

# **PEF 5710 - Otimização Estrutural**

## **1º Quadrimestre 2021**

### **Semana 09**

### **Otimização Topológica**

**Prof. Marcelo Araujo da Silva**  
**marcelo.araujo@ufabc.edu.br**

**Professor Titular:**  
**Dr. Reyolando M. L. R. F. Brasil**

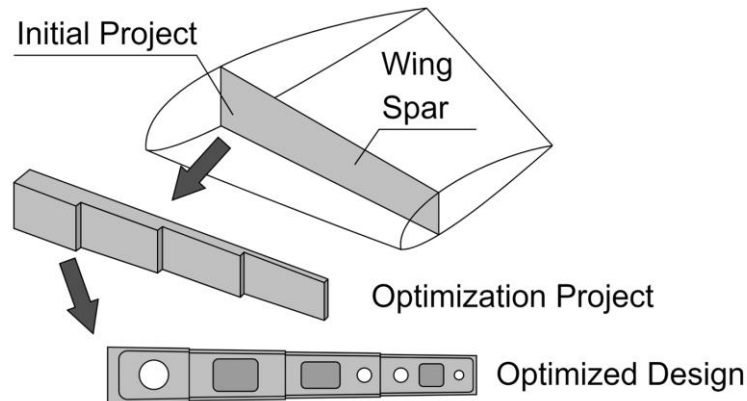


# Otimização Topológica

## Introdução

Em otimização topológica, o objetivo é gerar a melhor distribuição de material a partir de um espaço de projeto fixo. Esse procedimento é exemplificado na *Figura 1*.

Figura 1 - Processo de otimização topológica da longarina da asa de uma aeronave

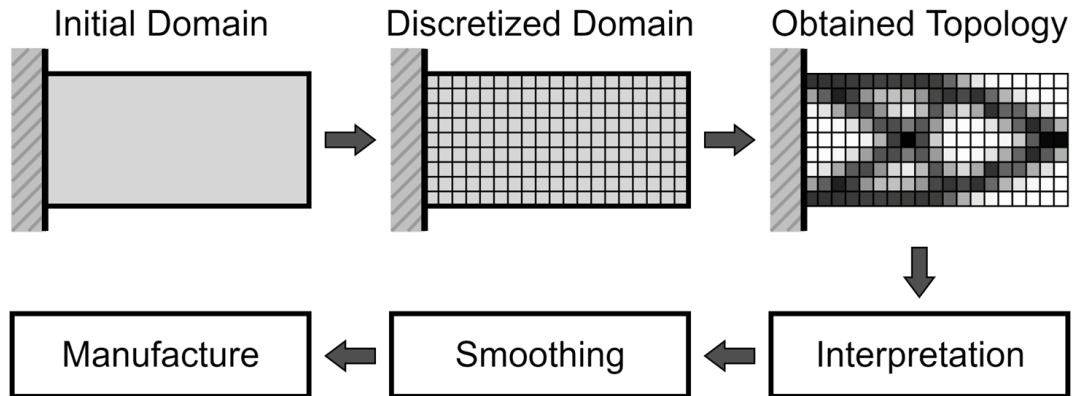


Fonte: Elaborado pelos autores

## Introdução

O procedimento é resumido na *Figura 2*, que ilustra o uso de MEF em conjunto com um algoritmo de otimização.

Figura 2 - Procedimento de projeto de otimização topológica



Fonte: Elaborado pelos autores

## MEF em Otimização Topológica

Os deslocamentos de um ponto em um sólido contínuo são modelados por um vetor  $\mathbf{u}$ . No caso de uma placa, em um domínio de eixos  $x$  e  $y$ , tem-se um vetor  $2 \times 1$ .

$$\mathbf{u} = \begin{Bmatrix} u(x, y, t) \\ v(x, y, t) \end{Bmatrix}.$$

A partir dos deslocamentos, as deformações são obtidas com a aplicação do operador diferencial  $\mathbf{D}$ :

$$\boldsymbol{\varepsilon} = \mathbf{D}\mathbf{u},$$

## MEF em Otimização Topológica

então, no caso de uma placa, tem-se o seguinte operador 3 x 2:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \mathbf{D}\mathbf{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix}.$$

