



Igor Tavares Girsas

Refinamento auto-adaptativo de malhas poligonais com base no Método dos Elementos Finitos

Trabalho de Conclusão de Curso

Departamento de Engenharia Mecânica
Programa de Graduação em Engenharia Mecânica

Prof. Ivan F.M. Menezes

Orientador

Departamento de Engenharia Mecânica — PUC-Rio

Rio de Janeiro, 2 de Julho, 2015

The most difficult thing is the decision to act, the rest is merely tenacity. – Amelia Earhart

AGRADECIMENTOS

A minha família, pelo constante e incondicional apoio, apesar de todos os tropeços.

Ao meu orientador, Prof. Ivan F.M. Menezes, pelo suporte, incentivo e correções. Por acreditar e aceitar o projeto, mesmo com um curto prazo disponível.

Ao Dr. Anderson Pereira, por todo auxílio no desenvolvimento do código, pelas discussões e esclarecimentos, além da disponibilidade em todos os momentos críticos.

Aos amigos de longa data Erick Mascarenhas, Henrique Saraiva e Nathalia Matola, pelo companheirismo, estímulo e paciência ao longo dos anos.

Aos amigos João Luiz Ramos e Luiz Santarelli, que com seu brilhantismo, me inspiraram e ajudaram no decorrer da faculdade, além de fazerem de suas, a minha segunda casa, sempre que preciso.

Ao amigo João Virgolino, por toda grosseria, risadas, explicações matemáticas complexas e Pires em dias aleatórios.

Aos amigos Cristina Sánchez, Davide Cilli, Geerjan Wolfs e Jonatan Snyders pelas grandes aventuras e histórias. Grazie mille e Dankjewel!

RESUMO

A análise de concentração de tensões é um tema de extrema importância em diversas áreas da engenharia, porém de difícil solução analítica. Com o desenvolvimento dos microcomputadores, a análise numérica destes problemas, por meio do Método dos Elementos Finitos, tornou-se comumente difundida.

Neste trabalho, investiga-se a evolução dos campos de tensão em uma malha poliédrica por meio de técnicas MEF, visando obter numericamente o valor das concentrações de tensão induzidas pela geometria do domínio escolhido.

Primeiro, são definidos os domínios de interesse de forma implícita. Elementos poliedrais são gerados, na sequência, pela triangulação de Delaunay e diagrama de Voronoi, sendo estes analisados de pela função de distância sinalizada, para verificação do pertencimento ao domínio.

Após a etapa inicial de geração de elementos, é feita simulação por elementos finitos, e então obtém-se as tensões médias em cada elemento, pela média das tensões de cada nó do mesmo, a partir do critério de Von Mises. Os elemento com maiores tensão, são automaticamente selecionado entrando numa rotina de refino da malha, até que se obtenha uma melhor percepção dos campos de tensão.

Finalmente, apresentam-se diversos exemplos que demonstram os recursos e o potencial da abordagem proposta e validam os os resultados obtidos.

Palavra Chaves: Concentração de tensões, Malhas poliédricas, Elementos Finitos.

ABSTRACT

Stress concentration analysis is a subject of utmost importance in several areas of engineering, nevertheless it has difficult analytical solution. Due to the development of microcomputers, the numerical analysis of such problems, using Finite Element Method, has become widespread commonly.

This paper investigates the evolution of stress fields in a polyhedral mesh, through FEM techniques, to obtain numerical results of stress concentrations due to the geometry changes in a chosen domain.

First, the areas of interest are implicitly defined. Polyhedral elements are generated as a result of Delaunay triangulation and Voronoi tessellation, been latter analyzed by the signed distance function to check if they belong to the chosen domain.

After the initial stage of element generation, finite elements simulation is performed, in order to obtain the mean stress values on each element, by averaging the stress of each node of a single element, based on Von Mises criteria. Elements with high stress are automatically selected, entering a mesh refinement routine to obtain a better understanding of the stress fields.

Finally, several examples are presented, demonstrating the features and the potential of the proposed approach, and also validating the results.

Keyword: Stress concentration; Polyhedral meshes; Finite element.

1	Introdução	12
1.1	Motivação e Objetivos	12
1.2	Escopo do Trabalho	13
2	Concentração de Tensão	15
2.1	Fator de Concentração de Tensão	17
2.2	Solução Analítica de Kirsch	18
2.3	Solução Analítica de Inglis	19
2.4	Estimativa de K_t pelo Método dos Elementos Finitos	20
3	Elementos Finitos	22
3.1	Cálculo dos Deslocamentos U	22
3.2	Formulação Isoparamétrica	25
3.3	Malha Estruturada e não Estruturada	27
4	Representação de Domínios	29
4.1	Função Distância Sinalizada (FDS) ou Signed Distance Function (SDF)	29
4.2	Representação Implícita e Criação de Domínios	30
5	Criação de Malhas Não Estruturadas	31
5.1	Diagrama de Voronoi ou Voronoi Tessellation	31
5.2	Triangulação de Delaunay	32
5.3	Algoritmo de Lloyd	33
5.4	Diagrama de Voronoi Centroidal ou Centroidal Voronoi Tessellation	34
6	Refinamento Automático de Malha	36
7	Resultados	39
7.1	Michell Domain	44
7.2	Ele Domain	47
8	Conclusão	49
	Referências	50
A	Apêndice	53

A.1	EleDomain	53
A.2	UnitSquareDomain	55
A.3	MichellDomain	56
A.4	Refinamento Automático de Malha	57

LISTA DE FIGURAS

1	Efeito da concentração de tensão	15
2	Representação do estado plano de tensões	16
3	Linhas de força desviando de descontinuidades na geometria	17
4	Placa plana com furo central	18
5	Placa plana com furo central elíptico	20
6	Comparação entre coordenadas	25
7	Elemento degenerado com quatro nós	26
8	Comparação entre estruturas de malha	27
9	Triangulação de Delaunay e Diagrama de Voronoi	32
10	Evolução das células de Voronoi dadas três iterações do algoritmo de Lloyd	34
11	Evolução da seleção de elementos de interesse	37
12	Modelo computacional para placa com furo central.	39
13	Campo de tensões iniciais para placa com furo central	41
14	Campo de tensões finais para placa com furo central	41
16	Evolução de K_t com 4 iterações	42
17	Evolução da malha, para placa com furo central, por meio do refinamento automático baseado no valor das tensões	43
18	Michell Domain. Condições de contorno	44
19	Zoom do campo de tensões inicial, para Michell Domain	44
20	Zoom do campo de tensões final, para Michell Domain, após 4 etapas de refinamento de malha	45
21	Evolução da malha, para Michell Domain, por meio do refinamento automático baseado no valor das tensões	46
22	EleDomain. Condições de contorno	47
23	Evolução da malha por meio do refinamento automático baseado no valor das tensões.	48

LISTA DE TABELAS

15	Evolução de número de elementos e da concentração de tensão	41
-----------	---	----

LISTA DE SÍMBOLOS

\mathbf{b}	Força de corpo
\mathbf{B}	Matriz deformação-deslocamento
\mathbf{C}	Matriz constitutiva
d_Ω	Fronteira de um domínio em um sistema discretizado
$\mathbf{f}^{(\varphi)}$	Vetor de forças agindo em um dado elemento
\mathbf{F}	Vetor de forças globais
\mathbf{K}	Matriz de rigidez global
K_t	Fator de concentração de tensão
$\mathbf{K}^{(\varphi)}$	Matriz de rigidez elementar
\mathbf{N}	Matriz de funções de forma
N_i^φ	Função de forma de um dado elemento
$\mathbf{q}^{(\varphi)}$	Vetor de forças elementares agindo em um dado elemento
s_Ω	Função distância
\mathbf{SN}	Matriz deformação-deslocamento
\mathbf{t}	Força de tração agindo em um dado elemento
\mathbf{u}	Vetor dos deslocamentos locais
$\mathbf{u}^{(\varphi)}$	Aproximação do campo dos deslocamentos \mathbf{u}
$\hat{\mathbf{u}}^{(\varphi)}$	Deslocamento avaliado nos nós de um dado elemento
\mathbf{U}	Vetor dos deslocamentos globais
\hat{V}_i	Células de Voronoi
\mathbf{z}^*	Centróide de uma célula de Voronoi para CVT
z_i	Centróide de uma célula de Voronoi
$\alpha_i(x, y)$	Coordenadas cartesianas
β	Região genérica e um dado domínio
γ_{xy}	Deformação de cisalhamento no plano xy
δ, ξ	Componentes das coordenadas ortonormais elíptico-hiperbólicas
ε	Deformação normal
$\boldsymbol{\varepsilon}$	Vetor das deformações normais
$\boldsymbol{\varepsilon}_0$	Vetor das deformações normais iniciais
$\varepsilon_{x,y}$	Deformações normais nas direções x e y
λ	Área de interesse para refinamento automático
ρ	Raio mínimo na raiz do entalhe
ν	Coefficiente de Poisson
σ	Tensão normal
$\boldsymbol{\sigma}$	Vetor das tensões normais
$\boldsymbol{\sigma}_0$	Vetor das tensões normais iniciais
$\sigma_{x,y}$	Tensões normais nas direções x e y
$\sigma_{1,2}$	Tensões principais
$\sigma_{m\acute{a}x}$	Tensão máxima
σ_n	Tensão normal
σ_{rr}	Tensão normal, em coordenadas polares, na direção r
$\sigma_{\theta\theta}$	Tensão normal, em coordenadas polares, na direção θ

$\tau_{r\theta}$	Tensão de cisalhamento, em coordenadas polares, no plano $r\theta$
τ_{xy}	Tensão de cisalhamento no plano xy
$\emptyset(s, t)$	Coordenadas isoparamétricas
ω	Densidade
Ω	Domínio de um sistema discretizado
φ	Número do elemento finito

INTRODUÇÃO

1.1- MOTIVAÇÃO E OBJETIVO

Peças mecânicas e elementos estruturais, muitas vezes têm características que causam mudanças bruscas de geometria. Sob cargas, essas mudanças induzem um aumento nos campos de tensão locais das peças de forma bastante significativa, sendo estes, geralmente, os locais onde as peças começam a falhar. A análise dos campos de tensão não é em nada trivial, existindo poucas soluções analíticas, restritas a geometrias simples.

Com o crescente desenvolvimento e barateamento dos microcomputadores, a técnica dos Elementos Finitos vem se destacando dentro da engenharia, passando a ser amplamente utilizada na avaliação do comportamento de estruturas.

Apesar dos significativos avanços alcançados no desenvolvimento da teoria matemática e de algoritmos para o Método dos Elementos Finitos, a discretização de um determinado problema depende, na maioria dos casos, do bom senso do analista e das experiências adquiridas na solução de problemas similares. No caso de os resultados serem julgados ruins, a discretização deve ser refeita. Visando solucionar este problema, diversas técnicas de refino automatizado de malha vem sendo propostas por diversos autores [1].

Para propor um método que facilite a análise de concentração de tensões, foi desenvolvido um algoritmo baseado no código PolyMesher, proposto por Talischi [3]. Seus princípios se fundamentam na descrição implícita de domínios genéricos e na geração de elementos finitos não estruturados pela técnica denominada “Voronoi Tessellation”, seguida de uma suavização pelo método iterativo de Lloyd [15]. Esta suavização tem por finalidade, gerar elementos mais regulares em termos geométricos. Este trabalho se propõe a ampliar e aplicar este código no tocante ao refinamento automático da malha, para então, analisar o comportamento de estruturas mecânicas sobre carregamentos uniformes, comparando com as soluções analíticas disponíveis na literatura, e aplicando posteriormente o código a estruturas com geometrias genéricas.

1.2 - ESCOPO DO TRABALHO

Para facilitar o entendimento, este trabalho foi dividido em diversos capítulos, cujo conteúdo é apresentado a seguir.

No capítulo 2 são introduzidos os conceitos relativos à concentração de tensão, assim como as soluções analíticas obtidas por Kirsh e Inglis.

No capítulo 3, as técnicas básicas referentes a elementos finitos, elementos isoparamétricos, malhas estruturadas e não estruturadas, são brevemente apresentadas.

No capítulo 4, os conceitos referentes à geração de malhas poliédricas, tais como a representação implícita de domínios genéricos são apresentados.

No capítulo 5, as técnicas referentes à geração de malhas poliédricas não estruturadas como Triangulação de Delaunay (TD), Diagrama de Voronoi (DV) e Centroidal Voronoi Tessellation (CVT) são descritas.

No capítulo 6, o refinamento da malha, assim como a abordagem para determinação dos elementos a serem refinados é apresentada, sendo os resultados obtidos discutidos no capítulo 7.

Finalmente, o capítulo 8 é reservado para a apresentação das conclusões obtidas por meio das formulações e metodologias utilizadas neste trabalho. Em seguida são apresentadas algumas sugestões para o desenvolvimento de pesquisas futuras.

