Appendix 1

MATLAB scripts

In this Appendix, sample MATLAB scripts are presented to solve proposed problems of Chapter 4, on Linear Programing, and Chapter 6, on the use of MATLAB Optimization Toolbox.

A.1 Chapter 4 – Linear Programming (Simplex method)

```
%linear programming - Simplex method – Chapter 4
% Prof. Reyolando Brasil - 2021
%
clear
clc
% problem dimensions
%
ninc=5; % number of design variable
neq=2; % number of constraint equations
%
% first tableu matrices input
%
a=[-1 1 -4 1 0;2 -1 2 0 1]; % constraint equations coefficients
b=[30 10]; % resorces vector
c=[-1 -2 1 0 0]; % objective function coefficients
f=0;
%
while(1)
%
% entering variable
%
```

```
centra=0;
for j=1:ninc
   if c(j) < centra
      centra=c(j);
      jentra=j;
   end
end
if centra==0, break, end
%
% leaving variable
%
razaoa=10000;
for i=1:neq
   razao=b(i)/a(i,jentra);
   if razao > 0
      if razao < razaoa
         razaoa=razao;
         isai=i;
      end
   end
end
%
% makes pivot unitary
%
pivo=a(isai,jentra);
for j=1:ninc
   a(isai,j)=a(isai,j)/pivo;
end
b(isai)=b(isai)/pivo;
```

```
%
% zeroes column
% above and under pivot
%
for i=1:neq
   if i ~= isai
      const=a(i,jentra);
      for j=1:ninc
         a(i,j)=a(i,j)-const*a(isai,j);
      end
      b(i)=b(i)-const*b(isai);
   end
end
const=c(jentra);
for j=1:ninc
   c(j)=c(j)-const*a(isai,j);
end
f=f-const*b(isai);
disp(a)
disp(b)
disp(c)
disp(f)
end
%
```

A.2 Chapter 6 – MATLAB Toolbox

A.2.1 Example 6.2.1 – Nonlinear cable problem

```
%file name = cable_opt.m
%cable nonlinear problem
```

```
%main program
%Total Potential Energy
%Prof. Reyolando Brasil – 2021
clear
clc
%Design variables:  x(1) = u   x(2) = v
%
Prob_data(1)=200;%vertical load V, KN
Prob_data(2)=100;%horizontal load H, KN
Prob_data(3)=1;%cable cross section, cm²
Prob_data(4)=21000;%Young's Modulus, KN/cm²
Prob_data(5)=100;%La length, cm
Prob_data(6)=100;%Lb length, cm
Prob_data(7)=50;%N0 KN
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
x0=[10,30];
%optimization function call
[x,f,ExitFlag,Output]=fminsearch('cable_obj',x0,options,Prob_data);
disp('     u       v')
disp(x)
disp('Energia Potencial Total Mínima em KJ')
disp(f)
%
%subroutine objective funtion
%file name = cable_obj.m
%cable nonlinear problem
%objective function
%Total Potential Energy
%Prof. Reyolando Brasil - february 2021
```

```
%Design variables:  x(1) = u   x(2) = v
function p=cable_obj(x,Prob_data)
%problem data
V=Prob_data(1);%vertical load V, KN
H=Prob_data(2);%horizontal load H, KN
A=Prob_data(3);%cable cross section, m²
E=Prob_data(4);%Young's Modulus,KN/m²
La=Prob_data(5);%La length, m
Lb=Prob_data(6);%Lb length, m
N0=Prob_data(7);%N0 KN
ka=E*A/La;kb=E*A/Lb;
%stretched length
LLa=sqrt(x(2)^2+(La+x(1))^2);
LLb=sqrt(x(2)^2+(Lb-x(1))^2);
%length change
da=LLa-La;db=LLb-Lb;
%axial forces
Na=ka*da;Nb=kb*db;
%strain energy
U=(N0+Na/2)*da+(N0+Nb/2)*db;
%work of external conservative forces
W=H*x(1)+V*x(2);
%Total Potential Energy
p=U-W;
```

