, Eq. (c). • • •

10.7 Exercises

1. Consider the 3 bar truss of Figure 10.2.2. The cross-sectional areas and moments of inertia of the three members are given, and we want to optimize the geometry of the truss to minimize the weight subject to the constraint that the truss does not collapse under either load case (consider both yielding and Euler buckling). Formulate the problem in a block-angular form.

2. Consider the portal frame of Figure 10.2.4. Formulate the minimum weight design of the frame subject to stress constraints and horizontal displacement limit of 10cm. The design variables are the cross-sectional dimensions for each of the three beams. Define global design variables to reduce the problem to a block-angular form.

3. Calculate the derivatives of $\bar{g}$ in Example 10.2.3 with respect to its other five arguments.

4. Obtain the solution of Example 10.4.1.

5. Solve Example 10.4.1 using the KS function.

6. Formulate the elastic design problem of the three-bar truss (Example 10.2.2) as a simultaneous-analysis and design problem.

10.8 References

[1] Giles, G.L. "Procedure for Automating Aircraft Wing Structural Design," J. of the Structural Division, ASCE, 97 (ST1), pp. 99–113, 1971.

[2 Sobieszczanski, J. and Loendorf, D., "A Mixed Optimization Method for Automated Design of Fuselage Structures", J. of Aircraft, 9 (12), pp. 805–811, 1972.

[3] Barthelemy, J.-F.,M., "Engineering Design Applications of Multilevel Optimization Methods," in Computer-Aided Optimum Design of Structures: Applications (eds. C.A. Brebbia and S. Hernandez), Springer-Verlag, pp. 113–122, 1989.

[4] Sobieszczanski-Sobieski, J., James, B.B., and Dovi, A.R., "Structural Optimization by Multilevel Decomposition", AIAA J., 23, 11, pp. 1775–1782, 1985.

[5] Thareja, R. R., and Haftka, R. T., "Efficient Single-Level Solution of Hierarchical Problems in Structural Optimization", AIAA J., 28, 3, pp. 506–514, 1990.

[6] Thareja, R., and Haftka, R.T., "Numerical Difficulties Associated with using Equality Constraints to Achieve Multilevel Decomposition in Structural Optimization," AIAA Paper No. 86-0854CP, Proceedings of the AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference, San Antonio, Texas, May 1986, pp. 21–28.

[7] Schmit L.A., and Mehrinfar, M., "Multilevel Optimum Design of Structures with Fiber-Composite Stiffened Panel Components", AIAA J., 20,1, pp. 138–147, 1982.

[8] Kirsch, U., "Multilevel Optimal Design of Reinforced Concrete Structures", Engineering Optimization, 6, pp. 207–212, 1983.

[9] Dantzig, G.B., and Wolfe, P., "The Decomposition Algorithm for Linear Program," Econometrica, 29, No. 4, pp. 767–778, 1961.

[10] Dantzig, G.B., "A Decomposition Principle for Linear Programs," in Linear Programming and Extensions, Princeton Press, 1963.

[11] Rosen, J.B., "Primal Partition Programming for Block Diagonal Matrices", Numerische Mathematik, 6, pp. 250–260, 1964.

[12] Geoffrion, A.M., "Elements of Large-Scale Mathematical Programming", in Perspectives on Optimization (A.M. Geoffrion, editor) Addison Wesley, pp. 25–64, 1972.

[13] Kirsch, U., "An Improved Multilevel Structural Synthesis Method", J. Structural Mechanics, 13 (2), pp. 123–144, 1985.

[14] Barthelemy, J.-F.M., and Sobieszczanski-Sobieski, J., "Extrapolation of Optimum Designs based on Sensitivity Derivatives," AIAA J., 21, pp. 797–799, 1983.

[15] Haftka, R.T., "An Improved Computational Approach for Multilevel Optimum Design", J. of Structural Mechanics, 12, 2, pp. 245–261, 1984.

[16] Sobieszczanski-Sobieski, J., James, B. B., and Riley, M. F., "Structural Sizing by Generalized, Multilevel Optimization", AIAA J., 25, 1, pp. 139–145, 1987.

[17] Fox, R. L., and Schmit, L. A., "Advances in the Integrated Approach to Structural Synthesis", J. of Spacecraft and Rockets, 3 (6), pp.858–866, 1966.

[18] Haftka, R.T., "Simultaneous Analysis and Design", AIAA J., 23, 7, pp. 1099-1103, 1985.

[19] Smaoui, H., and Schmit. L.A., "An Integrated Approach to the Synthesis of Geometrically Non-linear Structures," International Journal for Numerical Methods in Engineering, 26, pp. 555-570, 1988.

[20] Ringertz, U.T., "Optimization of Structures with Nonlinear Response," Engineering Optimization, 14, pp. 179–188, 1989.

[21] Haftka, R. T., and Kamat, M. P., "Simultaneous Nonlinear Structural Analysis and Design", Computational Mechanics, 4, 6, pp. 409–416, 1989.

[22] Chibani, L., Optimum Design of Structures, Springer-Verlag, Berlin, Heidelberg, 1989.

[23] Bendsøe, M.P., Ben-Tal, A., and Haftka, R.T., "New Displacement-Based Methods for Optimal Truss Topology Design," AIAA Paper 91-1215, Proceedings, AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference, Baltimore, MD, April 8–10, 1991, Part 1, pp. 684–696.

[24] Shin, Y., Haftka, R. T., and Plaut, R. H., "Simultaneous Analysis and Design for Eigenvalue Maximization", AIAA J., 26, 6, pp. 738–744, 1988.

[25] Pedersen, P., "On the Minimum Mass Layout of Trusses", AGARD Conference Proceedings, No. 36 on Symposium on Structural Optimization, Turkey, October, 1969, pp. 11.1-11.17, 1970.

[26] Vanderplaats, G.N., and Moses, F., "Automated Design of Trusses for Optimum Geometry", J. of the Structural Division, ASCE, 98, ST3, pp. 671–690, 1972.

[27] Spillers, W.R., "Iterative Design for Optimal Geometry", J. of the Structural Division, ASCE, 101, ST7, pp.1435–1442, 1975.

[28] Kirsch, U., "Synthesis of Structural Geometry using Approximation Concepts", Computers and Structures, 15, 3, pp. 305–314, 1982.

[29] Ginsburg, S., and Kirsch, U., "Design of Protective Structures against Blast", J. of the Structural Division, ASCE, 109 (6), pp. 1490–1506, 1983.

[30] Kirsch, U., "Multilevel Synthesis of Standard Building Structures," Engineering Optimization, 7, pp. 105–120, 1984.

[31] Kirsch, U., "A Bounding Procedure for Synthesis of Prestressed Systems," Computers and Structures, 20 (5), pp. 885–895, 1985.

[32] Sobieszczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," AIAA Journal, 28 (1), pp. 153–160, 1990.

Optimum Design of Laminated Composite Structures 11

Because of their superior mechanical properties compared to single phase materials, laminated fibrous composite materials are finding a wide range of applications in structural design, especially for lightweight structures that have stringent stiffness and strength requirements. Designing with laminated composites, on the other hand, has become a challenge for the designer because of a wide range of parameters that can be varied, and because the complex behavior and multiple failure modes of these structures require sophisticated analysis techniques. Finding an efficient composite structural design that meets the requirements of a certain application can be achieved not only by sizing the cross-sectional areas and member thicknesses, but also by global or local tailoring of the material properties through selective use of orientation, number, and stacking sequence of laminae that make up the composite laminate. The increased number of design variables is both a blessing and a curse for the designer, in that he has more control to fine-tune his structure to meet design requirements, but only if he can figure out how to select those design variables. The possibility of achieving an efficient design that is safe against multiple failure mechanisms, coupled with the difficulty in selecting the values of a large set of design variables makes structural optimization an obvious tool for the design of laminated composite structures.

Because of the need for sophisticated analysis tools for most realistic applications, designing with laminated composites largely relied on procedures that simply coupled those analyses with black-box optimizers. However a better understanding of the peculiarities associated with optimization of composites can best be illustrated through simple examples. In this chapter we emphasize examples that focus on basic concepts.

11.1 Mechanical Response of a Laminate

While laminated composite materials are attractive replacements for metallic materials for many structural applications that require high stiffness-to-weight and high strength-to-weight ratios, the analysis and design of these materials are considerably

more complex than those of metallic structures. One of the complexities in formulating the analysis of a laminated composite material is due to material anisotropy that requires an increased number of material constants for characterization of the mechanical response of the laminate. The generalized Hooke's law for an anisotropic material is given in terms of 21 independent stiffness coefficients. It is this aspect of composite materials which makes them attractive for optimal design and tailoring purposes. However, for a general structure with a three-dimensional stress state it is very difficult to solve the governing equations. Fortunately, most composite structures are plate-type structures which are composed of layers or plies of orthotropic material which can be characterized in terms of a smaller number of stiffness constants. In the following section, the basic equations that govern the mechanical response of an orthotropic lamina are summarized.

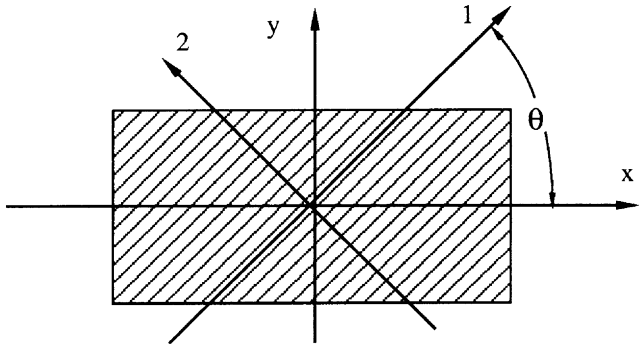


Figure 11.1.1 An orthotropic lamina with off-axis principal material directions.

11.1.1 *Orthotropic Lamina*

For an orthotropic material with the axes of orthotropy 1-2 aligned with the x-y coordinate axes ($\theta = 0$ in Fig. 11.1.1), the stress-strain relation in the principal material directions is given by the following set of equations with 9 independent constants.

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{31} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{Bmatrix} . \tag{11.1.1}$$

Furthermore, by assuming a plane stress state in each of the layers in the 1-2 principal material plane, we have

$$\sigma_3 = 0, \qquad \tau_{23} = 0, \quad \text{and} \quad \tau_{31} = 0, \tag{11.1.2}$$

which reduces the stress-strain relations to [1]

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix}, \tag{11.1.3}$$

where the Q_{ij}'s are called the reduced stiffnesses and are given in terms of four independent engineering material constants in principal material directions as

$$\begin{aligned} Q_{11} &= \frac{E_1}{1-\nu_{12}\nu_{21}}, \qquad Q_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}}, \\ Q_{12} &= \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} \quad = \quad \frac{\nu_{21}E_1}{1-\nu_{12}\nu_{21}}, \\ &\text{and} \quad Q_{66} = G_{12} \ . \end{aligned} \tag{11.1.4}$$

Since the orthotropic layers are generally rotated with respect to reference coordinate axes, the stress-strain relations given in the principal directions of material orthotropy Eq. (11.1.3) must be transformed to these axes. This transformation produces

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix}, \tag{11.1.5}$$

where the transformed reduced stiffnesses $\bar{Q}_{ij}$ are related to the Q_{ij} by

$$\begin{aligned} \bar{Q}_{11} &= Q_{11}\cos^4\theta + 2(Q_{12}+2Q_{66})\sin^2\theta\cos^2\theta + Q_{22}\sin^4\theta \, , \\ \bar{Q}_{12} &= (Q_{11}+Q_{22}-4Q_{66})\cos^2\theta\sin^2\theta + Q_{12}(\sin^4\theta+\cos^4\theta) \, , \\ \bar{Q}_{22} &= Q_{11}\sin^4\theta + 2(Q_{12}+2Q_{66})\sin^2\theta\cos^2\theta + Q_{22}\cos^4\theta \, , \\ \bar{Q}_{16} &= (Q_{11}-Q_{12}-2Q_{66})\sin\theta\cos^3\theta + (Q_{12}-Q_{22}+2Q_{66})\sin^3\theta\cos\theta \, , \\ \bar{Q}_{26} &= (Q_{11}-Q_{12}-2Q_{66})\sin^3\theta\cos\theta + (Q_{12}-Q_{22}+2Q_{66})\sin\theta\cos^3\theta \, , \\ \bar{Q}_{66} &= (Q_{11}+Q_{22}-2Q_{12}-2Q_{66})\sin^2\theta\cos^2\theta + Q_{66}(\sin^4\theta+\cos^4\theta) \ . \end{aligned} \tag{11.1.6}$$

Equations (11.1.6) are the basic building blocks of the *classical lamination theory* which will be discussed next. These equations, however, can be put into a simpler form in terms of the angular orientation of the principal axis of orthotropy with respect to the reference x–y coordinate system. Tsai and Pagano [2] defined the following material properties that are invariant with respect to ply orientation

$$\begin{aligned} U_1 &= \frac{1}{8}(3Q_{11}+3Q_{22}+2Q_{12}+4Q_{66}) \, , \\ U_2 &= \frac{1}{2}(Q_{11}-Q_{22}) \, , \\ U_3 &= \frac{1}{8}(Q_{11}+Q_{22}-2Q_{12}-4Q_{66}) \, , \\ U_4 &= \frac{1}{8}(Q_{11}+Q_{22}+6Q_{12}-4Q_{66}) \, , \\ U_5 &= \frac{1}{8}(Q_{11}+Q_{22}-2Q_{12}+4Q_{66}) \ . \end{aligned} \tag{11.1.7}$$

Using various trigonometric identities, we can write the transformed reduced stiffnesses of Eq. (11.1.6) as

$$\begin{aligned}
\bar{Q}_{11} &= U_1 + U_2 \cos 2\theta + U_3 \cos 4\theta \,, \\
\bar{Q}_{12} &= U_4 - U_3 \cos 4\theta \,, \\
\bar{Q}_{22} &= U_1 - U_2 \cos 2\theta + U_3 \cos 4\theta \,, \\
\bar{Q}_{16} &= -\frac{1}{2} U_2 \sin 2\theta - U_3 \sin 4\theta \,, \\
\bar{Q}_{26} &= -\frac{1}{2} U_2 \sin 2\theta + U_3 \sin 4\theta \,, \\
\bar{Q}_{66} &= U_5 - U_3 \cos 4\theta \,.
\end{aligned} \tag{11.1.8}$$

The above form of the reduced stiffnesses is simpler than those shown in Eq. (11.1.6) in terms of the ply orientation and, therefore, is useful for design optimization purposes where derivatives of the stiffnesses with respect to the orientation variables are needed.

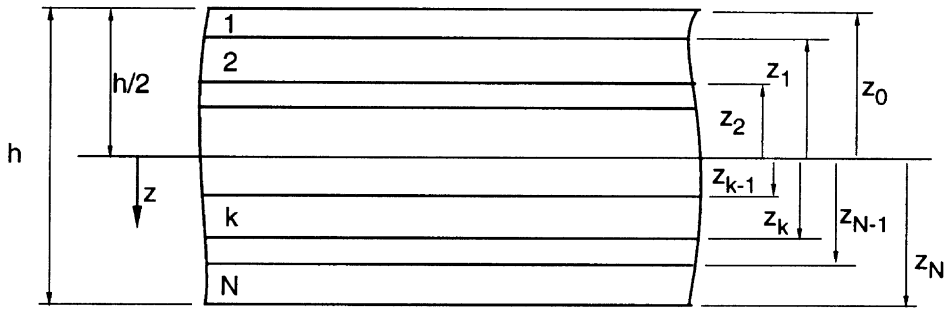


Figure 11.1.2 Laminate stacking convention.

11.1.2 *Classical Laminated Plate Theory*

Classical lamination theory (CLT) assumes that the N orthotropic layers described above are perfectly bonded together, as in Fig. 11.1.2, with a non-shear-deformable, infinitely thin bondline. Kirchhoff plate theory is used, which assumes a linear through-the-thickness variation of the in-plane displacements,

$$u = u_o - z\frac{\partial w_o}{\partial x}, \qquad v = v_o - z\frac{\partial w_o}{\partial y}, \tag{11.1.9}$$

and vanishing through-the-thickness strain components, $\epsilon_z = \gamma_{xz} = \gamma_{yz} = 0$ and $w = w_o$. The strain distribution is, therefore

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \epsilon_x^o \\ \epsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} + z \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix}, \tag{11.1.10}$$

where the superscript o indicates the mid-plane strains, and the curvatures κ are the mid-plane curvatures. Therefore, the stresses in the kth ply can be expressed in terms of the reduced stiffnesses of that particular ply by substituting Eq. (11.1.10) into the stress-strain relationship, Eq. (11.1.5)

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}_k \left(\begin{Bmatrix} \epsilon_x^o \\ \epsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} + z \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \right) . \tag{11.1.11}$$

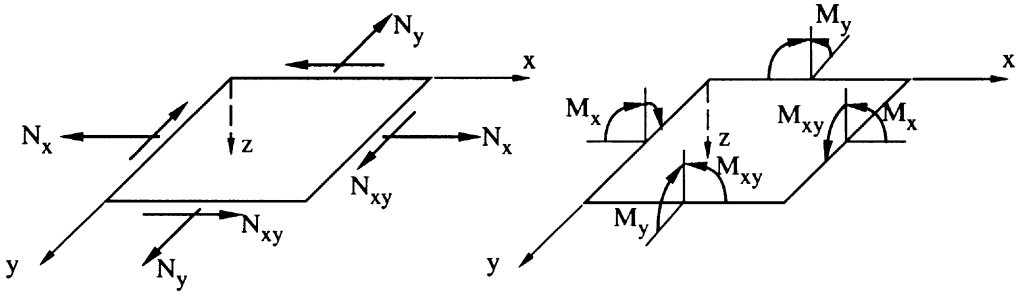


Figure 11.1.3 Stress and moment resultants in a laminate.

The net stress resultant and moment resultant (stress couple) per unit length of the cross section acting at a point in the laminate, see Fig. 11.1.3, are obtained by through-the-thickness integration of the stresses in each ply,

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \int_{-h/2}^{h/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k dz = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} dz , \tag{11.1.12}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \int_{-h/2}^{h/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k z dz = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} z dz . \tag{11.1.13}$$

Substituting the stress-strain relations of Eq. (11.1.11), we obtain the following constitutive relations for the laminate,

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x^o \\ \epsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} , \tag{11.1.14}$$

and

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x^o \\ \epsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} , \tag{11.1.15}$$

where

$$A_{ij} = \sum_{k=1}^{N} (\bar{Q}_{ij})_k (z_k - z_{k-1}), \tag{11.1.16}$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} (\bar{Q}_{ij})_k (z_k^2 - z_{k-1}^2), \tag{11.1.17}$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} (\bar{Q}_{ij})_k (z_k^3 - z_{k-1}^3) . \tag{11.1.18}$$

11.1.3 *Bending, Extension, and Shear Coupling*

The **A** and **D** matrices are the extensional and flexural stiffness matrices, respectively. The **A** matrix relates the in-plane stress resultants to the mid-plane strains, and the **D** matrix relates the moment resultants to the curvatures. The **B** matrix, on the other hand, relates the in-plane stress resultants to the curvatures and moment resultants to the mid-plane strains, and hence is called the bending-extension coupling matrix. This coupling matrix can be a useful tool in designing laminates for certain structural applications. If it is undesirable, the **B** matrix can be avoided by a symmetric placement of the plies with different orientations with respect to the mid-plane of a laminate. However, as noted by Caprino and Crivelli-Visconti [3] and by Gunnink [4], symmetry is a sufficient but not a necessary condition to avoid coupling. It is shown by Kandil and Verchery [5] that a certain class of laminates, such as laminates consisting of two symmetric sub-laminates with equal numbers of plies and equal but arbitrary fiber orientations, θ_1 and θ_2, for which the minimum number of layers is eight $[\theta_1/\theta_2/\theta_2/\theta_1/\theta_2/\theta_1/\theta_1/\theta_2]$, possess no bending-extension coupling. This may be important for design optimization purposes because symmetric placement of the plies may restrict certain combinations of the in-plane and bending stiffnesses.

In addition to the bending-extension coupling, certain elements of the **A**, **B**, and **D** matrices result in coupling response. When the A_{16} and A_{26} terms are not zero, there is a shear-extension coupling. The existence of D_{16} and D_{26} terms induces bending-twisting coupling, and bending-shear coupling as well as extension-twisting coupling results from non-zero B_{16} and B_{26} terms. Again, by proper selection of the laminate, these coupling terms can be eliminated. For example, by using negative angle plies for every positive angle ply used in the laminate one can eliminate the shear-extension coupling. Such laminates are referred to as balanced laminates. However, these same terms can also be manipulated to tailor the response of a laminate to the needs of a specified design application, as in the case of aeroelastic tailoring (see Section 11.4.2).

The **A**, **B**, and **D** matrices are commonly used in the literature in the form defined in Eqs. (11.1.16)-(11.1.18) together with the definitions of $(\bar{Q}_{ij})$ given by Eq. (11.1.6). However, for some design procedures, the use of sines and cosines of multiple angles (see Eqs. 11.1.8) proved to be more useful, especially for derivation

of the sensitivities of these matrices with respect to the angular orientation design variables. Starting with the integral form of the Eqs. (11.1.16)-(11.1.18), for example,

$$\{A_{11}, B_{11}, D_{11}\} = \int_{-h/2}^{h/2} \bar{Q}_{11}\{1, z, z^2\} dz \,, \tag{11.1.19}$$

and assuming each layer to be of the same material, we have

$$\{A_{11}, B_{11}, D_{11}\} = U_1\{h, 0, \frac{h^3}{12}\} + U_2 \int_{-h/2}^{h/2} \cos 2\theta\{1, z, z^2\} dz + U_3 \int_{-h/2}^{h/2} \cos 4\theta\{1, z, z^2\} dz \,. \tag{11.1.20}$$

Similar expressions can be found for the other stiffness terms, and are summarized in Table 11.1.1 where the expressions for the V's are the following

Table 11.1.1 : **A**, **B**, **D** Matrices in Terms of Lamina Invariants

	$V_{0\{\mathbf{A},\mathbf{B},\mathbf{D}\}}$	$V_{1\{\mathbf{A},\mathbf{B},\mathbf{D}\}}$	$V_{2\{\mathbf{A},\mathbf{B},\mathbf{D}\}}$	$V_{3\{\mathbf{A},\mathbf{B},\mathbf{D}\}}$	$V_{4\{\mathbf{A},\mathbf{B},\mathbf{D}\}}$
$\{A_{11}, B_{11}, D_{11}\}$	U_1	U_2	0	U_3	0
$\{A_{22}, B_{22}, D_{22}\}$	U_1	$-U_2$	0	U_3	0
$\{A_{12}, B_{12}, D_{12}\}$	U_4	0	0	$-U_3$	0
$\{A_{66}, B_{66}, D_{66}\}$	U_5	0	0	$-U_3$	0
$2\{A_{16}, B_{16}, D_{16}\}$	0	0	$-U_2$	0	$-2U_3$
$2\{A_{26}, B_{26}, D_{26}\}$	0	0	$-U_2$	0	$2U_3$

$$\begin{aligned}
V_{0\{\mathbf{A},\mathbf{B},\mathbf{D}\}} &= \{h, 0, \frac{h^3}{12}\} \,, \\
V_{1\{\mathbf{A},\mathbf{B},\mathbf{D}\}} &= \int_{-h/2}^{h/2} \cos 2\theta\{1, z, z^2\} dz \,, \\
V_{2\{\mathbf{A},\mathbf{B},\mathbf{D}\}} &= \int_{-h/2}^{h/2} \sin 2\theta\{1, z, z^2\} dz \,, \\
V_{3\{\mathbf{A},\mathbf{B},\mathbf{D}\}} &= \int_{-h/2}^{h/2} \cos 4\theta\{1, z, z^2\} dz \,, \\
V_{4\{\mathbf{A},\mathbf{B},\mathbf{D}\}} &= \int_{-h/2}^{h/2} \sin 4\theta\{1, z, z^2\} dz \,.
\end{aligned} \tag{11.1.21}$$

The above set of integrals can again be replaced by summations.

11.2 Laminate Design

The laminate stiffness matrices described in the previous section can be manipulated both by changing the number of layers and their orientations. Therefore, use of these quantities as design variables enables us to change the material properties of a laminate as well as its thickness. In many practical applications, bending-extension and shear-extension coupling is undesirable. Consequently, most laminates in use today are symmetric and balanced to eliminate these couplings. Balanced symmetric laminates are also much easier to analyze. For example, analysis of a laminate with bending-extension coupling is difficult because out-of-plane deformations associated with in-plane loads may be large and, therefore, require nonlinear analysis capability. Therefore, most of optimization work to date has been limited to balanced symmetric laminates. In the remainder of this chapter only such laminates are considered.

Most commercially available composite materials come in fixed ply thickness. Furthermore, most of the data available for laminate behavior is limited to ply orientations of 0-, 90-, and ±45-deg. For these reasons, laminate design is primarily an integer programming problem. However, most of the available optimization software is for continuous-valued design variables and the past work on laminate optimization are based on the use of such variables. The total thicknesses of contiguous plies of the same orientation, referred to as the ply thickness variables, were commonly used as design variables. Ply orientations were also occasionally used as design variables, with orientations taking any value between 0- and 90-deg. The final ply thicknesses (or orientations) can be rounded-off to integer multiples of the commertially available ply thickness (or convential ply orientations). However, for large number of design variables finding a rounded-off design that does not violate any constraint is often difficult. Also, the problem must be formulated with a given stacking sequence, rather than letting the optimization obtain the best stacking sequence. For these reasons, there is a growing interest in the application of integer programming methods to laminate design. We start this chapter with description of approaches that implement traditional continuous-valued variables, with integer programming applications described in section 11.3.

There are a number of design considerations for optimization of laminated plates depending on the intended application. One of the key considerations in terms of analysis and design is whether the plate is designed for in-plane or out-of-plane response. For the sake of simplicity we review these two cases separately.

11.2.1 *Design of Laminates for In-plane Response*

Ply Thickness Variables : One of the earliest efforts in designing laminates for in-plane strength and stiffness requirements is due to Schmit and Farshi [6] who considered a symmetric balanced laminate with fixed ply orientations. The thicknesses of the individual layers $t_i, i = 1, \ldots, I$ with different prescribed orientations were used as design variables. Because of the symmetric laminate restriction, only the thicknesses of one half, I, of the total number of layers, N, are used. The laminate is under the action of combined membrane force resultants, $N_{xk}, N_{yk}, N_{xyk}, k = 1, \ldots, K$ where K is the number of load cases.

