PMR5250 – Método de Otimização Topológica Aplicada ao Projeto Mecânico

Aula 10 – Método TOBS

Prof. Dr. Emílio C. N. Silva
Prof. Dr. Renato Picelli

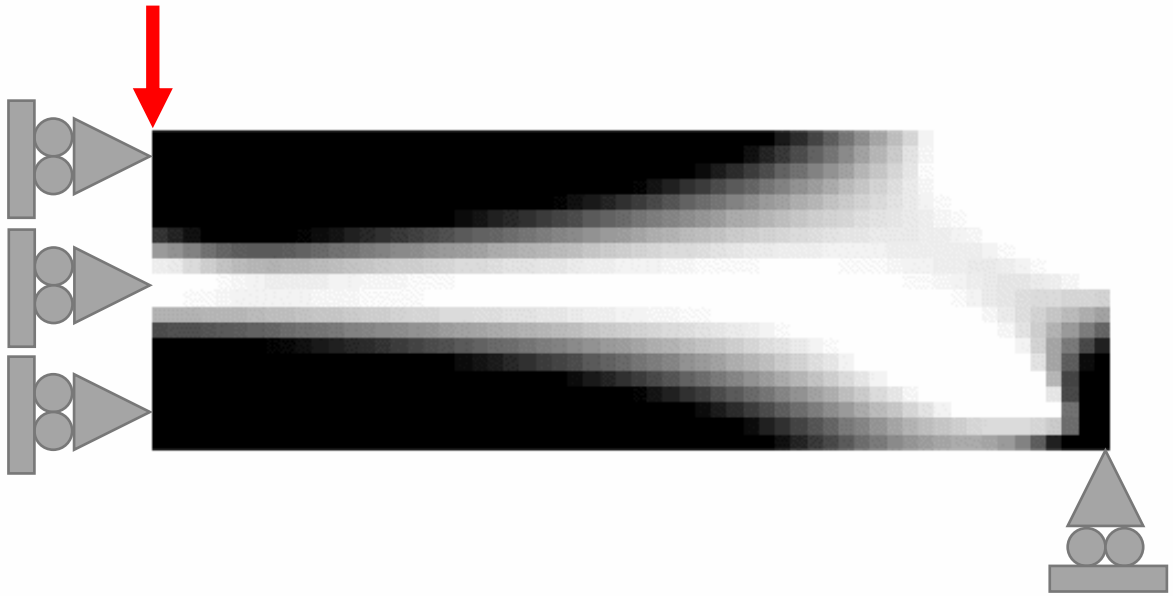
Departamento de Engenharia Naval e Oceânica
Escola Politécnica da Universidade de São Paulo

Resumo até o momento

- Análise de sensibilidade
  - Método das diferenças finitas
  - Método direto
  - Método adjunto
  - Método semi-analítico
- Método de densidade – baseado em SIMP
  - Aplicação de filtro numérico
  - Otimizadores OC e MMA

PNV USP

Algoritmo



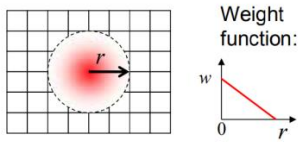
1. Definir propriedades de material de parâmetros de otimização;
2. Discretizar domínio de projeto;
3. Realizar análise por elementos finitos;

$$E_i = \rho_i^p E_0 \qquad \mathbf{Ku} = \mathbf{f}$$

4. Calcular sensibilidades;

$$\frac{df(\mathbf{x})}{dx_i} = -\frac{1}{2} p x_i^{p-1} \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i \qquad \frac{dg}{dx_i} = \frac{V_i}{V_0}$$

5. Aplicar filtro numérico;



6. Resolver otimização (otimizador);
7. Atualizar variáveis de projeto;
8. Análise de convergência.

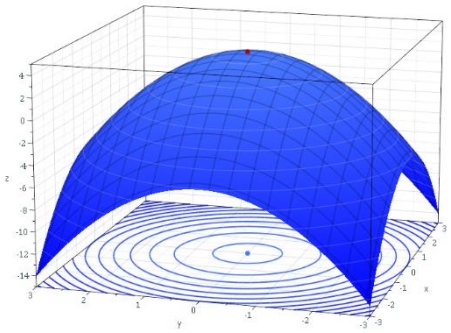
Método discreto – binário – {0,1}

Como resolver otimização topológica binária {0,1} com **programação matemática**?

PNV USP

Programação matemática

Em matemática, o termo *otimização*, ou *programação matemática*, refere-se ao estudo de problemas nos quais se busca minimizar ou maximizar uma função por meio de uma escolha sistemática de variáveis reais ou inteiras dentro de um espaço factível.



Algoritmos para se resolver o problema de otimização

$$\begin{aligned} \min \quad & f(x_i) \\ \text{s. t.} \quad & g_j(x_i) \leq 0 \\ & 0 < x_{min} \leq x_i \leq 1 \end{aligned}$$

PNV USP

Receita numérica

- Ingredientes numéricos:
  - 1. aproximação linear sequencial
  - 2. limites móveis
  - 3. filtro de sensibilidade
  - 4. otimizador de variáveis inteiras

PNV USP

Método TOBS

Finite Elements in Analysis and Design 139 (2018) 49–61

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

Topology optimization of binary structures using Integer Linear Programming

R. Sivapuram[a,*], R. Picelli[b]

[a] Structural Engineering, University of California, San Diego, La Jolla, CA, 92093, USA
[b] Cardiff School of Engineering, Cardiff University, Queen's Buildings, 14-17, The Parade, Cardiff, CF24 3AA, United Kingdom

O método TOBS

- Formulação do problema

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{Minimize}} \quad & f(\boldsymbol{x}), \\
\text{Subject to} \quad & g_i(\boldsymbol{x}) \leq \overline{g}_i, \;\; i \in [1, N_g], \\
& x_j \in \{0, 1\}, \;\; j \in [1, N_d],
\end{aligned}
$$

PNV USP

O método TOBS

- Funções aproximadas

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k + O(||\Delta \boldsymbol{x}^k||_2^2),$$

$$g_i(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \frac{\partial g_i(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k + O(||\Delta \boldsymbol{x}^k||_2^2),$$

PNV USP

O método TOBS

- Funções aproximadas

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^k) + \frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k,$$

$$g_i(\boldsymbol{x}) \approx g_i(\boldsymbol{x}^k) + \frac{\partial g_i(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k,$$

