

P.G.O



aula 11 / Otimização topológica

190924



Sumário (TP+P)

Otimização Topológica

Definição e classificação
Overview

Método SIMP
Fluxograma detalhado

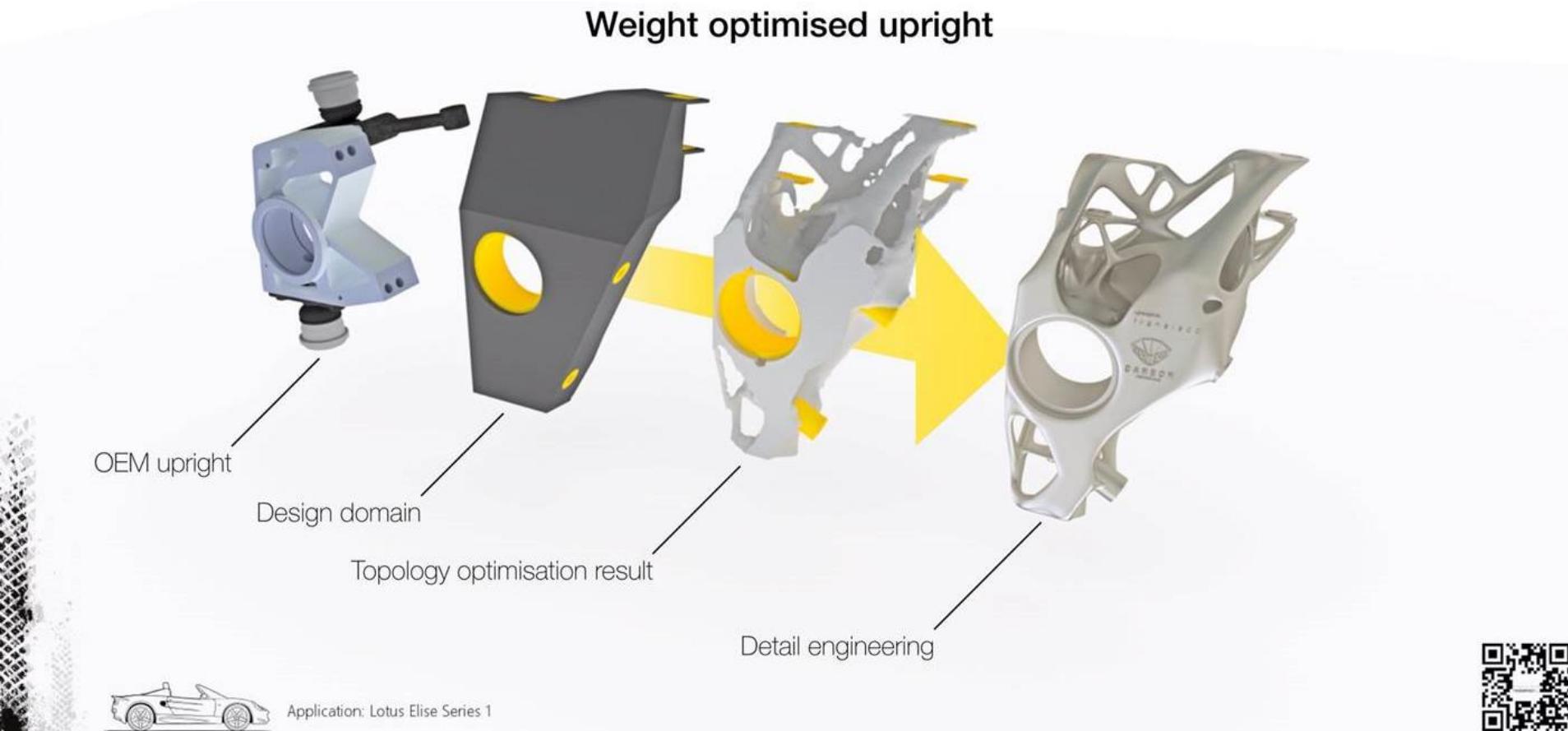
Labwork
Análise de um código TO SIMP



<https://forms.office.com/e/fR29HVDxs2>



Otimização Topológica





Otimização Topológica

Primeiros artigos científicos ~1990

Em 1994, a Altair recebeu o prémio “Technology of the Year” da IndustryWeek:

> OptiStruct, primeiro Código commercial de optimização topológica

Neste momento, em crescimento acentuado, com integração gradual na maioria dos códigos comerciais

- . Evolução do poder computacional
- . Plataformas integradas CAD/CAE
- . Design generativo
- . Fabrico aditivo (e BAAM)

DfAM e oportunidades

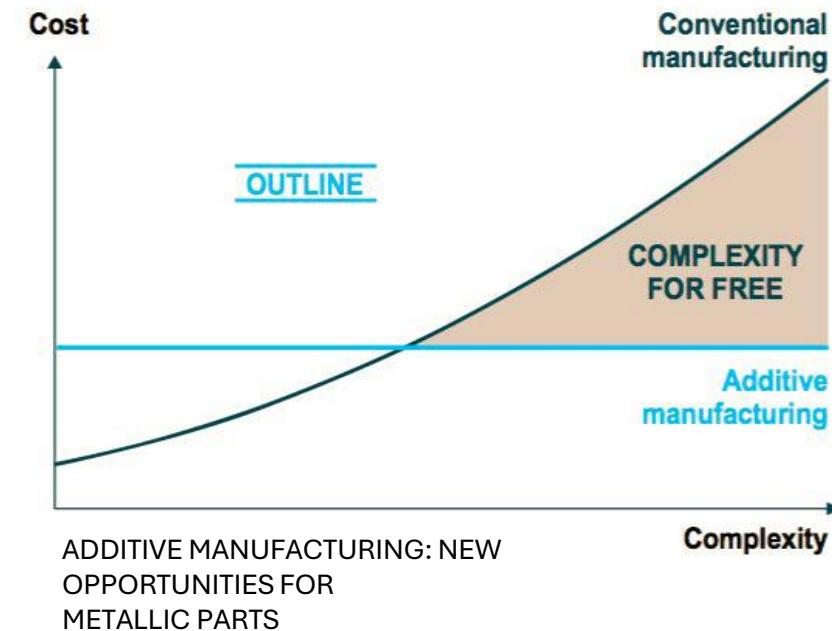
Design for Additive Manufacturing

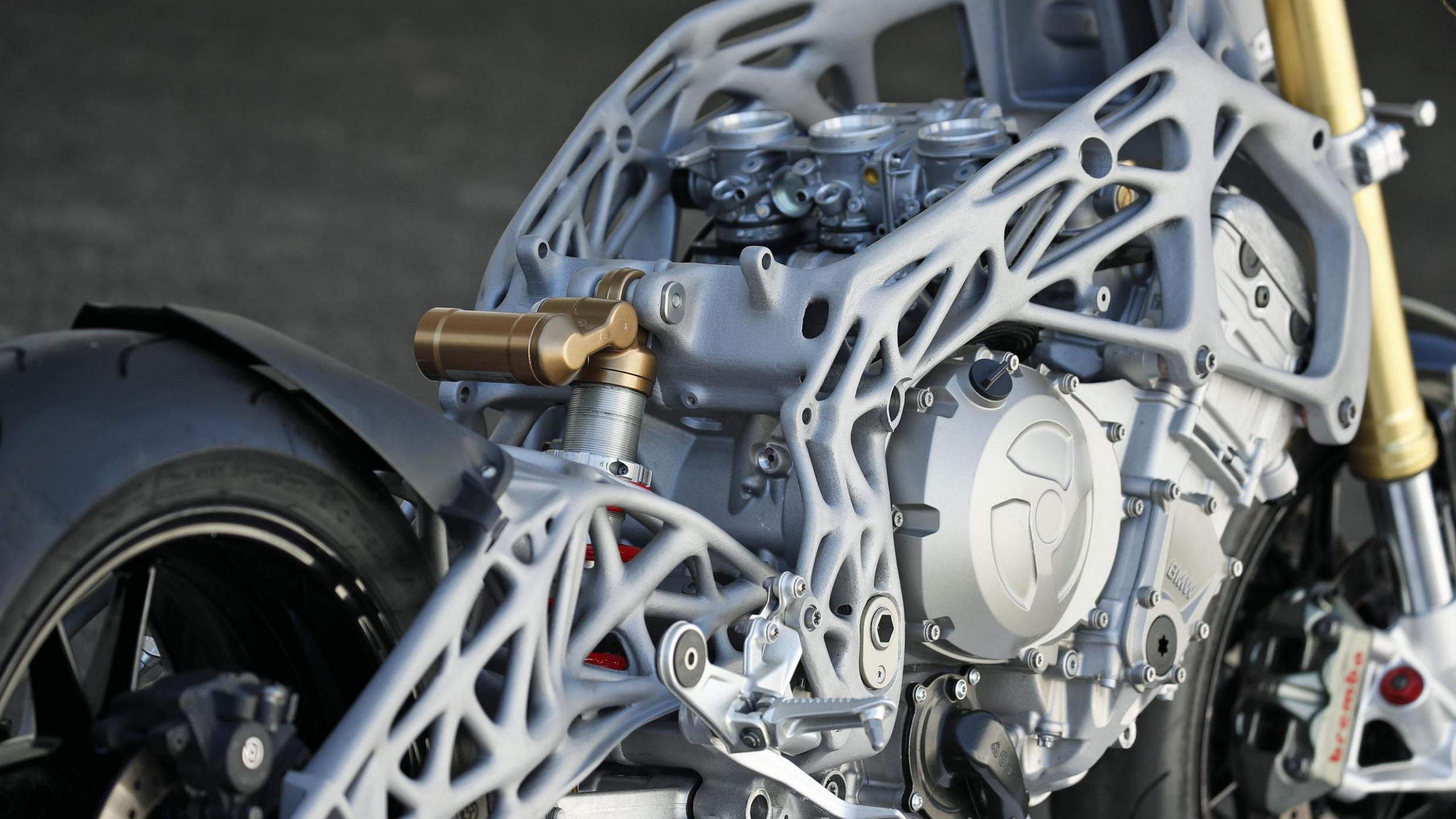
- > Optimização topológica
- > Projecto multiescala
- > Integração multi-material
- > Design for mass customization
- > Consolidação de componentes



DfAM e oportunidades

mudança de abordagem...







Otimização Topológica

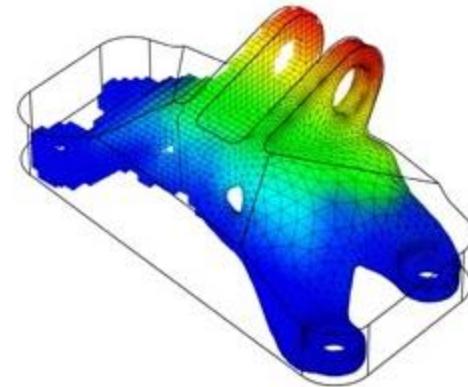




Otimização Topológica

GE jet engine bracket challenge

<https://grabcad.com/challenges/ge-jet-engine-bracket-challenge>





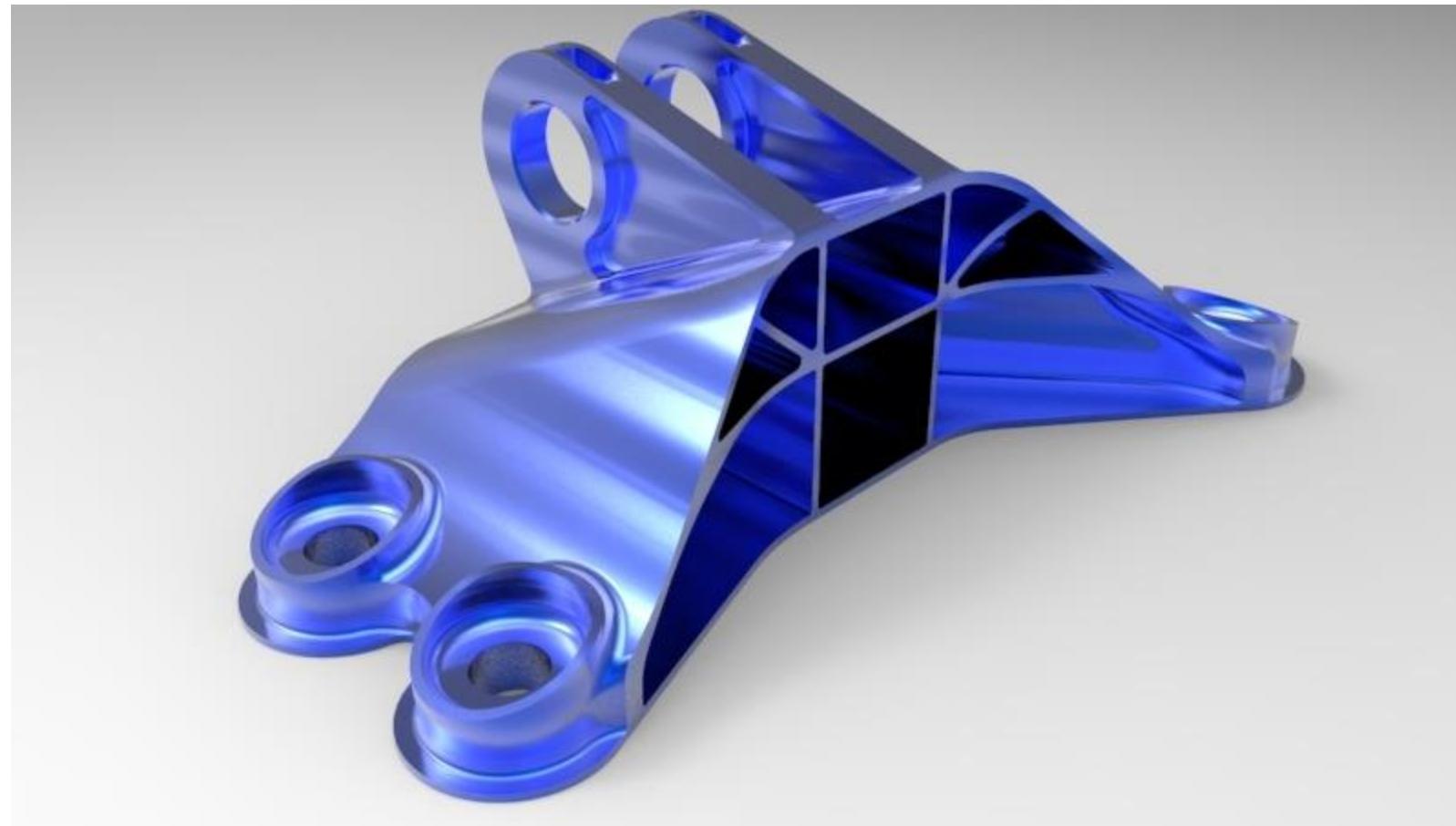
Otimização Topológica

GE jet engine bracket challenge
optimum?





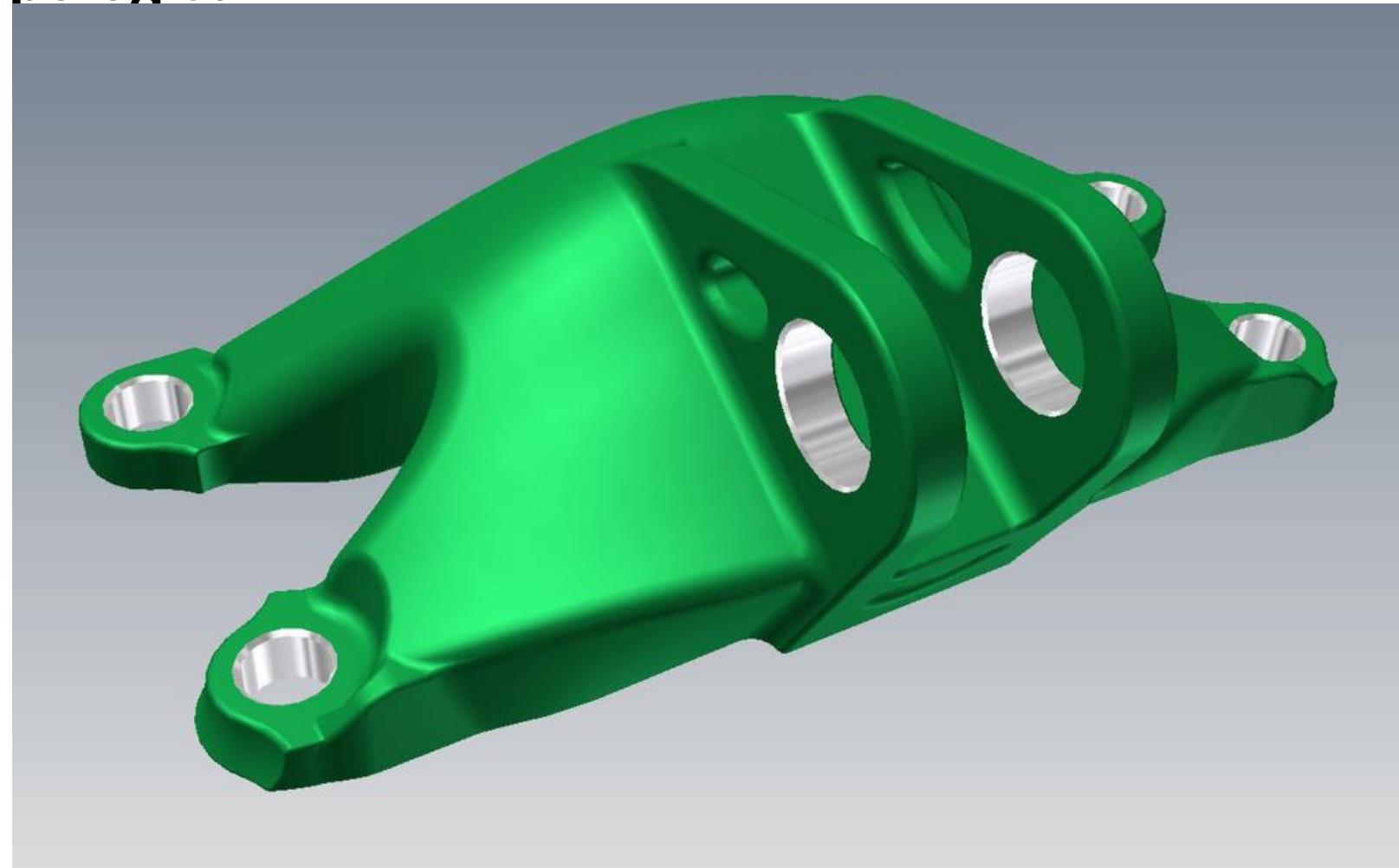
Otimização Topológica



1st place



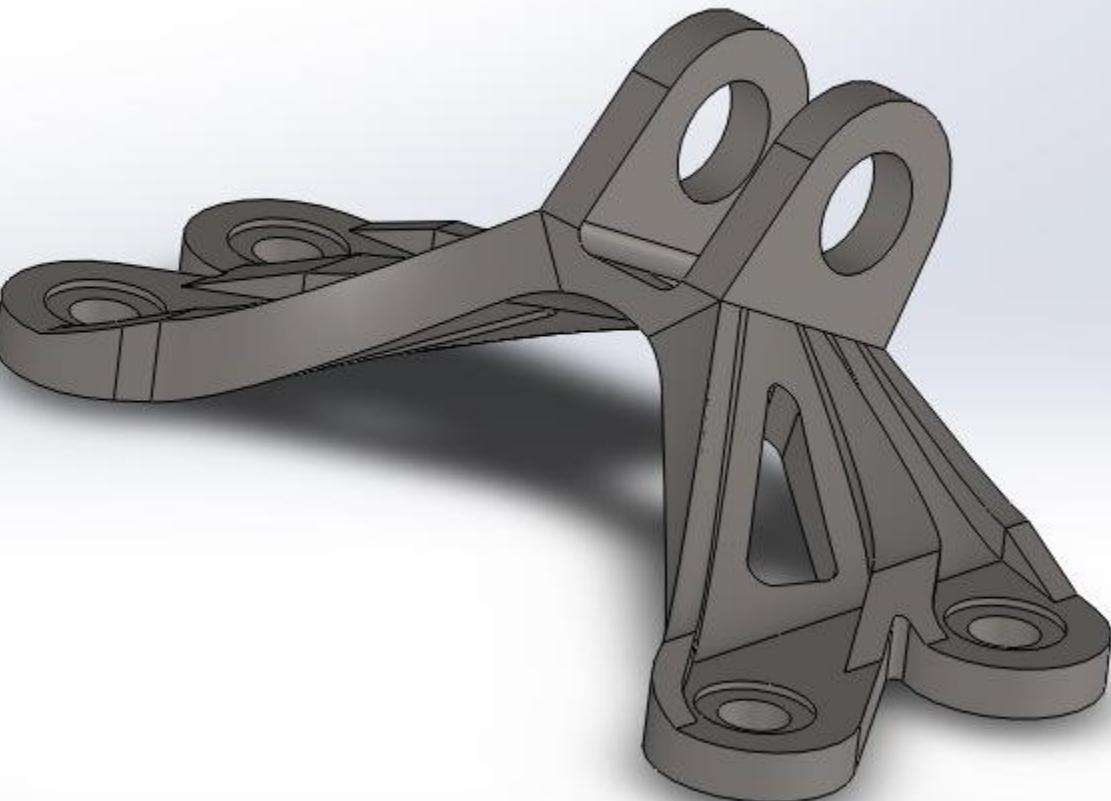
Otimização topológica



2nd place



Otimização Topológica



6th place



Design generativo

- sejam críticos!



dreamcatcher AUTODESK



Design generativo

Arup 3 starting shape

Define Explore

Outcome filters

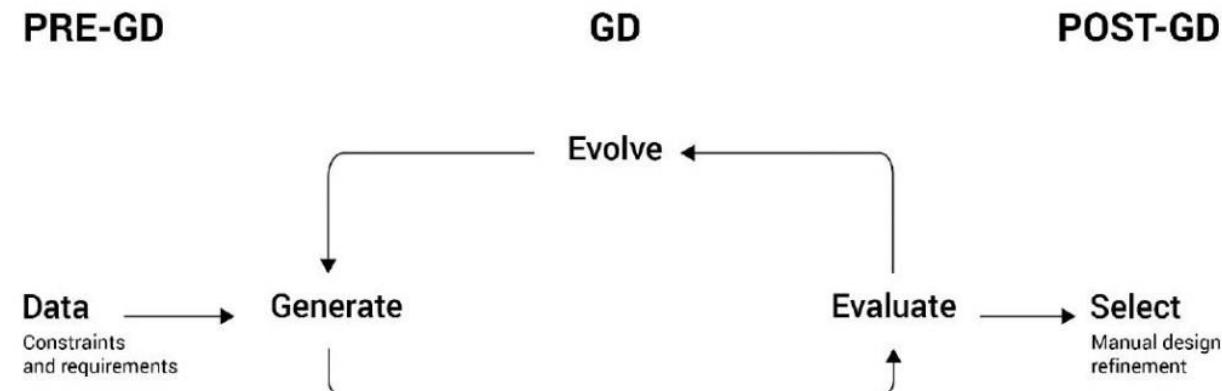
- Processing status
 - View completed
 - View processing
 - View queued
 - View failed
 - View canceled
- Study
 - FOS 3
 - FOS 7 for start
 - FOS 15 for start
 - Reverse Loads
 - Starting Shape ...
 - Starting SShape ...
 - Starting Shape R...
 - Study 1
- Objective ranges
 - Volume (mm³)
52,060.14 165,348.94
 - Mass (g)
156.91 1,102.66
 - Maximum displacement (mm)
0.00 0.93
 - Maximum von Mises stress (MPa)
16.5 2,097.6
 - Minimum factor of safety
0.48 18.00

Arup 3 starting shape 24 outcomes 24 completed

Sort by Maximum von Mises stress

Generative design is the use of algorithmic methods to generate feasible designs or outcomes from a set of performance objectives, performance constraints, and design space for specified use cases.

Performance objectives and constraints may include factors from multiple areas including operational performance, weight/mass, manufacturing, assembly or construction, usability, aesthetics, ergonomics, and cost.



