

20 de junho de 2014

OTIMIZAÇÃO ESTRUTURAL DE TRELIÇAS COM RESTRIÇÕES DE FLAMBAGEM E DE ESCOAMENTO DE MATERIAIS

Daniel Santhiago de Souza



OTIMIZAÇÃO ESTRUTURAL DE TRELIÇAS COM RESTRIÇÕES DE FLAMBAGEM E DE ESCOAMENTO DE MATERIAIS

Aluno: Daniel Santhiago de Souza

Orientador: Ivan Fabio Mota de Menezes

Trabalho apresentado como requisito parcial à conclusão do curso de Engenharia de Controle e Automação na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil



Agradecimentos

Ao meu orientador Ivan Menezes, pela atenção e paciência a cada reunião, sem ele a execução deste projeto não seria possível.

A coordenação das Engenharias e a todos os professores que me ajudaram nestes anos da graduação.

A minha família, meu pai minha mãe e minhas tias por toda a criação e educação que me deram até hoje.

Aos meus colegas da PUC, Fernando, Thiago, André, James e Vitor pelo apoio e companheirismo nesta longa jornada de engenharia.

Aos meus amigos Cristiano, Pedro Junqueira, Pedro Farias, Rodrigo, José Leonardo, Álvaro, Fabio, Pedro Arlen, Daniel e Marcos pela amizade, pelas conversas e todas as saídas.

Ao meu chefe Gutemberg por incentivar o meu estudo e acreditar no potencial do meu trabalho.

A tantas outras pessoas incríveis que conheci principalmente nestes últimos anos e sem as quais não haveria chegado até aqui.

A Deus, por ter me dado a vida e colocado em meu caminho todas as pessoas aqui citadas.

Resumo

Este trabalho consiste na implementação computacional, utilizando o MATLAB, de algoritmos indiretos de otimização não linear, para minimização do peso (volume) de treliças. Serão utilizados os métodos de penalidade e de barreira. O cálculo do equilíbrio estático da treliça será feito pelo método dos elementos finitos e as restrições do problema serão dadas em função das tensões nas barras, as quais devem ser inferiores às tensões de escoamento do material e às tensões críticas de flambagem. Os resultados obtidos comprovam a eficiência da implementação proposta.

Palavras-chave: Métodos de Penalidade; Otimização Estrutural; Treliças.

Structural Optimization of Trusses with buckling and yielding constraints

Abstract

This work consists on the computer implementation using MATLAB, indirect nonlinear optimization algorithms to minimize the weight (volume) of trusses. Methods of penalty and barrier will be used. The calculation of static equilibrium of the truss will be done by means of the finite element method and the constraints of the problem are given according to the tensions in the bars, which should be less than the yield strengths of the material and critical buckling stresses. The results obtained confirm the efficiency of the proposed implementation.

Keywords: Penalty method; Structural Optimization; Truss

Sumário

1.	Introdução.....	1
2.	Método dos Elementos Finitos	2
	a. Introdução.....	2
	b. Dados de entrada.....	2
	a. Matriz de rigidez do elemento.....	4
3.	Otimização de Treliças Planas.....	8
	a. Definição do problema.....	8
	b. Programação não-linear.....	9
	c. Métodos de otimização com restrições.....	9
	i. Método da penalidade.....	10
	ii. Método da barreira.....	10
	d. Métodos de otimização sem restrições	11
	i. Univariante.....	11
	ii. Powell.....	12
	e. Busca linear	12
	i. Método do passo incremental	12
	ii. Método da seção áurea	12
4.	Resultados.....	13
	a. Treliça de 2 barras.....	13
	i. Método da penalidade.....	13
	ii. Método da barreira.....	16
	i. Comparação Fmincon	23
	a. Treliça de 10 barras.....	23
	iii. Método da penalidade.....	24
	iv. Comparação Fmincon	24
5.	Conclusão.....	26
6.	Referências Bibliográficas	27

1. Introdução

A otimização, sob o ponto de vista matemático, consiste na minimização (ou maximização) de uma determinada função, por meio da escolha adequada e sistemática dos valores das variáveis dessa função, dentro de um conjunto viável. A programação matemática é a área da matemática que trata dos problemas de otimização. Neste trabalho serão implementados algoritmos de otimização, utilizando o ambiente MATLAB, para minimização do peso (volume) de treliças, por meio da escolha adequada dos diâmetros de suas barras, respeitando-se as restrições de equilíbrio estático e os limites das tensões de escoamento do material e de flambagem das barras.

2. Método dos Elementos Finitos

a. Introdução

O Método dos Elementos Finitos (MEF) é uma ferramenta numérica que permite a solução de problemas complexos de engenharia, os quais nem sempre apresentam uma solução analítica. No MEF, o domínio do problema é dividido em sub-regiões, denominadas elementos finitos, as quais se conectam entre si por meio dos nós. A solução do problema é obtida utilizando-se uma aproximação da resposta em cada elemento. Neste trabalho, os elementos finitos são as próprias barras das treliças.

b. Dados de entrada

O primeiro passo para o MEF consiste na leitura dos dados de entrada do problema. Normalmente, esses dados são fornecido em arquivos, na forma de tabelas, conforme ilustrado na Figura 2.1.

3						
1	0.0	0.0	0.0	1	1	1
2	60.0	0.0	0.0	1	1	1
3	30.0	10.0	0.0	0	0	1
2						
1	1	3	3.0e07	$\frac{\pi d_1^2}{4}$		
2	3	2	3.0e07	$\frac{\pi d_2^2}{4}$		
1						
3	0.0	-6.6e04	0.0			

Figura 2.1

Os dados da Figura 2.1 representam o mapeamento da treliça de três nós e dois elementos, mostrada na Figura 2.2.

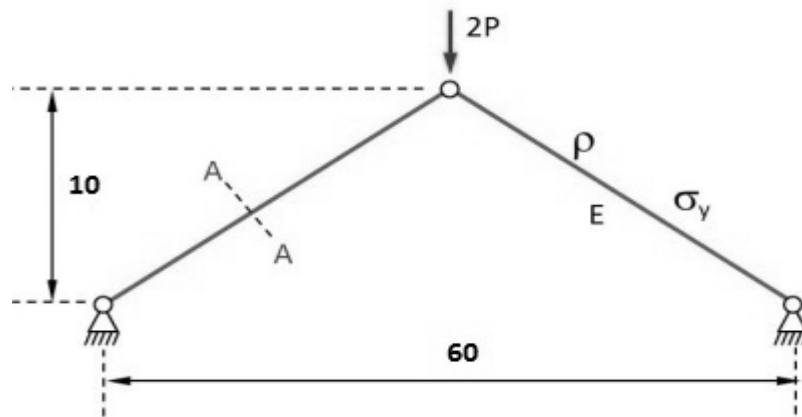


Figura 2.2

A primeira submatriz vista corresponde às coordenadas nodais (nós 1, 2 e 3):

3	X	Y	Z
1	0.0	0.0	0.0
2	60.0	0.0	0.0
3	30.0	10.0	0.0

Figura 2.3

O número em destaque representa a quantidade de nós da estrutura treliçada e, nas linhas seguintes, aparecem as coordenadas X-Y-Z de cada um desses nós.