PNV USP

O método TOBS

- Restrição de variável binária

$$\begin{cases} 0 \leq \Delta x_j^k \leq 1 & \text{if } x_j^k = 0, \\ -1 \leq \Delta x_j^k \leq 0 & \text{if } x_j^k = 1, \end{cases}$$

$$\Delta x_j^k \in \{-x_j^k, 1 - x_j^k\}.$$

O método TOBS

- Restrição para garantir que o erro de truncamento seja pequeno (restrição de truncamento)

$$||\Delta \boldsymbol{x}^k||_1 \leq \beta N_d.$$

O método TOBS

- Subproblema de otimização – problema aproximado

$$\underset{\Delta \boldsymbol{x}^k}{\text{Minimize}} \ \frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k,$$

$$\text{Subject to} \ \frac{\partial g_i(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}^k \leq \overline{g}_i - g_i(\boldsymbol{x}^k) := \Delta g_i^k,$$

$$||\Delta \boldsymbol{x}^k||_1 \leq \beta N_d,$$

$$\Delta x_j^k \in \{-x_j^k, 1 - x_j^k\}, \ j \in [1, N_d].$$

O método TOBS

- Limites móveis
  - Relaxação das restrições

$$\Delta g_i^k = \begin{cases} -\epsilon_i g_i\left(\boldsymbol{x}^k\right) & : \overline{g}_i < (1-\epsilon_i) g_i\left(\boldsymbol{x}^k\right), \\ \overline{g}_i - g_i\left(\boldsymbol{x}^k\right) & : \overline{g}_i \in [(1-\epsilon_i) g_i\left(\boldsymbol{x}^k\right), (1+\epsilon_i) g_i\left(\boldsymbol{x}^k\right)], \\ \epsilon_i g_i\left(\boldsymbol{x}^k\right) & : \overline{g}_i > (1+\epsilon_i) g_i\left(\boldsymbol{x}^k\right), \end{cases}$$

  - Restrição do erro de truncamento

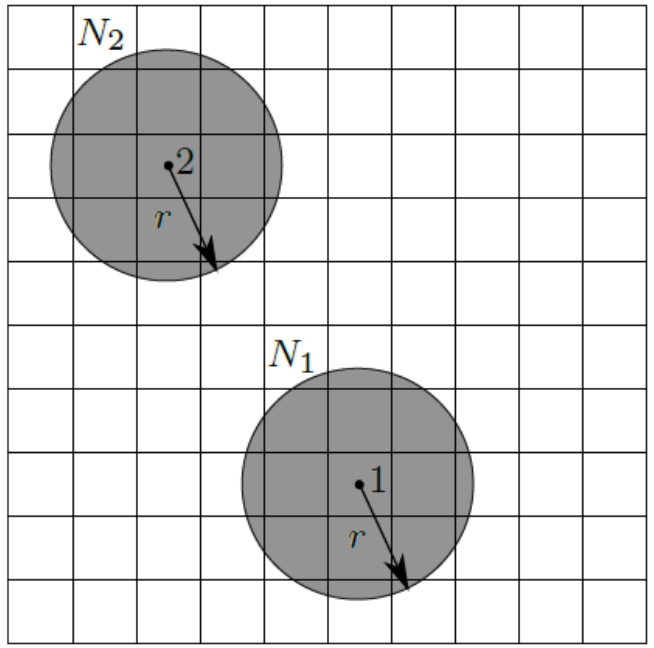
$$||\Delta \boldsymbol{x}^k||_1 \leq \beta N_d.$$

PNV USP

O método TOBS

- Filtro de sensibilidade

$$\frac{\partial f}{\partial \rho_e} = \frac{\sum_{m \in N} w_{nm} \frac{\partial f}{\partial y_m}}{\sum_{m \in N} w_{nm}}$$



O método TOBS

- Filtro de sensibilidade (estabilização temporal)

$$\frac{\widetilde{\partial f}}{\partial x_j}^{k} \leftarrow \frac{\frac{\widetilde{\partial f}}{\partial x_j}^{k} + \frac{\widetilde{\partial f}}{\partial x_j}^{k-1}}{2}$$

