



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL
CURSO DE GRADUAÇÃO EM ENGENHARIA CIVIL

MYRELLA VIEIRA CABRAL

**PROGRAMAS COMPUTACIONAIS PARA O REFINAMENTO H-
ADAPTATIVO: APLICAÇÃO A PROBLEMAS DE ENGENHARIA**

Recife
2018

MYRELLA VIEIRA CABRAL

**PROGRAMAS COMPUTACIONAIS PARA O REFINAMENTO H-ADAPTATIVO:
APLICAÇÃO A PROBLEMAS DE ENGENHARIA**

Monografia apresentada à Universidade Federal de Pernambuco como parte dos requisitos para obtenção de grau de Bacharel em Engenharia Civil.

Área de concentração: Estruturas

Orientador: Prof. Dr. Paulo Marcelo Vieira Ribeiro

Recife

2018

Catálogo na fonte

Bibliotecária Margareth Malta, CRB-4 / 1198

- C117p Cabral, Myrella Vieira.
Programas computacionais para o refinamento h-adaptativo: aplicação a problemas de engenharia / Myrella Vieira Cabral – 2018.
118 folhas, il., tabs.
- Orientador: Prof. Dr. Paulo Marcelo Vieira Ribeiro.
- TCC (Graduação) – Universidade Federal de Pernambuco. CTG. Departamento de Engenharia Civil e Ambiental, 2018.
Inclui Referências e Anexos.
1. Engenharia Civil. 2. Método dos Elementos Finitos. 3. Refinamento h-adaptativo. 4. Estimador de erro a posteriori. I. Ribeiro, Paulo Marcelo Vieira. (Orientador). II. Título.

624 CDD (22. ed.)

UFPE
BCTG/2018-308



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL
COORDENAÇÃO DO CURSO DE GRADUAÇÃO EM ENGENHARIA CIVIL

ATA DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO PARA CONCESSÃO DO GRAU DE ENGENHEIRO CIVIL

CANDIDATO: MYRELLA VIEIRA CABRAL

BANCA EXAMINADORA:

Orientador: Paulo Marcelo Vieira Ribeiro

Examinador 1: Ézio da Rocha Araújo

Examinador 2: Nadège Sophie Bouchonneau da Silva

TÍTULO DO TRABALHO DE CONCLUSÃO DE CURSO: PROGRAMAS COMPUTACIONAIS PARA O REFINAMENTO H-ADAPTATIVO: APLICAÇÃO A PROBLEMAS DE ENGENHARIA

LOCAL: 1º andar do prédio do Centro de Tecnologia e Geociências (CTG)

DATA: 27/07/2018 **HORÁRIO DE INÍCIO:** 10h.

Em sessão pública, após exposição de cerca de 30 minutos, o(s) candidato(s) foi (foram) arguido(s) oralmente pelos membros da banca com NOTA: 5,9 (deixar "Exame Final", quando for o caso).

1) ☒ **aprovado(s)** (nota > = 7,0), pois foi demonstrado suficiência de conhecimento e capacidade de sistematização no tema da monografia e o texto do trabalho aceito.

As revisões observadas pela banca examinadora deverão ser corrigidas e verificadas pelo orientador no prazo máximo de 30 dias (o verso da folha da ata poderá ser utilizado para portar revisões).

O trabalho com nota no seguinte intervalo, $3,0 \leq \text{nota} < 7,0$, será reapresentado, gerando-se uma nova ata; sendo o trabalho aprovado na reapresentação, o aluno será considerado **aprovado com exame final**.

2) ☐ **reprovado(s)**, (nota < 3,0)

Na forma regulamentar foi lavrada a presente ata que é assinada pelos membros da banca e pelo(s) candidato(s).

Recife, 27 de julho de 2018

Orientador:

Avaliador 1:

Avaliador 2:

Candidato 1:

Coordenação do Curso de Engenharia Civil-Civil
Rua Acadêmico Hélio Ramos s/nº, Cidade Universitária, Recife-PE CEP: 50740-530.
Fones: (081)2126.8220/8221 Fone/Fax: (081)2126.8229.

AGRADECIMENTOS

À Deus, por estar comigo em todos os momentos, pelo dom da vida e pelo seu amor.

Aos meus pais, por todo o incentivo e apoio e por fazerem parte de todas as minhas conquistas.

Ao meu professor orientador, Paulo Marcelo, que despertou em mim o interesse por esta área de pesquisa, me incentivou e assistiu durante a realização deste trabalho.

Aos meus colegas da equipe Mandacaru Aerodesign da UFPE, que contribuíram para o meu crescimento profissional e pessoal nos dois anos em que participei da equipe.

Aos amigos, familiares e a todos que de algum modo contribuíram para a conclusão deste ciclo.

RESUMO

Uma das maiores vantagens do Método dos Elementos Finitos (MEF) é a utilização de computadores convencionais para programação e análise de problemas de engenharia. No entanto, o procedimento de geração de malha (pré-processamento) muitas vezes exige um alto custo de processamento computacional e experiência do analista para geração de uma malha com elementos de tamanho adequado, principalmente em problemas onde há áreas mais sensíveis à análise, nas quais é necessário um número maior de elementos. Além da busca por uma melhor utilização dos recursos computacionais, procuram-se meios de garantir a precisão e convergência das soluções numéricas. O presente trabalho consiste no desenvolvimento de rotinas computacionais para o refinamento h-adaptativo de malhas aplicado a análise de problemas descritos pela Equação de Poisson e à Elasticidade Plana. Os códigos foram desenvolvidos no MATLAB e contam com rotinas para solução numérica pelo Método dos Elementos Finitos. A estimativa do erro *a posteriori* baseou-se na norma do erro em energia e na recuperação das derivadas de ordem superior pelo método das Médias Nodais Simples. Através do critério de equidistribuição do erro elementar e do erro percentual admissível em norma de energia, os níveis de refinamento dos elementos são determinados. Em seguida, uma série de medidas são tomadas para evitar o refinamento excessivo e promover uma melhor regularização da malha. A validação do código foi feita por comparação com soluções analíticas e resultados da literatura.

Palavras-chave: Método dos elementos finitos. Refinamento h-adaptativo. Estimador de erro *a posteriori*.

ABSTRACT

One of the main advantages of the Finite Element Method is the usage of conventional computers for programming and analysis of engineering problems. However, the mesh generation procedure often requires a high cost of computational processing and user experience to generate element sizes that provide an acceptable solution, especially when there are sensitive areas, where a higher number of elements is needed. Besides the search for better use of computational resources, it is also important to guarantee precision and convergence of the numerical solutions. The present work consists of the development of computational routines of h-adaptive mesh refinement applied to Poisson Equation and Plane Elasticity problems. The codes were developed in MATLAB and rely on routines for numerical solution using the Finite Element Method. The *posteriori* error estimate is based on error in the energy norm and recovery of the higher order derivatives by the method of Simple Nodal Averaging. By means of the equidistribution criterion of the elementary error and the permissible percentage error in the energy norm, the refinement levels of the elements are determined. Then a series of procedures are done to avoid excessive refinement and aid better shape of the mesh elements. The validation of the codes was made through the comparison with analytical solutions and literature results.

Keywords: Finite element method. H-adaptive refinement. *Posteriori* error estimate.

LISTA DE FIGURAS

Figura 1 - Subdivisão do contorno do corpo.....	18
Figura 2 - Elemento Triangular Linerar (KWON E BANG, 1997).	19
Figura 4 - Diagrama de corpo livre de um elemento bidimensional infinitesimal.....	21
Figura 5 - Deslocamentos e rotação das linhas de um elemento no plano xy.....	24
Figura 6 - Elemento CST	25
Figura 7 - Força externa distribuída.....	30
Figura 8 - Forças nodais equivalentes.....	30
Figura 9 - Análise no COMSOL com refinamento h-adaptativo.....	31
Figura 10- Triângulo vermelho refinado com triângulos vizinhos de transição.	32
Figura 11- Comparação entre as malhas: Original, Subdivisão de elemento e Regeneração, respectivamente. ZIENKIEWICZ & TAYLOR (2000).	32
Figura 12- Comparação entre o gradiente obtido via MEF (a) e o gradiente suavizado (b).....	36
Figura 13- Determinação do grau de refinamento.	40
Figura 14- Primeira etapa de regularização da malha.....	44
Figura 15- Segunda etapa de regularização da malha.....	45
Figura 16- Fluxograma dos procedimentos do programa PoissonG.....	48
Figura 17- Fluxograma dos procedimentos do programa ELT2D.	50
Figura 18- Evolução da malha – Condução de Calor.	51
Figura 19- Campo de temperatura.....	52
Figura 20- Erro na norma em energia versus NGL.....	52
Figura 21- Análise do esforço computacional.	53
Figura 22- Número de Nós versus Erro de aproximação (e) – Problema de condução de calor.....	54
Figura 23- Apresentação do problema de flexão de placas.....	55
Figura 24- Malha inicial -16 elementos.....	55
Figura 25- 1ª malha adaptada – 216 elementos.....	55
Figura 26- 2ª malha adaptada – 1386 elementos.....	56
Figura 27- Erro na norma em energia versus NGL.....	57
Figura 28- Deslocamentos verticais – solução numérica.	58
Figura 29- Deslocamentos verticais – solução analítica.	58

Figura 30- Número de Nós versus Erro de aproximação (e) – Deslocamentos verticais.	59
Figura 31- Malha inicial – 8 elementos.....	60
Figura 32- 1ª Malha adaptada – 128 elementos.	60
Figura 33- 2ª Malha adaptada – 1198 elementos.	60
Figura 34- Malha final do refinamento no domínio C – 2480 elementos.....	61
Figura 35- Apresentação do problema da viga engastada em balanço.	62
Figura 36- Evolução da malha da viga engastada em balanço.....	62
Figura 37- Erro na norma em energia versus NGL.....	64
Figura 38- Análise do esforço computacional.	64
Figura 39- Número de Nós versus Erro de aproximação (e) – Deslocamentos verticais.	66
Figura 40- Número de Nós versus Erro de aproximação (e) – Deslocamentos horizontais.....	66
Figura 41- Distribuição de deslocamentos em y da viga engastada em balanço.	67
Figura 42- Distribuição de deslocamentos em x da viga engastada em balanço.	67
Figura 43- Distribuição de σ_x da viga engastada em balanço.	67
Figura 44- Distribuição de σ_y da viga engastada em balanço.....	67
Figura 45- Distribuição de τ_{xy} da viga engastada em balanço.....	67
Figura 46- Distribuição do erro percentual de tensões normais mínimas.....	68
Figura 47- Distribuição do erro percentual de tensões normais máximas.	68
Figura 48- Histograma da distribuição do erro percentual das tensões normais.....	68
Figura 49- Apresentação do problema da chapa com furo.	69
Figura 50- Evolução da malha da chapa com furo.....	70
Figura 51- Erro na norma em energia versus NGL.....	71
Figura 52- Análise do esforço computacional.	72
Figura 53- Distribuição de deslocamentos em y da chapa com furo.	73
Figura 54- Distribuição de deslocamentos em x da chapa com furo.	73
Figura 55- Distribuição de σ_x da chapa com furo.....	74
Figura 56- Distribuição de σ_y da chapa com furo.....	74
Figura 57- Distribuição de τ_{xy} da chapa com furo.....	75
Figura 58- Distribuição do erro percentual de tensões normais máximas.	75
Figura 59- Distribuição do erro percentual de tensões normais mínimas.....	76
Figura 60- Histograma do erro percentual de tensões normais máximas.	76

Figura 61- Histograma do erro percentual de tensões normais mínimas.	77
Figura 62- Pontos de integração para os elementos triangulares planos.....	82

LISTA DE TABELAS

Tabela 1 - Resultados do refinamento adaptativo da placa sob flexão.	56
Tabela 2 - Resultados do refinamento uniforme da placa sob flexão.	56
Tabela 3 - Comportamento da rigidez torcional.	60
Tabela 4 - Comportamento da rigidez torcional.	61
Tabela 5 - Resultados do refinamento adaptativo da viga engastada em balanço.	63
Tabela 6 - Resultados do refinamento uniforme da viga engastada em balanço.	63
Tabela 7 - Resultados do refinamento adaptativo da chapa com furo.	70
Tabela 8 - Resultados do refinamento uniforme da chapa com furo.	71
Tabela 9 - Pontos e pesos (Hammer).	83

LISTA DE ABREVIATURAS

CST	<i>Constant Strain Triangle</i> (Elemento triangular de deformação constante)
HD	<i>Hard Disk</i> (Disco Rígido)
LST	<i>Linear Strain Triangle</i> (Elemento triangular linear)
MEF	Método dos Elementos Finitos
MNS	Média Nodal Simples
NGL	Número de Graus de Liberdade
RAM	<i>Random Access Memory</i> (Memória de Acesso Rápido)
REP	<i>Recovery by Equilibration of Patches</i> (Recuperação por Equilíbrio de Padrões)
SPR	<i>Superconvergent Patch Recovery</i> (Superconvergente de Recuperação de Padrões)
ZZ	Zienkiewicz e Zhu

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa e motivação	15
1.2	Objetivos gerais e específicos.....	16
2	REFERENCIAL TEÓRICO	17
2.1	Equação de Poisson	17
2.1.1	Aplicação do Método dos Elementos Finitos	19
2.2	Elasticidade Plana	21
2.2.1	Estado plano de tensão e estado plano de deformação	21
2.2.2	Estado bidimensional de tensão e deformação	23
2.2.3	Aplicação do Método dos Elementos Finitos	24
2.2.3.1	<i>Constant Strain Triangle (CST)</i>	24
2.2.3.2	<i>Matriz de rigidez</i>	27
2.2.3.3	<i>Forças de corpo</i>	28
2.2.3.4	<i>Forças externas</i>	29
2.2.3.5	<i>Forças totais do sistema</i>	30
2.2.3.6	<i>Equação básica</i>	30
2.3	Refinamento Adaptativo	31
2.4	Estimativa de erro no Método dos Elementos Finitos.....	33
2.4.1	Estimador de erro baseado na norma em energia	33
2.4.2	Método da Média Nodal Simples (MNS)	36
2.5	Definição dos critérios de convergência e dos parâmetros do processo adaptativo	37
2.5.1	Grau de refinamento	39
3	METODOLOGIA ADOTADA E PARTICULARIDADES DOS CÓDIGOS.....	42
3.1	Determinação do nível de refinamento para cada elemento	42
3.1.1	Desnível unitário interelementos	42
3.1.2	Regularização da malha.....	43
3.1.3	Dimensão mínima dos elementos da malha	45
3.2	Estrutura dos programas	45
3.2.1	<i>PoissonG</i>	46
3.2.2	<i>ELT2D</i>	48

4	RESULTADOS E DISCUSSÕES.....	51
4.1	Problemas descritos pela Equação de Poisson	51
4.1.1	Condução de Calor	51
4.1.2	Flexão de placas.....	54
4.1.3	Torção	59
4.2	Elasticidade Plana.....	61
4.2.1	Viga engastada em balanço	61
4.2.2	Chapa com furo	69
5	CONCLUSÕES.....	78
5.1	Sugestões para trabalhos futuros	79
	REFERÊNCIAS	80
	APÊNDICE A – INTEGRAÇÃO NUMÉRICA.....	82
	APÊNDICE B – CÓDIGO PARA O REFINAMENTO H-ADAPTATIVO APLICADO À ELASTICIDADE PLANA – <i>ELT2D</i>	84
	APÊNDICE C – CÓDIGO PARA O REFINAMENTO H-ADAPTATIVO APLICADO A PROBLEMAS DESCRITOS PELA EQUAÇÃO DE POISSON – <i>PoissonG</i>.....	103

1 INTRODUÇÃO

Os desafios atribuídos à engenharia estrutural pelos novos conceitos arquitetônicos e pelas obras modernas, cada vez mais imponentes, exigem que as ferramentas associadas ao projeto dessas estruturas se tornem cada vez mais sofisticadas e eficientes. Isto porque o projeto de engenharia precisa adequar-se à sua funcionalidade, de maneira que o custo da estrutura torne viável a execução da obra. Para isso é necessário que se conheça, da maneira mais representativa possível, os fenômenos associados ao seu comportamento.

Diante disso, técnicas numéricas de análise de estruturas foram desenvolvidas e vêm sendo aprimoradas ao longo dos anos. Sendo utilizadas para a solução de problemas de diversas naturezas de acordo com suas características e capacidade de representar e capturar singularidades associadas a cada contexto envolvido. Nos métodos com malha, a geometria é subdividida em pequenas partes, denominadas de elementos, os quais passam a representar o domínio contínuo do problema. Dessa forma, o Método dos Elementos Finitos (MEF) propõe que a superfície a ser analisada seja dividida em um número limitado de elementos, tornando possível a representação de uma função complexa como uma união de funções polinomiais.

Paralelamente ao desenvolvimento dos métodos numéricos, surgiram as pesquisas relacionadas à estimativa dos erros intrínsecos aos processos de modelagem computacional. São várias as fontes de erros presentes no MEF, entre os quais, tem-se os erros de aproximação do domínio, erros de aritmética finita e quadratura (provenientes de integração numérica e de arredondamentos) e erros de aproximação (decorrentes da substituição da solução exata do problema por uma função polinomial) (REDDY, 2006).

