MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
DEPARTAMENTO DE ENGENHARIA MECÂNICA

CÓDIGO MATLAB PARA A OTIMIZAÇÃO TOPOLÓGICA DE MECANISMOS FLEXÍVEIS PELA ABORDAGEM DE MAXIMIZAÇÃO DE ENERGIA DE DEFORMAÇÃO

por

Marcus Vinícius Simioni

Monografia apresentada ao Departamento de Engenharia Mecânica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do diploma de Engenheiro Mecânico.

Porto Alegre, novembro de 2013

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Engenharia Mecânica

CÓDIGO MATLAB PARA A OTIMIZAÇÃO TOPOLÓGICA DE MECANISMOS FLEXÍVEIS PELA ABORDAGEM DE MAXIMIZAÇÃO DE ENERGIA DE DEFORMAÇÃO

por

Marcus Vinícius Simioni

ESTA MONOGRAFIA FOI JULGADA ADEQUADA COMO PARTE DOS REQUISITOS PARA A OBTENÇÃO DO TÍTULO DE
ENGENHEIRO MECÂNICO
APROVADA EM SUA FORMA FINAL PELA BANCA EXAMINADORA DO DEPARTAMENTO DE ENGENHARIA MECÂNICA

Prof.ª Dr.ª Thamy Cristina Hayashi
Coordenadora do Curso de Engenharia Mecânica

Área de Concentração: **Mecânica dos Sólidos**

Orientador: Prof. Dr. Jun Sérgio Ono Fonseca

Comissão de Avaliação:

Prof.ª Dr.ª Leticia Fleck Fadel Miguel

Prof. Dr. Ignacio Iturrioz

Prof. Dr. Jakson Manfredini Vassoler

Porto Alegre, 25 de novembro de 2013.

SIMIONI, M.V. **CÓDIGO MATLAB PARA A OTIMIZAÇÃO TOPOLÓGICA DE MECANISMOS FLEXÍVEIS PELA ABORDAGEM DE MAXIMIZAÇÃO DE ENERGIA DE DEFORMAÇÃO**. 2013. 15 p. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

RESUMO

A Otimização Topológica é, dentre as metodologias de otimização, a que apresenta maior grau de complexidade. O fato de haver alteração simultânea de parâmetros e de forma, seja pela inclusão ou remoção de material no domínio, faz com que o controle da resposta, bem como a sua validação, seja um ponto a ser notado durante a análise do problema. Este trabalho desenvolve uma metodologia simplificada, com viés didático, com o objetivo de reproduzir resultados da Abordagem de Maximização de Energia de Deformação, utilizando como base um programa em linguagem MATLAB, conhecido na literatura especializada, desenvolvido para a Abordagem de Minimização de Flexibilidade. Neste trabalho se desenvolve uma exposição da abordagem-alvo, indicando as alterações feitas com o objetivo de diminuir o custo computacional. Procede-se para a exposição do código, explicando aspectos paramétricos de controle de convergência e complexidade. Por fim, se discute os resultados através da interpretação das topologias geradas de exemplos escolhidos.

PALAVRAS-CHAVE: Otimização Topológica, Mecanismos Flexíveis, MATLAB.

SIMIONI, M.V. **MATLAB CODE USING STRAIN ENERGY MAXIMIZATION APPROACH IN TOPOLOGY OPTIMIZATION TO THE DESIGN OF COMPLIANT MECHANISMS**. 2013. 15 p. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

ABSTRACT

Topological Optimization is among the optimization methodologies that has the highest degree of complexity. The fact that simultaneous change of parameters and shape, either by addition or removal of material in the domain, makes the control response as well as its validation a point to be noticed during the analysis of the problem . This paper develops a simplified methodology, with educational bias, in order to reproduce the results of Strain Energy Maximization Approach using a program based on a MATLAB language code, known in the literature, to the Flexibility Minimization Approach. This paper develops an exposure of the target approach, indicating the changes made with the aim of reducing the computational cost. Proceeds to the exposure of the code, explaining parametric control aspects of convergence and complexity. Finally, it is discussed the results from the interpretation of the generated topologies of selected examples .

KEYWORDS: Topology Optimization, Compliant Mechanisms, MATLAB.

SUMÁRIO

Pág.

1. INTRODUÇÃO 1
2 OBJETIVOS 2
3 METODOLOGIA 3
4 APRESENTAÇÃO DO PROBLEMA 7
5 RESULTADOS E DISCUSSÃO..... 9
6. CONCLUSÕES 12
REFERÊNCIAS BIBLIOGRÁFICAS 13
Apêndice I 17

1. INTRODUÇÃO

Pode-se argumentar que a busca analítica por um resultado de determinado problema, a partir de uma solução absoluta e irrefutável, perfaz um dos objetivos mais evidentes da natureza humana. No entanto, confrontam-se nesta busca efeitos de paralaxe que podem nublar a visão de ótimo de um problema, tal como considerar ótimo global uma solução de ótimo local, ou mesmo quando se trata de problemas de solução não dominada, ou ótimo de Pareto, onde não é possível otimizar um parâmetro sem comprometer outro.

Com desenvolvimento do cálculo diferencial, segundo [Liberzon, 2012], os séculos XVII e XVIII foram ricos em exemplos de soluções ótimas para antigos problemas da física e da engenharia. Em 1691, Johann Bernoulli obtém a descrição da curva catenária, definida como a forma ideal assumida por uma corrente de peso uniforme suspensa pelas extremidades. Em 1696, Bernoulli também propõe o problema da obtenção da curva braquistócrona, ou seja, a trajetória que minimizaria o tempo de deslocamento, sem atrito, de uma partícula submetida a um campo gravitacional constante. No ano de 1697, um ano após o lançamento do desafio, quatro matemáticos responderam com as soluções, a saber: Isaac Newton, Jakob Bernoulli, Gottfried Leibniz, Ehrenfried Walther von Tschirnhaus e Guillaume de l'Hôpital. No entanto, a solução mais interessante, do ponto de vista de otimização, é a do próprio Johann. Ao desenvolver uma solução baseada no *Princípio de Fermat* que continha uma nova família de curvas, as cicloides, Johann Bernoulli lança as bases do que, mais adiante, seria chamado o *Cálculo Variacional*. Em 1744, Leonhard Euler obtém a catenoide, superfície gerada a partir da revolução da curva catenária com relação a um eixo conveniente denominado diretriz. A curva catenoide, dentre outras propriedades, apresenta-se como a superfície mínima para a ação da tensão superficial, daí ser comum apresentar esta curva como a superfície de uma bolha de sabão. Um sólido gerado pela revolução desta curva em torno de sua diretriz é o sólido que melhor distribuiria calor, se as fronteiras fossem isoladas, ou tensão mecânica caso fosse aplicada uma força uniforme na seção de menor diâmetro.