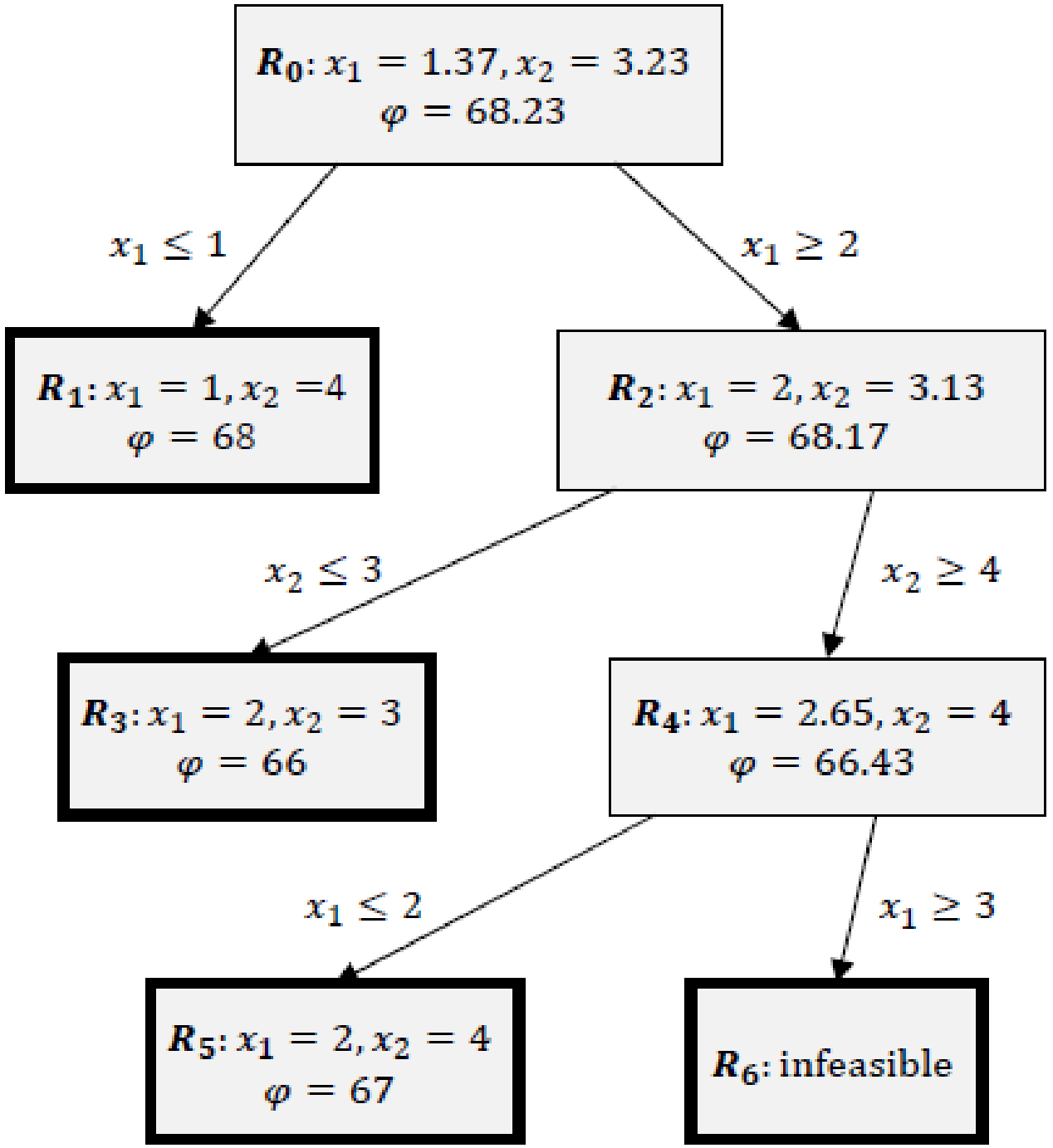
O método TOBS

- Otimização de programação inteira (*branch-and-bound*)
  - Intlinprog (Matlab)
  - CPLEX (IBM)

PNV USP

O método TOBS

- *Branch-and-bound*
  - Intlinprog (Matlab)
  - CPLEX (IBM)



O método TOBS

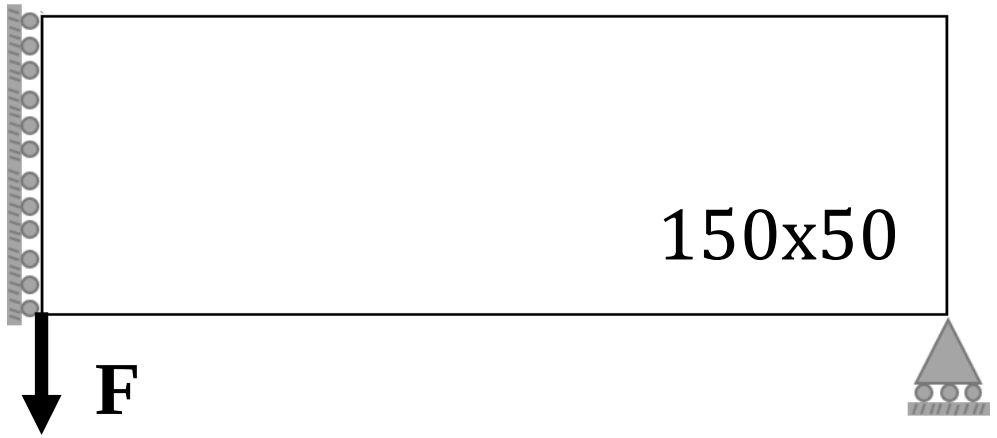
- *Branch-and-bound*
  - Intlinprog (Matlab)
  - CPLEX (IBM)

Atualização das pseudo-densidades: $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} + \Delta\boldsymbol{x}^{k}$

PNV USP

Exemplo

$$\begin{aligned} \text{Minimize} \quad & \mathbf{F}^T\mathbf{u} \\ \text{Subject to} \quad & V \leq 50\% \\ & x_i = \{0,1\} \end{aligned}$$



$E = 1.0$

$\nu = 0.3$

$\epsilon = 0.01$ → Relaxação da restrição

$\beta = 0.05$ → Restrição da truncagem

$r_{\min} = 5$

Educational Paper | Published: 27 September 2020

A 101-line MATLAB code for topology optimization using binary variables and integer programming

Renato Picelli, Raghavendra Sivapuram & Yi Min Xie

Structural and Multidisciplinary Optimization (2020) | Cite this article

- Raio do filtro, malha e penalidade.

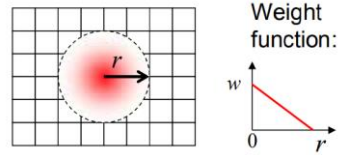
1. Definir propriedades de material de Parâmetros de otimização;
2. Discretizar domínio de projeto;
3. Realizar análise por elementos finitos;

$$E_i = E_{min} + x_i^p (E_0 - E_{min}) \qquad \mathbf{Ku} = \mathbf{f}$$

4. Calcular sensibilidades;

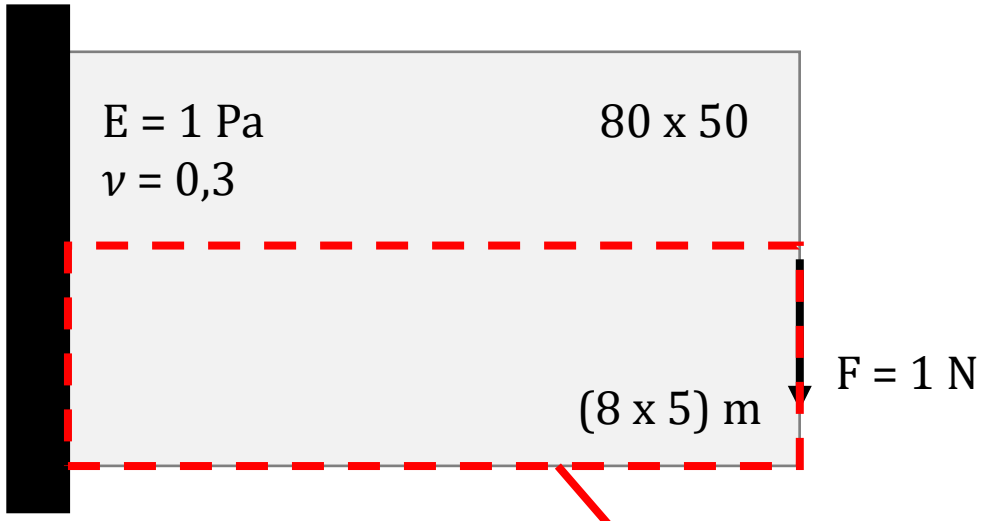
$$\frac{df(\mathbf{x})}{dx_i} = -\frac{1}{2} p x_i^{p-1} (E_0 - E_{min})\, \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i \qquad \frac{dg}{dx_i} = \frac{V_i}{V_0}$$

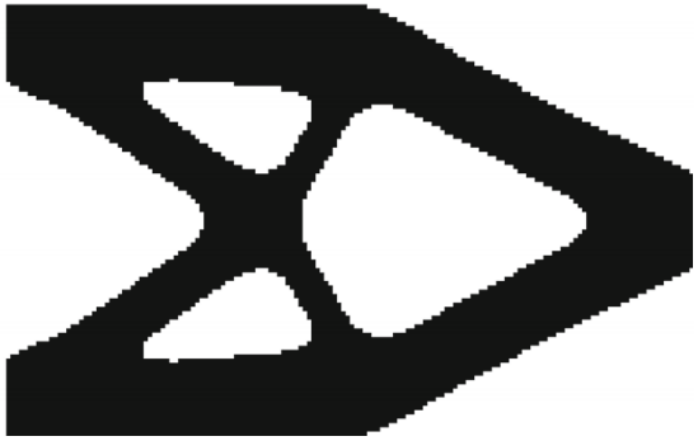
5. Aplicar filtro numérico;



6. Média histórica da sensibilidade (estabilização);
7. Resolver o subproblema linearizado;
8. Atualizar variáveis: $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k$
9. Convergiu?