Sabe-se que regiões mais refinadas, com maior número de elementos, permitem maior precisão ao passo que demandam elevado custo computacional. E dependendo do problema podem existir regiões mais sensíveis à análise, nas quais é necessário um número maior de graus de liberdade para que a solução seja obtida com a precisão desejada. Dentro deste contexto, surgiram as técnicas de refinamento adaptativo que tem por objetivo a formação de uma malha “ótima”, isto é, uma malha que minimize os erros de discretização com o menor número de graus de liberdade possível.

Dentre os diversos tipos de refinamento adaptativo de malhas de soluções em elementos finitos, neste trabalho optou-se pela utilização do refinamento h-adaptativo, que consiste na subdivisão dos elementos das regiões de maiores erros. A estimativa do erro é feita *a posteriori* e baseia-se na recuperação das derivadas de ordem superior pelo método das Médias Nodais

Simples e no cálculo do erro na norma em energia (ZIENCKIEWICZ & TAYLOR, 1989), que tem por objetivo a obtenção de um escalar que meça o erro em um sentido global para a malha corrente

Serão apresentados a fundamentação teórica do Método dos Elementos Finitos, aplicado a problemas descritos pela Equação de Poisson e a Elasticidade Plana, bem como o cálculo das estimativas de erro e o critério de discretização associados às estratégias de refinamento h-adaptativo e as particularidades dos programas desenvolvidos. Por fim, serão apresentados e discutidos cinco exemplos numéricos: flexão de placas, condução de calor, torção, viga engastada em balanço e chapa com furo sob tração.

1.1 Justificativa e motivação

Uma das maiores vantagens do MEF é a utilização de computadores convencionais para programação e análise de problemas diversos. Em particular, qualquer domínio físico com geometria complexa e condições de contorno prescritas pode ser analisado com o método (KWON e BANG, 1997). Por outro lado, o procedimento de geração de malhas (pré-processamento) muitas vezes exige um alto custo de processamento computacional para geração de uma malha que forneça uma solução aceitável, principalmente em problemas onde há áreas mais sensíveis à análise, nas quais é necessário um número maior de elementos para que a solução seja obtida com a precisão desejada. Além disso, vários programas não dispõem de uma estimativa do erro da análise, ficando a cargo do analista a verificação da confiabilidade dos resultados, quer seja por sucessivas discretizações, execuções e verificações, quer seja pela experiência do indivíduo naquele tipo de análise.

O desenvolvimento de algoritmos capazes de minimizar os erros associados à discretização do domínio e gerar, automaticamente, uma malha adequada ao problema é de grande valia. Diante disso, foram desenvolvidos dois programas de refinamento adaptativo de malha aplicados a problemas descritos pela Equação de Poisson e à Elasticidade Plana, que abrangem problemas presentes no cotidiano da engenharia. A finalidade do trabalho vai além do desenvolvimento das rotinas computacionais, ao promover a discussão e a implementação de estratégias de refinamento adaptativo acessíveis a alunos de graduação, espera-se contribuir para a criação de um ambiente propício ao desenvolvimento de novas ideias e técnicas.

1.2 Objetivos gerais e específicos

O objetivo deste trabalho consiste no desenvolvimento de rotinas computacionais de refinamento h-adaptativo de malha aplicado a problemas descritos pela Equação de Poisson e a problemas de Elasticidade plana. Também foi implementado os solucionadores MEF, necessários às análises. É importante destacar que as estratégias adaptativas implementadas podem ser aplicadas em outros tipos de problemas estacionários bidimensionais que utilizem malhas conformes.

Os objetivos específicos são:

- Desenvolver ferramentas computacionais de refinamento h-adaptativo de malhas;
- Demonstrar a eficiência do refinamento h-adaptativo e sua vantagem em termos de custo computacional;
- Aplicar a problemas bidimensionais de interesse prático.

2 REFERENCIAL TEÓRICO

Serão apresentadas a equação de Poisson e as equações básicas da teoria de elasticidade, seguidas pela aplicação do Método dos Elementos Finitos. Foram utilizados elementos triangulares, triangular linear e CST (*Constant Strain Triangle*), pois as expressões relacionadas aos elementos triangulares são comparativamente mais simples e os contornos das geometrias podem ser melhor aproximados. Além disso, esse tipo de elemento facilita a implementação das estratégias de refinamento h-adaptativo porque é mais fácil de tratar os *hanging nodes*, que surgem devido a diferença de nível de refinamento entre elementos vizinhos, comparado a elementos quadrangulares, por exemplo.

2.1 Equação de Poisson

As equações de Laplace e Poisson possuem múltiplas aplicações na engenharia. Por exemplo, essas equações diferenciais podem representar condução de calor, fluxo potencial e torção de estruturas (KWON & BANG, 1997). A Equação de Poisson é:

$$k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = g(x, y) \text{ em } \Omega \quad (1)$$

Onde a incógnita u é uma função ao longo do domínio Ω e $g(x,y)$ é uma função denominada fonte interna. Para o caso particular onde não há fonte interna, $g(x,y)=0$, obtém-se a Equação de Laplace:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \text{ em } \Omega \quad (2)$$

Para o domínio bidimensional, as condições de contorno são:

$$u = \bar{u} \text{ em } \Gamma_e \quad (3)$$

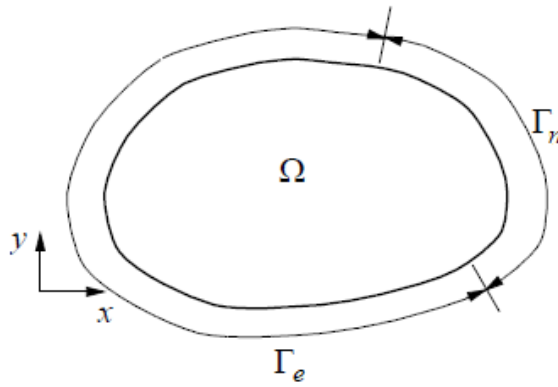
$$\frac{\partial u}{\partial n} = \bar{q} \text{ em } \Gamma_n \quad (4)$$

Onde $\bar{u}$ é a condição de contorno de Dirichlet, referente a imposição do valor de u na região Γ_e do contorno e $\bar{q}$ é a condição de contorno de Neumann, referente a imposição de fluxo em Γ_n , Fig. 1. Neste caso, considera-se o “equilíbrio” de fluxo em um elemento infinitesimal da região Γ_n do contorno. Portanto,

$$\Gamma_e \cup \Gamma_n = \Gamma \quad (5)$$

$$\Gamma_e \cap \Gamma_n = \emptyset \quad (6)$$

Figura 1 - Subdivisão do contorno do corpo.



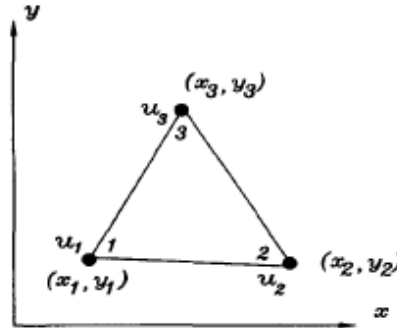
Fonte: Adaptado de SOUZA (2003).

Onde $\cup$ e $\cap$ denotam soma e interseção, respectivamente, e Γ é a condição de contorno total do domínio. A integração do peso residual da Eq. (1) e da condição de contorno, após a realização de integração por partes para redução da ordem de diferenciação da integral, resulta em (KNOW & BANG, 1997):

$$I = - \int_{\Omega} \left(\frac{\partial w}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial u}{\partial y} \right) d\Omega - \int_{\Omega} w g(x, y) d\Omega + \int_{\Gamma_n} w \frac{\partial u}{\partial n} d\Gamma \quad (7)$$

2.1.1 Aplicação do Método dos Elementos Finitos

Figura 2 - Elemento Triangular Linear (KWON E BANG, 1997).



Fonte: KWON E BANG (1997).

A discretização do domínio na Eq. (7) é feita usando elementos finitos bidimensionais. Um dos elementos mais simples é o elemento triangular linear, Fig.2, utilizado neste trabalho. A função de interpolação do elemento é linear em x e y e é expressa por:

$$u = a_1 + a_2x + a_3y \quad (8)$$

A função geral ψ , que interpola o campo a partir das variáveis nodais, é (LOGAN, 2007):

$$\{\psi\} = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \quad (9)$$

Onde $H_1(x, y)$, $H_2(x, y)$ e $H_3(x, y)$ são as funções de forma do elemento triangular linear. A primeira integral da Eq. (7) torna-se a matriz de rigidez, a segunda integral está relacionada a fonte interna e torna-se um vetor de fluxos e a terceira integral refere-se ao fluxo no contorno ao longo da fronteira natural, também resultando num vetor de fluxos.

Com as variáveis nodais indicadas na Eq. (9), a matriz de condução do elemento pode ser obtida por:

$$[K^e] = \int_{\Omega^e} [B]^T [\kappa] [B] d\Omega \quad (10)$$

Onde $[B]$ corresponde às derivadas das funções de forma H_i e $[\kappa]$ corresponde à matriz constitutiva:

$$[B] = \frac{1}{2A} \begin{bmatrix} (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix} \quad (11)$$

$$[\kappa] = \kappa(x, y) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (12)$$

A segunda integral da Eq. (7) é calculada apenas para os problemas descritos pela equação de Poisson, já que para Laplace a fonte interna de calor, $g(x, y)$, é nula em todo o domínio Ω . A integral é computada no elemento triangular linear da seguinte forma (KWON & BANG, 1997):

$$\{f^{(e)}\} = \int_{\Omega^e} \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix} g(x, y) d\Omega \quad (13)$$

A matriz global de condução é obtida pela Eq. (14), onde nel corresponde ao número de elementos da malha.

$$[K] = \sum_{e=1}^{nel} [K^{(e)}] \quad (14)$$

O vetor de fluxos nodais é obtido pela soma de todas as fontes dos elementos e é dado por:

$$[F] = \sum_{e=1}^{nel} \{f^{(e)}\} \quad (15)$$

A equação diferencial que governa o problema é transformada em um sistema de equações algébricas, Eq. (16), com as condições de contorno prescritas dadas pela Eq. (3). As

condições de contorno do fluxo de calor, Eq. (4) são computadas da mesma maneira que cargas distribuídas para um problema de análise de tensão, ou seja, são incluídas no vetor de fluxos nodais (LOGAN, 2007).

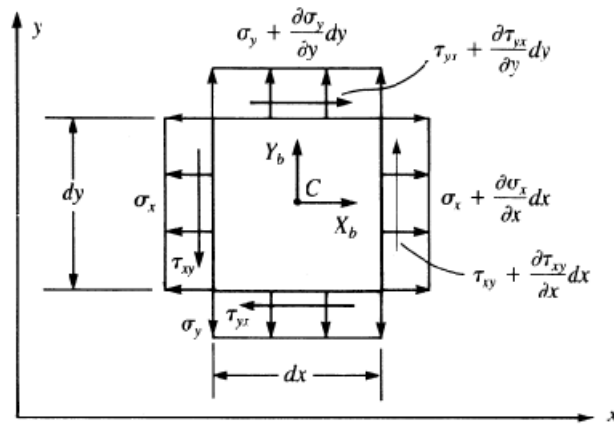
$$[F] = [K]\{u\} \quad (16)$$

2.2 Elasticidade Plana

2.2.1 Estado plano de tensão e estado plano de deformação

Considerando um diagrama de corpo livre de um elemento infinitesimal como mostrado na Fig. 3, o somatório das forças nos eixos vertical e horizontal tornam-se (KNOW & BANG, 2000):

Figura 3 - Diagrama de corpo livre de um elemento bidimensional infinitesimal.



Fonte: LOGAN (2007).

$$\sum F_x = (\sigma_x + \frac{\partial \sigma_x}{\partial x} dx) dy - \sigma_x dx dy + (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial y} dy) dx - \tau_{xy} dx dy + f_x dx dy = 0 \quad (17)$$

$$\sum F_y = (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial x} dx) dy - \tau_{xy} dx dy + (\sigma_y + \frac{\partial \sigma_y}{\partial y} dy) dx - \sigma_y dx dy + f_y dx dy = 0 \quad (18)$$

Onde f_x e f_y são as forças de corpo por unidade de área, σ_x é a tensão em x, σ_y é a tensão em y e τ_{xy} é a tensão cisalhante. Simplificando essas expressões, obtém-se as equações de equilíbrio:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x = 0 \quad (19)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y = 0 \quad (20)$$

Segundo LOGAN (2007), a terceira equação de equilíbrio que deve ser satisfeita é a equação de equilíbrio dos momentos em relação a um eixo normal ao plano xy. Tomando os momentos em relação ao ponto C da Fig. 3, tem-se:

$$\sum M_z = \tau_{xy} dy \frac{dx}{dy} + (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial x} dx) \frac{dx}{2} - \tau_{yx} dx \frac{dy}{2} - (\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy) \frac{dy}{2} = 0 \quad (21)$$

Simplificando e negligenciando os termos de alta ordem, obtém-se:

$$\tau_{xy} = \tau_{yx} \quad (22)$$

A relação entre tensões e deformações é dada pela expressão:

$$\{\sigma\} = [D]\{\varepsilon\} \quad (23)$$

Onde $\{\sigma\} = \{\sigma_x \ \sigma_y \ \tau_{xy}\}^T$ são as tensões e $\{\varepsilon\} = \{\varepsilon_x \ \varepsilon_y \ \gamma_{xy}\}^T$ são as deformações, as quais serão definidas na seção 2.2.2. Para a condição de Estado Plano de Tensão, tem-se: $\sigma_z = \tau_{xy} = \tau_{yx} = 0$, $\gamma_{xz} = \gamma_{yz} = 0$ e $\varepsilon_z \neq 0$. Ou seja, existe deformações tridimensionais no elemento, portanto, existe deformação no eixo z, mas não existe tensão. A matriz de propriedades do material é dada por:

$$[D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (24)$$

Onde E é o módulo de elasticidade do material e ν é o coeficiente de Poisson. Para a condição de Estado Plano de Deformação, tem-se: $\tau_{xy} = \tau_{yx} = 0$, $\varepsilon_z = \gamma_{xz} = \gamma_{yz} = 0$ e $\sigma_z \neq 0$. Ou seja, existem tensões tridimensionais atuando no elemento e não há deformação no eixo z . A matriz de propriedades do material é dada por:

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (25)$$

Em suma, na condição de estado plano de tensão, o elemento sofre deformações tridimensionais e na condição de estado plano de deformação, há tensões tridimensionais agindo sobre o elemento. Em todos os exemplos apresentados neste trabalho foi considerado o primeiro cenário, com espessura (dimensão ao longo de z) pequena e cargas apenas nas direções x e y .

2.2.2 Estado bidimensional de tensão e deformação

O estado principal de tensões, no qual ocorre as tensões normais máxima e mínima bidimensionais pode ser obtido segundo as expressões (GERE, 2001):

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} = \sigma_{\max} \quad (26)$$

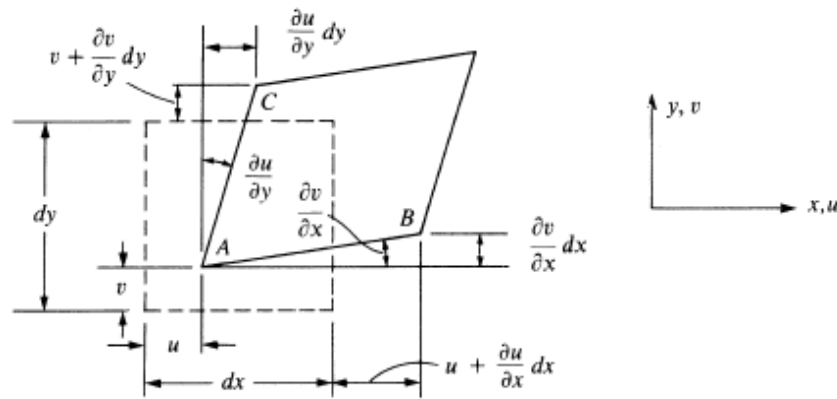
$$\sigma_2 = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} = \sigma_{\min} \quad (27)$$

A direção normal ao plano no qual ocorrem as tensões normais máximas e mínimas é definida pelo ângulo principal, θ_p , dado por:

$$\tan 2\theta_p = \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \quad (28)$$

A Figura 4 apresenta um elemento infinitesimal utilizado para representar o estado infinitesimal de deformação em algum ponto da estrutura.

Figura 4 - Deslocamentos e rotação das linhas de um elemento no plano xy.



Fonte: LOGAN (2007).