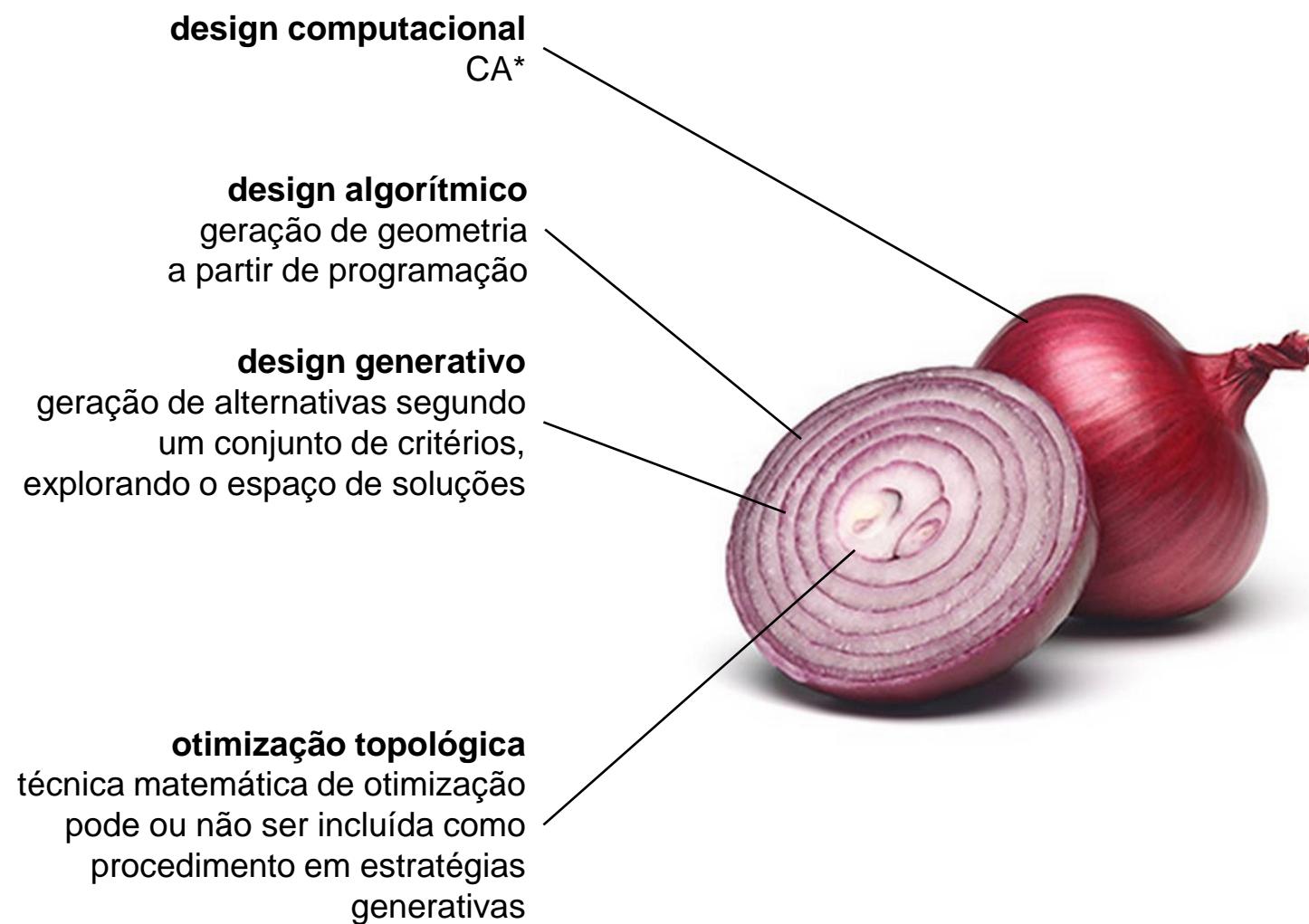
Generative Design is Doomed to Fail

Daniel Davis – 20 February 2020

<https://www.danieldavis.com/generative-design-doomed-to-fail/>



Design generativo



design computacional

CA*

design algorítmico

geração de geometria
a partir de programação

design generativo

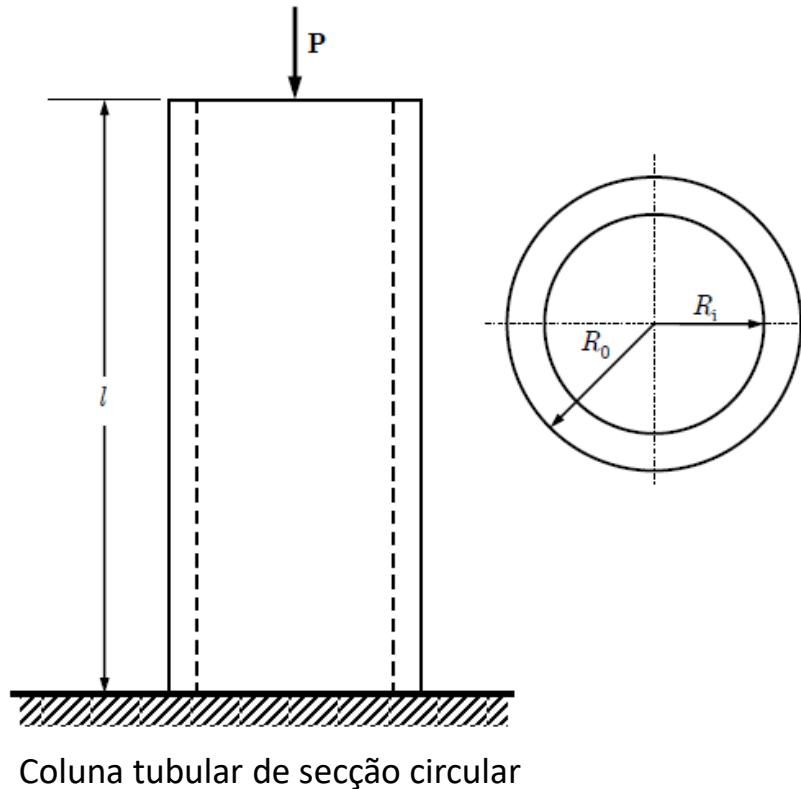
geração de alternativas segundo
um conjunto de critérios,
explorando o espaço de soluções

otimização topológica

técnica matemática de otimização
pode ou não ser incluída como
procedimento em estratégias
generativas

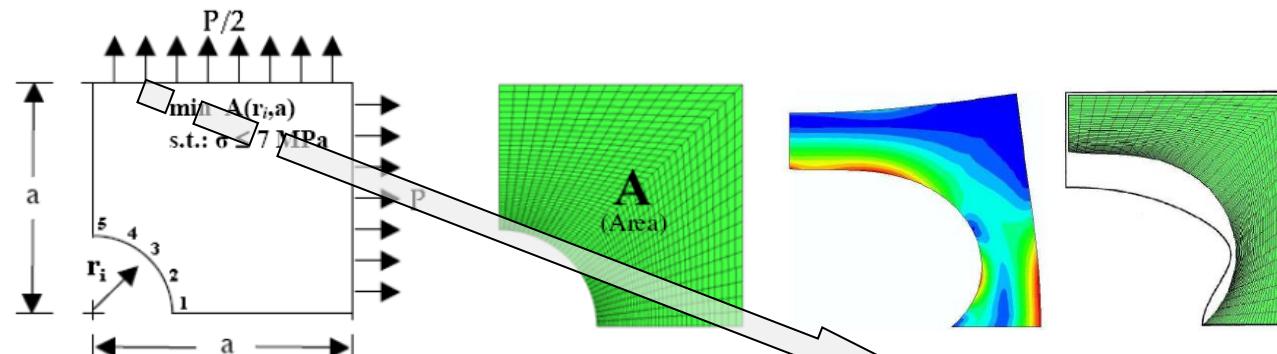
- O que é então otimização topológica?

Otimização topológica: o que não é

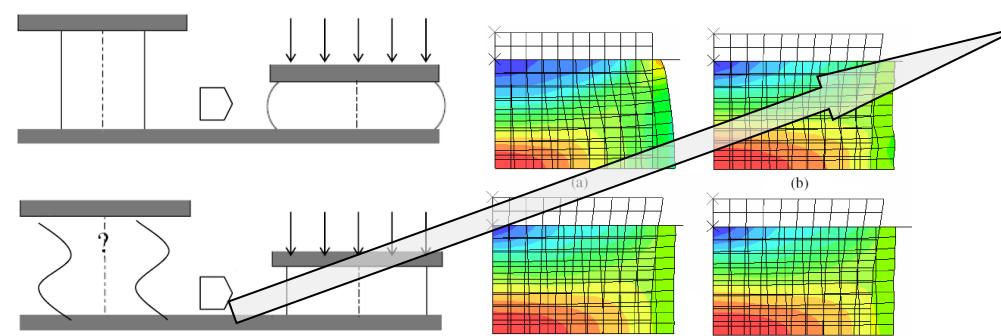


É possível parametrizar uma geometria definida com alteração apenas de dimensões ou parâmetros característicos?
→ **optimização dimensional**

Otimização topológica: o que não é



Problema de placa quadrada com furo central

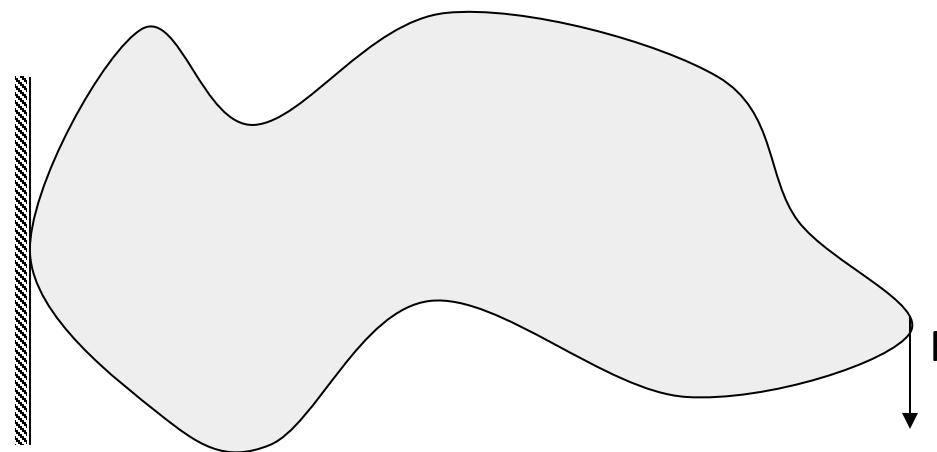


Problema de forjamento. Determinação da geometria inicial para obter uma dada geometria final

É possível parametrizar a geometria (forma) com um conjunto relativamente reduzido de variáveis?

→ optimização de forma

Otimização topológica

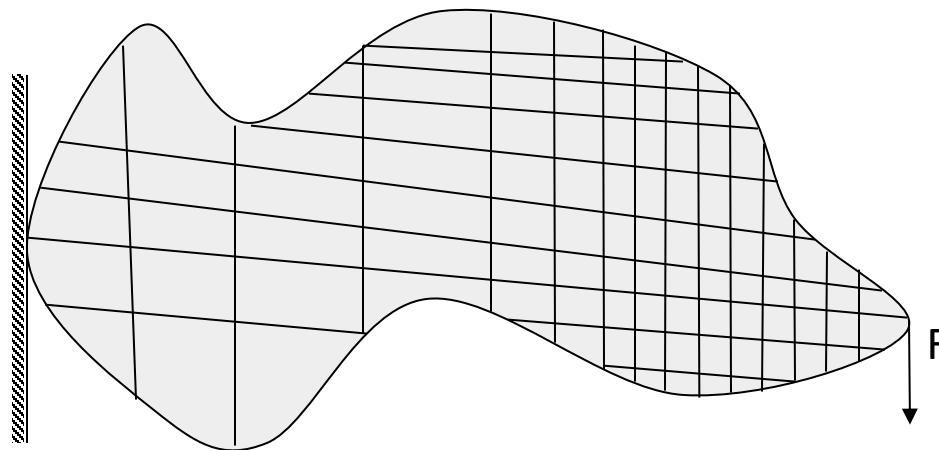


Qual a geometria que maximiza a rigidez do sólido?

(considerando eventuais furos, minimizando material, etc.)

É (quase) impossível parametrizar esta geometria (forma) com um número reduzido de variáveis

Otimização topológica: o que é

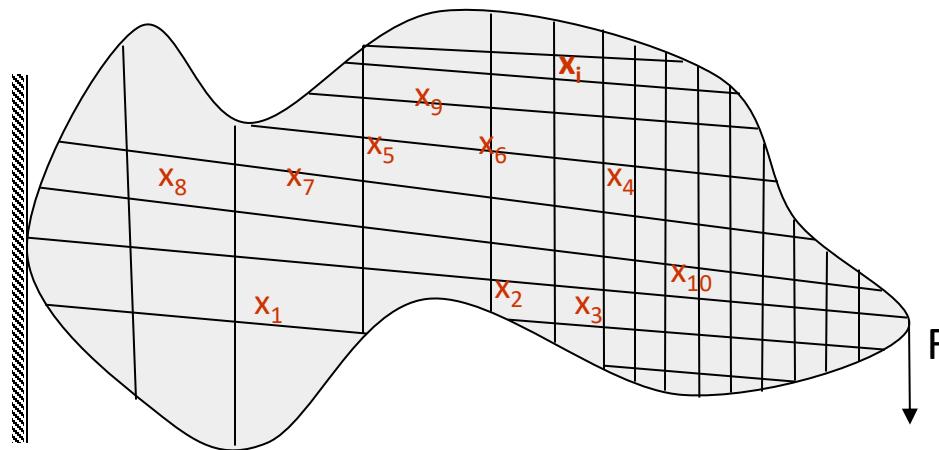


Qual a geometria que maximiza a rigidez do sólido?

(considerando eventuais furos, minimizando material, etc.)

Porque não discretizar este contínuo num número razoável de pequenos volumes?

Otimização topológica: o que é



Qual a geometria que maximiza a rigidez do sólido?

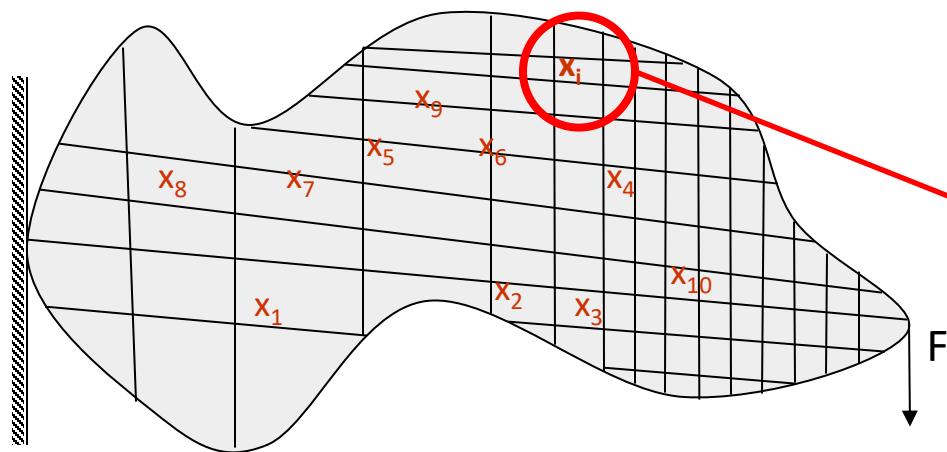
(considerando eventuais furos, minimizando material, etc.)

ground structure

Porque não discretizar este contínuo num número razoável de pequenos volumes?

.. e atribuir a cada volume uma variável de controlo que characterize a utilidade dessa porção do domínio?

Otimização topológica: o que é



Com um processo de optimização, é possível avaliar a importância de cada parte do domínio

| $x_i=1$, material;
| $x_i=0$, vazio.

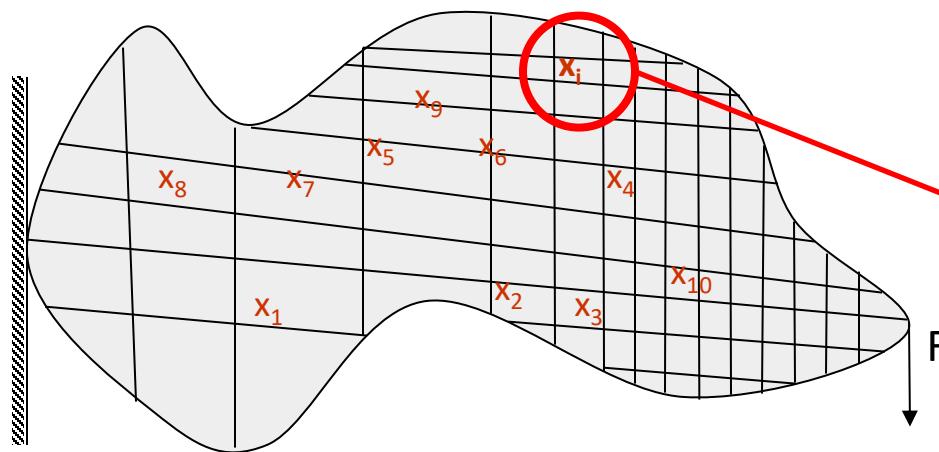
Qual a geometria que maximiza a rigidez do sólido?

(considerando eventuais furos, minimizando material, etc.)



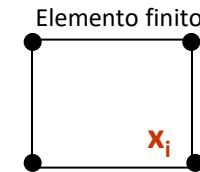
Semelhanças com algum método?

Otimização topológica: o que é



Qual a geometria que maximiza a rigidez do sólido?

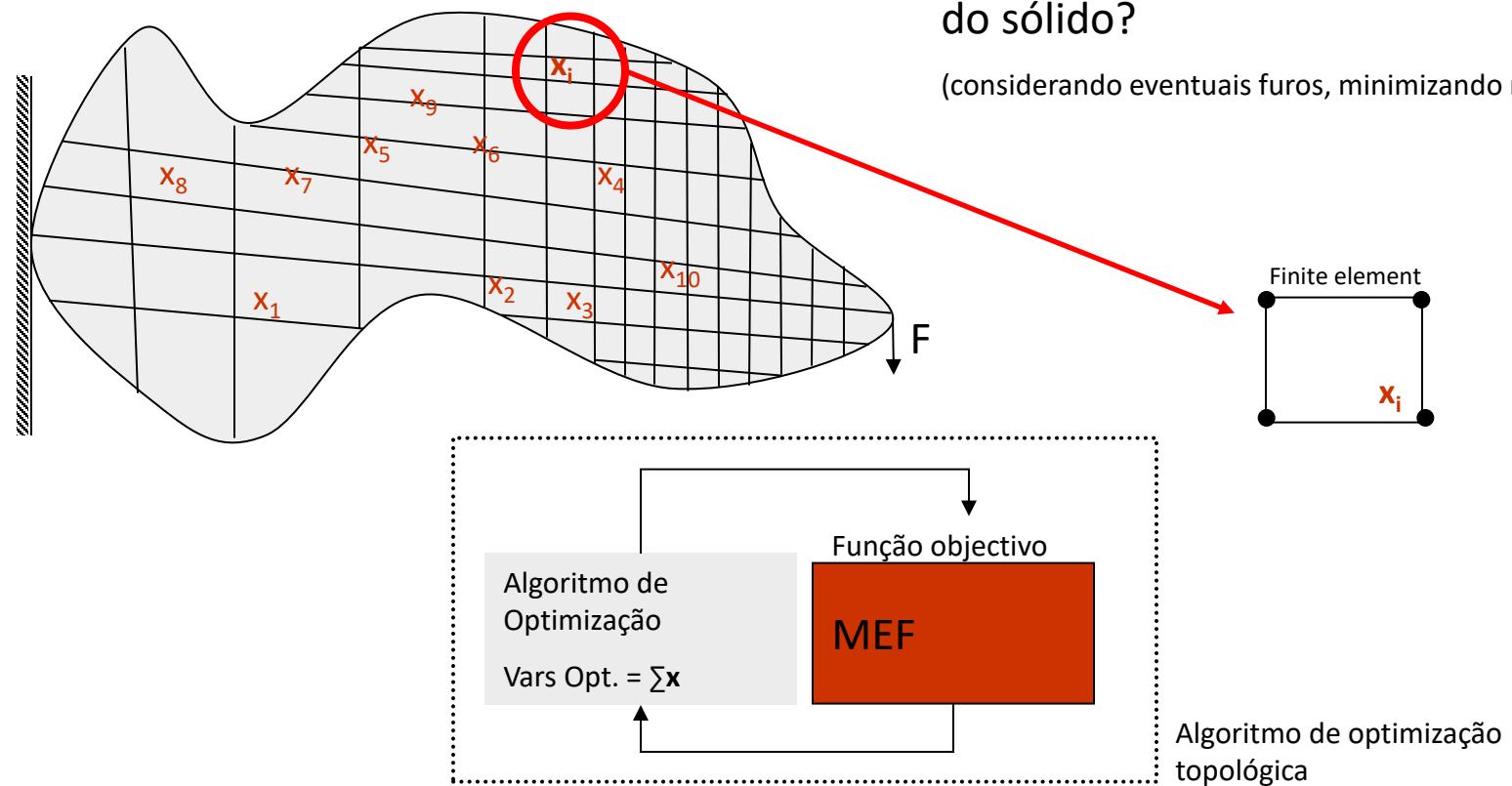
(considerando eventuais furos, minimizando material, etc.)



O Método dos Elementos Finitos ajusta-se bem a este problema

O MEF pode ser usado para avaliar a função objectivo (deformação, tensão, energia de deformação, conformidade, etc...)

Otimização topológica: o que é





Otimização topológica: definição

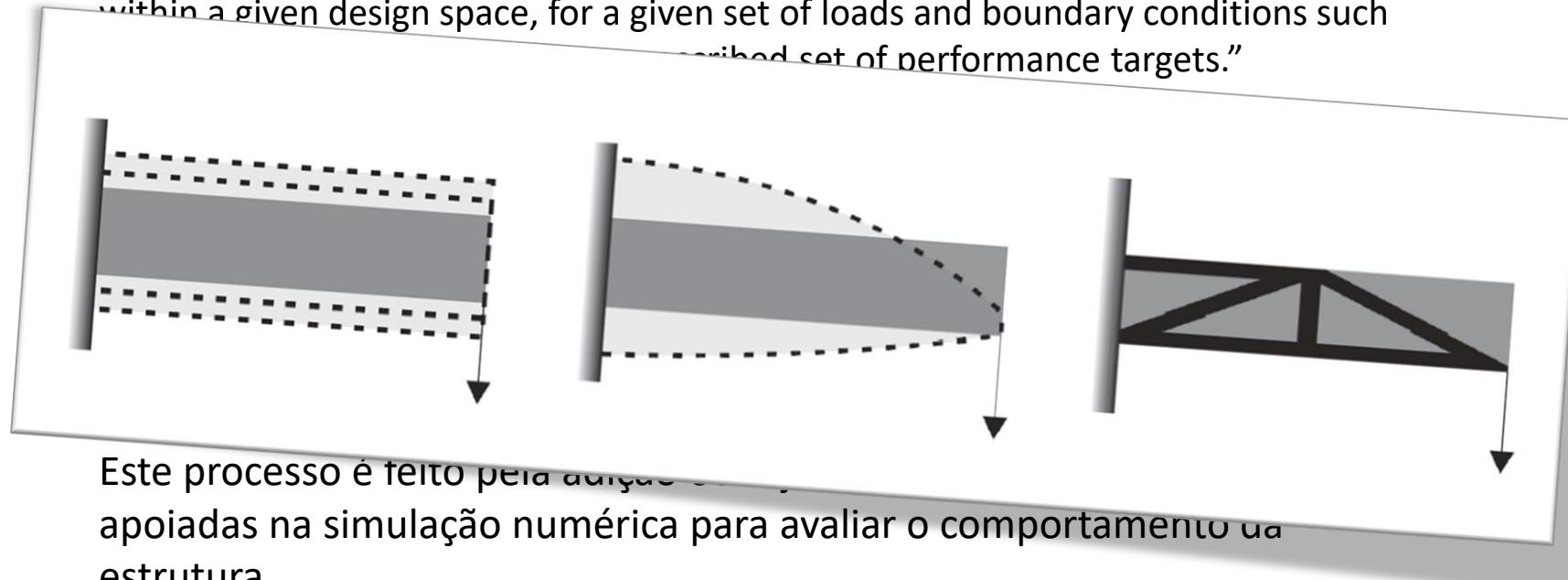
“**Topology optimization** is a mathematical approach that optimizes material layout within a given design space, for a given set of loads and boundary conditions such that the resulting layout meets a prescribed set of performance targets.”

i.e. a optimização topológica pode gerar a topologia óptima (forma geral) de uma estrutura sujeita a carregamentos mecânicos

Este processo é feito pela adição ou rejeição de material, com decisões apoiadas na simulação numérica para avaliar o comportamento da estrutura

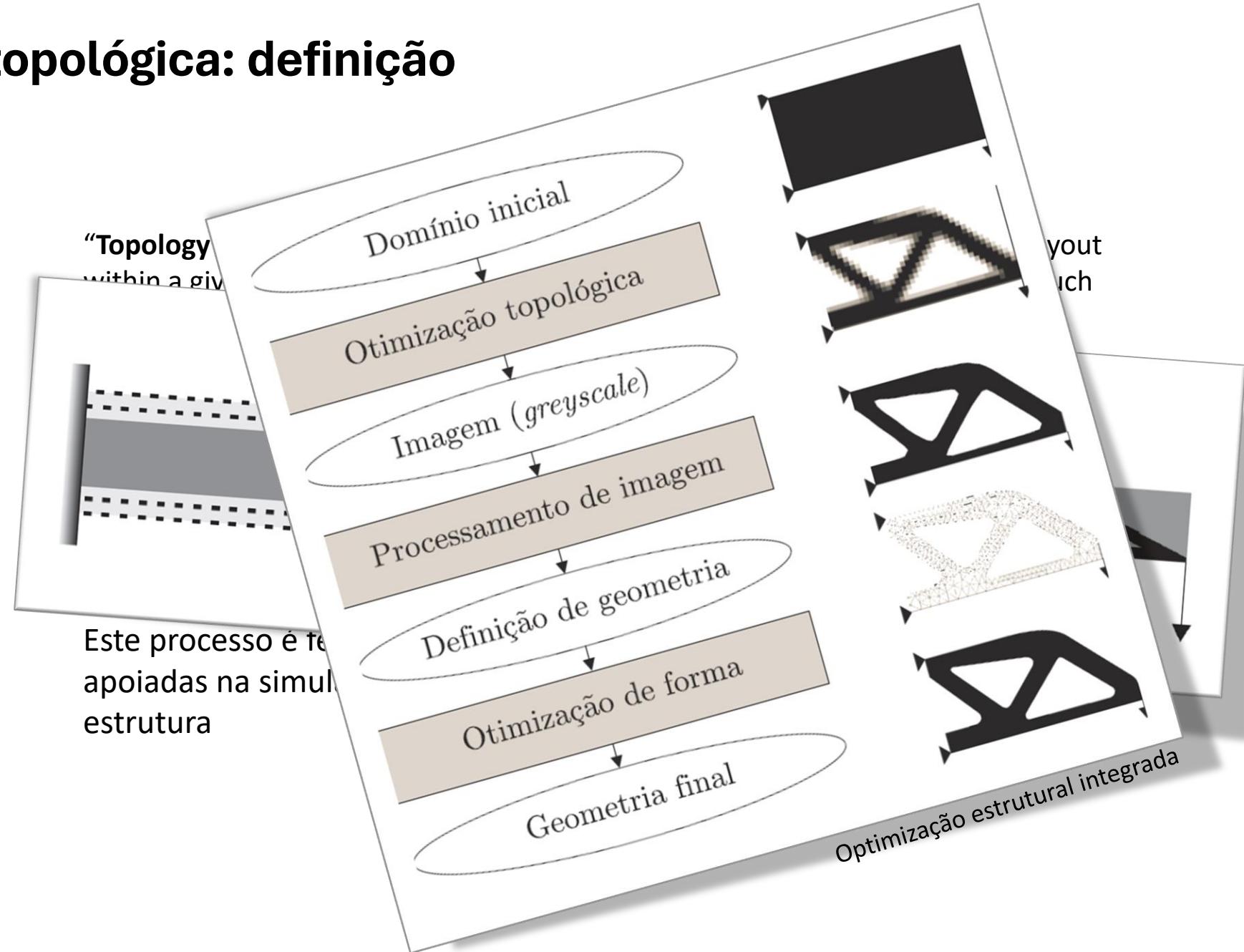
Otimização topológica: definição

“**Topology optimization** is a mathematical approach that optimizes material layout within a given design space, for a given set of loads and boundary conditions such as described set of performance targets.”