CONCENTRAÇÃO DE TENSÃO

Tensão é a grandeza física que expressa as forças internas que partículas vizinhas de um material contínuo exercem umas nas outras, sendo definida como a relação entre a força F aplicada a uma área unitária A e própria a área A [19], ou seja: $\sigma = \frac{F}{A}$.

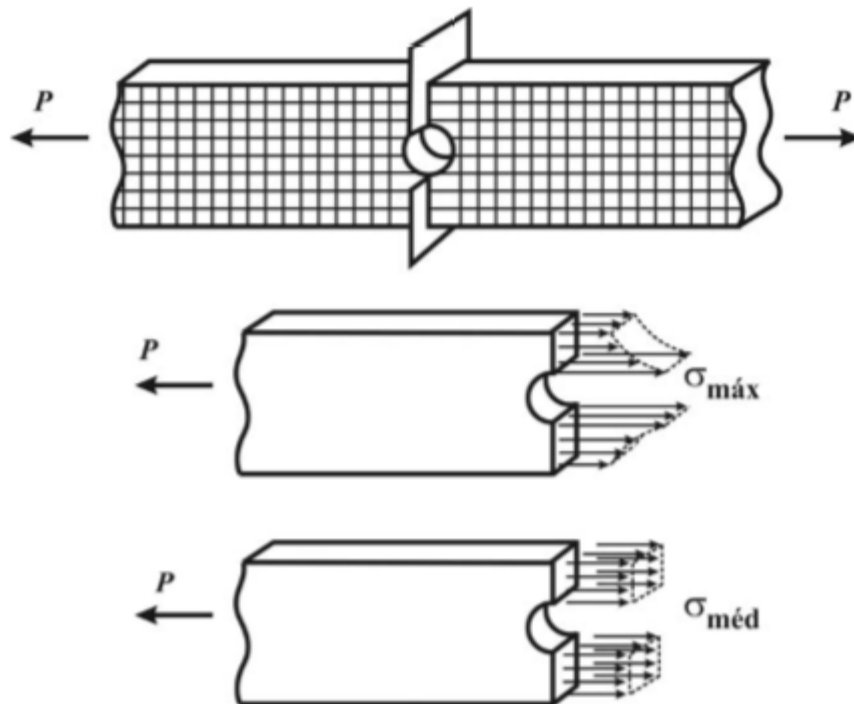


Figura 1 Efeito da concentração de tensão. A tensão média $\sigma_{\text{méd}}$ é aquela calculada desconsiderando a concentração de tensão. A tensão máxima $\sigma_{\text{máx}}$ é aquela que realmente está presente na estrutura.

O princípio de Saint Venant, ou princípio das cargas estáticas equivalentes, estabelece que as tensões de um corpo, em pontos “suficientemente distantes” de variações bruscas de geometria ou aplicação de cargas concentradas, depende apenas da resultante das forças aplicadas e não da distribuição das mesmas[2]. Com isso, podemos afirmar que as tensões na vizinhança dos pontos acima citados, não são aquelas calculadas globalmente para o elemento pelas formulações:

$$\sigma_{x,y} = \frac{E}{1 - \nu - 2\nu^2} [(1 - \nu)\varepsilon_{x,y}] \quad ; \quad \tau_{xy} = G\gamma_{xy} \quad (1)$$

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

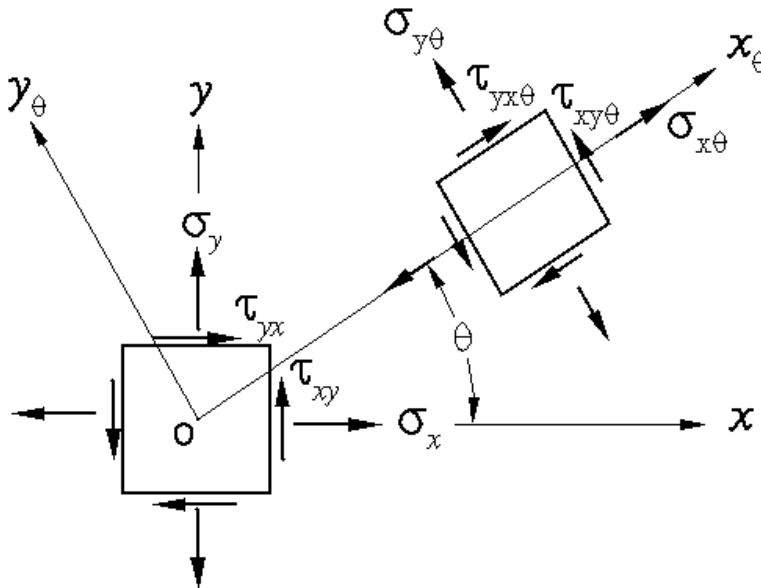


Figura 2 Representação do estado plano de tensões. A variável σ representa as tensões normais com suas componentes cartesianas x e y . A variável τ representa a tensão cisalhante agindo transversalmente às componentes cartesianas x e y .

Logo, estes calculos globais de tensão não são constantes para toda seção transversal do elemento analisado, necessitando de um fator de correção para o campo das tensões.

2.1 - FATOR DE CONCENTRAÇÃO DE TENSÃO

Como visto anteriormente, a distribuição de tensões, ao longo de seções de geometria não uniforme, não é necessariamente constante. Percebeu-se que quando o fluxo dos esforços desvia-se de descontinuidades da geometria, este se concentra na proximidade de tais obstáculos, sendo esta concentração a responsável pelo aumento local das tensões.

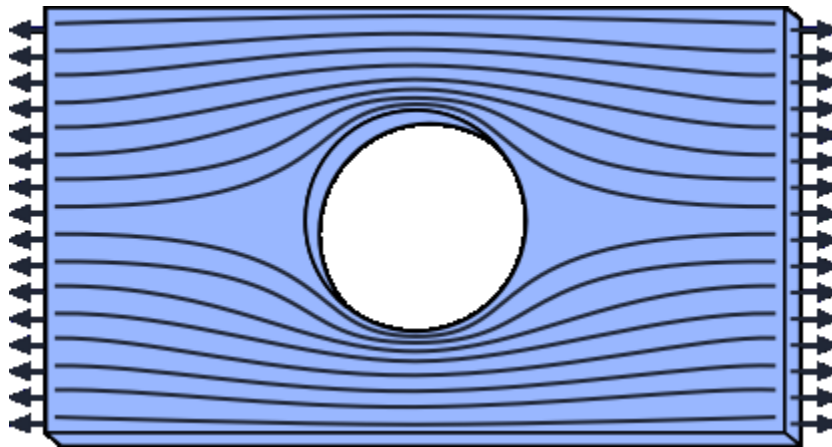


Figura 3 Linhas de força desviando de descontinuidades na geometria.

Tal fato pode ser verificado através da observação das linhas de força, pois estas são os caminhos de força constante que ligam os pontos de aplicação da carga, logo, as tensões são proporcionais as linhas de força que passam por uma unidade de área[5].

Para representar este fato, foi então introduzido o conceito de fator de concentração de tensão, usualmente denominado K_t , ou seja:

$$K_t = \frac{\sigma_{m\acute{a}x}}{\sigma_n} \quad (2)$$

Este fator é definido como a relação entre a máxima tensão $\sigma_{m\acute{a}x}$ que atua em uma dada seção, contendo um concentrador, e a tensão nominal σ_n atuante na mesma área, não fosse o concentrador.

Tecnicamete, sabendo o valor de $\sigma_{m\acute{a}x}$ que atua na estrutura, saberíamos determinar o fator de concentração de tensão presente. Entretanto, estes valores são extremamente difíceis de se obter analiticamente, normalmente recorrendo a soluções numéricas de elementos finitos. A próxima seção é dedicada à introdução das duas soluções analíticas para o campo de tensões com concentração.

2.2 – SOLUÇÃO ANALÍTICA DE KIRSCH

A primeira solução analítica para concentração de tensões foi apresentada por Kirsch, em 1898, para uma placa infinita com furo central [13].

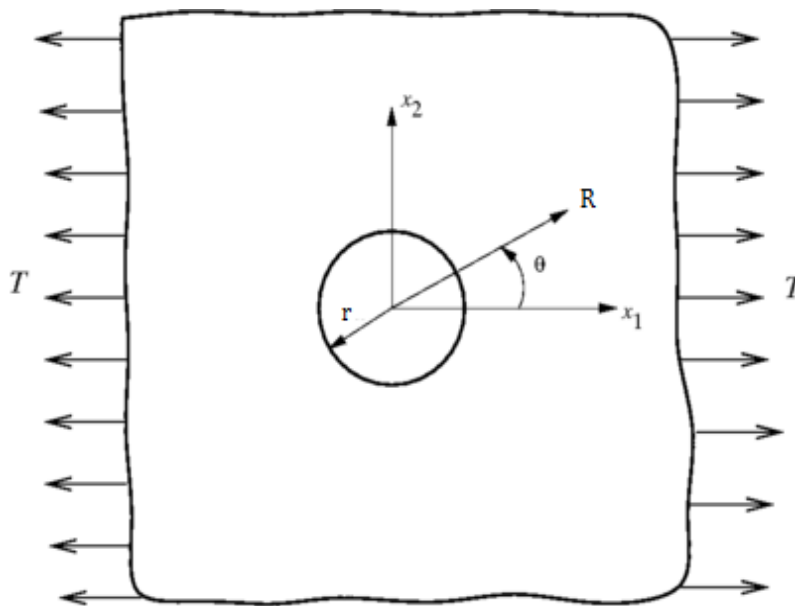


Figura 4 Placa plana com furo central.

Considerando tensão uniaxial distribuída, sendo r o raio do furo central e R uma posição radial genérica, foi obtida a expressão:

$$\begin{aligned}\sigma_{rr} &= \frac{\sigma_n}{2} \left(1 - \left(\frac{r}{R} \right)^2 \right) + \frac{\sigma_n}{2} \left(1 - 4 \left(\frac{r}{R} \right)^2 + 3 \left(\frac{r}{R} \right)^2 \right) \cos 2\theta \\ \sigma_{\theta\theta} &= \frac{\sigma_n}{2} \left(1 - \left(\frac{r}{R} \right)^2 \right) - \frac{\sigma_n}{2} \left(1 + 3 \left(\frac{r}{R} \right)^2 \right) \cos 2\theta \\ \tau_{r\theta} &= -\frac{\sigma_n}{2} \left(1 + 2 \left(\frac{r}{R} \right)^2 - 3 \left(\frac{r}{R} \right)^4 \right) \sin 2\theta\end{aligned}\tag{3}$$

Para o caso onde r e R são iguais (borda do furo), as expressões se reduzem a :

$$\sigma_{rr} = 0 \quad \tau_{r\theta} = 0 \quad \sigma_{\theta\theta} = \sigma_n(1 - 2 \cos 2\theta)\tag{4}$$

Logo, o ponto mais solicitado da estrutura ocorre quando $\theta = \pm 90^\circ$, resultando numa concentração de tensão igual a 3σ . Com esta primeira análise, pode-se então perceber que a mera introdução de um furo pode incrementar consideravelmente a tensão local de uma dada peça.

As equações parecem simples em princípio, porém é necessário salientar que sua resolução envolve a solução de mais de dezoito equações constitutivas compostas por complexas funções bi-harmônicas. Para a dedução completa, reportar às referências [5,13].

2.3 – SOLUÇÃO ANALÍTICA DE INGLIS

Esta solução envolveu a resolução das tensões em um furo elíptico, sendo apresentada por Inglis, em 1913 [9]. A solução encontrada não é nada trivial, valendo-se de coordenadas ortonormais elíptico-hiperbólicas (δ, ξ) mapeando o plano das tensões por meio de elipses δ e hipérboles ξ . Estudando um caso simplificado, temos que, para uma placa sob tensão uniaxial distribuída σ_n :

$$\sigma_{max} = \sigma_n \left(1 + \frac{2a}{b} \right) = \sigma_n \left(1 + 2 \sqrt{\frac{a}{\rho}} \right) \quad (5)$$

Sendo as variáveis a e b as dimensões características da elipse que forma o furo, e ρ o menor raio de curvatura presente na geometria.

Para a dedução completa da solução e casos complementares, reportar-se às referências [19,18].