The optimization problem is formulated as the following:

$$\textbf{minimize} \qquad W = \sum_{i=1}^{I} 2\rho_i t_i \tag{11.2.1}$$

$$\textbf{subject to} \qquad g^s_{ijk} = 1 - \left(P^{(i)}_j \epsilon_{1ik} + Q^{(i)}_j \epsilon_{2ik} + R^{(i)}_j \gamma_{12ik}\right) \geq 1\,, \tag{11.2.2}$$

$$A_{11} \geq A_{11l}, \quad A_{22} \geq A_{22l}, \quad A_{66} \geq A_{66l}\,, \tag{11.2.3}$$

$$\text{and} \quad t_i \geq 0\,, \tag{11.2.4}$$

$$\text{for } i = 1,\ldots,I\,, \quad j = 1,\ldots,J\,, k = 1,\ldots,K\,, \tag{11.2.5}$$

where $P^{(i)}_j, Q^{(i)}_j, R^{(i)}_j$; are coefficients which define the jth boundary of a failure envelope for each layer (i) in the strain space, and the $\epsilon_{1ik}, \epsilon_{2ik}$, and γ_{12ik} are the principal material-direction strains in the ith layer under the kth load condition. For a simple maximum strain criterion, which puts bounds on the maximum values of the strains in the principal material directions, the failure envelope has 6 facets with P, Q, and R defined as the inverse of the normal and shearing failure strains in the longitudinal and transverse directions to the fibers in tension and compression. Equations (11.2.3) prescribe lower limits A_{11l}, A_{22l}, and A_{66l} of the membrane stiffnesses of the laminate.

The approach used by Schmit and Farshi transforms the nonlinear programming problem described in Eqs. (11.2.1)-(11.2.5) into a sequence of linear programs (see section 6.1). The inequality constraint Eq. (11.2.2) representing the strength criterion is a nonlinear function of the thickness variables and, therefore, is linearized as

$$g^s_{ijkL}(\mathbf{t}) = g^s(\mathbf{t}_0) + \sum_{l=1}^{I} (t_l - t_{0l}) \left(P^{(i)}_j \frac{\partial \epsilon_{1ik}}{\partial t_l} + Q^{(i)}_j \frac{\partial \epsilon_{2ik}}{\partial t_l} + R^{(i)}_j \frac{\partial \gamma_{12ik}}{\partial t_l} \right)\,, \tag{11.2.6}$$

where the derivatives of the principal strains in the ith layer are related to the derivatives of the laminate strains through the transformation relations

$$\frac{\partial \boldsymbol{\varepsilon}_{ik}}{\partial t_l} = \mathbf{T}_i \frac{\partial \boldsymbol{\varepsilon}^o_k}{\partial t_l}\,, \tag{11.2.7}$$

where $\boldsymbol{\varepsilon}_{ik} = (\epsilon_{1ik}, \epsilon_{2ik}, \gamma_{12ik})^T$, and $\mathbf{T}_i$ is the transformation matrix for the ith layer defined by

$$\mathbf{T}_i = \begin{bmatrix} \cos^2\theta_i & \sin^2\theta_i & \cos\theta_i \sin\theta_i \\ \sin^2\theta_i & \cos^2\theta_i & -\cos\theta_i \sin\theta_i \\ -2\cos\theta_i \sin\theta_i & 2\cos\theta_i \sin\theta_i & (\cos^2\theta_i - \sin^2\theta_i) \end{bmatrix}\,. \tag{11.2.8}$$

For a given in-plane loading condition $\mathbf{N}_k = (N_{xk}, N_{yk}, N_{xyk})^T$, the derivative of the laminate strains with respect to the thickness variables can be determined by differentiating Eq. (11.1.14), $\mathbf{N} = \mathbf{A}\boldsymbol{\varepsilon}^o_k$, to obtain

$$\frac{\partial \mathbf{N}_k}{\partial t_l} = \frac{\partial \mathbf{A}}{\partial t_l} \boldsymbol{\varepsilon}^o_k + \mathbf{A} \frac{\partial \boldsymbol{\varepsilon}^o_k}{\partial t_l} = 0\,. \tag{11.2.9}$$

Since the **A** matrix is a linear function of the thickness variables (see Eq. 11.1.16), the derivative is simply equal to the transformed reduced stiffnesses of the ith layer

$$\frac{\partial \mathbf{A}}{\partial t_i} = \bar{\mathbf{Q}}_i \,, \tag{11.2.10}$$

so that from Eq. (11.2.9) we have

$$\frac{\partial \varepsilon_k^o}{\partial t_l} = -\mathbf{A}^{-1}\bar{\mathbf{Q}}_l \varepsilon_k^o \,. \tag{11.2.11}$$

Equation (11.2.6) together with (11.2.7) and (11.2.11) can be used to form the linear approximations at any stage of the design optimization.

In addition to the constraint approximation, Schmit and Farshi also used a constraint deletion technique by including only those constraints that are potentially critical at each stage of the constraint approximations.

Table 11.2.1 : Minimum weight laminates with stiffness constraints loaded in axial compression.

Layup and Layer Number	Orientation Angle deg	Initial Design t_i (in.)	Final Design t_f (in.)	Final Design %	Number of Plies (rounded)
$[0/\pm 45/90]_s$					
1	0^o	0.032281	0.018793	28.96	4
2	$+45^o$	0.032281	0.023048	35.52	6
3	-45^o	0.032281	0.023048	35.52	6
4	90^o	0.032281	0.000000	0	0
	$\sum t_i$	0.129124	0.064890		
$[0/\pm 45]_s$					
1	0^o	0.034194	0.012555	21.12	3
2	$+45^o$	0.034194	0.023441	39.44	6
3	-45^o	0.034194	0.023441	39.44	6
	$\sum t_i$	0.102583	0.059438		

Results of optimal designs for various conventional laminates with 0-deg, ±45-deg, and 90-deg ply orientations under various combinations of in-plane normal and shear loads presented in Ref. [6] demonstrate the importance of the choice of laminate stacking sequence on the optimum design. For example, for a laminate under uniaxial stress and limits on shear stiffness, it does make a difference whether we select a $[0/\pm 45/90]_s$ laminate or $[0/\pm 45]_s$ laminate even though at the end of the design iterations the thickness of the 90-deg plies of the first laminate vanishes. Results of these laminates obtained from Ref. [6] are summarized in Table 11.2.1. The final design of the first laminate has a critical strength constraint for the 90-deg

ply. Compared to the second laminate $[0/\pm 45]_s$ it is about 9% thicker due to an additional 0-deg ply required for the first laminate to prevent violation of the strength constraint in the 90-deg layers. In order to achieve a true optimal solution, therefore, the designer has to repeat the optimization process with different laminate definitions, especially by removing the layer(s) that converge to their lower bounds. However, the fact that a layer assumes a value different from its lower bound may not mean that the particular layer is essential for the optimal design. That is, it is quite possible that once a layer with a thickness different from its lower bound is removed, the optimization procedure can resize the remaining layers to achieve a weight lower than the one achieved before. This can make the design procedure difficult, because of the need to try all possible combinations of preselected angles. However, for most practical applications the presence of plies with fibers running in prescribed directions (such as fibers transverse to the load direction) is desirable. Therefore, lower limits which are generally different than zero are imposed, and ply removal is not an option. Multiple load conditions also tend to produce designs where ply removal may not be possible.

Ply Orientation Variables : In order to find the laminate stacking sequence which is best suited to the load condition under consideration, the ply orientations of the laminate as well as the ply thicknesses need to be used as design variables. Indeed many design codes treat both as design variables. In order to demonstrate the use of ply orientations as design variables, however, we concentrate on examples with only ply orientation variables. For optimization problems formulated as minimum-weight designs, the objective function is independent of the ply orientations. This might cause difficulties in converging to an optimum solution with some optimization algorithms. An alternative to the weight objective function minimization is the maximization of the laminate strength as demonstrated by Park [7] and Massard [8].

A quadratic first-ply failure (FPF) criterion based on an approximate failure envelope in the strain space [9] is used by Park [7] for laminates under various in-plane loading conditions (N_x, N_y, N_{xy}). This approximate failure envelope is given by

$$\epsilon_x^2 + \epsilon_y^2 + (1/2)\gamma_{xy}^2 = {b_0}^2 , \tag{11.2.12}$$

where b_0 is defined solely in terms of the stiffness and strength properties in the principal material directions. The objective function to be minimized is defined as

$$f = \epsilon_x^2 + \epsilon_y^2 + (1/2)\gamma_{xy}^2 , \tag{11.2.13}$$

which represents the square of the norm of the strain vector. The smaller the objective function value, the larger the loads that can be applied to the laminate before the failure envelope is violated, therefore, the stronger the laminate in FPF. One key feature of this approximate strain failure envelope is that it applies to laminate strains and does not require ply-level strain calculations. Only balanced symmetric laminates are considered in reference 7, and six different laminates were studied, five of which were the following conventional layups; $[-\theta, +\theta]_s$, $[-\theta, 0, +\theta]_s$, $[-\theta, 90, +\theta]_s$, $[-\theta, 0, 90, +\theta]_s$, and $[-\theta, -45, +45, +\theta]_s$. The sixth laminate was called a continuous laminate, and was assumed to have fiber orientation changing linearly from the top surface to the mid-plane of the laminate covering a range from $-\theta$ to θ-deg. orientations.

Results in [7] showed that under combined loading the best laminate, according to the FPF criterion, for large longitudinal loading without shear is the $[-\theta, 0, +\theta]_s$ type, and for large shear loading without the longitudinal load, the best is the $[-\theta, -45, +45, +\theta]_s$ laminate. The optimum angle for the $[-\theta, 0, +\theta]_s$ laminate depended on the magnitude of the transverse load N_y, and was equal to 0-deg for $N_y = 0$. As the transverse load is increased, the optimum angle reached 45-deg for $N_y = N_x/2$, and was equal to 60-deg for $N_y = N_x$. Similarly, for the $[-\theta, -45, +45, +\theta]_s$ laminate (with shear loading and no axial loading), the optimal angle was 45-deg for $N_y = 0$. As the transverse load N_y increased, the optimal angle increased and reached a value of about 73-deg for $N_y = N_{xy}$. The continuous laminate proved to have the best overall performance under combined longitudinal and shear loadings with a range ± 65-deg for $N_y = 0$.

The above results were intuitively appealing in that the fibers were mostly placed in a direction parallel to the applied loads. But such intuition may not always lead to optimal designs when working with composite materials. Consider, for example, using Hill's yield stress criterion interpreted for composite materials by Tsai [10],

$$f = \left(\frac{\sigma_1}{X}\right)^2 - \left(\frac{\sigma_1\sigma_2}{X^2}\right) + \left(\frac{\sigma_2}{Y}\right)^2 + \left(\frac{\tau_{12}}{S}\right)^2 \leq 1\,, \qquad (11.2.14)$$

for the strength prediction of a unidirectional composite. The quantities X, Y are the normal strengths in directions parallel and transverse to the fibers, and S is the shear strength of a ply. Brandmaier showed [11] that if the transverse normal strength Y is less than the shear strength S, optimal placement of the fibers is not along the principal stress directions, but depends on the values of the strength quantities as well as the applied stresses. This can be demonstrated (see Exercise 1) by expressing the principal stresses in terms of the applied stresses $\sigma_x, \sigma_y, \tau_{xy}$, and the fiber orientation θ, and equating the derivative of Eq. (11.2.14) with respect to the fiber orientation to zero.

A Graphical Tool for Optimum Design : A graphical procedure introduced by Miki [12, 13] for the design of laminates with prescribed in-plane stiffness properties is a highly practical tool for design optimization. The procedure is suitable for multiple balanced angle-ply laminates of the type $[(\pm\theta_I)_{N_I}/(\pm\theta_{I-1})_{N_{I-1}}/\ldots/(\pm\theta_1)_{N_1}]_s$ where the total number of plies in the laminate is $N = 2\sum_{i=1}^{I} N_i$. In addition to the balanced angle-ply sub-laminates, one unidirectional lamina with principal material axes aligned with the axes of the laminate can be included into the stacking sequence.

The major effort of this design procedure is the construction of a lamination parameter diagram which describes the allowable region of lamination parameters V_1^* and V_3^*. These parameters are obtained by normalizing the in-plane components of $V_{1\mathbf{A}}$ and $V_{3\mathbf{A}}$ in Eq. (11.1.21) by the total laminate thickness. For a laminate of total thickness h, in which the volume fraction of the plies with $\pm\theta_i$ orientation angles are v_i, the lamination parameters are given as

$$V_1^* = \frac{V_{1\mathbf{A}}}{h} = \sum_{k=1}^{I} v_k \cos 2\theta_k\,, \quad \text{and} \quad V_3^* = \frac{V_{3\mathbf{A}}}{h} = \sum_{k=1}^{I} v_k \cos 4\theta_k\,, \qquad (11.2.15)$$

where

$$v_i = \frac{2(z_i - z_{i-1})}{h}, \quad \text{and} \quad \sum_{i=1}^{I} v_i = 1 \ . \tag{11.2.16}$$

Because of the normalization, the values of the lamination parameters are always bounded, $-1 \leq V_1^*, V_3^* \leq 1$. For a laminate with only one fiber orientation angle, the lamination parameters are

$$V_1^* = \cos 2\theta, \quad \text{and} \quad V_3^* = \cos 4\theta \,, \tag{11.2.17}$$

and the two parameters are related as

$$V_3^* = 2V_1^{*2} - 1 \ . \tag{11.2.18}$$

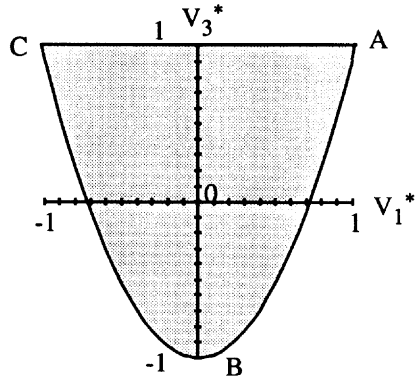


Figure 11.2.1 Lamination parameter diagram of a $[\pm\theta]_s$ laminate.

Values of all possible combinations of the lamination parameters are, therefore, located along the curve ABC in Fig. 11.2.1 described by Eq. (11.2.18). Note that the points A, B, and C correspond to laminates with [0], $[\pm 45]_s$, and [90] ply orientations, respectively. For each design point along the curve ABC the values of the effective engineering elastic constants can be obtained from

$$\begin{aligned} E_x &= \frac{1}{h}\left(\frac{A_{11}A_{22} - A_{12}^2}{A_{22}}\right), \\ E_y &= \frac{1}{h}\left(\frac{A_{11}A_{22} - A_{12}^2}{A_{11}}\right), \\ G_{xy} &= \frac{1}{h}A_{66}, \quad \text{and} \quad \nu_{xy} = \frac{A_{12}}{A_{22}}, \end{aligned} \tag{11.2.19}$$

where the elements of the extensional stiffness matrix of the laminate are determined from the following equations from Table (11.1.1)

$$\begin{Bmatrix} A_{11} \\ A_{22} \\ A_{12} \\ A_{66} \end{Bmatrix} = \begin{bmatrix} U_1 & V_{1\mathbf{A}} & V_{3\mathbf{A}} \\ U_1 & -V_{1\mathbf{A}} & V_{3\mathbf{A}} \\ U_4 & 0 & -V_{3\mathbf{A}} \\ U_5 & 0 & -V_{3\mathbf{A}} \end{bmatrix} \begin{Bmatrix} h \\ U_2 \\ U_3 \end{Bmatrix} , \tag{11.2.20}$$

and where the U_i are the orientation-invariant material properties, Eq. (11.1.7).

If the laminate consists of two or more fiber orientations, then it is shown by Miki [12] that Eq. (11.2.18) becomes an inequality

$$V_3^* \geq 2V_1^{*2} - 1 . \tag{11.2.21}$$

The allowable region of the lamination parameters is the area bounded by the curve ABC in Fig. 11.2.1, independent of the number of different ply orientations. Any point inside the lamination parameter diagram, therefore, corresponds to laminates with two or more fiber orientations. Because a point is defined by two parameters, this means that only two orientation angles θ_1 and θ_2 are sufficient for designing laminates for prescribed stiffness requirements. For balanced angle-ply laminates with more than two orientations, there will be many combinations of the ply orientations that will produce the same lamination parameters and, therefore, the same stiffness properties. Each point inside the design space is called a lamination point, and corresponds to a laminate with a specific stiffness properties. It is also possible to restrict permissible values of the various effective engineering stiffnesses (E_x, E_y, G_{xy}, and ν_{xy}) graphically. This is achieved by introducing contours of constant effective engineering stiffnesses, obtained from Eqs. (11.2.19) and (11.2.20), for each of the engineering constants

$$E_x \text{ contours}: \quad V_3^* = \frac{U_2^2 V_1^{*2} - U_2 E_x V_1^* + E_x U_1 - U_1^2 + U_4^2}{U_3(2U_1 + 2U_4 - E_x)} , \tag{11.2.22}$$

$$E_y \text{ contours}: \quad V_3^* = \frac{U_2^2 V_1^{*2} + U_2 E_y V_1^* + E_y U_1 - U_1^2 + U_4^2}{U_3(2U_1 + 2U_4 - E_y)} , \tag{11.2.23}$$

$$\nu_{xy} \text{ contours}: \quad V_3^* = \frac{\nu_{xy} U_2 V_1^* - \nu_{xy} U_1 + U_4}{(1 + \nu_{xy}) U_3} , \tag{11.2.24}$$

$$G_{xy} \text{ contours}: \quad V_3^* = \frac{U_5 - G_{xy}}{U_3} . \tag{11.2.25}$$

Contours of constant laminate effective engineering properties are shown in Fig. 11.2.2 . The figure indicates that, if no other constraints are specified, the maximum values of the E_x, E_y, and G_{xy} are all achieved for lamination points located around the boundary of the design space which require only one lamination angle. As expected, the maximum E_x and E_y are obtained for 0-deg and 90-deg laminates, and the maximum shear stiffness for $[\pm 45]_s$ laminate. However, determination of the value

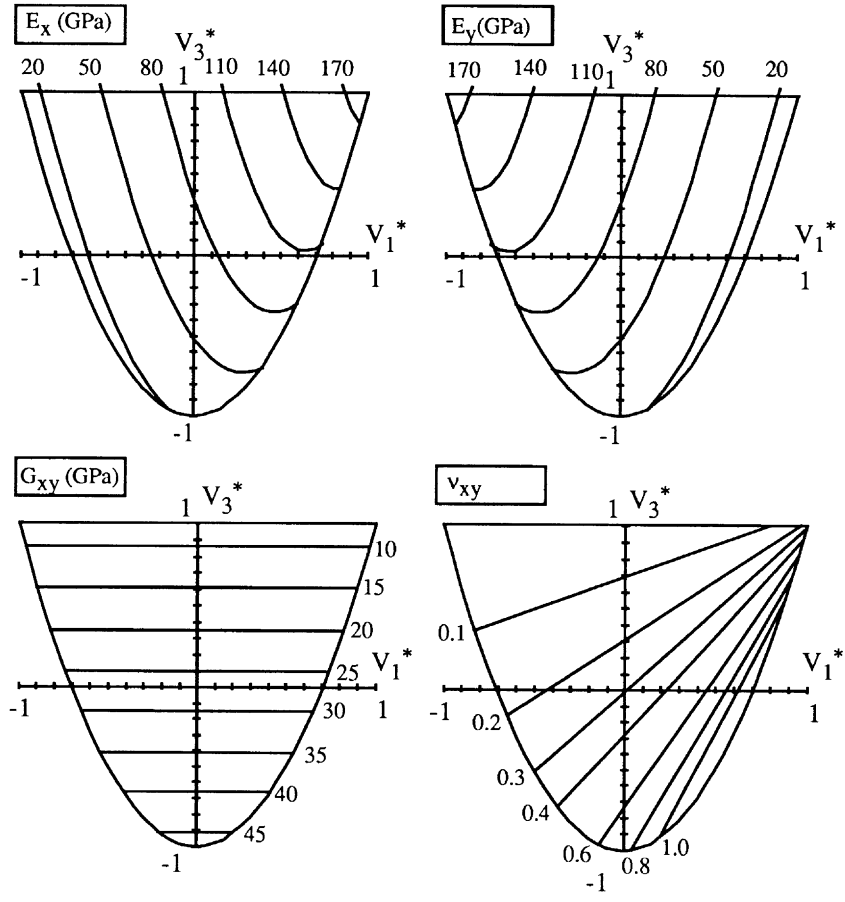


Figure 11.2.2 Contours of constant effective engineering elastic properties.

of lamination angle $[\pm\theta]_s$ that maximizes the effective Poisson's ratio is not straight forward and is a function of the lamina properties via Eqs. (11.1.7) and (11.1.4). For example, for T300/5208 graphite/epoxy and Scotchply 1002 glass/epoxy materials the laminates that produce the maximum Poisson's ratio are $[\pm 25]_s$ and $[\pm 31]_s$, respectively.

For design problems where one or more of the effective engineering constants are constrained, appropriate contours can be superimposed to identify the feasible design space and the lamination point that maximizes (or minimizes) the desired stiffness property (see Exercise 2).

11.2.2 *Design of Laminates for Flexural Response*

Ply Thickness Variables : For rectangular laminated plates under in-plane compressive loads, the strength constraint becomes unimportant if the size of the plate is large compared to the thickness. For such plates, elastic stability and vibration, which are governed by the flexural rigidities of the plate must be considered. One of the earliest studies that included the elastic stability constraint during the optimal design of composite plates is by Schmit and Farshi [14].

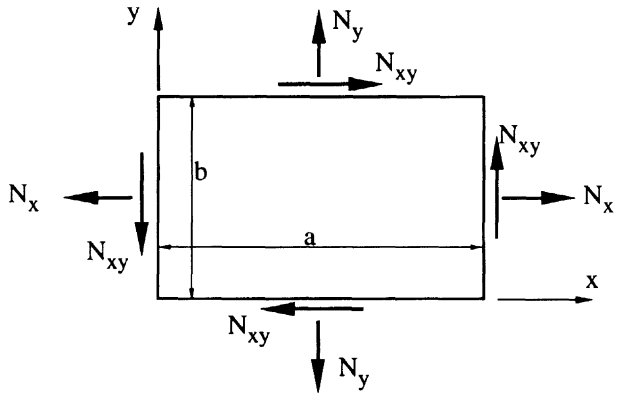


Figure 11.2.3 Laminated panel under in-plane loads.

For a symmetrically laminated, balanced orthotropic laminate with only thickness design variables, the elastic stability constraint is imposed in the form

$$g_b(\mathbf{t}) = 1 - \lambda_b(\mathbf{t}) \geq 0\,, \tag{11.2.26}$$

where $\mathbf{t}$ is a vector of design variables which are the thicknesses of individual layers in a laminate, and λ is the buckling load factor. For a balanced, symmetric laminate of dimensions a by b with simply supported boundaries under-in plane loads an assumed displacement function of the form

$$w(x,y) = \sum_{n=1}^{N}\sum_{m=1}^{M} W_{mn} \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b}\,, \tag{11.2.27}$$

gives $M \times N$ of these constraints representing the $M \times N$ possible modes of buckling associated with the transverse displacement patterns. This form of the displacements satisfies the boundary conditions exactly. A truncated series can be used for an approximate solution of the differential equation governing the buckling of a rectangular orthotropic plate

$$D_{11}\frac{\partial^4 w}{\partial x^4} + 2(D_{12}+2D_{66})\frac{\partial^4 w}{\partial x^2 \partial y^2} + D_{22}\frac{\partial^4 w}{\partial y^4} = \lambda(N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + 2N_{xy}\frac{\partial^2 w}{\partial x \partial y})\,, \tag{11.2.28}$$

where N_x, N_y, and N_{xy} are equal to the applied design loads. Substituting Eq. (11.2.27) into the equilibrium equation and applying Galerkin's method leads to an eigenvalue problem of the form

$$\mathbf{K}\mathbf{w} = \lambda \mathbf{K}_G \mathbf{w}\,, \tag{11.2.29}$$

where the eigenvector is composed of the unknown coefficients of the displacement function, $\mathbf{w} = \{W_{11} \ldots W_{1N}\ W_{21} \ldots W_{2N} \ldots\ldots W_{MN}\}^T$. The matrices $\mathbf{K}$ and $\mathbf{K}_G$ are given as

$$\mathbf{K} = \frac{ab}{4}[\delta_{mp}\delta_{nq} f_{mn}]\,, \qquad \left\{ \begin{matrix} m,p = 1,\ldots,M \\ n,q = 1,\ldots,N \end{matrix} \right\}, \tag{11.2.30}$$

where

$$f_{mn} = \pi^4 \left[D_{11}\left(\frac{m}{a}\right)^4 + 2(D_{12} + 2D_{66})\left(\frac{m}{a}\right)^2\left(\frac{n}{b}\right)^2 + D_{22}\left(\frac{n}{b}\right)^4 \right], \tag{11.2.31}$$

with δ_{mp}, δ_{nq} denoting Kronecker deltas and

$$\mathbf{K}_G = \left[8N_{xy}\xi_{mnpq} - \delta_{mp}\delta_{nq}\frac{\pi^2}{4}\left(\frac{b}{a}m^2 N_x + \frac{a}{b}n^2 N_y\right)\right] \qquad \left\{ \begin{matrix} m,p = 1,\ldots,M \\ n,q = 1,\ldots,N \end{matrix} \right\}, \tag{11.2.32}$$

where

$$\xi_{mnpq} = \left\{ \begin{matrix} 0 \text{ if } p = m \text{ or } q = n \\ \dfrac{mnpq}{(p^2 - m^2)(q^2 - n^2)}\delta^{pm}\delta^{qn} \end{matrix} \right\},$$

$$\delta^{pm} = \left\{ \begin{matrix} 1 \text{ if } (p+m) \text{ odd} \\ 0 \text{ if } (p+m) \text{ even} \end{matrix} \right\}, \quad \text{and} \quad \delta^{qn} = \left\{ \begin{matrix} 1 \text{ if } (q+n) \text{ odd} \\ 0 \text{ if } (q+n) \text{ even} \end{matrix} \right\}.$$

The indices p and q are used as a counter for the equations and m and n are the indices for the coefficients of the W_{mn}'s in each one of the equations. Therefore, no summation is implied over the indices m, n, p, q in the calculation of elements of the two matrices.