Simetria





solução: resolver o problema de ILP (programação linear inteira) somente na parte simétrica.

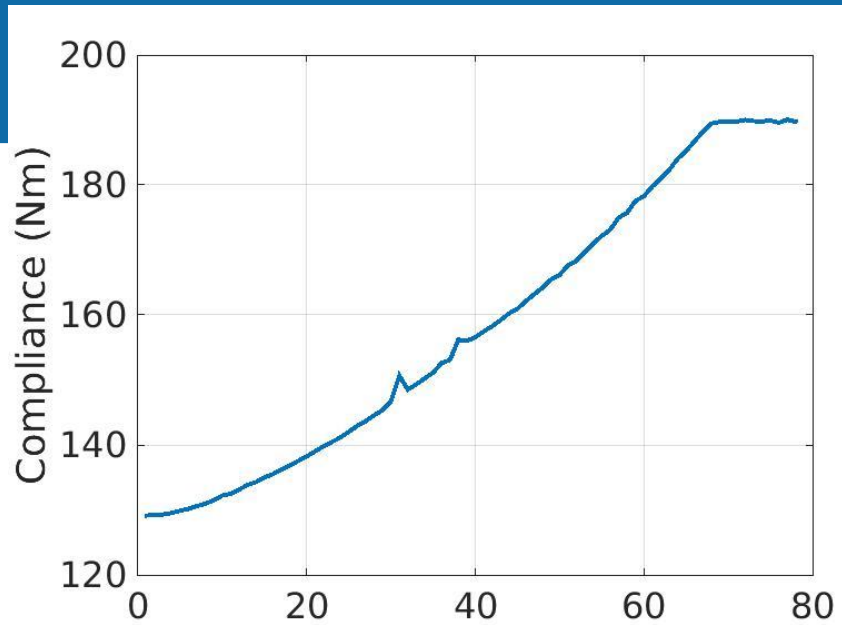
Critério de convergência

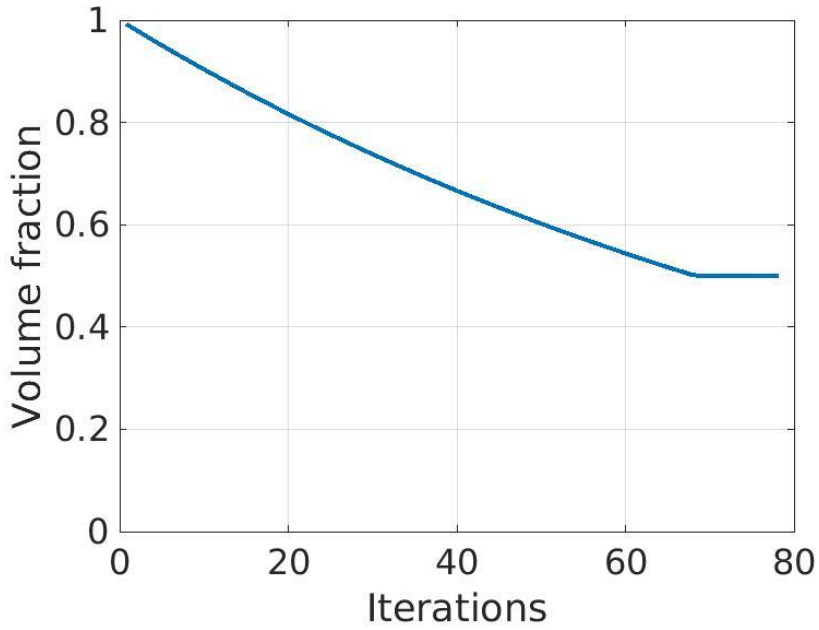
- Mudança na função objetivo.

$$error = \frac{\left|\sum_{i=1}^{N}(C_{k-i+1} - C_{k-N-i+1})\right|}{\sum_{i=1}^{N} C_{k-i+1}} \leqslant \tau$$

Exercício 1

Reproduzir esse resultado usando o seu código.





PNV USP

Exercício 1

Reproduzir esse resultado usando o seu código.

```
% Define mesh dimensions
Lx = 150;
Ly = 50;
% Number of elements in the mesh
nelx = 150;
nely = 50;

% Material properties
E = 1.0;    % Young's modulus
nu = 0.3;   % Poisson's ratio
%
penalty = 3;
volfrac = 0.5;
radius = 5;
%
beta = 0.05;
epsilons = 0.01;
```

PNV USP

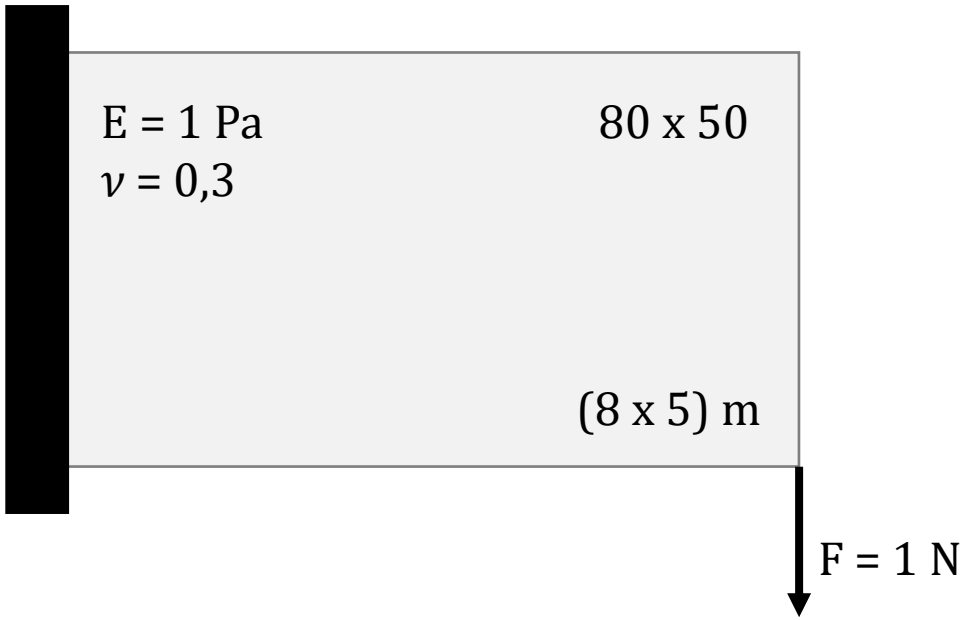
Comentários

- Programação inteira tem mais mínimos locais que otimização com variáveis contínuas.