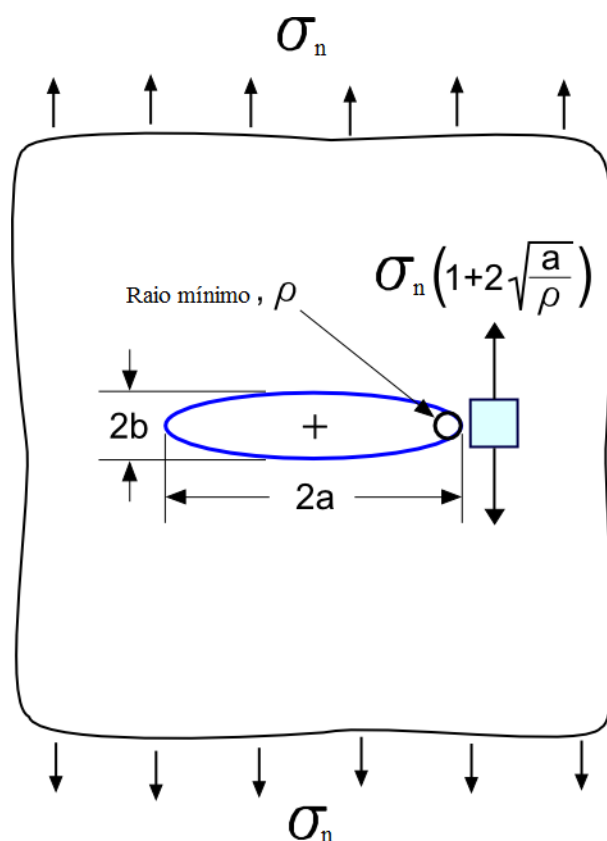


Figura 5 Placa plana com furo central elíptico.

2.4 – ESTIMATIVA DE K_t POR ELEMENTOS FINITOS

Como visto nas seções anteriores, o cálculo analítico dos campos de tensão na presença de descontinuidades geométricas bruscas, não é uma tarefa trivial. Com o advento dos microcomputadores, o cálculo numérico das tensões, por meio da técnica dos elementos finitos,

tornou-se comumente difundido. Esta técnica é um método global de análise, que calcula o campo dos deslocamentos, deformações e tensões, discretizando a geometria em componentes menores (elementos finitos), sendo introduzida no próximo capítulo deste trabalho.

ELEMENTOS FINITOS

O método de elementos finitos é um procedimento numérico que visa solucionar sistemas de equações diferenciais parciais, na qual um meio contínuo é discretizado em pequenos elementos chamados “Elementos finitos”. Resolver este sistema implica na resolução do sistema $\mathbf{KU} = \mathbf{F}$, onde \mathbf{K} representa a matriz de rigidez, \mathbf{U} o vetor dos deslocamentos e \mathbf{F} o vetor que contém as forças atuante no sistema [12,17,18].

3.1 - CÁLCULO DOS DESLOCAMENTOS \mathbf{U}

Considerando $\mathbf{u}^{(\varphi)}$ como uma aproximação do campo de deslocamentos \mathbf{u} de cada elemento φ , tal que:

$$\mathbf{u}^{(\varphi)}(x) = \sum_{i=1}^n u_i N_i(x) = \mathbf{N} \hat{\mathbf{u}}^{(\varphi)} \quad (6)$$

Temos que $\hat{\mathbf{u}}^{(\varphi)}$ representa o conjunto de deslocamentos avaliado nos nós de um dado elemento, n a quantidade de nós e φ o número do elemento. Uma vez conhecidos os deslocamentos, faz-se necessário calcular as deformações, que no \mathbb{R}^2 , são dadas pela relação:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathbf{S}\mathbf{u}^{(\varphi)} \sim \mathbf{S}\mathbf{N}\mathbf{u}^{(\varphi)} = \mathbf{B}\mathbf{u}^{(\varphi)} \quad (7)$$

As variáveis u e v correspondem às componentes x e y do deslocamento dos nós do elemento finito, sendo a matriz $\mathbf{S}\mathbf{N}$, também conhecida como matriz \mathbf{B} , a que representa a relação deformação-deslocamento, sendo expressa como[12]:

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix}^T \quad (8)$$

Definem-se o vetor tensão $\{\boldsymbol{\sigma}\}$ e o vetor deformação $\{\boldsymbol{\varepsilon}\}$ como:

$$\{\boldsymbol{\sigma}\} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \quad \{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (9)$$

Desconsiderando efeitos térmicos, pode-se definir a relação tensão-deformação por:

$$\{\boldsymbol{\sigma}\} = [\mathbf{C}]\{\boldsymbol{\varepsilon}\} \quad (10)$$

Onde a matriz $[\mathbf{C}]$ é simétrica e representa a matriz constitutiva. Sendo assim, temos a relação [11]:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (11)$$

Onde ν é o coeficiente de Poisson e E é o modulo de Young do material.

Após definir os campos de deslocamento $\mathbf{u}^{(\varphi)}$ e as tensoes $\{\boldsymbol{\sigma}\}$, podemos então definir as forças $\mathbf{f}^{(\varphi)}$ agindo localmente sobre o elemento, as quais são devido as forças elementares $\mathbf{q}^{(\varphi)}$ de corpo \mathbf{b} , tração \mathbf{t} , deformações iniciais $\boldsymbol{\varepsilon}_0$ e tensões iniciais $\boldsymbol{\sigma}_0$, sendo representadas por:

$$\mathbf{q}^{(\varphi)} = - \int_V \mathbf{N}^T \mathbf{b} dV + \int_{\Gamma} \mathbf{N}^T \mathbf{t} d\Gamma - \int_V \mathbf{B}^T \mathbf{C} \boldsymbol{\varepsilon}_0 dV + \int_V \mathbf{B}^T \mathbf{C} \boldsymbol{\sigma}_0 dV \quad (12)$$

Compactamente, representam-se as forças agindo nos nós de cada elemento como:

$$\mathbf{f}^{(\varphi)} = \mathbf{K}^{(\varphi)} \mathbf{u}^{(\varphi)} + \mathbf{q}^{(\varphi)} \quad (13)$$

Para um sistema inicialmente em equilíbrio, as forças elementares $\mathbf{q}^{(\varphi)}$ são nulas, com isso, temos:

$$\mathbf{f}^{(\varphi)} = \mathbf{K}^{(\varphi)} \mathbf{u}^{(\varphi)} \quad (14)$$

Podemos, finalmente, calcular a matriz de rigidez elementar $\mathbf{K}^{(\varphi)}$ de cada elemento φ , integrada na área, pelos pontos de Gauss, como:

$$\mathbf{K}^{(\varphi)} = \iint \mathbf{B}^T \mathbf{C} \mathbf{B} dx dy \quad (15)$$

A matriz de rigidez global \mathbf{K} é então contruída a partir das matrizes elementares $\mathbf{K}^{(\varphi)}$. O sistema de valores globais $\mathbf{KU} = \mathbf{F}$, é então resolvido para um deslocamento \mathbf{U} desconhecido, dado por: $\mathbf{U} = \mathbf{FK}^{-1}$. Uma vez conhecidos os deslocamentos \mathbf{U} , podemos então calcular as deformações $\boldsymbol{\varepsilon}$ e tensões $\boldsymbol{\sigma}$ pela equação 11.

3.2– FORMULAÇÃO ISOPARAMÉTRICA

A formulação isoparamétrica é amplamente utilizada na representação de malhas não retangulares, pois transforma superfícies degeneradas (globalmente) em superfícies conhecidas (localmente), com fácil manuseio matemático e baixo custo computacional.

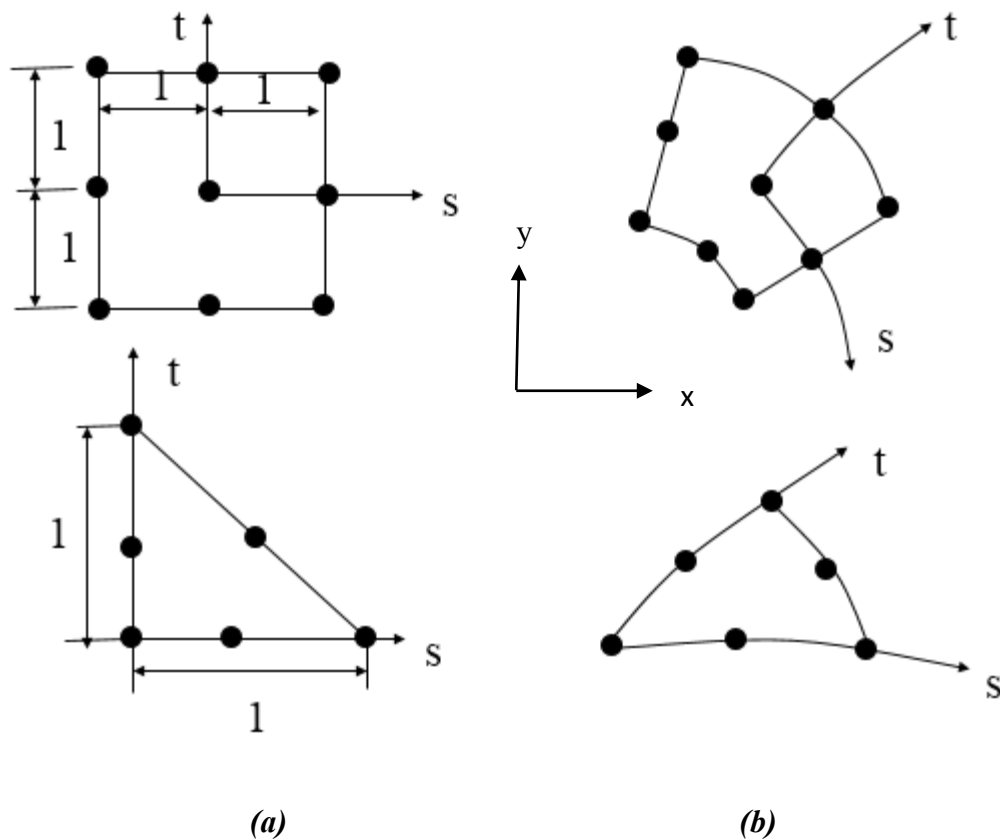


Figura 6 Comparação entre coordenadas. Na figura "a" o elemento e é mostrado em coordenadas locais, sendo na figura "b" em coordenadas globais.

No caso simples de um elemento com quatro nós (conforme mostrado abaixo), as funções de forma a ele associadas seriam:

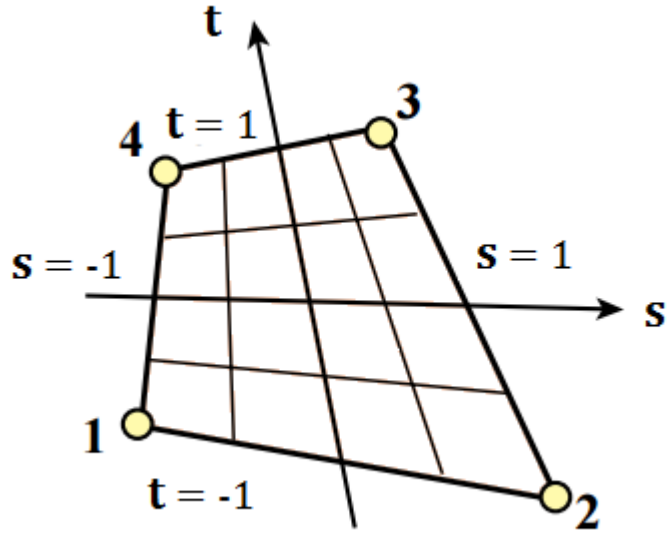


Figura 7 Elemento degenerado com quatro nós.

$$\begin{aligned}
 N_1^\varphi &= \frac{1}{4}(1-s)(1-t) & ; & & N_2^\varphi &= \frac{1}{4}(1+s)(1-t) \\
 N_3^\varphi &= \frac{1}{4}(1+s)(1+t) & ; & & N_4^\varphi &= \frac{1}{4}(1-s)(1+t)
 \end{aligned}
 \tag{16}$$

A transformação de coordenadas (x, y) para (s, t) , realizada pela equação 17 é chamada de mapeamento isoparamétrico. Este processo facilita os cálculos que envolvem um problema de elementos finitos, pois permite avaliar integrais em limites de integração simplificados:

$$I = \int_{x_1}^{x_2} \int_{y_1}^{y_2} f(x, y) dx dy \quad \rightarrow \quad \int_{-1}^1 \int_{-1}^1 \varphi(s, t) ds dt
 \tag{17}$$

Discussões aprofundadas sobre elementos finitos e suas técnicas são apresentadas nas referências [10,12].

3.3 MALHA ESTRUTURADA E NÃO ESTRUTURADA

Pode-se dizer que a representação não estruturada de malhas apresenta vantagens na discretização de geometrias complexas [18]. A diferença entre estes os dois tipos de malha (estruturada e não estruturada) se dá na forma pela qual seus dados/elementos são apresentados. Uma malha estruturada consiste em um conjunto de coordenadas que correspondem diretamente a elementos de uma matriz, sendo assim, pontos vizinhos do objeto discretizado correspondem a elementos vizinhos na matriz que os representa. Por exemplo, em \mathbb{R}^2 , a variável $\alpha(x,y)$ representa as coordenadas cartesianas x e y dos pontos em uma malha.