For a simply supported plate under biaxial compression loads only ($N_{xy} = 0$), the plate buckles when the load amplitude parameter λ reaches a critical value λ_{cr} given as

$$\lambda_{cr}(m,n) = \frac{\pi^2 \left[D_{11}\left(\frac{m}{a}\right)^4 + 2(D_{12} + 2D_{66})\left(\frac{m}{a}\right)^2\left(\frac{n}{b}\right)^2 + D_{22}\left(\frac{n}{b}\right)^4 \right]}{\left(\frac{m}{a}\right)^2 N_x + \left(\frac{n}{b}\right)^2 N_y}\,, \tag{11.2.33}$$

where m and n are the number of half waves in the x and y directions, respectively, that minimize λ_{cr}.

As they did for the strength constraint Eq. (11.2.2), Schmit and Farshi used a linear approximation for the buckling constraints in the form

$$g_L(\mathbf{t}) = 1 - \lambda_b(\mathbf{t}_0) - \sum_{i=1}^{I}(t_i - t_{0i})\frac{\partial \lambda_b}{\partial t_i}\bigg|_{\mathbf{t}=\mathbf{t}_0}\,. \tag{11.2.34}$$

Noting from Eqs. (11.2.32) that the matrix $\mathbf{K}_G$ is independent of the design variables, and using Eq. 7.3.5, we can show that the derivatives of the kth buckling load factor are given by

$$\frac{\partial \lambda_k}{\partial t_i} = \frac{\mathbf{w}_k^T \frac{\partial \mathbf{K}}{\partial t_i} \mathbf{w}_k}{\mathbf{w}_k^T \mathbf{K}_G \mathbf{w}_k} \,. \tag{11.2.35}$$

Since the matrix $\mathbf{K}$ is a function of the flexural stiffnesses, an explicit expression for the derivatives of $\mathbf{K}$ with respect to the design variables can be written as

$$\frac{\partial k_{pq}}{\partial t_i} = \frac{ab}{4} \left[\delta_{mp} \delta_{nq} \frac{\partial f_{mn}}{\partial t_i} \right] \,, \tag{11.2.36}$$

where

$$\frac{\partial f_{mn}}{\partial t_i} = \pi^4 \left[\frac{\partial D_{11}}{\partial t_i} \left(\frac{m}{a}\right)^4 + 2\left(\frac{\partial D_{12}}{\partial t_i} + 2\frac{\partial D_{66}}{\partial t_i}\right) \left(\frac{m}{a}\right)^2 \left(\frac{n}{b}\right)^2 + \frac{\partial D_{22}}{\partial t_i} \left(\frac{n}{b}\right)^4 \right] \,. \tag{11.2.37}$$

The partial derivatives of the flexural stiffnesses can be related to the partial derivatives of the in-plane stiffness matrix $\mathbf{A}$. For a quasi-homogeneous laminate in which the bending-twisting coupling terms, D_{16} and D_{26}, are ignored (these terms vanish as the number of ply groups increase), the in-plane and flexural moduli are related by (see page 204 of Ref. 9)

$$D_{ij} = \frac{h^2}{12} A_{ij} \,, \tag{11.2.38}$$

where h is the laminate thickness. The partial derivatives of the flexural stiffnesses are, therefore, given by

$$\frac{\partial D_{rs}}{\partial t_i} = \frac{1}{12} \left[\left(\frac{\partial A_{rs}}{\partial t_i} \right) h^2 + 2 A_{rs} h \right] \,, \qquad r,s = 1,2,6 \,, \tag{11.2.39}$$

where the derivative of the $\mathbf{A}$ matrix is given by Eq. (11.2.10).

Graphical Buckling Optimization : Just as with the in-plane lamination diagram discussed earlier, a diagram can be constructed, as shown by Miki [15], for designing laminates for buckling response. We define flexural lamination parameters as

$$W_1^* = \frac{12 V_{1\mathbf{D}}}{h^3} = \sum_{k=1}^{I} s_k \cos 2\theta_k \,, \quad \text{and} \quad W_3^* = \frac{12 V_{3\mathbf{D}}}{h^3} = \sum_{k=1}^{I} s_k \cos 4\theta_k \,, \tag{11.2.40}$$

where $I = N/2$, and

$$s_i = \left(\frac{2z_i}{h}\right)^3 - \left(\frac{2z_{i-1}}{h}\right)^3 \,. \tag{11.2.41}$$

Miki shows [15] that a relation of the same form as Eq. (11.2.21) is obtained

$$W_3^* \geq 2 W_1^{*2} - 1 \,. \tag{11.2.42}$$

Therefore, any balanced symmetric angle-ply laminate with multiple orientations can be represented as a point in a region bounded by

$$W_3^* = 2W_1^{*2} - 1 \ , \tag{11.2.43}$$

where the designs on the boundaries correspond to designs with only one lamination angle, $[(\pm\theta)_I]_s$, and

$$W_1^* = \cos 2\theta, \quad \text{and} \quad W_3^* = \cos 4\theta \ . \tag{11.2.44}$$

The diagram for the flexural lamination parameters can be used for designing laminates for maximum buckling load under uniaxial and biaxial loads. For prescribed values of the m and n, and a fixed ratio of applied transverse load to axial load it can be shown, by manipulating Eq. (11.2.33), that the contours of the critical load parameter λ_{cr} are straight lines in the flexural lamination diagram. However, a difficulty in using flexural lamination parameter in designing laminates with maximum buckling load is that m and n are seldom known a priory. Since these two numbers depend on the design variables, as well as the plate aspect ratio and the applied loads, it is not always possible to predict them accurately. For further discussion of the use of the flexural lamination parameter diagram for buckling maximization see Ref. [15]. Also, the following analytical discussion of the use of ply orientation variables for buckling problem explains the role of m and n.

Ply Orientation Variables : A number of researchers carried out analytical investigations of the optimization of various flexural response quantities such as vibration frequency [16-18], structural compliance [19], and buckling response [20] of simply supported laminated plates. For a plate with length a and width b, Pedersen [20] defined a parameter ϕ which is proportional to the square of natural frequency and buckling load, and inversely proportional to the out-of-plane displacements. The quantity ϕ, composed of a linear combination of the non-dimensional bending stiffnesses d_{ij} $(i, j = 1, 2, 6)$, is defined as

$$\phi = d_{11} + 2\eta^2(d_{12} + 2d_{66}) + \eta^4 d_{22} \, , \tag{11.2.45}$$

where η is a mode parameter defined as the ratio of the longitudinal and transverse half-wave lengths by

$$\eta = \frac{na}{mb} \, , \tag{11.2.46}$$

with m and n being the modal half-wave numbers in the x and y directions, respectively (see Eq. 11.2.27). The non-dimensional bending stiffnesses, d_{ij}, are defined in terms of the flexural stiffnesses as

$$\mathbf{d} = \frac{8(1 - \nu_{12}\nu_{21})}{E_1 h^3}\mathbf{D} \ . \tag{11.2.47}$$

For a laminate with fixed ply thicknesses, the maximization of the buckling load or the natural frequency, or minimization of the displacements, is achieved by obtaining the stationary value of the ϕ with respect to the ply orientations. That is

$$\frac{\partial \phi}{\partial \theta} = \frac{\partial d_{11}}{\partial \theta} + 2\eta^2\left(\frac{\partial d_{12}}{\partial \theta} + 2\frac{\partial d_{66}}{\partial \theta}\right) + \eta^4 \frac{\partial d_{22}}{\partial \theta} = 0 \ . \tag{11.2.48}$$

Restricting the laminate to be a balanced, symmetric angle-ply laminate and ignoring the bending-twisting coupling terms, we can put the bending stiffness matrix from Table 11.1.1 into a summation form

$$\begin{bmatrix} D_{11} \\ D_{22} \\ D_{12} \\ D_{66} \end{bmatrix} = \frac{h^3}{12} \begin{bmatrix} U_1 & W_1^* & W_3^* \\ U_1 & -W_1^* & W_3^* \\ U_4 & 0 & -W_3^* \\ U_5 & 0 & -W_3^* \end{bmatrix} \begin{bmatrix} 1 \\ U_2 \\ U_3 \end{bmatrix}, \tag{11.2.49}$$

where W_1^* and W_3^* are defined by Eq. (11.2.40). Using Eqs. (11.2.40), (11.2.48), and (11.2.49) we have

$$\frac{\partial \phi}{\partial \theta_k} = \frac{2}{3}(z_k^3 - z_{k-1}^3)\Big([U_2(1-\eta^4) + 4U_3(1-6\eta^2+\eta^4)\cos 2\theta]\sin 2\theta\Big)_k . \tag{11.2.50}$$

The stationary values of ϕ correspond to

$$\theta_k = 0, \quad \text{or } |\theta_k| = 90, \tag{11.2.51}$$

$$\text{or} \qquad |\theta_k| = \frac{1}{2}\cos^{-1}\left(\frac{U_2}{4U_3}\frac{(\eta^4-1)}{(1-6\eta^2+\eta^4)}\right) . \tag{11.2.52}$$

The existence of multiple values of the fiber orientation that yield stationary values for the quantity ϕ indicates local optima. The first two roots are independent of the material properties and the geometry. The solution in Eq. (11.2.52), on the other hand, contains the material properties and the mode parameter η, and is valid in a range (see Muc [21]) $\eta^2_{\min} < \eta^2 < \eta^2_{\max}$ where

$$\eta^2_{\min} = \frac{-6 \pm \sqrt{36 + 4[(U_2/4U_3)^2 - 1]}}{2[(U_2/4U_3) - 1]}, \text{ and } \quad \eta^2_{\max} = \frac{6 \pm \sqrt{36 + 4[(U_2/4U_3)^2 - 1]}}{2[(U_2/4U_3) + 1]}, \tag{11.2.53}$$

when θ reaches 0 and 90-deg , respectively.

The optimal values of the fiber angles for two different values of the $U_2/4U_3$ values are presented in Figure 11.2.4 from Eq. (11.2.52). The range of the $U_2/4U_3$ values used in the figure practically covers many commercially available composites including Graphite-Epoxy, Boron-Epoxy, Glass-Epoxy, and Aramid-Epoxy. Clearly the optimal fiber orientation is insensitive to the material properties, but strongly influenced by the mode shape parameter. For small or large values of the mode parameter η the optimal orientation is either θ_k = 0-deg or θ_k = 90-deg, and the optimal orientation is independent of the position of the layer in the laminate.

The influence of the mode parameter η on value of the optimal fiber orientation needs to be investigated further. The minimum value of ϕ which corresponds to the buckling mode shape with lowest buckling load is obtained for transverse wavelength parameter $n = 1$, but it is not always clear what value of the longitudinal wavelength parameter m leads to the lowest value of the parameter ϕ. For plate aspect ratios $r = a/b$ less than a critical value $(r_{cr})_1$ the wave number $m = 1$ gives the lowest value. For $r > (r_{cr})_1$ the wave number is determined such that it minimizes ϕ. The points

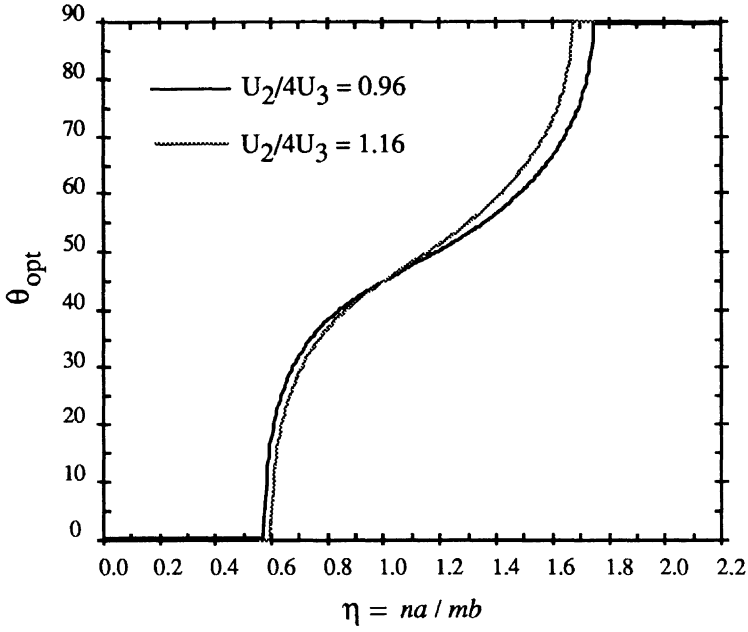


Figure 11.2.4 Optimal ply orientation as a function of the mode parameter η.

points of intersection of the curves of ϕ for mode parameters m and $m+1$ give the critical values of the plate aspect ratio [15]

$$(r_{cr})_m = \left[\frac{m^2\bar{m}^2(U_1 + U_2\cos 2\theta + U_3\cos 4\theta)}{U_1 - U_2\cos 2\theta + U_3\cos 4\theta}\right]^{\frac{1}{4}}, \tag{11.2.54}$$

where $\bar{m} = m+1$ is the wave number of the adjacent mode shape. However, it is demonstrated by Miki [15] that, in the range $(r_{cr})_m < r < (r_{cr})_{\bar{m}}$, laminates designed by assuming the mode shape to be m lead to a laminate which has lower buckling load corresponding to mode $\bar{m}$. Similarly, laminates designed by assuming the mode shape to be $\bar{m}$ give a laminate which has lower buckling load corresponding to mode m. This indicates that at the optimum both buckling loads are the same. In the range of r where two successive modes are simultaneously active, the optimum value of the fiber orientation is determined from $\phi(m) = \phi(\bar{m})$ and is given by

$$\cos 2\theta = \frac{U_2(r^4 + m^2\bar{m}^2) \pm \sqrt{U_2^2(r^4 + m^2\bar{m}^2)^2 - 8U_3(U_1 - U_3)(r^4 - m^2\bar{m}^2)^2}}{4U_3(r^4 - m^2\bar{m}^2)}. \tag{11.2.55}$$

The optimal orientation of the fibers, including the interaction of the adjacent modes, for a T300/5208 Graphite/Epoxy laminate as a function of the plate aspect

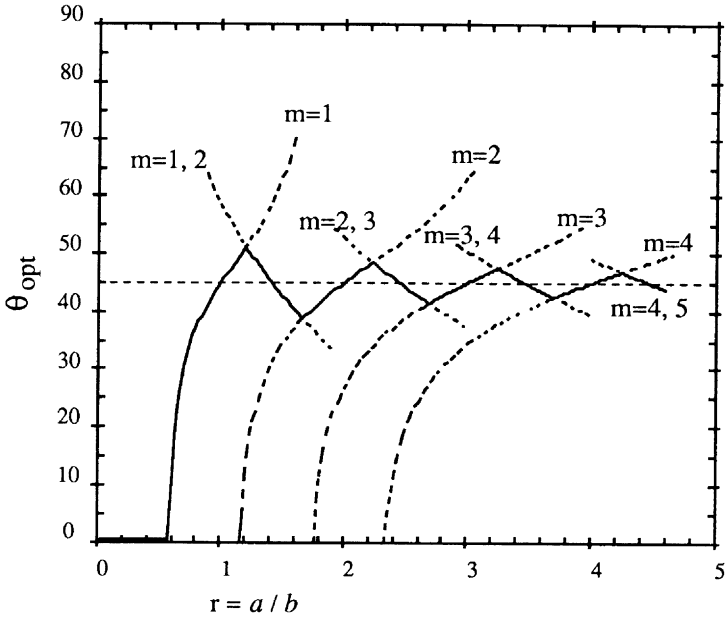


Figure 11.2.5 Optimal ply orientation as a function of plate aspect ratio.

ratio is shown in Figure 11.2.5. For aspect ratios greater than unity, the optimal angle oscillates around 45-deg. The amplitude of the oscillations decreases as the aspect ratio r is increased, therefore, for all practical purposes and for aspect ratios $r > 4$ the optimal angle can be assumed to be $\theta_{\text{opt}} = 45$-deg.

If the laminate is loaded under biaxial compression [20], for small aspect ratios, $r < 1.5$, the optimal fiber angle is similar to the case of uniaxial compression. For aspect ratios larger than 1.5, the value of the optimal angle increases rapidly as the ratio of the transverse load to the axial load (N_y/N_x) increases. For $N_y \geq 4N_x$, the optimal fiber orientation is 90-deg.

Importance of Laminate Stacking Sequence: When ply thickness design variables are used, the stacking sequence is selected ahead of time. As for in-plane loads, the optimum design can be influenced by a choice of whether include or not to include a particular ply orientation. However, for flexural response, the stacking sequence is more important because it strongly affects the **D** matrix while it has no effect on the **A** matrix. Fortunately, as shown below, the optimum design is insensitive to the choice of stacking sequence.

If the relative position of the boundaries between the plies are $\xi_k = z_k/h$ for a laminate with N plies, then

$$-\frac{1}{2} \leq \xi_{k-1} < \xi_k < \xi_{k+1} \leq \frac{1}{2}, \qquad (k = 1, \ldots, N-1)\ ; \tag{11.2.56}$$

the derivative of ϕ with respect to the ply boundary variable is

$$\frac{\partial \phi}{\partial \xi_k} = \frac{\partial d_{11}}{\partial \xi_k} + 2\eta^2\left(\frac{\partial d_{12}}{\partial \xi_k} + 2\frac{\partial d_{66}}{\partial \xi_k}\right) + \eta^4 \frac{\partial d_{22}}{\partial \xi_k} = 0 \,. \tag{11.2.57}$$

Since the contribution of the individual layers to the overall **D** matrix depends only on the distance of the layers to the laminate mid-plane, the derivative of the **D** matrix is expressed as

$$\frac{\partial D_{ij}}{\partial \xi_k} = \xi_k^2 (D_{ij_k} - D_{ij_{k+1}}) \,. \tag{11.2.58}$$

Here D_{ij_k} depends only on the properties and orientation of the k-th layer and (assuming the adjacent layers to be made of the same material so that the constant U terms are omitted) is defined by

$$\mathbf{D}_k = h^3 \begin{bmatrix} U_2 \cos 2\theta_k + U_3 \cos 4\theta_k & -U_3 \cos 4\theta_k & 0 \\ -U_3 \cos 4\theta_k & -U_2 \cos 2\theta_k + U_3 \cos 4\theta_k & 0 \\ 0 & 0 & -U_3 \cos 4\theta_k \end{bmatrix} . \tag{11.2.59}$$

Then, as shown by Cheng Kengtung [19] the derivative of the function ϕ can be expressed as

$$\frac{\partial \phi}{\partial \xi_k} = 2(\xi_k^2)\Big[-U_2(1-\eta^4) - 2U_3(1-6\eta^2+\eta^4)(\cos 2\theta_k + \cos 2\theta_{k+1})\Big](\cos 2\theta_k - \cos 2\theta_{k+1}) \,. \tag{11.2.60}$$

Since the sign of the derivative of ϕ is independent of the position of the boundary, we choose either the minimum or the maximum thickness for the k-th ply depending on the sign of the derivative. For example, if $\frac{\partial \phi}{\partial \xi_k} > 0$ we will use $\xi_k = \xi_{k\max}$ in order to maximize the buckling load. Furthermore, some specific combinations of the neighboring ply orientations lead to stationary values for the ϕ, see Eq. (11.2.60), indicating possible local minima. These roots are

$$|\theta_k| = |\theta_{k+1}| \,, \qquad \text{and} \tag{11.2.61}$$

$$\cos 2\theta_k + \cos 2\theta_{k+1} = \frac{1}{2}\frac{U_2}{U_3}\frac{(\eta^4 - 1)}{(1 - 6\eta^2 + \eta^4)} \,. \tag{11.2.62}$$

If the total thickness of the two plies is kept constant, the derivative is zero for these angles whatever the location of the boundary between the plies. Therefore, the buckling load is independent of the thickness distribution of the adjacent plies. Moreover, for a square laminate, $\eta = 1$, ϕ is constant for

$$\theta_{k+1} = \frac{\pi}{2} - \theta_k \,, \tag{11.2.63}$$

whatever the material properties.

Shin et al. showed [22] that for a symmetric laminate with fixed total thickness, the order of ply orientations can also be permitted in any desired way without changing the **D** matrix (see exercise 3). The individual ply thicknesses do change,

of course. In practice, the requirement of an integer number of plies forces changes in the **D** matrix, but if the total thickness is large compared to angle-ply thickness this effect will be small. This is demonstrated in Table 11.2.2, taken from [22], which presents six permutations of a plate made of 0, 90 and 45 plies. The total thickness for all six permutations is the same (normalized to one) and all have the same **D** matrix and the same buckling load. If the total number of plies is 50 the buckling loads of the six laminates vary by less then one percent (see Ref. 22).

Table 11.2.2 : Optimum Designs with Equivalent **D** Matrix

StackingSequence	t_1	$(t_1^\dagger)$	t_2	$(t_2^\dagger)$	t_3	$(t_3^\dagger)$
$[0/90/45]_s$	0.0366	(0.04)	0.1539	(0.16)	0.8095	(0.80)
$[0/45/90]_s$	0.0366	(0.04)	0.2496	(0.24)	0.7139	(0.72)
$[45/0/90]_s$	0.2228	(0.20)	0.0634	(0.08)	0.7139	(0.72)
$[45/90/0]_s$	0.2228	(0.20)	0.3044	(0.32)	0.4729	(0.48)
$[90/45/0]_s$	0.1399	(0.12)	0.3872	(0.40)	0.4729	(0.48)
$[90/0/45]_s$	0.1399	(0.12)	0.0506	(0.04)	0.8095	(0.84)

† Ply thicknesses are rounded such that each laminate has a total of 50 plies.

The insensitivity of the design to the choice of stacking sequence disappears when strength is also a consideration. In such cases the choice of stacking sequence is critical, and this topic is discussed in the next section.

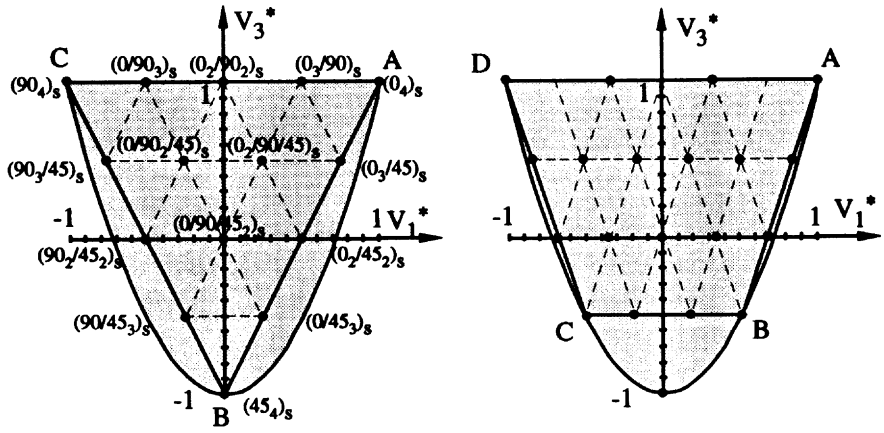
11.3 Stacking Sequence Design

The methods presented in the previous section yield results that are valuable for understanding the basic trends in laminate design. However, one of the major difficulties of a realistic design situation is the need for a practical laminate which is generally made up of plies with only 0-deg, 90-deg and ±45-deg orientations (or occasionally orientations with 15-deg increments between 0- and 90-deg), and thicknesses which are integer multiples of the ply thickness. Of course, deciding the number of plies of a specified orientation is not sufficient to define a laminate, but through-the-thickness location of the ply must be decided as well. This means that the basic design problem is to determine the stacking sequence of the composite laminate—a problem which calls for discrete programming techniques. In the following, we introduce various approaches that address this problem.

11.3.1 *Graphical Stacking Sequence Design*

The lamination parameter diagrams introduced in section 11.2 can be used for designing laminates with predetermined ply orientation angles. It is shown by Miki and Sugiyama [23] that the feasible region for laminates with fixed ply angles is a polygon with vertices located on the envelope of the lamination parameter diagram. If the design point is on the periphery of the diagram, the laminate is an angle ply

laminate with one fiber orientation. Therefore, given a set of permissible integer ply orientations, vertices of the polygons are placed at those locations that correspond to the selected angles. For example, the design spaces for laminates made up of plies with 0-deg, ±45-deg, and 90-deg orientations and 0-deg, ±30-deg, ±60-deg, and 90-deg orientations are shown in Fig. 11.3.1-a and 11.3.1-b, respectively. For laminates with 0, ±45, and 90-deg plies, the design space is a triangle with vertices at A, B, and C as shown in the figure. For ply orientations of 0-deg, ±30-deg, ±60-deg, and 90-deg, the design space is a trapezoid.



a) 0-, ±45-, and 90-deg plies b) 0-, ±30-, ±60-, and 90-deg plies

Figure 11.3.1 In-plane lamination diagram for laminates with integer ply orientations.

Points along the edges and interior points of the polygons correspond to laminates with combinations of two or more ply orientations, and their number is determined by the total number of layers in the laminate. If the total number of layers is N and $I = N/2$, then in addition to the vertices, we obtain $I - 1$ equally spaced design points along the edges and along the internal lines that join two vertices. From the nodes we obtained along the edges, we also draw lines parallel to the lines that join vertices. If such a line terminates at another discrete design point at the opposite end of the polygon, then it is easy to label the design that would be in the interior by looking at the designs at the two end points. For example, for an eight-ply (total) laminate with 0-deg, ±45-deg, and 90-deg angles (triangular design space), there are five equally spaced design points with fiber orientations varying incrementally from one vertex to another as shown in Fig. 11.3.1-a. Note that the design points inside the triangular region also follow an incremental pattern, but are combinations of the three available angles. Design points for a laminate with total six layers are shown in Fig. 11.3.1-b. Labeling of those designs is left to the reader (see exercise 4).

Just as for the in-plane lamination diagram, it is possible to construct the flexural lamination diagram for a laminate with prescribed fiber orientations. The boundaries

of the design space are same as the in-plane parameters; the prescribed angles are on the envelope of the lamination diagram and form the vertices of a polygon. However in this case the design points, which are combinations of the given angles, are not equally spaced (although combinations of the angles corresponding to two vertices are still located along the edge that connect these vertices) but are located through the use of Eq. (11.2.40).

11.3.2 *Penalty Function Formulation*

Buckling Design : The procedure described in section 5.7.4 for the use of a penalty function to achieve designs with discrete valued variables is demonstrated in this section for buckling maximization of laminates with fiber orientation variables. In order to establish results that can be used to compare with integer orientation designs, a series of results was generated for the continuous problems, see Gürdal and Haftka [24]. This was achieved by turning off the penalty terms for the non-discrete values of the design variables.

The problems solved are for $a = 20$ *in* by $b = 10$ *in* ($50.8\ cm \times 25.4\ cm$) rectangular plates of specified numbers of plies and fiber orientation design variables. The critical eigenvalues are maximized for applied compressive load of $N_x = 1$ with varying N_y/N_x ratios.