Das definições gerais de deformações normais e cisalhantes e fazendo uso da Fig. 4, obtém-se (LOGAN, 2007):

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad (29)$$

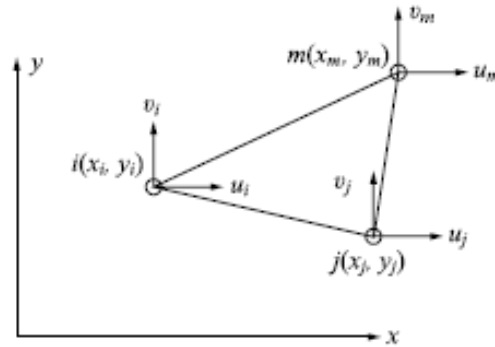
$$\varepsilon_y = \frac{\partial v}{\partial y} \quad (30)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (31)$$

2.2.3 Aplicação do Método dos Elementos Finitos

2.2.3.1 Constant Strain Triangle (CST)

Figura 5 - Elemento CST



Fonte: LOGAN (2007).

O elemento CST possui dois graus de liberdade por nó, Fig. 5, e as tensões e deformações atuantes são constantes no elemento. A função linear de deslocamentos, que assegura que a compatibilidade será satisfeita para cada elemento é dada por (LOGAN, 2007):

$$u(x, y) = a_1 + a_2x + a_3y \quad (32)$$

$$v(x, y) = a_4 + a_5x + a_6y \quad (33)$$

Onde $u(x, y)$ e $v(x, y)$ descrevem os deslocamentos em cada ponto interior (x_i, y_i) do elemento e os a 's representam os coeficientes generalizados. São utilizadas funções de forma N para expressar as funções de deslocamento no interior de um elemento. Usando as Eq. (32) e (33) e fazendo algumas operações, conforme LOGAN (2007), obtém-se a função geral de deslocamentos expressa em função dos valores dos deslocamentos nodais e das funções de forma, N .

$$\{\psi\} = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} = \begin{Bmatrix} N_i u_i + N_j u_j + N_m u_m \\ N_i v_i + N_j v_j + N_m v_m \end{Bmatrix} \quad (34)$$

Ou

$$\{\psi\} = \begin{bmatrix} N_i & 0 & N_j & 0 & N_m & 0 \\ 0 & N_i & 0 & N_j & 0 & N_m \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} \quad (35)$$

De maneira abreviada, a Eq. (35) torna-se $\{\psi\} = [N]\{d\}$. Deve-se, então, expressar as deformações e tensões dos elementos em termos de deslocamentos nodais. Partindo da Eq. (36), obtém-se a expressão dada pela Eq. (37) que relaciona as deformações com os deslocamentos nodais.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} \quad (36)$$

$$\{\varepsilon\} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & \frac{\partial N_j}{\partial x} & 0 & \frac{\partial N_m}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 & \frac{\partial N_j}{\partial y} & 0 & \frac{\partial N_m}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial y} & \frac{\partial N_j}{\partial x} & \frac{\partial N_m}{\partial y} & \frac{\partial N_m}{\partial x} \end{bmatrix} \{d\} \quad (37)$$

De maneira compacta, tem-se $\{\varepsilon\} = [B]\{d\}$. Como as deformações são constantes no elemento CST (*Constant Strain Triangle*), segundo KWON & BANG (1997), a avaliação das funções de forma resulta em:

$$[B] = \frac{1}{2A} \begin{bmatrix} (y_j - y_m) & 0 & (y_m - y_i) & 0 & (y_i - y_j) & 0 \\ 0 & (x_m - x_j) & 0 & (x_i - x_m) & 0 & (x_j - x_i) \\ (x_m - x_j) & (y_j - y_m) & (x_i - x_m) & (y_m - y_i) & (x_j - x_i) & (y_i - y_j) \end{bmatrix} \quad (38)$$

E as tensões no plano, em termos dos graus de liberdade dos nós, são dadas por:

$$\{\sigma\} = [D][B]\{d\} \quad (39)$$

2.2.3.2 Matriz de rigidez

Usando o princípio da energia potencial mínima, pode-se gerar equações para um elemento CST. Para o plano de tensão do elemento, a energia potencial é uma função dos deslocamentos nodais, $\{d\}$. LOGAN (2007) apresenta, de maneira detalhada, a expressão da energia potencial total, π_p , constituída por: energia de deformação, energia potencial de forças de corpo, energia potencial de cargas concentradas e energia potencial de cargas distribuídas. A derivada de π_p com respeito aos deslocamentos nodais, já que $\pi_p = \pi_p(d)$, é dada por:

$$\frac{\partial \pi_p}{\partial \{d\}} = \left[\iiint_V [B]^T [D][B] dV \right] \{d\} - \{f\} = 0 \quad (40)$$

Onde $\{f\}$ representa a força total do sistema atuante em um elemento. O primeiro termo da Eq. (41) refere-se às forças de corpo, o segundo termo refere-se às forças nodais concentradas e o terceiro termo às cargas distribuídas.

$$\{f\} = \iiint_V [N]^T \{X\} dV + \{P\} + \iint_S [N_S]^T \{T_S\} dS \quad (41)$$

Onde $\{X\}$ é o peso do corpo por unidade de volume ou matriz de densidade (geralmente em kN/m³), $\{P\}$ é a carga externa concentrada, $[N_S]$ representa a matriz de funções de forma avaliada ao longo da superfície na qual as cargas distribuídas estão atuando e $\{T_S\}$ representa as cargas distribuídas (geralmente em kN/m²).

A matriz de rigidez elementar corresponde ao primeiro termo do lado direito da Eq. (40) e é dada por:

$$[K^e] = \iiint_V [B]^T [D] [B] dV \quad (42)$$

Como o problema é bidimensional, a espessura do elemento é constante. Além disso, o elemento é CST, portanto, os termos integrandos não são uma função de x e y . Consequentemente, a Eq. (42) torna-se a Eq. (43), onde A é a área do elemento e t é a espessura (KWON & BANG, 1997).

$$[K^e] = tA[B][D][B] \quad (43)$$

A matriz de rigidez do sistema é, então, dada pela soma das contribuições das matrizes de rigidez de cada elemento:

$$[K] = \sum_{e=1}^{nel} [K^{(e)}] \quad (44)$$

2.2.3.3 Forças de corpo

Partindo do primeiro termo da Eq. (41), pode-se avaliar as forças de corpo nos nós de cada elemento (LOGAN, 2007):

$$\{f_b\} = \iiint_V [N]^T \{X\} dV \quad (45)$$

Onde

$$\{X\} = \begin{Bmatrix} X_b \\ Y_b \end{Bmatrix} \quad (46)$$

E X_b e Y_b são forças por unidade de volume (kN/m^3) nas direções x e y , respectivamente. Essas forças surgem devido ao peso próprio da estrutura (forças gravitacionais), velocidade angular (denominada de força de corpo centrífuga) ou forças inerciais e dinâmicas. Para o caso em que atuam apenas forças de corpo devido ao peso próprio da estrutura, como a força

gravitacional está na direção y , tem-se apenas Y_b ($X_b=0$). Simplificando a integração da Eq. (45), sem perda de generalidade, obtém-se (LOGAN, 2007):

$$\{f_b\} = \begin{Bmatrix} f_{bix} \\ f_{biy} \\ f_{bjx} \\ f_{bjy} \\ f_{bmx} \\ f_{bmy} \end{Bmatrix} = \begin{Bmatrix} X_b \\ Y_b \\ X_b \\ Y_b \\ X_b \\ Y_b \end{Bmatrix} \frac{At}{3} \quad (47)$$

2.2.3.4 Forças externas

As forças externas atuantes em um corpo podem ser concentradas ou distribuídas. Trataremos apenas de forças distribuídas, pois o código implementado não permite a presença de forças concentradas em apenas um nó. Quando houver força concentrada, esta deverá ser substituída por uma força distribuída estaticamente equivalente, ao longo da direção de atuação da força concentrada. Pelo terceiro termo da Eq. (41), pode-se avaliar as forças externas distribuídas como:

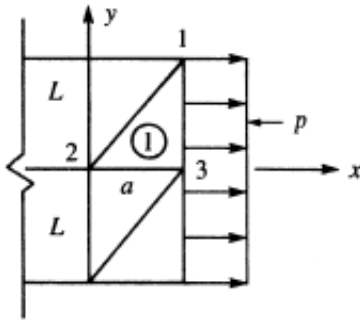
$$\{f_s\} = \iint_S [N_S]^T \{T_S\} dS \quad (48)$$

Onde

$$\{T_S\} = \begin{Bmatrix} p_x \\ p_y \end{Bmatrix} \quad (49)$$

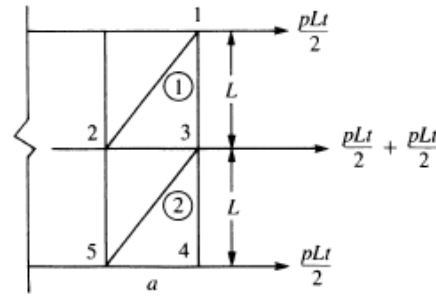
Sendo p_x e p_y os valores das forças externas distribuídas nas direções x e y , respectivamente. Para um elemento CST, as cargas distribuídas ao longo dos lados de um elemento podem ser tratadas como cargas concentradas atuantes nos nós associados ao lado carregado, transformando cargas distribuídas em cargas nodais estaticamente equivalentes, Fig. 6 e 7. No entanto, para elementos de alta ordem, como os LST (*Linear Strain Triangle*), a transformação da força distribuída em força nodal equivalente deve ser feita usando a Eq. (48), a qual foi obtida pelo princípio da energia potencial mínima (LOGAN, 2007).

Figura 6 - Força externa distribuída.



Fonte: LOGAN (2007).

Figura 7 - Forças nodais equivalentes.



Fonte: LOGAN (2007).

2.2.3.5 Forças totais do sistema

Conforme apresentado nas seções anteriores, a força total atuante no sistema é composta por forças de corpo e forças externas. Portanto, o vetor de forças totais é obtido pela soma das contribuições de cada uma dessas parcelas, Eq. (50), onde nel corresponde ao número de elementos da malha.

$$[F] = \sum_{e=1}^{nel} \left[\{f_s^{(e)}\} + \{f_b^{(e)}\} \right] \quad (50)$$

2.2.3.6 Equação Básica

É obtida pelo rearranjo da Eq. (40), proveniente da aplicação do princípio da energia potencial total mínima que estabelece: “*de todas as configurações cinematicamente possíveis, a deformação que produzir a energia potencial total mínima é a condição de equilíbrio estável*” (KWON & BANG, 1997).

$$[F] = [K]\{d\} \quad (51)$$

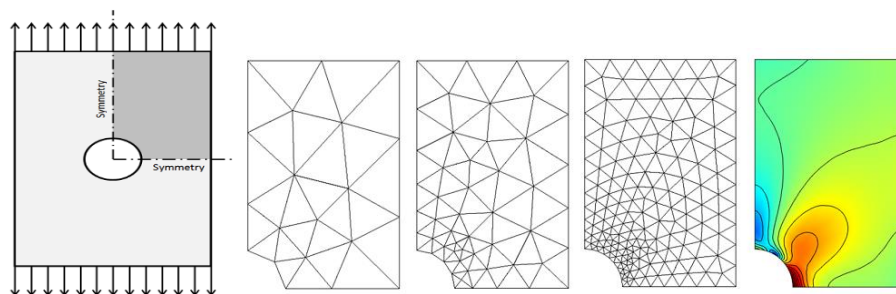
Onde $[K]$ é a matriz de rigidez do sistema, Eq. (44), $[F]$ é o vetor de forças totais atuantes no sistema, Eq. (50), e $\{d\}$ é o vetor de deslocamentos nodais.

2.3 Refinamento Adaptativo

Processos de refinamento adaptativo de discretização de malhas tem sido foco de intensivas pesquisas nas últimas décadas, mostrando-se uma ferramenta eficiente para a análise por elementos finitos de diversos problemas de Engenharia. Sabe-se que regiões mais refinadas permitem maior acurácia ao passo que demandam um maior custo computacional, por esse motivo, é importante que o refinamento seja distribuído de forma inteligente ao longo do domínio.

Existem diversos procedimentos para o refinamento de soluções em Elementos Finitos. Uma subdivisão clássica é apresentada a seguir: Refinamento-p, Refinamento-r e Refinamento-h. O primeiro propõe o refinamento da malha por meio da alteração da ordem do polinômio usado na interpolação, geralmente alterado hierarquicamente, mantendo-se o tamanho dos elementos. O segundo mantém o número total de nós constantes e ajusta a suas posições para obter uma aproximação ótima. Esse procedimento é teoricamente interessante, mas é difícil de usar na prática e é pouco recomendado. No Refinamento-h, alteram-se as dimensões e/ou formas dos elementos de maneira que em alguns locais os elementos possuem menores dimensões do que em outros, permitindo que as regiões críticas dos problemas recebam atenção especial, proporcionando assim máxima economia no processo de obtenção da solução desejada. A Fig. 8 apresenta um exemplo de Refinamento-h. Esses tipos de refinamento podem ser combinados. Segundo NOVOTNY & FANCELLO (1998), uma combinação bastante conhecida e eficiente no controle de erro para uma vasta classe de problemas é o refinamento h-p, que consiste na combinação das estratégias h e p. O inconveniente do método h-p é a necessidade de estruturas de dados complexas para a realização das operações.

Figura 8 - Análise no COMSOL com refinamento h-adaptativo.

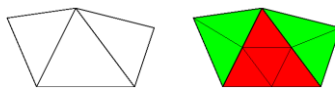


Fonte: COMSOL (2017).

O método de Refinamento-h, alvo da proposta atual, pode ser aplicado de várias maneiras. TAYLOR *et al* (2005) apresenta os dois métodos mais comuns :

- 1) Subdivisão de elemento: aplica o refinamento em locais com maiores erros. Os elementos existentes são divididos em menores, mantendo intacta as condições de contorno dos elementos originais. Pontos flutuantes, *hanging nodes*, são gerados no meio dos lados dos nós adjacentes ao elemento que foi refinado, sendo então necessário prover restrições nesses pontos flutuantes além de exigir uma maior habilidade de controle, tornando o cálculo mais complexo (RIVARA, 1984). Um método bastante conhecido, é o Red-Green (BANK & SHERMAN, 1981), que consiste em adicionar nós nas arestas dos elementos, gerando quatro novos triângulos geometricamente similares ao original, Fig. 9. Isso produz pontos extras ao longo das arestas dos triângulos vizinhos. Esses triângulos, que servem de transição entre a região que sofreu refinamento e o resto da estrutura, são divididos de modo que tornam-se irregulares e de baixa qualidade, sendo chamados de triângulos verdes.

Figura 9- Triângulo vermelho refinado com triângulos vizinhos de transição.



Fonte: JOHANN CERVENKA (1996).

- 2) Regeneração da malha: um novo tamanho de elemento é determinado em todos os domínios e uma malha totalmente nova é gerada. No geral, é o método mais eficiente e apresenta resultados superiores aos demais. No entanto, pode ser custoso especialmente para elementos tridimensionais e apresenta problema de transferência de dados de uma malha para outra. Para a maioria dos problemas práticos de engenharia, particularmente aqueles em que a forma do elemento é severamente distorcida durante a análise, a regeneração adaptativa de malha torna-se uma escolha natural, Fig. 10.

Figura 10- Comparação entre as malhas: Original, Subdivisão de elemento e Regeneração, respectivamente.



Fonte: ZIENKIEWICZ & TAYLOR (2000).

No desenvolvimento deste trabalho foi adotado o método de regeneração da malha. É determinado um tamanho ideal para os elementos da malha com base no critério da equidistribuição do erro elementar (ZIENKIEWICZ & TAYLOR, 2000), os elementos da malha são, então, subdivididos em elementos menores de acordo com o grau de refinamento determinado e uma nova malha é gerada. O estimador de erro empregado e as estratégias de refinamento utilizadas serão apresentados nas próximas seções deste trabalho.

2.4 Estimativa de erro no Método dos Elementos Finitos

Essa seção tem como finalidade central a exposição da formulação geral na qual está fundamentado o corrente trabalho. Todo o equacionamento dos estimadores de erro baseado na norma em energia será voltado para estimativas de erro levando em consideração o problema de elasticidade plana, o qual, posteriormente, é aplicado adequadamente a cada problema em análise.

2.4.1 Estimador de erro baseado na norma em energia

As estimativas de erro podem ser *a priori* ou *a posteriori*. De acordo com LINS (2011), estimativa de erro *a priori* são aquelas baseadas em conhecimento prévio das características da solução analítica do problema, no entanto, não fornecem uma quantificação do erro associado a uma resposta aproximada. Já as estimativas *a posteriori* baseiam-se na própria solução numérica para geração de melhores aproximações dos campos de temperatura, deslocamentos, tensões ou deformações, a depender o problema em questão. Esses valores são chamados de solução recuperada e são utilizados para guiar o processo adaptativo.