- Contudo, o método TOBS também é convergente e, dependendo, independente da malha.

PNV USP

Exercício 2

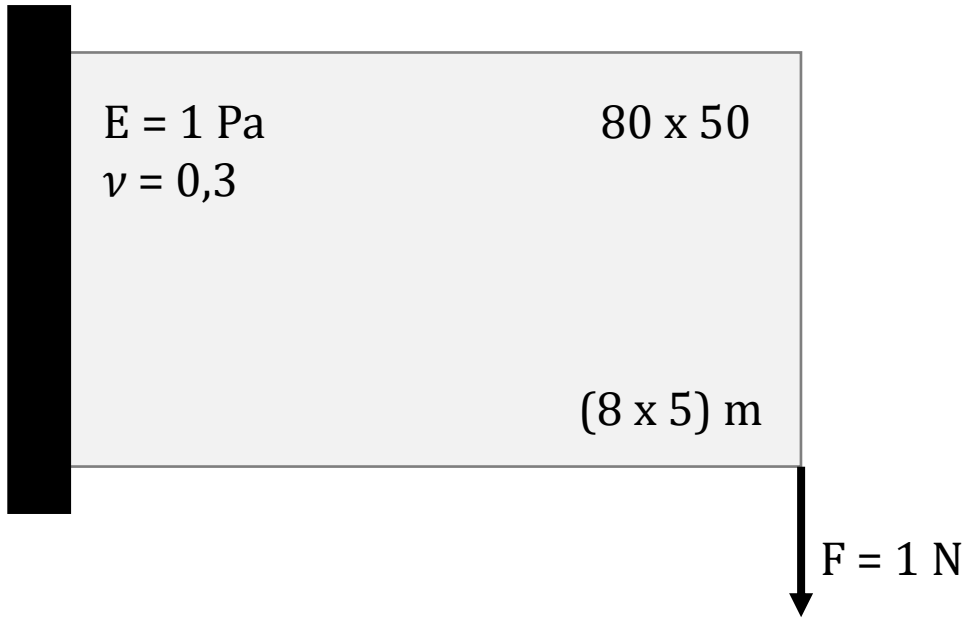


$$\begin{aligned} &\text{minimizar} && V(\mathbf{x}) \\ &\text{sujeito a} && \mathbf{F}^T\mathbf{U} \leq \bar{C} \\ & && x_i = \{0,1\} \end{aligned}$$

```
% Optimization parameters
radius = 0.3;                    % Filter radius in length unit
rho_min = 0.001;                 % Minimum density (for void elements)
compliance_constraint = 70;      % Compliance constraint
epsilons = 0.001;                % Constraint relaxation parameter
flip_limits = 0.10;              % Flip limits
tau = 0.001;                     % Convergence tolerance
```

PNV USP

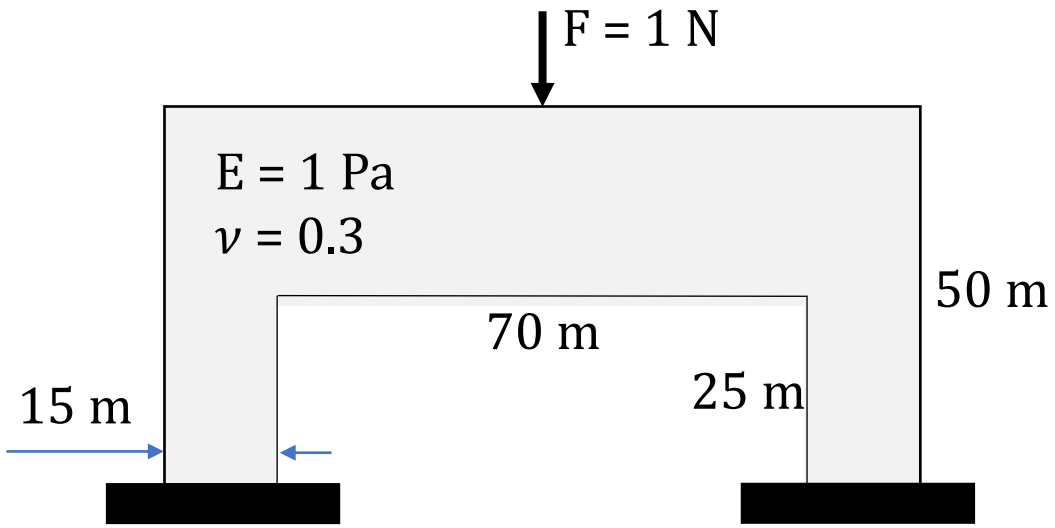
Exercício 2



$$\begin{aligned} &\text{minimizar} && \mathrm{V}(\mathbf{x}) \\ &\text{sujeito a} && \mathbf{F}^T\mathbf{U} \leq \bar{C} \\ & && x_i = \{0,1\} \end{aligned}$$

Rodar para três valores diferentes de $\bar{C}$ e explicar os resultados.

Exercício 3

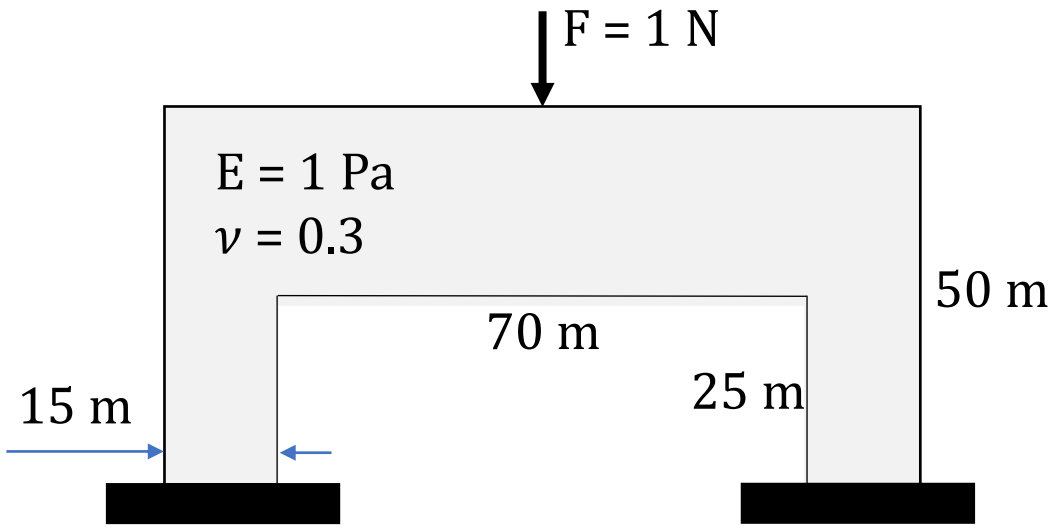


$$\begin{aligned} &\text{minimizar} && V(\mathbf{x}) \\ &\text{sujeito a} && \mathbf{F}^T\mathbf{U} \leq \bar{C} \\ & && x_i = \{0,1\} \end{aligned}$$

```
% Optimization parameters
radius = 3;                          % Filter radius in length unit
rho_min = 0.001^3;                   % Minimum density (for void elements)
compliance_constraint = 15;          % Compliance constraint
epsilons = 0.01;                     % Constraint relaxation parameter
flip_limits = 0.10;                  % Flip limits
tau = 0.001;                         % Convergence tolerance
```