O passo seguinte é a obtenção do vetor das tensões, a partir das deformações, usando, simplificadamente, a lei de Hooke, na forma matricial:

$$\boldsymbol{\sigma} = \mathbf{E}\boldsymbol{\varepsilon} = \mathbf{E}\mathbf{D}\mathbf{u}.$$

## MEF em Otimização Topológica

No caso de uma placa, o vetor das tensões é  $3 \times 1$ :

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix},$$

no estado plano de tensão, a lei de Hooke é expressa pela seguinte matriz  $3 \times 3$ :

$$\mathbf{E} = \frac{E_m}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix},$$

## MEF em Otimização Topológica

onde  $\nu$  ( $\nu$ ) é o coeficiente de Poisson e  $E_m$  é o módulo de elasticidade. Também é preciso obter as tensões máxima e mínima, dadas pela equação:

$$\sigma_{1,2} = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x + \sigma_y}{2}\right)^2 + \tau_{xy}^2}.$$

Também é necessário encontrar a tensão de cisalhamento máxima, dada por:

$$\tau_{max} = \sqrt{\left(\frac{\sigma_x + \sigma_y}{2}\right)^2 + \tau_{xy}^2}.$$



## O Algoritmo KNITRO

Um problema de otimização pode ser definido como: Determine  $\mathbf{x} \in \mathfrak{R}^n$  que minimize a função  $f(\mathbf{x})$  sujeita à

$$\text{Restrição de Igualdade: } g_j(\mathbf{x}) = 0; j = 1, l$$

$$\text{Restrição de Desigualdade: } g_j(\mathbf{x}) \leq 0; j = l + 1, m$$

A função Lagrangiana do problema é definida como

$$\Lambda(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}),$$

onde  $\mathbf{u} \in \mathfrak{R}^m$  é o vetor dos multiplicadores de Lagrange.



## O Algoritmo KNITRO

Neste caso, as variáveis de projeto são o vetor espessura  $v_t$ , representado anteriormente pelo vetor  $\mathbf{x}$  e o número de divisões nas direções  $x$  e  $y$ .

$$v_{t(k,1)} = [v_{t_1}, v_{t_2}, \dots], k = 1, \dots, nel$$

As restrições, aqui, são desigualdades não lineares e se referem à tensão admissível do material,  $\sigma_{adm} = 225 \times 10^6 \text{ N}$ , e à espessura mínima  $v_{tmin} = 0,0001 \text{ m}$ .

$$g_1 = \sigma_{max} \leq \sigma_{adm} \Rightarrow g_1 = \sigma_{max} - \sigma_{adm};$$

$$g_2 = v_t \geq v_{tmin} \Rightarrow g_2 = v_{tmin} - v_t.$$

## O Algoritmo KNITRO

Entende-se que, ao otimizar o volume, ou seja, minimizando-o, todo o projeto é otimizado. Logo, o volume ( $Vol$ ) será a função objetivo:

$$Vol = Area * v_t,$$

onde  $v_t$ , lembre-se, é a espessura da placa e a  $Area$  é dada por:

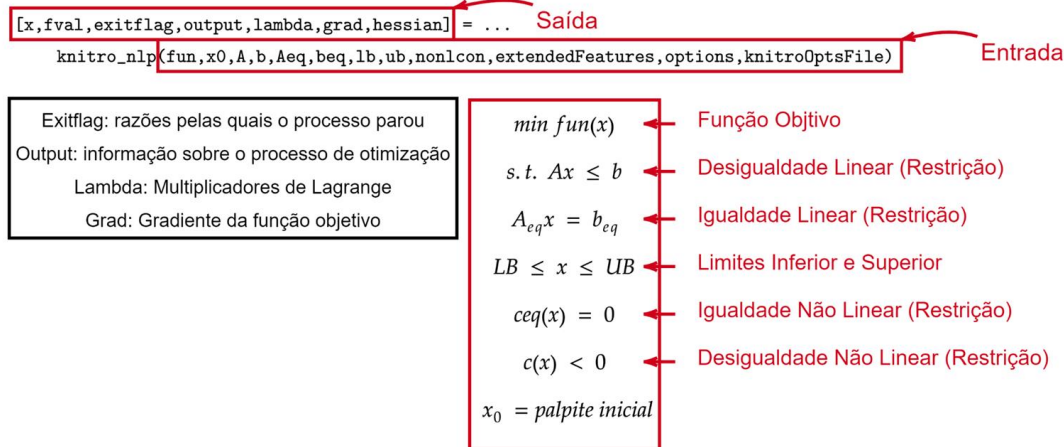
$$Area_{(1,k)} = \begin{bmatrix} dx_1 * dy_1 \\ \dots \end{bmatrix}, k = 1, \dots, nel$$

Para resolver o problema, o algoritmo **KNITRO** será empregado, uma vez que o mesmo tem uma interface fácil de ser utilizada no MATLAB.

## O Algoritmo KNITRO

Na *Figura 3* é ilustrado os argumentos do algoritmo.

Figura 3 - Argumentos da função KNITRO no MATLAB



Fonte: Elaborado pelos autores

## O Algoritmo KNITRO

As estruturas foram discretizadas 2 e 3 vezes as dimensões originais, com  $v_t = 0,1 m$ .

Figura 4 - Placa engastada com carga concentrada na ponta

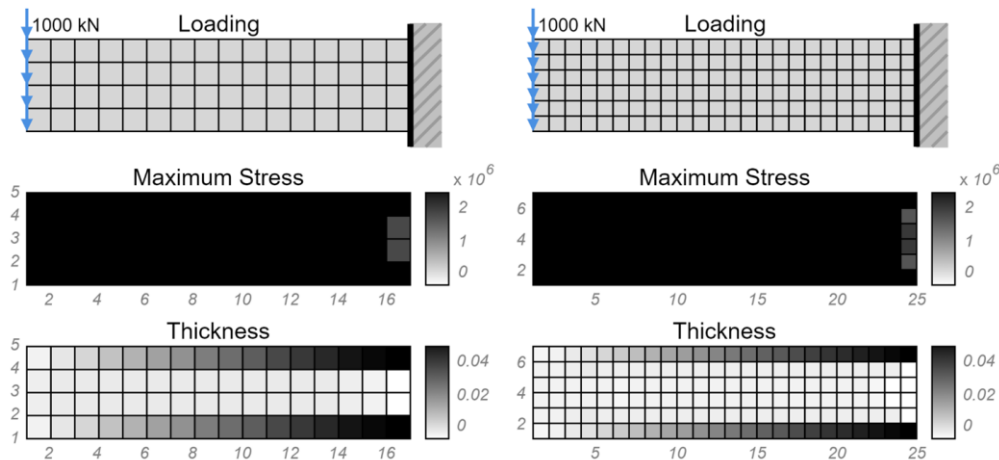


Tabela 1 - Placa engastada com carga concentrada na ponta

Dimensões	8 m x 2 m	
Volume Inicial	1,6 m <sup>3</sup>	
Discretização	x2	x3
Volume Final	0,201368 m <sup>3</sup>	0,194917 m <sup>3</sup>
Redução	87,41%	87,82%

Fonte: Elaborado pelos autores

## O Algoritmo KNITRO

As estruturas foram discretizadas 2 e 3 vezes as dimensões originais, com  $v_t = 0,1 m$ .