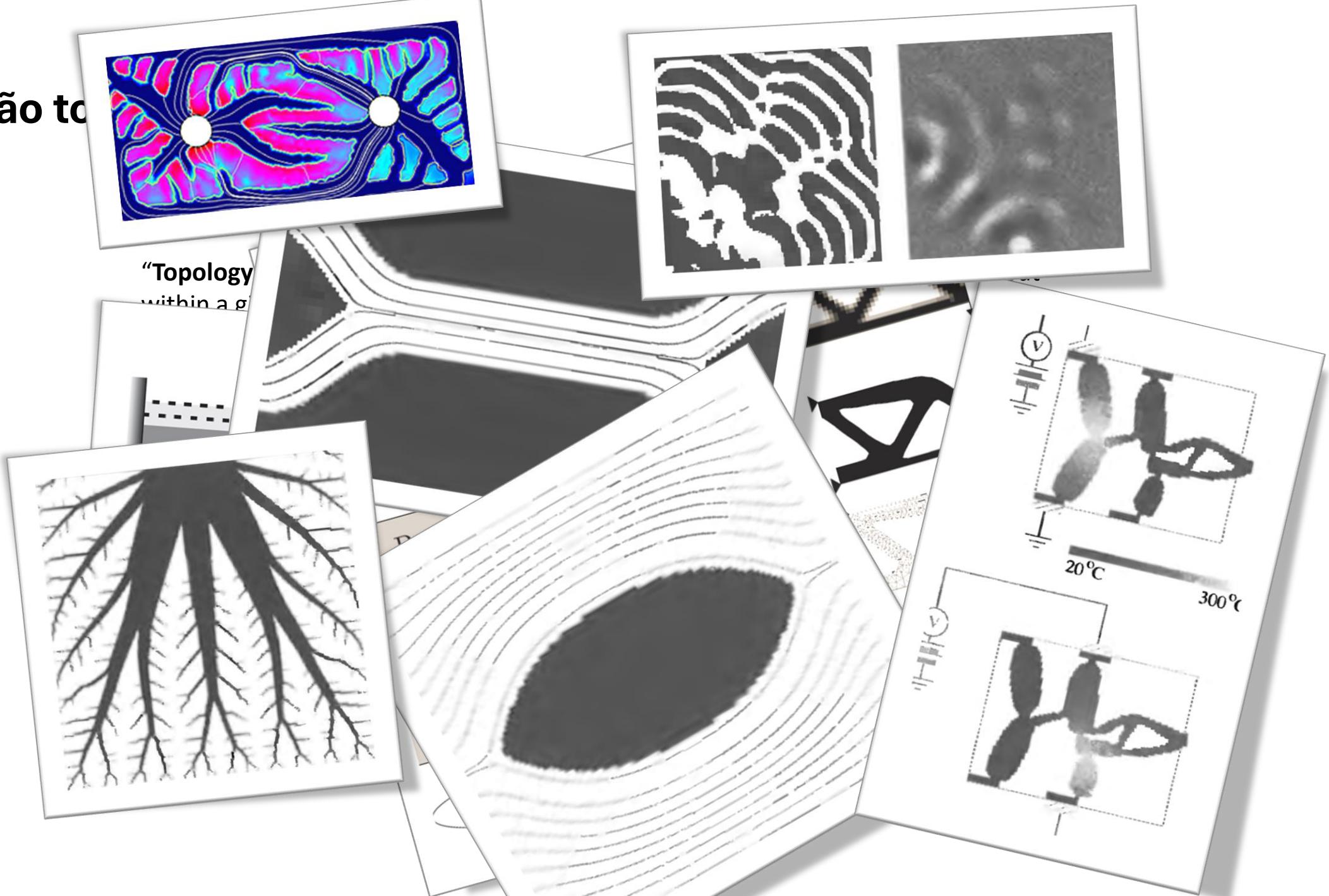
Este processo é feito pela análise de critérios de otimização apoiadas na simulação numérica para avaliar o comportamento da estrutura

Otimização topológica: definição

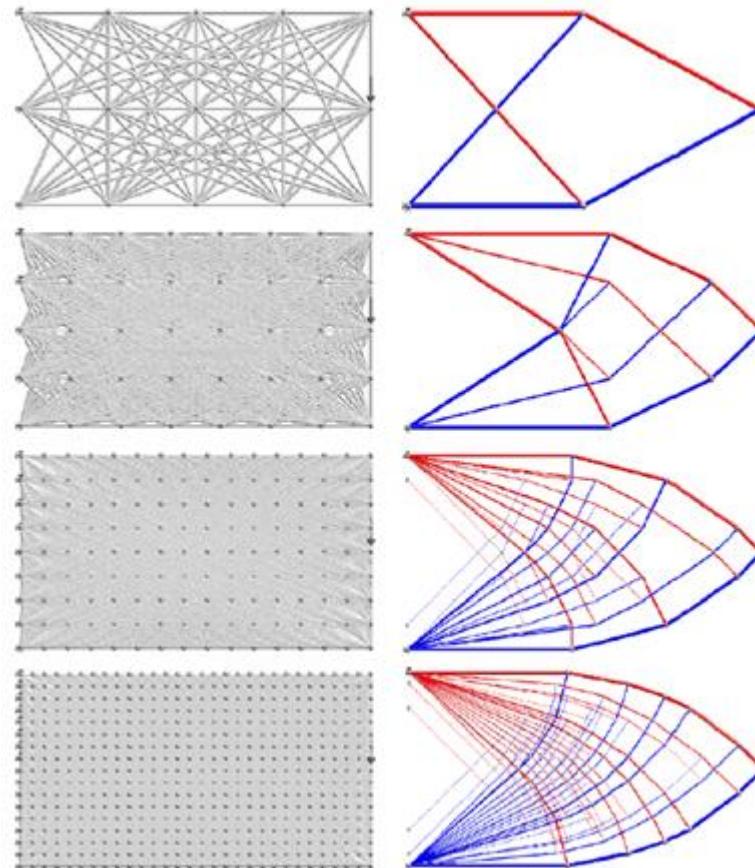




Otimização topológica

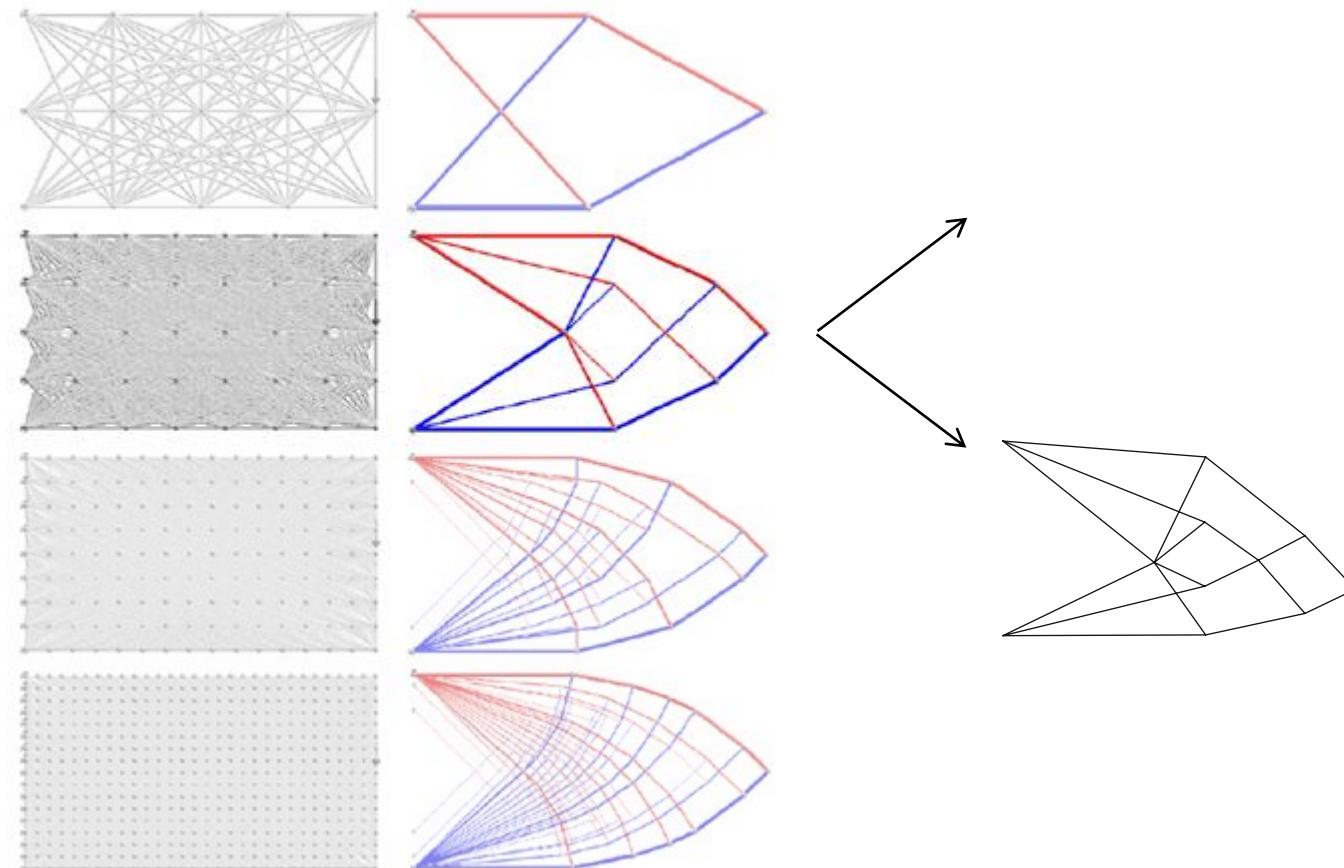


Otimização topológica: ainda é?



P. Martínez, P. Martí, O.M. Querin, Growth method for size,topology, and geometry optimization of truss structures, *Structural and Multidisciplinary Optimization* **33**(1):13-26, 2007

Otimização topológica: o que é

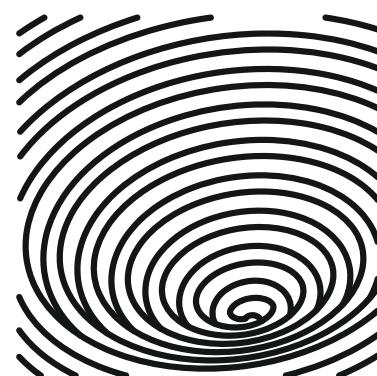


P. Martínez, P. Martí, O.M. Querin, Growth method for size,topology, and geometry optimization of truss structures, *Structural and Multidisciplinary Optimization* **33**(1):13-26, 2007



Otimização em Cálculo Estrutural

Otimização topológica



Elasticidade linear

Elasticidade linear

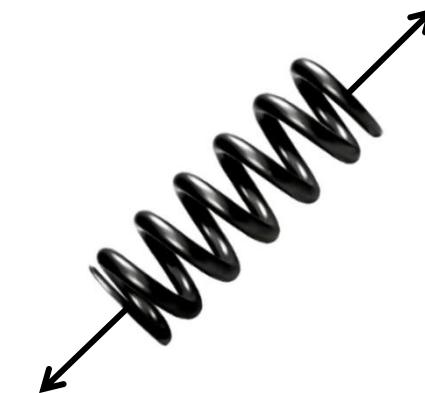


Trabalho executado pela força externa (compliance)

$$W = \int_{\Gamma} \mathbf{f}^T \mathbf{u} d\Gamma$$

Energia de deformação do sólido

$$S = \frac{1}{2} \int_{\Omega} \nabla \mathbf{u}^T \mathbf{D} \nabla \mathbf{u} d\Omega$$



Trabalho executado pela carga aplicada

$$W = f u, \quad u = \Delta L$$

Energia elástica da mola

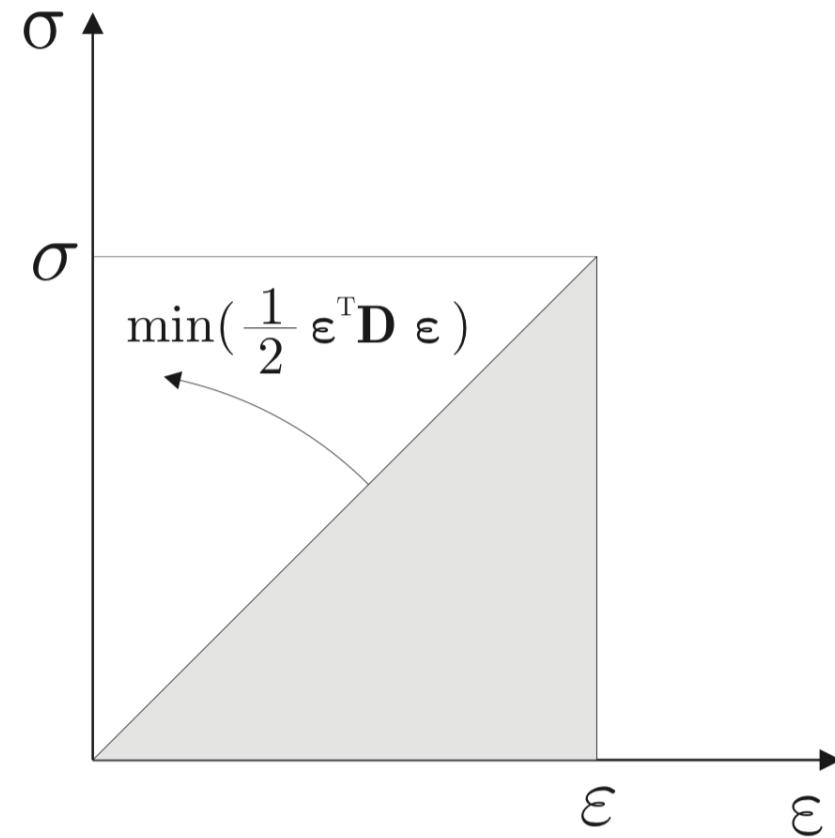
$$S = \frac{1}{2} k u^2$$

No equilíbrio ($\mathbf{k}\mathbf{u}=\mathbf{f}$), o campo de deslocamentos minimiza a energia potencial total

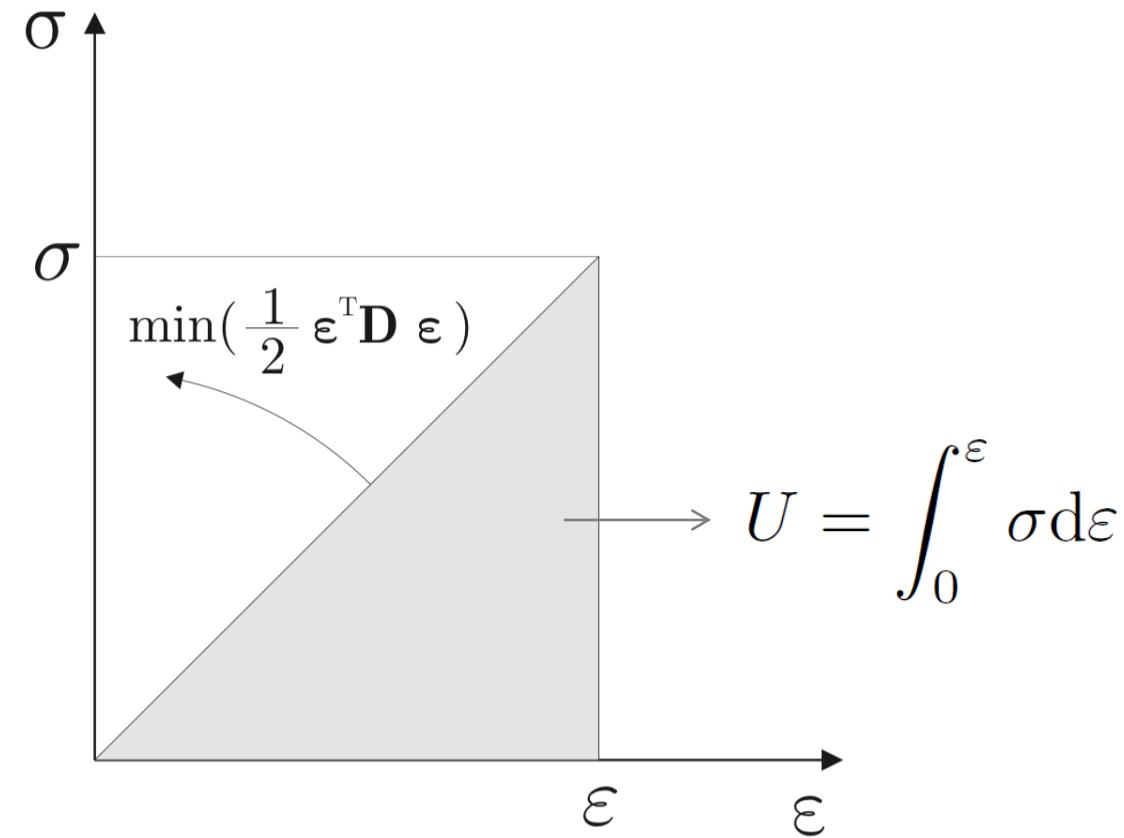
$$P = S - W = \frac{1}{2} \int_{\Omega} \nabla \mathbf{u}^T \mathbf{D} \nabla \mathbf{u} d\Omega - \int_{\Gamma} \mathbf{f}^T \mathbf{u} d\Gamma$$

[Berggen 06]

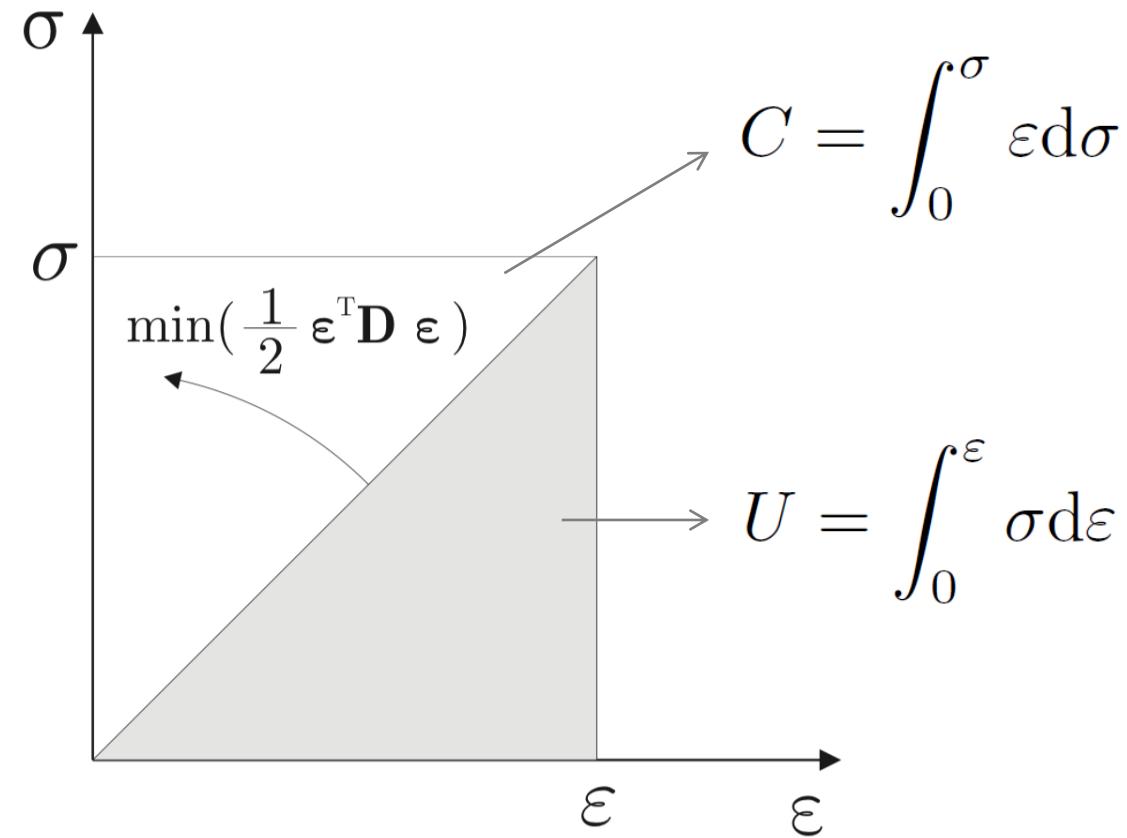
Elasticidade linear e energia de deformação



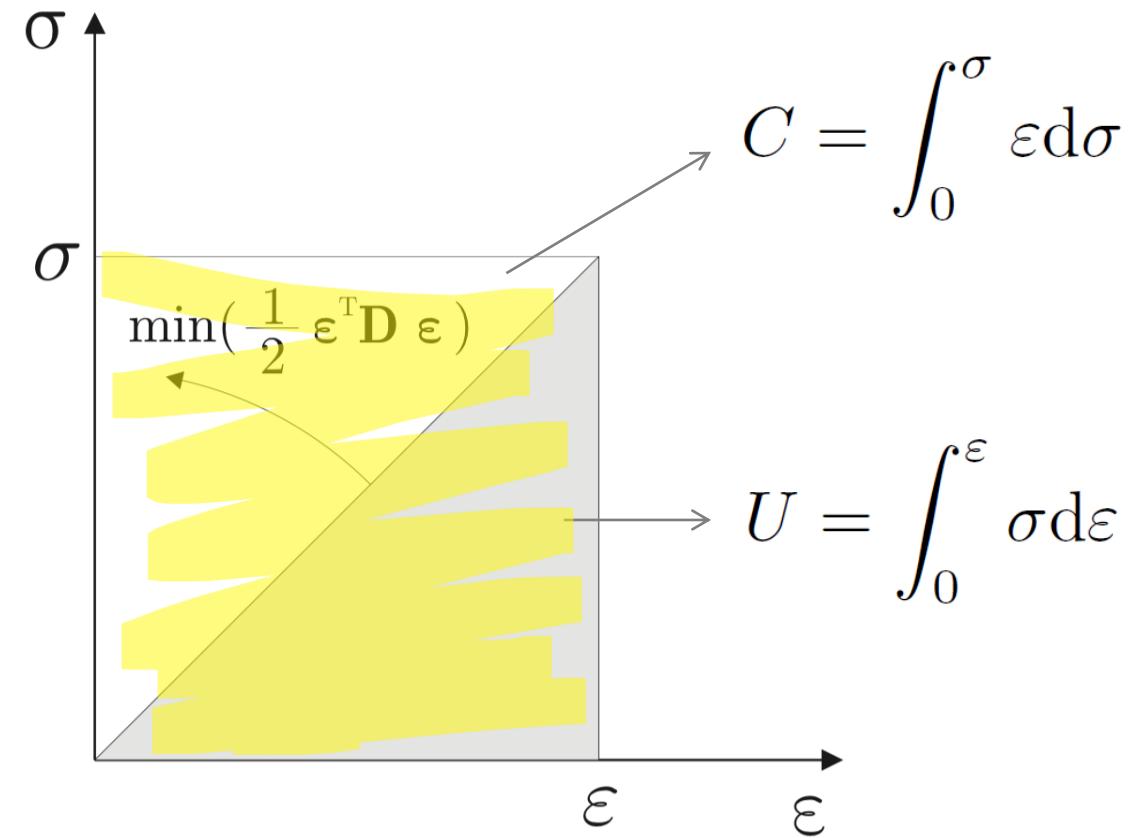
Elasticidade linear e energia de deformação



Elasticidade linear e energia de deformação complementar



Elasticidade linear e Trabalho das forças externa



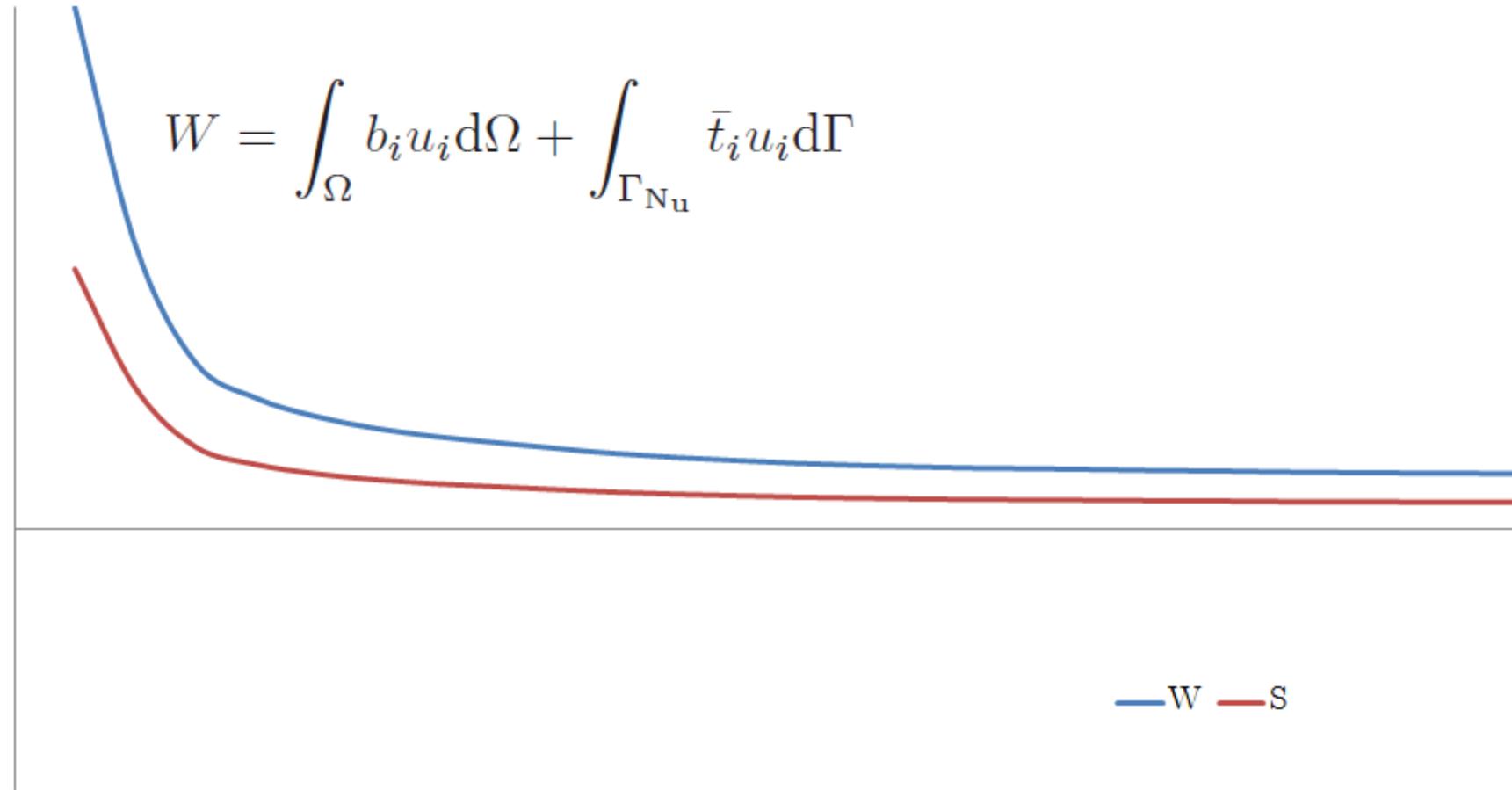
(iguais em elasticidade linear)