Malhas não estruturadas, por outro lado, não podem ser representadas como anteriormente explicado. Para cada ponto $\alpha_i(x,y)$, $i = 1, \dots, n \in \mathbb{R}^2$ da malha, informações adicionais relativas a conexão com os pontos que compõem o elemento finito devem ser explicitamente informadas na matriz de conectividade.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24

(a)

1	20	13	2
7	6	5	4
22	17	16	14
8	3	9	10
11	12	19	23
15	18	21	24

(b)

Figura 8: Comparação entre estruturas de malha. Na figura "a" os elementos seguem uma sequência lógica, não precisando de informações adicionais. Na figura "b", a malha apresenta uma estrutura randômica dos elementos, logo necessita da informação relativa aos elementos vizinhos.

Como representado na figura, para cada elemento $\hat{V}_i, i = 1, \dots, k$, é necessário saber as coordenadas de todos os pontos que o compõem, sendo \hat{V}_i definido por exemplo, como $\hat{V}_i(n_1, n_5, n_7)$, pois os nós dos elementos não seguem uma ordem definida [11].

REPRESENTAÇÃO DE DOMÍNIOS

4.1 - FUNÇÃO DISTÂNCIA SINALIZADA (FDS) OU SIGNED DISTANCE FUNCTION (SDF)

Vamos assumir Ω como um subconjunto de \mathbb{R}^2 . A função distância associada a Ω é o mapeamento $d_\Omega: \mathbb{R}^2 \rightarrow \mathbb{R}$ definido como:

$$d_\Omega(x) = s_\Omega(x) \min_{y \in \delta\Omega} \|x - y\| \quad (18)$$

Onde $\delta\Omega$ representa a borda do domínio Ω , $\|\cdot\|$ é a norma Euclidiana padrão em \mathbb{R}^2 e o sinal da função é definido como:

$$s_\Omega(x) = \begin{cases} -1, & x \in \Omega \\ +1, & x \in \mathbb{R}^2 \setminus \Omega \end{cases} \quad (19)$$

Sendo assim, é natural conjecturar que:

$$d(x) \begin{cases} < 0, & x \in \Omega \\ = 0, & x \in \delta\Omega \\ > 0, & x \notin \Omega \end{cases} \quad (20)$$

A SDF nos dá a menor distância entre um ponto $x \in \Omega$ e um ponto $y \in \delta\Omega$, apresentando sinal positivo se x está fora do domínio Ω e negativo no caso oposto [7].

4.2 - REPRESENTAÇÃO IMPLÍCITA E CRIAÇÃO DE DOMÍNIOS

A representação implícita nos permite identificar facilmente se um elemento faz parte do domínio, ou não, ao analisar sua SDF.

$$\Omega = \{\mathbf{x} \in \mathbb{R}^2 \mid d(\mathbf{x}) \leq 0\} \quad (21)$$

$$\delta\Omega = \{\mathbf{x} \in \mathbb{R}^2 \mid d(\mathbf{x}) = 0\}$$

Assim, para cada domínio, é necessário criar a função distância d_Ω que nos permita identificar todos os pontos $d_\Omega(\mathbf{x})$ de Ω . Domínios complexos Ω são criados pela combinação de domínios simples Ω_A e Ω_B (reta, círculo, quadrado), os quais apresentam funções distâncias d_A e d_B de fácil obtenção.

$$\textbf{União} \quad d_{\Omega_A \cup \Omega_B}(\mathbf{x}) = \min(d_{\Omega_A}(\mathbf{x}), d_{\Omega_B}(\mathbf{x}))$$

$$\textbf{Interseção} \quad d_{\Omega_A \cap \Omega_B}(\mathbf{x}) = \max(d_{\Omega_A}(\mathbf{x}), d_{\Omega_B}(\mathbf{x})) \quad (22)$$

$$\textbf{Diferença} \quad d_{\Omega_A \setminus \Omega_B}(\mathbf{x}) = \max(d_{\Omega_A}(\mathbf{x}), -d_{\Omega_B}(\mathbf{x}))$$

$$\textbf{Complementação} \quad d_{\mathbb{R}^2 \setminus \Omega_A}(\mathbf{x}) = -d_{\Omega_A}(\mathbf{x})$$

Com estas funções, podemos então identificar facilmente pontos que pertençam, ou não, ao domínio estudado [7,18].

CRIAÇÃO DE MALHAS NÃO ESTRUTURADAS

A definição das funções, diagramas e triangulações deste capítulo se fazem necessárias para podermos entender as etapas da formação da malha dos elementos finitos não estruturados deste projeto. Estas etapas podem ser enumeradas como:

- Geração de pontos aleatórios no domínio
- Triangulação de Delaunay & Diagrama de Voronoi
- Homogeneização das células pelo Algoritmo de Lloyd
- Diagrama de Voronoi Centroidal

Após estes passos, obtemos uma malha de elementos finitos poliédrica, não estruturada, homogênea, o que é fundamental para resolução de problemas de elementos finitos.

5.1- DIAGRAMA DE VORONOI OU VORONOI TESSELATION

Dado um conjunto de pontos $\{z_i\}_{i=1}^k$ que pertencem ao domínio Ω contido em \mathbb{R}^N , a região de Voronoi \hat{V}_i consiste em todos os pontos em Ω que são mais próximos de Z_i que qualquer outro ponto do conjunto. O conjunto $\{\hat{V}_i\}_{i=1}^k$ forma uma partição de Ω , sendo conhecido como Diagrama de Voronoi ou Voronoi Tessellation.

$$\widehat{V}_i = \{x \in \Omega \mid |x - z_i|_2 < |x - z_j|_2, \quad j = 1, \dots, k, \quad j \neq i\} \quad (23)$$

Os pontos $x = (x, y)$ são quaisquer pontos do domínio, onde os pontos $z_{i=1}^k$ chamados de pontos geradores, cada \widehat{V}_i é uma célula de Voronoi, sendo estes os elementos finitos poliedrais [7,20].

5.2- TRIANGULAÇÃO DE DELAUNAY

A triangulação de Delaunay é complementar ao diagrama de Voronoi e pode ser determinada como aquela onde, dado um conjunto de pontos $\{z_i\}_{i=1}^k$ do domínio Ω , com restrição P , apresenta $\{\{\widehat{V}_i\}_{i=1}^k, \{z_i\}_{i=1}^k\}$ como solução do problema [14]:

$$\min_{\{z_i\}_{i=1}^k \in P, \{\widehat{V}_i\}_{i=1}^k} \mathcal{F}(\{\widehat{V}_i\}_{i=1}^k, \{z_i\}_{i=1}^k) \quad (24)$$

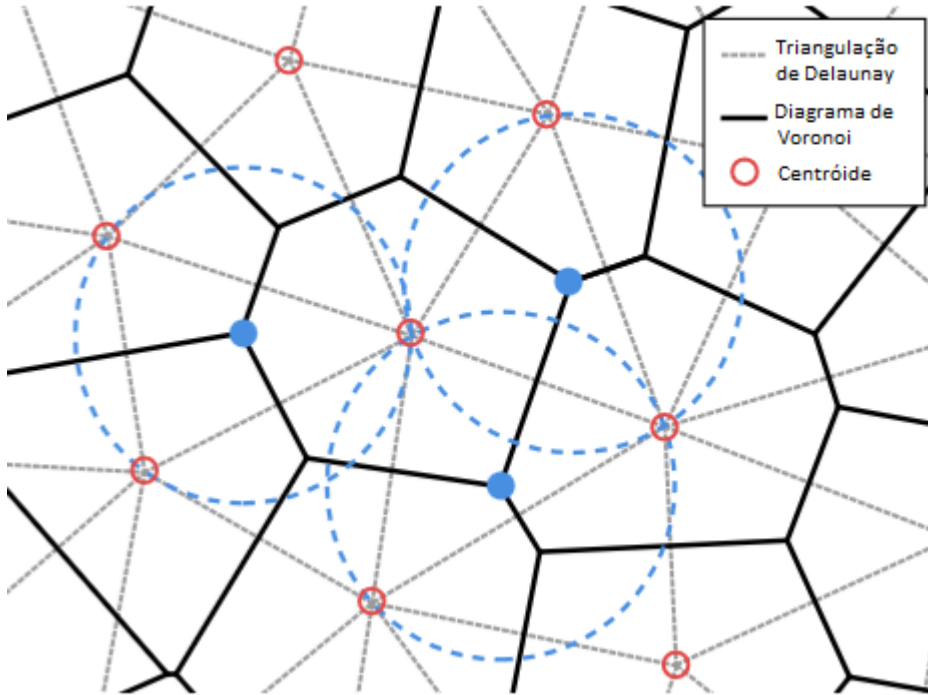


Figura 9: Triangulação de Delaunay e Diagrama de Voronoi [18].

Definição: A triangulação de Delaunay $TD(S)$ é obtida ao conectar, por um segmento de reta, quaisquer dois pontos p, q de S , gerando um círculo C que passe por p e q e não contenha nenhum outro seed de S no interior das suas fronteiras. As bordas de $TD(S)$ são chamadas bordas de Delaunay [8].

5.3- ALGORITMO DE LLOYD

Algoritmo de Lloyd ou iteração de Voronoi é normalmente usado neste contexto como forma de homogeneizar elementos finitos não estruturados, de um conjunto de pontos em espaços Euclidianos, garantindo formas convexas regulares. Para cada iteração, o algoritmo encontra os pontos correspondentes ao centróide das células de Voronoi $\{\hat{V}_i\}_{i=1}^k$, assim como seus pontos geradores \mathbf{z}_i , fazendo com que na iteração subsequente, o ponto gerador (seed) \mathbf{z}_i corresponda ao centróide da célula de Voronoi da iteração anterior. Logo, para cada iteração, o domínio é novamente particionado e rearranjado numa disposição mais homogênea [2].

Como pode ser visto abaixo, os pontos na cor vermelha representam os geradores \mathbf{z}_i das células de Voronoi $\{\hat{V}_i\}_{i=1}^k$, sendo os sinais “+” a posição do centróide da célula. Em (a) temos a distribuição inicial aleatória dos pontos geradores, sendo ao longo de (b), (c) e (d) percebida a gradativa homogeneização das células ao deslocar os pontos geradores em direção ao respectivo centróide do elemento.

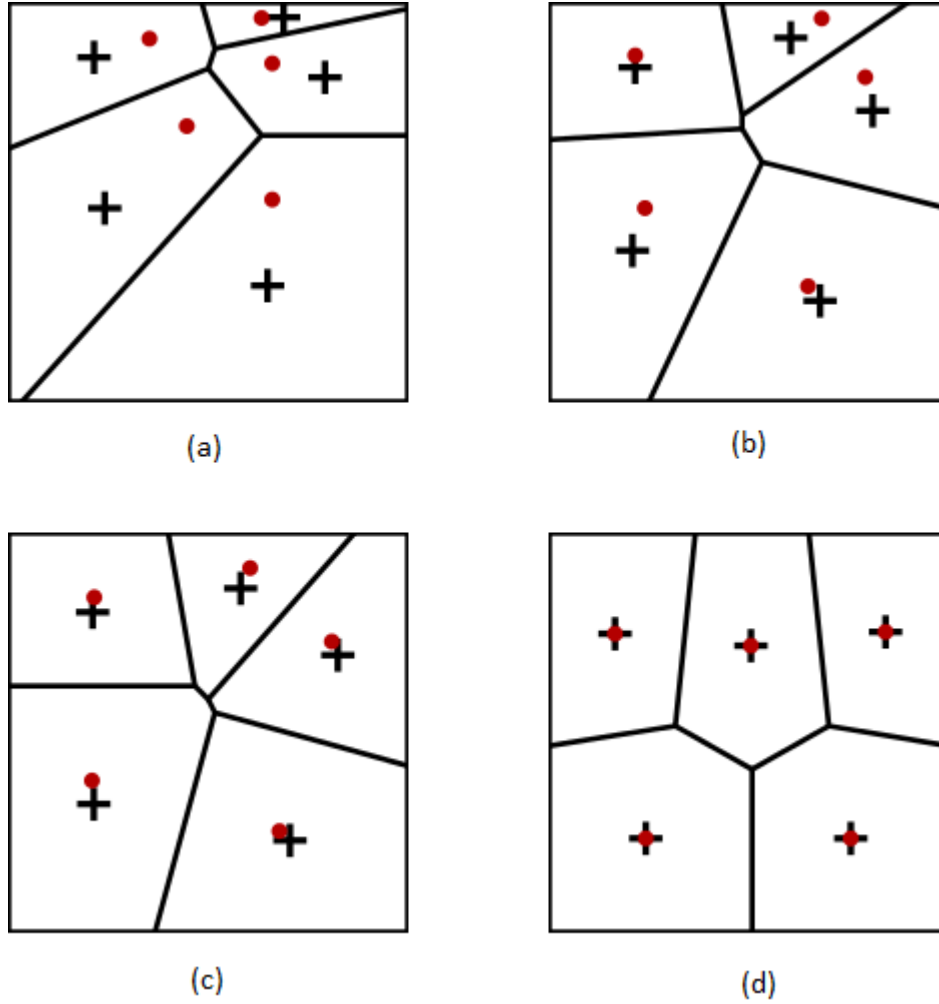


Figura 10: Evolução das células de Voronoi dadas três iterações do algoritmo de Lloyd.