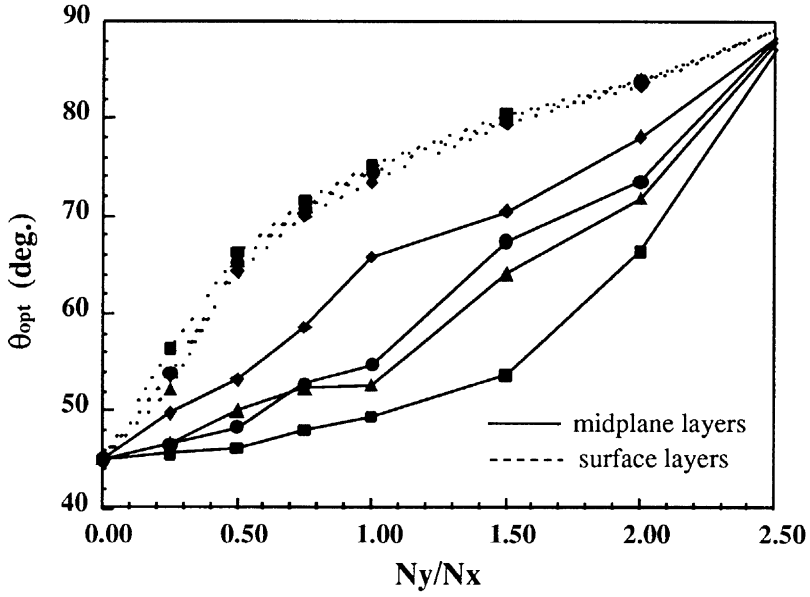


Figure 11.3.2 Optimum continuous fiber orientations for maximum buckling load.

Plates with four different thicknesses corresponding to 8, 12, 16, and 24 ply laminates were designed. The optimal orientations of the surface layer fibers (indicated

by dashed lines) and the layer adjacent to the mid-plane (solid lines) are shown in Fig. 11.3.2 for each of the four laminates. For uniaxial compression, $N_y = 0$ or $N_x = 0$ (or $N_y/N_x > 2.5$), the laminates have the same fiber orientation through the entire thickness which are ±45-deg and 90-deg, respectively. For intermediate load ratios, the fiber angles at the surface layer are larger than the mid-plane layers with the difference being largest for the thick 24-ply laminates. However, the fiber orientation of the surface layers appears to depend only on the load ratio, and not on the laminate thickness.

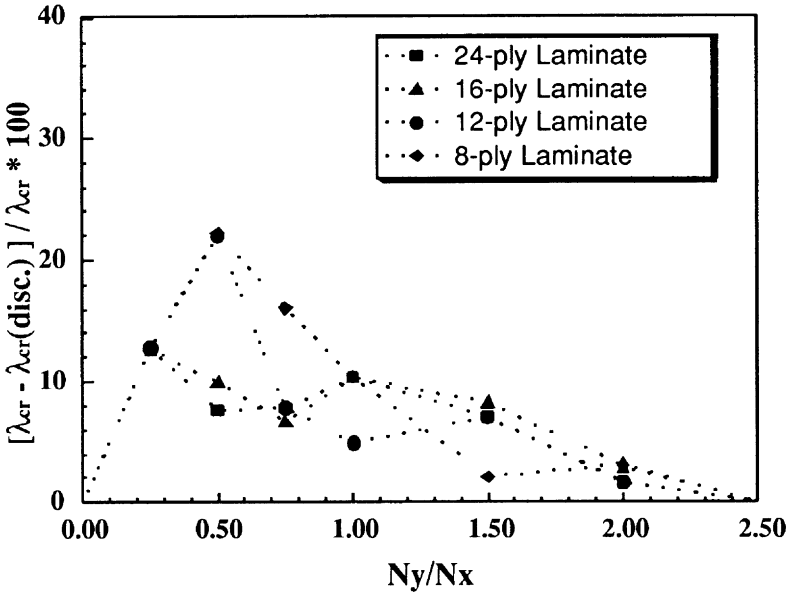


Figure 11.3.3 Buckling load reduction for laminates with 0-deg, ±45-deg, and 90-deg plies.

Next, the same design cases were repeated using discrete fiber orientations of 0-, ±45-, and 90-deg. Solutions were obtained with the penalty function approach, and checked by the branch-and-bound approach described in section 11.3.3. Plies with +45-deg orientation were required to be adjacent to −45-deg plies so as to minimize bending-twisting coupling. For the penalty function approach, it was convenient to require also the plies with 0- and 90-deg orientations to appear in pairs. Plots of the percentage reduction in buckling load due to the restrictions to discrete orientations are shown in Fig. 11.3.3 for the four laminates. Discrete valued designs are accompanied with a substantial buckling load reduction over at least a portion of the load ratio range considered. The largest penalty was for $N_y/N_x = 0.5$ (about 22% reduction), and the thin 8-ply and 12-ply laminates. However, buckling load reductions associated with different thicknesses appeared to be quite random.

The laminate stacking sequences obtained for the discrete valued designs are

Table 11.3.1 : Optimum stacking sequence for 8-ply laminates under biaxial compression.

$\frac{N_y}{N_x}$	Continuous Optima	Penalty Approach	Global Optima
0.0	$[\pm 45]_{2s}$	$[\pm 45]_{2s}$	-
0.25	$[\pm 53.7/\pm 49.8]_s$	$[\pm 45]_{2s}$	$[\pm 45/90_2]_s$
0.50	$[\pm 64.3/\pm 53.2]_s$	$[\pm 45]_{2s}$	$[\pm 45/90_2]_s$
0.75	$[\pm 70.0/\pm 58.6]_s$	$[90_2/\pm 45]_s$	-
1.00	$[\pm 73.5/\pm 65.8]_s$	$[90_2/\pm 45]_s$	-
1.50	$[\pm 79.4/\pm 70.5]_s$	$[90_2/\pm 45]_s$	-
2.00	$[\pm 83.4/\pm 78.1]_s$	$[90_2/\pm 45]_s$	-
2.50	$[\pm 89.2/\pm 88.4]_s$	$[90_4]_{4s}$	-

Table 11.3.2 : Optimum stacking sequence for 16-ply laminates under biaxial compression.

$\frac{N_y}{N_x}$	Continuous Optima	Penalty Approach	Global Optima
0.0	$[\pm 45]_{4s}$	$[\pm 45]_{4s}$	-
0.25	$[\pm 52.2/\cdots/\pm 46.5]_s$	$[\pm 45]_{4s}$	$[\pm 45_2/90_4]_s$
0.50	$[\pm 65.3/\cdots/\pm 60.0]_s$	$[90_2 \pm 45_3]_s$	$[\pm 45/90_6]_s$
0.75	$[\pm 70.9/\cdots/\pm 52.3]_s$	$[90_2/\pm 45_3]_s$	$[90_2/\pm 45_2/90_2]_s$
1.00	$[\pm 74.9/\cdots/\pm 52.6]_s$	$[90_4/\pm 45_2]_s$	$[90_2/\pm 45/90_4]_s$
1.50	$[\pm 80.0/\cdots/\pm 64.1]_s$	$[90_6/\pm 45]_s$	$[90_4/\pm 45_2]_s$
2.00	$[\pm 83.9/\cdots/\pm 71.8]_s$	$[90_6/\pm 45]_s$	$[90_4/\pm 45/90_2]_s$
2.50	$[\pm 89.2/\cdots/\pm 87.9]_s$	$[90]_{8s}$	-

presented in Table 11.3.1 and 11.3.2 for the 8-ply and the 16-ply laminates. Included in the table are the laminate stacking sequences for the continuous valued designs, the discrete designs obtained by using the modified penalty method, and the global optimal designs. If the design obtained by the penalty function approach is same as the global optimal design, the entry under the Global Optima column is left blank. The penalty approach is unable to reach the global optimum in some cases, especially for laminates with large numbers of plies. In every case, the discrete designs obtained by the penalty function approach followed a pattern such that the orientations of the outer plies were larger than those plies close to the mid-plane; this was similar to the trend observed for the continuous designs. Global optimal designs, on the other hand, had orientations that were more random. The differences in buckling loads ranged up to 14%, and illustrate the danger of looking for the discrete optimum near the continuous one.

11.3.3 *Integer Linear Programming Formulation*

The normalized integrals used for the graphical procedure as design variables, see Eqs. (11.2.14) and (11.2.39), may not be a good choice for more general design problems. In order to define the integrals that are needed for characterizing the laminate, a new set of variables that define the existence of a given orientation layer or the orientation of a specified layer are proposed by Haftka and Walsh [25]. Such variables are referred

to as ply-identity design variables. For example, if we have four possible orientations and N plies, we can use N design variables that take the values of 1 to 4 to define the stacking sequence. If symmetry is used this number can be reduced to $N/2$.

It is also possible to use zero-one ply identity design variables. For example, if the laminate is made up of 0-deg, 90-deg, and ±45-deg plies the stacking sequence can be defined in terms of four sets of ply-orientation-identity variables o_i, n_i, f_i^p and f_i^m, $i = 1, \cdots, N/2$, that are zero-one integer variables. The variables o_i, n_i, f_i^p or f_i^m is equal to one if there is a 0-deg, 90-deg, 45-deg or -45-deg ply, respectively, in the ith layer.

The advantage of these zero-one ply-identity variables is that the integrals, and therefore the **A** and **D** matrices are linear functions of these variables. The integrals $V_{0\mathbf{A}}$, $V_{1\mathbf{A}}$ and $V_{3\mathbf{A}}$ are given in terms of the ply identity variables and the thickness of a single ply t as

$$\begin{aligned} V_{0\mathbf{A}} &= \int_{-h/2}^{h/2} dz = 2t\sum_{k=1}^{N/2}(o_k + n_k + f_k^p + f_k^m), \\ V_{1\mathbf{A}} &= \int_{-h/2}^{h/2} \cos 2\theta dz = 2t\sum_{k=1}^{N/2}(o_k - n_k), \\ V_{3\mathbf{A}} &= \int_{-h/2}^{h/2} \cos 4\theta dz = 2t\sum_{k=1}^{N/2}(o_k + n_k - f_k^p - f_k^m). \end{aligned} \tag{11.3.1}$$

For the flexural response, the integrals $V_{0\mathbf{D}}$, $V_{1\mathbf{D}}$ and $V_{3\mathbf{D}}$ are expressed as

$$\begin{aligned} V_{0\mathbf{D}} &= \frac{2t^3}{3}\sum_{k=1}^{N/2} p_k[(\frac{z_k}{t})^3 - (\frac{z_{k-1}}{t})^3] = \frac{2t^3}{3}\sum_{k=1}^{N/2}[k^3 - (k-1)^3](o_k + n_k + f_k^p + f_k^m), \\ V_{1\mathbf{D}} &= \frac{2t^3}{3}\sum_{k=1}^{N/2} p_k \cos 2\theta_k[(\frac{z_k}{t})^3 - (\frac{z_{k-1}}{t})^3] = \frac{2t^3}{3}\sum_{k=1}^{N/2}[k^3 - (k-1)^3](o_k - n_k), \\ V_{3\mathbf{D}} &= \frac{2t^3}{3}\sum_{k=1}^{N/2} p_k \cos 4\theta_k[(\frac{z_k}{t})^3 - (\frac{z_{k-1}}{t})^3] = \frac{2t^3}{3}\sum_{k=1}^{N/2}[k^3 - (k-1)^3](o_k + n_k - f_k^p - f_k^m), \end{aligned} \tag{11.3.2}$$

where f_k^p and f_k^m do not appear in the expression for $V_{1\mathbf{A}}$ and $V_{1\mathbf{D}}$ since the cosine of 90 degrees is equal to zero. The variable p_k in Eq. (11.3.2) is unity if the kth ply is occupied and zero if it is empty. Constraints are applied during the optimization to ensure that p_k can be zero only for the outermost plies.

Stacking Sequence for Buckling Design : Since the buckling load for symmetric laminates under biaxial loads is a linear function of the flexural lamination parameters which are linear functions of the ply-identity variables (see Eqs. (11.2.32) and

(11.2.48)), the problem can, therefore, be posed as a linear integer programming problem.

Two formulations for the optimization problem are possible. The first is the optimization of a laminate with a fixed thickness for maximum buckling load, and the second is the optimization of a laminate for minimum thickness for a given buckling load. For the first optimization problem the lowest buckling load λ^* is maximized, over values of m and n. The objective λ^* is not a smooth function of the design variables, and the standard device (see section 2.4) for removing this problem is to add λ^* as a design variable and require it to be less than or equal to each $\lambda_{cr}(m,n)$. Thus, the optimization problem is formulated as

$$\begin{aligned}
&\textbf{find} && \lambda^*, \quad \text{and} \quad o_i,\ n_i,\ f_i^p,\ f_i^m, \quad i=1,\cdots,N/2\,,\\
&\textbf{to maximize} && \lambda^*\\
&\textbf{such that} && \lambda^* \le \lambda_{cr}(m,n), \qquad m=1,\cdots,m_f\,, \quad n=1,\cdots,n_f\,,\\
&&& o_i+n_i+f_i^p+f_i^m=1\,, \qquad i=1,\cdots,N/2\,,\\
&\textbf{and} && \sum_{i=1}^{N/2} f_i^p - f_i^m = 0\,.
\end{aligned} \tag{11.3.3}$$

The minimization over m and n is performed by checking for all values of m between 1 and m_f, and all values of n between 1 and n_f. The last constraint in Eq. (11.3.3) ensures that the number of 45-deg and -45-deg plies is the same, so that the laminate is balanced. The optimization problem of Eq. (11.3.3) is a integer linear programming problem, and the methods described in chapter 3.9 can be applied.

For the dual problem of weight minimization of a laminate capable of sustaining a specified load without buckling, the total number of layers must be variable. This seems to contradict the use of ply-identity variables which requires N to be known in advance. A remedy for this contradiction is to start with a number of layers large enough so that the initial design does not buckle, but permit some of the plies to be empty ($o_i+n_i+f_i^p+f_i^m \le 1$). Of course, plies that are permitted to be empty must be the outer plies of the laminate in order to maintain integrity of the laminate. The formulation takes the form

$$\begin{aligned}
&\textbf{find} && o_i,\ n_i,\ f_i^p,\ f_i^m, \quad i=1,\cdots,N/2\,,\\
&\textbf{to minimize} && \sum_{i=1}^{N/2}(o_i+n_i+f_i^p+f_i^m)\\
&\textbf{such that} && \lambda_{cr}(m,n) \ge 1, \qquad m=1,\cdots,m_f\,, \quad n=1,\cdots,n_f\,,\\
&&& o_i+n_i+f_i^p+f_i^m \le 1\,, \qquad i=1,\cdots,N/2\,,\\
&&& \sum_{i=1}^{N/2} f_i^p - f_i^m = 0\,,\\
&\textbf{and} && o_i+n_i+f_i^p+f_i^m \le o_{i-1}+n_{i-1}+f_{i-1}^p+f_{i-1}^m\,.
\end{aligned} \tag{11.3.4}$$

where the last constraint ensures that the empty plies are on the outside.

In general, the solution of the weight minimization problem is not unique. For a minimum weight design with N^* layers, it is possible to change the orientations of the fibers and come up with designs that will have the same weight but different buckling loads. Out of those feasible designs, ideally, one would like to choose the one that has the largest margin for the buckling constraint. This can be achieved by subtracting a small fraction of λ_{cr} from the objective function, so that the modified objective function serves the dual purpose of minimizing weight while maximizing the buckling load. For results on weight minimization designs, the reader is referred to Haftka and Walsh [25]. In the following paragraphs, results for buckling maximization will be presented.

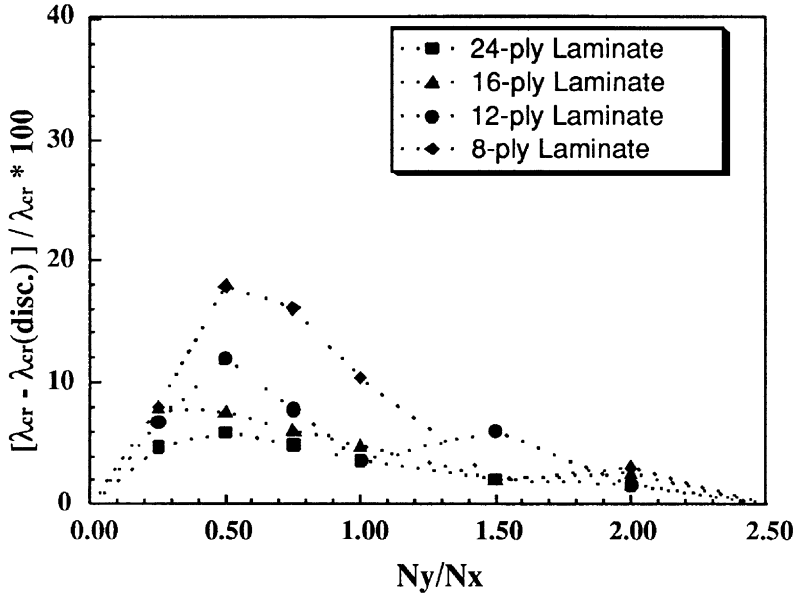


Figure 11.3.4 Buckling load reduction for globally optimal laminates with 0*-deg,* ±45*-deg, and* 90*-deg plies.*

For the results presented in this section the solution of Eqs. (11.3.3) is generated with the LINDO program [26] which employs the branch-and-bound algorithm described in section 3.9.1. First we present the biaxial load cases that were reported earlier in Table 11.3.1 and 11.3.1 as global optima. A plot similar to the plot shown in Fig. 11.3.3, this time for the global optimum designs obtained through the use of the linear integer programming approach, is shown in Fig. 11.3.4 for comparison. In general, there is a small amount of improvement in the buckling load reduction for most of the laminates. For example, the worst buckling load reduction (compared to the continuous designs) is still for the 8-ply laminate for a load ratio of $N_y/N_x = 0.5$, but it is only about 18% as compared to 22%. Also, there is an orderly progression with increasing laminate thickness. The smallest and the largest buckling load

reductions are associated with the 24-ply and the 8-ply laminates, respectively.

When the number of contiguous plies with the same orientation angle is large, composite laminates are known to experience matrix cracking. Therefore, it is desirable to limit the number of such contiguous plies. We demonstrate the use of such a constraints on the design obtained for $N_y/N_x = 2$. We start with the design that was presented in Table 11.3.2, $[90_4/\pm 45/90_2]_s$, which we imposed the constraint that the plies with different orientations appear in pairs. The critical load factor for this optimal design was $\lambda_{cr} = 36.19$. Next, we relax this requirement and redesign the plate so that we can have single plies with different orientations adjacent to one another. This yields a design which has 5 contiguous 90-deg plies, $[90_5/+45/-45/90]_s$. The critical load factor for this design is $\lambda_{cr} = 36.84$, a 1.8% increase compared to the design which restricts each orientation to be in pairs. The fact that 45-deg plies appear in a pair is of course coincidental. We then implement the contiguous ply requirement by adding the constraint

$$n_4 + n_5 + n_6 + n_7 + n_8 \leq 4\,. \tag{11.3.5}$$

The design obtained with this constraint is $[90_4/+45/90_2/-45]_s$ and has a slightly smaller load factor, $\lambda_{cr} = 36.59$, compared to the previous design. However, it still has a slightly larger load factor compared to the design from Table 11.3.2, but violates the requirements that off-axis angles appear in pairs. By introducing a constraint of the form

$$\begin{gathered} f_i^p - f_{i+1}^m = 0\,, \qquad i = 1, 2, \ldots, (I-1)\,, \\ \text{where } f_1^m = 0, \quad \text{and } f_I^p = 0\,, \end{gathered} \tag{11.3.6}$$

designs that have the 45-deg plies in plus and minus pairs can be achieved, without requiring the 0- and the 90-deg plies to be in pairs, and without exceeding 4 contiguous plies with the same orientations. In this particular case we obtain again the design presented in Table 11.3.2.

Stiffness and Buckling Design : In some cases it may be desirable to impose constraints on the stiffness of the plate. For example, a constraint requiring A_{11} to have a minimum value of A_{11}^0 can be written as

$$A_{11}/A_{11}^0 - 1 \geq 0\,. \tag{11.3.7}$$

As shown in [25] this constraint can be expressed as a linear function of the ply identity design variables similar to the buckling constraint. Therefore, it can be used as a constraint in the problem formulated by Eqs. (11.3.3). The effect of introducing a minimum stiffness requirement is checked for $N_y/N_x = 2$. The optimum laminate for this case, was dominated by 90-deg plies, and has only 16 percent of the axial stiffness A_{11} of an all 0-deg laminate. A requirement that A_{11} be at least 50 percent of the unidirectional laminate was added, with and without the requirement of no more than four contiguous plies. The results are compared to the original design in Fig. 11.3.5. It is seen that the stiffness requirement is satisfied by putting 0-deg plies near the plane of symmetry where they have only a minimal effect on the bending stiffnesses, and hence on the buckling load. The reduction in the buckling load is about 8 percent. For this design the effect of adding the requirements of no more

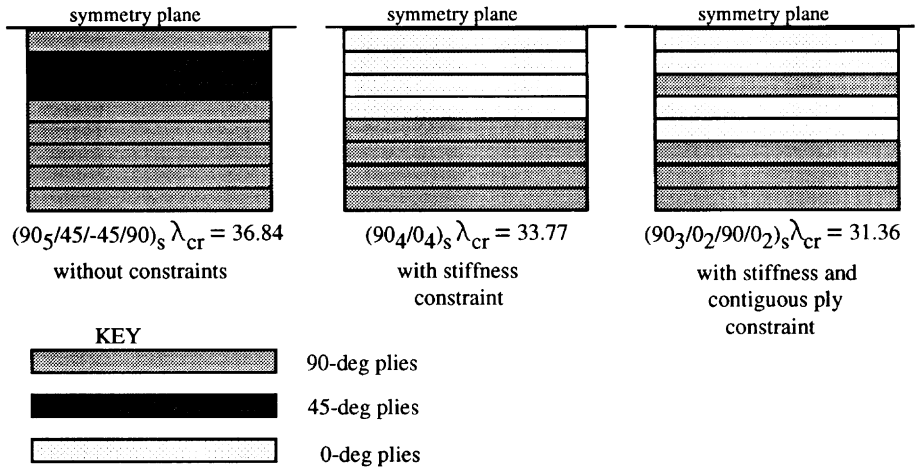


Figure 11.3.5 Effect of stiffness requirement on laminate design.

than 4 contiguous plies had a substantial effect (7 percent reduction) on the buckling load.

Stacking Sequence for Strength and Buckling Design : In the absence of applied shear loads, the laminate strains ϵ_x and ϵ_y can be calculated (for a load factor $\lambda = 1$) from

$$\epsilon_x = \frac{(A_{22}N_x - A_{12}N_y)}{(A_{11}A_{22} - A_{12}^2)}, \quad \text{and} \quad \epsilon_y = \frac{(A_{11}N_y - A_{12}N_x)}{(A_{11}A_{22} - A_{12}^2)}. \tag{11.3.8}$$

The strains for the kth ply may be calculated from the transformation

$$\begin{aligned} \epsilon_1^k &= \cos^2\theta_k\epsilon_x + \sin^2\theta_k\epsilon_y, \\ \epsilon_2^k &= \sin^2\theta_k\epsilon_x + \cos^2\theta_k\epsilon_y, \\ \gamma_{12}^k &= \sin 2\theta_k(\epsilon_y - \epsilon_x). \end{aligned} \tag{11.3.9}$$

Even though the extensional stiffnesses A_{ij} are linear functions of the design variables the strains calculated by Eq.(11.3.8), are nonlinear functions of these variables. These strains can be linearized, as shown by Nagendra et al. [27], by a linear Taylor series in A_{ij}. We have

$$\begin{aligned} \epsilon_L(\mathbf{x}) = \epsilon(\mathbf{x}_o) + \left(\frac{\partial\epsilon}{\partial A_{11}}\right)_{\mathbf{x}_o}(A_{11} - A_{11}^{\mathbf{x}_o}) + \left(\frac{\partial\epsilon}{\partial A_{12}}\right)_{\mathbf{x}_o}(A_{12} - A_{12}^{\mathbf{x}_o}) \\ + \left(\frac{\partial\epsilon}{\partial A_{22}}\right)_{\mathbf{x}_o}(A_{22} - A_{22}^{\mathbf{x}_o}), \end{aligned} \tag{11.3.10}$$

where ϵ is a typical strain component ($\lambda = 1$), ϵ_L is its linear approximation, and $A_{ij}^{\mathbf{x}_o}$ and A_{ij} are the extensional stiffnesses calculated at the nominal design point $\mathbf{x}_o$ and neighboring designs, respectively. The derivatives of the strain with respective to the extensional stiffnesses at the nominal design point are calculated in terms of the midplane strains and the extensional stiffnesses at the nominal design. The linear strain approximation can thus be constructed along a particular fiber orientation and transverse to it by evaluating the strains ϵ_1^k, ϵ_2^k and γ_{12}^k for each orientation (since the orientation is chosen apriori, either 0° or 45°) in terms of the midplane strains using Eq. (11.3.9). For example, the strains along and transverse to the 45° fibers and in shear can be derived as

$$\begin{aligned} \epsilon_1 \;=\; \epsilon_2 = \frac{1}{2}[\epsilon_x + \epsilon_y] \;&=\; \frac{1}{2}\left[\frac{(A_{22} - A_{12})N_x + (A_{11} - A_{12})N_y}{(A_{11}A_{22} - A_{12}^2)}\right], \\ \gamma_{12} \;=\; [\epsilon_y - \epsilon_x] \;&=\; \left[\frac{(A_{11} + A_{12})N_y - (A_{22} + A_{12})N_x}{(A_{11}A_{22} - A_{12}^2)}\right]. \end{aligned} \tag{11.3.11}$$

The derivatives needed for the strain approximation of Eq.(11.3.10) can then be obtained by differentiating Eq.(11.3.11). For example, the derivative of the strain along the 45° fiber with respect to A_{11} can be written as

$$\frac{\partial \epsilon_1}{\partial A_{11}} = \frac{1}{2}\frac{(A_{12} - A_{22})}{(A_{11}A_{22} - A_{12}^2)}\,\epsilon_x\,, \tag{11.3.12}$$

where A_{ij} are the extensional stiffnesses at the nominal design point. Similar strain derivatives with respect to A_{22} and A_{12} can be derived. The extensional stiffnesses are a linear function of the ply-identity design variables, thus the strain approximation is a linear function of the ply-identity variables. It is also important to note that the strains are initially calculated based on some reference value of the load. In order to implement the strain constraint they have to be multiplied by the value of the buckling load multiplier λ_c which is also a function of the design variables,

$$\lambda_c \epsilon_i \;\leq\; \epsilon_i^{ua}\,, \tag{11.3.13}$$

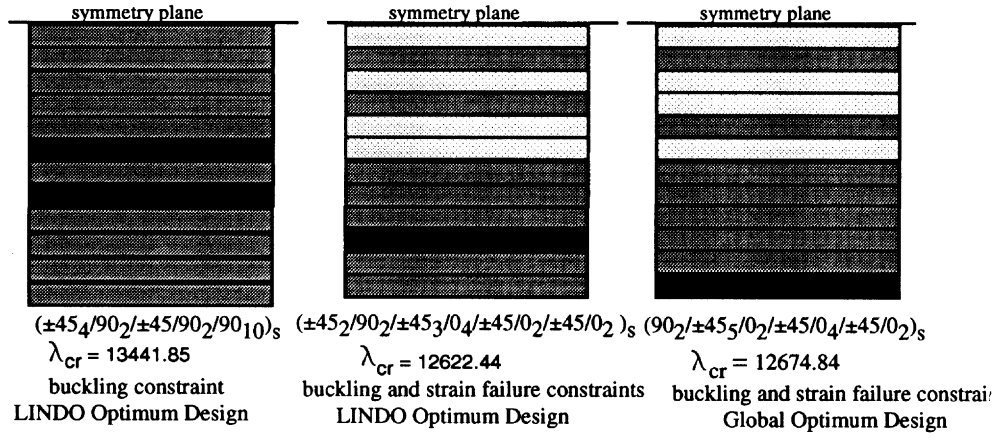
where ϵ_i^{ua} is the strain allowable. The strain constraint of Eq. (11.3.13) can be linearized by moving λ_c to the right hand side, and expanding $1/\lambda_c$ in linear Taylor's series to obtain

$$\lambda_o \epsilon_i + \frac{\lambda_c}{\lambda_o}\epsilon_i^{ua} \;\leq\; 2\epsilon_i^{ua}\,, \tag{11.3.14}$$

where λ_o is the buckling load factor for the nominal design.