A seguir, temos a matriz com as incidências das barras, ou seja, os nós inicial e final de cada elemento:

2		
1	1	3
2	3	2

Figura 2.4

Com essas informações é possível calcular o comprimento inicial de cada barra:

$$L_0 = \sqrt{(x_{\text{final}} - x_{\text{inicial}})^2 + (y_{\text{final}} - y_{\text{inicial}})^2 + (z_{\text{final}} - z_{\text{inicial}})^2} \quad (2.1)$$

Em seguida, a matriz com as condições de suporte nodal será utilizada para a definição dos graus de liberdade da estrutura, ou seja, a quantidade de equações do problema:

3		X	Y	Z
1	...	1	1	1
2	...	1	1	1
3	...	0	0	1

Figura 2.5

Em seguida, são fornecidas as informações sobre as propriedades físicas e geométricas dos materiais (no caso de treliças, o módulo de elasticidade e a área de cada barra, fornecida em função do seu respectivo diâmetro):

2			
1	...	3.0e07	$\frac{\pi d_1^2}{4}$
2	...	3.0e07	$\frac{\pi d_2^2}{4}$

Figura 2.6

Finalmente, a matriz de cargas concentradas nodais mostra quais nós possuem alguma carga aplicada e as componentes dessa carga em cada coordenada.

1	X	Y	Z
3	0.0	-6.6e04	0.0

Figura 2.7

a. Matriz de rigidez do elemento

Para o caso de elemento de treliça, é possível determinar uma forma explícita de sua matriz de rigidez, expressa no sistema global de coordenadas. Por simplicidade, vamos derivar a expressão dessa matriz para o caso plano (2D), porém, a extensão para 3D é igualmente simples. Vamos considerar um eixo s , coincidente com o eixo centroidal do elemento. O deslocamento ao longo do eixo da barra é dado por u_s , sendo $u_{s\text{final}}$ e $u_{s\text{inicial}}$ os deslocamentos nodais. O ângulo entre o eixo s e x será θ e com isso podemos decompor u_s como:

$$u_s = u_x \cos \theta + u_y \sin \theta \quad (2.2)$$

As forças impostas sobre os nós da treliça também são decompostos nas direções x e y .

$$F_x = F * \cos \theta \quad (2.3)$$

$$F_y = F * \sin \theta \quad (2.4)$$

Podemos escrever as equações acima na forma matricial para os nós finais e iniciais do elemento:

$$\begin{Bmatrix} u_{Sinicial} \\ u_{Sfinal} \end{Bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \begin{Bmatrix} u_{Xinicial} \\ u_{Yinicial} \\ u_{Xfinal} \\ u_{Yfinal} \end{Bmatrix} \quad (2.5)$$

$$\begin{Bmatrix} F_{xinicial} \\ F_{yinicial} \\ F_{xfinal} \\ F_{yfinal} \end{Bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix} \begin{Bmatrix} F_{inicial} \\ F_{final} \end{Bmatrix} \quad (2.6)$$

Lembrando a relação tensão-deformação em barras e o equilíbrio nodal das forças, temos:

$$\sigma = E * \varepsilon \quad (2.7)$$

$$\begin{cases} F_{inicial} = -E * A * \varepsilon \\ F_{final} = E * A * \varepsilon \end{cases} \quad (2.8)$$

Ao aplicar uma força em um nó de um elemento, esta se reflete no outro nó com o sentido oposto, sendo assim uma Tensão de compressão em um nó inicial é interpretada como uma tensão de tração no nó final e vice-versa.

$$\begin{cases} F_{inicial} = -E * A * \frac{(u_{Sfinal} - u_{Sinicial})}{L} \\ F_{final} = E * A * \frac{(u_{Sfinal} - u_{Sinicial})}{L} \end{cases} \quad (2.9)$$

Reescrevendo a expressão anterior na forma matricial:

$$\begin{Bmatrix} F_{inicial} \\ F_{final} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_{Sinicial} \\ u_{Sfinal} \end{Bmatrix} \quad (2.10)$$

Introduzindo as relações (2.10) e (2.5) na equação (2.6), teremos:

$$\begin{Bmatrix} F_{xinicial} \\ F_{yinicial} \\ F_{xfinal} \\ F_{yfinal} \end{Bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix} \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \begin{Bmatrix} u_{Xinicial} \\ u_{Yinicial} \\ u_{Xfinal} \\ u_{Yfinal} \end{Bmatrix} \quad (2.11)$$

$$\begin{Bmatrix} F_{x\text{inicial}} \\ F_{y\text{inicial}} \\ F_{x\text{final}} \\ F_{y\text{final}} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & -\cos \theta & -\sin \theta \\ -\cos \theta & -\sin \theta & \cos \theta & \sin \theta \end{bmatrix} \begin{Bmatrix} u_{x\text{inicial}} \\ u_{y\text{inicial}} \\ u_{x\text{final}} \\ u_{y\text{final}} \end{Bmatrix} \quad (2.12)$$

$$\begin{Bmatrix} F_{x\text{inicial}} \\ F_{y\text{inicial}} \\ F_{x\text{final}} \\ F_{y\text{final}} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} \cos^2 \theta & \cos \theta \sin \theta & -\cos^2 \theta & -\cos \theta \sin \theta \\ \sin \theta \cos \theta & \sin^2 \theta & -\sin \theta \cos \theta & -\sin^2 \theta \\ -\cos^2 \theta & -\cos \theta \sin \theta & \cos^2 \theta & \cos \theta \sin \theta \\ -\sin \theta \cos \theta & -\sin^2 \theta & \sin \theta \cos \theta & \sin^2 \theta \end{bmatrix} \begin{Bmatrix} u_{x\text{inicial}} \\ u_{y\text{inicial}} \\ u_{x\text{final}} \\ u_{y\text{final}} \end{Bmatrix} \quad (2.13)$$

A equação acima representa a relação entre a Força aplicada sobre um elemento e seu deslocamento em uma *treliça plana*. É dada a essa matriz o nome de *matriz de rigidez*:

$$K_i = \frac{EA}{L} \begin{bmatrix} \cos^2 \theta & \cos \theta \sin \theta & -\cos^2 \theta & -\cos \theta \sin \theta \\ \sin \theta \cos \theta & \sin^2 \theta & -\sin \theta \cos \theta & -\sin^2 \theta \\ -\cos^2 \theta & -\cos \theta \sin \theta & \cos^2 \theta & \cos \theta \sin \theta \\ -\sin \theta \cos \theta & -\sin^2 \theta & \sin \theta \cos \theta & \sin^2 \theta \end{bmatrix} \quad (2.14)$$

Conforme mencionado anteriormente, para o caso de treliças espaciais, basta introduzir uma terceira coordenada z e seguir a mesma metodologia adotada para o caso plano:

$$cx = \cos \theta = \frac{x_{\text{final}} - x_{\text{inicial}}}{L} \quad (2.15)$$

$$cy = \sin \theta = \frac{y_{\text{final}} - y_{\text{inicial}}}{L} \quad (2.16)$$

$$cz = \frac{z_{\text{final}} - z_{\text{inicial}}}{L} \quad (2.17)$$

Finalmente, para o caso 3D, a expressão da matriz de rigidez K_i é dada por:

$$K_i = \frac{EA}{L} \begin{bmatrix} cx^2 & cx * cy & cx * cz & -cx^2 & -cx * cy & -cx * cz \\ cy * cx & cy^2 & cy * cz & -cy * cx & -cy^2 & -cy * cz \\ cz * cx & cz * cy & cz^2 & -cz * cx & -cz * cy & -cz^2 \\ -cx^2 & -cx * cy & -cx * cz & cx^2 & cx * cy & cx * cz \\ -cy * cx & -cy^2 & -cy * cz & cy * cx & cy^2 & cy * cz \\ -cz * cx & -cz * cy & -cz^2 & cz * cx & cz * cy & cz^2 \end{bmatrix} \quad (2.18)$$

A partir da matriz de rigidez de cada barra, o próximo passo consiste no “espalhamento” dessas matrizes, de acordo com a numeração dos nós da treliça, para montagem da matriz de rigidez global da estrutura (K). Da mesma forma, é montado o vetor de carregamento global, denominado de F . Finalmente, os deslocamentos de cada nó da treliça são obtidos resolvendo-se o sistema $KU = F$. Em função desses deslocamentos nodais, são calculadas as tensões em cada barra, as quais serão utilizadas pelo algoritmo de otimização, para verificação dos limites de escoamento e flambagem.