Segundo ZIENKIEWICZ & TAYLOR (2000), o erro de aproximação (e) de uma análise numérica em Elementos Finitos depende da malha adotada e pode ser definido como a diferença entre a solução exata, u , e a solução aproximada, $\hat{u}$, obtida pelo MEF. Aplicando essa definição na função deslocamento, o erro de aproximação é expresso por:

$$e = u - \hat{u} \quad (52)$$

A especificação do erro local nem sempre é adequada e eventualmente equivocada. Por exemplo, os erros em tensões medidos sobre uma carga pontual serão localmente infinitos. Com

o objetivo de obter um valor escalar que meça a função do erro em um sentido global para a corrente malha, adota-se uma medida em forma de norma. Neste trabalho, foi utilizado a norma do erro em energia, introduzido por ZIENKIEWICZ & ZHU (1987). Para problemas elásticos, pode ser obtida por:

$$\|e\| = \left[\int_{\Omega} (Se)^T D Se d\Omega \right]^{1/2} \quad (53)$$

Onde $\|e\|$ é o erro medido em energia, S é um operador diferencial linear, sendo $\{Se\}$ o gradiente da solução. Neste caso, S relaciona os deslocamentos com as deformações:

$$\varepsilon = Su \quad (54)$$

$$\hat{\varepsilon} = S\hat{u} \quad (55)$$

Sendo D a matriz de propriedades do material, as tensões podem ser obtidas por:

$$\sigma = D\varepsilon \quad (56)$$

$$\hat{\sigma} = D\hat{\varepsilon} \quad (57)$$

De maneira alternativa, pode-se escrever a norma do erro em energia das seguintes formas (Silva, 2015):

$$\|e\| = \left[\int_{\Omega} (\varepsilon - \hat{\varepsilon})^T D (\varepsilon - \hat{\varepsilon}) d\Omega \right]^{1/2} \quad (58)$$

$$\|e\| = \left[\int_{\Omega} (\varepsilon - \hat{\varepsilon})^T (\sigma - \hat{\sigma}) d\Omega \right]^{1/2} \quad (59)$$

$$\|e\| = \left[\int_{\Omega} (\sigma - \hat{\sigma})^T D^{-1} (\sigma - \hat{\sigma}) d\Omega \right]^{1/2} \quad (60)$$

Uma maneira de quantificar o erro medido na norma em energia é o erro relativo percentual em energia, η , descrito pela razão entre o erro na norma em energia e a energia total acumulada no domínio, $\|U\|$, dado por:

$$\eta = \frac{\|e\|}{\|U\|} 100\% \quad (61)$$

Onde

$$\|U\| = \left(\int_{\Omega} \varepsilon^T D \varepsilon d\Omega \right)^{1/2} \quad (62)$$

Porém, como a solução exata u não é conhecida, busca-se empregar uma solução com uma taxa de convergência superior a solução gerada diretamente pelo MEF, ε^* ou σ^* , denominadas solução melhorada. Dessa forma, a norma do erro em energia, obtida em relação à solução melhorada, pode ser escrita como (SILVA, 2015):

$$\|e^*\| = \left[\int_{\Omega} (\sigma^* - \hat{\sigma})^T D^{-1} (\sigma^* - \hat{\sigma}) d\Omega \right]^{1/2} \quad (63)$$

Essa técnica de substituição da solução exata por uma solução melhorada, obtida por recuperação, é utilizada pelos estimadores de erro baseados em recuperação. Este tipo de estimador tem sido empregado por diversos autores, como OÑATE & BUGEDA (1993) e CASTELLAZZI *et al.* (2010).

Supondo que a solução analítica não é conhecida, utiliza-se para o cálculo de $\|U\|$ o conceito de que o erro é ortogonal à solução aproximada (CIARLERT, 2002). Neste caso:

$$\|U\|^2 = \|\hat{U}\|^2 + \|e^*\|^2 \quad (64)$$

A qualidade dos estimadores de erro é medida pelo índice de efetividade, θ , descrito pela razão entre o valor do erro estimado pela Eq. (63) e o valor do erro verdadeiro estimado pela Eq. (53). Em casos onde não se conhece a solução analítica do problema, esta é substituída

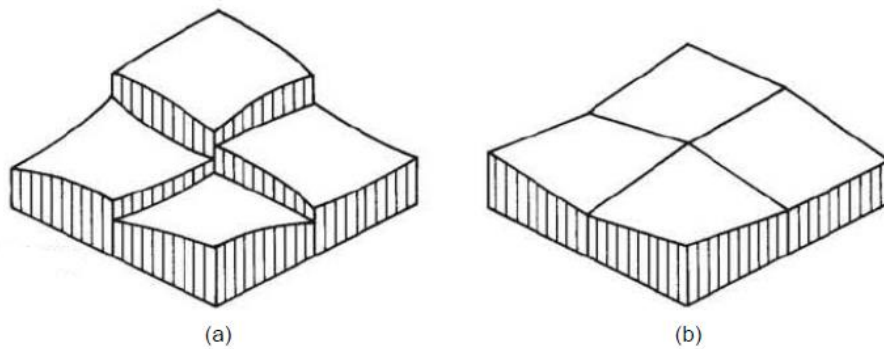
por uma solução obtida de uma malha bem refinada. Assim, caso a solução recuperada convirja a uma taxa mais elevada que a solução de MEF, uma estimativa assintoticamente exata é alcançada. O que se espera é que o índice de efetividade tenda a unidade à medida que o erro estimado tende a zero. A qualidade do processo de recuperação adotado determina a qualidade do estimador somente se a solução recuperada convergir para a solução analítica a uma taxa superior à solução do MEF (ZIENKIEWICZ & TAYLOR, 2000).

2.4.2 Método da Média Nodal Simples (MNS)

Existem vários estimadores de erro baseados em recuperação: MNS, ZZ, SPR, REP, entre outros. Dentre esses métodos de recuperação do gradiente, foi adotado o MMS neste trabalho, por apresentar reduzido custo computacional e por sua facilidade de implementação. Este consiste, inicialmente, no cálculo do gradiente da solução, por exemplo, o campo de tensões (em problemas de elasticidade) e do fluxo de calor (em problemas de temperatura). Sabendo que para um nó compartilhado por um total de nel elementos existem, via de regra, nel valores de $\hat{\sigma}$, pode-se estimar o vetor de médias nodais das tensões recuperadas, σ_m^* , para o nó como (COOK *et al.*, 2002):

$$\sigma_m^* = \frac{1}{nel} \sum_{j=1}^{nel} (\hat{\sigma}_i)_j \quad (65)$$

Figura 11- Comparação entre o gradiente obtido via MEF (a) e o gradiente suavizado (b).



Fonte: Adaptado de HINTON & CAMPBELL (1974).

Onde $(\hat{\sigma}_i)_j$ é o vetor de tensões obtidos pelo MEF para i-ésimo nó do j-ésimo elemento do conjunto. Um campo suavizado e contínuo entre os elementos pode ser construído através dos fluxos de calor médios nodais utilizando as mesmas funções de interpolação N . Desta forma, a parcela de um campo de tensões suavizadas sobre um elemento pode ser escrita como (SILVA, 2015):

$$\sigma^* = N\sigma_m^* \quad (66)$$

Onde σ^* é o vetor das tensões suavizadas no elemento. No entanto, para este trabalho, o campo de tensões suavizadas foi calculado como sendo a média dos valores das tensões recuperadas nos nós, devido à correspondência numérica desta operação.

2.5 Definição dos critérios de convergência e dos parâmetros do processo adaptativo

Segundo ZIENKIEWICZ & ZHU (1987) para uma solução ser considerada convergente e aceitável, ela deve satisfazer dois critérios: um em nível global e outro em nível local. Para a convergência global, é necessário que o erro global na norma em energia para a malha corrente k , $\|e_k\|$, não ultrapasse um percentual de energia de deformação total, $\|U_k\|$, ou seja:

$$\eta \leq \bar{\eta} \quad (67)$$

Onde η é dado pela Eq. (61) e $\bar{\eta}$ é o valor do erro relativo percentual admissível global especificado pelo analista.

Para o critério local elementar, adotou-se a equidistribuição do erro elementar que baseia-se no conceito de equidistribuição do erro, ou seja, cada elemento deve ter o mesmo valor do erro (ZIENKIEWICZ & ZHU, 1987). Sendo n_e o número de elementos da malha, pode-se calcular o erro em todo o domínio por:

$$\|e^*\|^2 = \sum_{e=1}^{n_e} \|e_e^*\|^2 \quad (68)$$

Um erro médio, e_m , pode ser definido pela Eq. (69).

$$n_e \cdot e_m^2 = \sum_{e=1}^{n_e} \|e_e^*\|^2 \quad (69)$$

Ao relacionar este erro médio com a percentagem de erro admissível, $\bar{\eta}$, pode-se definir o erro médio almejado, e_m^* , para cada elemento na próxima malha como:

$$e_m^* = \bar{\eta} \left[\frac{\|U\|^2}{n_e} \right]^{1/2} \quad (70)$$

Onde $\|U\|^2$ é dado pela Eq. (64). Para os elementos lineares, fazendo-se uso da teoria de erros *a priori*, sem a presença de singularidades na solução, podemos admitir que o erro associado a um elemento é proporcional à dimensão característica deste. Assim, podemos relacionar erro e dimensão característica através da expressão (ZIENKIEWICZ & TAYLOR, 2000; ARAÚJO, 2004):

$$\|e_e^*\|^k = c \cdot d_e^k \quad (71)$$

Onde k indica a malha corrente, c representa uma constante de proporcionalidade e d_e é a dimensão característica do elemento.

Como pretende-se obter o erro médio, e_m^* , calculado pela Eq. (70) para cada elemento da solução da malha seguinte ($k+1$), tem-se (ARAÚJO, 2004):

$$e_m^* = \|e_e^*\|^{k+1} = c \cdot d_e^{k+1} \quad (72)$$

Relacionando as Eq. (71) e (72), obtém-se a Eq. (73), que determina a dimensão necessária para que se obtenha a qualidade de solução especificada através de $\bar{\eta}$, identificando tanto as regiões que deveriam ser refinadas quanto as que deveriam ser empobrecidas. No entanto, o código implementado não realiza o desrefinamento da malha, portanto, quando indicar desrefino o elemento permanecerá com o tamanho inalterado.

$$d_e^{k+1} = \frac{d_e^k \cdot e_m^*}{\|e_e^*\|} \quad (73)$$

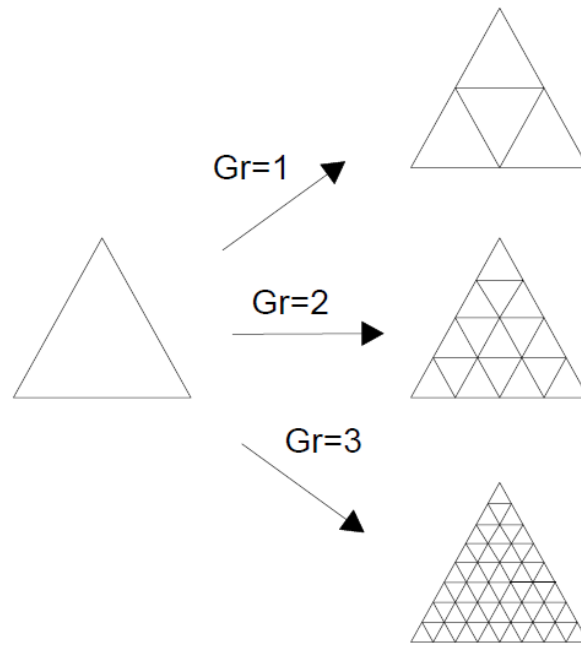
Os resultados da Eq. (73) conduzem a um campo de dimensões descontínuas ao longo do domínio. Desta forma, algum tipo de média ou procedimento de suavização deve ser adotado para se ter um campo contínuo de distribuição de densidade de malha desejada. Além disso, como o erro médio desejado, e_m^* , é calculado com o uso dos gradientes suavizados e não com os gradientes exatos e nem o processo de obtenção da nova malha é algo exato, o erro admissível, $\bar{\eta}$, pode não ser encontrado para a solução aproximada sobre a malha adaptada. Sendo, então, necessário uma nova análise de erros após a simulação numérica sobre a nova malha, com o intuito de verificar se o erro almejado foi atingido ou não. Não sendo alcançado, deve-se proceder uma nova adaptação até que $\bar{\eta}$ seja atingido (ARAÚJO, 2004).

2.5.1 Grau de refinamento

O grau de refinamento é determinado pela comparação entre as dimensões d_e^k e d_e^{k+1} , isto é, a dimensão característica atual do elemento e a dimensão desejada que foi indicada pela análise de erros. Esse parâmetro determina quais elementos serão refinados e o nível de refinamento de cada um deles, conduzindo, portanto, o processo de refinamento adaptativo.

A dimensão característica adotada neste trabalho foi $d = \sqrt{2 \cdot A}$, sendo A a área do elemento triangular. A Figura 12 indica a relação entre o grau de refinamento e as dimensões do elemento após a adaptação da malha.

Figura 12- Determinação do grau de refinamento.



Fonte: A Autora.

Observe que, ao dividir a área do elemento por 4, a dimensão característica passa a ser metade da dimensão original. Portanto, a ordem da dimensão obtida é de 2^{Gr} menor do que a dimensão original, sendo Gr o grau de refinamento desejado. Desse modo, a expressão que relaciona o nível de refinamento com as dimensões da nova malha é (ARAÚJO, 2004):

$$d_e^{k+1} = \frac{d_e^k}{2^{Gr}} \quad (74)$$

Onde

$$Gr = \frac{\log\left(\frac{d_e^k}{d_e^{k+1}}\right)}{\log(2)} \quad (75)$$

No entanto, como a solução da Eq. (75) é um número real e o grau de refinamento deve ser um número inteiro, será adotado o arredondamento deste valor. Se o grau de refinamento for negativo, indicando desrefino, será atribuído o valor 0, indicando que o elemento não deve

ser refinado, pois o código implementado não realiza desrefinamento. Para casos em que Gr é positivo, se a parte decimal do número for inferior a 0,25 toma-se o valor arredondado eliminando a parte decimal. Entretanto, se o valor for maior que 0,25, será feito o truncamento, porém o valor de Gr será o valor obtido adicionado de uma unidade. Esses critérios adotados são semelhantes ao proposto por ARAÚJO (2004) e são de caráter conservador, pois conduzem a malhas com maior número de elementos.

3 METODOLOGIA ADOTADA E PARTICULARIDADES DOS CÓDIGOS

Serão descritas as estratégias implementadas para o refinamento h-adaptativo das malhas, bem como as particularidades e a estrutura dos programas desenvolvidos. Foi utilizado um computador pessoal com processador Intel Core i7-6500U com memória RAM de 8GB e HD de 1TB.

3.1 Determinação do nível de refinamento para cada elemento

Como visto na seção 2.5, é possível estimar a distribuição das dimensões dos elementos necessárias para atingir uma tolerância pré-estabelecida, a partir da estimativa global dos erros, do indicador dos erros de cada elemento e do critério de equidistribuição dos erros. O grau de refinamento de cada elemento é, então, definido pela relação entre a dimensão do elemento da malha corrente e a dimensão estimada. No entanto, uma série de medidas são tomadas para evitar o refinamento excessivo e promover uma melhor regularização da malha.

3.1.1 Desnível unitário interelementos

Entende-se por desnível unitário interelementos a diferença entre os valores do nível de refinamento particular de cada elemento indicado pela análise de erros e os valores dos níveis de seus vizinhos. Desta forma, o processo de refinamento da malha foi elaborado de maneira que os elementos vizinhos não possuam uma diferença de nível de refinamento superior à unidade (ARAÚJO, 2004).

Apenas um *loop* em todos os elementos não é suficiente para garantir o desnível unitário global. Com a finalidade de evitar perda da eficiência pelo número de *loops* em todos os elementos da malha, apenas os vizinhos dos elementos que tiveram seu grau de refinamento alterado são armazenados e passam por uma nova verificação, na qual é verificado o desnível unitário entre os vizinhos e os vizinhos deles. Portanto, o desnível unitário contém duas etapas: 1 – Verificação do desnível unitário de cada elemento da malha com relação aos seus vizinhos e armazenamento dos elementos vizinhos àqueles que sofreram alteração do grau de refinamento; 2 – Verificação do desnível unitário entre os vizinhos dos elementos que tiveram seu grau de refinamento alterado e os vizinhos destes vizinhos.

É arbitrado ao usuário definir o grau máximo de refinamento por etapa. A limitação do grau de refinamento é importante porque em problemas com singularidades, os quais são encontrados com frequência nas mais diversas aplicações de engenharia, a análise de erros indica para os elementos da região de singularidade dimensões extremamente pequenas a fim de minimizar o erro naquela região. Utilizando o critério de desnível unitário, haverá propagação do nível de refinamento do elemento da singularidade por toda a malha e uma malha demasiadamente densa será gerada. Isto implicaria em uma ótima aproximação, porém com grande perda de eficiência computacional. Segundo ARAÚJO (2004), processos sucessivos de adaptação com refinamento limitado levam a uma solução tão boa quanto esta e com muito mais eficiência.