The linear strain constraint of Eq. (11.3.14) can now be added to the problem formulation of Eqs. (11.3.3) for designing laminates that are buckling and strength failure resistant. Since the formulation involves a local approximation for the strength constraint, sequential linear programming needs to be used. In using sequential linear programmig, imposing move limits is generally recommended so that designs geneated based on approximate constraints remain in or near the feasible design space. In the case of *zero/one* ply-identity variables, imposing move limits on the design variables is not practical. Hence move limits were applied as bounds on the extensional stiffnesses

Ny = 0.25 lb/in.



N_y = 0.5 lb/in.

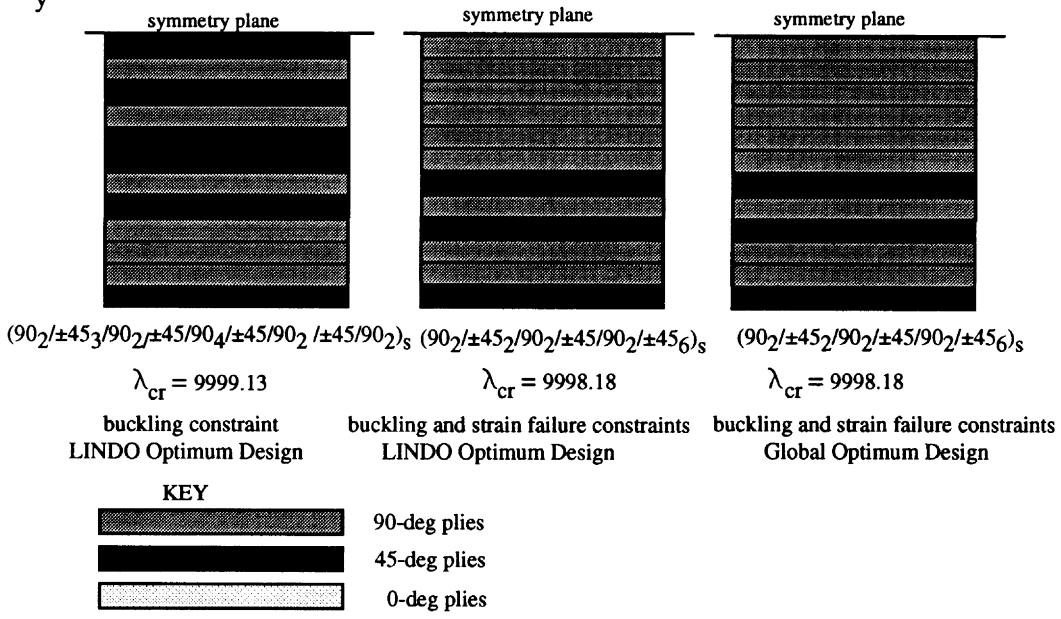


Figure 11.3.6 Maximum buckling load designs with strength constraints.

A_{ij} expressed in terms of the ply-identity variables. This requires addition of six more constraints to the problem

$$A_{ij}^L \leq A_{ij} \leq A_{ij}^U, \qquad i,j = 1,2\,. \tag{11.3.15}$$

Designs with strength constraints were obtained for laminates that are thicker than those considered in the previous cases so that the buckling loads are likely to

violate the strain failure constraints. Design results for 48-ply laminates under two different combination of biaxial loads ($N_y/N_x = 0.25$ and $N_y/N_x = 0.5$), for $N_x =$ 0.25 lb/in (175 N/m) are presented in Fig. 11.3.6, along with the results for designs with no strain constraint. Since the method used involves local approximations, the final design may be a locally optimal design. Designs with a higher confidence of being globally optimum can be generated by using one of the probabilistic search algorithms for nonlinear programming problems with discrete valued design variables (see chapter 4). The last design in each of the load cases presented in Fig. 11.3.6 is generated using the genetic algorithm discussed in section 4.4.2 and verified to be actually the global optimum design. Compared to the design without strength failure constraint, the failure load factor decreased by 6.05% for $N_y = 0.25$. Although the design for this load case was only a local maximum, the load factor differed from the global optimum design only by a fraction of a percent. For the load ratio of 0.5, the design without the strain constraint violated the shear strength by 7%. The design obtained from the sequential integer linear programming approach was also the global optimum.

11.3.4 *Probabilistic Search Methods*

Probabilistic search methods such as simulated annealing and genetic algorithms have a number of parameters that can be tuned to tailor the method to the problem at hand. For simulated anealing these parameters include the initial temperature and the rate of cooling. For genetic algorithms the tuning parameters are the probabilities of the various genetic operators, such as mutation, as well as population size and convergence criteria. The design of unstiffened laminates using Classical Lamination theory is a good problem for tuning such parameters because it is so computationally inexpensive to optimize.

For simulated annealing Lombardi [28] studied the effect of initial temperature and cooling rate on the performance of the algorithm for the buckling load maximization problem described in the previous section. The performance of the algorithm was judged by two criteria: computational cost and reliability in finding the global optimum. The problem tends to have a large number of solutions (stacking sequences) with very similar buckling loads. For this reason, a success was defined as a solution which is within 0.1% of the maximum buckling load. Results were obtained for 32-ply plates where plies were grouped in stacks of two 0-deg, 90-deg or $\pm$45-deg plies. For symmetric laminates this requires to define the angles of 8 stacks for a total of $3^8 = 6561$ possiblilities. The simulated annealing algorithm required about 1000 analyses for high reliability, which is a sizable fraction of the design space. However, when the number of plies was incleased from 32 to 64, the number of required analyses increased only to about 3000, while the number of possible designs increased to $3^{16} = 43$ million.

Le Riche and Haftka [29] solved the same buckling maximization problem for 48- and 64-ply laminates using genetic algorithms. Tuning the probabilites of the genetic operators as well as the population size could reduce substantially the number of required analyses. For 48-ply laminates, for example, the number of required analyses was found to be about of 200–300. One advantage of the genetic algorithm is that

it yields several near optimal designs, rather than one optimum. For example, for a plate with $a = 20$in, $b = 5$in, $N_x = 1$lb/in, and $N_y = 0.5$lb/in two of the best designs were: $[90_2, \pm 45_2, 90_2, \pm 45, 90_2, \pm 45_6]_s$, and $[\pm 45, 90_4, \pm 45, 90_2, \pm 45_5, 90_2, \pm 45]_s$. The first laminate has a buckling load of $\lambda_c = 9998$, while the second buckles at $\lambda_c = 9976$. For a designer, the differences between the laminates, such as the presence of ± 45-deg plies on the outside, or the reduced percentage of 90-deg plies in the second laminate may be more important than the 0.2% difference in buckling loads.

11.4 Design Applications

11.4.1 *Stiffened Plate Design*

Laminated plates stiffened by longitudinal and transverse members are one of the most common structural components. Use of stiffeners makes it possible to resist highly directional loads, and to introduce multiple load paths that may provide protection against damage and crack growth under both compressive and tensile loads. The biggest advantage of the stiffeners, though, is the increased bending stiffness of the panel with a minimum of additional material, which makes these structures highly desirable for out-of-plane loads and destabilizing compressive loads. In addition to placement of the stiffeners to resist directional loads, the use of composite materials makes it possible to further tailor the stiffness and strength characteristics of the individual elements (such as webs, flanges, and skin) of a stiffened plate to meet various structural requirements. This local tailoring is achieved through selection of ply orientations and thicknesses for the different sections of the plate. Also the use of composite materials makes it possible to adopt stiffener cross-sectional geometries which may be expensive to manufacture using metallic materials.

However, the complex behavior of stiffened composite plates makes it difficult to adopt the simplifying assumptions used for the analysis of flat laminates which often lead to closed-form solutions. Therefore, design optimization of such plates typically requires use of numerical algorithms. In this section we will discuss the design of stiffened composite plates under compressive and shear loadings, and subject to mainly buckling constraints.

In one of the early studies of optimum design of stiffened plates, Stroud and Agranoff [30] considered a longitudinally stiffened plate composed of an assembly of orthotropic plate elements. The plate configurations were limited to corrugated and hat-stiffened plates, but the same procedure used in Ref. 30 can be extended to other geometries such as the ones shown in Figure 11.4.1. The simplified analysis was based on buckling of orthotropic plates with simply supported boundary conditions. Both global and local modes of buckling were considered. The global buckling analysis modeled the stiffened plate as an orthotropic plate with smeared stiffeners, assumed to buckle as a wide column. For local buckling, each element of the plate was considered separately as a narrow strip of orthotropic plate with simply supported boundary

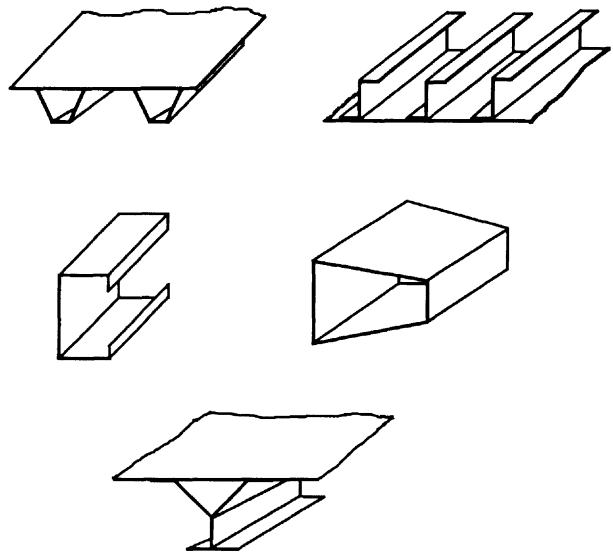


Figure 11.4.1 Examples of typical stiffened panel concepts.

conditions along the lines of attachment to adjacent elements. That is, the rotational restraint between panel elements such as stiffener and skin was ignored, and the continuity of the buckling mode shapes between different elements was not accounted for. Equations for the buckling loads resulting from these assumptions are presented in Table 11.4.1 for plates loaded by compressive and shear loads.

The local buckling equations in the table are applied to each of the plate elements of width b and length L. The length L of each element is assumed to be much larger than the width of the elements for both longitudinal compression and shear loadings. The D_{ij}'s are the bending stiffness coefficients (Eq. 11.1.18) of the respective plate elements. For global buckling under longitudinal compression, the panel is treated as a wide column with the loaded edges simply supported and the unloaded edges free. The longitudinal stiffness of the column is equal to the smeared longitudinal stiffness of the panel, EI. For the shear loading case, the stiffened panel is modeled as a uniform thickness orthotropic laminate (with smeared orthotropic properties, D_1, D_2, and D_3) infinitely long in the transverse direction and simply supported along the loaded edges. The smeared stiffness terms (EI, D_1, D_2, and D_3) in the global buckling relations strongly depend on the cross-sectional configuration of the stiffeners. The calculation of these smeared stiffnesses for complicated stiffened panel geometries is quite involved and requires various kinematic assumptions depending on the applied loads. The derivation of some of the smeared stiffness terms is demonstrated in Ref. 30 for corrugated and hat-stiffened panels.

The design problem of Ref. 30 was formulated as a mathematical programming problem with panel mass per unit width being the objective function. Design variables

Table 11.4.1 : Overall and Local Buckling Equations from Reference 30

Loading	Equation	Reference
Global Buckling		
Longitudinal Compression	$N_{x,\mathrm{cr}} = \dfrac{\pi^2 EI}{pL^2} \dfrac{1}{1 + \dfrac{\pi^2 EI}{2L^2 b_2 A_{66_2}}}$	Eq. (92), [31] Eq. (3), [32]
Shear	$\zeta = \dfrac{\sqrt{D_1 D_2}}{D_3}$	Eqs. (2.2.2-21),
For $\zeta > 1$,	$N_{xy,\mathrm{cr}} = (\dfrac{2}{L})^2 (D_1^3 D_2)^{\frac{1}{4}} (8.125 + \dfrac{5.05}{\zeta})$	(2.2.2-22), [33]
For $\zeta < 1$,	$N_{xy,\mathrm{cr}} = (\dfrac{2}{L})^2 \sqrt{D_1 D_3}(11.7 + 0.532\zeta + 0.938\zeta^2)$	pp. 468-471, [34]
Combined	$\dfrac{N_x}{N_{x,\mathrm{cr}}} + \left(\dfrac{N_{xy}}{N_{xy,\mathrm{cr}}}\right)^2 = 1$	Eq. (105.8), [34]
Local Buckling		
Longitudinal Compression	$N_{x,\mathrm{cr}} = \dfrac{2\pi^2}{b^2}\left[(D_{11}D_{22})^{\frac{1}{2}} + D_{12} + 2D_{66}\right]$	Eq. (92), [31] Eq. (3), [32]
Shear	$\zeta = \dfrac{\sqrt{D_{11}D_{22}}}{D_{12} + 2D_{66}}$	Eqs. (2.2.2-21),
For $\zeta > 1$,	$N_{xy,\mathrm{cr}} = (\dfrac{2}{b})^2 (D_{11} D_{22}^3)^{\frac{1}{4}} (8.125 + \dfrac{5.05}{\zeta})$	(2.2.2-22), [33] ;
For $\zeta < 1$,	$N_{xy,\mathrm{cr}} = (\dfrac{2}{L})^2 \sqrt{D_{22}(D_{12} + 2D_{66})}$ $(11.7 + 0.532\zeta + 0.938\zeta^2)$	pp. 468-471, [34]
Combined	$\dfrac{N_x}{N_{x,\mathrm{cr}}} + \left(\dfrac{N_{xy}}{N_{xy,\mathrm{cr}}}\right)^2 = 1$	Eq. (105.8), [34]

were the element widths and thicknesses of the layers that make up the elements. The design constraints were buckling load, strength and stiffness requirements, and lower and upper bounds on some of the panel dimensions. A general purpose optimization code AESOP [35], which is based on exterior penalty function formulation, was used for the design optimization.

A more rigorous design procedure [36] based on a stiffened panel buckling and vibration analysis code VIPASA [37, 38] and a mathematical programming code based on the method of feasible directions algorithm (see Section 5.6) CONMIN [39] was introduced to improve some of the assumptions made in Ref. 30. The analysis code VIPASA is capable of computing buckling loads of structures comprised of flat rectangular plate elements connected together along their longitudinal edges. As opposed to the procedure used in Ref. 30, the analysis accounts for the physical connection between the adjacent elements by maintaining the continuity of the buckle

patterns across the intersection of neighboring plate elements. Buckling solutions are based on exact thin-plate equations with D_{16} and D_{26} anisotropic stiffness terms so that bending-twisting coupling is allowed. Individual plate elements may be isotropic, orthotropic, or anisotropic. However, the laminates that make up the elements are limited to balanced symmetric layups such that bending-extension and extension-shearing couplings are eliminated. Another limitation of the analysis is the buckling boundary conditions. Although the unconnected longitudinal edges may take various boundary conditions, the boundary conditions along the loaded edges are limited to simply supported conditions. Any combination of longitudinal, transverse, and shearing loads that are constant along the length of the panel may be applied, see Fig. 11.4.2. However, as will be discussed later, in the case of applied shear loads the limitation of the simply supported boundary condition at the loaded edge may result in inaccuracies in the buckling load calculations.

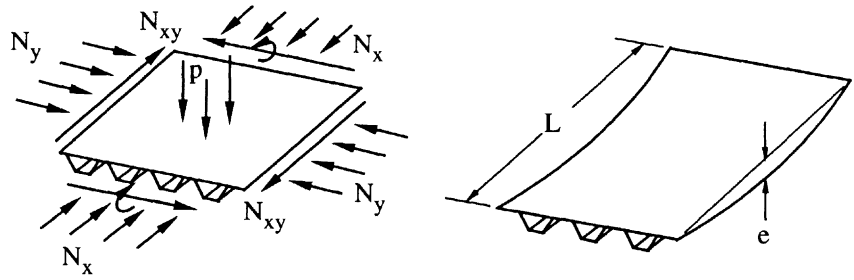


Figure 11.4.2 Loading conditions and initial imperfections.

The VIPASA analysis program was eventually used by Stroud and Anderson as the basis of a design code PASCO [40, 41] which is commonly used for preliminary design of uniaxially-stiffened panel structures. PASCO uses the nonlinear mathematical programming code CONMIN [39] for optimization. The design problem is formulated so as to minimize the panel mass for a given set of loadings. Constraints include upper and lower bounds on design variables, lower bounds on material strength and buckling loads, lower and upper bounds on overall bending, extensional, and shear stiffnesses, and lower bounds on vibration frequencies. In addition to the design condition described for VIPASA analysis (N_x, N_y, N_{xy}), PASCO includes applied bending moment (M_x), lateral pressure (p), overall bow-type initial imperfections, and temperature loadings. The effects of the bending strains, resulting from the applied bending moment, pressure, initial imperfection, or the temperature, are included in the strain failure analysis by superimposing them on the uniform strains resulting from the in-plane loads. The bending strains resulting from the applied pressure and bow-type imperfections are calculated based on a beam-column approach [42] by calculating the corresponding bending moment at the panel midlength. This maximum bending moment is conservatively assumed to act over the entire panel length.

This approach is in line with the VIPASA requirement that the prebuckling stress distribution be constant along the panel length. For more detailed discussion of the bending moments see Ref. 40. Use of multiple sets of design conditions is also allowed in PASCO. The set of design variables consists of the widths, b, the ply thicknesses, t, and orientations, θ, of any of the plate elements that make up the panel. Reducing the number of design variables by linking of some of the element dimensions or ply orientations through linear relations is also possible. PASCO is also capable of implementing approximations for the buckling and vibration constraints through first-order Taylor series expansion of those constraints, and set move limits for the design variables. This aspect of the code makes it computationally efficient and very attractive for preliminary design purposes, and lets the designer compare various design concepts in a cost-effective manner.

Example 11.4.1

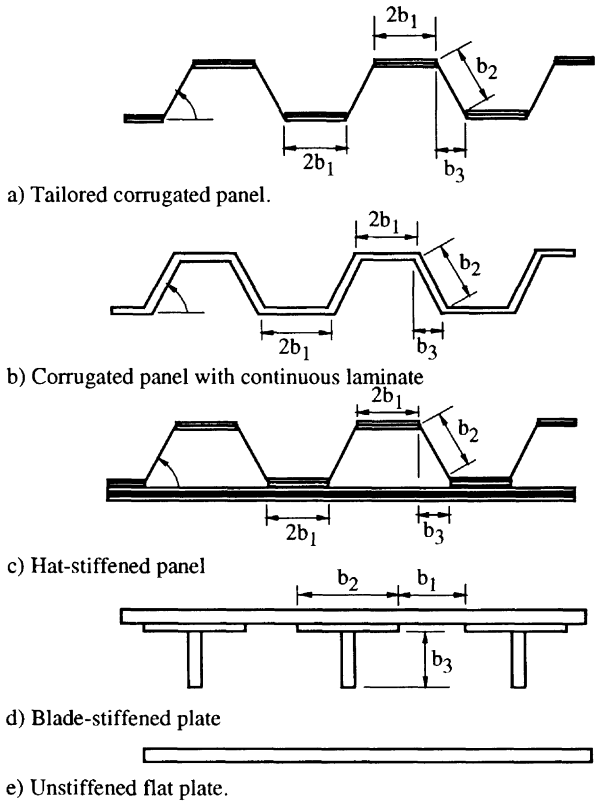


Figure 11.4.3 Design configurations.

This example by Swanson and Gürdal [43] is a comparison of the structural efficiencies of optimally designed composite wing rib panel configurations typical of a center-wing-box fuel-cell closeout rib of large transport-type aircraft. Rib dimensions of 28 inches high by 80 inches wide are used. The panel configurations are chosen to be practical and applicable to cost-effective manufacturing techniques. These configurations are shown in Fig. 11.4.3, and include a tailored corrugated panel, a corrugated panel with a continuous laminate throughout its length and width, and a hat-stiffened panel. A corrugated panel is relatively easy to manufacture since it has continuous plies which run throughout the configuration that form integral stiffeners without requiring fasteners. It is also suitable for the thermoforming process which is a potentially economical manufacturing technique for thermoplastic materials. Also included are a blade-stiffened panel, which is the most commonly used concept for wing rib applications, and a flat unstiffened plate which is used as a baseline configuration for comparison.

The constraints considered in this example include those associated with material strength, buckling, and geometric limits. The material failure criterion chosen is the maximum strain failure criterion. The buckling criterion implemented is based on a common design practice used for wing structures that does not allow the components to buckle at design limit loads. Thus, the design of the wing rib does not consider any postbuckling-load-carrying capability of the panel.

The design variables are the thicknesses of plies with different ply orientations in the different sections of the panels. Conventional ply angles of ±45-deg, 0-deg, and 90-deg orientations are chosen. Also, detailed cross-sectional dimensions are used as sizing variables to determine the best cross-sectional geometry. Hercules AS4/3502 preimpregnated graphite-epoxy tape is chosen as a typical graphite-epoxy material.

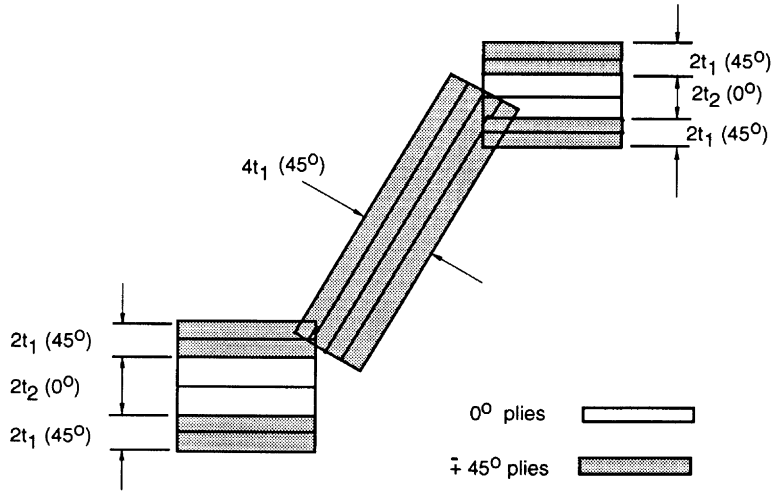


Figure 11.4.4 Tailored corrugated panel model.

The geometry of the repeating elements is typically defined by the plate element width design variables b_1 through b_4 as shown in Fig. 11.4.3. For the corrugated panels, for example, both the upper and lower corrugation caps are assumed to be of equal width due to symmetry. The plate element widths, b_2 and b_3, define the corrugated panel web angle. The panel webs are made of only ±45-deg plies, Fig. 11.4.4, that run continuously across the width of the cross section. Such continuous plies help reduce manufacturing costs and eliminate stress concentrations that could occur at the ±45-deg ply termination points. In the plate elements which make up the caps, 0-deg plies are included between the layers of ±45-deg fibers. Thus, the entire laminate is defined by two thickness design variables, t_1 and t_2, relating to the 45-deg and 0-deg plies, respectively. Cross-sectional details of the other configurations can be obtained from Ref. 43.

The loads considered in Ref. 43 are combined in-plane axial compression (N_x), shear (N_{xy}), and pressure (p) loads with magnitudes typical of an inboard wing rib fuel closeout cell for a large transport aircraft. In the present example a load index of N_x/L, where L is the panel length, is used with values ranging from 0.3 to 1000 lb/in^2. This range includes loadings above and below typical rib loads so that design trends for panels for other subcomponents, such as a wing skin, are covered.

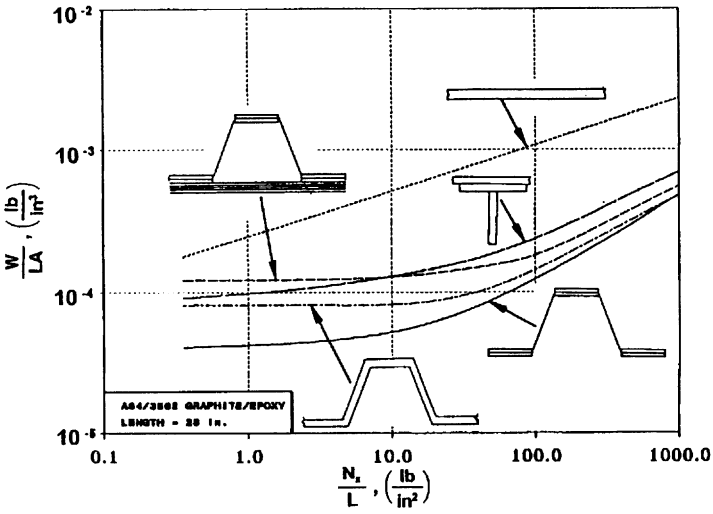


Figure 11.4.5 Structural efficiency of axial compression loaded panels.