A.2.2 Example 6.2.2 – Eccentrically loaded tubular column

```
%Main Program
% file name column.m
% optimization of tubular steel column with eccentric loading
```

```
% Prof. Reyolando Brasil 2021
clear
clc
% options
options=optimset('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
% limits of design variables, thickness and radius
Lb=[0.01 0.005];Ub=[1 0.2];
% initial design
x0=[1 0.2];
% call the constrained optimization routine
[x,FunVal,ExitFlag,Output]=...
   fmincon('coluna_objf',x0,[],[],[],[],Lb,Ub,'coluna_conf',options)
%
%Subroutine with the objective function
% file name column_objf.m
% objective function
% Prof. Reyolando Brasil 2021
function f=column_objf(x)
% renaming design variables
x1=x(1);x2=x(2);
% data
L=5.0; % column length (m)
rho=7850; % steel density (kg/m^3)
% objective function
A=2*pi*x1*x2;
f=A*L*rho; %column mass
%
%Subroutine with the constraint functions
% file name column_conf.m
```

```
% constraints
% Prof. Reyolando Brasil 2021
function [g,h]=column_conf(x)
% renaming design variables
x1=x(1);x2=x(2);
%data
P=50000; % vertical compressive load (N)
E=210e9; % elastic modulus (Pa)
L=5.0; % column length (m)
Sy=250e6; % allowable stress (Pa)
Delta=0.25; % allowable displacement at column top
% geometric characteristics
A=2*pi*x1*x2; % section area
W=pi*x1^2*x2; % bending modulus
I=pi*x1^3*x2; % moment of inertia
e=0.02*x1; % load eccentricity
%inequality constraints
g(1)=P/A*(1+e*A/W*sec(L*sqrt(P/E/I)))/Sy-1; % allowable stress
g(2)=1-pi^2*E*I/4/L^2/P; % buckling
g(3)=e*(sec(L*sqrt(P/E/I))-1)/Delta-1; % displacement at column top
g(4)=x1/x2/50-1; % Radius/thickness ratio
% equality constraints (none)
h=[];
```

A.2.3 Example 6.2.3 – Statically loaded redundant truss

```
%file name = tre_redun_opt.m
%redundant truss optimization
%main program
%Prof. Reyolando Brasil - 2021
```

```
clc
clear
%problem data
Prob_data(1)=1;%gravity load P, KN
Prob_data(2)=1000;%density, kg/m³
Prob_data(3)=10;%alowble stress, KN/m²
Prob_data(4)=100;%Young's Modulus, KN/m²
%options
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
%Lower and upper bounds of design variables
Lb=[0 0];Ub=[1 1];
%initial design
x0=[0.1 0.1];
%optimization function call
[x,FunVal,ExitFlag,Output]=...
   fmincon('tre_redun_obj',x0,[],[],[],[],Lb,Ub,'tre_hiper_con',options,Prob_data)
%
% subroutine objective function
%file name = tre_redun_obj.m
%redundant truss optimization
%objective function
%Prof. Reyolando Brasil - 2021
function f=tre_redun_obj(x,Prob_data)
%design variables
x1=x(1);%vertical bar transverse section area, m²
x2=x(2);%diagonal bar transverse section area, m²
%material parameters
rho=Prob_data(2);%density, kg/m³
%objective function, truss total mass (kg)
```

```
f=rho*(3*x1+2*sqrt(2)*x2);
%
%subroutine constraint functions
%file name = tre_redun_con.m
%redundant truss optimization
%constraint equations
%Prof. Reyolando Brasil - 2021
function [g,h]=tre_redun_con(x,Prob_data)
%design variables
x1=x(1);%vertical bar transverse section area, m²
x2=x(2);%diagonal bar transverse section area, m²
%problem data
PP=Prob_data(1);%gravity load P, KN
Ta=Prob_data(3);%alowble stress, KN/m²
E=Prob_data(4);%Young's Modulus, KN/m²
%solution
K=E*[x1+x2*sqrt(2)/4 -x1;-x1 x1+x2*sqrt(2)/4];%stiffness matrix
P=[0;-PP];%loading vector
p=K\P;%displacements vector
%bars normal forces
N1=E*x1*(p(1)-p(2));
N4=-E*x2*p(2)/2;
N5=-E*x2*p(1)/2;
%inequality constraints
g(1)=(N1/x1)/Ta-1;
g(2)=(N4/x2)/Ta-1;
g(3)=(N5/x2)/Ta-1;
%equality constraints (none)
h=[];
```