PNV USP

Exercício 3



$$\begin{aligned} \text{minimizar} \quad & V(\mathbf{x}) \\ \text{sujeito a} \quad & \mathbf{F}^T\mathbf{U} \leq \bar{C} \\ & x_i = \{0,1\} \end{aligned}$$

Para resolver esse problema, altere a solução (projeto) inicial para representar a topologia estrutural acima. Crie um vetor com os números de cada elemento passivo (que permanecerão $x_i = 0$ para manter o vão da estrutura vazio). Esses elementos passivos devem ter sua sensibilidade prescritas como 0. Isso inativará os elementos na otimização.

PNV USP

Exemplo do portal (domínio passivo - não otimizável)

- Elementos passivos

```
passive = zeros(length(fea.mesh.incidence),1);
```

  - Loop nos elementos
    - Se o centróide do elemento está no vão:

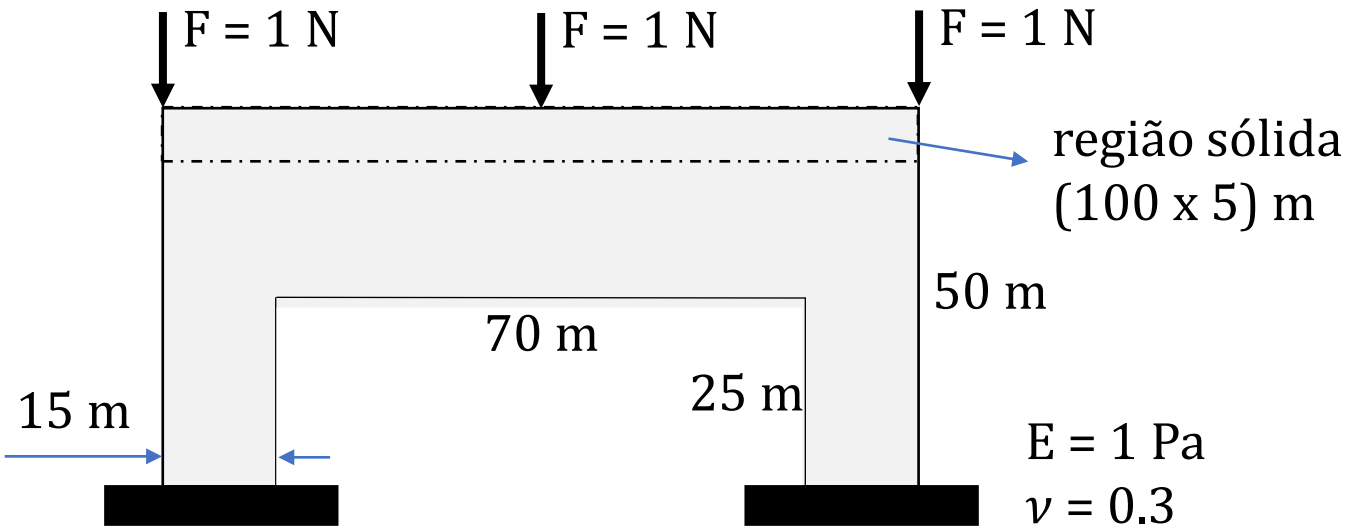
```
passive(i) = 1;
```

  - fim

```
passive = find(passive == 1);
```

PNV USP

Exercício 4



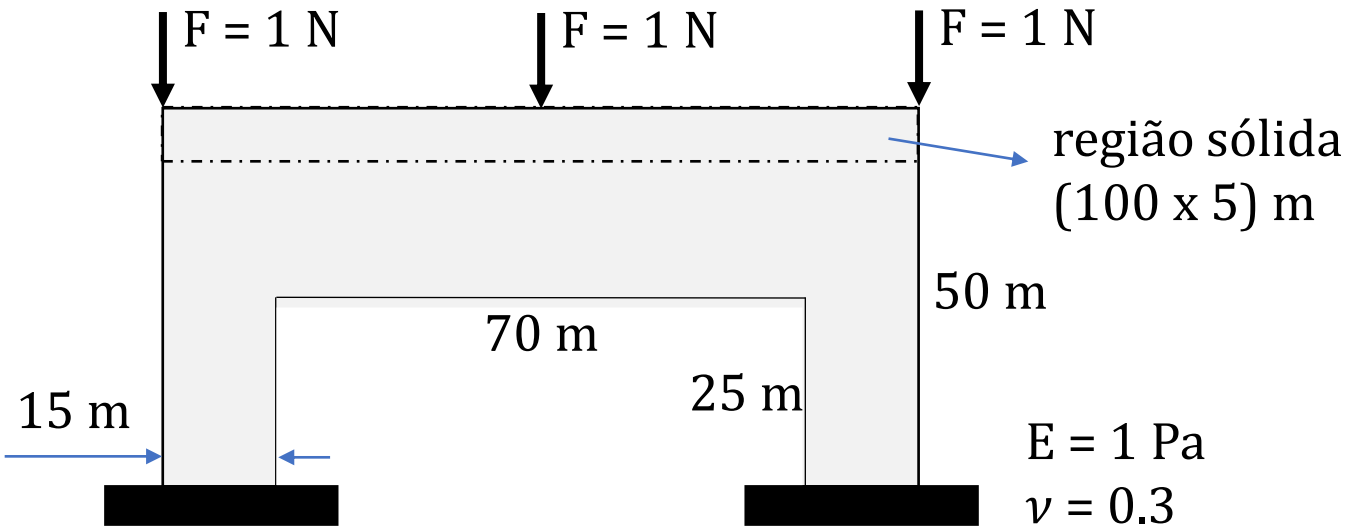
$$\text{minimizar} \quad V(\mathbf{x})$$
$$\text{sujeito a} \quad \mathbf{F}^T\mathbf{U} \leq \bar{C}$$
$$x_i = \{0,1\}$$