5.4 - DIAGRAMA DE VORONOI CENTROIDAL OU CENTROIDAL VORONOI TESSELLATION (CVT)

Dada uma região β , com densidade ω , função dos pontos \mathbf{y} , definidos em β , o centróide \mathbf{z}^* de β é então definido como:

$$\mathbf{z}^* = \frac{\int_{\beta} \mathbf{y} \, \omega(\mathbf{y}) \, d\mathbf{y}}{\int_{\beta} \omega(\mathbf{y}) \, d\mathbf{y}} \quad (25)$$

Logo, os k pontos $\mathbf{z}_i, i = 1, \dots, k$, no domínio Ω , é possível definir sua região de Voronoi associada (célula de Voronoi) $\widehat{V}_i, i = 1, \dots, k$, que forma a discretização do domínio Ω .

Definição: Dado um conjunto de pontos $\mathbf{z}_{i=1}^k$ do domínio Ω e uma função de densidade ρ positiva, definida em Ω , um diagrama de Voronoi é dito CVT se:

$$\mathbf{z}_i = \mathbf{z}_i^*, \quad i = 1, \dots, k \quad (26)$$

Ou seja, os pontos \mathbf{z}_i são os pontos geradores da região de Voronoi \widehat{V}_i , são os próprios centroides de \widehat{V}_i [14].

REFINAMENTO AUTOMATICO DE MALHA

As técnicas de refino adaptativo de malhas de elementos finitos visa melhorar os resultados numéricos obtidos pelo mesmo, que por muitas vezes pode apresentar valores grosseiros. Tradicionalmente o refino é obtido por meio da mudança da localização dos nós, sem alterar a topologia da malha (refino r), aumento do número de elementos (refino h), aumento da ordem polinomial dos elementos (refino p) ou combinações dos diferentes métodos (refino hp) [1].

Devido às malhas poliedrais, com numero inconstantes de nós por elementos, o processo de refino neste projeto se dá por meio da técnica hp, pois a medida que novos elementos surgem, estes preenchem o domínio, adequando a quantidade de nós para que as premissas do diagrama de Voronoi sejam respeitadas.

As técnicas de refino foram aplicadas a elementos que respektassem dois critérios:

1. Elementos dentro de uma zona de interesse pré definida
2. Elementos de maiores tensões

Inicialmente, identifica-se uma região de interesse λ , onde à priori, exista concentração de tensão, e seleccionam-se todos os Z_i geradores nela contidos.

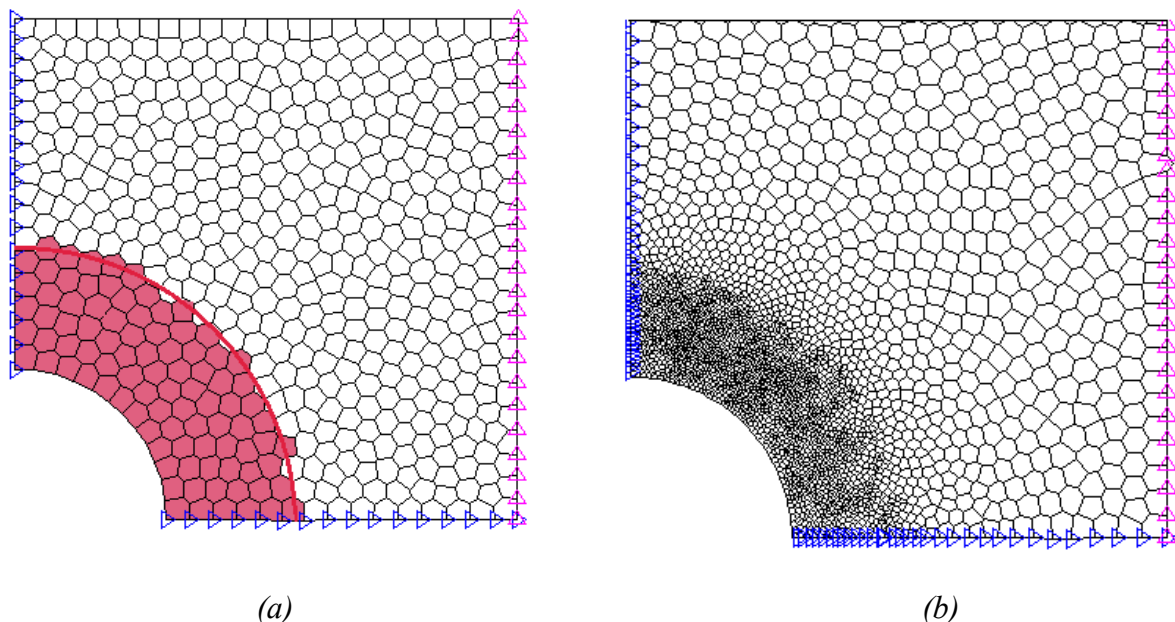


Figura 11 Evolução da seleção de elementos de Interesse. Em rosa, a região pré definida λ .

Este requisito restringe o código, pois, para cada geometria, deve-se explicitamente definir uma região de interesse. Tal fato, porém, maximiza o refino dos elementos de interesse, evitando que pontos de aplicação de carga ou pontos provenientes de erros numéricos inerentes às aproximações do sistema, sejam selecionados.

Após a seleção dos Z_i geradores da região, o código define todos nós que compõem a célula de voronoi \widehat{V}_i , gerada por Z_i , como novos geradores. Com isto, ocorre um incremento no número de elementos na malha.

Uma vez refinada a seção, são analisadas as tensões referentes a cada elemento, sendo selecionados àqueles 30% com maior tensão contidos em λ . Este processo se repete até que a tensão do elemento mais solicitado na iteração $i - 1$ fique dentro de uma faixa de erro, em

relação a tensão do mesmo elemento na iteração i . O algoritmo descrito é apresentado abaixo, na forma de pseudocódigo:

Algoritmo Refinamento de Malha

```
1: Procedimento MeshRefinement (Pontos geradores)
2: PolyMesher (Domínio, N# de Elementos, N# de Iterações )
3: Análise MEF
4: Definição, à priori, da região de interesse contendo centroides (P)
5:   For  $i = 1$  to MaxIter do
6:     Cada nó de P é um novo ponto gerador
10:    PolyMesher (Domínio, N# de Elementos, N# de Iterações, Nós de P )
11:    Análise MEF
12:    If  $\text{abs}(\text{MaxTension}_i - \text{MaxTension}_{i-1}) \leq \text{erro}$  then
13:      Break
14:    End if
15:    Sort (Tensões)
16:     $P = \text{Tensões} (0.7 * \text{ceil}(\text{length}(\text{Tensões})): \text{length}(\text{Tensões}))$ ;
17:  End for
18: End Procedimento
```

RESULTADOS

Buscando validar o método aplicado, inicialmente, foi realizada a simulação do caso analisado por Kirsch [13], confrontando e investigando os resultados. Este modelo consiste de uma placa com furo circular central, que devido à simetria, pode ser representado como exemplificado pela figura abaixo:

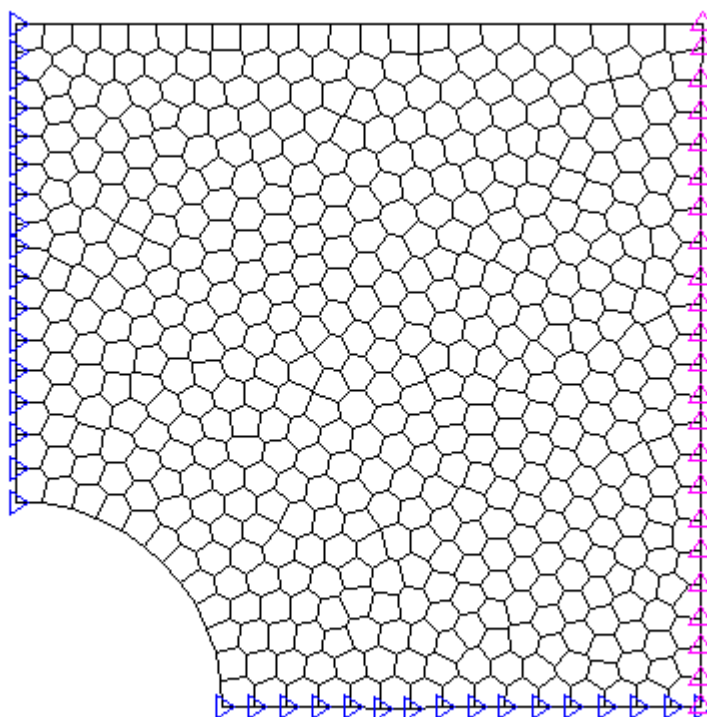


Figura 12 Modelo computacional para placa com furo central.

O modelo foi dimensionado conforme proposto por [6], onde os triângulos azuis representam as condições de contorno do problema, e os triângulos rosas os pontos de aplicação da carga.

As condições de contorno das duas faces são distintas, pois seus respectivos pontos são fixos relativos apenas a uma coordenada. Os pontos da face esquerda apresentam seu deslocamento em x restrito, estando este livre em y ; já os pontos da face inferior, apresentam, de forma oposta, deslocamento em y restrito e livre em x .

É notória a melhora na visualização do campo de tensão com o refino da malha. A figura 13 apresenta a condição inicial de tensões para o problema de Kirsch.

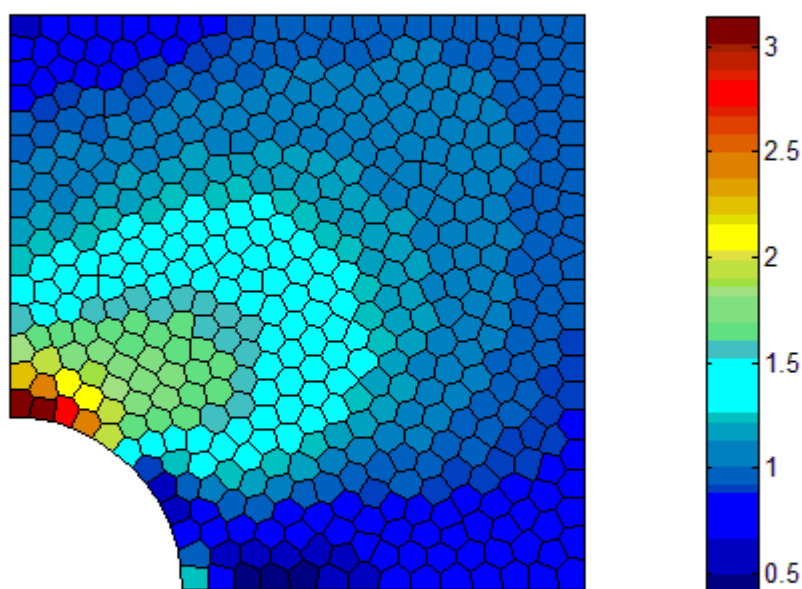


Figura 13 Campo de tensões iniciais para placa com furo central

A figura 14, apresenta o campo final de tensões obtidos após 4 etapas de refino da malha, e posterior homogeneização dos elementos pelo método de Lloyd.

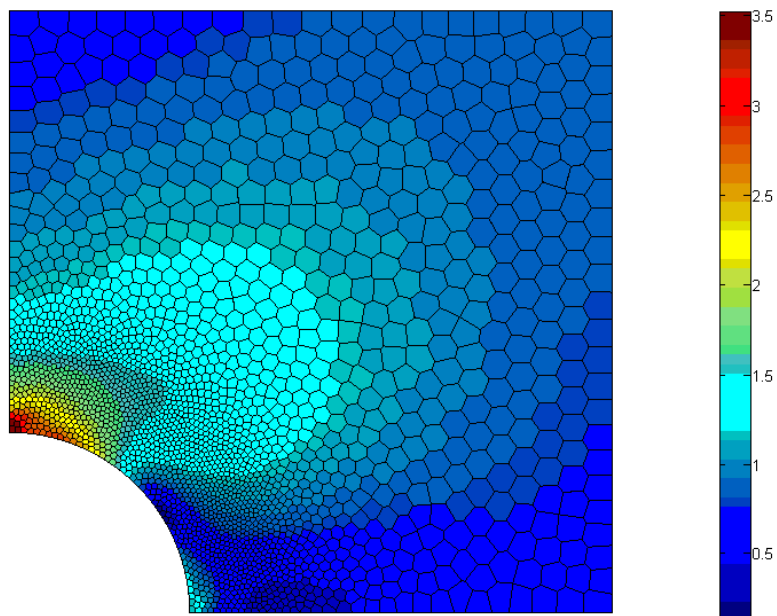


Figura 14 Campo de tensões iniciais para placa com furo central

A figura 17 apresenta a evolução da malha com o refino local dos elementos de maior tensão. A medida que os campos de tensão altos são refinados, incrementa-se o número de elementos discretizando a área, o que gera uma melhor precisão no cálculo. A tabela a seguir, mostra a evolução dos parâmetros da malha com as iterações.