A.2.4 Example 6.2.4 – Frequency optimization of a redundant truss

```
%main program
%file name = tre_freq_opt.m
%redundant truss optimization for frequency constraints
%Prof. Reyolando Brasil - 2021
clc
clear
%problem data
Prob_data(1)=1000;%density, kg/m³
Prob_data(2)=100000;%Young's Modulus, N/m²
%options
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
%Lower and upper bounds of design variables
Lb=[0.01 0.01];Ub=[1 1];
%initial design
x0=[0.01 0.01];
%optimization function call
[x,FunVal,ExitFlag,Output]=...
   fmincon('tre_freq_obj',x0,[],[],[],[],Lb,Ub,'tre_freq_con',options,Prob_data)
%
%subroutine objective function
%file name = tre_freq_obj.m
%redundant truss optimization with frequency constraints
%Prof. Reyolando Brasil - 2021
function f=tre_freq_obj(x,Prob_data)
%design variables
x1=x(1);%vertical bar transverse section area, m²
x2=x(2);%diagonal bar transverse section area, m²
```

```
%material parameters
rho=Prob_data(1);%density, kg/m³
%objective function, truss total mass (kg)
f=rho*(3*x1+2*sqrt(2)*x2);
%
%subroutine constraint equations
%file name = tre_freq_con.m
%redundant truss optimization with frequency constraints
%Prof. Reyolando Brasil - 2021
function [g,h]=tre_freq_con(x,Prob_data)
%design variables
x1=x(1);%vertical bar transverse section area, m²
x2=x(2);%diagonal bar transverse section area, m²
%problem data
rho=Prob_data(1);%density, kg/m³
E=Prob_data(2);%Young's Modulus, N/m²
%solution of eigenvalue problem
K=E*[x1+x2*sqrt(2)/4 -x1;-x1 x1+x2*sqrt(2)/4];%stiffness matrix
M=rho*[x1+x2*sqrt(2)/2 0;0 x1+x2*sqrt(2)/2];%mass matrix
frs=eig(K,M);%squared frequencies in rad/s
fhz=sqrt(sort(frs))/2/pi;%frequencies in Hz
%
%inequality constraints
g(1)=1-fhz(1);%f1 larger then 1 Hz
g(2)=fhz(2)-2;%f2 less then 2 Hz
%equality constraints (none)
h=[];
```

A.2.5 Example 6.2.5 – Thickness optimization of a rectangular steel plate simply supported under uniformly distributed loading and its own weight

```
%main program
%file name = plate_opt.m
%optimization of simply supported rectangular steel plate
%uniformly loaded and own weight
%design variable thickness x, m
%Prof. Reyolando Brasil 2021
clc
clear
%options
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
%Lower and uper bounds of plate thickness
Lb=0.01;Ub=0.1;%m
%initial design = initial thickness
x0=0.025;%m
Prob_data(1)=4;%lenght in x direction, m
Prob_data(2)=4;%lenght in y direction, m
Prob_data(3)=10000;%uniform loading, N/m²
Prob_data(4)=7850;%steel density, kg/m³
Prob_data(5)=15e7;%steel alowble stress, N/m²
Prob_data(6)=210e9;%steel Young's Modulus, N/m²
Prob_data(7)=0.3;%Poisson's ratio
%optimization function call
[x,FunVal,ExitFlag,Output]=...
   fmincon('plate_obj',x0,[],[],[],[],Lb,Ub,'plate_con',options,Prob_data)
%
%subroutine objective function
%file name = plate_obj.m
%optimization of simply supported rectangular steel plate
%uniformly loaded and own weight
```

```
%design variable thickness x, m
%objective function
%Prof. Reyolando Brasil february 2021
function f=plate_obj(x,Prob_data)
a=Prob_data(1);%lenght in x direction, m
b=Prob_data(2);%lenght in y direction, m
rho=Prob_data(4);%steel density, kg/m³
%objective function, the plate total mass, kg
f=rho*a*b*x;
%
%subroutine constraint functions
%file name = plate_con.m
%optimization of simply supported rectangular steel plate
%uniformly loaded and own weight
%design variable thickness x, m
%Prof. Reyolando Brasil 2021
function [g,h]=plate_con(x,Prob_data)
%problem data
a=Prob_data(1);%lenght in x direction, m
b=Prob_data(2);%lenght in y direction, m
q=Prob_data(3);%uniform loading, N/m²
rho=Prob_data(4);%steel density, kg/m³
Ta=Prob_data(5);%steel alowble stress, N/m²
E=Prob_data(6);%steel Young's Modulus, N/m²
nu=Prob_data(7);%Poisson's ratio
%
D=E*x^3/12/(1-nu^2);
w=x^2/6;
%
```

```
r=round(b/a,1);%b/a ratio, between 1 and 2, 11 possible values, round to first decimal

k=10*(r-1)+1;%table position

qmp=q+10*rho*x;%gravity acceleration 10 m/s²

%

%Table 8 "Theory of Plates an Shells", Timoshenko

%

alfa=[0.00406;  0.00485;  0.00564;    0.00638;  0.00705;  0.00772;  0.00830;  0.00883;  0.00931; 0.00974; 0.01013];

%

beta=[0.0479; 0.0554; 0.0627;  0.0694; 0.0755; 0.0812; 0.0862; 0.0908; 0.0948; 0.0985; 0.1017];

%

vmax=alfa(k)*qmp*a^4/D;%maximum vertical displacement, m

%

Mx_max=beta(k)*qmp*a^2;%maximun bending moment Mx, Nm/m

%constraints

g(1)=(Mx_max/w)/Ta-1;

g(2)=400*vmax/a-1;

%

h=[];
```