3. Otimização de Treliças Planas

a. Definição do problema

Na otimização de treliças, deseja-se reduzir seu peso total (ou seja, a soma dos pesos de cada elemento), considerando-se algumas restrições nos valores das tensões.

A função objetivo é definida pela seguinte equação:

$$f = \sum_{i=1}^{nTotalElementos} \rho_i * L_i * A_i \quad (3.1)$$

A tensão axial nas barras é dada pela fórmula:

$$\sigma_i = E_i * \frac{(L_{final} - L_o)}{L_o} \quad (3.2)$$

Quando submetidas a esforços de compressão axial, uma barra pode sofrer o fenômeno de instabilidade conhecido como "flambagem". A condição que permite determinar quando este fenômeno ocorre é obtida pela fórmula de Euler:

$$\sigma_{cr} = \frac{\pi^2 * E * I_i}{L_{final}^2 * A_i} \quad (3.3)$$

$$I_i = \frac{\pi * d_i^4}{64} \quad (3.4)$$

$$\sigma_i < \sigma_{cr} \quad (3.5)$$

Obs: A inequação (3.5) deve ser satisfeita para que a barra não sofra flambagem.

Nas expressões (3.1) a (3.5) temos:

f = função objetivo (peso total da treliça).

ρ_i = massa específica da barra i

L_i = comprimento da barra i

A_i = Área da seção transversal da barra i

$nTotalElementos$ = número de barras na treliça

σ_i = Tensão axial na barra i

E = Módulo de elasticidade da barra i

L_{final} = Comprimento final da barra i após deformação

L_o = Comprimento inicial da barra i

σ_{cr} = Tensão de flambagem da barra i

I_i = Momento de inércia da barra i

b. Programação não-linear

A programação não linear consiste nos métodos de otimização aplicados a problemas cujas funções envolvidas (função objetivo e restrições) são não lineares, como no caso deste trabalho. As restrições envolvidas no problema de minimização do peso de treliças respeitando-se os limites das tensões de escoamento e de flambagem são dadas por:

$$\sigma_i - \sigma_{cr} < 0 \quad (3.6)$$

$$\sigma_i - \sigma_y < 0 \quad (3.7)$$

$$d_i > 0.0001 \quad (3.8)$$

Onde essas três inequações se repetem para cada barra que compõe a treliça.

c. Métodos de otimização com restrições

Os métodos de penalidade são procedimentos de otimização de problemas com restrições que buscam uma solução numérica pela resolução de problemas de minimização *sem restrições*. Essa otimização é obtida, adicionando-se à *função objetivo* uma parcela que estabelece uma penalidade baseada nas restrições do projeto. Dependendo da escolha dos parâmetros de entrada, a função pode se tornar mal condicionada numérica e computacionalmente. A expressão abaixo representa a forma geral da chamada *pseudo função objetivo*:

$$\phi(\tilde{x}, r) = f(\tilde{x}) + r * p(\tilde{x}) \quad (3.8)$$

onde:

ϕ = pseudo função objetivo.

\tilde{x} = variáveis de projeto.

r = parâmetro de penalidade.

p = função penalidade.

i. Método da penalidade

No método da penalidade (também conhecido como “Penalidade Exterior”), a função objetivo é penalizada quando o ponto estiver fora da região viável. A função $p(\tilde{x})$ é dada por:

$$p(x) = \sum_{i=1}^m \{\max[0, C_i(x)]\}^2 + \sum_{j=1}^n [h_j(x)]^2 \quad (3.9)$$

Se todas as restrições são satisfeitas(nenhuma foi violada), então $p(x) = 0$. A pseudo função, neste caso, será equivalente a função objetivo. Caso contrário a restrição violada será penalizada de acordo com a expressão (3.9).

O algoritmo correspondente ao Método de Penalidade pode ser escrito como:

1. Entrar com os valores iniciais de \tilde{x}_0 , r_0 e β .
2. Inicializar contador $k=0$.
3. Determinar os valores de \tilde{x}^* que minimizam a pseudo função objetivo
 $\phi(\tilde{x}_k, rp_k) = f(\tilde{x}) + \frac{1}{2} rp * p(\tilde{x})$ usando o método de Otimização sem restrições.
4. Se $\left| \frac{1}{2} * rp_k * p(\tilde{x}_{k+1}) \right| < \text{tolerância}$, então FIM
 a. $\tilde{x}^* = \tilde{x}^{k+1}$
5. $rp_{k+1} = \beta * rp_k$; $k = k + 1$; Retornar ao passo 3.

ii. Método da barreira

No método da função de barreira (também conhecido como “Penalidade Interior”), a função é penalizada dentro da região viável de forma mais severa quanto mais próximo o ponto \tilde{x} estiver da fronteira. A função $p(\tilde{x})$ é dada por:

$$p(x) = \sum_{i=1}^m -\frac{1}{C_i(x)} \quad (3.10)$$

Ou, usando a expressão alternativa:

$$p(x) = \sum_{i=1}^m \log[-C_i(x)] \quad (3.11)$$

Lembrando que, para o método de barreira, usamos apenas restrições de desigualdade $C_i(x)$. Quando houver restrições de igualdade a expressão geral ficará na forma:

$$\phi(\tilde{x}_k, rp_k) = f(\tilde{x}) + \frac{1}{2} * rp * \sum_{j=1}^n [h_j(x)]^2 + rb * \sum_{i=1}^m -\frac{1}{C_i(x)} \quad (3.12)$$

O algoritmo correspondente ao Método de Barreira pode ser escrito como:

1. Entrar com os valores iniciais de \tilde{x}_0 e r_0 e β .
2. Inicializar contador $k=0$.
3. Determinar os valores de \tilde{x}^* que minimizam a pseudo função objetivo $\phi(\tilde{x}_k, rp_k) = f(\tilde{x}) + rb * p(\tilde{x})$ usando o método de Otimização sem restrições.
4. Se $|rb_k * p(\tilde{x}_{k+1})| < \text{tolerância}$, então FIM
 - a. $\tilde{x}^* = \tilde{x}^{k+1}$
5. $rp_{k+1} = \beta * rp_k$; $k = k + 1$; Retornar ao passo 3.

d. Métodos de otimização sem restrições

Os métodos clássicos de otimização sem restrições (OSR) podem ser divididos em três classes. Os de ordem zero que são utilizados quando a função objetivo a ser trabalhada não é diferenciável ou é altamente não-linear, pois suas derivadas são de difícil obtenção com precisão. Sendo assim a precisão do resultado é pequena. Os métodos mais conhecidos de ordem zero são o Univariate e o método de Powell. Os métodos de primeira ordem usam os valores da função objetivo e seus gradientes, ou seja, a derivada em relação a cada variável de projeto. Os principais métodos desta classe são *Steepest Descent* (máximo declive), *Fletcher-Reeves* e *BFGS*. Por fim, os métodos de segunda ordem utilizam os valores da função objetivo, os gradientes e a matriz Hessiana. O exemplo mais conhecido deste método é o de *Newton Raphson*.