Caso de carregamento simples.

```
% Optimization parameters
radius = 3;                     % Filter radius in length unit
rho_min = 0.001^3;              % Minimum density (for void elements)
compliance_constraint = 50;     % Compliance constraint
epsilons = 0.01;                % Constraint relaxation parameter
flip_limits = 0.10;             % Flip limits
tau = 0.001;                    % Convergence tolerance
```

PNV USP

Exercício 4

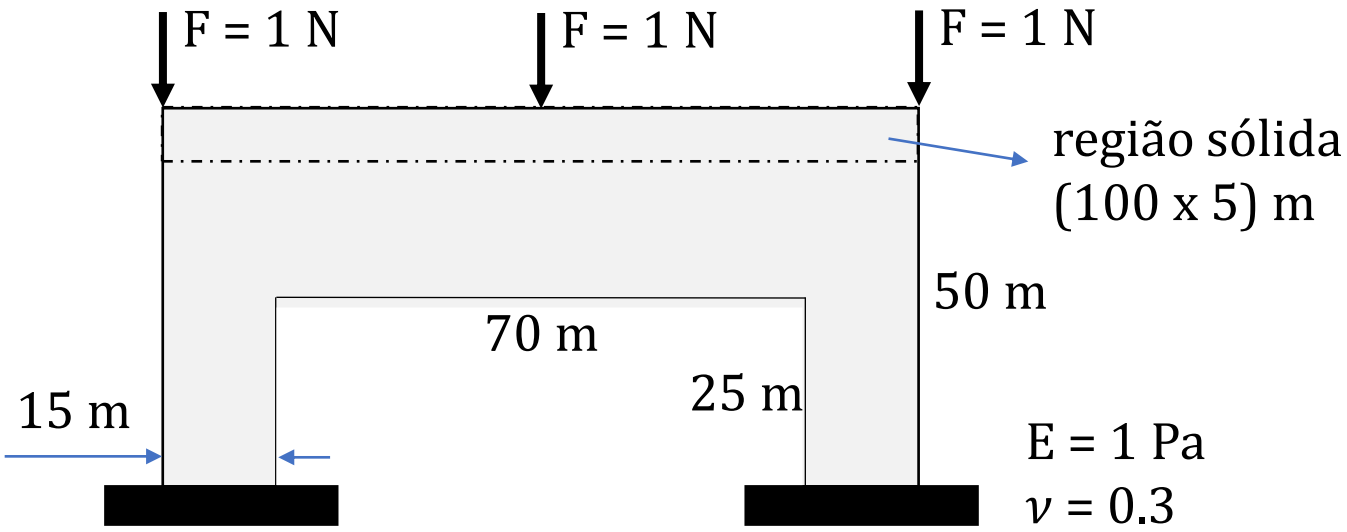


$$\begin{aligned} \text{minimizar} \quad & V(\mathbf{x}) \\ \text{sujeito a} \quad & \mathbf{F}^T\mathbf{U} \leq \bar{C} \\ & x_i = \{0,1\} \end{aligned}$$

Incluir a lista de elementos que se manterão sólido. A diferença para o exercício anterior é que a lista de elementos passivos mudou (aumentou).

PNV USP

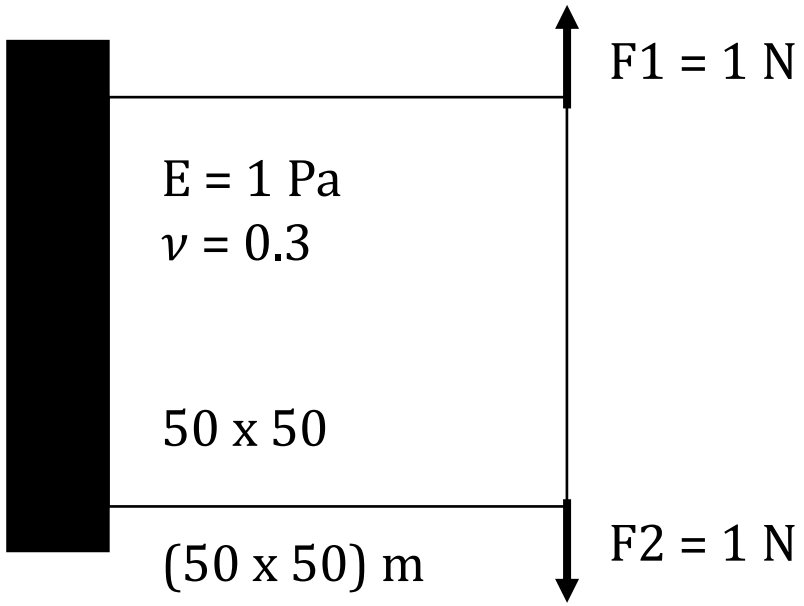
Exercício 4



Incluir a lista de elementos que se manterão sólido. A diferença para o exercício anterior é que a lista de elementos passivos mudou (aumentou).

PNV USP

Outros casos – carregamentos múltiplos



O problema de carregamento simples é quando F_1 e F_2 agem simultaneamente.

O problema de carregamentos múltiplos é quando F_1 e F_2 podem agir em diferentes momentos. Assim, a estrutura deve ser projetada para suportar ambos os carregamentos separadamente. Nesse caso,

$f(\mathbf{x}) = w1 * f_1(\mathbf{x}) + w2 * f_2(\mathbf{x})$; consequentemente,

$$\frac{df}{dx_i} = w1 * \frac{df_1}{dx_i} + w2 * \frac{df_2}{dx_i}.$$

Outros casos – restrição de deslocamento

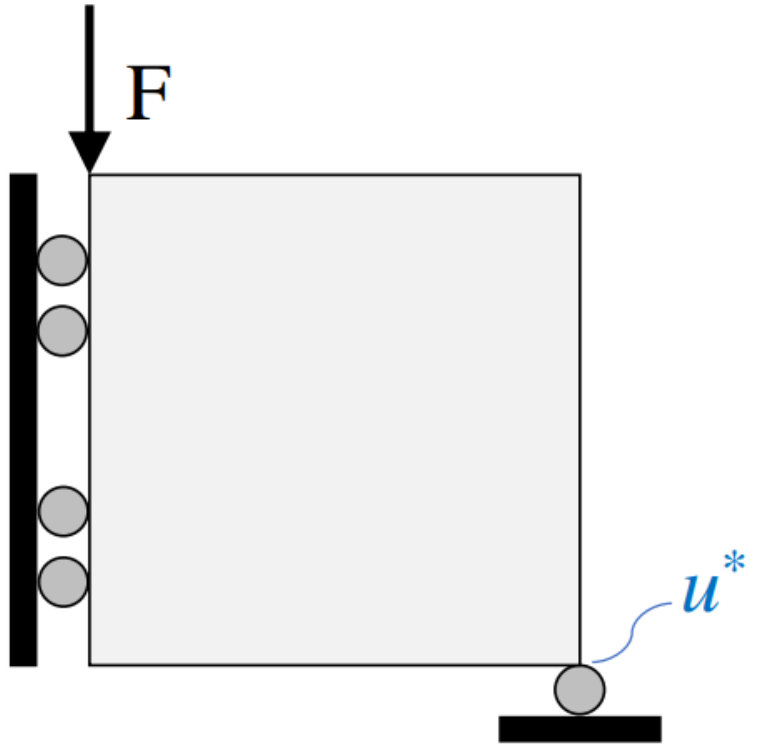
$$\underset{\boldsymbol{x}}{\text{Minimize}} \quad C(\boldsymbol{x}) = \boldsymbol{u}^T \boldsymbol{K}\boldsymbol{u},$$