Elasticidade linear e energia de deformação

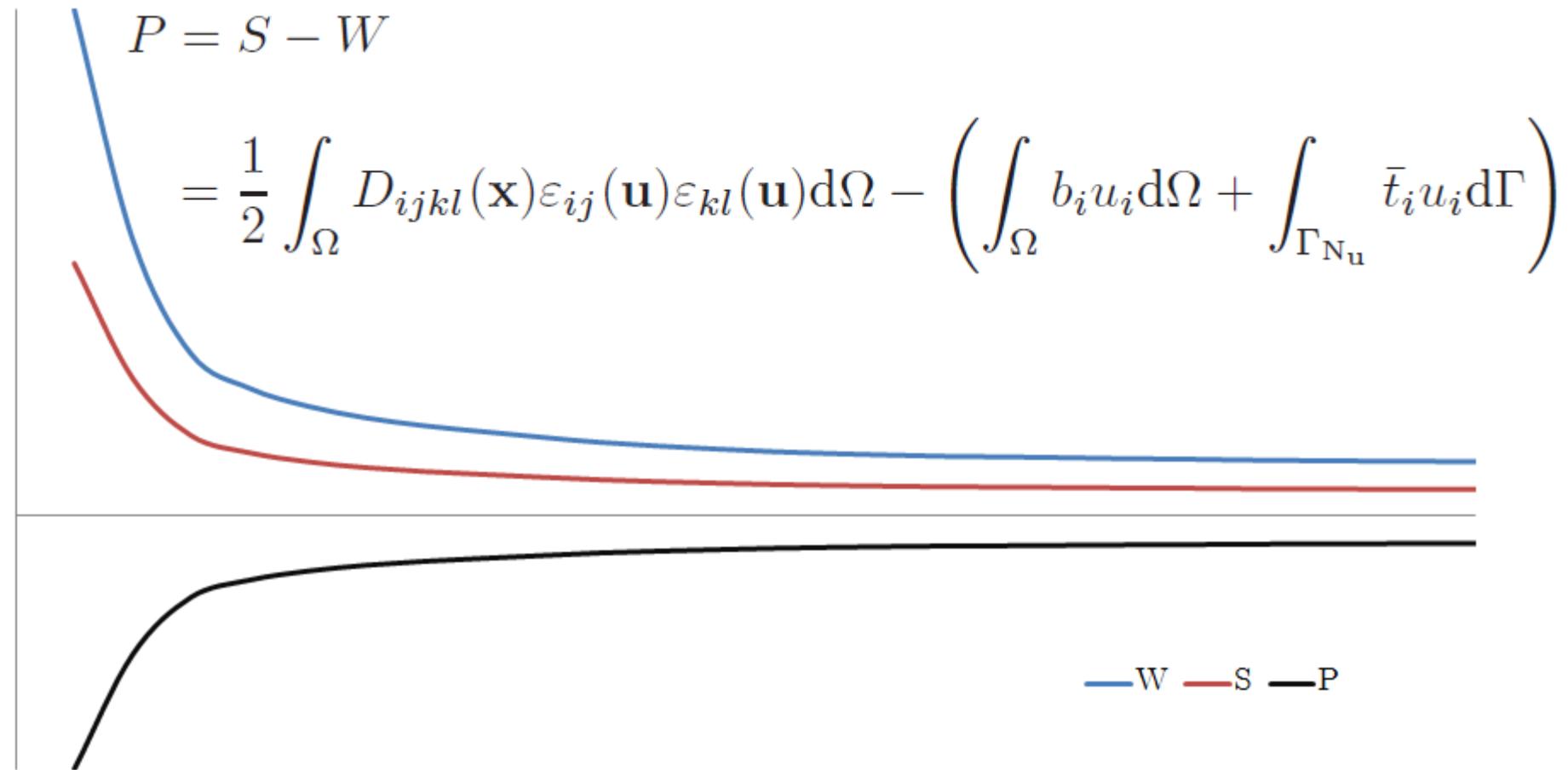
$$S = \frac{1}{2} \int_{\Omega} D_{ijkl}(x) \varepsilon_{ij}(u) \varepsilon_{kl}(u) d\Omega$$



Elasticidade linear e compliance



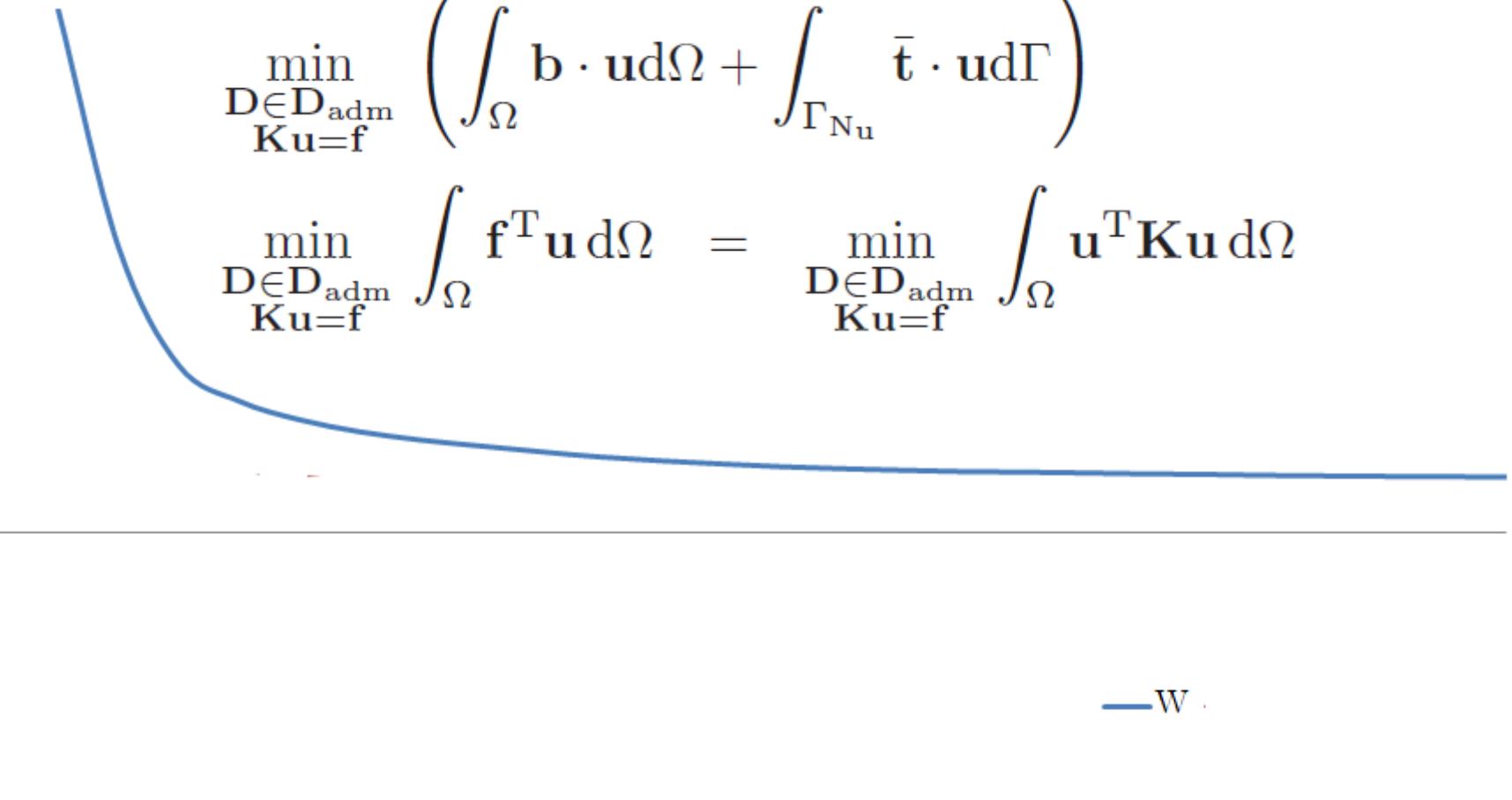
Elasticidade linear e energia potencial mínima



Elasticidade linear e otimização topológica

$$\min_{\substack{D \in D_{\text{adm}} \\ Ku=f}} \left(\int_{\Omega} b \cdot u d\Omega + \int_{\Gamma_{N_u}} \bar{t} \cdot u d\Gamma \right)$$

$$\min_{\substack{D \in D_{\text{adm}} \\ Ku=f}} \int_{\Omega} f^T u d\Omega = \min_{\substack{D \in D_{\text{adm}} \\ Ku=f}} \int_{\Omega} u^T Ku d\Omega$$



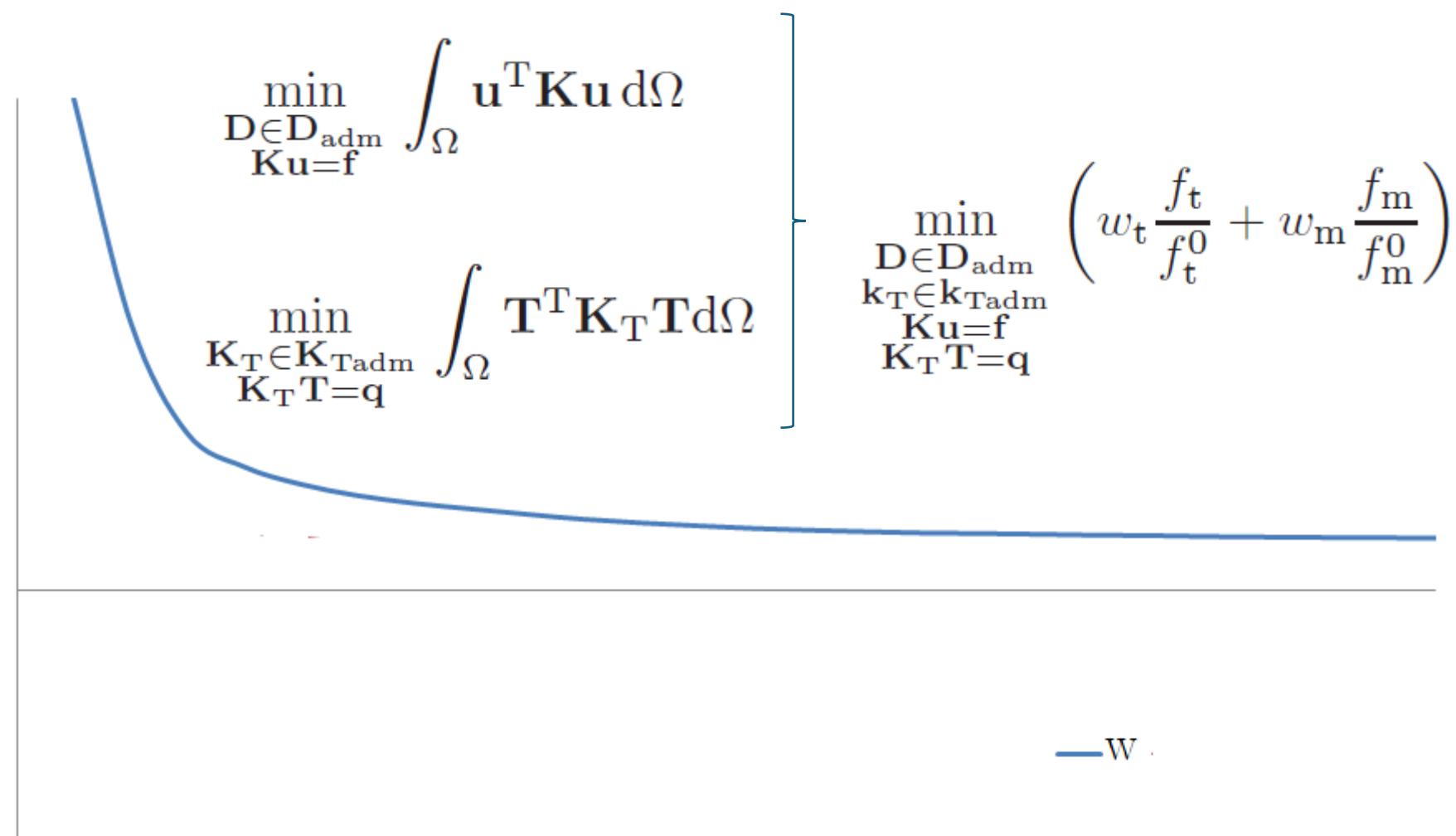
W

Otimização topológica e problemas semelhantes

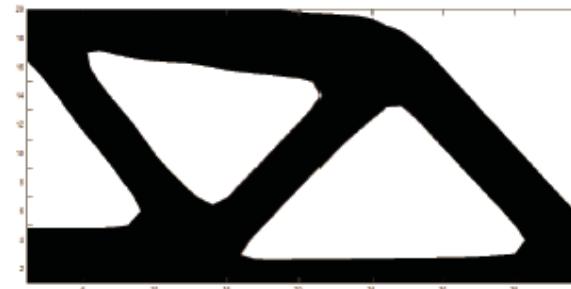
$$\min_{\substack{K_T \in K_{T\text{adm}} \\ q_i = -k_{ij} \frac{dT}{dx_j}}} \int_{\Gamma_{N_T}} \bar{q} T d\Gamma .$$
$$\min_{\substack{K_T \in K_{T\text{adm}} \\ K_T T = q}} \int_{\Omega} T^T K_T T d\Omega$$

— W

Otimização topológica multi-objetivo



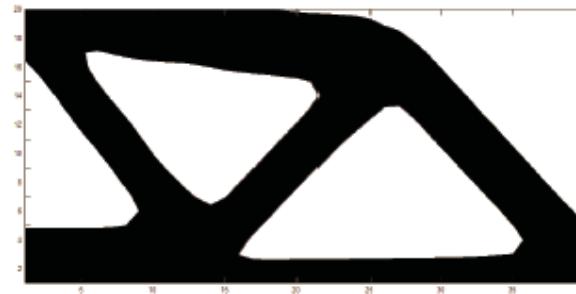
Otimização topológica



Com um processo de optimização, é possível avaliar a importância de cada parte do domínio

| $x_i=1$, material;
| $x_i=0$, vazio.

Otimização topológica e variáveis discretas e relaxação



As nossas variáveis passam a ser definidas
como **densidades**

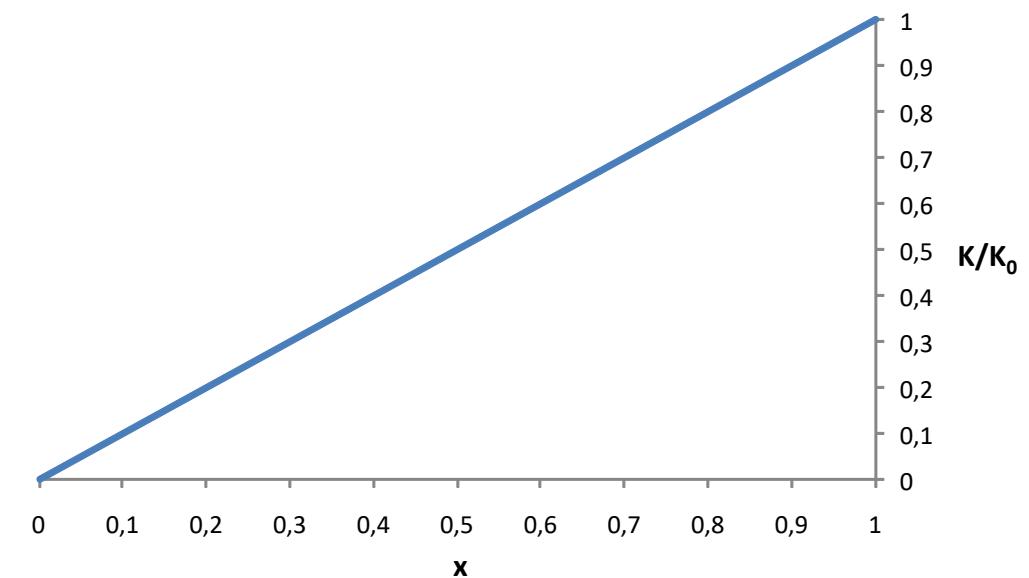
| $x_i=1$, material;
| $x_i=0$, vazio.

$$K = x K_0$$

Solid Isotropic Material with Penalization (SIMP)

- relaxar o problema original de variáveis discretas

$$0 < x_{\min} < x \leq 1$$



Otimização topológica: SIMP

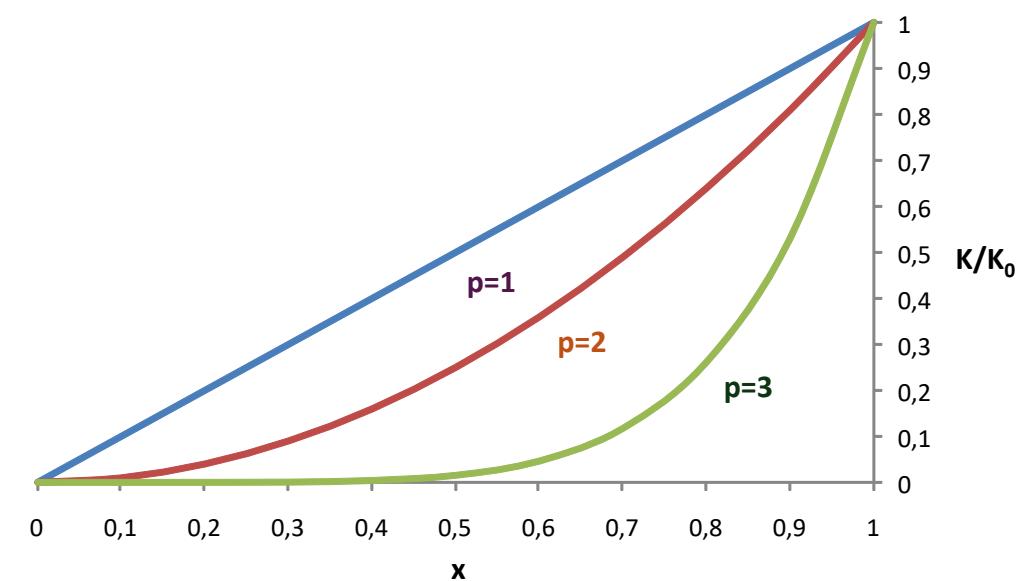
Solid Isotropic Material with Penalization (SIMP)

- relaxar o problema original de variáveis discretas

$$0 < x_{\min} < x \leq 1$$

- penalizar as densidades intermédias na distribuição de material

$$K = x^p K_0$$



Otimização topológica: SIMP

Solid Isotropic Material with Penalization (SIMP)

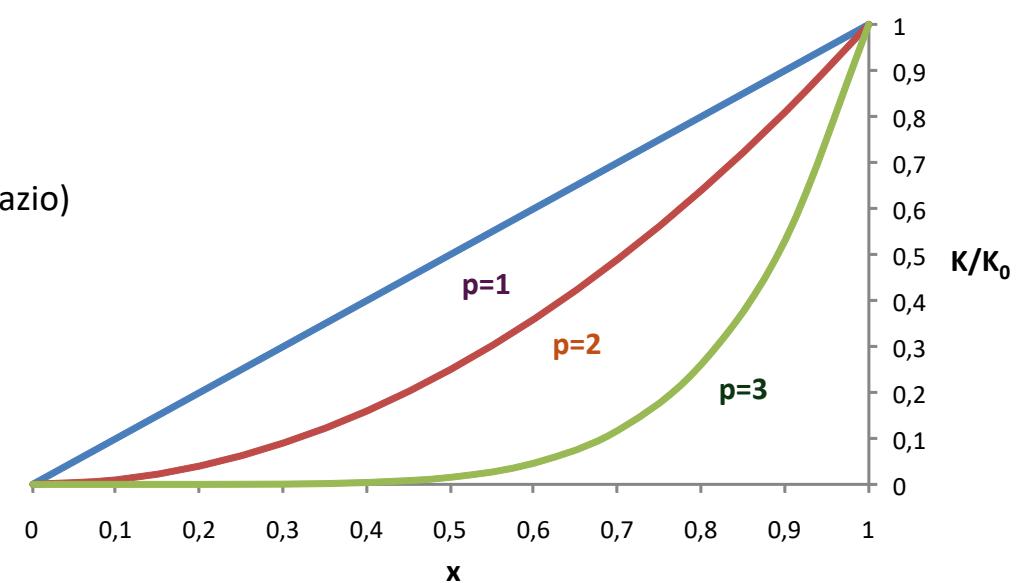
- relaxar o problema original de variáveis discretas

$$0 < x_{\min} < x \leq 1$$

- penalizar as densidades intermédias na distribuição de material

$$K = x^p K_0 \quad (\text{material/vazio})$$

$$K = x^p K_1 + (1-x^p) K_2 \quad (\text{2 materiais, sem vazio})$$



Otimização topológica: SIMP

Minimização de conformidade (*compliance*) usando **SIMP**

$$\min_{\mathbf{x}} : c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N (x_e)^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \rightarrow \text{Função Objectivo!}$$

sujeito a :

$$\frac{v(x)}{V_0} = f_v$$

$$: \quad \mathbf{K} \mathbf{U} = \mathbf{F}$$

$$: \quad \mathbf{0} \leq \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{1}$$

Medida de **compliance**
(minimização da energia de deformação, maximização da rigidez)

Variável de optimização

> Quantidade de material

> “densidade” > ρ

Potência de penalização

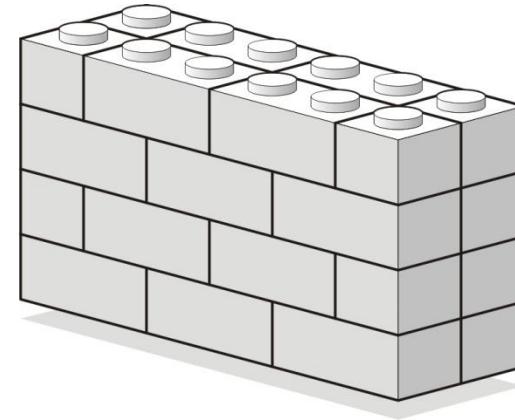
> Penaliza as densidades intermédias
(usualmente indesejadas)

Restrição de volume

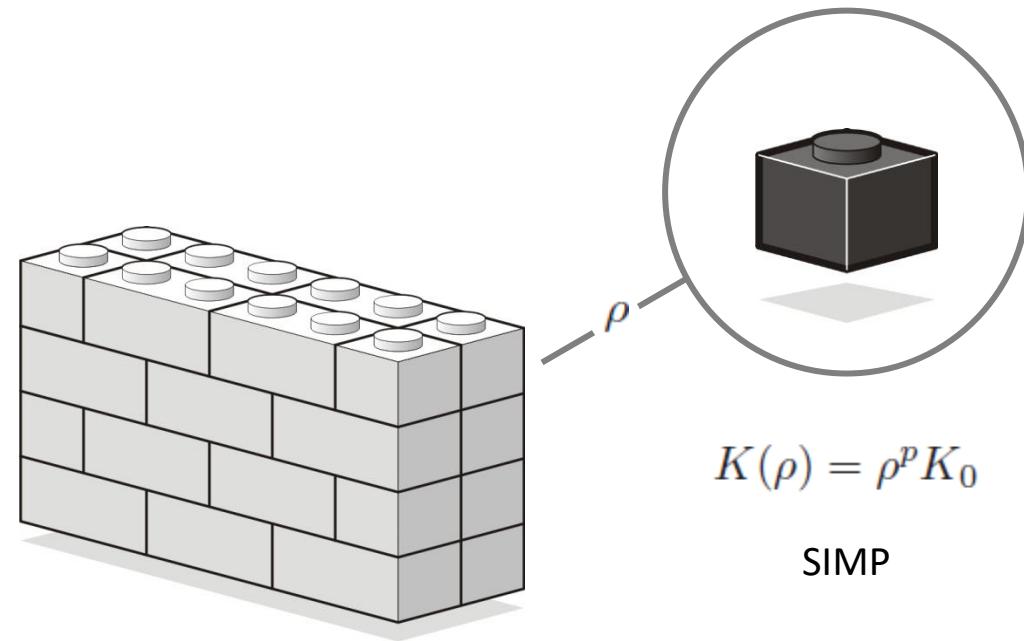
> minimiza-se a flexibilidade para um dado volume total