Figura 5 - Placa biapoada com carga concentrada no lado

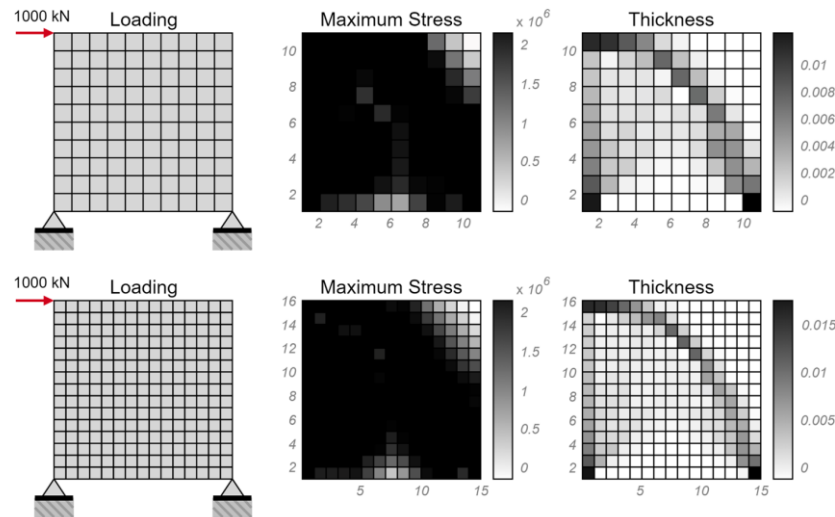


Tabela 2 - Placa biapoada com carga concentrada no lado

Dimensões	5 m x 5 m	
Volume Inicial	2,5 m <sup>3</sup>	
Discretização	x2	x3
Volume Final	0,055845 m <sup>3</sup>	0,056045 m <sup>3</sup>
Redução	97,77%	97,76%

Fonte: Elaborado pelos autores

## O Algoritmo KNITRO

As estruturas foram discretizadas 2 e 3 vezes as dimensões originais, com  $v_t = 0,1 m$ .

Figura 6 - Placa biapojada com carregamento uniforme

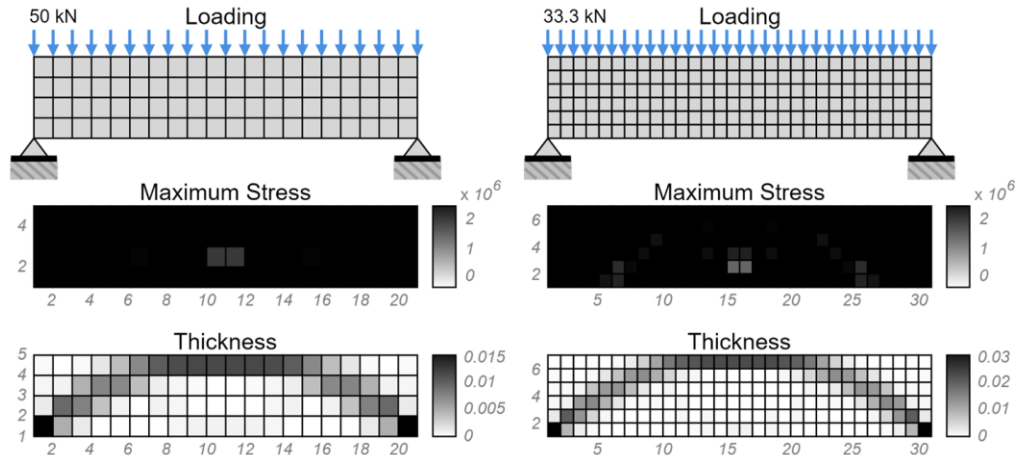


Tabela 3 - Placa biapojada com carregamento uniforme

Dimensões	10 m x 2 m	
Volume Inicial	2 m <sup>3</sup>	
Discretização	x2	x3
Volume Final	0,072808 m <sup>3</sup>	0,108060 m <sup>3</sup>
Redução	96,36%	94,60%

Fonte: Elaborado pelos autores

## O Algoritmo KNITRO

As estruturas foram discretizadas 2 e 3 vezes as dimensões originais, com  $v_t = 0,1 m$ .