$$\text{Subject to} \quad \frac{V(\boldsymbol{x})}{V_0} \leq \overline{V},$$

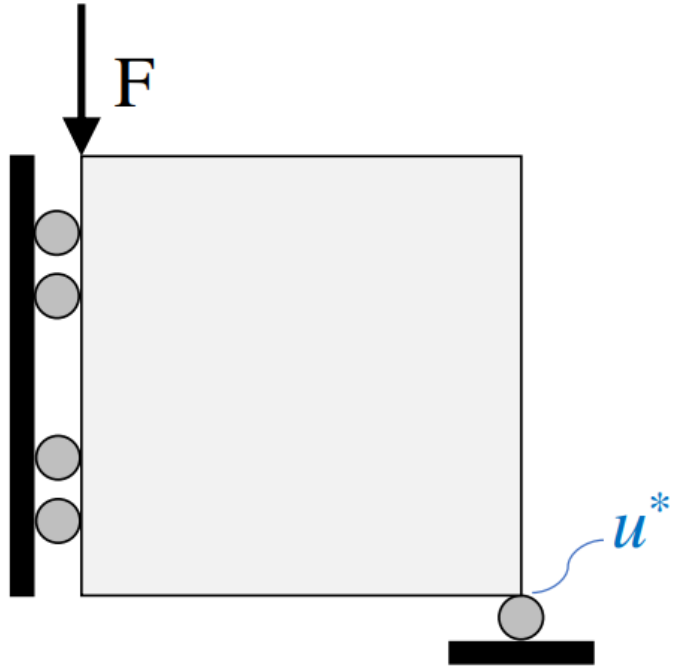
$$u^* \leq \bar{u},$$

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F},$$

$$x_j \in \{0, 1\}, \quad j \in [1, N_d]$$



Outros casos – restrição de deslocamento



Problema adjunto:

$$\mathbf{K}\boldsymbol{\lambda} = -\frac{\partial \emptyset}{\partial \mathbf{U}} \longrightarrow \boldsymbol{K}\boldsymbol{\lambda} = -\boldsymbol{P}$$

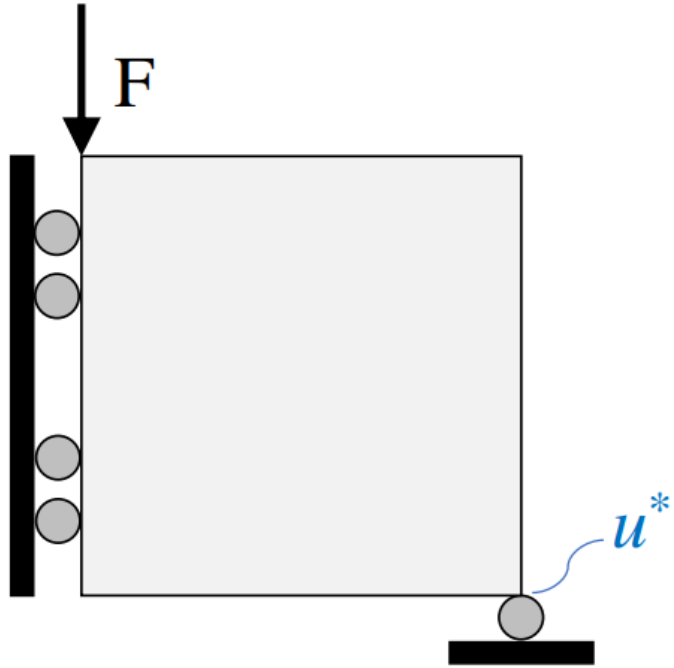
$$\emptyset(x_i) = u^*$$

Expressão final:

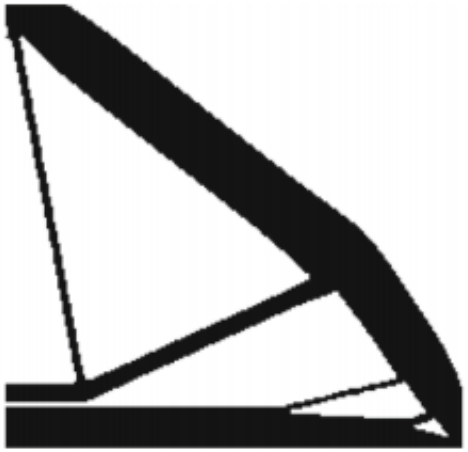
$$\frac{d\emptyset}{dx_i} = \frac{\partial \emptyset}{\partial x_i} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{K}}{\partial x_i}\mathbf{U} \longrightarrow \frac{\partial u^*}{\partial x_j} = p x_j^{p-1}(E_0 - E_{\min})\boldsymbol{\lambda}_j^T \boldsymbol{k}_0 \boldsymbol{u}_j$$

PNV USP

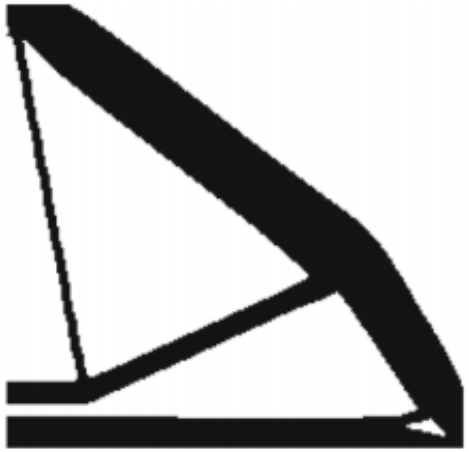
Outros casos – restrição de deslocamento



(a) $u^* \leq 10$



(b) $u^* \leq 12$



(c) $u^* \leq 14$

(d) $u^* \leq 100$

Outros casos – restrição de deslocamento