The effect of axial compression load intensity on the structural efficiency and geometry of all the panel configurations considered in the present study is shown in Figs. 11.4.5. The tailored corrugated panel concept with different laminates in the corrugation crowns and webs is the most structurally efficient configuration. The corrugated panel concept with a continuous laminate is the next most structurally

efficient concept, followed by the blade-stiffened panel concept, the hat-stiffened panel concept, and the unstiffened flat panels (see Fig. 11.4.5). The weight differences in this load range are due largely to the modeling of the laminates that define the panel geometry. Each configuration is modeled such that a minimum number of plies necessary to define the geometry is used, and that number differs for each model. For low axial load intensity, all configurations, excluding the unstiffened plate, are constrained by the same minimum gage ply thickness of 0.005 inches on all the plies. Therefore, the weight of a panel is almost directly proportional to the number of layers in the cross section and is independent of the intensity of the load. • • •

One drawback of PASCO is possible inaccuracy in modeling the boundary conditions under shear loads. Boundary conditions on the panel ends perpendicular to the stiffeners are assumed to be simply supported and cannot be changed. Without the shearing loads, the buckling pattern consists of a series of straight nodal lines that coincide with the loaded edges of the panel. When shear is applied to the panels, the buckling pattern consists of a series of skewed nodal lines and the buckling load calculated for this load case may deviate from to the buckling load of a simply supported plate. In particular, if a single buckling half-wave of length λ forms along the panel length, L, the PASCO analysis can severely underestimate the buckling load.

An optional smeared stiffness solution [38] is included in PASCO for the $\lambda = L$ case to provide a more accurate solution when a shear load is present. The smeared stiffness approach was shown [44] to be an improved solution but not always a conservative one. Additionally, in order to achieve an optimally designed stiffened panel configuration, the full cross-sectional detail must be retained to account for local stiffener buckling, while at the same time, maintaining the simple support boundary conditions at the loaded edges. The smeared stiffener solution in PASCO does not account for such detail. An improved analysis exists in the VICON, (**VI**PASA with **CON**straints) program [45, 46] which modifies the VIPASA buckling analysis to include supports at arbitrary locations along the panel length through the use of Lagrange multipliers. By specifying the supports at intervals corresponding to the ends of the desired panel length, the simple support boundary conditions can be enforced at the panel ends when shear is applied. The VICON analysis has recently been implemented in a design code VICONOPT by Butler and Williams [47].

The design requirement that does not allow buckling of the panels at the limit load is appropriate for wing and empennage cover panels because of nonstructural considerations such as maintaining a good aerodynamic surface. However, fuselage panels of metallic aircraft structures are commonly designed to buckle below their ultimate loads. The lack of sufficient information on the postbuckling response of composite panels hindered the application of such a design philosophy in the past. Realization of the possible weight saving kindled interest in designing postbuckled panels in recent years (see Dickson et al. [48, 49] and Shin et al. [50]. A nonlinear theory for the prediction of behavior of locally imperfect stiffened panels has been incorporated by Bushnell into the design optimization program PANDA2 [51]. However, because of the complexity and serious computational cost involved in postbuckling analysis of stiffened panel structures, optimal design of such panels is still far from being a routine practice.

11.4.2 *Aeroelastic Tailoring*

Another major area of design optimization application is the aeroelastic tailoring of aircraft wing structures which involve aeroelastic constraints. Aeroelastic tailoring involves the use of structural deformations to improve the structural and aerodynamic characteristics of a lifting surface. A suggested standard definition [52] is;

Aeroelastic tailoring is the embodiment of directional stiffness into an aircraft structural design to control aeroelastic deformation, static or dynamic, in such a fashion as to affect the aerodynamic and structural performance of that aircraft in a beneficial way.

The beneficial behavior characteristics are those associated with the aeroelastic twist, aeroelastic camber, improved flutter and divergence speeds, reduced aeroelastic roll control losses, and increased strength [53].

The subject of aeroelastic tailoring has gained popularity during the past decade because of advancements in the structural optimization field and increased use of composite materials in aircraft structures. Composite wing designs are often more flexible than metal ones, which makes them more susceptible to aeroelastic effects. However, composite materials often provide the designer with an opportunity to improve aerodynamic performance by tailoring the material response, through the use of ply thickness and orientation design variables, to generate favorable aeroelastic effects. While there is an increased flexibility in tailoring the design, the increased number of design variables and complex response characteristics of composite materials make the difficult wing design problem even more difficult [54, 55]. This is where the use of advanced optimization techniques come into picture. Although it is still considered costly, application of rigorous optimization algorithms to the detailed structural model of a lifting surface may make it possible to achieve the desired performance improvements. Many of the early studies, however, relied on simplification of the structural model to make the design affordable. These simplifications included, in some cases, beam models for the structural representation. A survey of the applications of structural optimization techniques to problems of design under aeroelastic constraints is presented by Haftka [56].

One of the early efforts in introducing structural optimization into aeroelastic tailoring is the TSO program developed by McCullers and Lynch [57]. The program originated under the name WASP (Wing Aeroelastic Synthesis Procedure) [58], and uses a mathematical programming procedure based on a penalty approach (see Section 5.7) for converting the constrained problem into a series of unconstrained problems. The unconstrained minimizations are performed via the Davidon-Fletcher-Powell algorithm (see Section 4.2). Modeling of the wing structure is based on plate analysis with a Ritz solution technique. The objective function may be any combination of weight, lift curve slope, control surface effectiveness, flutter speed, fundamental natural frequency or deflections. The design variables are coefficients of polynomials which control both the orientations of the various plies and their thicknesses. Use of a polynomial description of the design parameters along with the Ritz procedure makes the application the mathematical programming method manageable

for optimization purpose. The TSO program was used for several design studies for aeroelastic tailoring applications to existing aircraft [54, 58, 59]

Another popular program for the design of lifting surfaces subject to strength and aeroelastic constraints is the finite element based program FASTOP developed by Grumman [60]. The program employs optimality criteria methods (see Chapter 9), and is also capable of handling flutter constraints. Optimality criteria methods are very efficient for designs subject to a single constraint. Thus, despite the costly finite element analysis involved, the cost of optimization was kept manageable through the use of sequential treatment of constraints. First the stress constraints are treated by the non-optimal Fully Stressed Design (FSD, see Section 9.1), followed by a 'uniform-cost-effectiveness' optimality criterion (Section 9.3) for each of the aeroelastic constraints. The process is repeated with the strength and aeroelastic constraints until convergence is achieved. Design variables are limited to thickness or cross-sectional areas, and ply orientations are not allowed to change during the design.

A more recent finite element based design program is ASTROS (Automated Structural Optimization Systems) [61] developed by Northrop under an Air Force contract. ASTROS is designed as an automated procedure to address interdisciplinary requirements during preliminary design of aerospace structures. The structural analysis module of ASTROS is derived from the public domain version of the NASTRAN finite element code and forms the core of the procedure. The structural analysis module is used to obtain structural response to applied mechanical, gravitational, aerodynamic, induced thermal, and time dependent loads. Design constraints include limits on stresses, strains, displacements, modal frequencies, flutter response, aeroelastic lift effectiveness, and aileron effectiveness. Design variables that can be used in the process are element areas and thicknesses, structural inertias and concentrated masses. Membrane and bending elements used in the structural analysis provide full-composite modeling capability. Individual ply thicknesses of the material can be used as design variables, but the ply orientation design variables are not allowed. In order to reduce the number of design variables and to assure physically meaningful dimensions design variable linking is used. The design variable linking is implemented together with a procedure that divides the design variables into two groups that are identified as global and local variables. A global design variable can be specified as a weighted sum of a number of local design variables. Similar to TSO, shape function type of linking can be used to define shapes such as a smooth thickness variation along the span direction. The design optimization module used in ASTROS is the ADS (Automated design Synthesis) [62] program. All sensitivities of the objective function and of the constraints are calculated based on analytical derivatives. Both direct and adjoint-variable methods (see Chapter 7) are available.

11.5 Design Uncertainties

Although composite materials provide a vast, and probably so far underutilized, freedom in tailoring structural response to suit the needs of the designer, they also display certain problems uncharacteristic of conventional materials. Optimally designed

structures are known to be sensitive to changes in load conditions and imperfections. Because of increased number of variables which enable designers to tailor the design closer to the desired specifications, this sensitivity may be heightened for composite structures. The simplest example of sensitivity to changes in the load condition is the case of a laminate designed to carry uniaxial loads [63]. For this application, it can easily be demonstrated that the best design is the one that has all the layers oriented along the load direction. It is also well known that this design is extremely poor for carrying loads transverse to the fiber direction. Therefore, any change in the direction of the applied design load is likely to result in a failure, whereas a similar design made of a conventional isotropic material would be capable of carrying a transverse load of magnitude equal to the original design load.

Another complication in designing optimal composite structures is sometimes the difficulty in identifying and imposing proper strength constraints. Not only the load and stress distributions are functions of the ply thickness and fiber orientation variables, but the strength properties are also dependent on these variables. Failure of composite laminates is largely due to highly localized stresses. The number of possible local failure modes is large, and these failure modes are generally micromechanically governed and complex. Fiber breaking, matrix cracking, fiber-matrix debonding, and separation of individual layers can result in surface and through-the-thickness cracks, splits, and delaminations. Under compressive loads, even the instability of fibers on a microscopic scale (often referred as fiber microbuckling) was proposed as a failure mechanism, although based on more recent studies compression failures for high-performance composites are believed to be strength-related failures. Furthermore, failure modes can interact with one another making the strength prediction even more difficult.

Some of the basic assumptions used for simplification of the laminate stress analysis that reduce the three-dimensional nature of the laminated composites to two dimensions may also cause loss of information important for failure predictions. It is well known that laminated composite plates can locally display a three-dimensional stress state. The most common examples of these three-dimensional effects are free-edge stresses, and interlaminar stresses at the stiffener-skin interface of stiffened panels. It is important that designers be aware of such local effects during the formulation of the optimization problem and include appropriate constraints to account for them.

It is only fair to claim that some of the design-related issues of composites failures are not well understood. Sometimes strength quantities that are needed for implementation of a certain stress constraint may not be available. For example, based on their experience with metallic materials, designers often look for a compressive material strength limit that they can include in an optimization problem. It can be argued that the compressive failure strength is a highly problem-dependent quantity, rather than a material strength parameter. In some applications, the lack of understanding and availability of predictive models for certain design considerations may hamper the design effort. For example, unlike metallic materials, composites have been found to be sensitive to low-velocity impact loadings. Currently, there is no predictive model that can realistically be used for designing laminates under impact damage conditions. Some of these topics are still under development and constitute a major effort in the area of mechanics of composite materials.

Under these difficulties, designers sometimes resort to practical guidelines. Rather than using ply orientation angles as design variables, designers often fix them to prescribed practical angles such as 0-deg, ±45-deg, and 90-deg. Even if the applied loading is highly directional, such as panels under uniaxial loadings, presence of plies other than the ones aligned along the load direction provides increased safety for off-design load conditions such as unexpected transverse loads. In order to assure that the thickness design variables associated with those plies that are placed based on intuitive guidelines do not disappear, either lower bound on those thicknesses are used or additional loads are specified. For example, application of a certain percentage of the axial load as shear load leads to non-zero thickness for ±45-deg layers even if the lower bound on those layers is zero.

The selection of a stacking sequence for a laminate is also guided by intuitive considerations. For example, use of ±45-deg plies as the outside layers of a laminate is preferred because of damage tolerance considerations. Another practical guideline is not to allow more than 4 identical contiguous plies. This guideline helps to reduce the interlaminar stresses between plies with different orientations. In order to satisfy such ply stacking sequence rules, an iterative procedure may be used as outlined in Ref. [64]. If the branch-and-bound algorithm with ply identity variables is used, this requirement can easily be implemented through the use of Eq. (11.3.5) as described earlier.

11.6 Exercises

1. For a unidirectional laminate under uniform applied stresses, σ_x, σ_y, and τ_{xy}, show that the stationary values of the Tsai-Hill function

$$f = \left(\frac{\sigma_1}{X}\right)^2 - \left(\frac{\sigma_1\sigma_2}{X^2}\right) + \left(\frac{\sigma_2}{Y}\right)^2 + \left(\frac{\tau_{12}}{S}\right)^2 , \tag{11.6.1}$$

are achieved for

$$a\cos 2\theta + \sin 2\theta = 0 ,$$

and

$$a\sin 2\theta - \cos 2\theta = b ,$$

where

$$a = \frac{2\tau_{xy}}{\sigma_y} - 1, \quad \text{and} \quad b = \frac{\sigma_y + \sigma_x}{\sigma_y - \sigma_x}\frac{1-\alpha^2}{\beta^2 - \alpha^2 - 2} ,$$

and

$$\alpha = X/Y , \quad \text{and} \quad \beta = X/S ,$$

where X, Y are the normal strengths parallel and transverse directions to the fibers and S is the shear strength.

2. Using the graphical procedure described in section 2, determine the orientations and thickness ratio of a balanced angle-ply symmetric, $[(\pm\theta_1)_{v_1}/(\pm\theta_2)_{v_2}]$, T300-5208

graphite-epoxy laminate with maximum effective shear stiffness G_{xy}. The laminate must also meet the following stiffness requirements

$$E_x \geq 17.5\ 10^6\ psi, \quad E_y \geq 5.8\ 10^6\ psi, \quad \text{and} \quad 0.1 \geq \nu_{xy} \geq 0.3\ .$$

Engineering properties of T300-5208 graphite-epoxy material along its principal material directions are

$$E_1 = 26.25\ 10^6\ psi,\ E_2 = 1.49\ 10^6\ psi,\ G_{12} = 1.04\ 10^6\ psi, \text{ and } \nu_{12} = 0.28\ .$$

3. Show that a quasi-isotropic laminate $[0_i^o, 90_i^o, -45_i^o, +45_i^o]$ can be replaced by $[90_j^o, 0_k^o, -45_i^o, +45_i^o]$ with an identical **D** matrix by suitably selecting j and k so that $j + k = 2i$ (note: j and k may be non-integers).

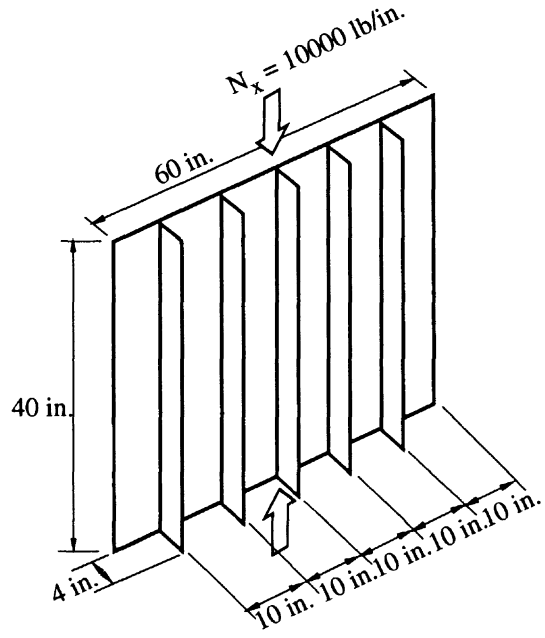


Figure 11.6.1 Blade stiffened panel under uniaxial compression.

4. For a laminate that is made up of integer number of plies with 0-, ±30-, ±60-, and 90-deg orientations, the design space is shown in Fig. 11.3.1-b.

a) Complete the figure by putting the stacking sequences of laminates next to the appropriate discrete design points on the figure.

b) If the laminate is required to have a Poisson's ratio ν_{xy} greater than 0.3, determine the stacking sequence that maximizes the transverse modulus E_y.

5. The skin laminate of a simply supported blade stiffened panel shown in Figure 11.6.1 is a $[\pm 45_n]_s$ construction, and the stiffeners are made of unidirectional laminae. Determine the longitudinal smeared stiffness EI which can be used for the global buckling load calculation presented in Table 11.3.1. Assuming the thicknesses of individual plies to be continuously variable, determine the minimum weight design for an axial compression of $N_x = 10000\ lb/in$. Consider only buckling constraints.

11.7 References

[1] Jones, R. M., Mechanics of Composite Materials, McGraw-Hill Book Co., New York, pp. 45–57, 1975.

[2] Tsai, S. W., and Pagano N. J., "Invariant Properties of Composite Materials," in Composite Materials Workshop, (Eds. Tsai, S.W., Halpin, J.C., Pagano, N.J.) Technomic Publishing Co., Westport, pp. 233–253, 1968.

[3] Caprino, G., and Crivelli–Visconti, I., "A Note on Specially Orthotropic Laminates," J. Comp. Matls., 16, pp. 395–399, 1982.

[4] Gunnink, J. W., "Comment on A Note on Specially Orthotropic Laminates," J. Comp. Matls., 17, pp. 508–510, 1983.

[5] Kandil, N., and Verchery, G., "New Methods of Design for Stacking Sequences of Laminates ," Proceedings of the International Conference on "Computer Aided Design in Composite Material Technology," Eds. Brebbia, C. A., de Wilde, W. P., and Blain, W. R., pp. 243–257, 1988.

[6] Schmit, L. A., and Farshi, B., "Optimum Laminate Design for Strength and Stiffness," Int. J. Num. Meth. Engng., 7, pp. 519–536, 1973.

[7] Park, W. J., "An Optimal Design of Simple Symmetric Laminates Under the First Ply Failure Criterion," J. Comp. Matls., 16, pp. 341–355, 1982.

[8] Massard, T. N., "Computer Sizing of Composite Laminates for Strength," J. Reinf. Plastics and Composites, 3, pp. 300–345, 1984.

[9] Tsai, S. W., and Hahn, H. T., Introduction to Composite Materials, Technomic Publishing Co., Inc., Lancaster, Pa., pp. 315–325, 1980.

[10] Tsai, S. W., "Strength Theories of Filamentary Structures," in R. T. Schwartz and H. S. Schwartz (eds.), Fundamental Aspects of Fiber Reinforced Plastic, Wiley Interscience, New York, pp. 3–11, 1968.

[11] Brandmaier, H. E., "Optimum Filament Orientation Criteria," J. Composite Materials, 4, pp. 422–425, 1970.

[12] Miki, M., "Material Design of Composite Laminates with Required In–Plane Elastic Properties," Progress in Science and Engineering of Composites, Eds., T.

Hayashi, K. Kawata, and S. Umekawa, ICCM–IV, Tokyo, Vol. 2, pp. 1725–1731, 1982.

[13] Miki, M., "A Graphical Method for Designing Fibrous Laminated Composites with Required In-plane Stiffness," *Trans.* JSCM, 9, 2, pp. 51–55, 1983.

[14] Schmit, L. A., and Farshi, B., "Optimum Design of Laminated Fibre Composite Plates," Int. J. Num. Meth. Engng., 11, pp. 623–640, 1977.

[15] Miki, M., "Optimum Design of Laminated Composite Plates Subject to Axial Compression," Composites' 86: Recent Advances in Japan and the United States, Eds., Kawata, K., Umekawa, S., and Kobayashi, A., Proc. Japan–U.S. CCM–III, Tokyo, pp. 673–680, 1986.

[16] Bert, C. W., "Optimal Design of a Composite–Material Plate to Maximize its Fundamental Frequency," J. Sound and Vibration, 50 (2), pp. 229–237, 1977.

[17] Rao, S. S., and Singh K., "Optimum Design of Laminates with Natural Frequency Constraints," J. Sound and Vibration, 67 (1) pp. 101–112, 1979.

[18] Mesquita, L., and Kamat, M. P., "Optimization of Stiffened Laminated Composite Plates with Frequency Constraints," Eng. Opt., 11, pp. 77–88, 1987.

[19] Cheng Kengtung, "Sensitivity Analysis and a Mixed Approach to the Optimization of Symmetric Layered Composite Plates," Eng. Opt., 9, pp. 233–248, 1986.

[20] Pedersen, P., "On Sensitivity Analysis and Optimal Design of Specially Orthotropic Laminates," Eng. Opt., 11, pp. 305–316, 1987.

[21] Muc, A., "Optimal Fiber Orientation for Simply–Supported, Angle–Ply Plates Under Biaxial Compression," Comp. Struc., 9, pp. 161–172, 1988.

[22] Shin, Y. S., Haftka, R. T., Watson, L. T., and Plaut, R. H., "Design of Laminated Plates for Maximum Buckling Load", J. Composite Materials, 23, pp. 348–369, 1989

[23] Miki, M., and Sugiyama, Y., "Optimum Design of Laminated Composite Plates Using Lamination Parameters," Proceedings of the AIAA/ASME/ASCE/AHS/ ASC 32th Structures, Structural Dynamics, and Materials Conference, Baltimore, MA., Part I, pp. 275–283, April, 1991.

[24] Gürdal, Z. and Haftka, R. T., "Optimization of Composite Laminates," presented at the NATO Advanced Study Intitute on Optimization of Large Structural Systems, Berchtesgaden, Germany, Sept. 23 - Oct. 4, 1991.

[25] Haftka, R.T., and Walsh, J.L., "Stacking-Sequence Optimization for Buckling of Laminated Plates by Integer Programming, AIAA Journal (in Press).

[26] Schrage, L., Linear, Integer and Quadratic Programming with LINDO, 4th Edition, The Scientific Press, Redwood City CA., 1989.

[27] Nagendra, S., Haftka, R. T., and Gürdal, Z., "Optimization of Laminate Stacking sequence with Stability and Strain Constraints," submitted for presentation at

the AIAA/ASME/ASCE/AHS/ASC 33th Structures, Structural Dynamics, and Materials Conference, Dallas, TX., April, 1992.

[28] Lombardi, M., "Ottimizzazione di Lastre in Materiale Composito con l'uso di un Metodo di Annealing Simulato," Tesi di Laurea, Department of Structural Mechanics, University of Pavia, 1990.

[29] Le Riche, R., and Haftka, R.T., "Optimization of Laminate Stacking-Sequence for Buckling Load Maximization by Genetic Algorithm," submitted for presentation at the AIAA/ASME/ASCE/AHS/ASC 33th Structures, Structural Dynamics, and Materials Conference, Dallas, TX., April, 1992.

[30] Stroud, W. J., and Agranoff, N., "Minimum-mass Design of Filamentary Composite Panels Under Combined Loadings: Design Procedure Based on Simplified Buckling Equations," NASA TN D-8257, 1976.

[31] Timoshenko, S., Theory of Elastic Stability, McGraw-Hill, New York, 1936.

[32] Stein, M., and Mayers, J., "Compressive Buckling of Simply Supported Curved Plates and Cylinders of Sandwich Construction," NACA TN 2601, 1952.

[33] Advanced Composites Design Guide. Vols. I-V, Third Edition, U.S. Air Force, Jan. 1973.

[34] Lekhnitskii, S. G., Anisotropic Plates. Translated by Tsai, S. W., and Cheron, T., Gordon and Breach Sci. Publ., Inc., New York, 1968.

[35] Hague, D. S., and Glatt, C. R., "A Guide to the Automated Engineering and Scientific Optimization Program, AESOP, " NASA CR-73201, April 1968.

[36] Stroud, W. J., Agranoff, N., and Anderson, M. S., "Minimum-Mass Design of Filamentary Composite Panels Under Combined Loads: Design Procedure Based on a Rigorous Buckling Analysis, " NASA TN D-8417, July 1977.

[37] Wittrick, W. H., and Williams, F. W., "Buckling and Vibration of Anisotropic or Isotropic Plate Assemblies Under Combined Loadings, " Int. J. Mech. Sci., 16, 4, pp. 209-239, April 1974.

[38] Plank, R. J., and Williams, F. W., "Critical Buckling of Some Stiffened Panels in Compression, Shear and Bending, " Aeronautical Q., *XXV*, Part 3, pp. 165-179, August 1974.

[39] Vanderplaats, G. N., "CONMIN - A Fortran Program for Constrained Function Minimization, User's Manual," NASA TM X-62, 282, 1973.

[40] Stroud, W. J., and Anderson, M. S., "PASCO: Structural Panel Analysis and Sizing Code, Capability and Analytical Foundations, " NASA TM 80181, November 1981.

[41] Anderson, M. S., Stroud, W. J., Durling, B. J., and Hennessy, K. W., "PASCO: Structural Panel Analysis and Sizing Code, User's Manual, " NASA TM 80182, November 1981.

[42] Giles, G. L., and Anderson, M. S., "Effects of Eccentricities and Lateral Pressure on the Design of Stiffened Compression Panels," NASA TN D–6784, June 1972.

[43] Swanson, G. D., and Gürdal, Z., "Structural Efficiency Study of Graphite–Epoxy Aircraft Rib Structures," J. Aircraft, 27 (12), pp. 1011-1020, 1990.

[44] Stroud, W.J., Greene, W.H., and Anderson, M.S., "Buckling Loads of Stiffened Panels Subjected to Combined Longitudinal Compression and Shear: Results Obtained With PASCO, EAL, and STAGS Computer Programs," NASA TP 2215, January 1984.

[45] Williams, F.W., and Kennedy, D., "User's Guide to VICON, VIPASA with Constraints," Department of Civil Engineering and Building Technology, University of Wales Institute of Science and Technology, August, 1984.

[46] Williams, F.W., and Anderson, M.S., "Incorporation of Lagrangian Multipliers into an Algorithm for Finding Exact Natural Frequencies or Critical Buckling Loads," Int. J. Mech. Sci., 25, 8, pp. 579–584, 1983.

[47] Butler, R., and Williams, F.W., "Optimum Design Features of VICONOPT, an Exact Buckling Program for Prismatic Assemblies of Anisotropic Plates," Proceedings of the AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, Part 2, pp. 1289–1299, 1990.

[48] Dickson, J. N., Cole, R. T., and Wang, J. T. S., "Design of Stiffened Composite Panels in the Postbuckling Range, " In Fibrous Composites in Structural Design, Eds. Lenoe, E. M., Oplinger, D. W., and Burke, J. J., Plenum Press, New York, pp. 313–327, 1980.

[49] Dickson, J. N., and Biggers, S. B., "Design and Analysis of a Stiffened Composite Fuselage Panel, " NASA CR–159302, August 1980.

[50] Shin, D.K., Gürdal, Z., and Griffin, O. H. Jr., "Minimum-Weight Design of Laminated Composite Plates for Postbuckling Performance, Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32th Structures, Structural Dynamics, and Materials Conference, Baltimore, Maryland, Part I, pp. 257-266, 1991.