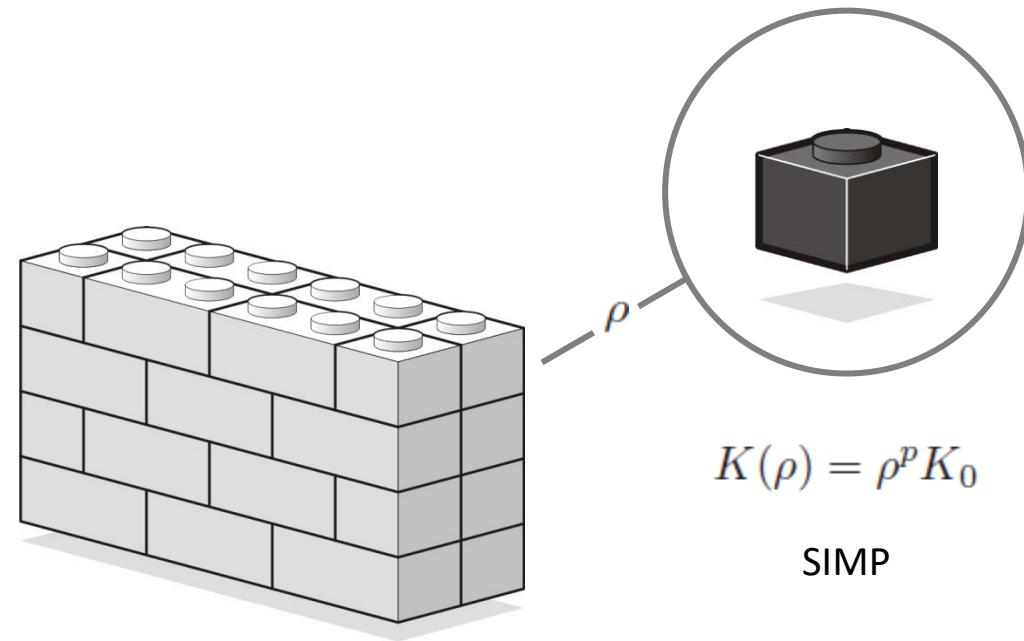
Otimização topológica: SIMP



Otimização topológica: SIMP

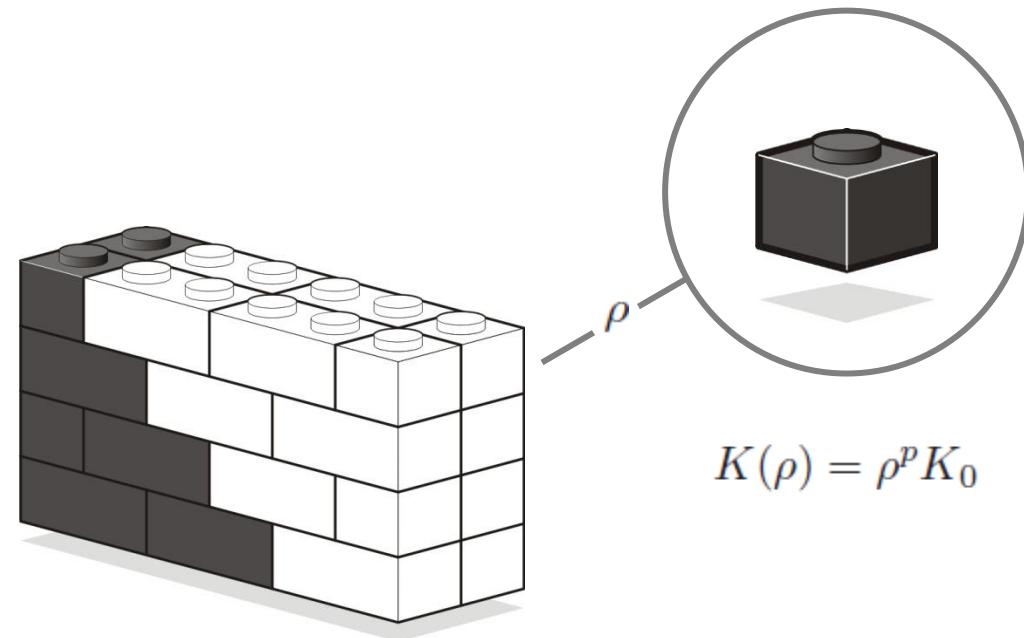


Otimização topológica: SIMP



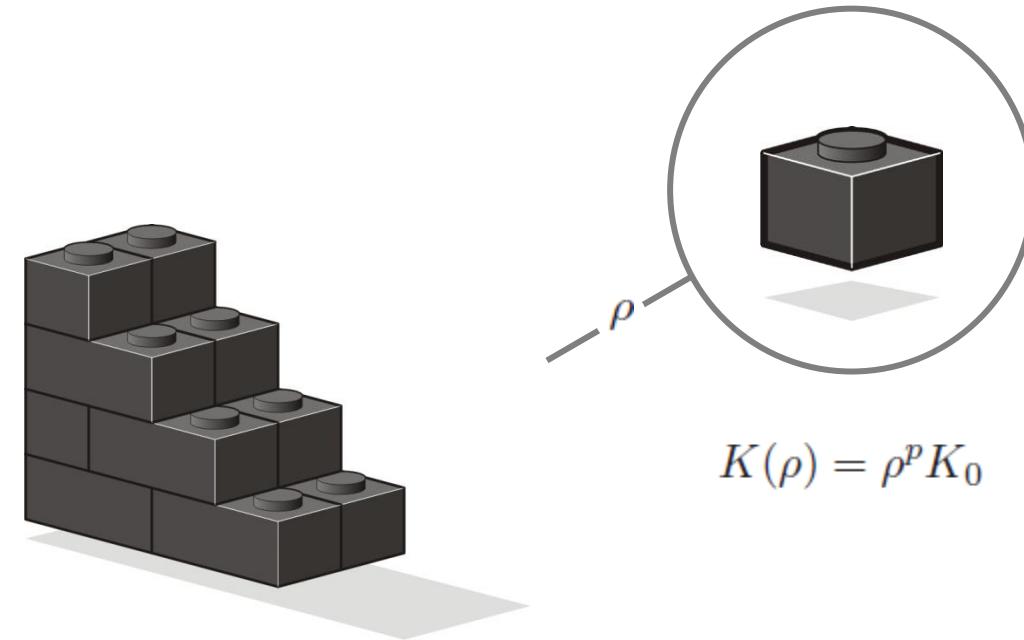
$$\min_{\begin{array}{l} 0 < \rho_{\min} \leq \rho_i \leq 1, i=1, \dots, n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_v \\ \mathbf{Ku} = \mathbf{f} \end{array}} f(\rho), \quad f(\rho) = \int_{\Omega} \mathbf{u}^T \mathbf{K} \mathbf{u} d\Omega = \sum_{e=1}^n \rho_e^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$

Otimização topológica: SIMP



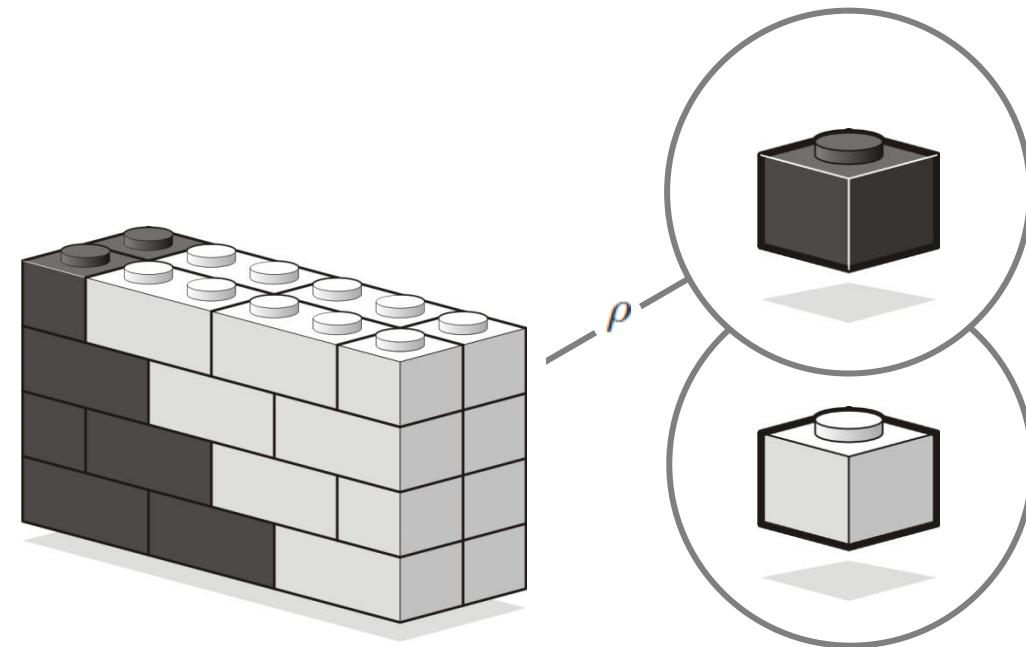
$$\min_{\begin{array}{l} 0 < \rho_{\min} \leq \rho_i \leq 1, i=1, \dots, n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_v \\ \mathbf{Ku} = \mathbf{f} \end{array}} f(\rho), \quad f(\rho) = \int_{\Omega} \mathbf{u}^T \mathbf{K} \mathbf{u} d\Omega = \sum_{e=1}^n \rho_e^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$

Otimização topológica: SIMP



$$\min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1, \dots, n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_v \\ \mathbf{Ku} = \mathbf{f}}} f(\rho), \quad f(\rho) = \int_{\Omega} \mathbf{u}^T \mathbf{K} \mathbf{u} d\Omega = \sum_{e=1}^n \rho_e^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$

Otimização topológica: SIMP



$$K(\rho) = \rho^p K_1 + (1 - \rho^p)K_2$$

Otimização topológica: SIMP

Minimização de conformidade (*compliance*) usando **SIMP**

$$\min_{\mathbf{x}} : c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N (x_e)^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \rightarrow \text{Função Objectivo!}$$

sujeito a :

$$\frac{v(x)}{V_0} = f_v$$

$$: \quad \mathbf{K} \mathbf{U} = \mathbf{F}$$

$$: \quad \mathbf{0} \leq \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{1}$$

Medida de **compliance**
(minimização da energia de deformação, maximização da rigidez)

Variável de optimização

> Quantidade de material

> “densidade” > $\boxed{?}$

Potência de penalização

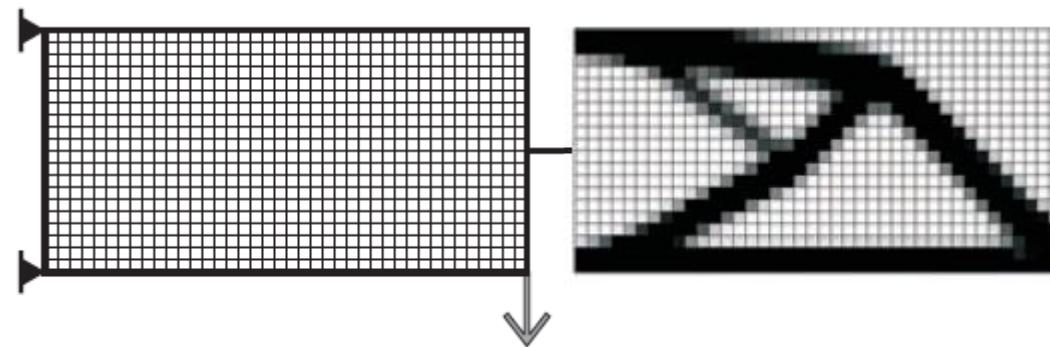
> Penaliza as densidades intermédias
(usualmente indesejadas)

Restrição de volume

> minimiza-se a flexibilidade para um dado volume total

Otimização topológica: SIMP

$$\begin{array}{l} \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1,\dots,n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T Ku d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{array}$$

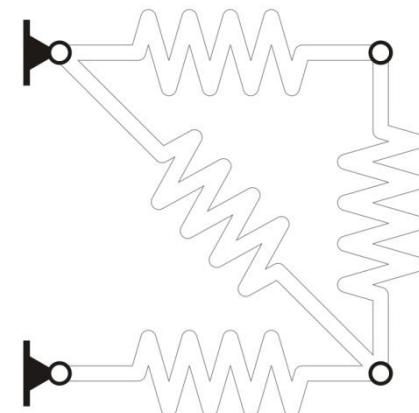


$$K(\rho) = \rho^p K_1$$

$$K(\rho) = \rho^p K_1 + (1 - \rho^p) K_2$$

Otimização topológica: sensibilidade

$$\begin{array}{c} \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1,\dots,n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T K u d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{array}$$

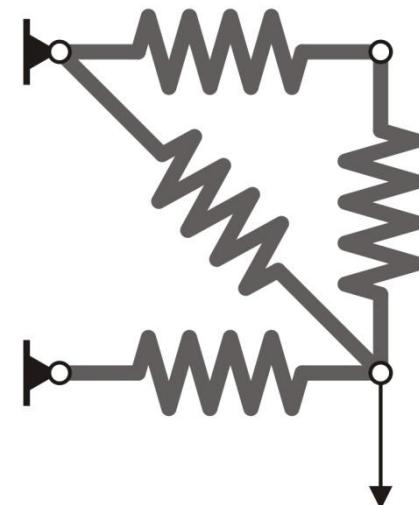


$$\frac{\partial K(\rho)}{\partial \rho} = p \rho^{p-1} K_0 .$$

$$\frac{\partial f(\rho)}{\partial \rho} = -p \rho_e^{p-1} u_e^T k_0 u_e ,$$

Otimização topológica: sensibilidade

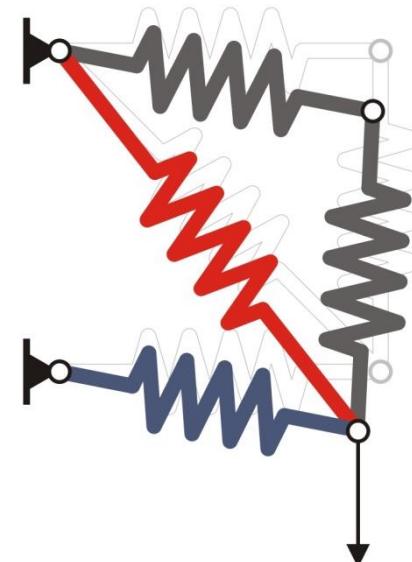
$$\begin{array}{c} \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1,\dots,n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T Ku d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{array}$$



$$\frac{\partial f(\rho)}{\partial \rho} = -p \rho_e^{p-1} u_e^T k_0 u_e ,$$

Otimização topológica: sensibilidade

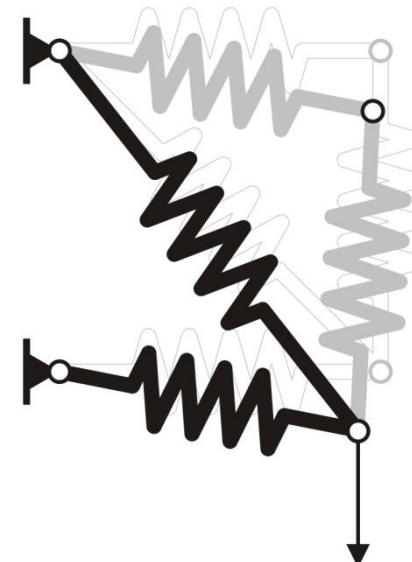
$$\begin{array}{c} \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1,\dots,n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T K u d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{array}$$



$$\frac{\partial f(\rho)}{\partial \rho} = -p \rho_e^{p-1} u_e^T k_0 u_e ,$$

Otimização topológica: sensibilidade

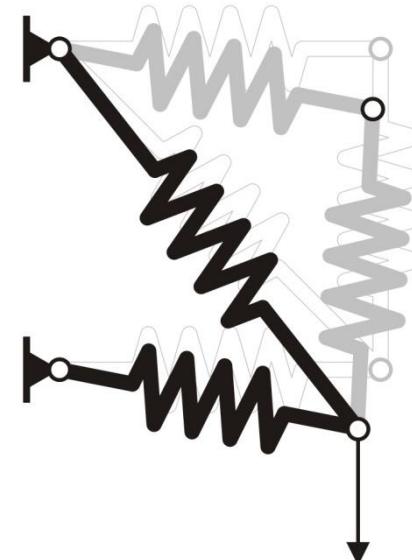
$$\begin{array}{c} \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1,\dots,n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T K u d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{array}$$



$$\frac{\partial f(\rho)}{\partial \rho} = -p \rho_e^{p-1} u_e^T k_0 u_e ,$$

Otimização topológica: sensibilidade

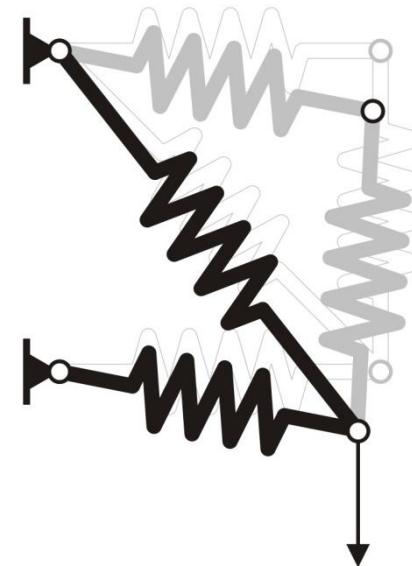
$$\begin{aligned} & \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1, \dots, n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T K u d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{aligned}$$



Já agora, alguma ideia mais “heurística” para resolver este mesmo problema?

Otimização topológica: sensibilidade

$$\begin{aligned} & \min_{\substack{0 < \rho_{\min} \leq \rho_i \leq 1, i=1, \dots, n \\ \sum_{i=1}^n \frac{\rho_i v_i}{V_\Omega} \leq f_V \\ Ku=f}} \int_{\Omega} u^T K u d\Omega = \sum_{e=1}^n \rho_e^p u_e^T k_0 u_e \end{aligned}$$



Já agora, alguma ideia mais “heurística” para resolver este mesmo problema?

Sugestões de pesquisa:
. (B)ESO
. Hard-kill e Soft-kill.



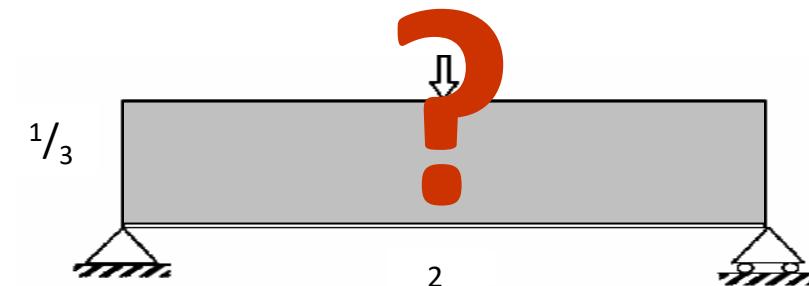
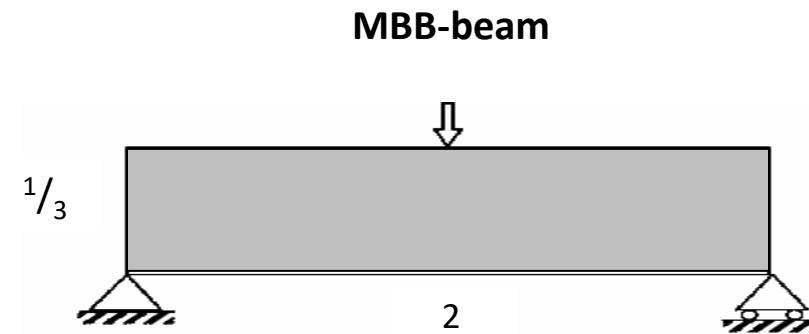
Otimização topológica



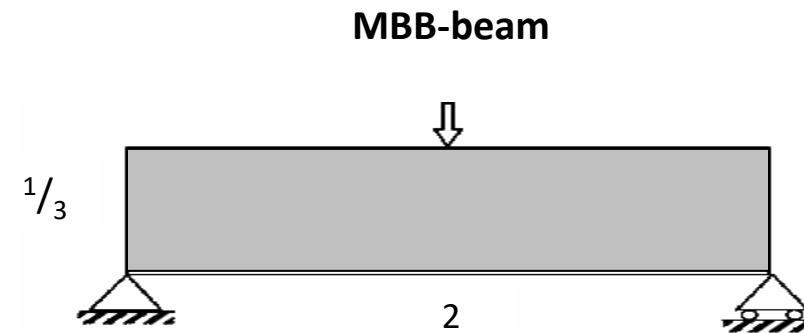
Exemplos



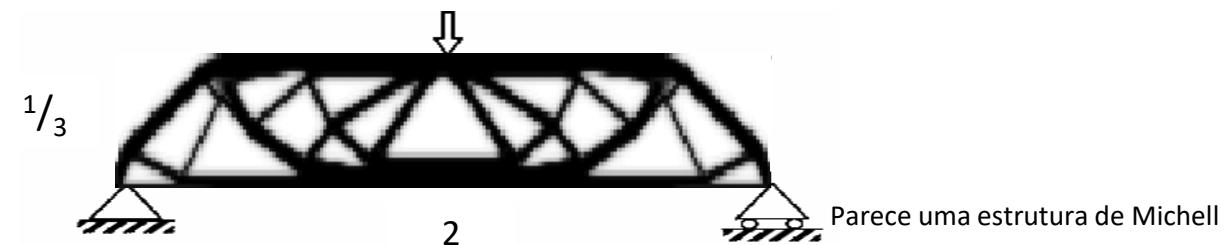
Exemplos

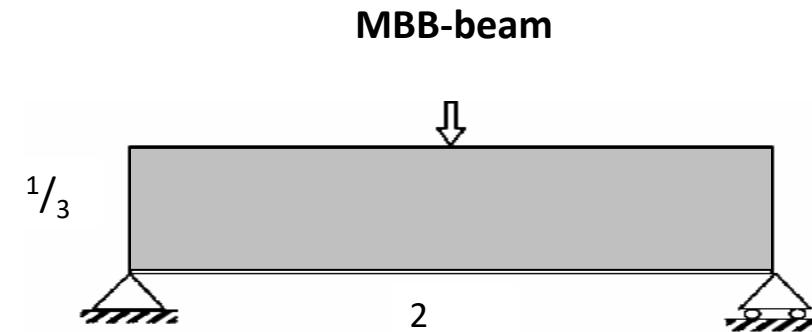


Topologia óptima para
maximização da rigidez com
40% de volume?

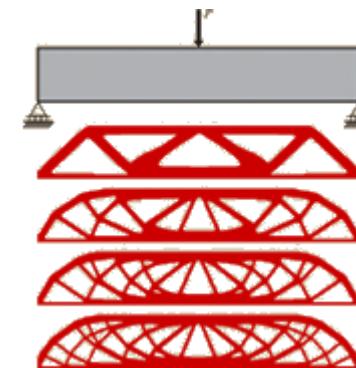


Otimização topológica resultou na estrutura ótima seguinte





Os resultados dependem do refinamento de malha (mínimos locais)

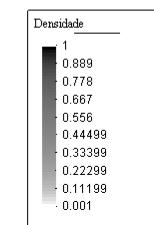


Parece uma estrutura de Michell



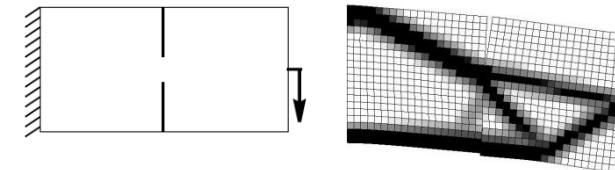
Exemplos

Cantilever-beam

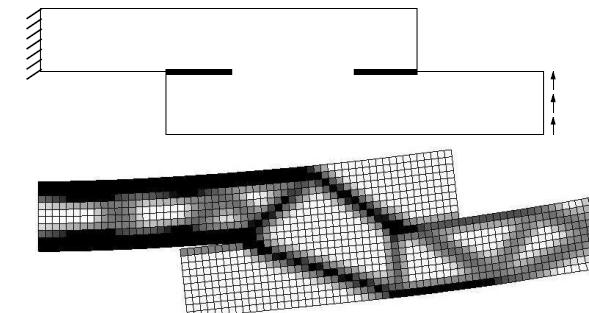


1. Rectangular domain with two contact surfaces (B.

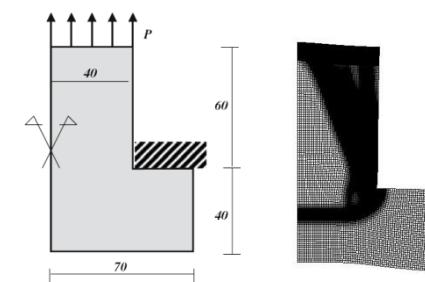
Desmorat, Int. J. Sol Struct. 44, 1132-1144, 2007)



2. Linked Bars (B. Desmorat, Int. J. Sol Struct. 44, 1132-1144, 2007)

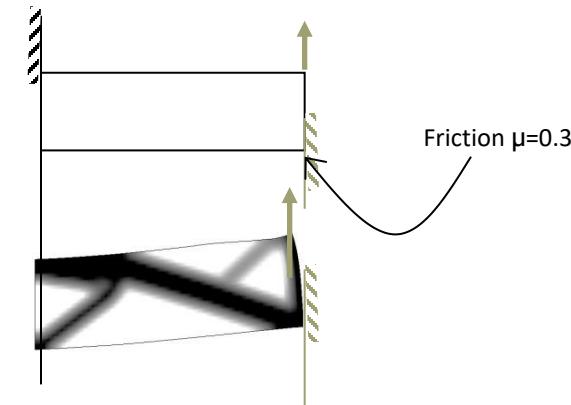


3. L-Shaped test (E.A. Fancello, Struct. Multidisc Optim. 32, 229-240, 2006)



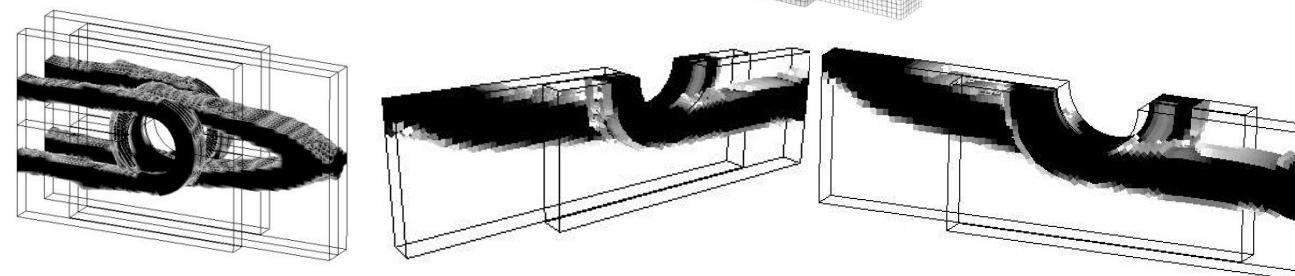
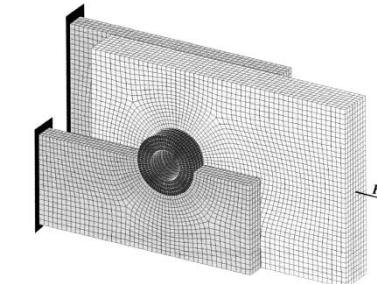
Exemplos