Por exemplo, ao limitar o grau de refinamento em 2, a propagação do nível se dá apenas para a camada de vizinhos de um elemento central (que possui a singularidade) e, numa próxima malha, novamente apenas os elementos ao redor da singularidade recebem grau dois de refinamento. Deve-se ainda observar que mesmo limitando-se o grau de refinamento, em cada análise, consegue-se obter uma excelente discretização ao redor da singularidade, garantindo uma boa solução. Portanto, ao limitar o grau de refinamento, limita-se a propagação do nível de refinamento. Além disso, como visto nas seções anteriores, a análise de erros também se baseia em aproximações, de maneira que a tolerância do erro desejada pode não ser atingida com a primeira malha adaptada, sendo necessária, por vezes, a realização de mais de uma etapa de adaptação da discretização do domínio. Portanto, a utilização de malhas desnecessariamente densas implicaria, mais uma vez, em perda de eficiência computacional (ARAÚJO, 2004).

Diante disto, durante a implementação do programa foi estabelecido que o grau máximo de refinamento permitido por etapa é 3, ficando a cargo do usuário a definição do grau de refinamento máximo para a análise em questão, desde que $1 \leq Gr_{\max} \leq 3$.

3.1.2 Regularização da malha

O critério de regularização da malha adotado tem por objetivo a formação de uma malha com melhor qualidade. Ele consiste em corrigir o nível de refinamento de um elemento quando pelo menos dois dos seus vizinhos possuírem grau de refinamento elevado em uma unidade em relação a ele (ARAÚJO, 2004).

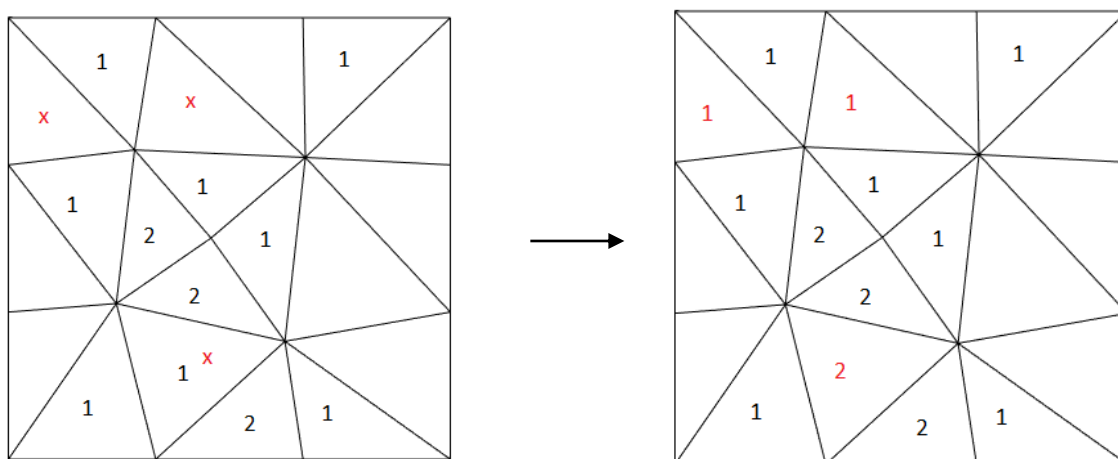
Assim como o desnível unitário, o processo de regularização da malha possui duas etapas. A Fig. 13 mostra a configuração da malha após a primeira etapa de manutenção do

desnível unitário. Verificou-se que os elementos marcados precisavam ter seus graus corrigidos. Após a correção, os vizinhos destes elementos que sofreram alteração do grau de refinamento (destacados em vermelho) são verificados. Constatou-se que um dos seus vizinhos, elemento marcado na Fig. 14, que não precisava de correção na primeira etapa, precisou ter seu nível corrigido na segunda etapa devido à alteração do grau de refinamento do seu vizinho.

Em suma, os critérios de desnível unitário e de regularização da malha são realizados em duas etapas. A primeira consiste na verificação do desnível unitário e armazenamento dos elementos vizinhos daqueles que sofreram alteração do grau de refinamento, seguido pela verificação do critério de regularização da malha e armazenamento dos elementos vizinhos àqueles que sofreram alteração do grau de refinamento. A segunda etapa consiste na verificação do desnível unitário e da regularização da malha entre os vizinhos dos elementos que tiveram seu grau de refinamento alterado e os vizinhos destes vizinhos.

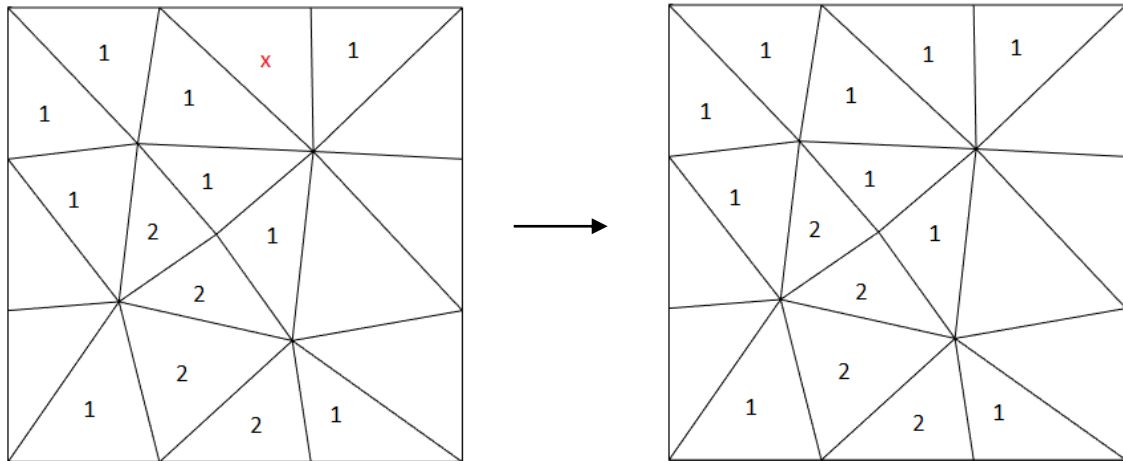
Ainda assim, pode haver algum elemento que não satisfaça o critério de regularização da malha, pois após a segunda etapa seria necessário a verificação dos elementos vizinhos aos elementos que foram corrigidos e assim sucessivamente até que nenhum elemento sofresse alteração do grau de refinamento. No entanto, isso implicaria num número muito maior *loops* e no aumento do custo computacional. Além disso, as etapas implementadas mostraram-se suficientes para atingir a finalidade de melhoria da qualidade da malha.

Figura 13- Primeira etapa de regularização da malha.



Fonte: A autora.

Figura 14- Segunda etapa de regularização da malha.



Fonte: A Autora.

3.1.3 Dimensão mínima dos elementos da malha

Sabe-se que para problemas com singularidade é possível que haja refinamentos excessivos em uma região do domínio com pouca alteração do erro global da solução, uma medida adotada com respeito à presença de singularidades, foi a limitação do tamanho mínimo dos elementos da malha. Ao atingir uma dimensão menor que $d_{\min}$, o elemento é mantido com a dimensão corrente. Isso evita a geração de elementos com dimensões extremamente pequenas e que conduziriam a uma malha com muitos elementos na busca de reduzir o erro total à tolerância desejada. Se a tolerância for muito pequena e a singularidade muito forte o procedimento pode não convergir, requerendo sempre mais e mais refinamento na região singular (SAMPAIO ET AL, 1993; LYRA, 1988).

Além disso, a presença de singularidades pode, através do efeito de “poluição do erro”, levar à não determinação adequada da qualidade da solução fora da região singular (ARAÚJO, 2004). Ao limitar o tamanho mínimo dos elementos, permite-se que outras singularidades sejam encontradas e que outras regiões do domínio sejam mais contempladas.

3.2 Estrutura dos programas

Foram desenvolvidos dois programas no MATLAB (2015), nomeados *PoissonG* e *ELT2D*, para o refinamento adaptativo de soluções em elementos finitos de problemas descritos pela equação de Poisson e problemas de elasticidade plana, respectivamente. Os programas funcionam de maneira independente e devem ser utilizados de acordo com a necessidade do

usuário, isto é, a depender do problema a ser analisado. Os procedimentos de refinamento adaptativo apresentados anteriormente são comuns aos dois programas. A seguir serão apresentadas as particularidades de cada um deles.

3.2.1 PoissonG

A primeira etapa de desenvolvimento do código foi a implementação e validação do solucionador MEF, seção 2.1.1. Só então iniciou-se a implementação do refinamento adaptativo, acompanhado da definição do estimador de erro *a posteriori*, do método de recuperação de gradiente e dos critérios discretização e regularização da malha. Descritos nas seções 2.3, 2.4 e 2.5.

A Figura 15 apresenta um fluxograma dos procedimentos do programa. Inicialmente o usuário deve fornecer os dados de entrada: dados do problema, condições de contorno (podem ser descritas por uma função ou por um vetor com os valores prescritos dos nós do contorno), erro percentual admissível, $\bar{\eta}$, global e local elementar (que podem ser iguais ou distintos), dimensão característica mínima permitida para os elementos da malha, $d_{\min}$, e o grau máximo de refinamento permitido por etapa, $Gr_{\max}$, tal que $1 \leq Gr_{\max} \leq 3$. O usuário tem a opção de gerar a malha com um gerador de malha do próprio MATLAB, inserido no código, ou importar a malha do GiD.

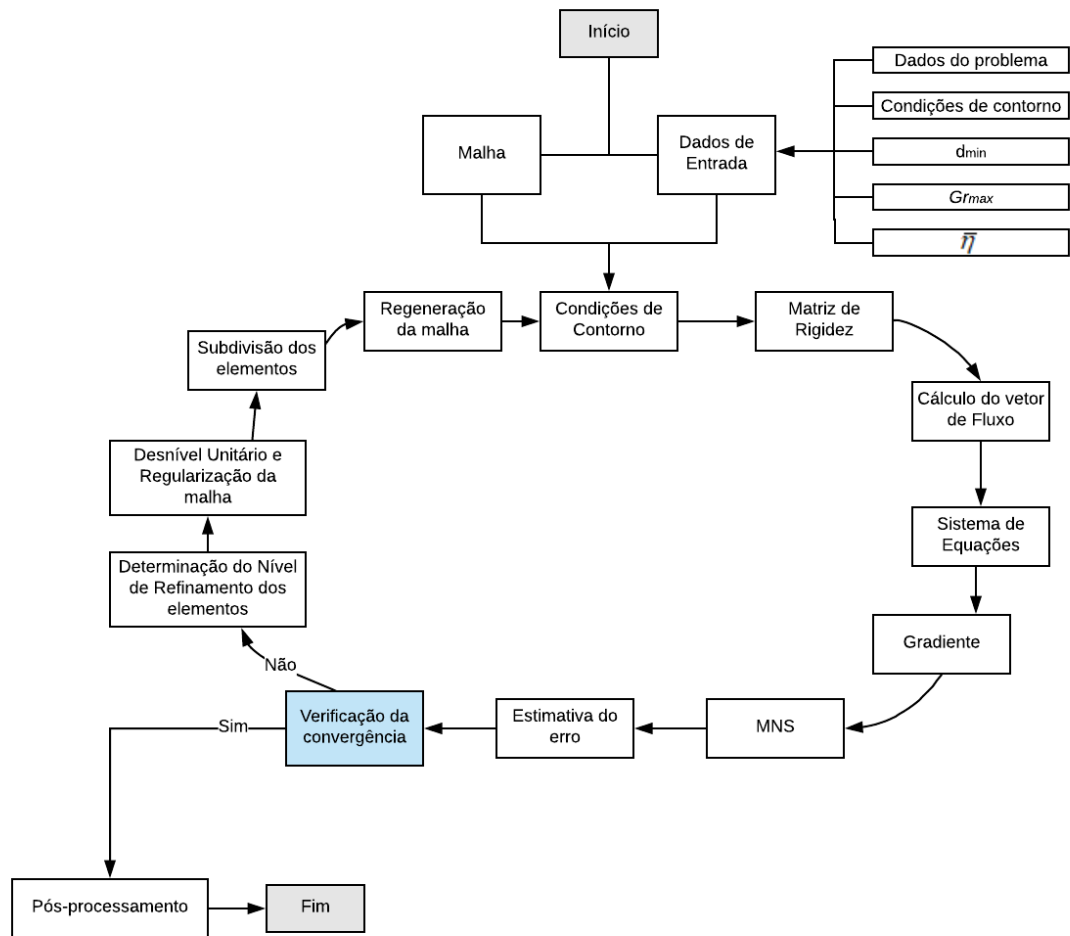
Se optar por gerar a malha no próprio MATLAB, o usuário deverá informar a dimensão máxima permitida para os lados dos elementos da malha, $h_{\max}$. A ideia é que o código possa ser autônomo, não necessitando de um segundo programa para geração da malha. No entanto, a função utilizada na geração de malha, nativa do MATLAB, não permite o controle da qualidade dos elementos e, a depender do domínio e do $h_{\max}$ adotado, a malha gerada é bastante irregular. Sabendo que a qualidade dos elementos da malha inicial interfere na qualidade dos elementos filhos gerados no processo adaptativo e, portanto, na eficiência da obtenção da solução com a precisão desejada, em muitos casos torna-se mais viável utilizar uma malha inicial proveniente do programa gerador de malha, GiD. Neste caso, o usuário deve importar do GiD as coordenadas dos nós e a matriz de conectividade dos elementos.

Segue-se, então, o cálculo da matriz de rigidez dos elementos, Eq. (10), montagem da matriz de rigidez global, Eq. (14), cálculo do vetor de fluxos, Eq. (13), e montagem e resolução do sistema de equações, Eq. (16). Após a obtenção da solução pelo MEF, é realizado o cálculo dos gradientes e do campo suavizado pelo método MNS, utilizado no cálculo do erro na norma

em energia, conforme explicitado nas seções 2.4.1 e 2.4.2. É importante destacar que as formulações apresentadas nestas seções estão aplicadas aos problemas de elasticidade plana, mas podem estendidas para os problemas descritos pela equação de Poisson. Por exemplo, em um problema de condução de calor, o gradiente da solução é denominado fluxo de calor e é obtido pelo produto da matriz de condutividade térmica, Eq. (12), pela matriz de derivadas das funções de forma, Eq. (11), e pelo vetor de temperaturas nodais.

O critério de convergência do processo adaptativo é dado pela Eq. (67). Se o erro relativo percentual da malha corrente ultrapassar o erro relativo percentual admissível, $\bar{\eta}$ ela deverá ser refinada. Utilizando o critério de equidistribuição do erro elementar e relacionando-o com a percentagem de erro admissível local elementar, que também deve ser especificado pelo usuário, determina-se o erro médio almejado para os elementos, Eq. (70). Com isso, estima-se o tamanho que o elemento deve ter para atingir o erro médio almejado e o seu grau de refinamento é determinado pela relação entre a dimensão do elemento da malha corrente e a dimensão estimada.

Entretanto, devem ser tomadas medidas para evitar o refinamento excessivo e para promover uma melhor regularização da malha. Para isso, foram adotados os critérios de desnível unitário interelementos e de regularização da malha apresentados na seção 3.2. Após a verificação desses critérios para todos os elementos da malha, são realizadas as subdivisões dos elementos de acordo com o nível de refinamento estabelecido. Uma nova malha é gerada e repete-se o processo até que $\eta \leq \bar{\eta}$.

Figura 15- Fluxograma dos procedimentos do programa *PoissonG*.

Fonte: A Autora.

3.2.2 *ELT2D*

Os procedimentos de implementação e a estrutura de *ELT2D* são semelhantes a *PoissonG*. A diferença mais notável está presente nas funções relacionadas ao solucionador MEF do código. O elemento utilizado aqui é o CST, que possui dois graus de liberdade por nó.