A partir do desenvolvimento do Cálculo Variacional, em meados do século XVIII, surgem formalmente as primeiras condições de otimalidade, como os multiplicadores de Lagrange, a equação de Euler-Lagrange como condição necessária de primeira ordem, a condição necessária de segunda ordem de Legendre e todo um ferramental matemático que viria a pavimentar a otimização como um todo.

No entanto, conforme as hipóteses simplificadoras dos modelos foram sendo removidas, de modo a tratar de problemas mais realísticos, a complexidade da abordagem por meio de cálculo de variações cresceu de forma a desestimular a pesquisa com viés de engenharia, porque nem sempre foi possível resolver os sistemas de equações diferenciais parciais associados aos funcionais dos cálculos de variações.

Em [Michell ,1904], considerando o trabalho de Maxwell e a sua formulação para o princípio dos Trabalhos Virtuais, inaugura-se a Otimização Estrutural na discussão dos limites para a economia de material na construção de estruturas reticuladas. Calculou-se as linhas de isotensão principais para uma força aplicada em um ponto num domínio infinito, este que se encontraria sujeito a restrições de deslocamento. No fim, chegou-se à conclusão de que a estrutura ótima, para este caso, seria uma treliça, ou seja, estrutura composta por barras que, por definição, são livres da ação de momentos fletores. Com este trabalho Michell se torna pioneiro na Otimização Estrutural e seus ótimos obtidos para estruturas de carregamentos simples são utilizadas ainda hoje como *benchmark* em programas computacionais [Sigmund, 2000].

Embora os casos clássicos como a catenária e a braquistócrona apresentem solução de rigor e beleza peculiares, as soluções na abordagem do Cálculo Variacional que se mostram fiéis a um modelo realístico e que podem ser resolvidas completamente são poucas e de difícil implementação. Por este motivo se observa uma transição de conduta na qual a otimização recairia mais sobre a variação de parâmetros em elementos discretos, seja optando por estruturas discretas, como é o caso das treliças, ou optando-se pela prática, cada vez mais frequente, de se proceder à análise numérica pela discretização do domínio em elementos

finitos, esta última estimulada pelos ganhos tecnológicos das últimas décadas na capacidade de processamento computacional.

Os métodos de discretização de equações de equilíbrio em um domínio tornam o custo computacional da análise crescente e este acaba criando a demanda por novos ferramentais de cálculo. Aí é que surgem novos algoritmos para adaptar computacionalmente as condições de otimalidade, como é o caso das condições KKT (Karush–Kuhn–Tucker). Surgem uma ampla variedade de métodos de resolução por processo iterativo, inerente ao cálculo numérico, que buscam novas estratégias de convergência, dos quais pode-se citar de [Haftka e Gürdal, 1992] o Método de Newton, o Método Simplex, a Programação Linear Sequencial, o Método da Penalização Interior, etc.

Chegado a este ponto, antes ainda de seguir ao problema de otimização, cabe apresentar os diferentes aspectos da Otimização Estrutural. Nesta área os problemas se dividem em otimização paramétrica, otimização de forma e otimização topológica. Como o nome indica, a otimização paramétrica está interessada na variação de apenas um parâmetro, tal como uma área de seção que minimize ou maximize uma grandeza física, como a flexibilidade ou energia de deformação. A principal característica da otimização paramétrica é que o domínio do modelo e as variáveis de projeto, a priori, são conhecidas. A otimização de forma está interessada numa família de curvas, tal como no caso das curvas catenoides, geralmente descrevendo a forma ótima de uma geometria para determinado estado de carregamento. A otimização topológica de estruturas sólidas tem por objetivo desenvolver simultaneamente a otimização de parâmetros e de forma, incluindo-se vacâncias de material no domínio.

Por último, define-se mecanismos flexíveis os componentes cuja mobilidade advém inteiramente da flexibilidade de seus membros, ao contrário dos mecanismos convencionais que devem a sua flexibilidade a dobradiças, rolamentos ou a juntas deslizantes. Entre outras vantagens está que, uma vez construído o mecanismo, é fácil de montar e a manutenção é simplificada já que detém poucas peças e não necessita de lubrificação. Podem ser concebidos pela discretização do domínio contínuo (elementos finitos) ou por aplicação de estruturas discretas, como treliças [Bendsøe e Sigmund, 2004]. Este trabalho seguirá a primeira proposição, utilizando Otimização Topológica para a remoção de material do domínio simultânea a uma análise simplificada de Elementos Finitos, a qual resolverá a equação de equilíbrio para cada iteração, de forma a se garantir que o ponto ótimo encontrado pelo algoritmo realmente tenha relevância física.

2. OBJETIVOS

O objetivo deste trabalho é desenvolver um código simplificado no programa MATLAB para a abordagem de Maximização de Energia de Deformação para o Projeto de Mecanismos Flexíveis, proposto em [Cardoso e Fonseca, 2004], inspirado no que foi desenvolvido para a abordagem de minimização de flexibilidade em [Sigmund, 2001]. A primeira abordagem, originalmente desenvolvida em linguagem C++, se diferencia da abordagem de flexibilidade pela imposição de que parte da energia entregue na entrada do sistema seja acumulada na forma de energia de deformação elástica pela estrutura. A hipótese assumida é que, para que um mecanismo seja totalmente flexível, deve-se maximizar tanto quanto possível a quantidade de energia de deformação acumulada na estrutura, considerando restrições de entrada e saída. Com isso, se deseja impedir o surgimento de rótulas, comuns em abordagens como a minimização de flexibilidade.

A ideia central no desenvolvimento da abordagem proposta em uma linguagem de programação de alto nível, como é o caso do MATLAB, é fortalecer o aspecto didático na apresentação do processo numérico inerente à otimização topológica, se valendo também da facilidade em se construir gráficos no programa, aliviando condutas mais específicas de programação. Por outro lado, esta opção traz consigo a perda de velocidade de processamento e uma necessidade de se conhecer o *software* de forma mais aprofundada para se manipular erros numéricos.