Os métodos de OSR, procuram inicialmente uma direção d de busca para minimização da função. A forma de se calcular a direção de busca é diferente para cada método de otimização sem restrições. Conhecida esta direção, a próxima etapa consiste em calcular o tamanho do 'passo' a ser dado na direção d . A partir daí temos um problema de uma única variável (α), dado por:

$$x_{k+1} = x_k + \alpha_k * d_k$$

Para o presente trabalho, serão usados apenas os métodos de Ordem Zero.

i. Univariate

Por se tratar de um método de OSR, devemos usar primeiramente uma direção de busca que minimize a função objetivo f . Por padrão, o método apresentado utiliza direções unitárias para cada variável de projeto.

$$d_k = e_k^x$$

onde:

$$k = 1..nTotalElementos$$

$$e_k^x = \{0,0,1, \dots, 0\}$$

Teremos então uma iteração para cada variável de projeto. O conjunto de direções nos dará uma matriz identidade de tamanho $nTotalElementos$. Se após n iterações, a convergência não for obtida, um

novo ciclo deve ser iniciado. O cálculo do tamanho do passo α_k é dado por uma busca unidimensional também conhecido como busca linear.

ii. Powell

O método *Powell* é capaz de acelerar o método Univariante por meio do chamado "movimento padrão" a cada ciclo de $n_{TotalElementos}$ iterações do método Univariante. Cada nova direção substitui uma direção Univariante e, quando todas estas forem substituídas, o processo é reiniciado com um novo ciclo Univariante.

e. Busca linear

Conhecida também pelos nomes de 'busca sobre a linha' ou 'busca unidimensional', a busca linear tem como objetivo achar o mínimo ou uma aproximação deste, para uma função de uma única variável. Para o projeto apresentado, será usado o Método do Passo Incremental em conjunto com o Método da Seção Áurea.

i. Método do passo incremental

- Determinar se o incremento inicial é positivo ou negativo
 - Se $f(0) < f(\delta)$ $\xRightarrow{\text{então}}$ Função está crescendo em $f(0)$ $\xRightarrow{\text{então}}$ Ponto de mínimo encontra-se a esquerda de 0. Incremento inicial será negativo
 - Se $f(0) > f(\delta)$ $\xRightarrow{\text{então}}$ Função está decrescendo em $f(0)$ $\xRightarrow{\text{então}}$ Ponto de mínimo encontra-se a direita de 0. Incremento inicial será positivo.
- A fórmula para o Passo incremental é dada abaixo:

$$\alpha_{k+1} = \alpha_k \pm 2^{k-1} * \delta$$
- O Valor de α_k vai sendo modificado até que o valor da função f se torne maior do que o imediatamente anterior ou seja:
 - Se $f(\alpha_{k+1}) > f(\alpha_k)$ $\xRightarrow{\text{então}}$ $\alpha^* = \alpha_k$

Para o uso do método da seção áurea teremos α^* e o seu passo anterior como os limites inferior e superior dependendo da direção de *busca unidirecional*.

ii. Método da seção áurea

- Determinar um intervalo de busca α^L e α^U . (Extraído do passo incremental)
- Dividir o intervalo em 2 pontos:
 - $\alpha_E = \alpha^L + (1 - R_A) * \beta$
 - $\alpha_D = \alpha^L + (R_A) * \beta$

onde:

$$R_A = \text{Razão Áurea} = \frac{(\sqrt{5} - 1)}{2}$$

$$\beta = \alpha^U - \alpha^L$$

- Avaliar a função nos dois pontos
 - Se $f(\alpha_E) \geq f(\alpha_D)$ $\xRightarrow{\text{então}}$ Descartar o intervalo da esquerda.
 - Se $f(\alpha_D) > f(\alpha_E)$ $\xRightarrow{\text{então}}$ Descartar o intervalo da direita.
- Redefinir o intervalo
 - Para descartar o intervalo da esquerda. $\alpha^L = \alpha_E$
 - Para descartar o intervalo da direita $\alpha^U = \alpha_D$

4. Resultados

Os resultados do problema se mostraram satisfatórios e atenderam aos valores de tolerância pré-estabelecidos. Para validação dos métodos implementados neste trabalho, foi usado em um script separado, o toolbox de otimização do MATLAB denominado *FMINCON*.

a. Treliça de 2 barras

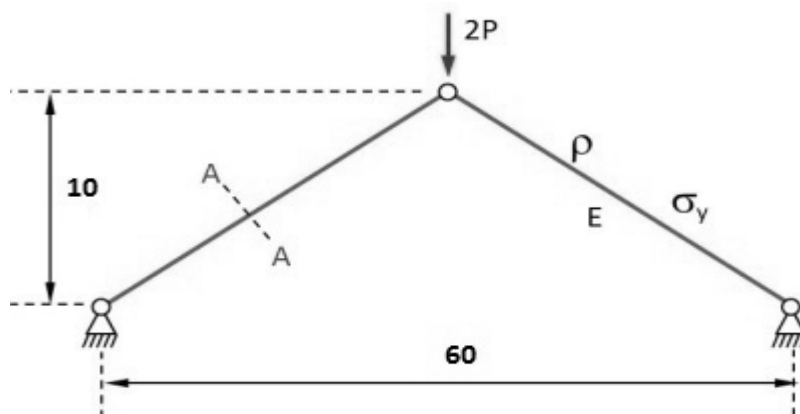


Figura 4.1

Os dados do problema são:

$$\rho = 0,3$$

$$P = 33.000$$

$$E = 3 * 10^7$$

$$\sigma_y = 10^5$$

Para estes parâmetros, os resultados obtidos foram:

i. Método da penalidade

ITERAÇÃO	d1	d2	Valor na Função	Rp
6	1,1511	1,1511	19,6788	10000
5	1,1498	1,1498	19,6349	1000
4	1,138	1,138	19,2347	100
3	1,0718	1,0718	17,0523	10
2	0,92509	0,92514	12,689	1
1	0,75665	0,75669	8,4684	0,1
0	1	1	14,8372	0

Figura 4.2

Cada linha da tabela representa uma iteração do método de otimização com restrições, sendo a primeira linha o valor final e a última coluna o coeficiente de penalidade que é incrementado a cada ciclo.