Iterações	Número de Elementos	K_t
0	500	3,226
1	626	3,423
2	814	3,481
3	1232	3,517
4	2124	3,519

Figura 15 Evolução de número de elementos e da concentração de tensão

Percebe-se um comportamento assintótico da tensão do elemento mais solicitado, ao incrementar o numero de elementos discretizando a malha. Tal comportamento pode ser visto no gráfico abaixo:

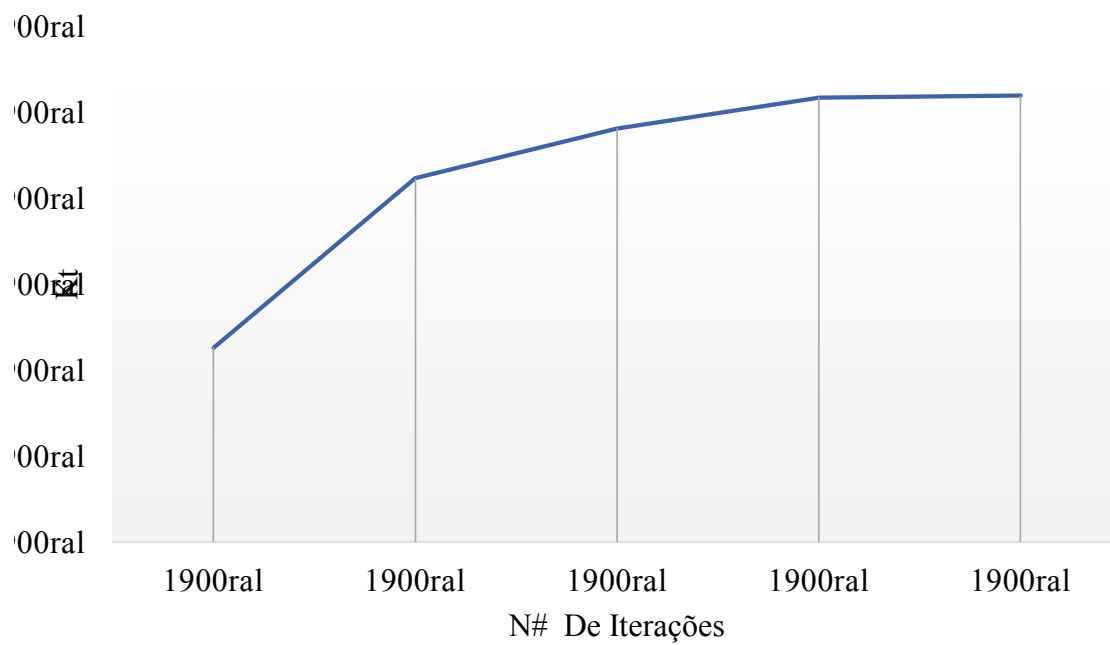


Figura 16 Evolução de Kt com 4 iterações

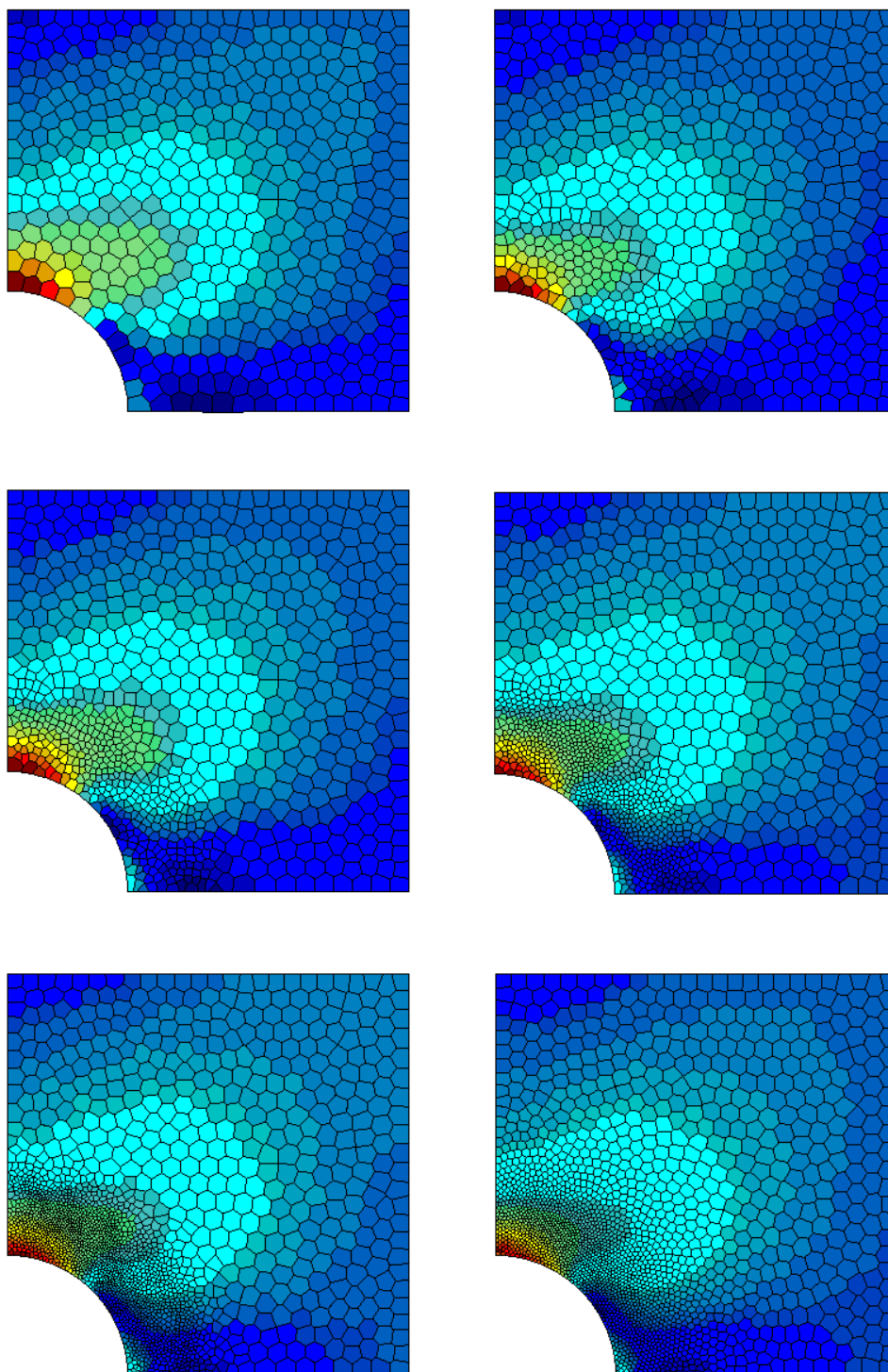


Figura 17 Evolução da malha, para placa com furo central, por meio do refinamento automático baseado no valor das tensões.

Nas próximas seções, serão apresentados os resultados da simulação dos demais domínios construídos

7.1 - MICHELL DOMAIN

Este domínio é descrito como segue na figura abaixo:

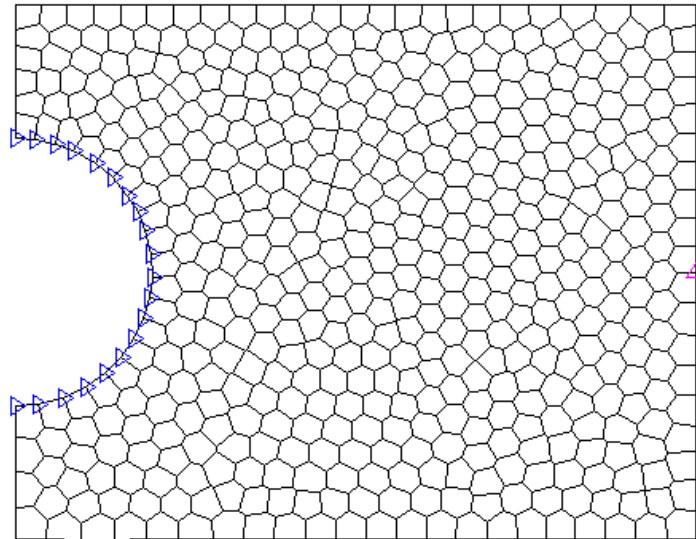


Figura 18 Michell Domain. Condições de contorno.

É interessante notar o refinamento dos elementos na vizinhança das variações bruscas de geometria, e consequente concentração de tensão. Na imagem abaixo, nota-se uma real melhora na percepção do campo das tensões a medida que refina-se a malha:

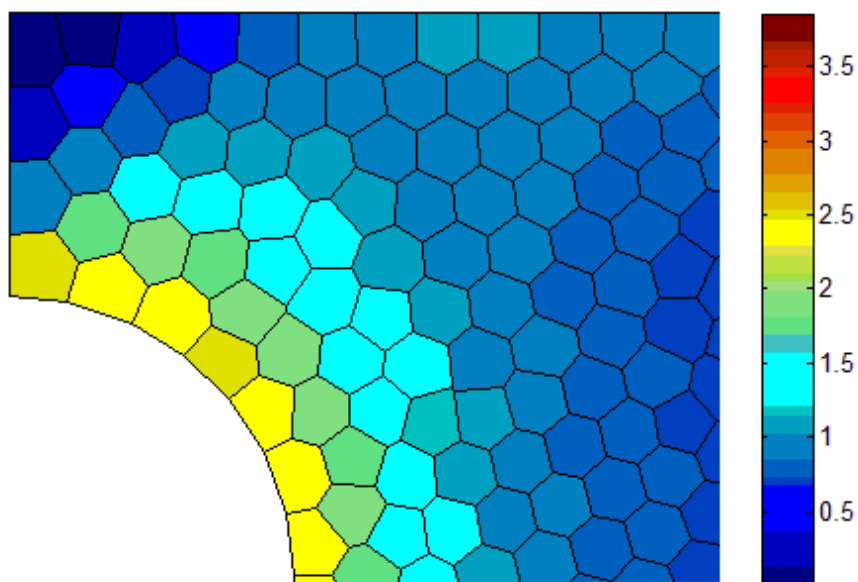


Figura 19 Zoom do campo de tensões inicial, para Michell Domain.

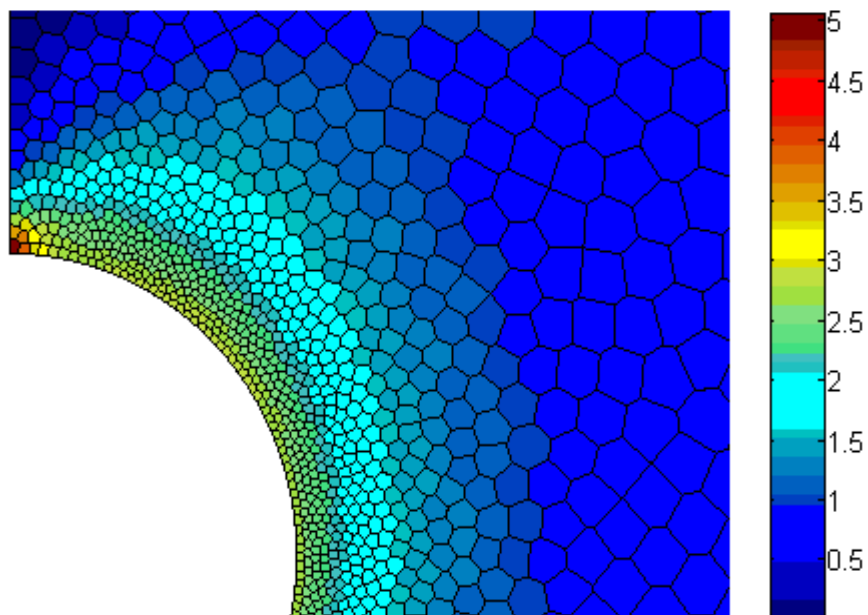


Figura 20 Zoom do campo de tensões final, para Michell Domain, após 4 etapas de refinamento de malha.

Nota-se um aumento na precisão do campo de tensões, o que pode ser visto pelo elemento com maior valor de tensão, no entalhe da geometria. Inicialmente, este apresentava K_t de 2,2 e ao refinarmos a malha, vemos que este apresenta valor de 4,9.

Para melhor perceber a melhora com o refinamento automático, a figura a seguir mostra a evolução da malha com 4 iterações e uma homogeinização final, pelo algoritmo de Lloyd.

A figura 21 apresenta a evolução da malha com o refino local dos elementos de maior tensão. A medida que os campos de tensão altos são refinados, incrementa-se o número de elementos discretizando a área, o que gera uma melhor precisão no cálculo.