3. METODOLOGIA

3.1. OTIMIZAÇÃO TOPOLÓGICA

3.1.1. PARAMETRIZAÇÃO DE PROJETO

A metodologia utilizada neste trabalho para o projeto de mecanismos flexíveis é a otimização topológica. O que se define, a priori, são as forças prescritas e os seus respectivos pontos de aplicação, os pontos de apoio da estrutura a ser projetada, as dimensões do domínio material, a quantidade de material base no domínio, indicada pela fração de volume e o tipo de material base sobre o qual será projetado o mecanismo. No entanto, para se proceder à programação, é preciso definir a forma como o algoritmo irá suprimir material de dentro do domínio. Neste trabalho seguiu-se [Sigmund, 2001] na opção pelo modelo SIMP, cuja sigla pode ser traduzida literalmente como "Material Sólido Isotrópico com Penalização". Este modelo atua convertendo o problema inteiro de se atribuir valores para a densidade do material em um problema de otimização de parâmetro, a pseudodensidade, que pode ser interpretada, após a penalização, em valores 0-1. A expressão (1)

$$\begin{aligned} &\mathbf{E}(x) = \rho(x)^p \mathbf{E}^0, \\ &p > 1, \rho \in [0,1] \end{aligned} \qquad (1)$$

representa a dependência do tensor constitutivo de quarta ordem **E(x)**, em cada ponto do domínio do projeto, com respeito à distribuição espacial da pseudodensidade material *ρ(x)* e às propriedades constitutivas do material base, $\mathbf{E^0}$. O valor *p* do expoente da pseudodensidade é o coeficiente de penalização. Para que o modelo SIMP esteja ativo, o valor de *p* deve ser maior do que 1. Assim, os valores intermediários de pseudodensidade são reduzidos a valores próximos de 0 ou de 1.

Para ilustrar a relação entre a densidade de energia de deformação, W , com as propriedades do material, considera-se a formulação para uma estrutura elástica linear com densidade uniforme, ρ. Em (2)

$$W = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{E} : \boldsymbol{\varepsilon} = \frac{1}{\rho}\boldsymbol{\varepsilon}^0 : \rho\mathbf{E^0} : \frac{1}{\rho}\boldsymbol{\varepsilon}^0 = \frac{1}{\rho}W^0, \qquad (2)$$

tem-se $\boldsymbol{\varepsilon}$ como o tensor de deformação e, da mesma forma que em (1), os expoentes 0 em $\boldsymbol{\varepsilon}^0$ e W^0 indicam o tensor de deformação e a energia de deformação, respectivamente, no material base, ou seja, quando ρ=1.

Nota-se em (2) a utilidade, para fins de otimização, da relação inversa entre a densidade do material e a energia de deformação. Se a formulação matemática (4) opera alterando a pseudodensidade do elemento, tendo como objetivo maximizar a energia de deformação da estrutura, dada em (3), sabemos que deve ρ deve decair conforme avança o processo iterativo.

$$U = \int_{\Omega} W d\Omega . \qquad (3)$$

Em (3), tem-se *U* como energia de deformação na estrutura e Ω representa as variáveis geométricas do domínio de projeto.

3.1.2. FORMULAÇÃO MATEMÁTICA

A função objetivo e sua expressão, na forma discreta, para cada elemento *e* no domínio, segue

$$\Phi = \sum_{e}^{N} \rho_e^p \mathbf{U}_\mathbf{e}^T \mathbf{K}_\mathbf{e} \mathbf{U}_\mathbf{e}, \qquad (4)$$

onde Φ é a função objetivo, $\mathbf{U}_\mathbf{e}$ é o vetor de deslocamento local e $\mathbf{K}_\mathbf{e}$ é a matriz de rigidez local. A pseudodensidade é uma função definida em todo o domínio. Em (4), diferentemente de em (6), o valor da pseudodensidade é aplicada a um elemento *e*, este que só pode assumir um único valor de pseudodensidade. Assim, a natureza vetorial da pseudodensidade da eq.(6) é justificada pela necessidade de se atribuir valores de pseudodensidade, elemento a elemento, em todo o domínio de projeto.

Definida a função objetivo, a formulação matemática do problema de otimização fica:

$$\begin{aligned} &\text{Max:} \quad \Phi(\rho) \qquad (5)\\ &\text{Suj.} \quad : \sum_{e}^{N} \rho_e V_e \le V_{\max} \\ &\qquad :\mathbf{K}(\rho)\mathbf{U}=\mathbf{F} \\ &\qquad : 0 < \rho_{\min} \le \rho_e \le 1 \end{aligned}$$

V é o volume do estrutura, V_o é o volume do domínio e *f* é a fração de volume, ou seja, a quantidade de material que se quer manter. **F** é o vetor de força, **U** é o vetor de deslocamentos global e **K** é a matriz de rigidez global. Os valores de pseudodensidade surgem como $\rho_{\min}$ e ρ_e, e a desigualdade na qual estão inseridos é válida para qualquer elemento do domínio.

3.1.3. ANÁLISE DE SENSIBILIDADES

A sensibilidade da função objetivo com respeito às variáveis de projeto é avaliada pela abordagem adjunta [Cardoso e Fonseca, 2004]. A função objetivo (4) é reescrita adicionando-se

$$\Phi = \sum_{e=1}^{N} \left[\boldsymbol{\rho}^p \left(\mathbf{H}_\mathbf{e} \mathbf{u}_\mathbf{g} \right)^T \mathbf{K}_\mathbf{e} \left(\mathbf{H}_\mathbf{e} \mathbf{u}_\mathbf{g} \right) \right] + \boldsymbol{\lambda}_\mathbf{g}^\mathrm{T} \left(\mathbf{K}_\mathbf{g} \mathbf{u}_\mathbf{g} - \mathbf{f}_\mathbf{g} \right) \qquad (6)$$

onde $\mathbf{H}_\mathbf{e}$ é o operador localização, ou seja, vetor que liga o vetor de deslocamento do elemento, $\mathbf{u}_\mathbf{e}$, ao vetor de deslocamento global, $\mathbf{u}_\mathbf{g}$. $\mathbf{K}_\mathbf{g}$ é a matriz de rigidez global, $\mathbf{f}_\mathbf{g}$ é o vetor global de forças externas e $\boldsymbol{\lambda}_\mathbf{g}$ é o vetor adjunto global. A equação de equilíbrio foi incorporada à eq. (4) e o seu resultado deve ser igual a zero, logo deve não interferir na equação original.