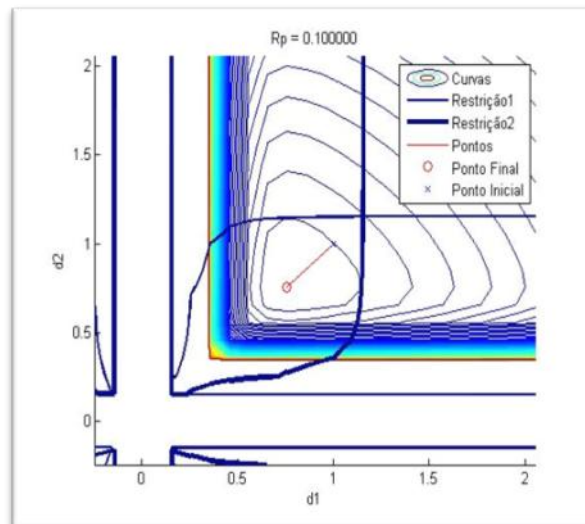


Figura 4.3

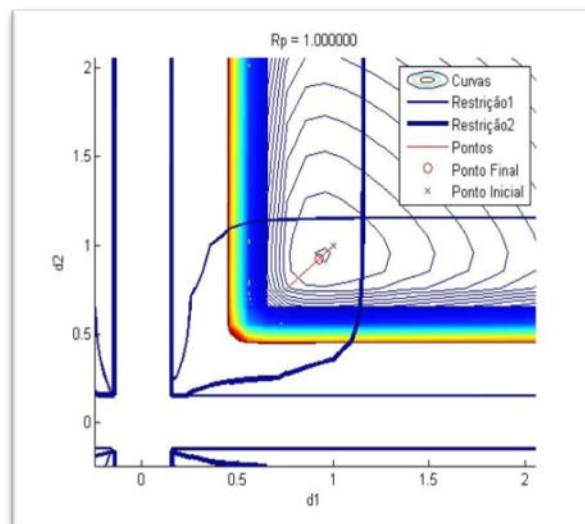


Figura 4.4

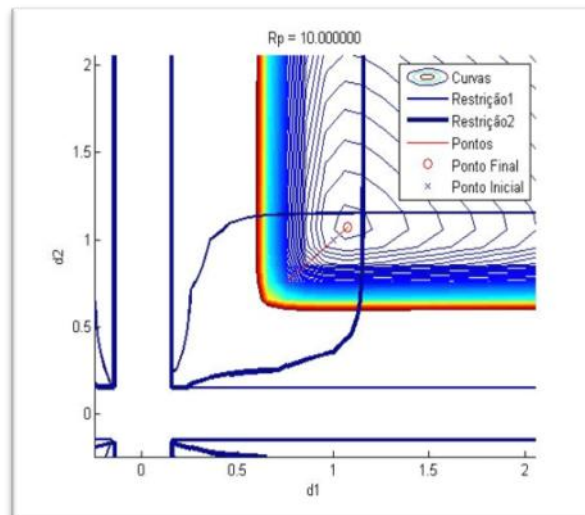


Figura 4.5

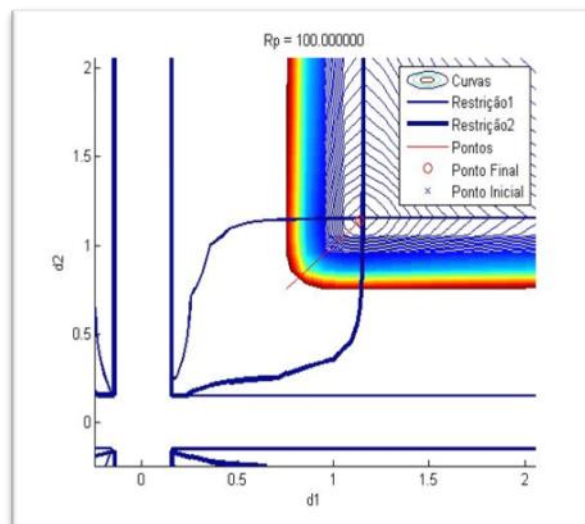


Figura 4.6

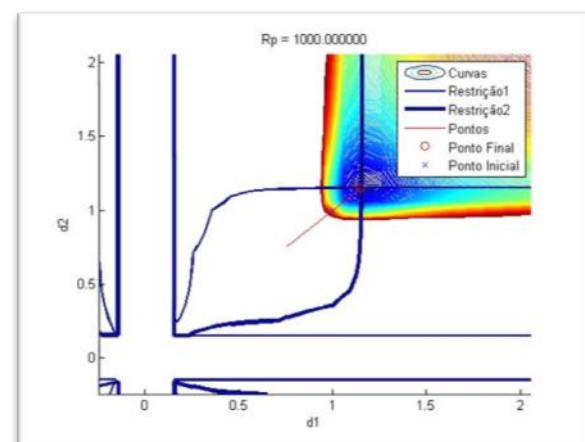


Figura 4.7

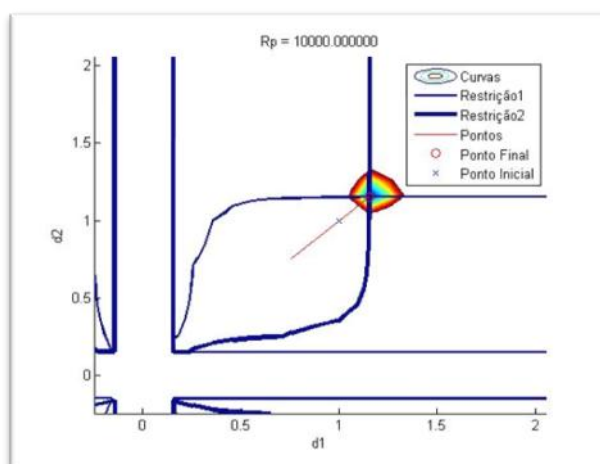


Figura 4.8

Acima vemos nos gráficos a evolução a cada interação do ponto bem como, a *pseudo-função* que vai sendo atualizada para satisfazer as restrições. O parâmetro R_p , tem como finalidade penalizar a *pseudo-função* a cada interação e o seu valor pode ser visto no título de cada gráfico.

ii. Método da barreira

ITERAÇÃO	d1	d2	Valor na Função	Rb
17	1,1513	1,1513	19,6877	0,001526
16	1,1514	1,1514	19,6898	0,003052
15	1,1516	1,1516	19,6961	0,006104
14	1,1519	1,1519	19,708	0,012207
13	1,1526	1,1526	19,7324	0,024414
12	1,154	1,154	19,7807	0,048828
11	1,1568	1,1568	19,8769	0,097656
10	1,155	1,155	19,8152	0,19531
9	1,1656	1,1657	20,1835	0,39063
8	1,1857	1,1858	20,8874	0,78125
7	1,2217	1,2217	22,177	1,5625
6	1,282	1,2819	24,4249	3,125
5	1,3753	1,3753	28,1201	6,25
4	1,5094	1,5094	33,886	12,5
3	1,6905	1,6905	42,5192	25
2	1,9244	1,9244	55,1201	50
1	2,2179	2,2179	73,2384	100
0	2	2	59,5419	0

Figura 4.9

Cada linha da tabela representa uma interação do método de otimização com restrições, sendo a primeira linha o valor final e ultima coluna o parâmetro de penalidade que é diminuído a cada ciclo.