A.2.6 Example 6.2.6 – Redundant wood planar portal frame

```
%main program

%file name = port_wood_opt.m

%optimization of redundant wood planar portal frame

%Prof. Reyolando Brasil - 2021

clc

clear

%portal frame data

h=3;%clumn height, m
```

```
L=6;%beam span, m
b=0.075;%section b dimension, m
P=15;%design load at bem mid span, KN
%options
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
%Variables lower and upper bounds
Lb=[0.15 0.15];Ub=[0.4 0.4];
%initial dimensions
x0=[0.15 0.15];
%optimization function call
[x,FunVal,ExitFlag,Output]=...
   fmincon('port_wood_obj',x0,[],[],[],[],Lb,Ub,'port_wood_con',options,h,L,b,P)
%
%subroutine objective function
%file name = port_wood_obj.m
%optimization of redundant wood planar portal frame
%Prof. Reyolando Brasil - 2021
function f=port_wood_obj(x,h,L,b,P)
%design variables
x1=x(1);%column section d dimension, m
x2=x(2);%beam section d dimension, m
%wood parameters
rho=1000;%density, kg/m³
%objective function, portal frame total mass
f=rho*(2*h*b*x1+L*b*x2);
%
%subroutine constraint equations
%file name = port_wood_con.m
%optimization of redundant wood planar portal frame
```

```
%Prof. Reyolando Brasil - 2021
function [g,h]=port_wood_con(x,h,L,b,P)
%
%design variables
x1=x(1);%column section d dimension, m
x2=x(2);%beam section d dimension, m
Ac=b*x1;%column section area, m²
Av=b*x2;%beam section area, m²
Wc=b*x1^2/6;%column section flexural modulus, m³
Wv=b*x2^2/6;%beam section flexural modulus, m³
Ic=b*x1^3/12;%column section moment of inertia, m^4
Iv=b*x2^3/12;%beam section moment of inertia, m^4
%wood parameters
E=15e6;%efective Young's Modulus, KN/m²
fcd=20e3;%design resistance, KN/m²
%axial forces and bending moments
X=(P*L^2*h/8/Iv)/(2*h^3/3/Ic+L*h^2/Iv);
Nc=P/2;%KN
Mc=X*h;%KNm
%
FE=pi^2*E*Ic/h^2;%Euler's buckling load
e=(Mc/Nc+h/300)*(FE/(FE-Nc));
Mc=Nc*e;
%
Nv=X;%KN;
Mv=P*L/4-X*h;%KNm
%
FE=pi^2*E*Iv/L^2;%Euler's buckling load
e=(Mv/Nv+L/300)*(FE/(FE-Nv));
```

```
Mv=Nv*e;
%
%inequlitiy constraints
g(1)=(Nc/Ac+Mc/Wc)/fcd-1;
g(2)=(Nv/Av+Mv/Wv)/fcd-1;
%equality constraints (none)
h=[];
```

6.2.7 Example 6.2.7 –Linearly constrained profit maximization of a toy factory production

```
%file name = profit_opt.m
%toy factory maximum profit
%
%main program
%design variables
%x1=number of type A aircraft sold
%x2=number of type B aircraft sold
%Prof. Reyolando Brasil 2021
clc
clear
%options
options=optimset ('LargeScale','off','TolCon',1e-8,'TolX',1e-8);
%Lower and uper bounds
Lb=[0 0 0 0 0];Ub=[15 15 15 15 15];%
%initial design = initial thickness
x0=[1 1 1 0 0];%m
Aeq=[1 1 1 0 0;1/28 1/14 0 1 0;1/14 1/24 0 0 1];
beq=[16 1 1];
%optimization function call
[x,FunVal,ExitFlag,Output]=...
  fmincon('profit_obj',x0,[],[],Aeq,beq,Lb,Ub,[],options)
```

```
%
%file name = profit_obj.m
%airplane factory maximum profit
%
%design variables
%x1=number of type A aircraft sold
%x2=number of type B aircraft sold
%objective function
%Prof. Reyolando Brasil 2021
function [f]=profit_obj(x)
%objective function
c=[-400 -600 0 0 0];
f=c*x';
%
```