[51] Bushnell, D., "PANDA2 — Program for Minimum Weight Design of Stiffened, Composite, Locally Buckled Panels," Comput. Struct., 25 (4), pp. 469–605, 1987.

[52] Shirk, M. H., Hertz, T. J., and Weisshaar, T. A., "Aeroelastic Tailoring – Theory, Practice, and Promise, " J. Aircraft, 23 (1), pp. 6–18, 1986.

[53] Lynch, R. W., and Rogers, W. A., "Aeroelastic Tailoring of Composite Materials to Improve Performance, " Proceedings of the AIAA/ASME/SAE, 17th Structures Structural Dynamics and Materials Conference, King of Prussia, PA., May 5-7, pp. 61–68, 1976.

[54] McCullers, L. A., "Automated Design of Advanced Composite Structures, " Proceedings of the ASME Structural Optimization Symposium, AMD–7, pp. 119–133, 1974.

[55] McCullers, L. A., and Lynch, R. W., "Dynamic Characteristics of Advanced Filamentary Composite Structures, " AFFDL-TR-73-111, vol. II, Sept. 1974.

[56] Haftka, R. T., "Structural Optimization with Aeroelastic Constraints: A Survey of US Applications, " Int. J. of Vehicle Design, 7 (3/4), pp. 381-392, 1986.

[57] McCullers, L. A., and Lynch, R. W., "Composite Wing Design for Aeroelastic Tailoring Requirements, " Air Force Conference on Fibrous Composites in Flight Vehicle Design, Sept. 1972.

[58] Fant, J. A., "An Advanced Composite Wing for the F-16, " paper presented at the 22nd National SAMPE Symposium and Exhibition, San Diego, pp. 773-783, April 1977.

[59] Gimmestad, D., "Aeroelastic Tailoring of a Composite Winglet for KC-135," AIAA Paper No. 81-0607, presented at the AIAA/ASME/ASCE/AHS 22nd Structures, Structural Dynamics and Materials Conference, Atlanta, GA., Part 2, pp. 373-376, April 1981.

[60] Wilkinson, K., Markowitz, J., Lerner, E., George, D., and Batill, S. M., "FASTOP: A Flutter and Strength Optimization Program for Lifting Surface Structures, " J. Aircraft, 14 (6), pp. 581-587, 1977.

[61] Neill, D.J., Johnson, E.H., and Canfield, R., "ASTROS—A Multidisciplinary Automated Structural Design Tool," J. Aircraft, 27, 12, pp. 1021-1027, 1990.

[62] Vanderplaats, G. N., "ADS — A Fortran Program for Automated Design Synthesis, " NASA-CR-177985, Sept. 1985.

[63] Stroud, W. J., "Optimization of Composite Structures," NASA TM 84544, August 1982.

[64] Nagendra, S., Haftka, R. T., Gürdal, Z., and Starnes, J. H., Jr., "Design of a Blade-Stiffened Composite Panel with a Hole," Composite Structures, Vol. 18 (3), pp. 195-219, 1991.

Name Index

Aarts, E. 148, 157
Abadie, J. 177, 206
Adali, S. 20
Adelman, H.M. 226, 251, 302, 303, 352 385
Agranoff, N. 451, 466
Anderson, M.S. 454, 466, 467
Aragon, C.R. 157
Armand, J.L. 57, 67, 69
Arora, J.S. 21, 176, 206, 254, 346
Ashley, H. 20
Atrek, E. 253
Avriel, M. 131, 154
Axelsson, O. 155

Balasubramanyam, K. 61, 69
Balling, R.J. 149, 158
Barnett, R.L. 38, 66
Barthelemy, B. 303, 346
Barthelemy, J-F. M. 206, 220, 250, 251, 304, 390, 400, 412, 413
Bartholomew, P. 252
Baruh, H. 304
Batill, S.M. 468
Beale, E.M.L. 136
Beckers, P. 20
Belsare, S. 253
Bendsøe, M.P 20, 48, 66, 240, 251, 252 406, 413
Bennett, J.A. 251
Ben-Tal, A. 413
Berke, L. 252, 350, 366, 385, 386
Bert, C.W. 465
Bertsekas, D.P. 199, 207
Biggers, S.B. 467
Bindolino, G. 290, 303
Bjorck, A. 172, 206
Blackburn, C.L. 385
Blain, W.R. 464
Booker, L. 151, 158
Botkin, M.E. 251
Box, G.E.P. 250
Brach, R.M. 68
Braibant, V. 20, 214, 250, 386
Bråmå, T. 253
Brandt, A. 19
Brandmaier, H.E. 426, 464
Brill, E.D. 157
Brotchie, J.F. 384
Broyden, C.G. 138–140, 155, 156
Bruno, R.J. 158
Burke, J.J. 467
Burns, N.H. 113
Burrell, B. P. 114
Bushnell, D. 20, 458, 467
Butler, R. 458, 467

Calladine, C.R. 74, 113
Calo, J.M. 304
Camarda, C.J. 303
Cameron, G.E. 386
Canfield, R.A. 253, 468
Caprino, G. 420, 464
Cardani, C. 277, 303

Cardoso, J. B. 346
Carpentier, J. 177, 206
Cauchy, A. 132, 155
Cerny, V. 147, 157
Chang, K.-J. 251
Charmichael, D.G. 21
Charnes, A. 113
Chen, D. H. 124, 125, 155
Chen, G.-S. 149, 158
Cheng, Kentung 437, 465
Cherkaev, A.V. 69
Chibani, L. 405, 413
Choi, K.K. 303, 345, 346
Chon, C.T. 303
Cilly, F.H. 384
Cobb, W.G.C. 253
Cohen, G.A. 345
Cohn, M.Z. 77, 113
Cole, R.T. 467
Cornell, C.A. 384
Crivelli-Visconti, I. 464
Curtis, A.R. 144, 156

Dahlquist, G. 172, 206
Dailey, R.L. 281, 303
Dantzig, G. 88, 113, 399, 413
Davidon, W.C. 127, 155
Dayaratnam, P. 385
de Wilde, W.P. 464
Decker, D.W. 157
De Jong, K.A. 151, 158
Dems, K. 346
Dennis, J.E. 142, 155, 156
Desai, R. 251
Dickson, J.N. 458, 467
Dixon, S.C. 385
Doig, A.G. 107, 114
Dorn, W.C. 241, 252
Dovi, A.R. 412
Draper, N.R. 250
Dupree, D.M. 252
Durling, B.J. 466
Dwyer, W. 385

Edgeworth 6, 20
Elperin, T. 149, 158
Emerton, R. 385
Eschenauer, H.A. 21, 346

Fadel, G.M. 221, 251
Falk, J.E. 354, 385
Fant, J.A. 468
Farshi, B. 209, 210, 249, 422, 430, 431 464, 465
Fiacco, V. 188, 206
Flaggs, D.L. 20
Fletcher, R. 127, 134, 137, 140, 155, 156 201, 207
Fleury, C. 20, 215, 250, 252, 254, 354 357, 385, 386
Fomin, S.V. 66
Fox, R. L. 405, 413
Frauenthal, J.C. 57, 67
Freedman, B. A. 114
Friedmann, P.P. 20
Fuchs, M.B. 213, 225, 249-251

Gajewski, A. 57, 68
Garfinkel, R.S. 105, 114
Ge, R. 157
Gelfand, I.M. 66
Gelatt, C.D. 157
Gellatly, R.A. 252, 386
Geoffrion, A.M. 413
George, D. 468
Ghalib, M.A. 66
Ghosh, S.K. 77, 113
Giles, G.L. 251, 385, 390, 412, 467
Gill, P.E. 145, 155, 157, 170, 206, 302
Gimmestad, D. 468
Ginsburg, S. 414
Glatt, C.R. 466
Goldberg, D.E. 146, 150, 152, 157, 158
Goldfarb, D. 140, 156
Gomrooy, R.E. 252
Grace, D.W. 125, 155
Grandhi, R.V. 20, 254
Greenberg, H.J. 113, 252
Greene, W.H. 299, 304, 467
Grierson, D.E. 361, 385, 386
Griewank, A.O. 157
Griffin, O.H., Jr. 207, 467
Gunnink, J.W. 420, 464
Gürdal, Z. 207, 440, 456, 465, 467 468

Haber, R. B. 337, 346
Haftka, R.T. 20, 21, 57, 60, 67, 110, 114 191, 206, 227, 250–254, 277 302–304, 346, 385, 396, 405, 412 413, 440, 445, 450, 465, 466, 468
Hague, D.S. 466
Hahn, H.T. 464
Haj Ali, R.M. 213, 250
Hajela, P. 20, 152, 158
Haley, S.B. 225, 251
Hancock, H. 66
Hansen, E. 157
Hansen, S.R. 250
Hariran, M. 253
Hartley, R.L. 158
Haug, E.J. 19, 179, 206, 276, 303, 345
Hayashi, T. 465
Hayduk, R.J. 115, 154
Hennessy, K.W. 466
Hertz, T.J. 467
Hestenes, M.R. 134, 155, 199, 207
Hext, G. R. 123, 124, 155
Hildebrand, F.B. 66
Himsworth, F. R. 123, 124, 155
Holland, J.H. 150, 158
Holmes, A.M.C. 20
Holnicki-Szulc, J. 225, 251
Hornbuckle, J.C. 67
Huang, N.C. 67, 139, 141

Icerman, L.J. 60, 68
Iott, J. 302
Irvine, H. M. 113
Isakson, G. 253

James, B.B. 412, 413
Johnson, D.S. 147, 157
Johnson, E.H. 253, 468
Johnson, E.L. 114
Johnson, L.W. 54, 67
Johnson, O.G. 155
Jones, R.M. 464
Junkins, J.L. 156

Kamat, M.P. 57, 60, 67, 68, 69, 115, 154 156, 157, 345, 405, 413, 465
Kandil, N. 420, 464
Kao, J.-J. 157
Kao, P.-J. 251
Karmarkar, N. 100–104, 113
Kawata, K. 465
Keller, J.B. 54, 66, 67
Kelley, C.T. 157
Kennedy, D. 467
Khot, N.S. 252, 350, 365, 366, 369 382, 385, 386
Kiefer, J. 154
Kikuchi, N. 240, 251
Kincaid, R.K. 149, 158
Kirkpatrick, S. 147, 157
Kirsch, U. 20, 83, 113, 223, 242, 251, 252, 399, 400, 413, 414
Kiusalaas, J. 68, 253
Kobayashi, A. 379
Kodiyalam, S. 250
Kohn, R.V. 240, 251
Komkov, V. 303, 345
Korst, J. 157
Koski, J. 21
Kovács, L.B. 114
Kramer, M.A. 304
Kreisselmeier, G. 160, 206, 291, 304, 404
Kruzelecki, J. 20
Kwok, H.H. 156

Laarhoven, P.J.M. van 146, 157
Land, A.H. 107, 114
Lansing, W. 385
Lawler, E.L. 114
Lee, W.H. 385, 386
Lekhnitskii, S.G. 466
Le Riche, R. 450, 466
Lerner, E. 468
Lev, O. E. 19
Lin, T.Y. 113
Lodier, B. 57, 67
Loendorf, D.D. 390, 412
Lombardi, M. 450, 466
Lowder, H.E. 213, 250
Luenberger, D. 113, 155
Lurie, K.A. 69
Lust, R.V. 249, 250
Lynch, R.W. 252, 459, 467, 468

Majid, K.I. 84, 113
Makky, S.M. 40, 66

Mangiavacchi, A. 68
Mantegazza, P. 277, 290, 303
Markowitz, J. 468
Massard, T.N. 425, 464
Massonet, C.E. 79, 113
Masur, E.F. 54, 67
Matthies, H. 143, 156
May, S.A. 149, 158
Mayers, J. 466
McCormick, G.P. 188, 206
McCormick, P.J. 20
McCullers, L.A. 252, 459, 467, 468
McGeoch, L.A. 157
Mead, R. 124, 155
Mehrinfar, M. 412
Meketon, M. S. 114
Mesquita, L. 465
Metropolis, N. 146, 147, 157
Micchelli, C.A. 155
Miele, A. 68
Miki, M. 426, 432, 438, 464, 465
Mills-Curran, W.C. 213, 249, 281, 303
Mitchell, A.G.M. 384
Miura, H. 218, 243, 250, 252–254
Moe, J. 206
Mohanty, B.P. 68
More, J.J. 156
Morris, D. 83, 113
Moses, F 18, 21, 414
Mota Soares C.A. 346, 386
Mrøz, Z. 60, 68, 345, 346
Muc, A. 434, 465
Munksgaard, N. 155
Murray, W. 145, 155, 157, 206, 302
Murthy, D.V. 227, 251, 277, 290, 303, 304

Nachlas, J.A. 251
Nagendra, G. 253
Nagendra, S. 447, 465, 468
Nahar, S. 149, 158
Narayanaswami, R. 352, 385
Neal, B. G. 113
Neill, D.J. 253, 468
Nelder, J. A. 124, 155
Nelson, R.B. 277, 303
Nemhauser, G.L. 105, 114
Niordson, F.I. 19, 67, 68
Noor, A.K. 213, 250

Olhoff, N. 20, 54, 57, 60, 67, 68, 69, 251
Ojalvo, I.U. 281, 303
Onada 18, 21
Ong, T.G. 251
Oren, S.S. 155
Osyczka, A. 21

Padula, S.L. 149, 158
Paeng, J.K. 253
Pagano, N.J. 417, 464
Palmer, A.C. 21
Pan, T.-S. 158
Parbery, R.D. 61, 69
Pardo, H. 252
Pareto, V. 6, 20
Parimi, S.R. 77, 113
Paris, G.H. 113
Park, W.J. 425, 464
Parme, R.L. 113
Pars, L.A. 66
Patnaik, S. 385
Paul, G. 155
Pedersen P. 19, 414, 433, 465
Pfeffer, J.T. 157
Phelan, D.G. 346
Pierson, B.L. 19, 68
Plank, R.J. 466
Plaut, R.H. 54, 60, 67, 68, 345, 413, 465
Polak, E. 137, 155
Powell, M.J.D. 117, 124, 127, 136, 137 142, 144, 154–156, 202, 207
Powell, S. 114
Prager, W. 19, 54, 60, 66, 68, 365, 386
Prasad, B. 20, 57, 60, 67, 214, 215, 250 253
Pritchard, J.I. 226, 251

Rabitz, H. 304
Ranalli, E. 385
Rao, S.S. 19, 152, 158, 465
Rasmussen, H. 54, 67
Rasmussen, J. 240, 252
Razani, R. 385
Reddy, G.B. 253
Reddy, J.N. 66
Reddy, V.S. 366, 386

Reeves, C.M. 134, 137, 155
Reid, J.K. 144, 156
Reinschmidt, K.F. 241, 252, 384
Riley, K.M. 206
Riley, M.F. 251, 304, 413
Rinaldi, G. 114
Ringertz, U.T. 405, 413
Rizzo, T. 250
Rogers, J.L., Jr. 253
Rogers, L.C. 277, 303
Rogers, W.A. 467
Rosen, J.B. 176, 178, 206, 399, 400, 413
Rosenbluth, A.W. 157
Rosenbluth, M.N. 157
Rozvany, G.I.N. 240, 251, 366, 386
Russel, A.D. 241, 252
Rutenbar, R.A. 157

Sahni, S. 158
Salajeghah, E. 250
Salama, M. 158
Saleem, Z. 125, 155
Salinas, D. 67
Samtani, M.P. 152, 158
Sandridge, C. A. 304
Saunders, M.A. 302
Save, M.A. 79, 113
Schevon, C. 157
Schmit, L.A. 19, 209, 210, 218, 241–243 249, 250, 252, 254, 357, 384, 385 399, 405, 412, 413, 422, 430, 431 464, 465
Schnabel, R.B. 142, 145, 155
Schrage, L. 114, 465
Schubert, L.K. 143, 156
Seong, H.G. 346
Shanno, D.F. 140, 142, 143, 156
Shephard, M.S. 251
Sheu, C.Y. 19, 67, 241, 252
Shield, R.T. 60, 68
Shin, D.K., 195, 207, 458, 467
Shin, Y.S. 413, 437, 465
Shirk, M.H. 467
Shore, C.P. 214, 250
Shragowithz, E.V. 158
Siegel, S. 352, 385
Simitses, G.J. 57, 67, 68
Singh K. 465
Smaoui, H. 405, 413
Smith, C.V. 67
Sobieszczanski-Sobieski, J. 20, 160, 174 206, 213, 249, 253, 390, 395, 400, 404 408, 412–414
Spendley, W. 123, 124, 155
Spillers, W.R. 61, 69, 414
Stadler, W. 6, 20, 21
Starnes, J.H. 191, 206, 250, 468
Stein, M. 466
Steinberg, Y. 225, 251
Steinhauser, R. 160, 206, 291, 304, 404
Stiefel, E. 134, 155
Storaasli, O.,O. 213, 249
Strang, G. 101, 114, 143, 156, 240, 251
Stroud, W.J. 353, 385, 451, 454, 466, 467 468
Sugiyama, Y. 438, 465
Sutter, T.R. 278, 303
Swanson, G.D. 456, 467
Szeto, W.T. 251
Szu, H. 152, 158

Tadikonda, S. 304
Tadjbaksh, I. 54, 66
Taye, S. 223, 251
Taylor, J.E. 20, 48, 54, 60, 65–67, 68
Teller, A.H. 157
Teller, E. 157
Thareja, R. 254, 396, 412
Thierauf, G. 346
Tischler, V.A. 252
Timoshenko, S. 466
Todd, M. J. 114
Toint, Ph.L. 143, 144, 156
Tomlin, J.A. 114
Topping, B.H.V. 242, 252
Tsach, U. 385
Tsai, S.W. 417, 464
Tseng, C.H. 254
Turner H.K., 67
Turner, M.J. 60, 65, 68

Vanderbei 114
VandenBrink, D.J. 157
Vanderplaats, G.N. 206, 219, 238, 239, 248

250, 253, 254, 414, 466, 468
Vecchi, M.P. 157
Venkayya, V.B. 19, 158, 252, 352, 365, 366 373, 380, 385, 386
Verchery, G. 420, 464

Wallerstein, D. 253
Walsh, G.R. 155
Walsh, J.L. 110, 114, 253, 303, 442, 445 465
Wang, B.P. 278, 303
Wang, J.T.S. 467
Ward, P. 253
Washizu, K. 66
Wasiutynski, Z. 19
Watson, L.T. 156, 157, 250, 465
Weisshaar, T.A. 467
Wellen, H.K. 253
Wilde, D.J., 1, 19
Wilkinson, J.H. 252, 303
Wilkinson, K. 252, 386, 468
Williams, D. 299, 304
Williams, F.W. 458, 466, 467
Wittrick, W.H. 466
Wolfe, P., 206, 399, 413
Woo, T.H. 215, 250
Wood, D.E. 114
Wright, M.H. 155, 206, 302
Wu, A.K. 68
Wu, C.C. 21

Yang, R.J. 251, 346
Yerry, M.A. 251

Zarghamee, M.S. 68
Zeleny, M. 20
Zeman, P. 113
Zoutendijk, G. 206
Zyczkowski, M. 20, 57, 68

Subject Index

ACCESS program 242
active constraints 13
active set 13
adjoint method (see sensitivity derivatives)
ADS program 243, 460
analysis (see also simultaneous analysis and design)
  •nonlinear equations 142
  •reanalysis 222–225
  •system of linear equations 142
approximations
  •conservative-convex 213–214
  •explicit 210
  •generic 211–222
  •global 210, 219–222
  •higher order 215
  •linear 210, 211
  •linear force 219, 239
  •local 210, 211–219
  •local-global 221
  •mid-range 219–222
  •multipoint 221
  •quadratic 215
  •reciprocal 213, 361, 366, 369, 375
  •reciprocal quadratic 215
  •two-point 221
approximations and move limits 210, 229
arch problems 77
artificial variables 94
ASOP program 242
ASTROS program 243, 460

beam problems 2, 38, 40, 77
BFGS update 140
bisection technique 121
block-angular decomposition 387, 389–399
block-diagonal decomposition 387, 389
bracketing 116
branch and bound algorithm 107
buckling problems 52–57, 61–63, 382 440–458
buckling load sensitivity 274–283, 323–333

cable problems 26, 31
calculus of variations 2, 29–61
central-difference derivative approximation 256
circular plate, frequency 60
classical tools 23–69
collapse load 72
collapse mechanisms 80
column problems 52–57, 61
composite laminates 415–468
  •aeroelastic tailoring 459–460
  •buckling problems 440–458
  •classical lamination theory 417, 418
  •coupling between bending, extension and shear 420
  •corrugated panel 456
  •design of stiffened panels 451–458
  •design uncertainties 460–462
  •genetic algorithm 450

- •graphical solution 426–429, 432–433, 438–440
- •integer linear programming 442–450
- •hat-stiffened panels 456
- •laminate design for flexural response 430–438
- •laminate design for in-plane response 422–429
- •mechanical response 415–421
- •orthotropic laminates 416
- •penalty function formulation 440–442
- •ply-orientation design variables 425, 433
- •ply-thickness design variables 422, 430
- •probabilistic search methods 450
- •quadratic first-ply failure 425
- •removing plies 424
- •simulated annealing 450
- •smeared stiffness 452
- •stacking convention 418
- •stacking sequence design 436–451
- •stiffened plate design 451–458
- •stiffness design 446
- •Tsai-Hill yield criterion 426
- •wing-rib panel configurations 456

concave function 166
condition error 256
conjugate directions (see Powell's conjugate directions method)
conjugate gradient methods 134–137 155, 156, 405
CONMIN program 184, 243
constraint approximations
- •conservative-convex 213–214
- •explicit 210
- •generic 211–222
- •global 210, 219–222
- •higher order 215
- •linear 210, 211
- •linear force 219, 239
- •local 210, 211–219
- •local-global 221
- •mid-range 219–222
- •multipoint 221
- •quadratic 215
- •reciprocal 213, 361, 366, 369, 375
- •reciprocal quadratic 215
- •two-point 221

constraint normalization 12
constraint relaxation 232
conservative approximation 213-214, 363
conservative concave approximation 214
convex approximation 214, 363
convex function 166, 214
convex linearization 215, 363
convex polytope 87
convex problems 166–168
convexity 166
coordination of multilevel optimization 387, 399–401
critical point constraint 292
cubic interpolation 121, 123
decomposition methods
- •block angular structure 389–399
- •block diagonal structure 389
- •computational benefits 392–399
- •global or coupling variables 396
- •implicit elimination of variables 396–399
- •local variables 396
- •relation to multilevel methods 387–388
- •response calculation 406–411
- •sensitivity calculation 406–411
- •substructures 390

derivative calculations (see sensitivity derivatives)
DESAP Program 242
design model 211
design-variable normalization 12
domain parametrization 337–339
DFP update 139
differential calculus 23–28
discrete-valued design variables 105, 195–198, 357–361
dual methods 353–365
- •Falk's dual 354, 355
- •first order approximations 361
- •integer programming 357–361
- •linear programming 96–100, 354
- •separable problems 355

dummy load method 35, 264
dynamic compliance 60
dynamic programming 21

eigenvalue sensitivity 276–290
•adjoint or modal technique 277, 289
•direct approach 277
•eigenvectors 276, 277
•modified modal method 278
•Nelson's method 277
•non-hermitian 283–289
•nonlinear problems 290
•normalization condition 289
•reduced-basis approach 285
•repeated eigenvalues 276, 281–283
•vibration 276–283
eigenvalue reanalysis 226–228
elastically supported column 67
envelope functions 160, 291, 404
equality constraints 9
Euler-Bernoulli beams 57
Euler-Bernoulli columns 52
Euler-Lagrange equations 31, 32, 37
explicit approximation 210
extended interior penalty function 190–192
exterior penalty function 187–190
extreme point 87
fast reanalysis techniques 222–228
FASOR program 305
FASTOP program 242, 371, 460
feasible direction 163
•method 182–186
feasible domain 13
Fibonacci search 118–119
finite difference approximations 256–263
•accuracy and derivative magnitude 261
•accuracy and step size selection 259
•central-difference approximation 256
•condition error 256
•iteratively solved problems 259–261
•forward-difference approximation 256
•round-off error 256
•truncation error 256
flutter sensitivity derivatives 290
Fourier series 61
frame structures 77–81
fully stressed design 15, 348–352, 390
functions of one variable 115–123
Gauss-Newton method 193
generalized reduced gradient method 177
generic approximations 211–222
genetic algorithms 146, 149–152, 450
•reproduction 151
•crossover 151
•mutation 151
geometric programming 21
geometrical optimization 239
global approximations 210, 219–222
global optimization 145
global sensitivity equation 408
golden section search 118–120
gradient projection method 176–182, 369
Green's function method 295
ground structure 240
Hessian matrix 24, 137, 138
•inverse Hessian approximations 138–140
•rank one updates 138
•rank two updates 139
•sparse update techniques 156
•sparsity 143
higher-order approximations 215
Huang's family of updates 139, 141
I-DEAS program 243
IDESIGN program 243
implicit approximation 210
inactive constraints 13
inequality constraints 9
infeasible domain 13
integer programming 4
•branch and bound algorithm 107
•linear programming 104–110
•penalty function method 195–198, 440–442
integrated analysis and design 21
interior penalty function 190, 191
internal boundaries or holes 240
intervening variables 212
Karmarkar's Algorithm 100–104
Kresselmeier-Steinhauser function 160, 291, 404
Kuhn-Tucker conditions 12, 99, 161–170 366
Kuhn-Tucker multipliers (see Lagrange multipliers)
Lagrange multipliers 13, 33, 34, 37–39