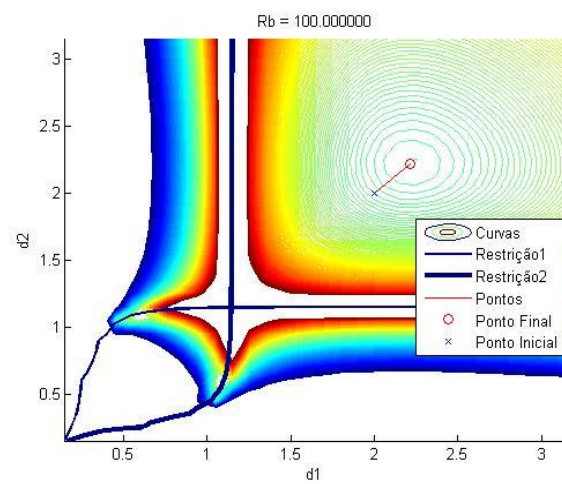


Figura 4.10

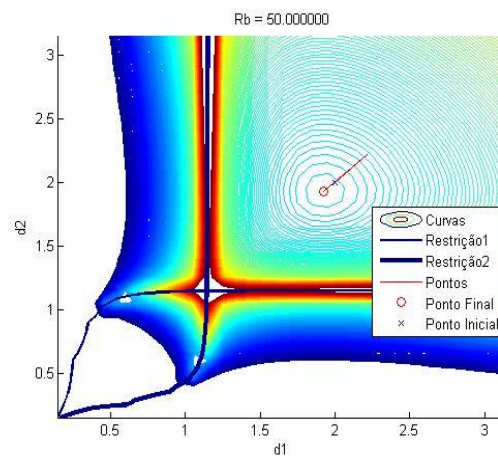


Figura 4.11

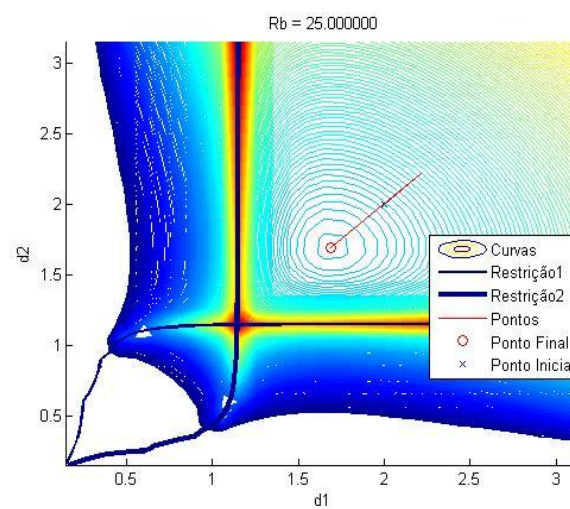


Figura 4.12

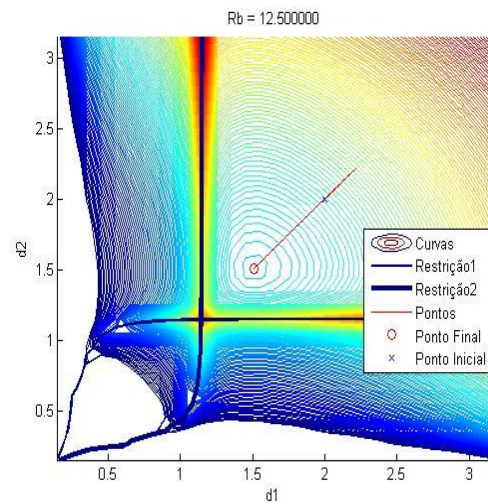


Figura 4.13

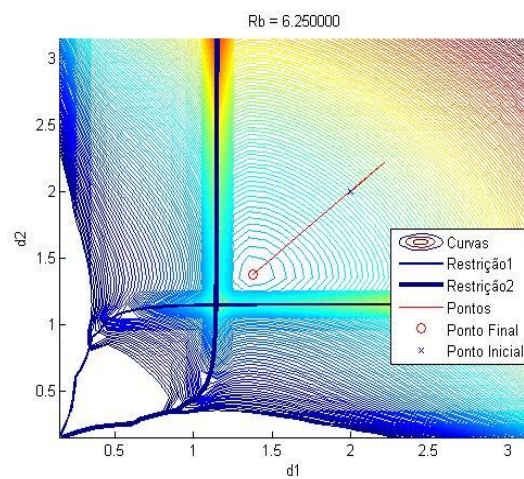


Figura 4.14

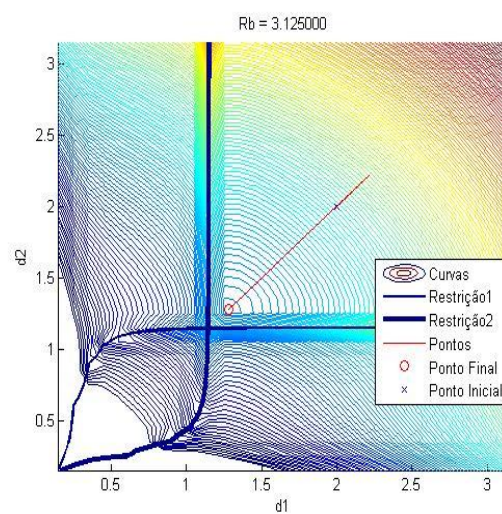


Figura 4.15

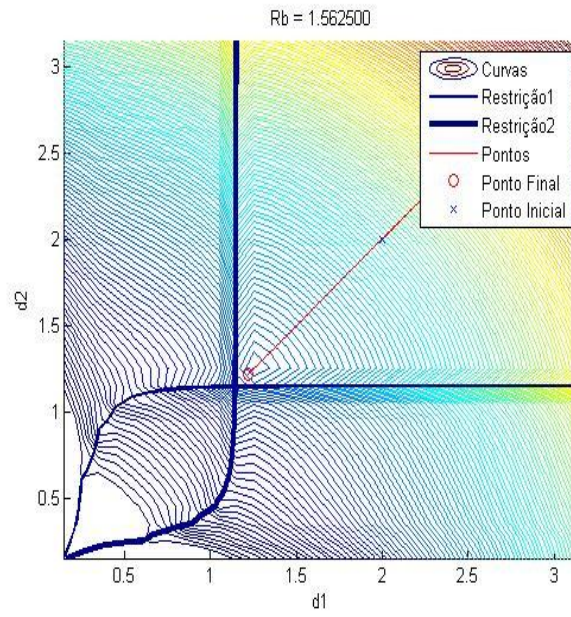


Figura 4.16

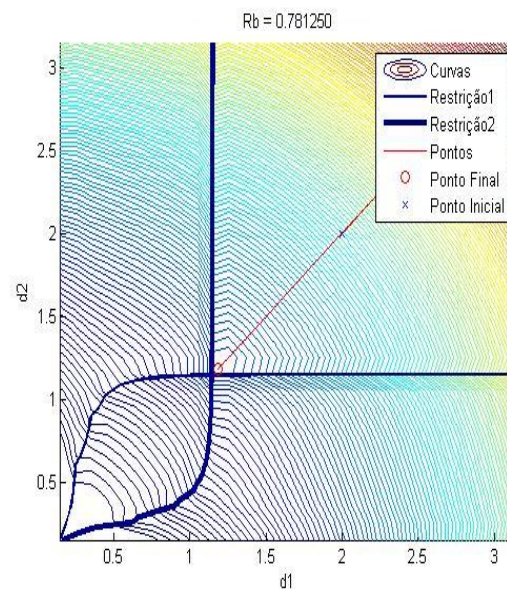


Figura 4.17

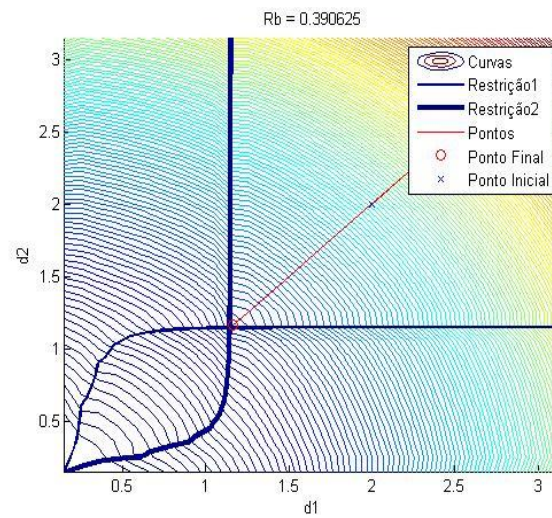


Figura 4.18

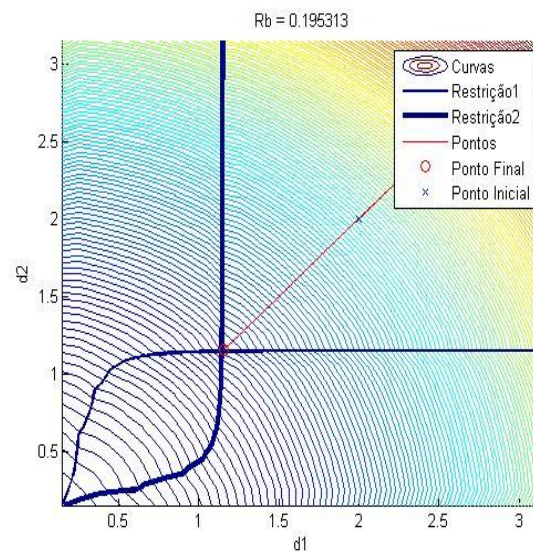


Figura 4.19

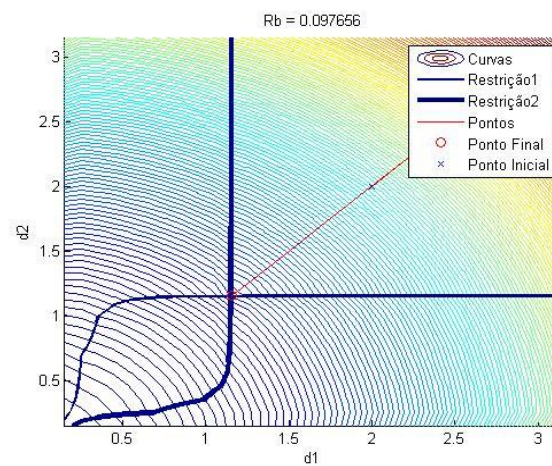


Figura 4.20