Conforme [Cardoso e Fonseca, 2004], derivando-se a eq. (6) com relação à pseudodensidade e agrupando-se os termos comuns, chega-se na seguinte expressão para o cálculo do vetor adjunto:

$$\boldsymbol{\lambda}_{\mathbf{g}}^{\mathbf{T}}\mathbf{K}_{\mathbf{g}} = \sum_{e=1}^{N}\left[-(2\rho_{\mathrm{e}}^{\mathrm{n}}\mathbf{u}_{\mathbf{e}}{}^{T}\mathbf{K}_{\mathbf{e}})\mathbf{H}_{\mathbf{e}}\right] \qquad (7)$$

Assim, a expressão final para a sensibilidade da função objetivo, com relação à pseudodensidade fica:

$$\frac{d\Phi}{d\rho} = (n+1)\rho_i^n \mathbf{u}_i^T \mathbf{K}_i^0 \mathbf{u}_i + \left(\mathbf{H}_i \boldsymbol{\lambda}_{\mathbf{g}}\right)^T \mathbf{K}_i^0 \mathbf{u}_i \qquad (8)$$

3.1.4. MODIFICAÇÕES DA ABORDAGEM

Conforme o código foi sendo desenvolvido no MATLAB, constatou problemas com o condicionamento da matriz de rigidez global, $\mathbf{K}_{\mathbf{g}}$, mais especificamente em (7), quando do cálculo do vetor adjunto. Desta forma, visando a obtenção de topologias, sem grandes prejuízos à análise em curso, optou-se por modificar alguns princípios da abordagem de maximização de energia de deformação para uma formulação próxima da maximização de deslocamento de saída, conforme dedução que segue

$$\begin{aligned}
\frac{d\boldsymbol{\Phi}}{d\rho} &= \mathbf{H}^{(2)}\frac{\partial \mathbf{U}}{\partial \rho_e} = \mathbf{H}^{(2)}\frac{\partial(\mathbf{K}^{-1}\mathbf{F})}{\partial \rho_e} \qquad (9)\\
&= \mathbf{H}^{(2)}\frac{\partial(\mathbf{K}^{-1})}{\partial \rho_e}\mathbf{F}\\
&= \mathbf{H}^{(2)}\left(-\mathbf{K}^{-1}\frac{\partial(\mathbf{K}\ )}{\partial \rho_e}\mathbf{K}^{-1}\right)\mathbf{F}\\
&= -\mathbf{H}^{(2)}\mathbf{K}^{-1}\frac{\partial(\mathbf{K}\ )}{\partial \rho_e}\mathbf{U}^{(1)},
\end{aligned}$$

onde $\mathbf{H}^{(2)}$ é um operador de localização.

A modificação adotada para a eq. (8) será no que se refere ao segundo termo desta, que incluiu o vetor adjunto global. Ao invés de se utilizar aquela formulação, opta-se pela definição em (10) de um vetor de pseudoforça, de modo a se evitar o cálculo do vetor adjunto global, $\boldsymbol{\lambda}_{\mathbf{g}}$.

$$\mathbf{K}^{-1}\mathbf{H}^{(2)} = \mathbf{U}^{(2)} \qquad (10)$$

Assim, alterando-se a sensibilidade da função objetivo, dada em (8), levando-se em conta as considerações expostas em (9) e (10), a sensibilidade da função objetivo passa a ser:

$$\frac{d\boldsymbol{\Phi}}{d\rho} = (p+1)\rho_i^p \mathbf{u}_e^{(2)^T}\mathbf{K}_i^0\mathbf{u}_e^{(1)}. \qquad (11)$$

3.1.5. RESTRIÇÕES

Outra modificação feita na abordagem se refere à quantidade de restrições. No presente trabalho, por simplicidade, a única restrição ativa para o problema abordado é com relação ao volume, cuja formulação segue:

$$V = \sum_{e}^{N} \rho_e V_e \qquad (11)$$

a correspondente sensibilidade fica

$$\frac{dV}{d\rho_i} = V_i \qquad (12)$$

3.1.6. PROGRAMAÇÃO MATEMÁTICA

Por simplicidade, este trabalho adotou a rotina similar ao método de Bissecção, para atualizar os valores de pseudodensidades no processo iterativo:

```
function
[xnew]=OC2(nelx,nely,passive,x,volfrac,dPHI)
l1 = 0; l2 = 100000; move = 0.005;n=0.8;

while ((l2-l1)/(l2+l1)>1e-9)&(l2>eps)
  lmid = 0.5*(l2+l1);
  xnew = max(0.01,max(x-
move,min(1.,min(x+move,x.*max(1e-10,-
PHI./lmid).^n))));
  if sum(sum(xnew)) - volfrac*nelx*nely > 0;
    l1 = lmid;
  else
    l2 = lmid;
  end
end
```

(13)

onde *xnew* é a pseudodensidade na saída da iteração, *x* é a pseudodensidade na entrada, *volfrac* é a fração de volume, *move* é o passo de variação de parâmetros, o expoente *n* é um valor arbitrário para que tem por objetivo facilitar a convergência e os valores *l1* , *l2* e *lmid* são os limites inferior, superior e intermediário do multiplicador de Lagrange. Sabe-se que o volume da estrutura, dado por *sum(sum(xnew))*, é uma função decrescente com relação ao multiplicador de Lagrange [Bendsøe e Sigmund, 2004]. O algoritmo de bissecção é inicializado com valores arbitrários para os limites superior e inferior dos multiplicadores de Lagrange. Os intervalo que limita o multiplicador de Lagrange é dividido ao meio sucessivamente até que esteja dentro do critério de convergência.

3.1.7. FILTROS DE INDEPENDÊNCIA DE MALHA

Em [Cardoso e Fonseca, 2003], se discutem maneiras de controlar a complexidade da topologia a ser gerada, visando evitar a formação de fenômenos como a instabilidade de tabuleiro e a interdependência da topologia com o refino da malha de elementos finitos e se propõe um novo tipo de filtro que seria aplicado sobre os limites móveis do critério de ótimo do modelo Programação Linear Sequencial (PLS). As figuras 3.1.7 a), b) e c) ilustram a formação de instabilidade de tabuleiro e a dependência da topologia em relação ao refino da malha.