4. Cantilever beam in contact with friction



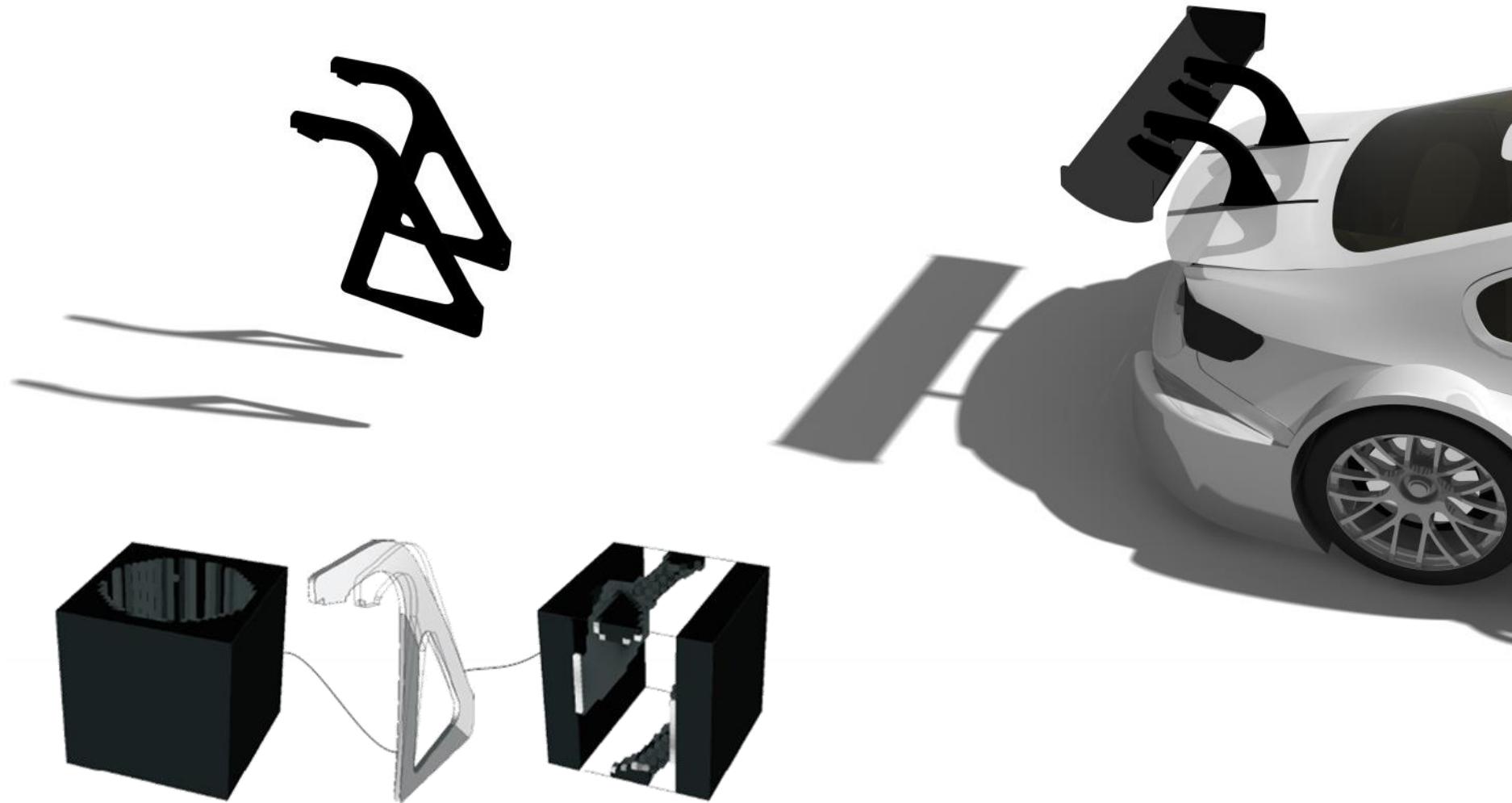
5. Flexible pinned union

(J. Folgado, *Modelos Computacionais para análise e projecto de próteses ortopédicas*, PhD Thesis, IST, 2004)



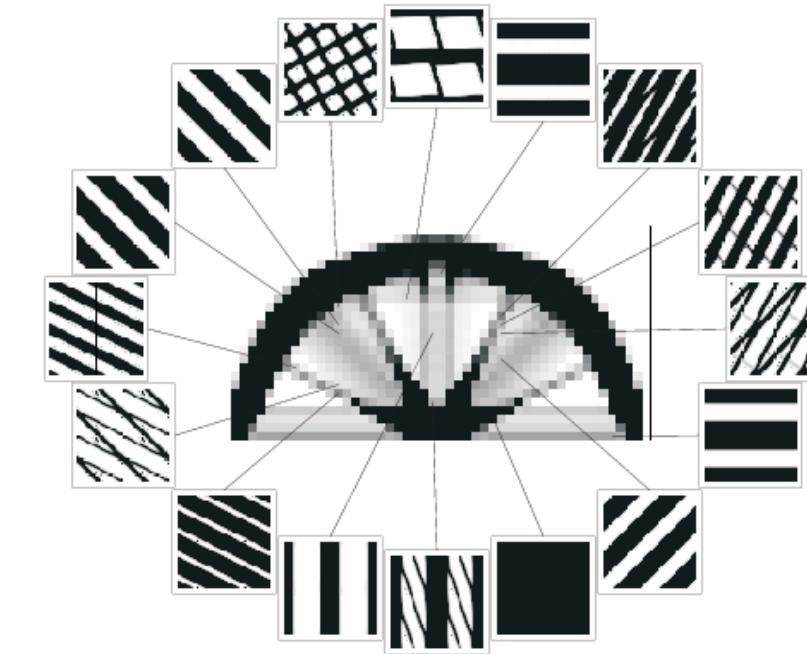
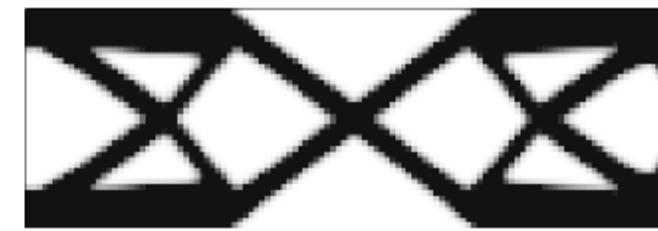


Exemplos

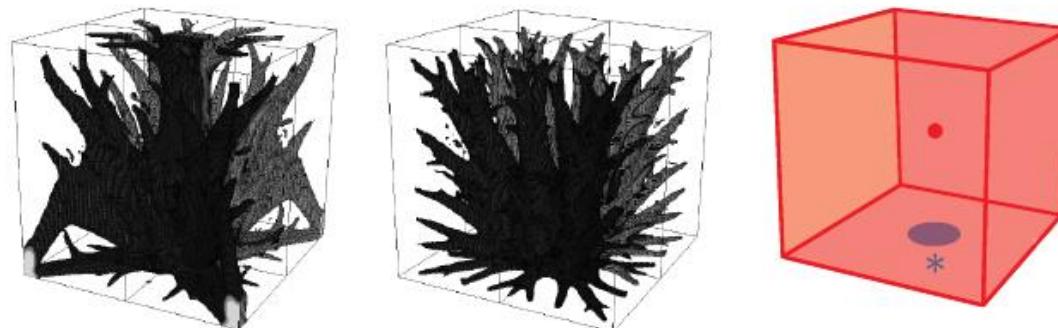
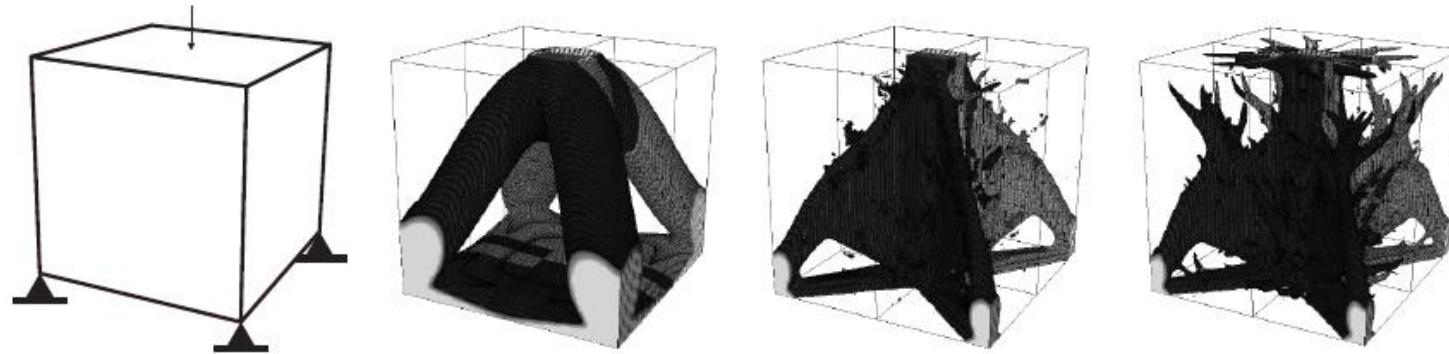




Exemplos



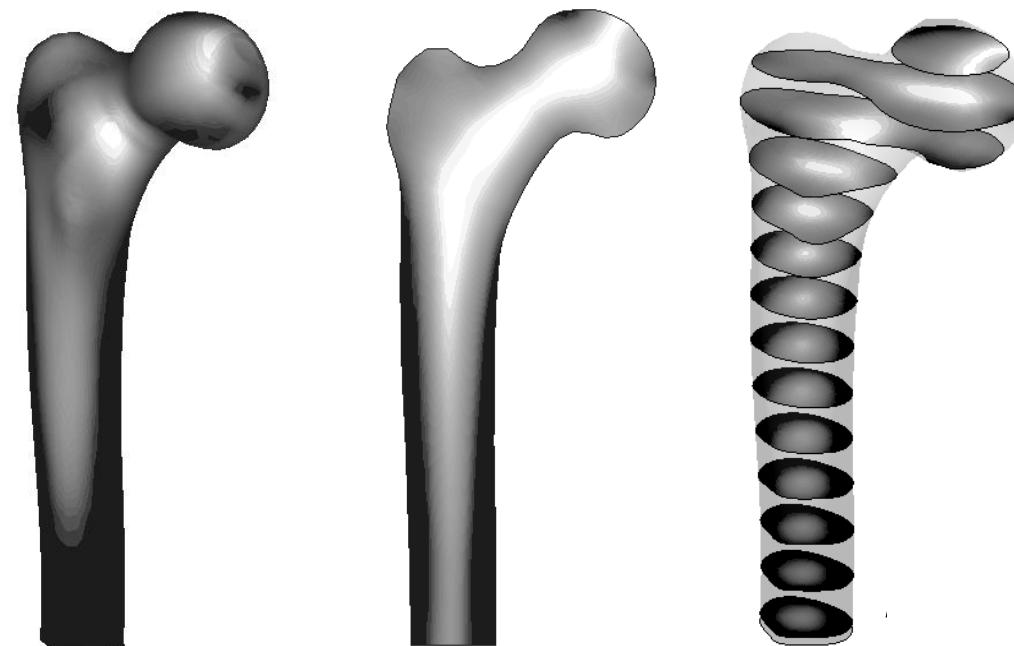
Exemplos: multiobjetivo



Resultados para a distribuição aparente de densidade óssea



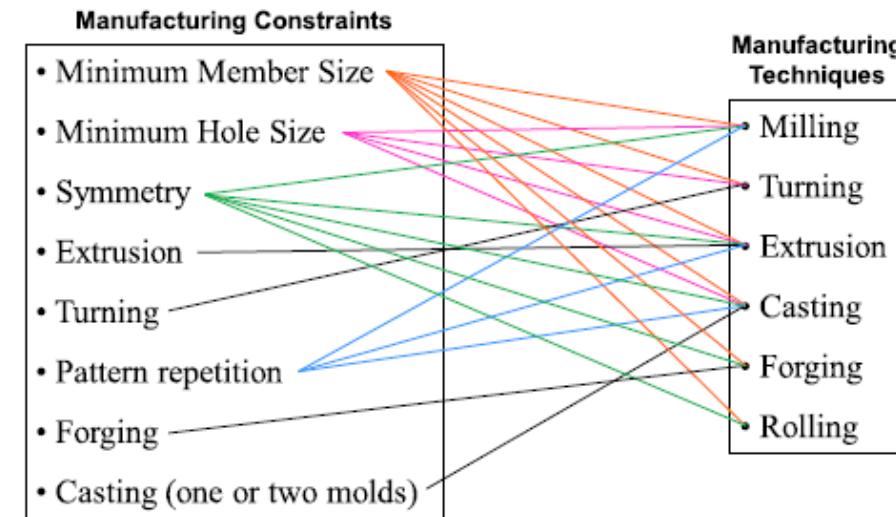
elementos hexaédricos; fração volúmica de 55%.



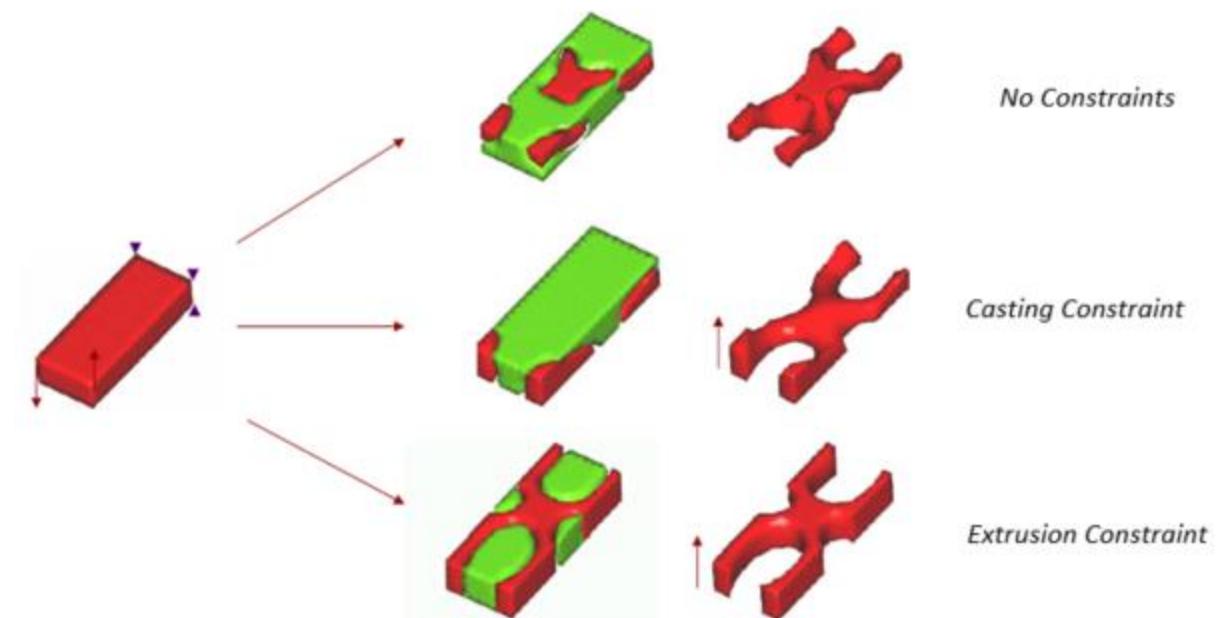
Boa aproximação à distribuição de densidade óssea de um fémur real.

Apparent density distribution of the trabecular bone using a volume fraction of 55% ($E \approx 1.185 \times 10^4 \text{ N mm}$ and $\lambda = 1.284 \text{ N/mm}^2$).
Longitudinal and transversals cuts of the femur's model.

Exemplos: outras restrições

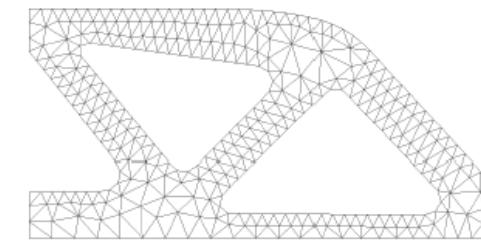
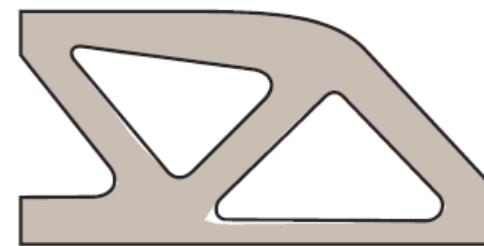
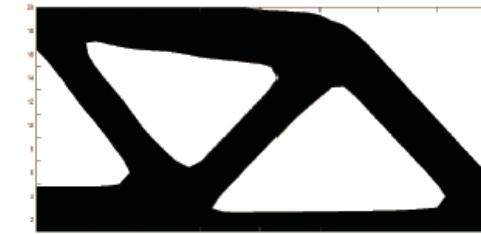
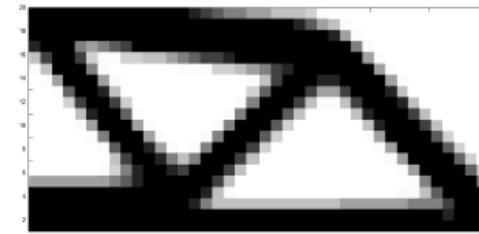


Exemplos: outras restrições



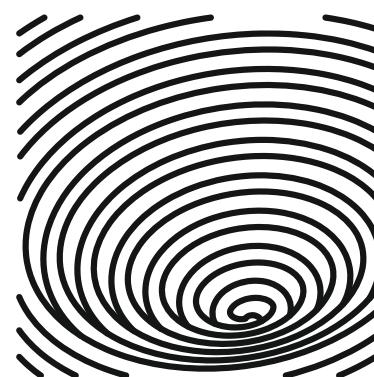


Pós-processamento e workflow





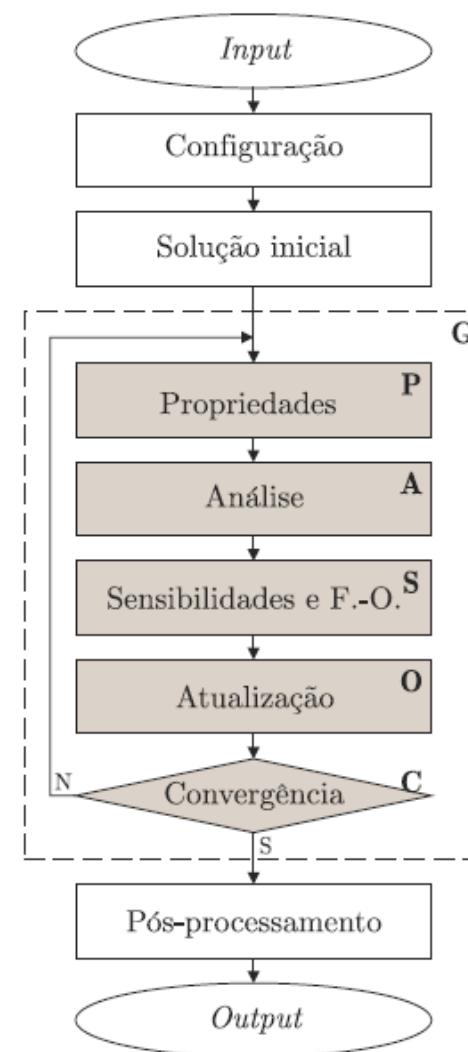
Otimização topológica



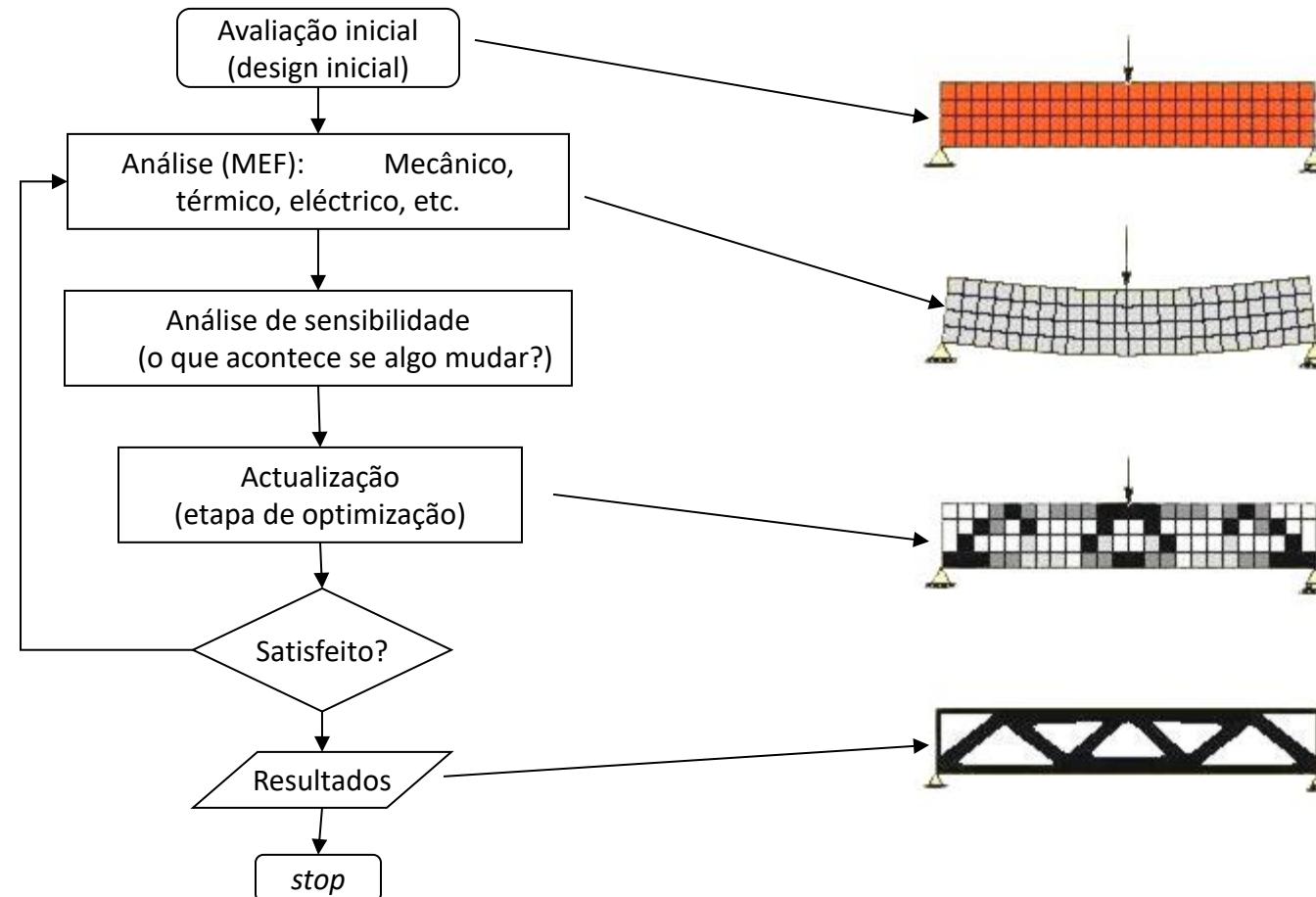
Desenvolvimento numérico



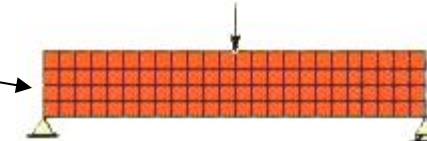
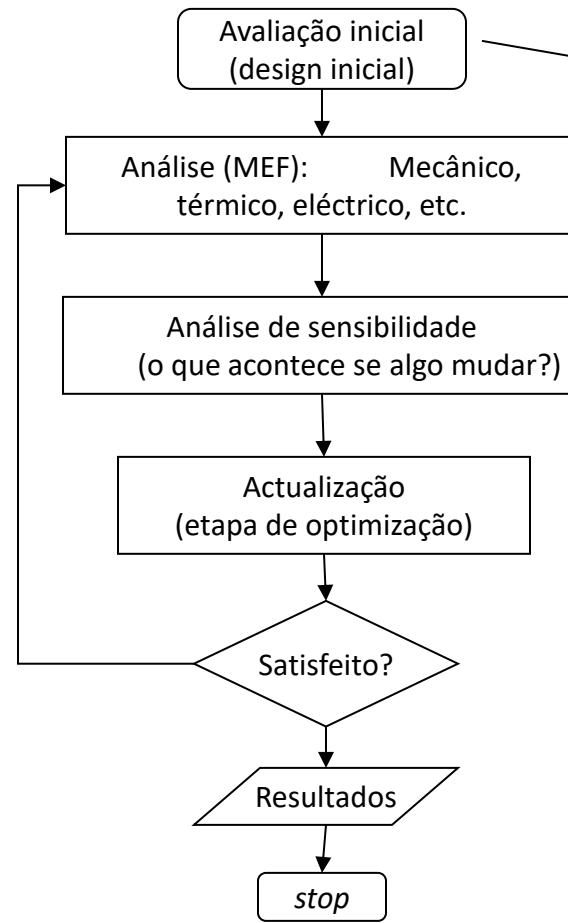
Fluxograma



Fluxograma



Fluxograma: solução inicial



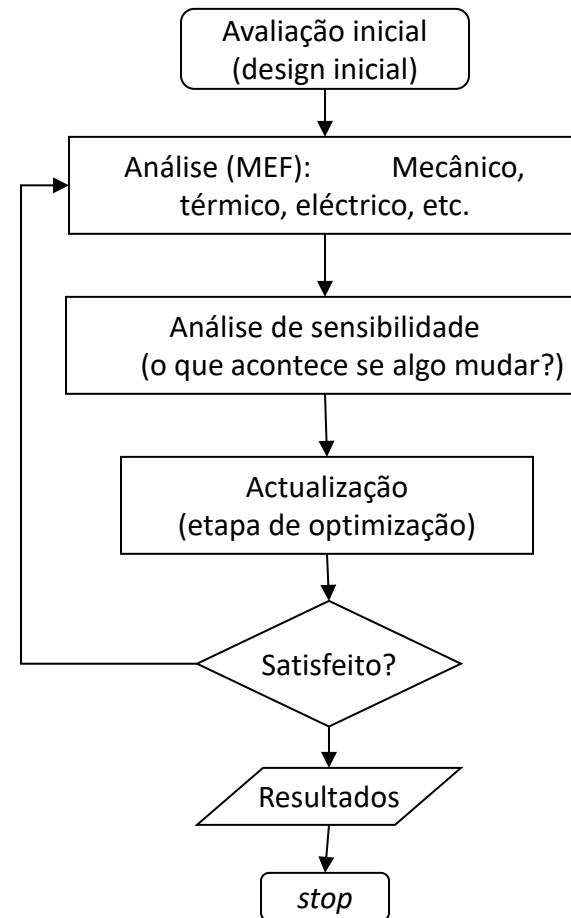
Restrição de volume

$$\frac{v(x)}{V_0} = f_v$$

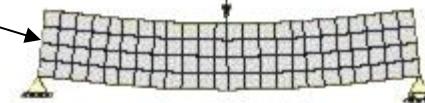
deve ser verificada em todas as iterações, incluindo a primeira

> neste caso, a abordagem mais simples é impor este valor em todas as variáveis x
 $= v/V_0 = f$

Fluxograma: avaliação da FO



$$c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N (x_e)^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$