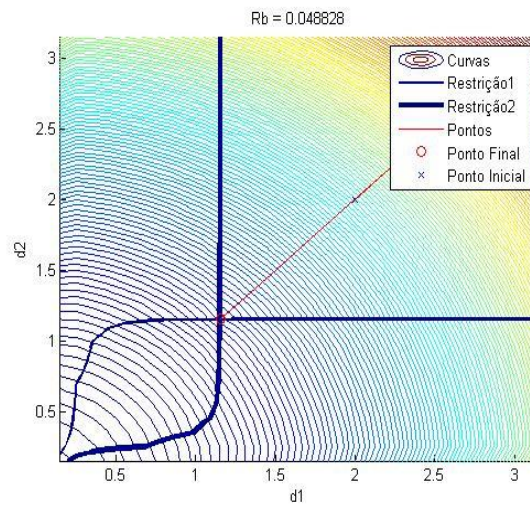


Figura 4.21

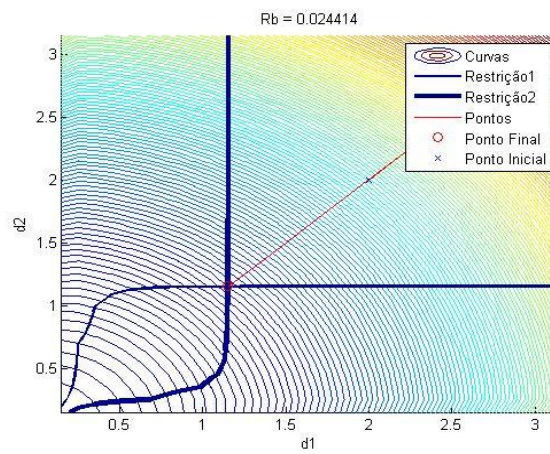


Figura 4.22

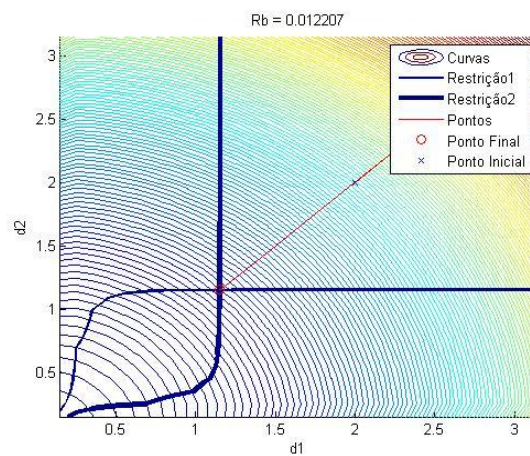


Figura 4.23

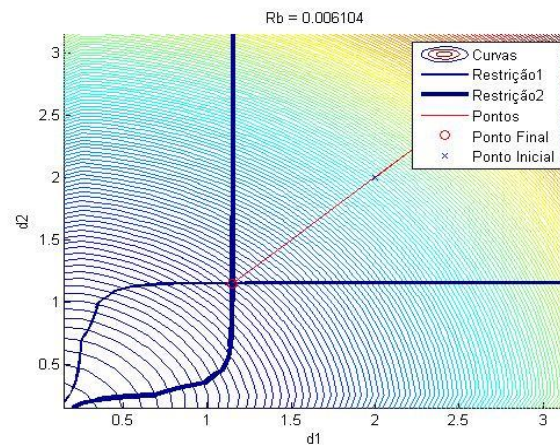


Figura 4.24

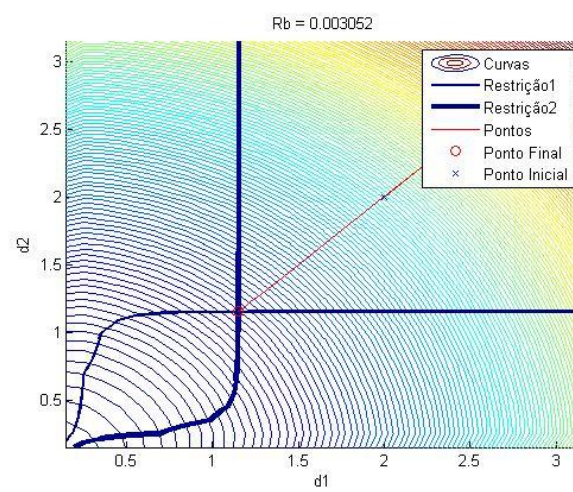


Figura 4.25

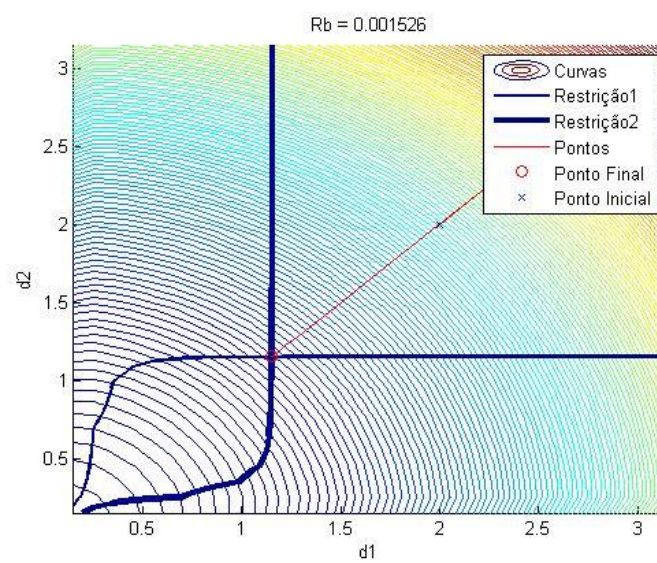


Figura 4.26

Acima vemos nos gráficos a evolução a cada interação do ponto bem como, a *pseudo-função* que vai sendo atualizada para satisfazer as restrições. O Parâmetro R_b , tem como finalidade penalizar a *pseudo-função* a cada interação e o seu valor pode ser visto no título de cada gráfico.