No entanto, ao invés de se seguir a recomendação de filtro dada na conclusão de [Cardoso e Fonseca, 2004], até porque o critério de ótimo não é o método PLS, o filtro adotado neste trabalho é do tipo espacial e depende de um dado de entrada arbitrário, $r_{\min}$,

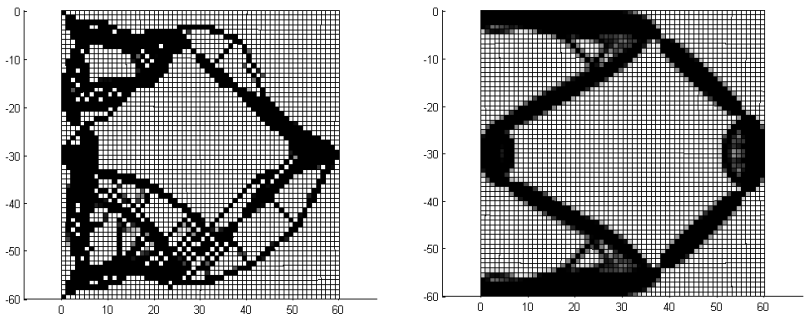
desenvolvido no código descrito em [Sigmund, 2001]. Este filtro modifica a sensibilidade de um elemento *k*, conforme equacionamento que segue:

$$\frac{\partial \hat{\Phi}}{\partial \rho_k} = \frac{1}{\rho_k \sum_{i=1}^{N} \hat{H}_i} \sum_{i=1}^{N} \hat{H}_i \rho_i \frac{\partial \Phi}{\partial \rho_i} \qquad (13)$$

$$\hat{H}_i = r_{\min} - dist(k,i), \{i \in N \mid dist(i,k) \leq r_{\min}\}, k = 1,...,N.$$

o operador *dist(k,i)* é definido como a distância entre o centro do elemento *k* e o centro do elemento *i*, o operador de convolução $\hat{H}_i$ é nulo fora do filtro e decai linearmente como aumento da distância do elemento *k*. Nota-se que a sensibilidade não se altera se $r_{\min}$ é igual a zero e a malha terá o mesmo valor de sensibilidade se $r_{\min}$ se aproxima do infinito.

Outro aspecto que é importante frisar no filtro de sensibilidades é a sua ação de controle de complexidade de topologia. Sem a utilização desta ferramenta não seria possível interpretar um resultado viável e o método, como um todo, estaria comprometido com seu viés de engenharia.

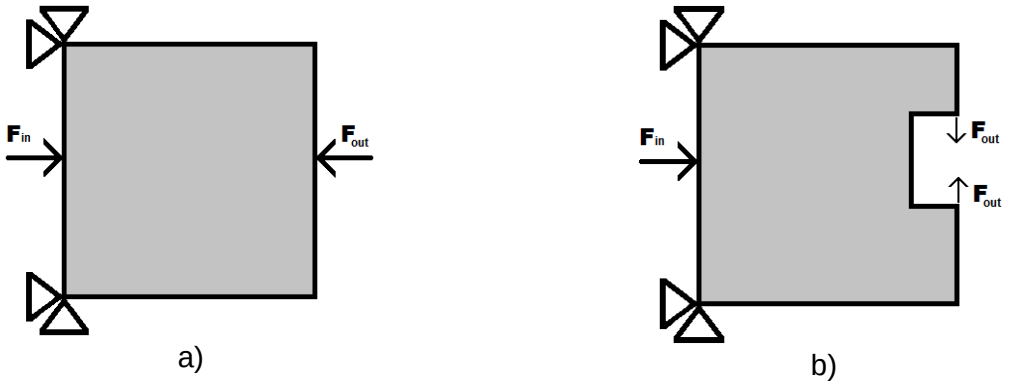


a) b)

Figura 3.1.7.1 Em uma malha de 60x60 elementos a) Topologia gerada com $r_{\min}$ =0.8, indica o surgimento de instabilidade de tabuleiro em várias regiões. Em b) Topologia gerada para $r_{\min}$ =1.1, demais variáveis inalteradas com respeito a a).

4. APRESENTAÇÃO DO PROBLEMA

Os problemas selecionados foram os mecanismos flexíveis *Inversor de Força* e o *Gripper*. Em ambos o domínio material é discretizado em malha quadrada de 3.600 elementos finitos, do tipo Q4, de dimensões 60x60 mm. Os pontos fixos nos domínios são as extremidades superior e inferior dos lados esquerdos das malhas, conforme ilustração das figura 4.1 a) e b). As forças F_{in} e F_{out} são prescritas, ambas de módulo 200 N, no caso a), aplicadas na mesma direção, porém em sentidos opostos. No caso b), F_{out} é um par de vetores de módulo 100 N, na direção vertical e também em sentidos contrários. No caso b) também há uma distância de 10 mm, entre os vetores F_{out}, correspondente ao curso de fechamento do *Gripper.*



a) b)

Figura 4.1 a) Esquema inicial do domínio do Inversor de Força. Em b) Esquema inicial do domínio do Gripper.

4.1. APRESENTAÇÃO DO CÓDIGO

Até aqui foi apresentado um breve descritivo de cada etapa de resolução do algoritmo. O diagrama de blocos do programa, relacionando uma etapa a outra, é conforme segue:

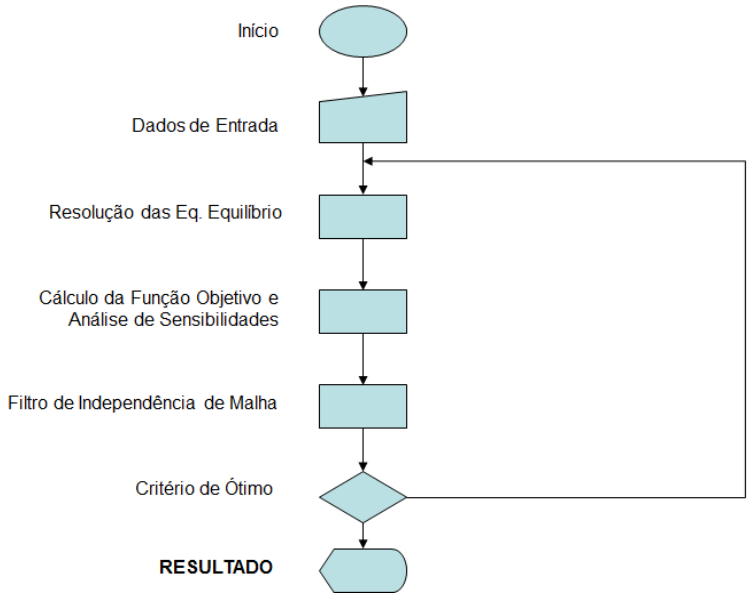


Figura 4.1.1 Diagrama de Blocos do código.

Na inicialização se define um valor arbitrário de pseudodensidade para todos os elementos da malha. No código em questão, assume-se que o valor é igual à fração de volume.

Os dados de entrada, no programa, são *iter* , contendo o número de iterações, *nelx*, com o número de elementos em na direção horizontal, *nely*, número de elementos na direção vertical, *volfrac*, a fração de volume, ou seja, quanto que se quer manter do volume inicial da malha, no volume final da estrutura. Ainda tem-se a *penal*, que é um valor de penalização das pseudodensidade, neste trabalho o valor 2 apresentou bons resultados em termos de convergência e *rmin*, explicado no item 3.1.7.