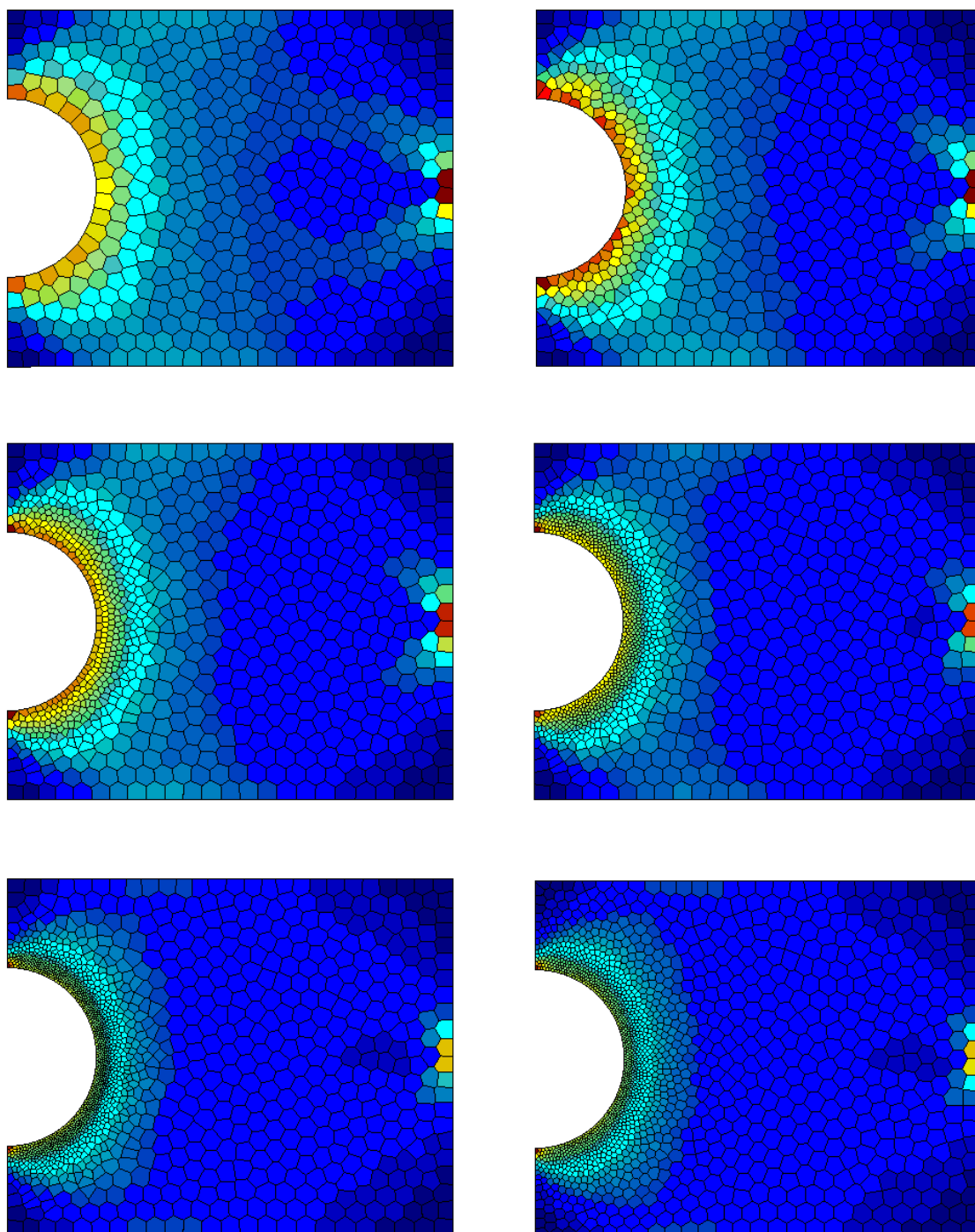


Figura 21 Evolução da malha, para Michell Domain, por meio do refinamento automático baseado no valor das tensões.

7.2 - ELE DOMAIN

Este domínio foi desenvolvido para melhor entender a concentração de tensões em seções com perfil “L”, onde há variação brusca na geometria. Estas seções são comumente usadas na indústria, sendo seu entendimento de fundamental importância.

A imagem abaixo apresenta a geometria do domínio, assim como seus pontos fixos (triângulos azuis) e pontos de aplicação da força distribuída (triângulos rosas).

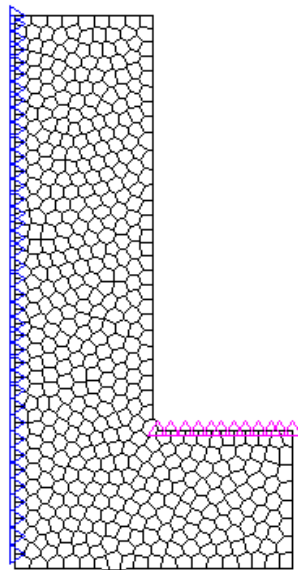


Figura 22 *EleDomain. Condições de contorno.*

A seguir seguem os resultados da simulação para 3 iterações e uma homogeneização.

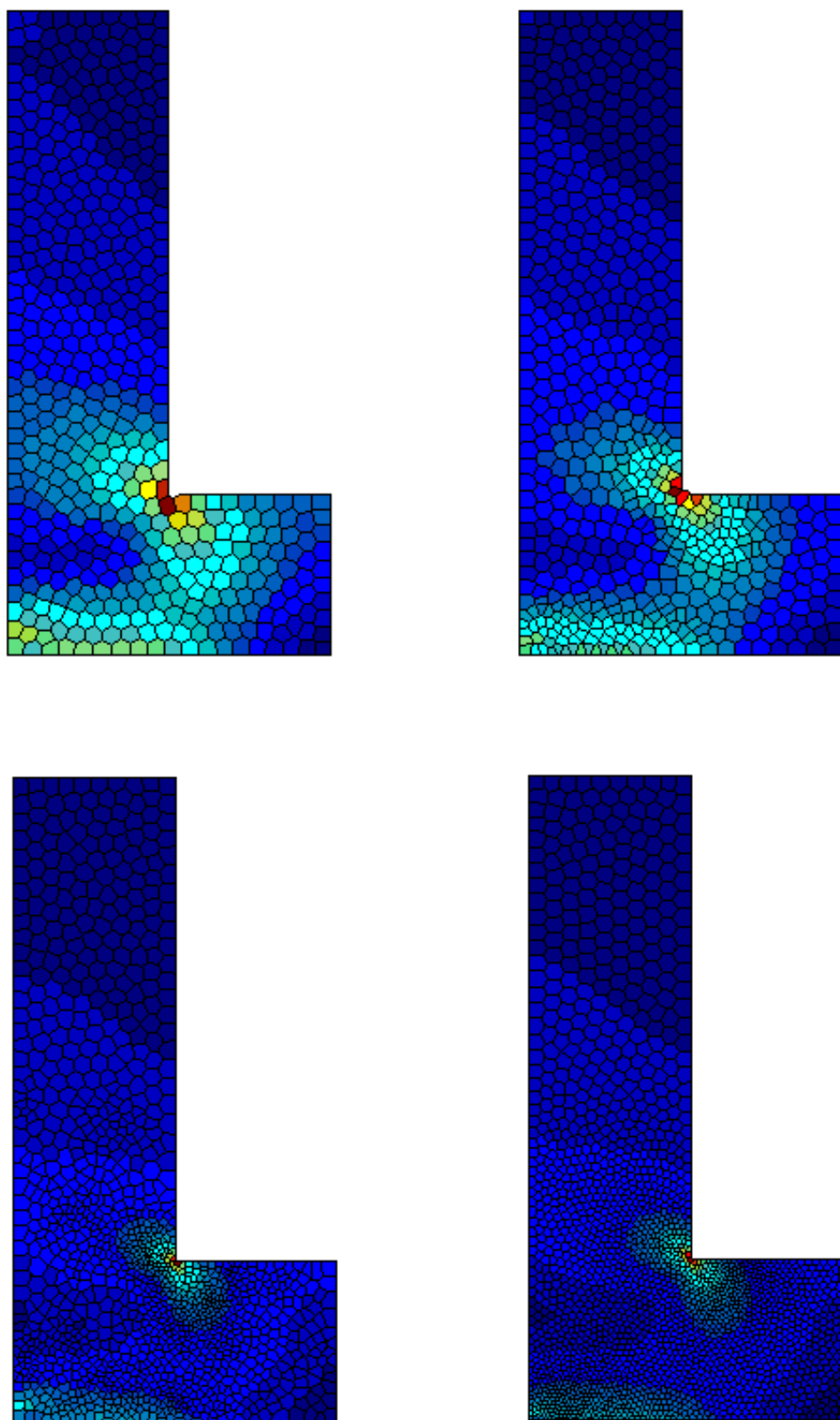


Figura 23 Evolução da malha por meio do refinamento automático baseado no valor das tensões.

CONCLUSÃO

Neste projeto, o Método dos Elementos Finitos, aplicado a malhas poliédricas, é utilizado para analisar os campos de tensão de estruturas mecânicas. A análise foi feita em domínios definidos implicitamente, o que simplifica a identificação dos elementos pertencentes do mesmo.

O método de discretização dos domínios utilizado, apresenta diversas vantagens em relação aos tradicionais. Além de gerar elementos que se assemelham ao formato de estruturas naturais, estes permitem a representação de domínios complexos assim como possibilita captura, com melhor precisão, das nuances da geometria.

Os resultados numéricos e padrões gráficos obtidos, se comportam como previsto em literatura, apresentando altos valores de K_t nas mudanças bruscas de geometria [6,12,17]. Ressalta-se porém, que o valor numérico na simulação do problema de Kirsch [13] apresentou diferença de 17,3% em relação a solução analítica, obtendo K_t de 3,519. O campo de tensões final, entretanto, se comportou como esperado, o que nos leva a acreditar em instabilidades numéricas nas etapas de cálculo (como por exemplo singularidade na matriz de rigidez \mathbf{K}).

Existe muito espaço para futuras melhorias, seja no código ou na abordagem do problema. Cita-se em específico a forma de cálculo das tensões nos elementos, a qual foi calculada nos pontos nodais, o que consequentemente introduz erros na simulação

REFERÊNCIAS

1. A.A. Novotny, E.A. Fancello. Um refinamento h , p e hp adaptativo na análise de flexão de placas semi-espessas. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. Vol. 14,1, 25-48, 1998.
2. A.E.H. Love, "A treatise on the mathematical theory of elasticity" Cambridge University Press, 1927.
3. C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes. PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. Struct Multidisc Optim (2012) 45:309–328. Published online: 8 January 2012 © Springer-Verlag 2011
4. Daniel Sieger, Pierre Alliez and Mario Botsch. Optimizing Voronoi Diagrams for Polygonal Finite Element Computations. In Proceeding of the 19th international meshing roundtable, pp. 335-350. Springer, 2010.
5. de Castro, J.T.P, Meggiolaro, M.A; Topper, T.H. Fadiga - Técnicas e Práticas de Dimensionamento Estrutural sob Cargas Reais de Serviço: Volume I - Iniciação de Trincas, 2009.
6. Fish J, Belytschko T, A First Course in Finite Elements, 2007.
7. G. H. Paulino, I. F. M. Menezes, J. B. Cavalcante Neto, L. F. Martha. A methodology for adaptive finite element analysis: Towards an integrated computational environment. Computational Mechanics 23 (1999) 361-388 © Springer-Verlag 1999.

8. Guibas, L. J.; Knuth, D. E.; Sharir, M. Randomized incremental construction of delaunay and voronoi diagrams. *Algorithmica*, v.7, n.1-6, p. 381–413, 1992.
9. Inglis, C.E., "Stresses in Plates Due to the Presence of Cracks and Sharp Corners," *Transactions of the Institute of Naval Architects*, Vol. 55, pp. 219-241, 1913.
10. J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74, 2002.
11. J. Thompson, B. Soni, N. Weatherill (Eds.), *Handbook for Grid Generation*, CRC Press, Boca Raton, 1999.
12. K. J. Bathe. *Finite Element Procedures*. Prentice Hall, second edition, 1995
13. Kirsch, E.G., "Die Theorie der Elastizität und die Bedürfnisse der Festigkeitslehre," *Zeitschrift des Vereines deutscher Ingenieure*, Vol. 42, pp. 797-807, 1898.
14. Qiang Du, Max Gunzburger. Grid generation and optimization based on centroidal Voronoi tessellations. *Applied Mathematics and Computation* 133 (2002) 591–607.
15. S. Lloyd, Least square quantization in PCM, *IEEE Trans. Inform. Theory* 28 (1982) 129–137.
16. S. Fortune, Voronoi diagrams and Delaunay triangulations, in: D. Du, F. Hwang (Eds.), *Computing in Euclidean Geometry*, World Scientific, River Edge, NJ, 1992, pp. 193–233.
17. T. J. R. Hughes. *The Finite Element Method*. Dover Publications, 2000
18. Thedin, Régis S. *Topology Optimization using Polyhedral Meshes*. 2014. Master thesis, Pontifical Catholic University of Rio de Janeiro.