Figura 7 - Placa biapoada com carregamento concentrado

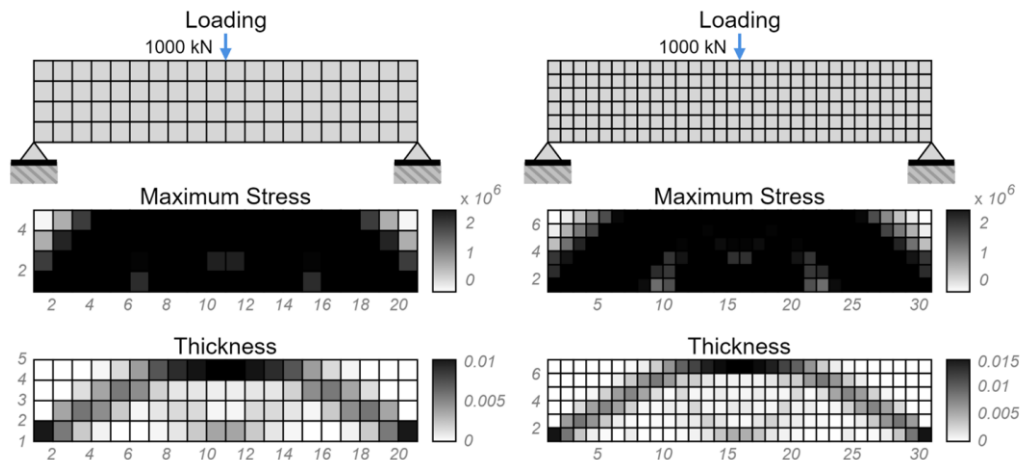


Tabela 4 - Placa biapoada com carregamento concentrado

Dimensões	10 m x 2 m	
Volume Inicial	2 m <sup>3</sup>	
Discretização	x2	x3
Volume Final	0,061673 m <sup>3</sup>	0,061700 m <sup>3</sup>
Redução	96,92%	96,91%

Fonte: Elaborado pelos autores



## Confiabilidade

Com os resultados, é possível analisar a *confiabilidade estrutural*. Em nosso caso, tome a equação abaixo:

$$M = m_{el} - m_e,$$

onde  $m_{el}$  é a massa limite do projeto e  $m_e$  é a massa obtida no processo de otimização topológica. Perceba que a estrutura apresenta uma massa maior que a limite, “falha”, se:

$$M < 0.$$

Esse procedimento será repetido  $n$  vezes e, assim, é possível obter a média  $\bar{M}$  e o desvio padrão  $\hat{M}$ .

## Confiabilidade

Com essas informações, é possível determinar o *índice de confiabilidade*

$$\beta = \frac{\bar{M}}{\widehat{M}}.$$

Baseada no índice de confiabilidade, a *probabilidade de falha*, quando todas as distribuições são normais, é calculada por

$$P = \Phi(-\beta),$$

onde  $\Phi$  é *função cumulativa normal padrão* (CDF), dada por

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{z^2}{2}} dz.$$

## Confiabilidade

### **Processo de Monte Carlo - Gerar Números Aleatórios no MATLAB**

O Processo de Monte Carlo é baseado na geração de números aleatórios.

Para gerar números aleatórios da *distribuição normal* no MATLAB, pode-se utilizar os comandos:

- $normrnd(\bar{X}, \hat{X})$  - para uma média  $\bar{X}$  e desvio padrão  $\hat{X}$ ;
- $randn()$  - para a normal padrão.

A variável aleatória do problema, no caso, é a tensão admissível  $\sigma_{adm}$ , cuja média e desvio padrão são, respectivamente:

$$\bar{\sigma}_{adm} = 260 \text{ MPa} \quad \text{e} \quad \hat{\sigma}_{adm} = 26 \text{ MPa}.$$

## Confiabilidade

### ***Processo de Monte Carlo - Gerar Números Aleatórios no Excel***

Para gerar números aleatórios da *distribuição normal* usando o *Processo de Monte Carlo* no *Excel* usa-se a função:

$$= INV.NORM.N(ALEATÓRIO(), \bar{X}, \hat{X}),$$

onde:

- *INV.NORM.N()* - retorna o inverso da *CDF* normal para  $\bar{X}$  e  $\hat{X}$ ;
- *ALEATÓRIO()* - retorna um número aleatório real  $\geq 0$  e  $< 1$ .