A resolução das equações de equilíbrio é feita a partir de resolução da malha em elementos finitos. Os dados de entrada são as constantes materiais, modificadas pela

pseudodensidade. Como saída o programa apresenta os vetores de deslocamento global, $\mathbf{U}$,e a matriz de rigidez global, $\mathbf{K_g}$.

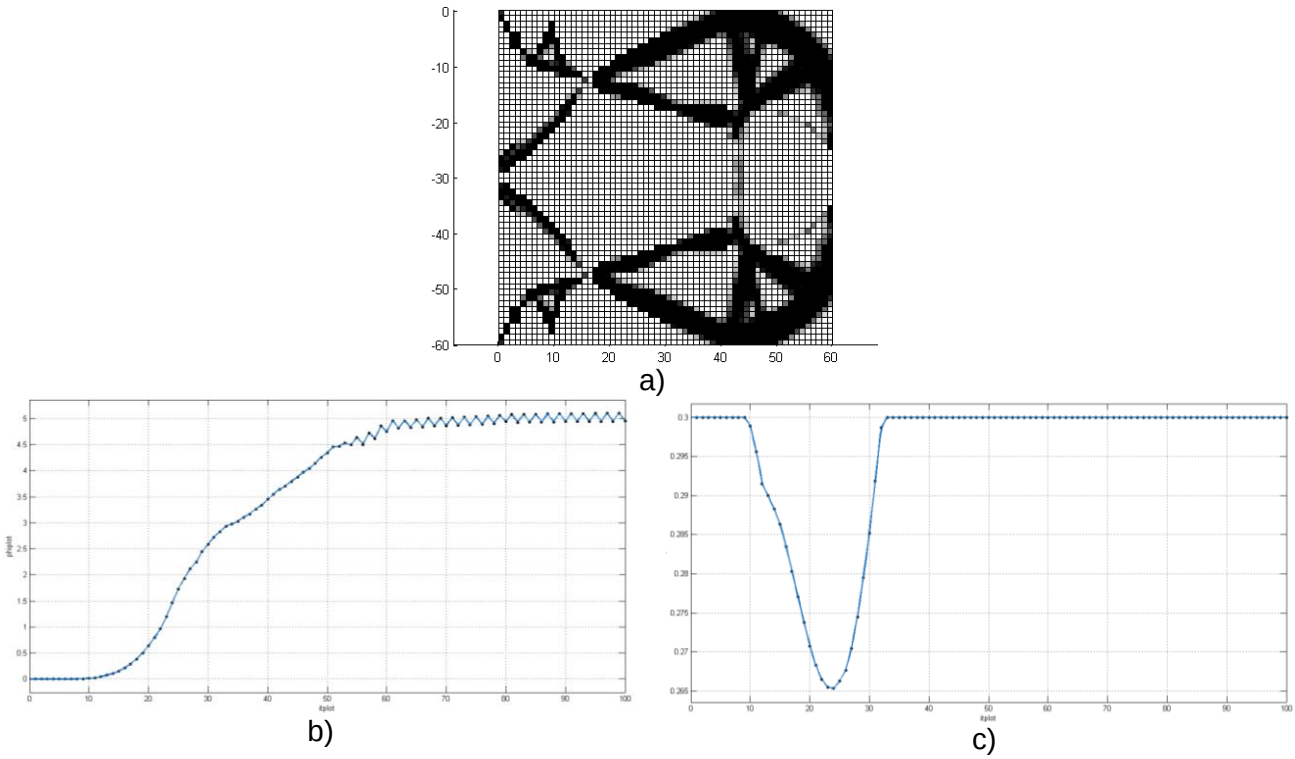
Demais passos são o cálculo do valor da função objetivo, $\mathbf{\Phi}$, e a sua respectiva sensibilidade, $\frac{d\mathbf{\Phi}}{d\rho}$, a partir dos valores obtidos para o vetor de deslocamento.

Parte-se, então, para o filtro de sensibilidade, explicado no item 3.1.7, e o critério de ótimo, explicado no item 3.1.6.

O próximo passo é a plotagem do resultado em imagem do tipo *raster*.

5. RESULTADOS E DISCUSSÃO

O primeiro resultado obtido foi para o *Gripper*, cujo domínio foi apresentado na figura 4.1 b).



a)

b) c)

Figura 5.1 a) Topologia obtida para o *Gripper*. Em b), tem-se gráfico de número de iterações pelo valor da Função Objetivo. Em c), tem-se gráfico de número de iterações pelo valor do da fração de volume.

Como dados de entrada para a topologia gerada na fig. 5.1 a) tem-se *iter* =100, *nelx*=*nely*=60, *volfrac*=0,3, *penal*=2 e *rmin*=1,15. Outros dado relevante é o passo escolhido para o critério de ótimo, *move* = 0,025.

Nota-se na fig. 5.1 b) que a função objetivo foi maximizada. Em fig. 5.1 c), percebe-se a conduta do critério de ótimo que, após algumas interações iniciais, passa a aumentar o valor da função objetivo sem obedecer a restrição de volume. Primeiramente a restrição decai, indicando movimento ascendente dos limites de bissecção impostos ao multiplicador de Lagrange. Chega-se a um patamar mínimo e então os valores de restrição passam a subir, indicando agora movimento descendente dos limites de bissecção. Só quando

o tem-se o retorno ao valor de restrição colocado inicialmente, ou seja, *volfrac*=0,3, é que se pode dizer que o valor da função objetivo está no seu patamar máximo.

Nota-se que a fig. 5.1 a) apresenta falhas, como manchas cinzentas, e rótulas, ou seja, pontos de flexibilidade concentrada. Pode-se dizer que, para a falha das manchas, o tipo de filtro não é o mais adequado para se obter este tipo de topologia. A falha na forma de rótula indica que o programa proposto não está maximizando a energia de deformação da estrutura.