Na etapa de pré-processamento, a malha inicial deverá ser gerada no GiD e o usuário deve importar as coordenadas dos nós e a matriz de conectividades dos elementos triangulares. O usuário deve, então, fornecer os dados do problema: características do material e forças externas aplicadas. Também deve informar se deseja considerar as forças de corpo na análise, se sim, deve fornecer as densidades de peso nas direções x e y , Eq. (46). As condições de contorno permitem apoios do primeiro gênero, segundo gênero ou engaste, ficando a cargo do usuário a especificação dos nós ou arestas que recebem tais apoios. O analista também deve

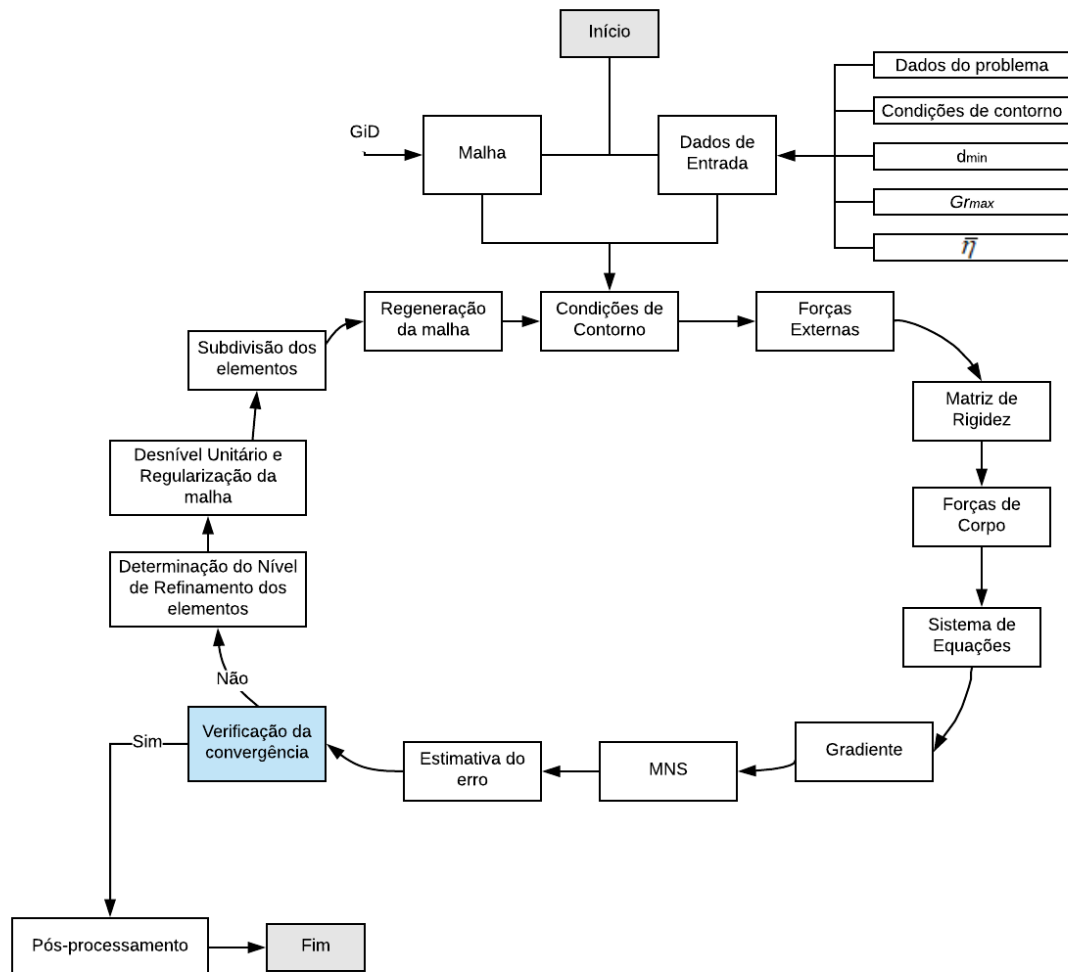
informar o $d_{\min}$, $Gr_{\max}$ e o erro percentual admissível, $\bar{\eta}$ global e local elementar (que podem ser iguais ou distintos).

Segue-se, então, o cálculo da matriz de rigidez dos elementos, Eq. (43), montagem da matriz de rigidez global, Eq. (44), cálculo das forças de corpo, Eq. (47), montagem do vetor de forças totais do sistema (forças externas mais forças de corpo), Eq. (50) e resolução do sistema de equações, Eq. (51).

Após a obtenção da solução pelo MEF, é realizado o cálculo dos gradientes e do campo suavizado pelo método MNS, utilizado no cálculo do erro na norma em energia, conforme explicitado nas seções 2.4.1 e 2.4.2. O critério de convergência do processo adaptativo é dado pela Eq. (67). Se o erro relativo percentual da malha corrente ultrapassar o erro relativo percentual admissível, $\bar{\eta}$, ela deverá ser refinada. Utilizando o critério de equidistribuição do erro elementar e relacionando-o com a percentagem de erro admissível local elementar, que também deve ser especificado pelo usuário, determina-se o erro médio almejado para os elementos, Eq. (70). Com isso, estima-se o tamanho que o elemento deve ter para atingir o erro médio almejado e o seu grau de refinamento é determinado pela relação entre a dimensão do elemento da malha corrente e a dimensão estimada.

Os critérios adotados em *ETL2D* para evitar o refinamento excessivo e promover uma melhor regularização da malha, desnível unitário interelementos e regularização da malha, foram apresentadas na seção 3.2 e são os mesmos aplicados a *PoissonG*. Após a verificação desses critérios para todos os elementos da malha, são realizadas as subdivisões dos elementos de acordo com o nível de refinamento estabelecido. Uma nova malha é gerada e repete-se o processo até que $\eta \leq \bar{\eta}$, isto é, até que o erro relativo percentual global da malha seja menor ou igual ao erro percentual admissível.

Figura 16- Fluxograma dos procedimentos do programa *ELT2D*.



Fonte: A Autora.

4 RESULTADOS E DISCUSSÕES

Os resultados foram divididos em problemas descritos pela Equação de Poisson e Elasticidade plana. Serão apresentados um total de cinco exemplos numéricos para evidenciar e discutir a eficiência do processo de refinamento h-adaptativo, utilizado com o intuito de obter soluções com baixo erro de discretização a um baixo custo computacional. Também é apresentado um comparativo entre o refinamento h-adaptativo e o refinamento uniforme.

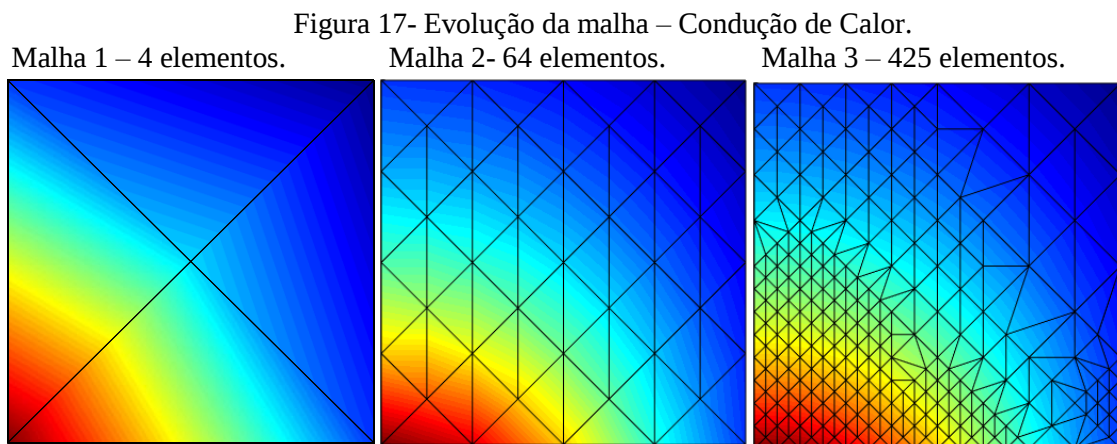
4.1 Problemas descritos pela Equação de Poisson

Serão apresentados alguns exemplos numéricos de problemas bidimensionais descritos pela equação de Poisson: condução de calor, flexão de placas (redução da equação diferencial de quarta ordem em duas equações de segunda ordem) e torção.

4.1.2 Condução de calor

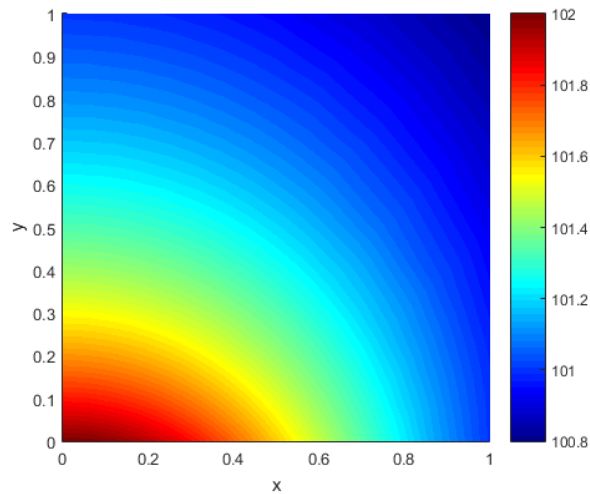
O domínio é um quadrado de lado 1m, $\Omega = [0,1]^2$. A solução analítica é dada pela Eq. (76), onde $a=0$ e $b=1$, a qual satisfaz $\nabla^2 T = 0$. Adotou-se $\bar{\eta} = 0.05$ para o erro percentual admissível global e local elementar. O grau máximo de refinamento por etapa adotado foi $Gr_{\max} = 2$. A malha inicial e as demais foram geradas no próprio MATLAB e são mostradas na Fig. 17 e a Fig. 18 apresenta o campo de temperatura obtido ao final do refinamento.

$$T(x, y) = 100 + \frac{2(1+y)}{(a+x)^2 + (b+y)^2} \quad (76)$$



Fonte: A Autora.

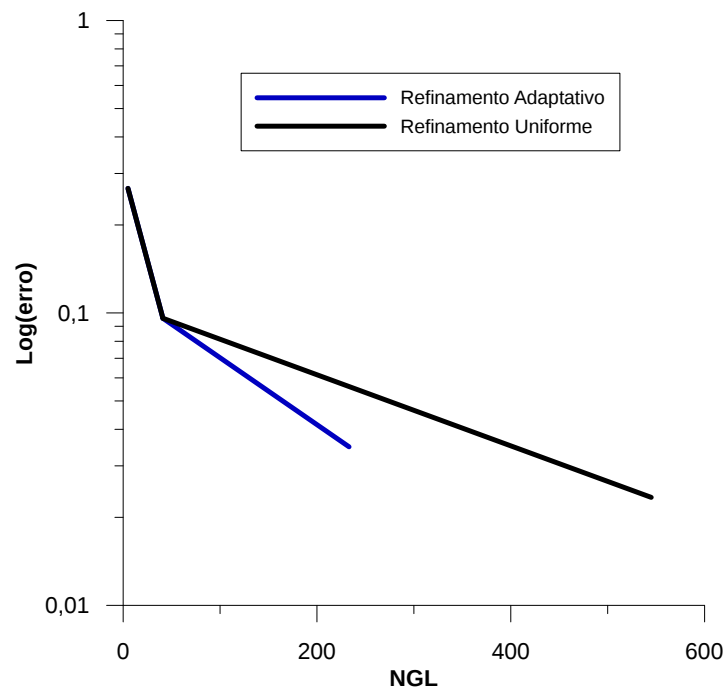
Figura 18- Campo de temperatura.



Fonte: A Autora.

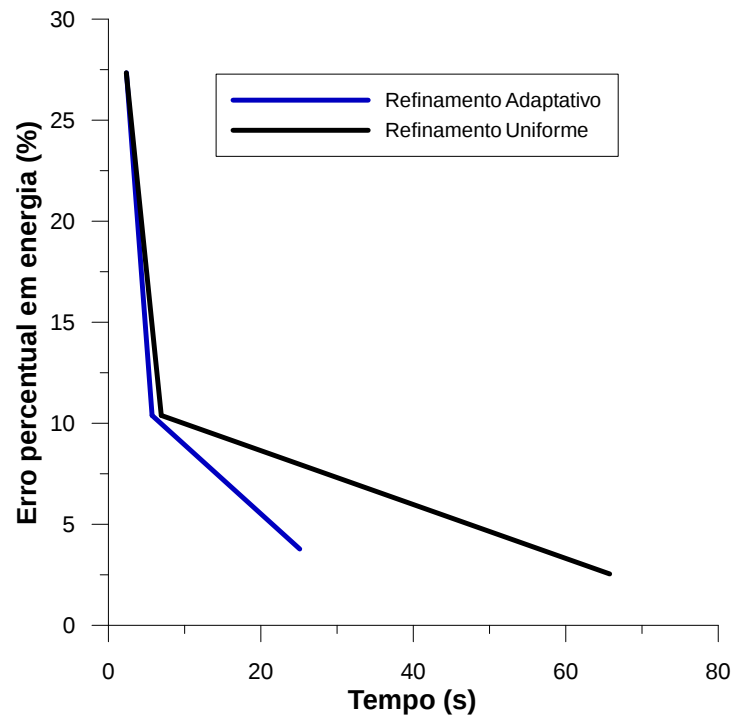
A Figura 19 apresenta um comparativo entre o refinamento h-adaptativo e o refinamento uniforme. Nota-se que o refino adaptativo é mais eficiente, pois alcança o erro admissível determinado para a análise com menos da metade do número de graus de liberdade do uniforme. Além disso, pela Fig. 20, observa-se que o tempo de processamento do adaptativo é cerca de metade do tempo do uniforme, evidenciando, mais uma vez a eficiência do refinamento adaptativo na obtenção da solução com a precisão exigida a um menor custo de processamento computacional.

Figura 19- Erro na norma em energia versus NGL.



Fonte: A Autora.

Figura 20- Análise do esforço computacional.

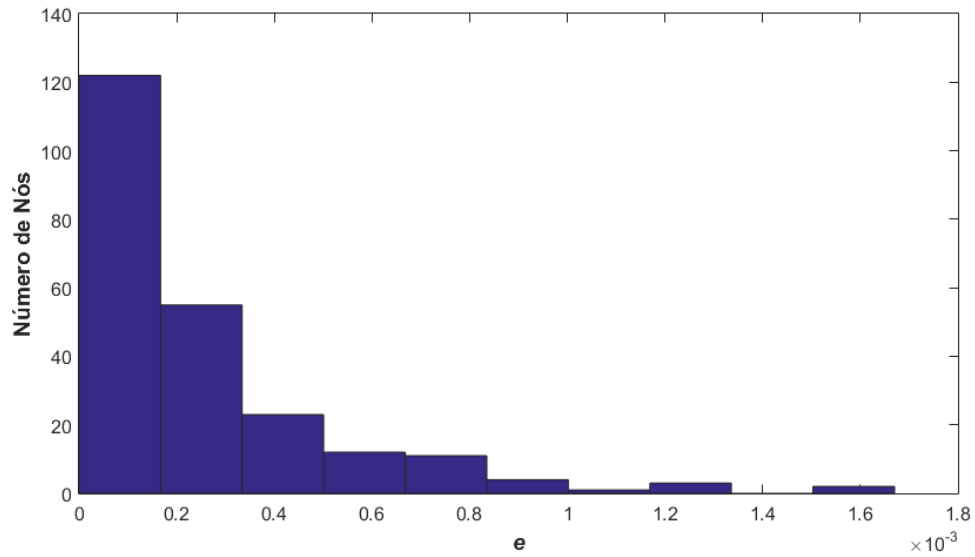


Fonte: A Autora.

A qualidade dos estimadores de erro é medida pelo índice de efetividade, θ , espera-se que o estimador seja assintoticamente exato, ou seja, que seu índice de efetividade tenda a unidade quando o erro estimado tender a zero. O índice de efetividade da malha final foi $\theta = 0,996$, indicando, portanto, que solução recuperada converge para a solução analítica a uma taxa superior à solução pelo MEF.

Os valores numéricos das temperaturas obtidas com a estratégia adaptativa foram comparados com o valor da solução exata, obtidos pela Eq. (76). Analisando o histograma da Fig. 21, que relaciona o número de nós com o valor do erro de aproximação, Eq. (52), percebe-se que os erros são da ordem de 10^{-3} e que a grande maioria dos nós apresentam erros muito próximos de zero.

Figura 21- Número de Nós versus Erro de aproximação (e) – Problema de condução de calor.



Fonte: A Autora.

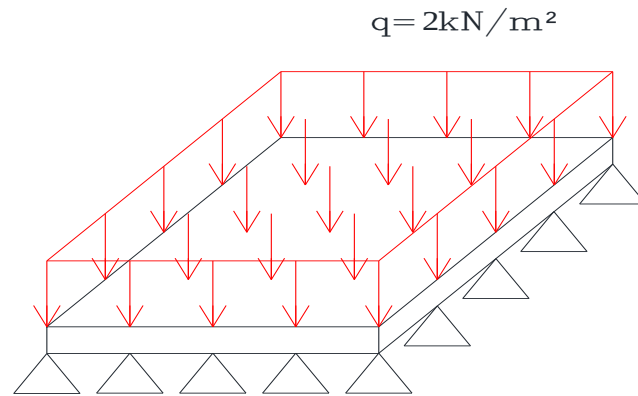
4.1.1 Flexão de placas

O problema de flexão em uma placa pode ser reduzido ao problema de flexão em uma membrana, Eq. (77) e (78), onde M são os momentos, w são os deslocamentos, q é a carga aplicada, $q=-2\text{kN/m}$, e D é a rigidez à flexão da placa (TIMOSHENKO, 1989). A placa da Fig. 22 está simplesmente apoiada em todos os lados, portanto, $M=0$ e $w=0$ ao longo do contorno. Possui 20 cm de espessura, $E= 200 \text{ MPa}$ e $\nu = 0,3$. O cálculo do erro na norma em energia foi obtido a partir do campo suavizado da solução da Eq. (77).

$$\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2} = -q \quad (77)$$

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = -\frac{M}{D} \quad (78)$$

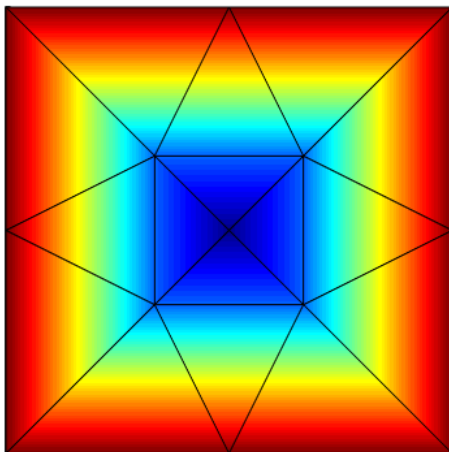
Figura 22- Apresentação do problema de flexão de placas.