## Confiabilidade

A *Tabela 5* apresenta, de forma resumida, o *Processo de Monte Carlo* para um número de avaliações  $N$ .

Tabela 5 - Resumo do *Processo de Monte Carlo*

$i$	$X_1$	$X_2$	...	$X_n$	$M$	$M(X) \leq 0$
<b>1</b>	$x_1$	$x_2$	...	$x_n$	$M_1$	0 ou 1
<b>2</b>	$x_1$	$x_2$	...	$x_n$	$M_2$	0 ou 1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
<b><math>N</math></b>	$x_1$	$x_2$	...	$x_n$	$M_N$	0 ou 1

## Confiabilidade

Pela definição, pode-se calcular a *probabilidade de falha* pela equação

$$P_f \approx \frac{n(M(\mathbf{X}) \leq 0)}{N}. \quad \text{O índice de confiabilidade por } \beta = \frac{\overline{M}}{\widehat{M}}.$$

Cem iterações foram realizadas para os processos. Utilizando o valor final do volume e a densidade do material, é possível obter a massa limite, inicialmente aumentada em 10% do valor de base:

$$m_{el} = (Vol * 7800) * (1 + 10\%).$$

## Confiabilidade

A *Tabela 6* apresenta os resultados, para os dois métodos empregados no processo, no caso da placa engastada com carga concentrada na ponta.

Tabela 6 - Placa engastada com carga concentrada na ponta

Processo de Monte Carlo		
Discretização	x2	x3
$\beta$	2,6553	2,7288
$P$	0,39614%	0,31871%
Método Clássico		
Discretização	x2	x3
$P_f$	1%	2%
Runtime	0,265 horas	2,403 horas

## Confiabilidade

A *Tabela 7* apresenta os resultados, para os dois métodos empregados no processo, no caso da placa biapoiada com carga concentrada no lado.

Tabela 7 - Placa biapoiada com carga concentrada no lado

Processo de Monte Carlo		
Discretização	x2	x3
$\beta$	2,5092	2,9796
$P$	0,60507%	0,14430%
Método Clássico		
Discretização	x2	x3
$P_f$	2%	0%
Runtime	2,075 horas	27,9 horas





## Confiabilidade

A *Tabela 8* apresenta os resultados, para os dois métodos empregados no processo, no caso da placa biapoiada com carregamento uniforme.

Tabela 8 - Placa biapoiada com carregamento uniforme

Processo de Monte Carlo		
Discretização	x2	x3
$\beta$	2,844	2,674
$P$	0,22273%	0,37477%
Método Clássico		
Discretização	x2	x3
$P_f$	0%	1%
Runtime	0,576 horas	12,24 horas

## Confiabilidade

A *Tabela 9* apresenta os resultados, para os dois métodos empregados no processo, para o caso da placa biapoiada com carregamento concentrado.

Tabela 9 - Placa biapoiada com carregamento concentrado

Processo de Monte Carlo		
Discretização	x2	x3
$\beta$	2,5157	2,6779
$P$	0,59400%	0,37040%
Método Clássico		
Discretização	x2	x3
$P_f$	1%	1%
Runtime	0,485 horas	13,89 horas

## Confiabilidade

Para mais conteúdo a respeito do Processo de Monte Carlo, você pode assistir a aula 3 da disciplina de *Confiabilidade de Componentes e Sistemas* ministrada no 1º Quadrimestre de 2021 na Universidade Federal do ABC no link abaixo.

PROCESSO DE MONTE CARLO

Recordando Variáveis Aleatórias

A Figura 1 ilustra um parafuso típico, suas dimensões podem ser entendidas como variáveis aleatórias em razão de imperfeições no processo de fabricação etc.

Figura 1 - Esquema de um parafuso típico.

$D(X_j)$   $L(X_j)$

Fonte: Elaborado pelos autores.

Onde:

- $L(X_j)$  - Variável aleatória associada ao comprimento;
- $D(X_j)$  - Variável aleatória associada ao diâmetro.

UMABC 03

Link: <https://www.youtube.com/watch?v=UMRIp1bySWY>

**Dúvidas?**



**Obrigado!**