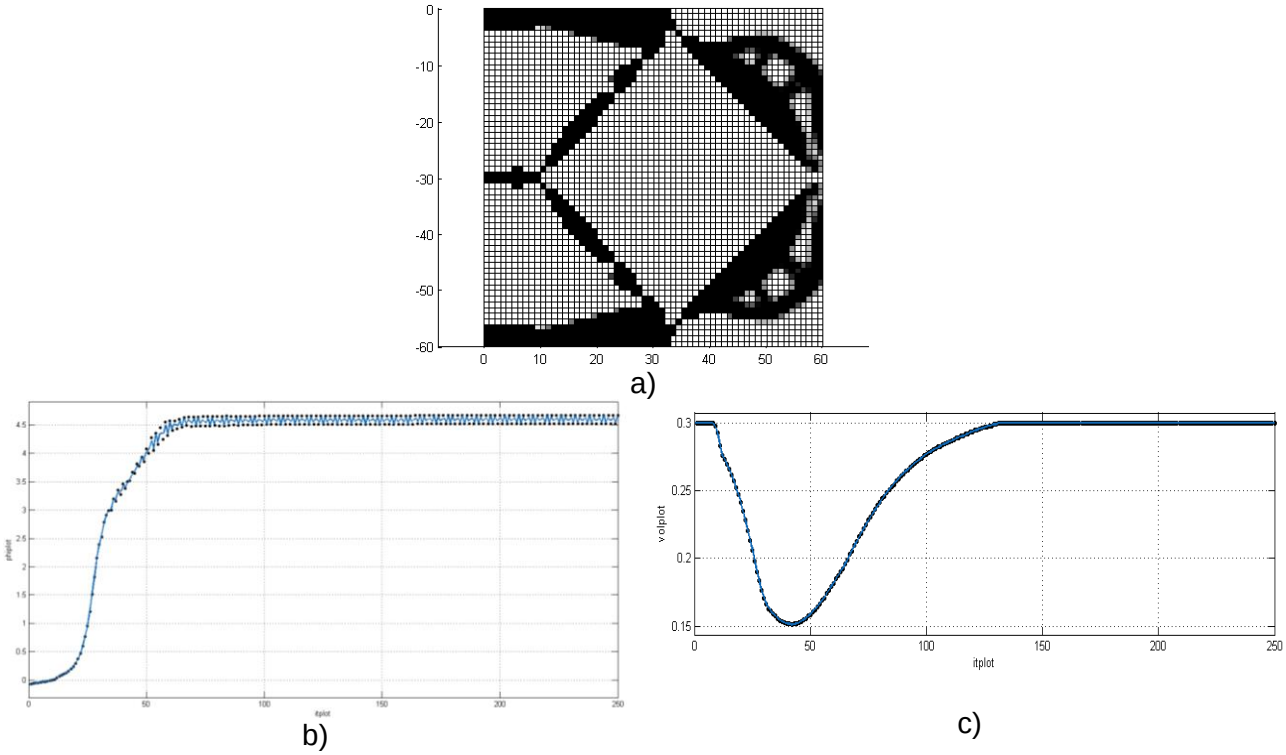


Figura 5.2 a) Topologia obtida para o *Inversor de Força*. Em b), tem-se gráfico de número de iterações pelo valor da Função Objetivo. Em c), tem-se gráfico de número de iterações pelo valor do da fração de volume.

Como dados de entrada para a topologia gerada na fig. 5.2 a) tem-se *iter* =250, *nelx*=*nely*=60, *volfrac*=0,3, *penal*=2 e *rmin*=1,15. O passo escolhido para o critério de ótimo foi *move* = 0,025.

De forma análoga ao exemplo anterior, na fig. 5.2 b) tem-se que a função objetivo foi maximizada. Porém, nota-se em fig. 5.2 c) que o valor a conduta do critério de ótimo é muito mais lenta e, por este motivo, houve necessidade de se aumentar a quantidade de iterações. A topologia resultante se mostrou bastante complexa, embora não apresente regiões cinzentas. Esse indicativo demonstra que o filtro está sendo bem aplicado, o que motivou a geração da próxima topologia, do mesmo tipo, porém com valores de iteração maiores, *rmin* maior e passo menor.

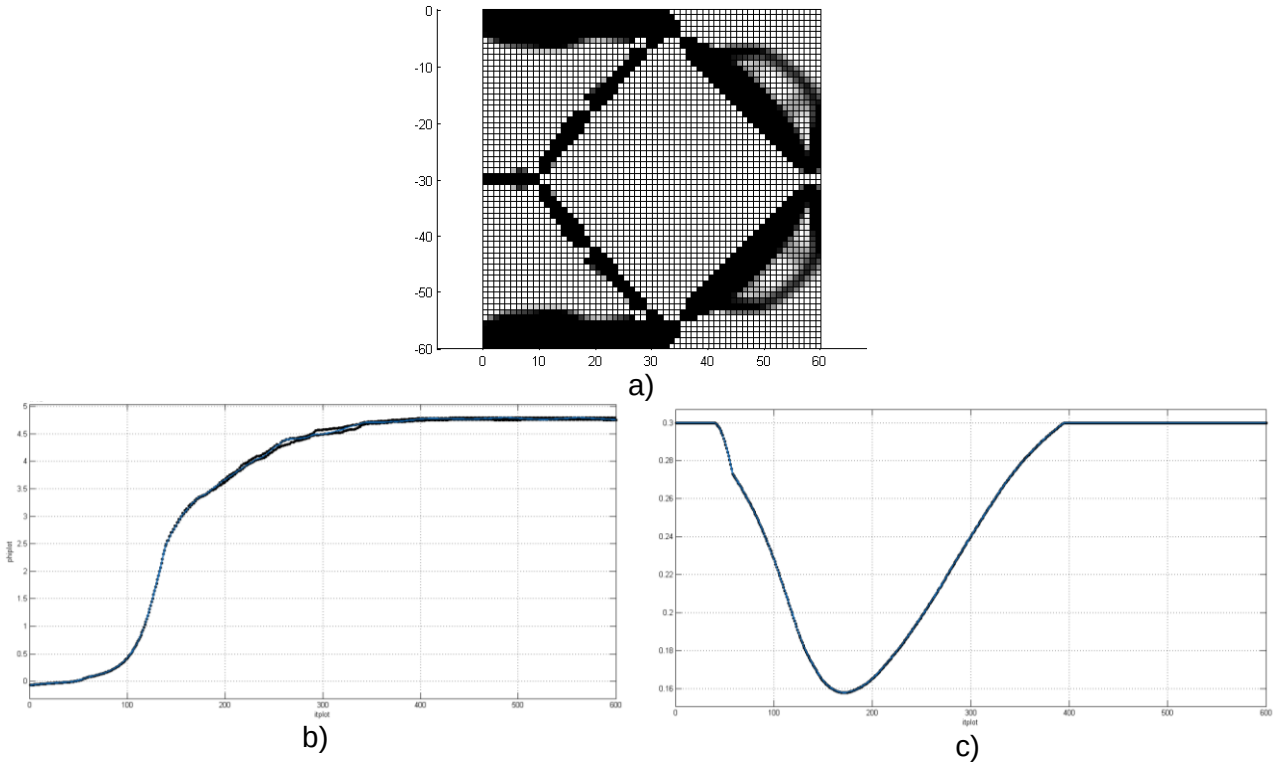


Figura 5.3 a) Topologia obtida para o *Inversor de Força*. Em b), tem-se gráfico de número de iterações pelo valor da Função Objetivo. Em c), tem-se gráfico de número de iterações pelo valor do da fração de volume.