iii. Comparação Fmincon

Temos abaixo tabelas comparando os resultados dos métodos penalidade, barreira e o código Fmincon respectivamente.

d1	d2	Valor na Função
1,1511	1,1511	19,6788

Figura 4.27

d1	d2	Valor na Função
1,1513	1,1513	19,6877

Figura 4.28

d1	d2	Valor na Função
1.1575	1.1575	19.9652

Figura 4.29

b. Treliza de 10 barras

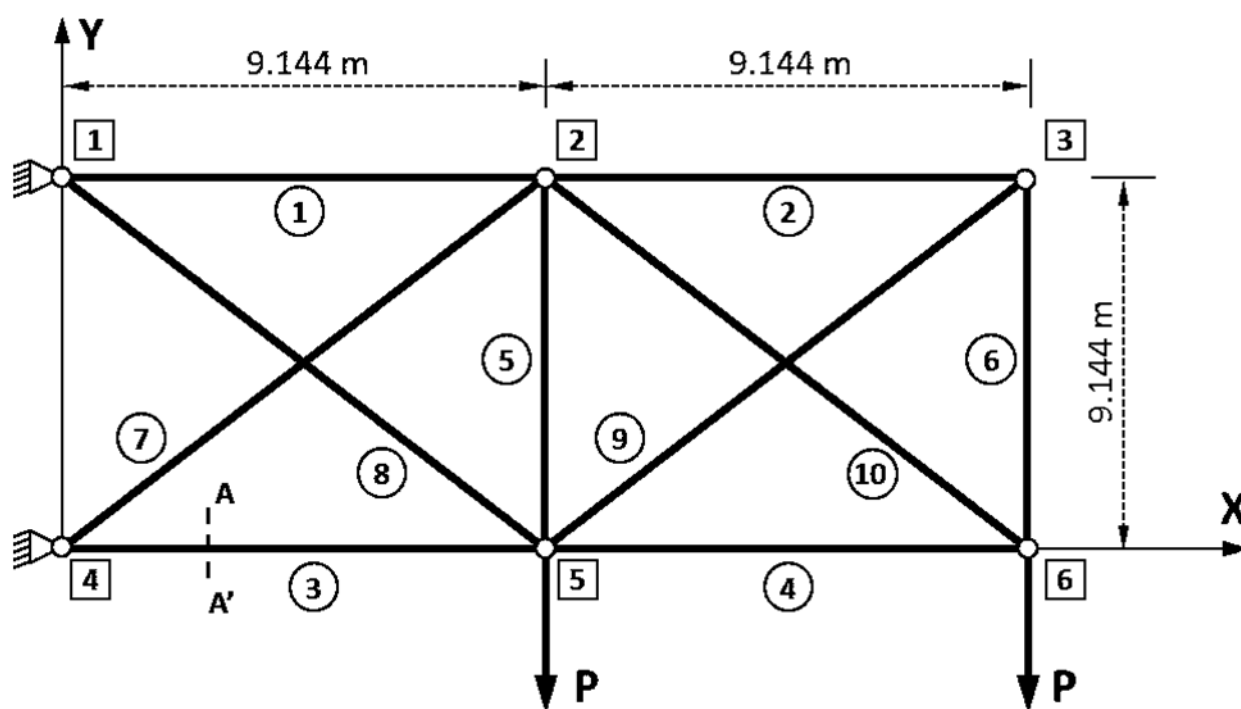


Figura 4.30

Os dados do problema são:

$$\rho = 27,143$$

$$P = 44,822$$

$$E = 68.947.600,000$$

$$\sigma_y = 172.369,000$$

Para estes parâmetros, os resultados obtidos foram:

i. Método da penalidade

ITERAÇÃO	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
10	0,081134	-0,00165	0,15268	0,1284	0,000369	-0,0034	0,16701	0,068257	0,020589	0,068433
9	0,081086	-0,00165	0,15266	0,12782	0,000245	-0,0034	0,16701	0,068246	0,020588	0,068427
8	-0,08112	-0,00254	0,15269	0,12776	3,07E-07	-0,00537	0,16695	0,068296	0,020088	0,068133
7	-0,08111	-0,00253	0,15264	0,12776	-4,4E-05	-0,00538	0,16687	0,068208	0,028249	0,068441
6	-0,08101	-0,00253	0,15241	0,12764	3,9E-07	-0,00543	0,16647	0,068093	0,028238	0,068414
5	-0,08022	-0,00251	0,15139	0,127	3,9E-07	-0,00835	0,16466	0,067152	0,028501	0,067624
4	0,074408	0,009256	0,14834	0,12272	4,47E-07	-0,00912	0,15748	0,070881	0,029921	0,060603
3	0,020196	0,014434	0,1082	0,05904	-0,16687	-0,00886	0,10438	0,025454	0,03743	0,032575
2	0,055608	0,038783	0,12923	0,020401	-0,02479	0,046586	0,027404	0,050899	0,038521	0,035166
1	0,013051	0,025107	0,10656	0,12221	-0,0468	0,008108	0,024314	0,030096	0,030623	0,03341
0	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03

Figura 4.31

ITERAÇÃO	Valor na Função	Rp
10	19,427	976562,5
9	19,3947	195312,5
8	19,3779	39062,5
7	19,4845	7812,5
6	19,4194	1562,5
5	19,0841	312,5
4	17,8202	62,5
3	12,3981	12,5
2	6,4498	2,5
1	6,7162	0,5
0	2,0487	0,1

Figura 4.32

As tabelas acima representam os resultados a cada iteração do método da penalidade para a treliça de 10 barras

ii. Comparação Fmincon

Temos abaixo tabelas comparando os resultados dos método penalidade e o código Fmincon respectivamente.

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	Valor na Função
0,081134	-0,00165	0,15268	0,1284	0,000369	-0,0034	0,16701	0,068257	0,020589	0,068433	19,427

Figura 4.33

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	Valor na Função
0.0001	0.0001	0.1545	0.1136	0.0001	0.0001	0.1655	0.0001	0.1333	0.0001	19.6228

Figura 4.34

5. Conclusão

Neste trabalho foram implementados os métodos de otimização não linear, com restrições, conhecidos como Métodos de Penalidade. O objetivo do trabalho foi minimizar o peso de treliças respeitando-se as restrições de flambagem das barras e escoamento dos materiais. Foram implementados os métodos de penalidade exterior e interior (método da barreira). Para o processo de otimização sem restrições foram utilizados os métodos de ordem zero (univariante e Powell) e para as buscas unidimensionais foi adotado o método da Seção Áurea combinado com o método do passo incremental. Os exemplos apresentados neste trabalho foram comparados com o método do FMINCON, disponível no software MATLAB. Os resultados obtidos comprovam a validade e a eficiência das implementações realizadas.

6. Referências Bibliográficas

- 1 – L. E. Vaz, A. Pereira e I. F. M. Menezes, Notas de Aula, 2012.
- 2 – G. Vanderplaats, Numerical Optimization Techniques for Engineering Design with Applications, McGraw-Hill, 1984.
- 3 – R. T. Haftka e Z. Gurdal, Elements of Structural Optimization, 3rd Edition, Kluwer Academic Press, 1992.
- 4 – J. Nocedal e S. J. Wright, Numerical Optimization, 2nd Edition, Springer, 2006.
- 5 – M. S. Bazaraa, H. D. Sherali e C. M. Shetty, Nonlinear Programming: Theory and Algorithms, 2nd Edition, John Wiley & Sons, Inc., 1993.