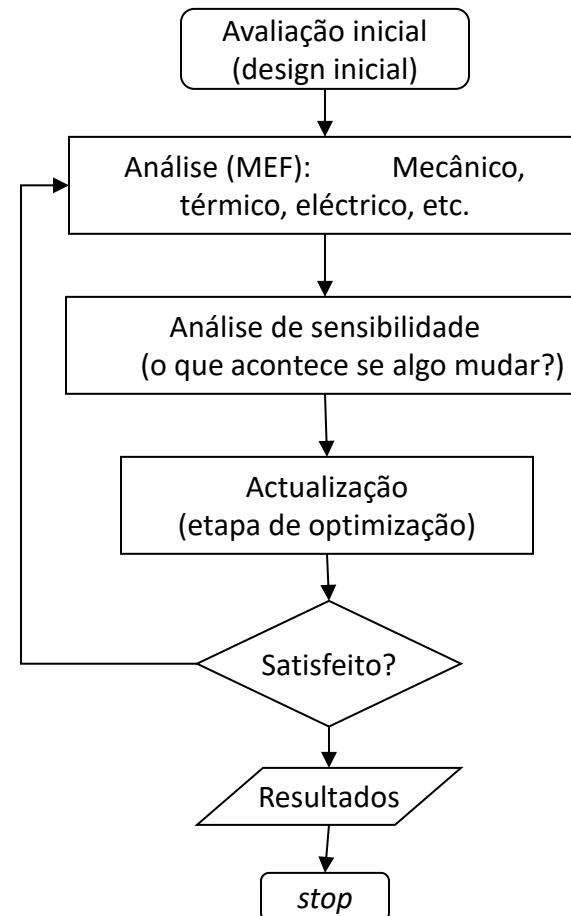
Problema típico de elementos finitos,
mas a rigidez elementar é multiplicada
por x_e^p

$$c(\mathbf{x}) = \sum_{e=1}^N \mathbf{u}_e^T [(x_e)^p \mathbf{k}_0] \mathbf{u}_e$$

Calculam-se os deslocamentos para cada
elemento e o trabalho das forças
externas

> Função-objectivo

Fluxograma: análise da sensibilidade



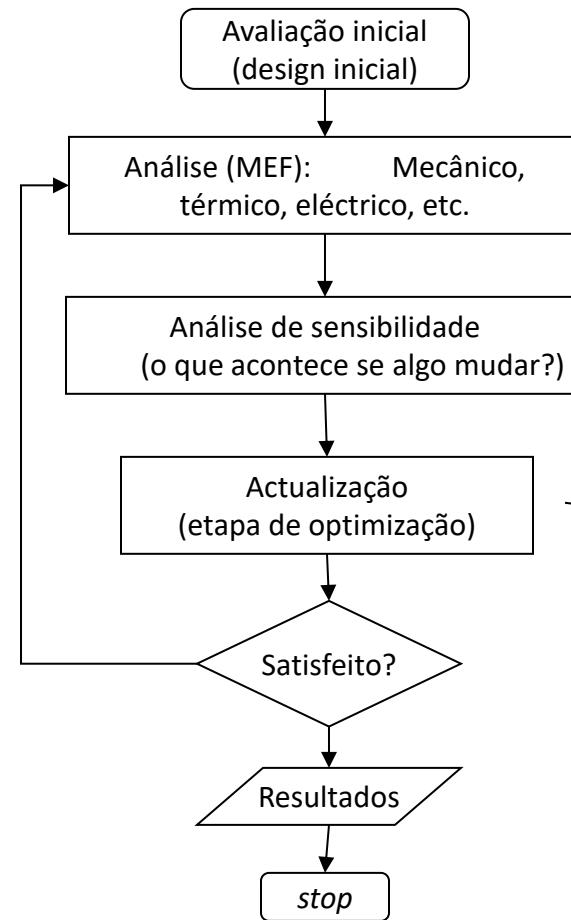
Função-objectivo dada então por

$$c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N (x_e)^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$

Sensibilidade (gradiente) pode ser calculada como

$$\frac{\partial c}{\partial x_e} = -p(x_e)^{p-1} \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$$

Fluxograma: atualização



Actualização feita recorrendo a um esquema de ponto fixo (critério de óptimo)

$$x_e^{\text{new}} =$$

$$\begin{cases} \max(x_{\min}, x_e - m) & \text{if } x_e B_e^n \leq \max(x_{\min}, x_e - m), \\ x_e B_e^n & \text{if } \max(x_{\min}, x_e - m) < x_e B_e^n < \min(1, x_e + m), \\ \min(1, x_e + m) & \text{if } \min(1, x_e + m) \leq x_e B_e^n, \end{cases}$$



Onde B é uma função de verificação que usa a sensibilidade e verifica a restrição de volume

x pode ser actualizada utilizando diversos métodos (critérios de óptimo, SLP, MMA, CONLIN, etc.)

Fluxograma: critério de ótimo

Método de critério de óptimo (esquema heurístico de actualização)

$$x_e^{\text{new}} =$$

$$\begin{cases} \max(x_{\min}, x_e - m) & \text{if } x_e B_e^\eta \leq \max(x_{\min}, x_e - m), \\ x_e B_e^\eta & \text{if } \max(x_{\min}, x_e - m) < x_e B_e^\eta < \min(1, x_e + m), \\ \min(1, x_e + m) & \text{if } \min(1, x_e + m) \leq x_e B_e^\eta, \end{cases}$$

limite-móvel

Coeficiente de amortecimento numérico

Parâmetro de verificação

Fluxograma: critério de ótimo

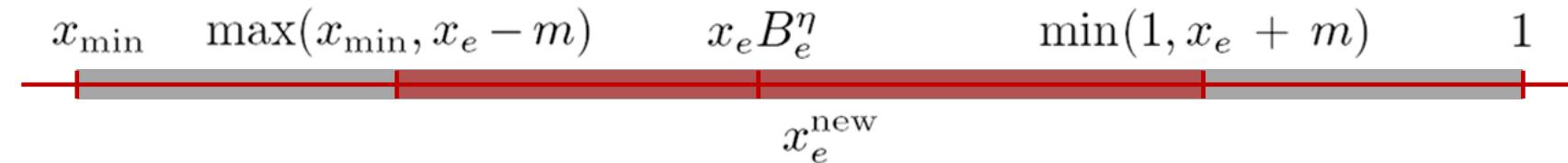
$$x_e^{\text{new}} =$$

$$\begin{cases} \max(x_{\min}, x_e - m) & \text{if } x_e B_e^\eta \leq \max(x_{\min}, x_e - m), \\ x_e B_e^\eta & \text{if } \max(x_{\min}, x_e - m) < x_e B_e^\eta < \min(1, x_e + m), \\ \min(1, x_e + m) & \text{if } \min(1, x_e + m) \leq x_e B_e^\eta, \end{cases}$$

limite-móvel

Coeficiente de amortecimento numérico

Parâmetro de verificação



Fluxograma: critério de ótimo

$$x_e^{\text{new}} =$$

$$\begin{cases} \max(x_{\min}, x_e - m) & \text{if } x_e B_e^\eta \leq \max(x_{\min}, x_e - m), \\ x_e B_e^\eta & \\ \min(1, x_e + m) & \text{if } \max(x_{\min}, x_e - m) < x_e B_e^\eta < \min(1, x_e + m), \end{cases}$$

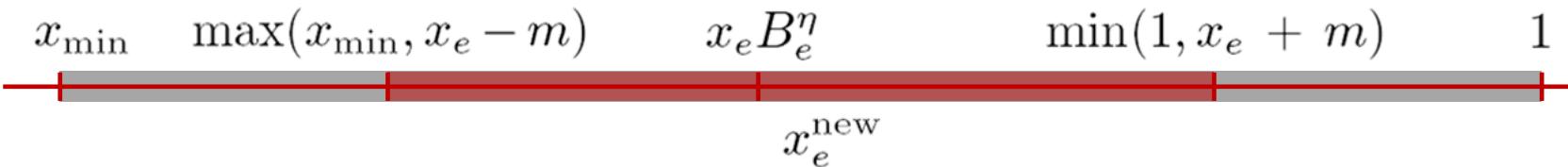
limite-móvel

Coeficiente de amortecimento numérico

Parâmetro de verificação

Se B_e tender para 1, pelo que, quando converge,

$$\min(1, x_e^{\text{new}}) = x_e B_e^\eta$$



Fluxograma: critério de ótimo

B pode ser definido por

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}},$$

→ Sensibilidade da FO
→ Sensibilidade da restrição

Existe um valor do multiplicador λ que verifica a restrição de volume.

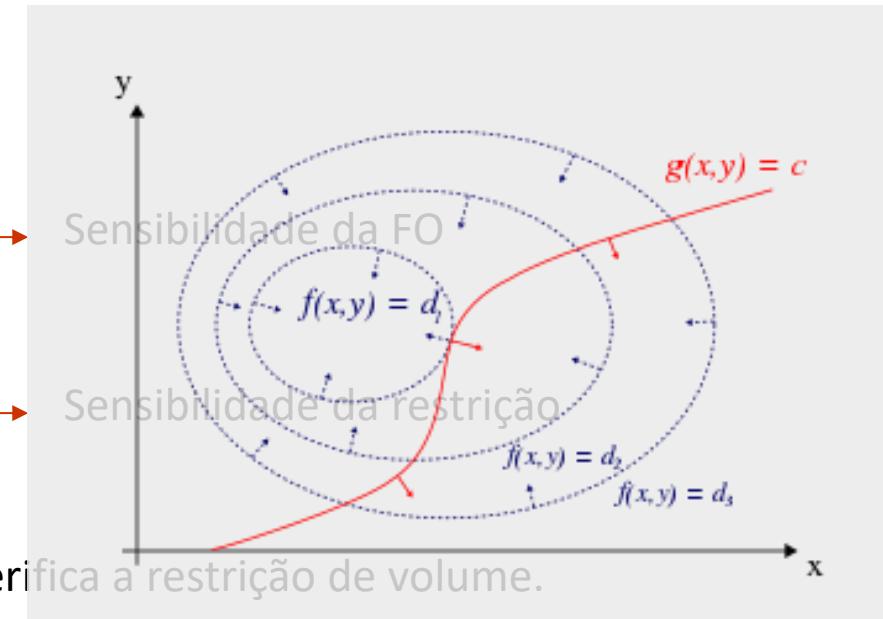
Este esquema é desenvolvido a partir da condição de estacionariedade do método dos Multiplicadores de Lagrange

$$\frac{\partial c}{\partial x_e} + \lambda \frac{\partial V}{\partial x_e} = 0$$

Fluxograma: critério de ótimo

B pode ser definido por

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}},$$



Existe um valor do multiplicador λ que verifica a restrição de volume.

Este esquema é desenvolvido a partir da condição de estacionariedade do método dos Multiplicadores de Lagrange

$$\frac{\partial c}{\partial x_e} + \lambda \frac{\partial V}{\partial x_e} = 0$$

Fluxograma: critério de ótimo

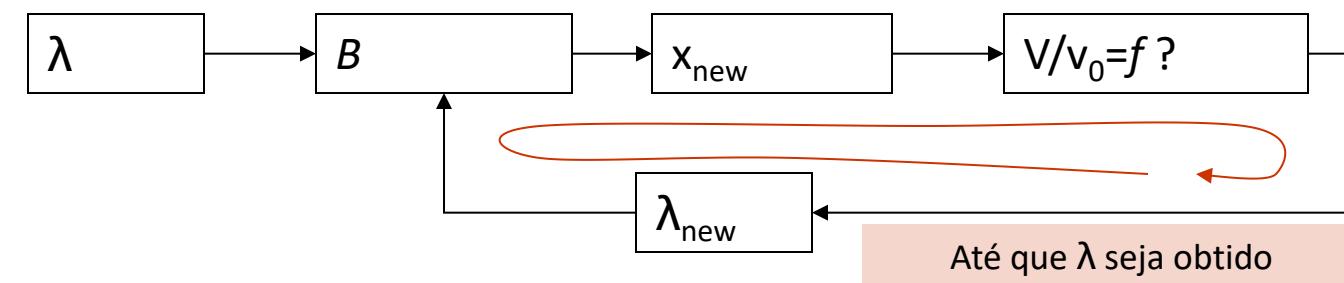
B pode ser definido por

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}},$$

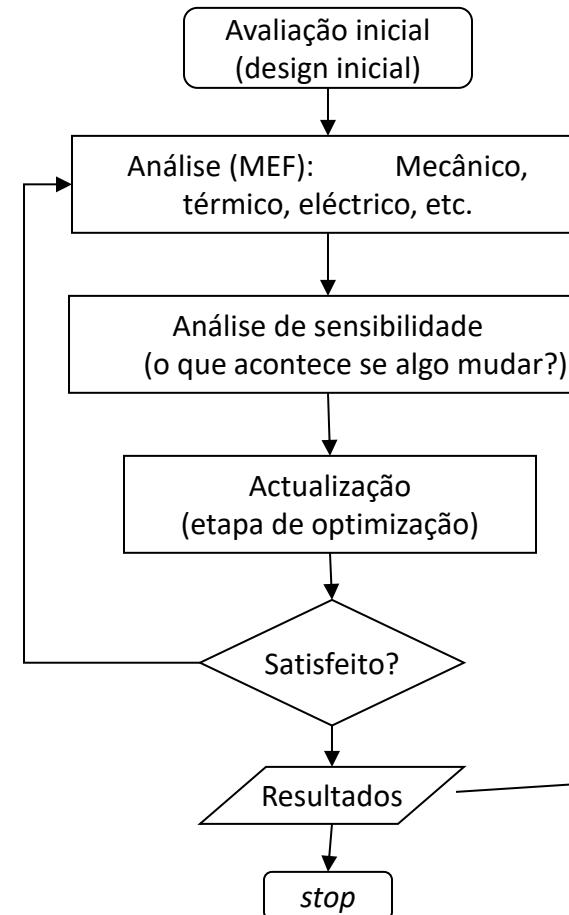
Sensibilidade da FO
Sensibilidade da restrição

Existe um valor do multiplicador λ que verifica a restrição de volume.

Pode ser actualizado num ciclo interno (e.g. algoritmo de bissecção)



Fluxograma: pós-processamento

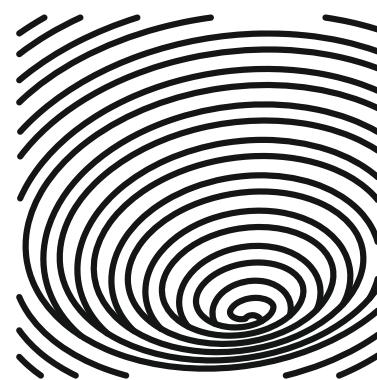


O pós-processamento pode ser feito
com diversas ferramentas
(GiD, Tecplot, Abaqus CAE, etc.)





Otimização topológica



Lab

Fornecido e analisado um código MatLab®

A 99 line topology optimization code written in Matlab

O. Sigmund, Struct. Multidisc. Optim. 21, 120–127, 2001.



Labwork: códigos para teste

Fornecido e analisado um código MatLab®

A 99 line topology optimization code written in Matlab

O. Sigmund, Struct. Multidisc. Optim. 21, 120–127, 2001.

topopt.m
+
topplot.m
toptherm.m
+
topopt.py

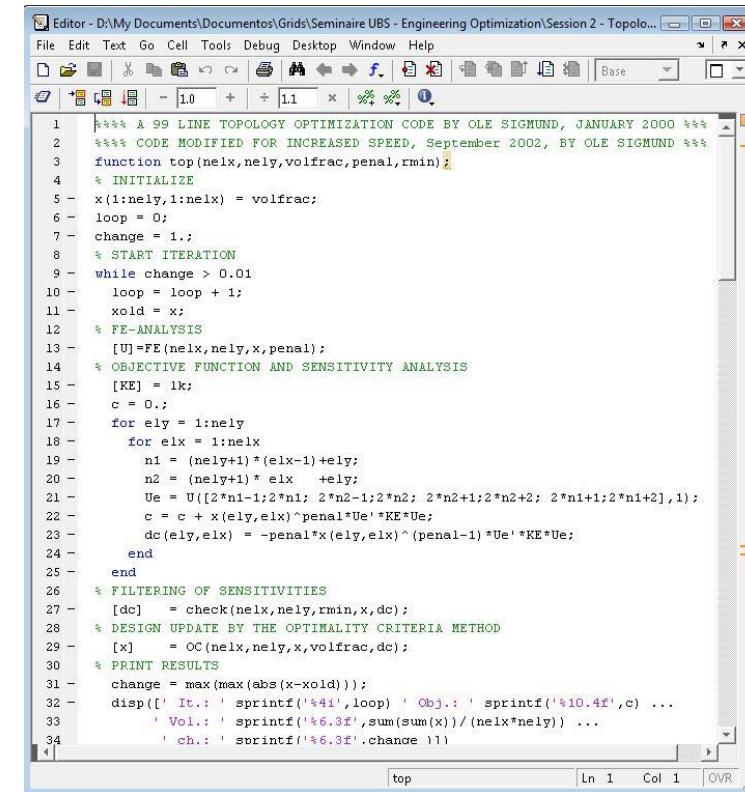




Material de suporte

top.m

- 1) Abrir o código **topopt.m** com **MatLab**, **Mathematica**, **Scilab**, **Octave**, **Rlab** ou equivalente



```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;

% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
c = 0.;

% FE-ANALYSIS
[U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[KE] = lk;
for ely = 1:nely
for elx = 1:nelx
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely;
Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
c = c + x(ely,elx)^penal*Ue'*KE*Ue;
dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
end
end
% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change)]);

```



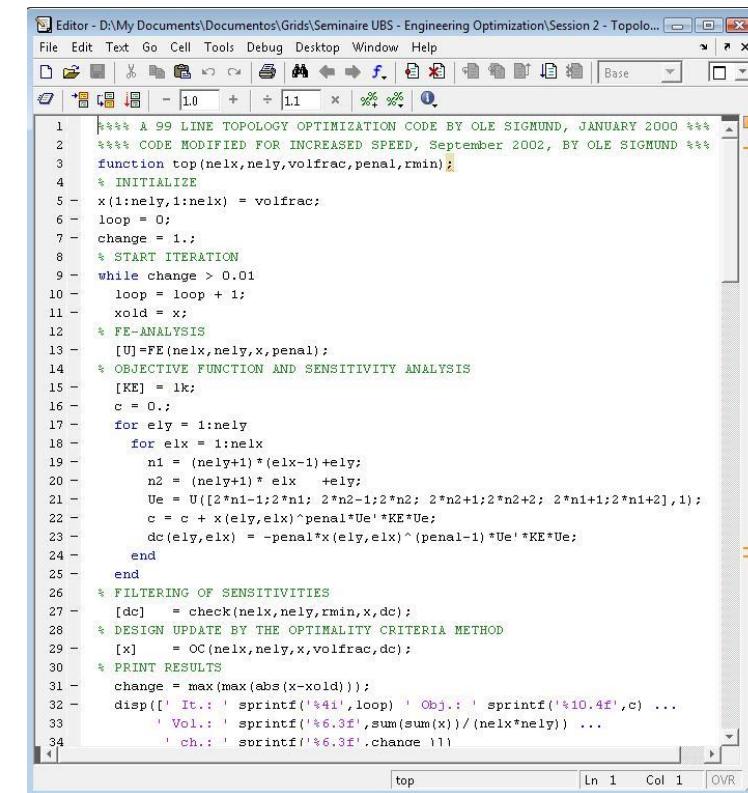
Material de suporte

top.m

1) Abrir o código **topopt.m** com **MatLab**,
Mathematica, Scilab, **Octave**, Rlab ou
equivalente

2) Executar o seguinte comando:

topopt(60,20,0.5,3.0,1.5)

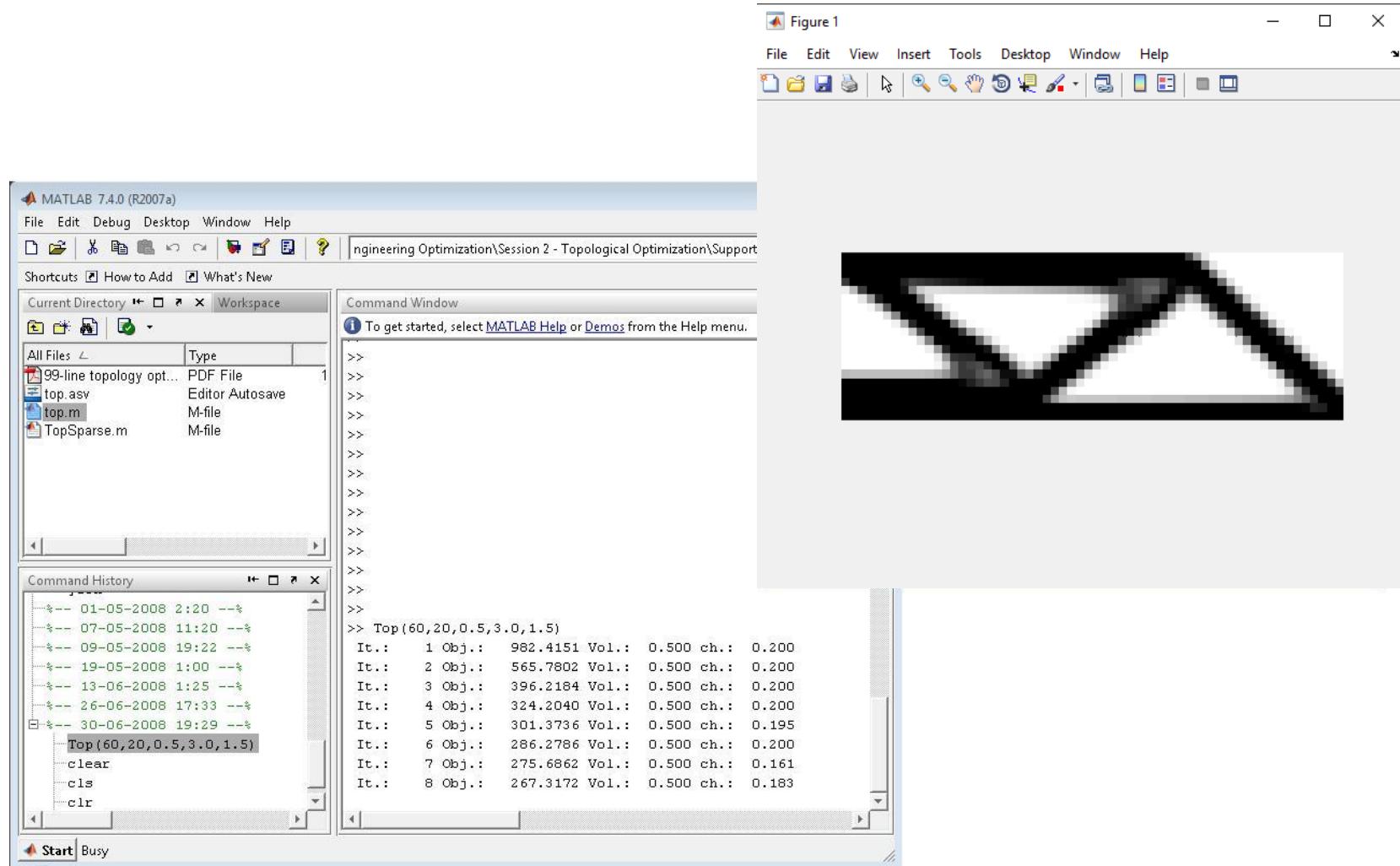


```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;

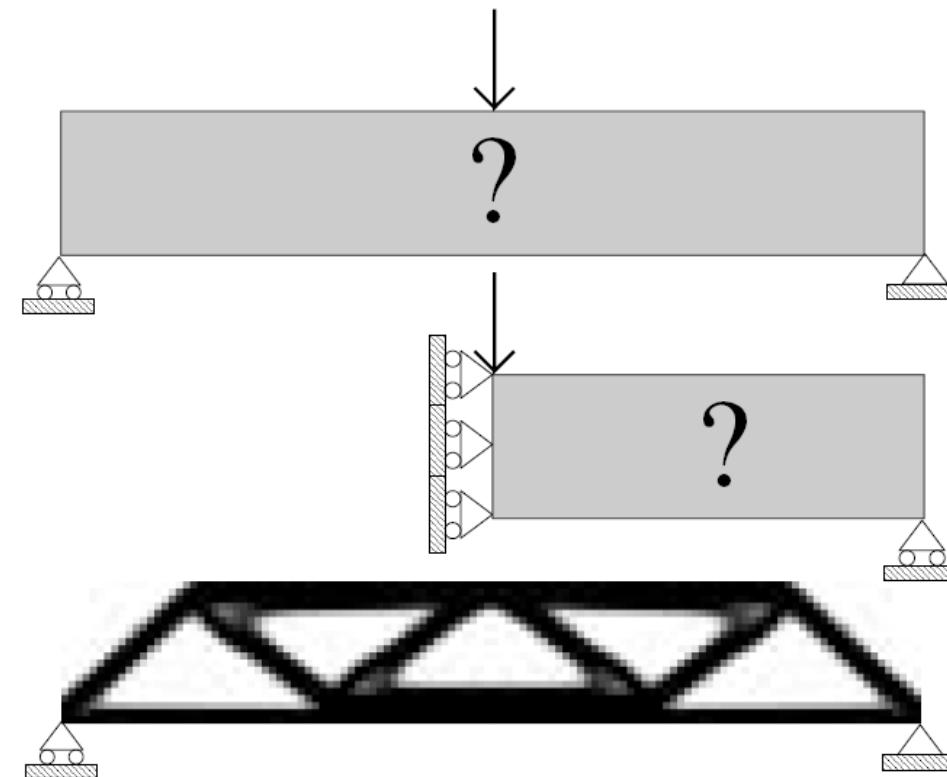
% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
c = 0.;

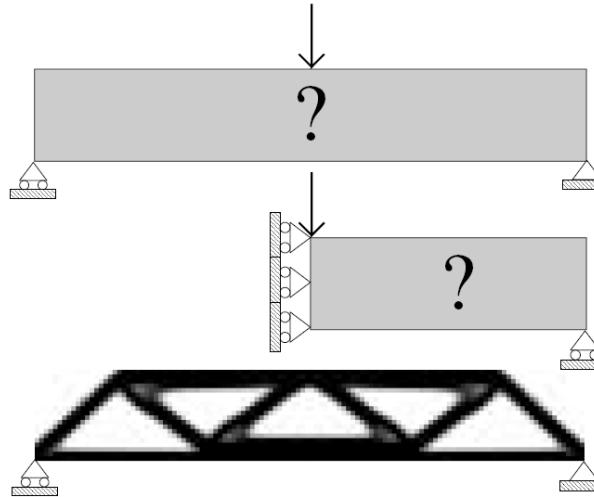
% FE-ANALYSIS
[U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[KE] = lk;
for ely = 1:nely
for elx = 1:nelx
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely;
Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
c = c + x(ely,elx)^penal*Ue'*KE*Ue;
dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
end
end
% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change)]);

```



Está a resolver este problema:



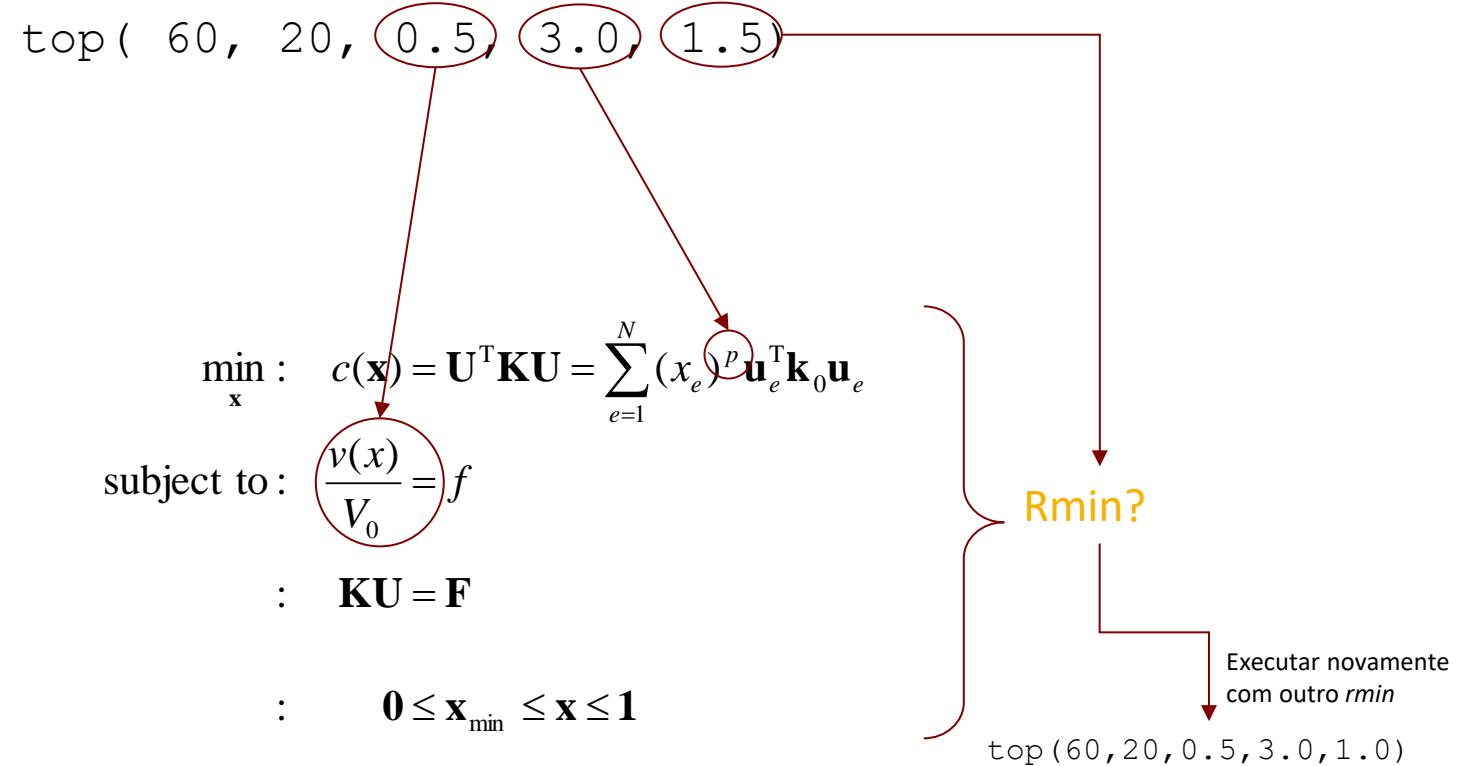


O comando

`top(60,20,0.5,3.0,1.5)`

é equivalente a

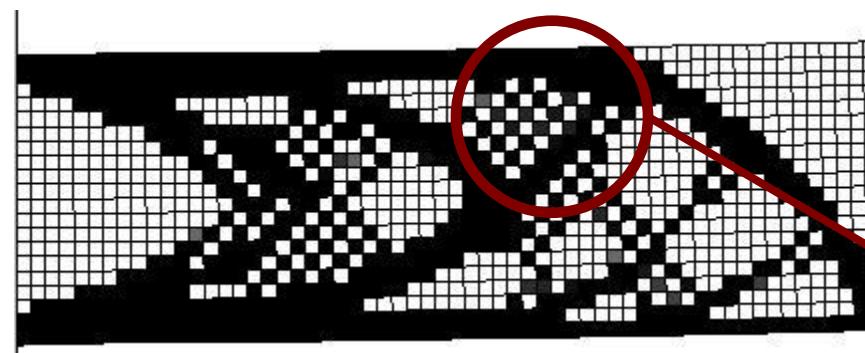
`top(nelx,nely,volfrac,penal,rmin)`



O comando

`top(60, 20, 0.5, 3.0, 1.0)`

resulta em

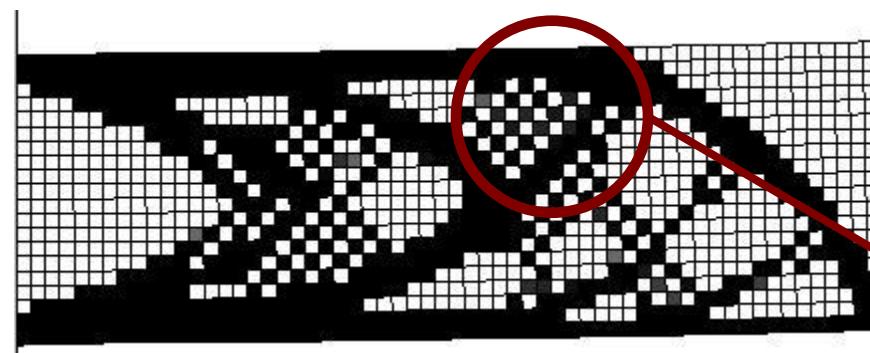


Instabilidade
numérica!
> **Checkerboard**

O comando

`top(60, 20, 0.5, 3.0, 1.0)`

resulta em



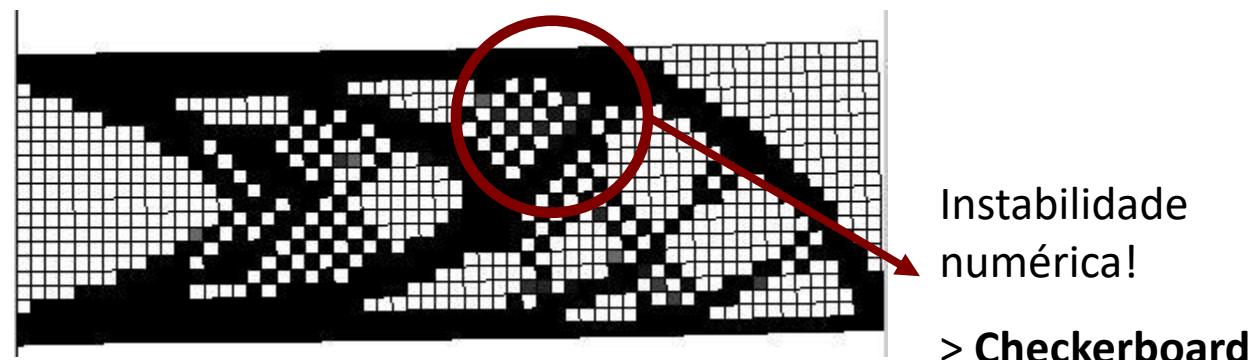
Instabilidade
numérica!
> Checkerboard

- > mínimos locais?
- > problema convexo?
- > métodos de gradiente...

O comando

`top(60, 20, 0.5, 3.0, 1.0)`

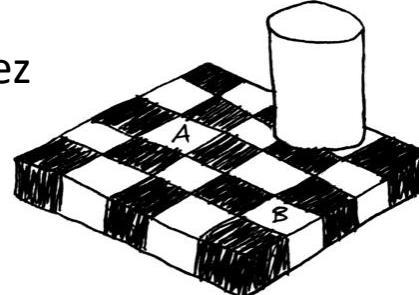
resulta em



Por que razão pode este padrão ser mais rígido do que uma solução homogénea de cinzentos?

Este problema resulta de um fenómeno de rigidez excessiva desta configuração

> recurso a filtros de sensibilidade



este tipo de filtro de independência de malha pode ser implementado alterando as sensibilidades como

$$\widehat{\frac{\partial c}{\partial x_e}} = \frac{1}{x_e \sum_{f=1}^N \hat{H}_f} \sum_{f=1}^N \hat{H}_f x_f \frac{\partial c}{\partial x_f}.$$

Operador de convolução (peso) H_f pode escrever-se como

$$\hat{H}_f = r_{\min} - \text{dist}(e, f),$$

$$\{f \in N \mid \text{dist}(e, f) \leq r_{\min}\}, \quad e = 1, \dots, N,$$

onde o operador $\text{dist}(e, f)$ é definido como a distância entre os centros dos elementos e e f

este tipo de filtro de independência de malha pode ser implementado alterando as sensibilidades como

$$\widehat{\frac{\partial c}{\partial x_e}} = \frac{1}{x_e \sum_{f=1}^N \hat{H}_f} \sum_{f=1}^N \hat{H}_f x_f \frac{\partial c}{\partial x_f}.$$

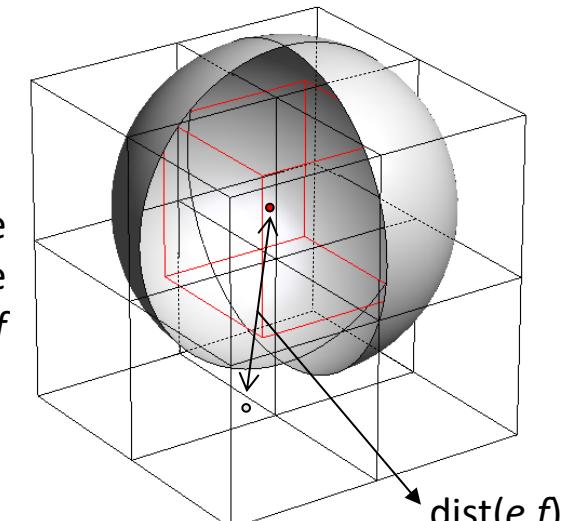


$$\hat{H}_f = r_{\min} - \text{dist}(e, f),$$

$\text{dist}(e, f)$ é a distância entre os centros dos elementos e e f

H_f é zero para a da região de influência do filtro

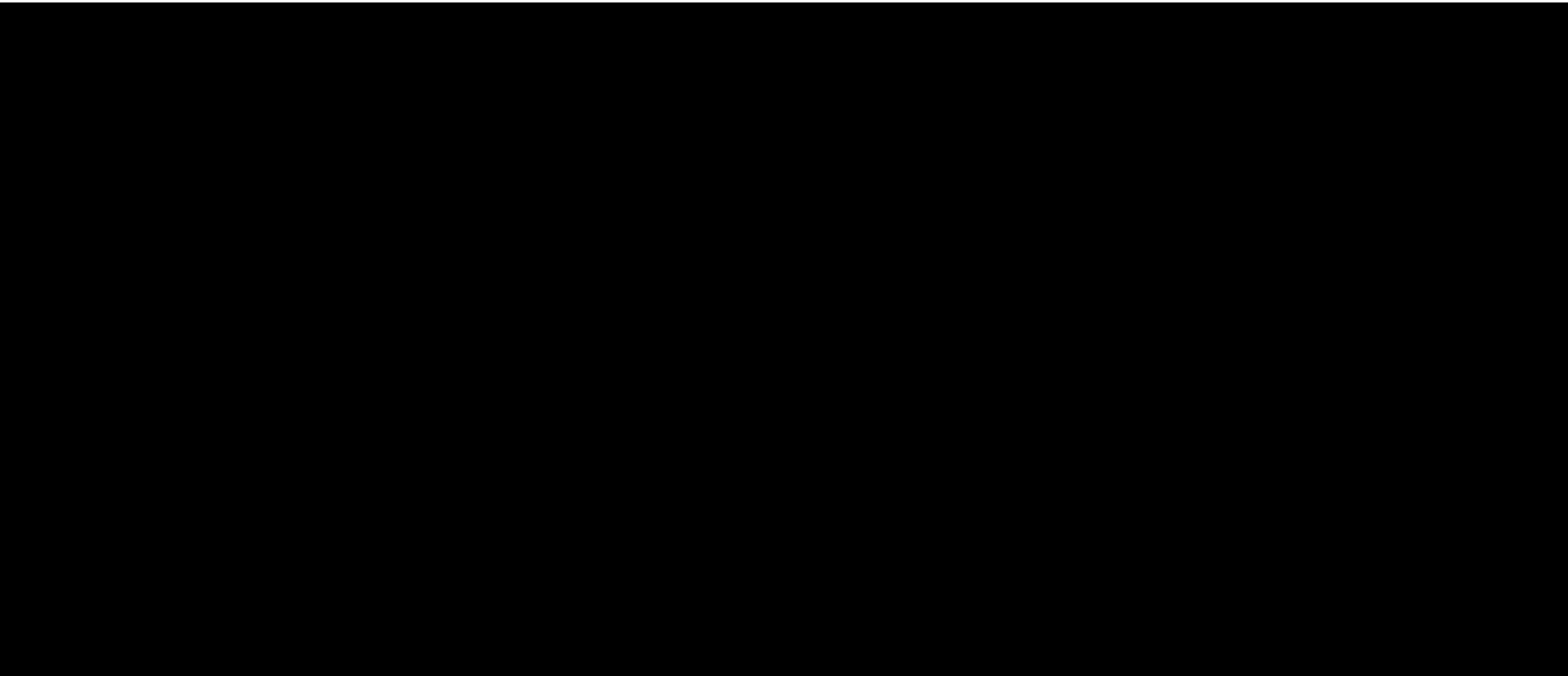
Nesta forma, a superfície de ponderação é de decaimento linear



Análise do código top.m



top.m



Análise do código top.m



top.m

```
%%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE %%%%%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
    loop = loop + 1;
    xold = x;
    % FE-ANALYSIS
    [U]=FE(nelx,nely,x,penal);
```

Análise do código top.m



top.m

```
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[KE] = 1k;
c = 0.;
for ely = 1:nely
    for elx = 1:nelx
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2;
        2*n1+1;2*n1+2],1);
        c = c + x(ely,elx)^penal*Ue'*KE*Ue;
        dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
    end
end
% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('%4i',loop) ' Obj.: '
sprintf('%10.4f',c) ...
' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change)])
```

Análise do código top.m



top.m

```
% PLOT DENSITIES
% colormap(gray); imagesc(-x); axis equal; axis tight; axis
off;pause(1e-6);
colormap(gray); axis equal;
for ely = 1:nely
for elx = 1:nelx
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)* elx +ely;
Ue = 0.005*U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2;
2*n1+1;2*n1+2],1);
ly = ely-1; lx = elx-1;
xx = [Ue(1,1)+lx Ue(3,1)+lx+1 Ue(5,1)+lx+1 Ue(7,1)+lx ]';
yy = [-Ue(2,1)-ly -Ue(4,1)-ly -Ue(6,1)-ly-1 -Ue(8,1)-ly-1]';
patch(xx,yy,-x(ely,elx))
end
end
drawnow;clf;
end
drawnow;
```

Análise do código top.m



top.m

```
%%%%% OPTIMALITY CRITERIA UPDATE %%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    xnew = max(0.001,max(x-
move,min(1.,min(x+move,x.*sqrt(-dc./lmid))))));
    if sum(sum(xnew)) - volfrac*nelx*nely > 0;
        l1 = lmid;
    else
        l2 = lmid;
    end
end
```

Análise do código top.m



top.m

```
%%%%% MESH-INDEPENDENCY FILTER %%%%%%
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
    for j = 1:nely
        sum=0.0;
        for k = max(i-
floor(rmin),1):min(i+floor(rmin),nelx)
            for l = max(j-
floor(rmin),1):min(j+floor(rmin),nely)
                fac = rmin-sqrt((i-k)^2+(j-l)^2);
                sum = sum+max(0,fac);
                dcn(j,i) = dcn(j,i) +
max(0,fac)*x(l,k)*dc(l,k);
            end
        end
        dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
    end
end
```

Análise do código top.m



top.m

```
%%%%% FE-ANALYSIS %%%%%%
function [U]=FE(nelx,nely,x,penal)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1),
2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1),1); U =
zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;
2*n2+2; 2*n1+1; 2*n1+2];
        K(edof,edof) = K(edof,edof) +
x(ely,elx)^penal*KE;
    end
end
```

Análise do código top.m



top.m

```
%%%%%% FE-ANALYSIS %%%%%%
%
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
%
F(2,1) = -1;
%
fixeddofs =
union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
%
% F(2*(nelx+1)*(nely+1),1)=-1;
%
% fixeddofs = [1:2*(nely+1)];
%
%
alldofs = [1:2*(nely+1)*(nelx+1)];
%
freedofs = setdiff(alldofs,fixeddofs);
%
% SOLVING
%
U(freedofs,:) = K(freedofs, freedofs) \
F(freedofs,:);
%
U(fixeddofs,:)= 0;
```

Análise do código top.m



top.m

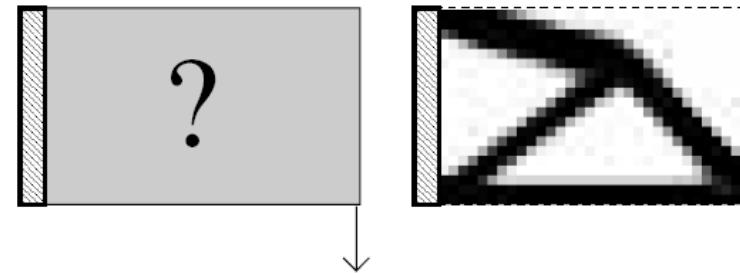
```
%%%%% ELEMENT STIFFNESS MATRIX %%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6    1/8+nu/8  -1/4-nu/12 -1/8+3*nu/8 ...
     -1/4+nu/12 -1/8-nu/8  nu/6          1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1)  k(2)  k(3)  k(4)  k(5)  k(6)  k(7)  k(8)
                  k(2)  k(1)  k(8)  k(7)  k(6)  k(5)  k(4)  k(3)
                  k(3)  k(8)  k(1)  k(6)  k(7)  k(4)  k(5)  k(2)
                  k(4)  k(7)  k(6)  k(1)  k(8)  k(3)  k(2)  k(5)
                  k(5)  k(6)  k(7)  k(8)  k(1)  k(2)  k(3)  k(4)
                  k(6)  k(5)  k(4)  k(3)  k(2)  k(1)  k(8)  k(7)
                  k(7)  k(4)  k(5)  k(2)  k(3)  k(8)  k(1)  k(6)
                  k(8)  k(3)  k(2)  k(5)  k(4)  k(7)  k(6)  k(1)];
```

Alterar o código top.m

para obter o seguinte problema:



top.m



Para resolver este exemplo, alterar as linhas 79 e 80 para

```
79 F(2*(nelx+1)*(nely+1),1) = -1;  
80 fixeddofs = [1:2*(nely+1)];
```

O comando passa a ser

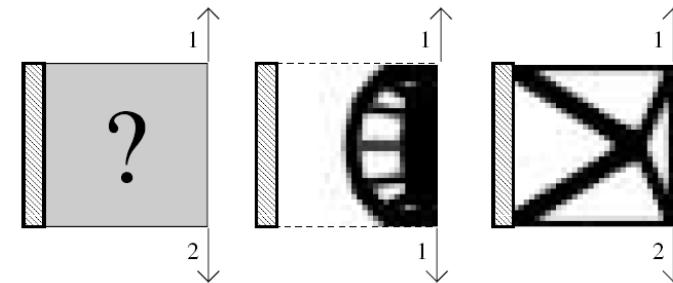
```
top(32,20,0.4,3.0,1.2)
```

Alterar o código top.m

Problema multicarregamento?



top.m



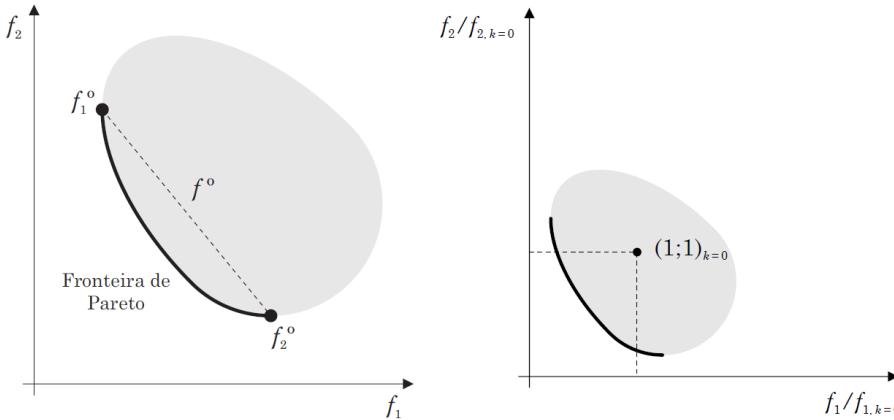
Funciona?

```
69 F = sparse(2*(nely+1)*(nelx+1),2);  
U = sparse(2*(nely+1)*(nelx+1),2);  
  
79 F(2*(nelx+1)*(nely+1),1) = -1.;  
F(2*(nelx)*(nely+1)+2,1) = 1.;
```

E usar o comando

```
top(30,30,0.4,3.0,1.2).
```

Otimização multiobjetivo



minimizar $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]$
 sujeito a $\mathbf{x} \in S$.



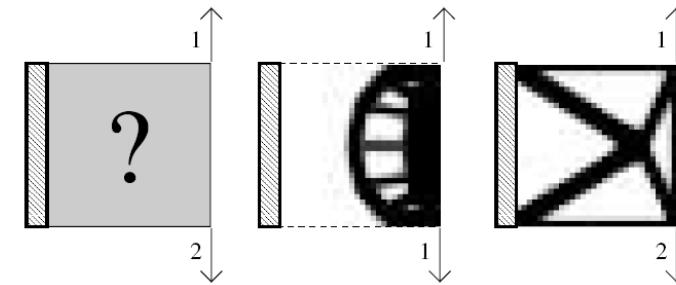
minimizar $\mathbf{f}(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \dots + w_m f_m(\mathbf{x})$,
 sujeito a $\mathbf{x} \in S$,

Alterar o código top.m

Problema multicarregamento?



top.m



testar 1

Soma de carregamentos?

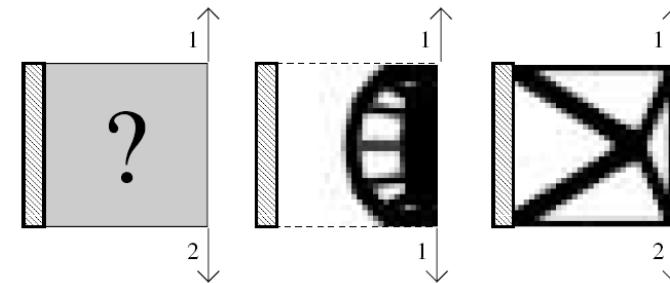
```
69 F = sparse(2*(nely+1)*(nelx+1),2);  
U = sparse(2*(nely+1)*(nelx+1),2);  
  
79 F(2*(nelx+1)*(nely+1),1) = -1.;  
F(2*(nelx)*(nely+1)+2,1) = 1.;
```

E usar o comando

`top(30,30,0.4,3.0,1.2).`

Alterar o código top.m

Problema multicarregamento:



```

19b dc(ely,elx) = 0.;
19c for i = 1:2
20   Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2;
              2*n2+1;2*n2+2;2*n1+1;2*n1+2] ,i);
21   c = c + x(ely,elx)^penal*Ue'*KE*Ue;
22   dc(ely,elx) = dc(ely,elx) -
              penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
22b end

```

```

69 F = sparse(2*(nely+1)*(nelx+1),2);
U = sparse(2*(nely+1)*(nelx+1),2);

79 F(2*(nelx+1)*(nely+1),1) = -1. ;
F(2*(nelx)*(nely+1)+2,2) = 1. ;

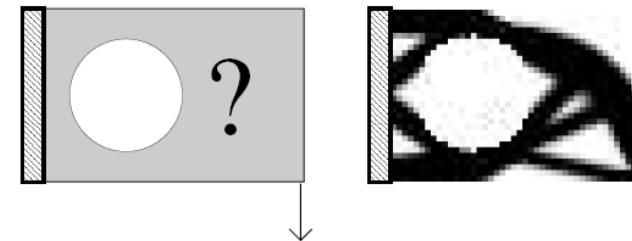
```

E usar o comando

`top(30,30,0.4,3.0,1.2).`

Alterar o código top.m

Elementos passivos:



top.m

Alterar

42b xnew(find(passive)) = 0.001; **Após lin. 4**

O input é

top(45,30,0.5,3.0,1.5),

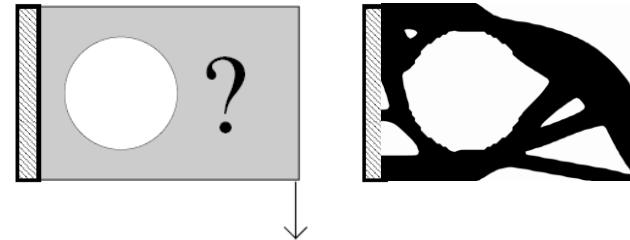
{

```
for ely = 1:nely
    for elx = 1:nelx
        if sqrt((ely-nely/2.)^2+(elx-nelx/3.)^2) <
            nely/3.

            passive(ely,elx) = 1;
            x(ely,elx) = 0.001;
        else
            passive(ely,elx) = 0;
        end
    end
end
```

Alterar o código top.m

E suavização? `Contourf`?



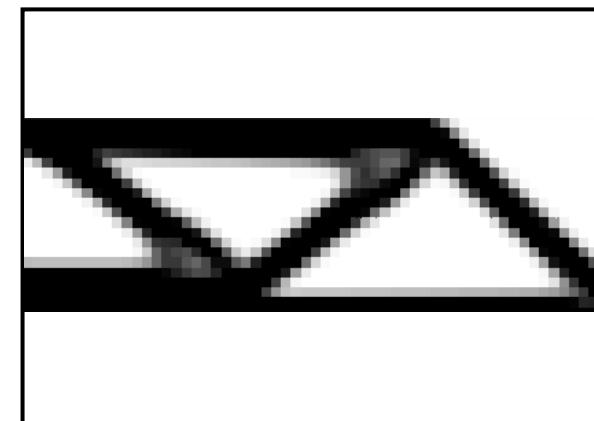
`top.m`

Alterar o código top.m

E... como seria com um domínio menos restringido?



top.m



- . Testar **topplot** e **topTH**
- . Cria o teu próprio problema

Analisar a influência dos parâmetros *nelx*,
nely, *volfrac*, *penal* e *rmin* nas soluções



Labwork: software comercial



AUTODESK®
FUSION 360™



Novembro 2024