19. Wai-Fah Chen and Da-Jian Han (2007), "Plasticity for Structural Engineers". J. Ross Publishing ISBN 1-932159-75-4
20. Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang. On centroidal voronoi tessellation—energy smoothness and fast computation. ACM Trans. on Graphics, 28(4):1–17, 2009.

A

Algoritmo

Nesta seção são apresentados os códigos referentes aos domínios criados, e ao algoritmo de refino automático.

A.1 - ELEDOMAIN

```
%----- PolyMesher -----%
% Ref: C Talischi, GH Paulino, A Pereira, IFM Menezes, "PolyMesher: A      %
%       general-purpose mesh generator for polygonal elements written in  %
%       Matlab," Struct Multidisc Optim, DOI 10.1007/s00158-011-0706-z    %
%-----%
function [x] = EleDomain(Demand,Arg)
    BdBx = [0 0.5 0 1];
    switch(Demand)
        case('Dist'); x = DistFnc(Arg,BdBx);
        case('BC'); x = BndryCnds(Arg{:},BdBx);
        case('BdBx'); x = BdBx;
        case('PFix'); x = FixedPoints(BdBx);
    end
%----- COMPUTE DISTANCE FUNCTIONS -----%
function Dist = DistFnc(P,BdBx)
    d1 = dRectangle(P,BdBx(1),BdBx(2),BdBx(3),BdBx(4));
    d2 = dRectangle(P,BdBx(2)/2,BdBx(2),BdBx(2)/2,BdBx(4));
    Dist = dDiff(d1,d2);
%----- SPECIFY BOUNDARY CONDITIONS -----%
function [x] = BndryCnds(Node,Element,BdBx)
    Supp = []; Load = []; temp = [];
    p = 0.25; %distributed Load
%-----Definição dos pontos fixos-----%
    TotalNodes = find((Node(:,2))>0.0001 & Node(:,1)<0.0001);
    Supp = ones(size(TotalNodes,1),3);
    Supp(:,1) = TotalNodes;
%-----Definição dos pontos de aplicação da carga distribuida-----%
-
    for i=1:size(Node,1)
        if ((Node(i,1)>=0.25)&&(abs(Node(i,2)-0.25)<1e-5))
            temp = [temp; [i Node(i,1)]];
        end
    end
    [~,iy] = sort(temp(:,1)); %we need to sort the node according to x-
coordinate
    temp = temp(iy,[1,2]);
    Load = zeros(size(temp,1),3);
```

```

Load(:,1) = temp(:,1);
for e = 1:size(temp,1)-1 %loop through segments
    d = abs(temp(e,2)-temp(e+1,2));
    Load(e,3) = Load(e,3) + 0.5*d*p;
    Load(e+1,3) = Load(e+1,3) + 0.5*d*p;
end
x = {Supp,Load};
%----- SPECIFY FIXED POINTS
function [PFix] = FixedPoints(BdBox)
    PFix = [];
%-----%

```

A.2 - UNITSQUAREDOMAIN

```
%----- PolyMesher -----%
% Ref: C Talischi, GH Paulino, A Pereira, IFM Menezes, "PolyMesher: A
%       general-purpose mesh generator for polygonal elements written in
%       Matlab," Struct Multidisc Optim, DOI 10.1007/s00158-011-0706-z
%-----%
function [x] = UnitSquareDomain(Demand,Arg)
    BdBx = [0 1 0 1];
    switch(Demand)
        case('Dist'); x = DistFnc(Arg,BdBx);
        case('BC'); x = BndryCnds(Arg{:},BdBx);
        case('BdBx'); x = BdBx;
        case('PFix'); x = [];
    end
%----- COMPUTE DISTANCE FUNCTIONS -----%
function Dist = DistFnc(P,BdBx)
    d1 = dRectangle(P,BdBx(1),BdBx(2),BdBx(3),BdBx(4));
    d2 = dCircle(P,0,0,0.3);
    Dist = dDiff(d1,d2);
%----- SPECIFY BOUNDARY CONDITIONS -----%
function [x] = BndryCnds(Node,Element,BdBx)
    Supp = []; Load = []; temp = [];
    p = 1; %distributed Load
%-----Definição dos pontos fixos-----%
    BaseNodes = find((Node(:,1)-0.3)>0.0001 & Node(:,2)<0.0001);
    LateralNodes = find((Node(:,2)-0.3)>0.0001 & Node(:,1)<0.0001);
    TotalNodes = vertcat(BaseNodes,LateralNodes);
    Supp = ones(size(TotalNodes,1),3); Supp(1:length(BaseNodes),2)=0;
    Supp(length(BaseNodes)+1:end,3)=0;
    Supp(:,1) = TotalNodes;
%-----Definição dos pontos de aplicação da carga distribuida-----%
    for i=1:size(Node,1)
        if (abs(Node(i,1)-1)<1e-5)
            temp = [temp; [i Node(i,2)]];
        end
    end
    [~,ix] = sort(temp(:,2)); %we need to sort the node according to y-
coordinate
    temp = temp(ix,[1,2]);
    Load = zeros(size(temp,1),3);
    Load(:,1) = temp(:,1);
    for e = 1:size(temp,1)-1 %loop through segments
        d = abs(temp(e,2)-temp(e+1,2));
        Load(e,2) = Load(e,2) + 0.5*d*p;
        Load(e+1,2) = Load(e+1,2) + 0.5*d*p;
    end
    x = {Supp,Load};
%-----%
```

A.3 - MICHELLDOMAIN

```
%----- PolyMesher -----%
% Ref: C Talischi, GH Paulino, A Pereira, IFM Menezes, "PolyMesher: A
%       general-purpose mesh generator for polygonal elements written in
%       Matlab," Struct Multidisc Optim, DOI 10.1007/s00158-011-0706-z
%-----%
function [x] = MichellDomain(Demand,Arg)
    BdBx = [0 5 -2 2];
    switch(Demand)
        case('Dist'); x = DistFnc(Arg,BdBx);
        case('BC'); x = BndryCnds(Arg{:},BdBx);
        case('BdBx'); x = BdBx;
        case('PFix'); x = FixedPoints(BdBx);
    end
%----- COMPUTE DISTANCE FUNCTIONS
function Dist = DistFnc(P,BdBx)
    d1 = dRectangle(P,BdBx(1),BdBx(2),BdBx(3),BdBx(4));
    d2 = dCircle(P,0,0,BdBx(4)/2);
    Dist = dDiff(d1,d2);
%----- SPECIFY BOUNDARY CONDITIONS
function [x] = BndryCnds(Node,Element,BdBx)
    eps = 0.1*sqrt((BdBx(2)-BdBx(1))*(BdBx(4)-BdBx(3))/size(Node,1));
    CircleNodes = find(abs(sqrt(Node(:,1).^2+Node(:,2).^2)-1.0)<eps);
    Supp = ones(size(CircleNodes,1),3);
    Supp(:,1) = CircleNodes;
    MidRightFace = sqrt((Node(:,1)-BdBx(2)).^2+...
        (Node(:,2)-(BdBx(3)+BdBx(4))/2).^2);
    [foo,MidRightFace] = sort(MidRightFace);
    Load = [MidRightFace(1),0,1];
    x = {Supp,Load};
%----- SPECIFY FIXED POINTS
function [PFix] = FixedPoints(BdBx)
    PFix = [5 0];
%-----%
```


A.4 - REFINAMENTO AUTOMÁTICO DE MALHA

```
% -----
% -----REFINAMENTO AUTOMÁTICO DE MALHA-----
% -----
function P = MeshRefinement(p)
R = 0.5; %para unitssquare
%R = 1.5; %para michell
if ~exist('p','var'), p = [0,0]; end;
Domain = @UnitSquareDomain;
%-----Crio a malha no domínio escolhido
[Node,Element,Supp,Load,P] = PolyMesher(Domain,500,100);
%-----Acho os pontos dentro da região de refio inicial
Pc = PolyMshr_CntrdPly(Element,Node,size(Element,1));
D = find(sqrt((Pc(:,1)-p(1)).^2+(Pc(:,2)-p(2)).^2)-R<0);
%-----Ploto a malha inicial relativo as tensões
[fem] = CreateFEMStruct(Node,Element,Supp,Load);
[U,S,fem] = FEAnalysis(fem);
clf; axis equal; axis off; hold on;
MaxNVer = max(cellfun(@numel,Element)); %Max. num. of vertices in mesh
PadWNaN = @(E) [E NaN(1,MaxNVer-numel(E))]; %Pad cells with NaN
ElemMat = cellfun(PadWNaN,Element,'UniformOutput',false);
ElemMat = vertcat(ElemMat{:}); %Create padded element matrix
patch('Faces',ElemMat,'Vertices',Node,'FaceColor','flat','FaceVertexCData',S);
;
%-----Acho o elemento com Maior tensão
for i = 1:4
NNodeInD = sum(cellfun(@numel,Element(D(:))));
NodeInD = zeros(NNodeInD,1);
hold on; index = 0;
for e=1:length(D)
el = D(e); NNodeElem = length(Element{el});
NodeInD(index+1:index+NNodeElem) = Element{el};
index = index + NNodeElem;
end
NodeInD = unique(NodeInD);
NodeInD = NodeInD(:);
d = Domain('Dist',NodeInD);
%-----Retiro dos nos escolhidos aqueles que estão na borda do domínio
I = find(d(:,end)<-1e-3);
NewNodes = NodeInD(I,:);
ID = (size(P,1)+1):(size([P;NewNodes],1));
%-----Refino a estrutura dado os nós de interesse
[Node,Element,Supp,Load,P] = PolyMesher(Domain,0,50,[P;NewNodes],ID);
%-----Crio nova FEM
[fem] = CreateFEMStruct(Node,Element,Supp,Load);
[~,S,fem] = FEAnalysis(fem);
%-----ploto a malha refinada
clf; axis equal; axis off; hold on;
MaxNVer = max(cellfun(@numel,Element)); %Max. num. of vertices in mesh
PadWNaN = @(E) [E NaN(1,MaxNVer-numel(E))]; %Pad cells with NaN
ElemMat = cellfun(PadWNaN,Element,'UniformOutput',false);
ElemMat = vertcat(ElemMat{:}); %Create padded element matrix
patch('Faces',ElemMat,'Vertices',Node,'FaceColor','flat','FaceVertexCData',S);
;
%-----Redefino os nós de interesse
Pc = PolyMshr_CntrdPly(Element,Node,size(Element,1));
```

```

D = find(sqrt((Pc(:,1)-p(1)).^2+(Pc(:,2)-p(2)).^2)-R<0);
ind = 1:length(Element); ind = ind';
Ss = [S ind]; SS=sortrows(Ss); SS = SS(:,2);
lim = length(D); tent = zeros(lim,2);
for e=1:length(D)
    tent(e,:) = [SS(e) D(e)];
end
SS=sortrows(tent);
SS = SS(:,2);
D= SS(0.7*ceil(length(SS)):length(SS));
end
%-----homogeneização da malha para todos os P
[Node,Element,Supp,Load,P] = PolyMesher(Domain,0,70,P);
[fem] = CreateFEMStruct(Node,Element,Supp,Load);
[~,S,fem] = FEAnalysis(fem);
clf; axis equal; axis off; hold on;
MaxNVer = max(cellfun(@numel,Element)); %Max. num. of vertices in mesh
PadWNaN = @(E) [E NaN(1,MaxNVer-numel(E))]; %Pad cells with NaN
ElemMat = cellfun(PadWNaN,Element,'UniformOutput',false);
ElemMat = vertcat(ElemMat{:}); %Create padded element matrix
patch('Faces',ElemMat,'Vertices',Node,'FaceColor','flat','FaceVertexCData',S);
;

end
%-----defino a função de busca de centroide
function [Pc,A] = PolyMshr_CntrdPly(Element,Node,NElem)
Pc=zeros(NElem,2); A=zeros(NElem,1);
for el = 1:NElem
    vx=Node(Element{el},1); vy=Node(Element{el},2); nv=length(Element{el});
    vxS=vx([2:nv 1]); vyS=vy([2:nv 1]); %Shifted vertices
    temp = vx.*vyS - vy.*vxS;
    A(el) = 0.5*sum(temp);
    Pc(el,:) = 1/(6*A(el,1))*[sum((vx+vxS).*temp),sum((vy+vyS).*temp)];
end
end
%-----defino a função de criação da FEM -----
function [fem] = CreateFEMStruct(Node,Element,Supp,Load)
fem = struct(...
    'NNode',size(Node,1),... % Number of nodes
    'NElem',size(Element,1),... % Number of elements
    'Node',Node,... % [NNode x 2] array of nodes
    'Element',{Element},... % [NElement x Var] cell array of elements
    'Supp',Supp,... % Array of supports
    'Load',Load,... % Array of loads
    'Nu0',0.3,... % Poisson's ratio of solid material
    'E0',1.0,... % Young's modulus of solid material
    'Reg',0 ... % Tag for regular meshes
);
end

```