Fonte: A Autora.

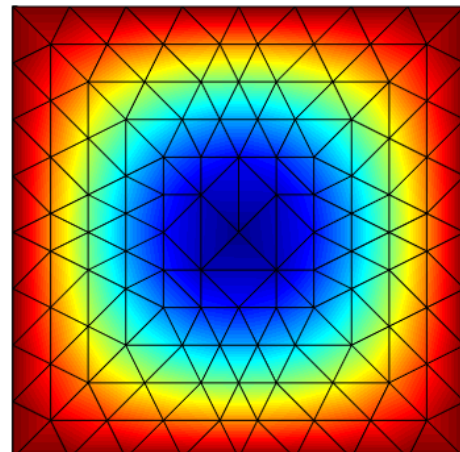
A malha inicial e as demais foram geradas no MATLAB. Optou-se por uma malha inicial pobre, com apenas 16 elementos, para que não houvesse refino inicial proveniente do usuário, ficando a cargo do código. Nesta análise, o valor do erro percentual em energia admitido foi 5%, tanto para o erro global quanto para o erro local elementar. O grau máximo de refinamento adotado foi $Gr_{max}=2$. A convergência ocorreu na segunda malha adaptada, as Fig. 23, 24 e 25 apresentam a sequência das malhas. A Tabela 1 apresenta os resultados obtidos para cada malha gerada ao longo do processo de refinamento efetuado e a Tabela 2 apresenta os resultados do refinamento uniforme, que foi realizado com o intuito de estabelecer um comparativo com o refinamento adaptativo e verificar sua eficiência.

Figura 23- Malha inicial -16 elementos.



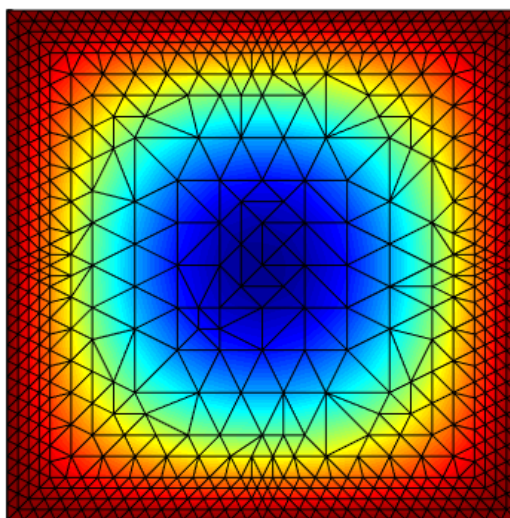
Fonte: A Autora.

Figura 24- 1ª malha adaptada – 216 elementos.



Fonte: A Autora.

Figura 25- 2ª malha adaptada – 1386 elementos.



Fonte: A Autora.

Tabela 1 - Resultados do refinamento adaptativo da placa sob flexão.

Malha	Número de elementos	NGL	$\ e\$	θ	η (%)	Desl. y (máx) (10^{-5}m)
1	16	13	0,156	1,011	41,56	-3,774
2	216	125	0,037	0,995	10,00	-5,400
3	1386	758	0,013	0,968	3,44	-5,496

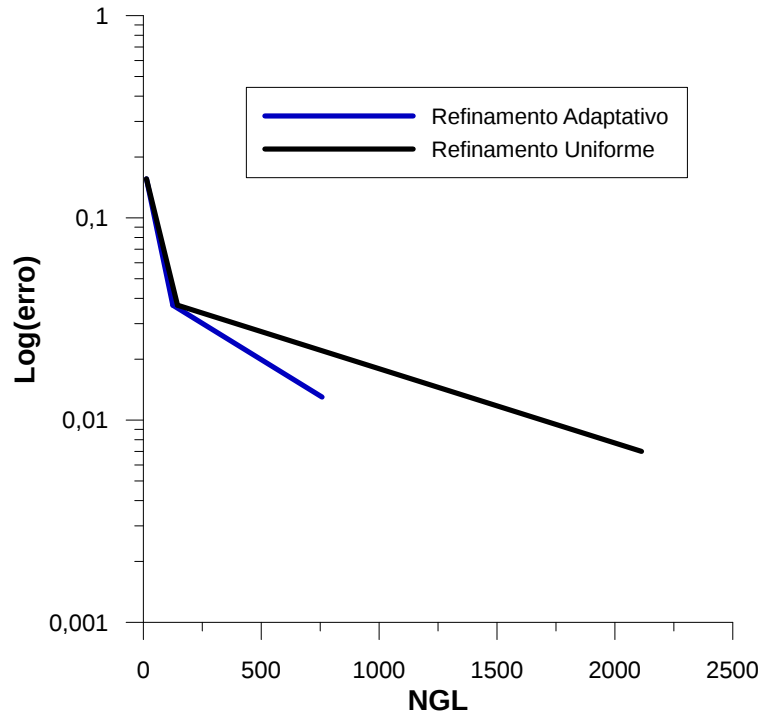
Tabela 2 - Resultados do refinamento uniforme da placa sob flexão.

Malha	Número de elementos	NGL	$\ e\$	η (%)	Desl. y (máx) (10^{-5}m)
1	16	13	0,156	41,56	-3,774
2	256	145	0,037	9,90	-5,413
3	4096	2113	0,007	1,97	-5,537

Pelas tabelas acima, observa-se que o erro percentual admissível e o deslocamento máximo possuem uma boa convergência para um número muito inferior de elementos em comparação com o refino uniforme. O índice de efetividade, θ , calculado pela comparação com a solução obtida por uma malha bem refinada, tende a 1, indicando que a estimativa do erro é assintoticamente exata. O gráfico do Erro Relativo Percentual em Energia (η_{adm}) versus

Número de Graus de Liberdade (NGL), Fig. 26, evidencia, mais uma vez, a eficiência do refinamento adaptativo em alcançar bons resultados a um menor custo computacional, alcançando o erro admissível determinado para a análise com menos da metade do número de graus de liberdade do uniforme.

Figura 26- Erro na norma em energia versus NGL.



Fonte: A Autora.

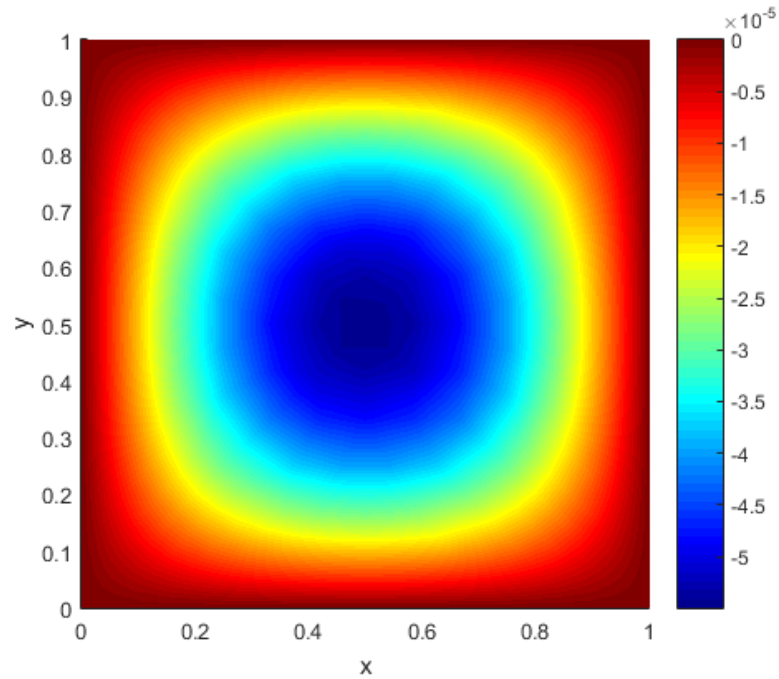
Os valores numéricos dos deslocamentos verticais obtidos com a estratégia adaptativa foram comparados o valor da solução analítica, Eq.(79) (TIMOSHENKO, 1989).

$$w = \frac{16q}{\pi^6 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}}{mn \left(\frac{m^2}{a^2} + \frac{n^2}{b^2} \right)^2} \quad (79)$$

Onde $m=1,3,5,\dots$ e $n=1,3,5,\dots$. Como esta série converge rapidamente e aproximações satisfatórias são obtidas avaliando-se apenas os primeiros termos, a série foi truncada no quarto termo. O valor do deslocamento máximo obtido pela solução analítica foi $-5.544 \cdot 10^{-5}$ e o obtido pela solução numérica foi $-5.496 \cdot 10^{-5}$. As Figuras 27 e 28 apresentam o campo de deslocamentos obtido pela solução numérica com a malha final adaptada e pela solução

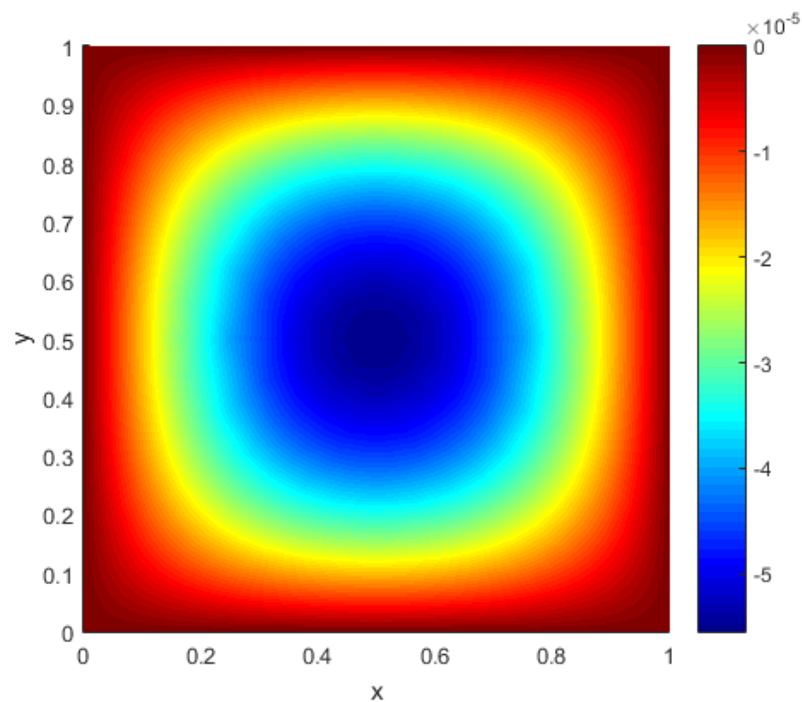
analítica, respectivamente. Pelo o histograma do erro de aproximação, Fig. 29, constata-se que os erros de aproximação, Eq. (52), são da ordem de 10^{-7} e que a grande maioria dos nós apresentam erros muito próximos de zero.

Figura 27- Deslocamentos verticais – solução numérica.



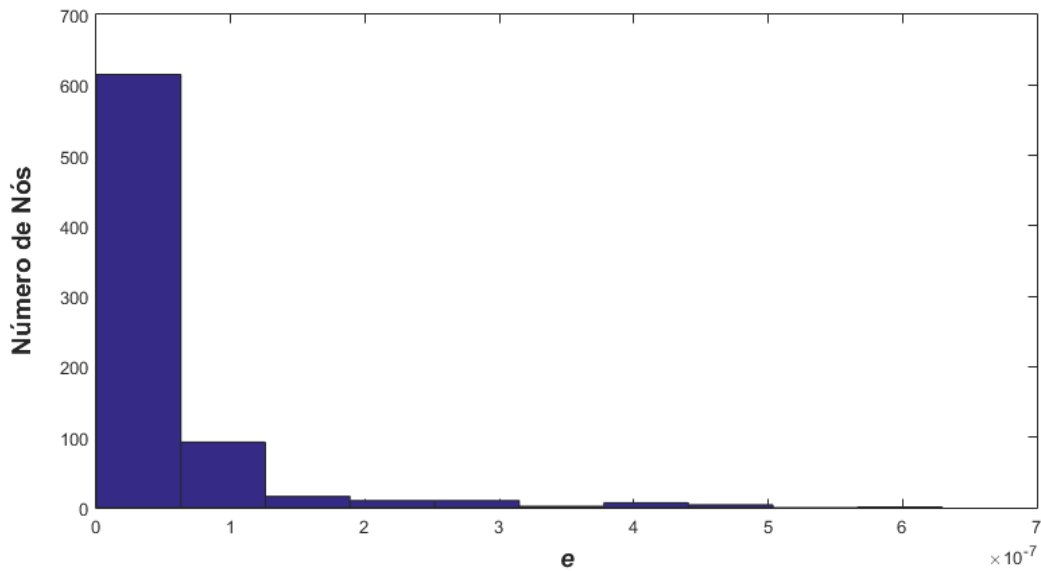
Fonte: A Autora.

Figura 28- Deslocamentos verticais – solução analítica.



Fonte: A Autora.

Figura 29- Número de Nós versus Erro de aproximação (e) – Deslocamentos verticais.



Fonte: A Autora.

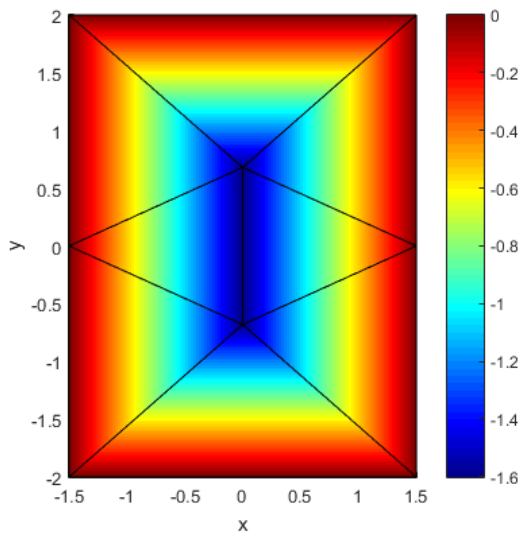
4.1.3 Torção

A rigidez torcional de uma seção qualquer também pode ser determinada pela equação de Poisson, Eq. (1), onde ϕ é a função de tensão e $g(x,y) = 2Gk$. Tomando $Gk=1$ e $\phi=0$ no contorno, obtém-se a rigidez torcional:

$$K = -2 \int_{\Omega} \phi d\Omega \quad (80)$$

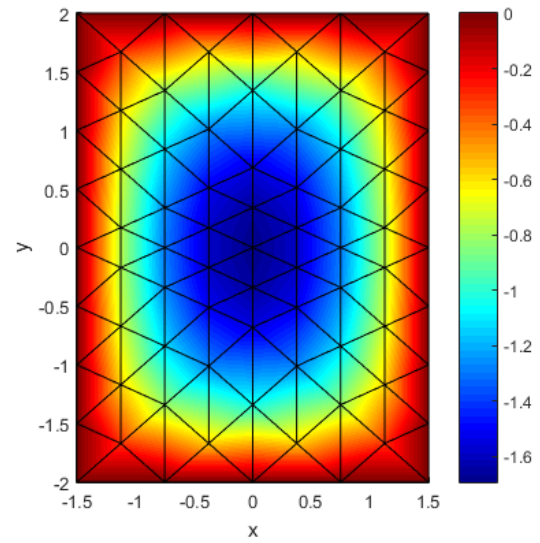
As Figuras 30, 31 e 32 mostram a evolução da malha pelo processo adaptativo e o comportamento da função de tensão no domínio retangular, $[-1,5;-2] \times [1,5;2]$.

Figura 30- Malha inicial – 8 elementos.



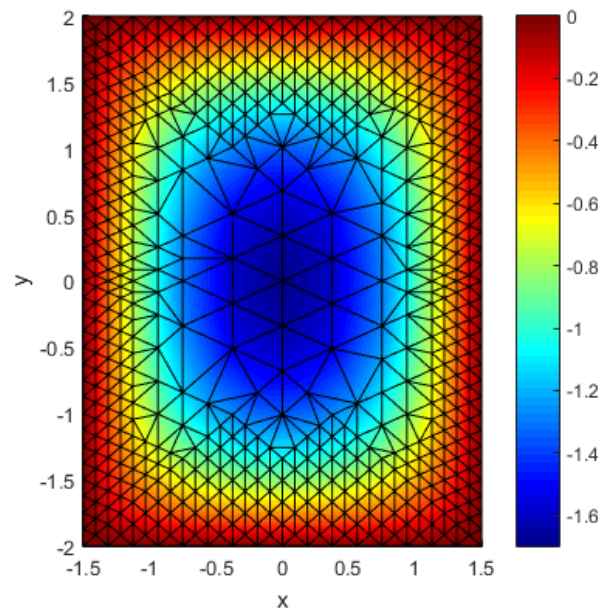
Fonte: A Autora.

Figura 31- 1ª Malha adaptada – 128 elementos.



Fonte: A Autora.

Figura 32- 2ª Malha adaptada – 1198 elementos.



Fonte: A Autora.

Adotou-se $\bar{\eta}=0,05$ para o erro percentual admissível global e local elementar e $Gr_{\max}=2$. A convergência foi atingida na segunda malha adaptada, Tabela 3.