Como dados de entrada para a topologia gerada na fig. 5.3 a) tem-se *iter* =600, *nelx*=*nely*=60, *volfrac*=0,3, *penal*=2 e *rmin*=1,8. O passo escolhido para o critério de ótimo foi *move* = 0,0025.

Na fig. 5.3 b) tem-se que a função objetivo foi maximizada, com menor incerteza do que nos exemplos anteriores, o que se deve principalmente à redução do passo. A conduta do critério de ótimo é ainda mais lenta que no caso anterior, na fig. 5.3 c). A quantidade de iterações foi significativamente maior do que nos outros casos. Pode-se perceber, no entanto, que a topologia resultante, apesar das rótulas, se mostrou um pouco menos complexa que no caso anterior. As regiões cinzentas apresentadas são admissíveis, podendo ser reinterpretadas no caso de se construir um mecanismo com esta topologia.

6. CONCLUSÕES

Pelos resultados obtidos podemos concluir que há uma forte influência dos dados de entrada na formação da topologia, sobretudo o parâmetro do filtro de sensibilidade, este que praticamente define a viabilidade desta topologia como estrutura.

As alterações feitas com o objetivo de diminuir o custo computacional alteraram significativamente os resultados, não podendo-se dizer que o que o algoritmo está fazendo é a maximização de energia de deformação, mas algo próximo à maximização de energia mútua ou maximização de deslocamento de saída.

Embora não tenha sido possível concluir com êxito o objetivo do trabalho, pode-se dizer, ainda assim, que a opção pela programação de alto nível para observar aspectos conceituais da otimização topológica foi útil e apresentou topologias bem interessantes para os exemplos escolhidos.

REFERÊNCIAS BIBLIOGRÁFICAS

Liberzon, D., "**Calculus of Variations and Optimal Control Theory – A Concise Introduction",** Princeton University Press, p. 28-32, 2012.

Michell, A. G. M., "**The limits of economy of material in frame-structures",** Philosophical Magazine, Vol. 8(47), p. 589-597, 1904.

Sigmund, O., "**Topology optimization: A tool for the tailoring of structures and materials",** Philosophical Transactions – Mathematical Physical and Engineering Sciences, 358(1765): 211-228, 2000.

Haftka, R.T., Gürdal,Z.,**"Elements of Structural Optimization**. **Series: Solid Mechanics and Its Applications**", Vol. 11. 3rd rev. and expanded ed., 1992

Bendsøe, M. P.; Sigmund, O.; "**Topology Optimization – Theory, Methods and Applications",** Springer-Verlag, 2nd edition, 2004.

Cardoso, E. L.; Fonseca, J. S. O.; "**Strain energy maximization approach to the design of fully compliant mechanisms using topology optimization",** Latin American Journal of Solids and Structures, Vol. 8(47), p. 263-275, 2004.

Sigmund, O.; "**A 99 line topology optimization code written in Matlab",** Springer-Verlag, Structural and Multidisciplinary Optimization, Vol. 21, p. 120-127, 2001.

Cardoso, E. L.; Fonseca, J. S. O.; "**Complexity Control in the Topology Optimization of Continuum Structures",** ABCM, Vol. XXV, p. 293-301, July-September 2003.

APÊNDICE I – CÓDIGO MATLAB

```
function maxendef(iter,nelx,nely,volfrac,penal,rmin)

%% INICIALIZAÇÃO
x(1:nely,1:nelx) = volfrac;
PHI(1:nely,1:nelx)=0;
dPHI(1:nely,1:nelx)=0;
loop = 0;
change = 1.;

volplot=sparse(iter,1);
itplot  =sparse(iter,1);

%% PROCEDIMENTO ITERATIVO
while loop <iter
  loop = loop + 1;
  xold = x;
  tic;
 %% CONDIÇÕES DE CONTORNO

dofs        = 2*(nelx+1)*(nely+1);
alldofs     = [1:dofs];
passive     = zeros(nelx,nely);
fixeddofs   = union([1:1:2],[2*(nely+1)-1:1:2*(nely+1)]);
freedofs    = setdiff(alldofs,fixeddofs);


%% Primeira iteração de Elementos Finitos
  [U,Uout,K,F]=FE(dofs,alldofs,freedofs,fixeddofs,nelx,nely,x,penal);
  [KE] = lk;

%% Laços definidos Elemento-a-Elemento
  for ely = 1:nely
    for elx = 1:nelx
      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];

 %% MATRIZ DE LOCALIZAÇÃO [He] DA FUNÇÃO OBJETIVO
      for i = 1:dofs
       for j = 1:8
         if edof(j)==alldofs(i)
            He(j,i)=1;
         else
            He(j,i)=0;
         end
       end
      end

 %% DESLOCAMENTO DO ELEMENTO
   UE = (He*U);
   Ue1=UE(:,1);
   Ue2=UE(:,2);
```

```
 %% FUNÇÃO OBJETIVO
PHI(ely,elx)= ((x(ely,elx)^(penal))*Ue1'*KE*Ue2);


 %% SENSIBILIDADE DA FUNÇÃO OBJETIVO
dPHI(ely,elx)= ((((penal+1)*x(ely,elx)^(penal))*Ue1'*KE*Ue2));


    end
  end


% FILTRO DE SENSIBILIDADE
  [dPHI]   = check(nelx,nely,rmin,x,dPHI);

% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
  [x]    = OC2(nelx,nely,passive,x,volfrac,dPHI);

volplot(loop,1)=sum(sum(x))/(nelx*nely);
phiplot(loop,1)=sum(sum(PHI));
itplot(loop,1)  =loop;

% PRINT RESULTS
  change = max(max(abs(x-xold)));
  disp([' It.: ' sprintf('%4i',loop) ' Obj.: '
sprintf('%10.4f',1e4*sum(sum(PHI))) ...
       ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
        ' ch.: ' sprintf('%6.3f',change )...
        ' it. time: ' sprintf('%6.3f',toc )])


if loop==iter
% PLOT DENSITIES
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
end

cftool(itplot,volplot);
cftool(itplot,phiplot);
end
```