161–173, 354, 368
•as dual variables 354
•as price of constraints 174
•computation of 170–173, 371, 376
•technique 34, 162
Lagrangian function 162, 234, 354
laminate design
•aeroelastic tailoring 459–460
•buckling problems 440–458
•classical lamination theory 417, 418
•coupling between bending, extension and shear 420
•currugated panel 456
•design of stiffened panels 451–458
•design uncertainties 460–462
•genetic algorithm 450
•graphical solution 426–429, 432–433, 438–440
•integer linear programming 442–450
•hat-stiffened panels 456
•design for flexural response 430–438
•design for in-plane response 422–429
•mechaical response 415–421
•orthotropic laminates 416
•penalty function formulation 440–442
•ply-orientation design variables 425, 433
•ply-thickness design variables 422, 430
•probabilistic search methods 450
•quadratic first-ply failure 425
•removing plies 424
•simulated annealing 450
•smeard stiffness 452
•stacking convention 418
•stacking sequence design 436–451
•stiffened plate design 451–458
•stiffness design 446
•Tsai-Hill yield criterion 426
•wing-rib panel configurations 456
limit analysis and design 72–81
•arches 77
•beams 77-79
•frames 77-81
•trusses 73–77
limit load 72
limit load sensitivity 274–275, 323–326
linear approximation 210, 211
linear force approximation 219, 239
line search 14, 115–123
linear static response reanalysis 222–225
linear programming 10
•duality in 96–100
•graphical solution 86–88
•integer programming 104–110
•standard form 88
•use of 72–86
local approximation 210–219
local constraints 44
local-global approximation 221
logarithmic derivative 261
lower bound theorem 74
marginal prices 174
material derivative 334
mathematical programming 3
mesh generators 240
Metropolis algorithm (see simulated annealing)
mid-range approximations 219–222
min-max approach 44–49
move limits 210, 229
multicriteria optimization 5–9, 20, 21
multilevel techniques
•coordination 387, 399–401
•decomposition 387–399
•derivatives of subsystem optima 399–401
•discontinuous sensitivities 400
•envelope function approach 401, 404
•narrow-tree problems 387, 404–406
•penalty function approach 401–404
•wide-tree structure 387
multiple objective functions 5–9
multiplier methods 198–201
multipoint approximations 221
necessary conditions for optimality 24, 49
Nelson's method 277
NEWSUMT Program 243
NEWSUMT-A Program 243
NISAOPT program 243
Newton's method 122–123, 137–138, 157, 274
Newton-type methods 157
normalization of design variables

and constraints 12
Objective function 5
•linear 10
•multiple 5–9
one dimensional line search 14
OPSTAT Program 242
OPT Program 243
OPTCOMP Program 242
optimality criteria methods
•displacements constraints 366–370
•fully stressed design 348–352
•intuitive 348–353
•Lagrange multiplier estimation 371 376
•scaling based 372–374, 380–381
•several constraints 375–382
•single constraint 365–374
•stability constraints 382
•stress-ratio technique 351, 378, 390
•uniform cost-effectiveness resizing 371, 460
•uniform strain-energy density 352
OPTIMUM Program 242
optimization packages 242–243
•ABAQUS 243
•ACCESS 242
•ADS 243, 460
•ANSYS 243
•ASKA 243
•ASOP 242
•ASTROS 243, 460
•CONMIN 184, 243
•DESAP 242
•DOC 243
•DOT 243
•EAL 242
•FASTOP 242, 371, 460
•GENESIS 243
•I-DEAS 243
•IDESIGN 243
•NASTRAN 243
•NEWSUMT 243
•NEWSUMT-A 243
•NISA II 243
•NISAOPT 243
•OPSTAT 242
•OPT 243
•OPTCOMP 242
•OPTIMUM 242
•OPTSYS 243
•PANDA2 372
•PARS 242
•PASCO 454–458
•PROSSS 242
•SHAPE 243
•SPAR 242
•STARS 242
•STROPT 243
•TSO 242, 459
•WASP 242, 459
PANDA2 program 458
PARS Program 242
PASCO program 454–458
penalty function
•asymptotic behavior 188, 192
•exterior 187–190
•interior 190, 191
penalty function methods 186–198
•extrapolation procedure 188, 192
•extended interior 190–192
•ill-conditioning 189
•integer programming problems 195–198, 440–442
•unconstrained minimization 193–195
plastic design (see limit analysis and design)
plate problems 4, 20, 56, 60
ply-orientation variables 425, 433
ply-thickness variables 422, 430
positive definiteness 24
Powell's conjugate directions method 124 127–131
preconditioned conjugate gradient methods 137, 155, 405
prestressed concrete design 81–83
probabilistic search 145–152 (see genetic algorithms and simulated annealing)
projected Lagrangian methods 201–204
projection matrix 171
PROSSS Program 242
pseudo loads 271, 306, 308
QR factorization 172
quadratic extended penalty function 191

quadratic interpolation 117, 123
quadratic approximation 215
quadratic programming 169–170
quasi-Newton methods 138-145, 156, 157
  •rank-one updates 138–139
  •rank-two updates 139–140
Rayleigh quotient 52, 53, 58, 64, 65, 227
reanalysis techniques 222–228
reciprocal approximation 213, 361, 366 369, 375
reciprocal quadratic approximation 215
reduced gradient method 176–182
response surface 219
Rosen's decompostion algorithm 400
Rosen's gradient projection method 176–177
safeguarded polynomial interpolation 123
sandwich beams 60
scaling-based resizing 372–374, 380–381
second derivatives of static response 268
semi-analytical method 269–273
sensitivity derivatives
  •adjoint method 264, 274, 277, 294 312, 320, 331, 343
  •buckling load 274–283, 323–333
  •decomposition 406–411
  •direct method 264, 274, 277, 293, 308 319, 327, 339
  •dummy-load method 264
  •finite-difference approximations 256–263
  •flutter 290
  •global sensitivity equation 408
  •linear static analysis 263–273 306–317
  •linear structural dynamics 298–300
  •limit loads 274, 323–326
  •nonlinear analysis 273, 290, 318
  •second derivatives 268
  •shape sensitivity 269–273, 334–345
  •structural dynamics 298–300
  •variational sensitivity (see variational sensitivity analysis)
  •vibration 276–283, 327–333
sensitivity derivatives of eigenvalue problems 276–290
  •adjoint or modal technique 277, 289
  •direct approach 277
  •eigenvectors 276, 277
  •modified modal method 278
  •Nelson's method 277
  •non-hermitian 283–289
  •nonlinear problems 290
  •normalization condition 289
  •reduced-basis approach 285
  •repeated eigenvalues 276, 281–283
  •vibration 276–283
sensitivity derivatives of limit load 274–275
sensitivity derivatives of optimum solutions 173–175
sensitivity derivatives of static response 263–275, 306–327
  •accuracy problems 269, 343
  •analytical first derivatives 263–268
  •adjoint method 264, 274, 312, 343
  •computational effort 267
  •direct method 264, 274, 308, 319, 338
  •nonlinear analysis 273, 318
  •second derivatives 268
  •semi-analytical method 269–273
sensitivity derivatives of transient response 291-300
  •adjoint method 294
  •critical point constraint 292
  •direct method 293
  •equivalent constraints 291
  •equivalent integrated constraint 291
  •Green's function method 294–298
  •linear structural dynamics 298–300
  •mode-acceleration method 299
  •mode-displacement method 298
separable problems 355
sequential approximate optimization 209
sequential linear programming 210 228–236, 423
sequential nonlinear approximate optimization 236–239
sequential quadratic programming 201–204
sequential simplex method 123–127
series solution 61–63
seventy-two-bar truss 247
SHAPE program 243

shape optimization 5, 20, 239–242
shape sensitivity derivatives 269–273 333–345
shell problems 20
simplex method 88–96
•artificial variables 94
•sequential 123–127
•tableau 93
simulated annealing 146–149, 450
•cooling schedule 147
•Metropolis algorithm 146, 147
simultaneous analysis and design 11, 15 21, 404–406
simultaneous mode design approach 353
slack variables 88
stability problems 52–57, 61–63, 382 440–458
stacking sequence effects in composite design 436–451
stiffened panel design 451–458
STARS program 242
statically determinate trusses 83
steepest descent method 132–134
stress ratio technique 351, 378, 390
STROPT program 243
structures of maximum stiffness 50, 51
sufficient conditions for optimality 24 49–61, 164
surplus variables 88
ten-bar truss 237, 244, 350
test problems 244–247
•ten-bar truss 237, 238, 244, 350
•twenty-five-bar truss 245
•seventy-two-bar truss 247
topological optimization 20, 239–242
transient response sensitivity 291–300
truncation error 256
truss problems 73–77, 83–85, 395
Tsai-Hill yield criterion 426
TSO Program 242, 459
twenty-five-bar truss 245
two-point approximations 221
unconstrained optimization 115–157
•functions of one variable 115–123
•functions of several variables 123–142
•with penalty functions 193–195
uniform cost-effectiveness criterion 371, 460
unimodal function 118
univariate search 116
updated versus fixed modes 285–289
usable feasible direction 163, 183
variable-metric methods 138, 155, 156
variational sensitivity analysis 305–346
•adjoint method 312, 320, 331, 343
•direct method 308, 319, 327, 339
•linear static analysis 306–319
•limit loads 323–326
•nonlinear static analysis 318–322
•static shape sensitivity 334–345
•unfavorable computational experience 343
•vibration and buckling 323–333
vibration problems 57, 60
vibration sensitivity calculation 276–283 327–333
•effect of updating modes 285–289
•reduced-basis approach 285
VICON program 458
VICONOPT program 458
VIPASA program 453
WASP Program 242, 459
wing design 194

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids. The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

1. R.T. Haftka, Z. Gürdal and M.P. Kamat: *Elements of Structural Optimization.* 2nd rev.ed., 1990 ISBN 0-7923-0608-2
2. J.J. Kalker: *Three-Dimensional Elastic Bodies in Rolling Contact.* 1990 ISBN 0-7923-0712-7
3. P. Karasudhi: *Foundations of Solid Mechanics.* 1991 ISBN 0-7923-0772-0
4. N. Kikuchi: *Computational Methods in Contact Mechanics.* (forthcoming) ISBN 0-7923-0773-9
5. Y.K. Cheung and A.Y.T. Leung: *Finite Element Methods in Dynamics.* (forthcoming) ISBN 0-7923-1313-5
6. J.F. Doyle: *Static and Dynamic Analysis of Structures.* With an Emphasis on Mechanics and Computer Matrix Methods. 1991 ISBN 0-7923-1124-8; Pb 0-7923-1208-2
7. O.O. Ochoa and J.N. Reddy: *Finite Element Modelling of Composite Structures.* (forthcoming) ISBN 0-7923-1125-6
8. M.H. Aliabadi and D.P. Rooke: *Numerical Fracture Mechanics.* ISBN 0-7923-1175-2
9. J. Angeles and C.S. López-Cajún: *Optimization of Cam Mechanisms.* 1991 ISBN 0-7923-1355-0
10. D.E. Grierson, A. Franchi and P. Riva: *Progress in Structural Engineering.* 1991 ISBN 0-7923-1396-8
11. R.T. Haftka and Z. Gürdal: *Elements of Structural Optimization.* 3rd rev. and exp. ed. 1992 ISBN 0-7923-1504-9; Pb 0-7923-1505-7

Kluwer Academic Publishers – Dordrecht / Boston / London

Mechanics

FLUID MECHANICS AND ITS APPLICATIONS

Series Editor: R. Moreau

Aims and Scope of the Series

The purpose of this series is to focus on subjects in which fluid mechanics plays a fundamental role. As well as the more traditional applications of aeronautics, hydraulics, heat and mass transfer etc., books will be published dealing with topics which are currently in a state of rapid development, such as turbulence, suspensions and multiphase fluids, super and hypersonic flows and numerical modelling techniques. It is a widely held view that it is the interdisciplinary subjects that will receive intense scientific attention, bringing them to the forefront of technological advancement. Fluids have the ability to transport matter and its properties as well as transmit force, therefore fluid mechanics is a subject that is particularly open to cross fertilisation with other sciences and disciplines of engineering. The subject of fluid mechanics will be highly relevant in domains such as chemical, metallurgical, biological and ecological engineering. This series is particularly open to such new multidisciplinary domains.

1. M. Lesieur: *Turbulence in Fluids.* 2nd rev. ed., 1990 ISBN 0-7923-0645-7
2. O. Métais and M. Lesieur (eds.): *Turbulence and Coherent Structures.* 1991 ISBN 0-7923-0646-5
3. R. Moreau: *Magnetohydrodynamics.* 1990 ISBN 0-7923-0937-5
4. E. Coustols (ed.): *Turbulence Control by Passive Means.* 1990 ISBN 0-7923-1020-9
5. A. A. Borissov (ed.): *Dynamic Structure of Detonation in Gaseous and Dispersed Media.* 1991 ISBN 0-7923-1340-2
6. K.-S. Choi (ed.): *Recent Developments in Turbulence Management.* 1991 ISBN 0-7923-1477-8

Kluwer Academic Publishers – Dordrecht / Boston / London

Mechanics

From 1990, books on the subject of *mechanics* will be published under two series:
***FLUID* MECHANICS AND ITS APPLICATIONS**
Series Editor: R.J. Moreau
***SOLID* MECHANICS AND ITS APPLICATIONS**
Series Editor: G.M.L. Gladwell

Prior to 1990, the books listed below were published in the respective series indicated below.

MECHANICS: DYNAMICAL SYSTEMS
Editors: L. Meirovitch and G.Æ. Oravas

1. E.H. Dowell: *Aeroelasticity of Plates and Shells.* 1975 ISBN 90-286-0404-9
2. D.G.B. Edelen: *Lagrangian Mechanics of Nonconservative Nonholonomic Systems.* 1977 ISBN 90-286-0077-9
3. J.L. Junkins: *An Introduction to Optimal Estimation of Dynamical Systems.* 1978 ISBN 90-286-0067-1
4. E.H. Dowell (ed.), H.C. Curtiss Jr., R.H. Scanlan and F. Sisto: *A Modern Course in Aeroelasticity.* *Revised and enlarged edition see under Volume 11*
5. L. Meirovitch: *Computational Methods in Structural Dynamics.* 1980 ISBN 90-286-0580-0
6. B. Skalmierski and A. Tylikowski: *Stochastic Processes in Dynamics.* Revised and enlarged translation. 1982 ISBN 90-247-2686-7
7. P.C. Müller and W.O. Schiehlen: *Linear Vibrations.* A Theoretical Treatment of Multi-degree-of-freedom Vibrating Systems. 1985 ISBN 90-247-2983-1
8. Gh. Buzdugan, E. Mihăilescu and M. Radeş: *Vibration Measurement.* 1986 ISBN 90-247-3111-9
9. G.M.L. Gladwell: *Inverse Problems in Vibration.* 1987 ISBN 90-247-3408-8
10. G.I. Schuëller and M. Shinozuka: *Stochastic Methods in Structural Dynamics.* 1987 ISBN 90-247-3611-0
11. E.H. Dowell (ed.), H.C. Curtiss Jr., R.H. Scanlan and F. Sisto: *A Modern Course in Aeroelasticity.* Second revised and enlarged edition (of Volume 4). 1989 ISBN Hb 0-7923-0062-9; Pb 0-7923-0185-4
12. W. Szemplińska-Stupnicka: *The Behavior of Nonlinear Vibrating Systems.* Volume I: Fundamental Concepts and Methods: Applications to Single-Degree-of-Freedom Systems. 1990 ISBN 0-7923-0368-7
13. W. Szemplińska-Stupnicka: *The Behavior of Nonlinear Vibrating Systems.* Volume II: Advanced Concepts and Applications to Multi-Degree-of-Freedom Systems. 1990 ISBN 0-7923-0369-5
Set ISBN (Vols. 12–13) 0-7923-0370-9

MECHANICS OF STRUCTURAL SYSTEMS
Editors: J.S. Przemieniecki and G.Æ. Oravas

1. L. Frýba: *Vibration of Solids and Structures under Moving Loads.* 1970 ISBN 90-01-32420-2
2. K. Marguerre and K. Wölfel: *Mechanics of Vibration.* 1979 ISBN 90-286-0086-8

Mechanics

3. E.B. Magrab: *Vibrations of Elastic Structural Members.* 1979 ISBN 90-286-0207-0
4. R.T. Haftka and M.P. Kamat: *Elements of Structural Optimization.* 1985
   Revised and enlarged edition see under Solid Mechanics and Its Applications, Volume 1
5. J.R. Vinson and R.L. Sierakowski: *The Behavior of Structures Composed of Composite Materials.* 1986 ISBN Hb 90-247-3125-9; Pb 90-247-3578-5
6. B.E. Gatewood: *Virtual Principles in Aircraft Structures.* Volume 1: Analysis. 1989 ISBN 90-247-3754-0
7. B.E. Gatewood: *Virtual Principles in Aircraft Structures.* Volume 2: Design, Plates, Finite Elements. 1989 ISBN 90-247-3755-9

Set (Gatewood 1 + 2) ISBN 90-247-3753-2

MECHANICS OF ELASTIC AND INELASTIC SOLIDS

Editors: S. Nemat-Nasser and G.Æ. Oravas

1. G.M.L. Gladwell: *Contact Problems in the Classical Theory of Elasticity.* 1980 ISBN Hb 90-286-0440-5; Pb 90-286-0760-9
2. G. Wempner: *Mechanics of Solids with Applications to Thin Bodies.* 1981 ISBN 90-286-0880-X
3. T. Mura: *Micromechanics of Defects in Solids.* 2nd revised edition, 1987 ISBN 90-247-3343-X
4. R.G. Payton: *Elastic Wave Propagation in Transversely Isotropic Media.* 1983 ISBN 90-247-2843-6
5. S. Nemat-Nasser, H. Abé and S. Hirakawa (eds.): *Hydraulic Fracturing and Geothermal Energy.* 1983 ISBN 90-247-2855-X
6. S. Nemat-Nasser, R.J. Asaro and G.A. Hegemier (eds.): *Theoretical Foundation for Large-scale Computations of Nonlinear Material Behavior.* 1984 ISBN 90-247-3092-9
7. N. Cristescu: *Rock Rheology.* 1988 ISBN 90-247-3660-9
8. G.I.N. Rozvany: *Structural Design via Optimality Criteria.* The Prager Approach to Structural Optimization. 1989 ISBN 90-247-3613-7

MECHANICS OF SURFACE STRUCTURES

Editors: W.A. Nash and G.Æ. Oravas

1. P. Seide: *Small Elastic Deformations of Thin Shells.* 1975 ISBN 90-286-0064-7
2. V. Panc: *Theories of Elastic Plates.* 1975 ISBN 90-286-0104-X
3. J.L. Nowinski: *Theory of Thermoelasticity with Applications.* 1978 ISBN 90-286-0457-X
4. S. Łukasiewicz: *Local Loads in Plates and Shells.* 1979 ISBN 90-286-0047-7
5. C. Fiřt: *Statics, Formfinding and Dynamics of Air-supported Membrane Structures.* 1983 ISBN 90-247-2672-7
6. Y. Kai-yuan (ed.): *Progress in Applied Mechanics.* The Chien Wei-zang Anniversary Volume. 1987 ISBN 90-247-3249-2
7. R. Negruţiu: *Elastic Analysis of Slab Structures.* 1987 ISBN 90-247-3367-7
8. J.R. Vinson: *The Behavior of Thin Walled Structures.* Beams, Plates, and Shells. 1988 ISBN Hb 90-247-3663-3; Pb 90-247-3664-1

Mechanics

MECHANICS OF FLUIDS AND TRANSPORT PROCESSES

Editors: R.J. Moreau and G.Æ. Oravas

1. J. Happel and H. Brenner: *Low Reynolds Number Hydrodynamics.* With Special Applications to Particular Media. 1983 ISBN Hb 90-01-37115-9; Pb 90-247-2877-0
2. S. Zahorski: *Mechanics of Viscoelastic Fluids.* 1982 ISBN 90-247-2687-5
3. J.A. Sparenberg: *Elements of Hydrodynamics Propulsion.* 1984 ISBN 90-247-2871-1
4. B.K. Shivamoggi: *Theoretical Fluid Dynamics.* 1984 ISBN 90-247-2999-8
5. R. Timman, A.J. Hermans and G.C. Hsiao: *Water Waves and Ship Hydrodynamics.* An Introduction. 1985 ISBN 90-247-3218-2
6. M. Lesieur: *Turbulence in Fluids.* Stochastic and Numerical Modelling. 1987 ISBN 90-247-3470-3
7. L.A. Lliboutry: *Very Slow Flows of Solids.* Basics of Modeling in Geodynamics and Glaciology. 1987 ISBN 90-247-3482-7
8. B.K. Shivamoggi: *Introduction to Nonlinear Fluid-Plasma Waves.* 1988 ISBN 90-247-3662-5
9. V. Bojarevičs, Ya. Freibergs, E.I. Shilova and E.V. Shcherbinin: *Electrically Induced Vortical Flows.* 1989 ISBN 90-247-3712-5
10. J. Lielpeteris and R. Moreau (eds.): *Liquid Metal Magnetohydrodynamics.* 1989 ISBN 0-7923-0344-X

MECHANICS OF ELASTIC STABILITY

Editors: H. Leipholz and G.Æ. Oravas

1. H. Leipholz: *Theory of Elasticity.* 1974 ISBN 90-286-0193-7
2. L. Librescu: *Elastostatics and Kinetics of Aniosotropic and Heterogeneous Shell-type Structures.* 1975 ISBN 90-286-0035-3
3. C.L. Dym: *Stability Theory and Its Applications to Structural Mechanics.* 1974 ISBN 90-286-0094-9
4. K. Huseyin: *Nonlinear Theory of Elastic Stability.* 1975 ISBN 90-286-0344-1
5. H. Leipholz: *Direct Variational Methods and Eigenvalue Problems in Engineering.* 1977 ISBN 90-286-0106-6
6. K. Huseyin: *Vibrations and Stability of Multiple Parameter Systems.* 1978 ISBN 90-286-0136-8
7. H. Leipholz: *Stability of Elastic Systems.* 1980 ISBN 90-286-0050-7
8. V.V. Bolotin: *Random Vibrations of Elastic Systems.* 1984 ISBN 90-247-2981-5
9. D. Bushnell: *Computerized Buckling Analysis of Shells.* 1985 ISBN 90-247-3099-6
10. L.M. Kachanov: *Introduction to Continuum Damage Mechanics.* 1986 ISBN 90-247-3319-7
11. H.H.E. Leipholz and M. Abdel-Rohman: *Control of Structures.* 1986 ISBN 90-247-3321-9
12. H.E. Lindberg and A.L. Florence: *Dynamic Pulse Buckling.* Theory and Experiment. 1987 ISBN 90-247-3566-1
13. A. Gajewski and M. Zyczkowski: *Optimal Structural Design under Stability Constraints.* 1988 ISBN 90-247-3612-9

Mechanics

MECHANICS: ANALYSIS

Editors: V.J. Mizel and G.Æ. Oravas

1. M.A. Krasnoselskii, P.P. Zabreiko, E.I. Pustylnik and P.E. Sbolevskii: *Integral Operators in Spaces of Summable Functions.* 1976 ISBN 90-286-0294-1
2. V.V. Ivanov: *The Theory of Approximate Methods and Their Application to the Numerical Solution of Singular Integral Equations.* 1976 ISBN 90-286-0036-1
3. A. Kufner, O. John and S. Pučík: *Function Spaces.* 1977 ISBN 90-286-0015-9
4. S.G. Mikhlin: *Approximation on a Rectangular Grid.* With Application to Finite Element Methods and Other Problems. 1979 ISBN 90-286-0008-6
5. D.G.B. Edelen: *Isovector Methods for Equations of Balance.* With Programs for Computer Assistance in Operator Calculations and an Exposition of Practical Topics of the Exterior Calculus. 1980 ISBN 90-286-0420-0
6. R.S. Anderssen, F.R. de Hoog and M.A. Lukas (eds.): *The Application and Numerical Solution of Integral Equations.* 1980 ISBN 90-286-0450-2
7. R.Z. Has'minskiĭ: *Stochastic Stability of Differential Equations.* 1980 ISBN 90-286-0100-7
8. A.I. Vol'pert and S.I. Hudjaev: *Analysis in Classes of Discontinuous Functions and Equations of Mathematical Physics.* 1985 ISBN 90-247-3109-7
9. A. Georgescu: *Hydrodynamic Stability Theory.* 1985 ISBN 90-247-3120-8
10. W. Noll: *Finite-dimensional Spaces.* Algebra, Geometry and Analysis. Volume I. 1987 ISBN Hb 90-247-3581-5; Pb 90-247-3582-3

MECHANICS: COMPUTATIONAL MECHANICS

Editors: M. Stern and G.Æ. Oravas

1. T.A. Cruse: *Boundary Element Analysis in Computational Fracture Mechanics.* 1988 ISBN 90-247-3614-5

MECHANICS: GENESIS AND METHOD

Editor: G.Æ. Oravas

1. P.-M.-M. Duhem: *The Evolution of Mechanics.* 1980 ISBN 90-286-0688-2

MECHANICS OF CONTINUA

Editors: W.O. Williams and G.Æ. Oravas

1. C.-C. Wang and C. Truesdell: *Introduction to Rational Elasticity.* 1973 ISBN 90-01-93710-1
2. P.J. Chen: *Selected Topics in Wave Propagation.* 1976 ISBN 90-286-0515-0
3. P. Villaggio: *Qualitative Methods in Elasticity.* 1977 ISBN 90-286-0007-8

Mechanics

MECHANICS OF FRACTURE

Editors: G.C. Sih

1. G.C. Sih (ed.): *Methods of Analysis and Solutions of Crack Problems.* 1973 ISBN 90-01-79860-8
2. M.K. Kassir and G.C. Sih (eds.): *Three-dimensional Crack Problems.* A New Solution of Crack Solutions in Three-dimensional Elasticity. 1975 ISBN 90-286-0414-6
3. G.C. Sih (ed.): *Plates and Shells with Cracks.* 1977 ISBN 90-286-0146-5
4. G.C. Sih (ed.): *Elastodynamic Crack Problems.* 1977 ISBN 90-286-0156-2
5. G.C. Sih (ed.): *Stress Analysis of Notch Problems.* Stress Solutions to a Variety of Notch Geometries used in Engineering Design. 1978 ISBN 90-286-0166-X
6. G.C. Sih and E.P. Chen (eds.): *Cracks in Composite Materials.* A Compilation of Stress Solutions for Composite System with Cracks. 1981 ISBN 90-247-2559-3
7. G.C. Sih (ed.): *Experimental Evaluation of Stress Concentration and Intensity Factors.* Useful Methods and Solutions to Experimentalists in Fracture Mechanics. 1981 ISBN 90-247-2558-5

MECHANICS OF PLASTIC SOLIDS

Editors: J. Schroeder and G.Æ. Oravas

1. A. Sawczuk (ed.): *Foundations of Plasticity.* 1973 ISBN 90-01-77570-5
2. A. Sawczuk (ed.): *Problems of Plasticity.* 1974 ISBN 90-286-0233-X
3. W. Szczepiński: *Introduction to the Mechanics of Plastic Forming of Metals.* 1979 ISBN 90-286-0126-0
4. D.A. Gokhfeld and O.F. Cherniavsky: *Limit Analysis of Structures at Thermal Cycling.* 1980 ISBN 90-286-0455-3
5. N. Cristescu and I. Suliciu: *Viscoplasticity.* 1982 ISBN 90-247-2777-4

Kluwer Academic Publishers - Dordrecht / Boston / London