Tabela 3 - Comportamento da rigidez torcional.

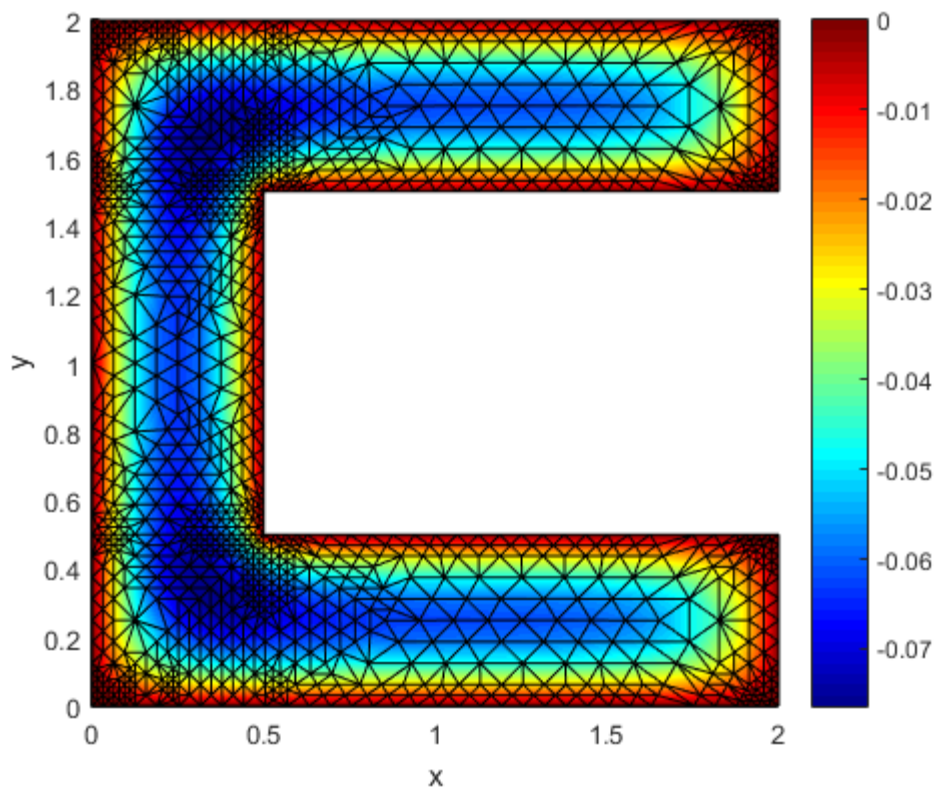
	Malha 1	Malha 2	Malha 3	Valor analítico
K	14,9861	18,8490	19,3823	19,4385

No exemplo da Fig. 33, região em forma de C, adotou-se $\bar{\eta}=0,1$ e $Gr_{\max}=2$. A convergência foi atingida na segunda malha adaptada, Tabela 4. Obteve-se $K=0,2016$ com uma malha final com 2480 elementos, sendo 0,2028 o valor da solução analítica (ROARK *et al.*, 2002).

Tabela 4 - Comportamento da rigidez torcional.

	Malha 1	Malha 2	Malha 3	Valor analítico
K	0,1120	0,1959	0,2016	0,2028

Figura 33- Malha final do refinamento no domínio C – 2480 elementos.



Fonte: A Autora.

4.2 Elasticidade Plana

Serão apresentados dois exemplos numéricos de problemas clássicos de elasticidade bidimensional: viga engastada em balanço e chapa com furo sob tração.

4.2.1 Viga engastada em balanço

Foi analisado o problema da viga engastada com carregamento em sua extremidade livre, ilustrado na Fig. 34. Esse problema é semelhante ao proposto em ZIENKIEWICZ E ZHU (1987), trata-se de uma viga de 10m de comprimento, 1m de altura, com uma carga vertical de 1kN/m aplicada na extremidade direita. A carga está distribuída ao longo dos nós da face direita da viga para evitar singularidades e, conseqüentemente, concentração de tensões, que provocariam a instabilidade do modelo. Como a viga possui 1m de altura, a carga distribuída de 1kN/m ao longo da face direita equivale a uma carga concentrada, $P=1\text{kN}$. O material adotado foi o aço com Módulo de Elasticidade, $E= 200 \text{ GPa}$ e Coeficiente de Poisson, $\nu =0,3$. O peso próprio da estrutura não foi considerado nesta análise.

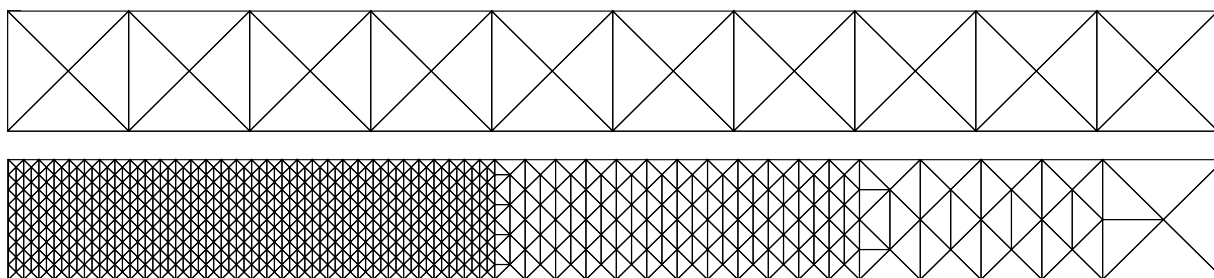
Figura 34- Apresentação do problema da viga engastada em balanço.



Fonte: A Autora.

A malha inicial foi gerada pelo GiD, Fig. 35, de maneira que os elementos fossem regulares e a malha fosse simétrica, para amenizar as distorções dos elementos ao longo das iterações. Nesta análise, o valor do erro percentual em energia admitido foi 15%, tanto para o erro global quanto para o erro local elementar. Adotou-se $Gr_{max}=3$ e a convergência já foi alcançada na primeira malha gerada. A Tabela 5 apresenta os resultados obtidos para a malha gerada no processo de refinamento adaptativo efetuado e a Tabela 6 apresenta os resultados do refinamento uniforme, que foi realizado com o intuito de estabelecer um comparativo com o refinamento adaptativo e verificar sua eficiência.

Figura 35- Evolução da malha da viga engastada em balanço.



Fonte: A Autora.

Tabela 5 - Resultados do refinamento adaptativo da viga engastada em balanço.

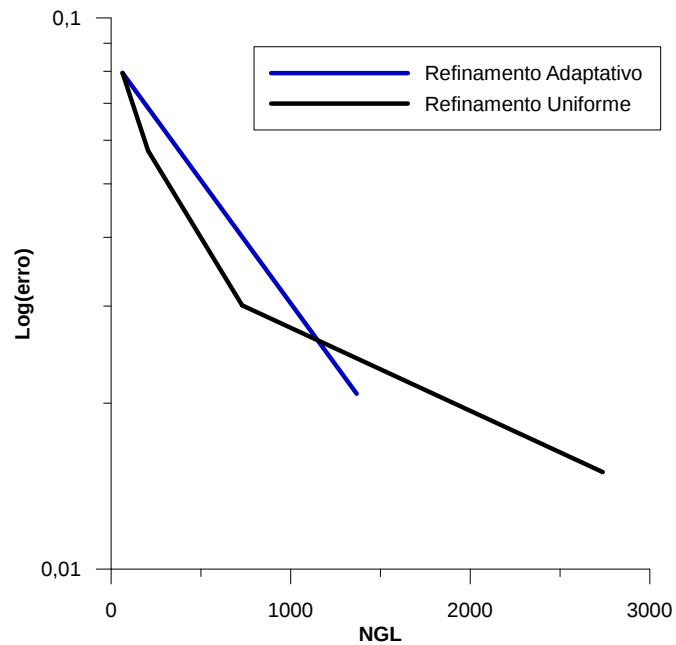
Malha	Número de elementos	NGL	$\ e\$	θ	η (%)	Desl. y (máx) (10^{-5}m)
1	40	64	0,0795	1,146	59,07	-1,180
2	1259	1368	0,0208	1,032	14,69	-1,970

Tabela 6 - Resultados do refinamento uniforme da viga engastada em balanço.

Malha	Número de elementos	NGL	$\ e\$	η (%)	Desl. y (máx) (10^{-5}m)
1	40	64	0,0795	59,07	-1,180
2	160	206	0,0574	40,11	-1,718
3	640	730	0,0301	21,20	-1,930
4	2560	2738	0,0150	10,58	-1,991
5	10240	10594	0,0075	5,28	-2,007

Pelas tabelas acima, observa-se que o erro percentual admissível e o deslocamento da extremidade direita possuem uma boa convergência para um número muito inferior de elementos em comparação com o refino uniforme. O índice de efetividade, θ , calculado pela comparação com a solução obtida por uma malha bem refinada, tende a 1, indicando que o estimador de erro é assintoticamente exato. O gráfico do Erro na Norma em Energia ($\|e\|$) versus Número de Graus de Liberdade (NGL), Fig. 36, evidencia a eficiência do adaptativo em alcançar bons resultados a um menor custo computacional. Conforme esperado, a curva referente ao refinamento adaptativo aproxima-se mais rapidamente do eixo x, indicando que a solução converge mais rapidamente.

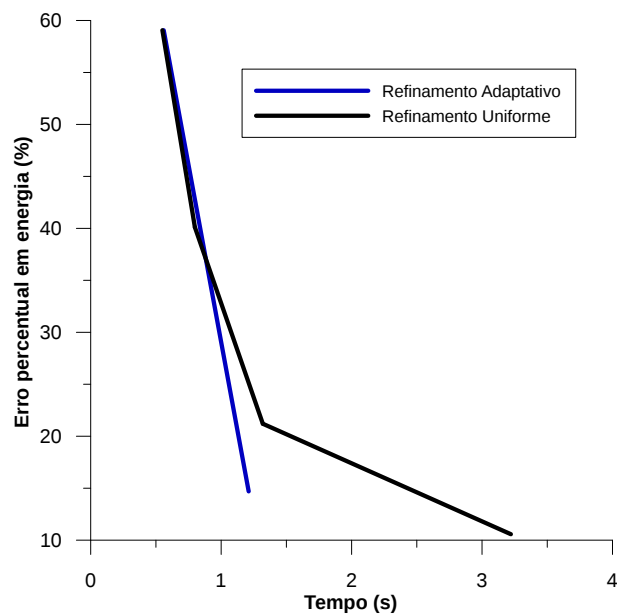
Figura 36- Erro na norma em energia versus NGL.



Fonte: A Autora.

O gráfico da Fig. 37 faz um comparativo do esforço computacional do refinamento adaptativo em relação ao uniforme. Evidenciando, mais uma vez, a convergência mais rápida do adaptativo a um menor custo computacional.

Figura 37- Análise do esforço computacional.



Fonte: A Autora.

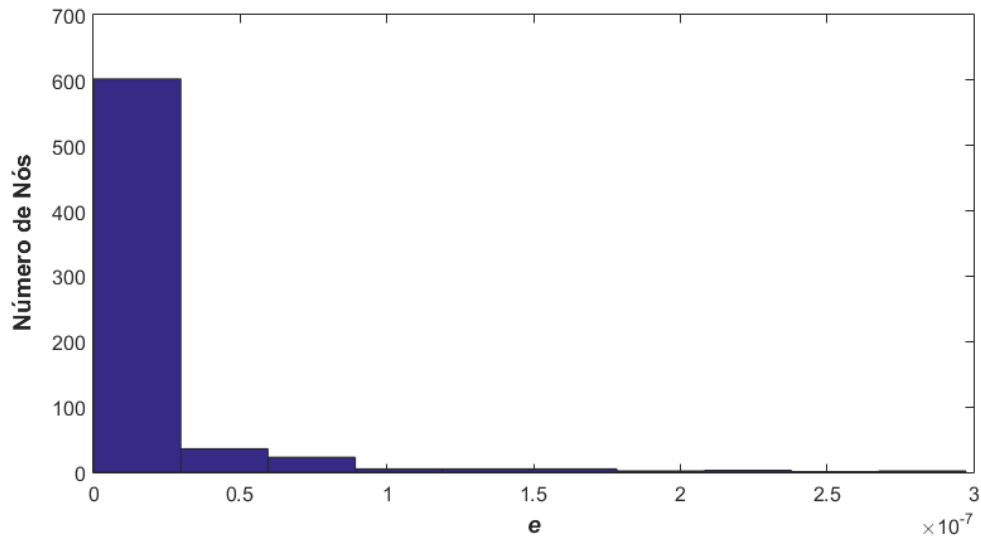
Para validação da tendência do refinamento adaptativo na viga engastada em balanço foi feita uma comparação com malhas de outros trabalhos, a exemplo da obtida por KRISHNAMOORTHY & UMESH (1993), que mostram que o refinamento caminha para a região próxima ao engaste, na qual concentram-se os menores elementos. O solucionador MEF implementado foi validado pela comparação com os resultados dos deslocamentos horizontais e verticais obtidos pela solução analítica, Eq. (81) e (82), respectivamente (TIMOSHENKO, 1951).

$$u = -\frac{Px^2y}{2EI} - \frac{\nu Py^3}{6EI} + \frac{Py^3}{6IG} + \left(\frac{PL^2}{2EI} - \frac{Pc^2}{2IG} \right) y \quad (81)$$

$$v = \frac{\nu Pxy^2}{2EI} + \frac{Px^3}{6EI} - \frac{PL^2x}{2EI} + \frac{PL^3}{3EI} \quad (82)$$

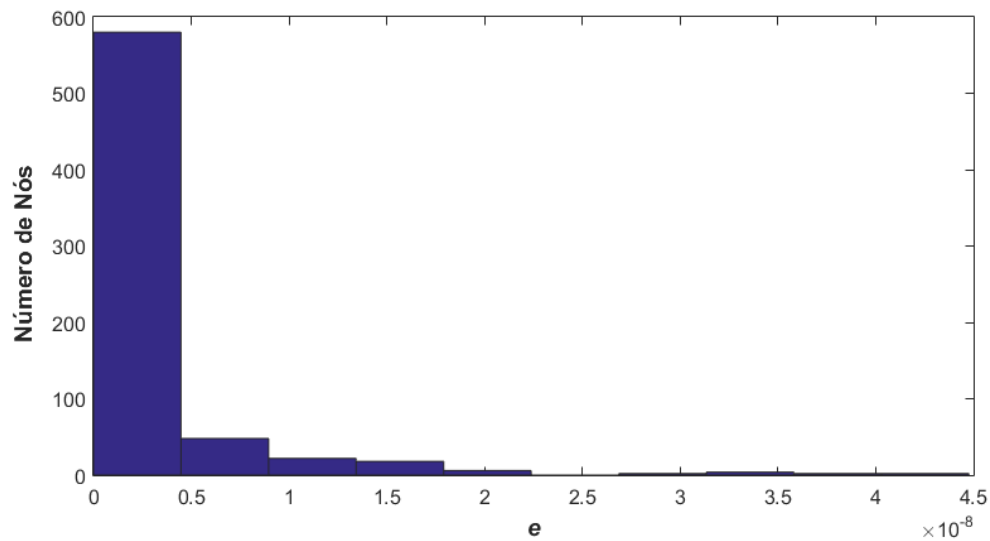
Onde x e y são as coordenadas do ponto aonde o deslocamento está sendo calculado, I é o momento de inércia da viga e G é o módulo de cisalhamento. O deslocamento máximo vertical obtido na extremidade direita da viga foi $2,0 \cdot 10^{-5} \text{m}$. O código implementado mostrou um deslocamento máximo vertical de $1,970 \cdot 10^{-5} \text{m}$, para a malha final adaptada com 1259 elementos. Os histogramas das Fig. 38 e 39 mostram a distribuição dos erros de aproximação, Eq. (52), nos nós da malha. Percebe-se que a grande maioria dos nós apresentam erro muito próximo de zero e que os maiores erros são da ordem de 10^{-7} e 10^{-8} para os deslocamentos verticais e horizontais, respectivamente. Portanto, tanto a tendência de refinamento quanto os resultados numéricos fornecidos pelo código são consistentes.

Figura 38- Número de Nós versus Erro de aproximação (e) – Deslocamentos verticais.



Fonte: A Autora.

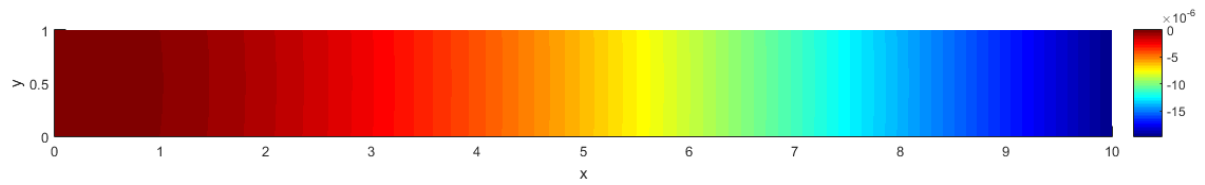
Figura 39- Número de Nós versus Erro de aproximação (e) – Deslocamentos horizontais.



Fonte: A Autora.

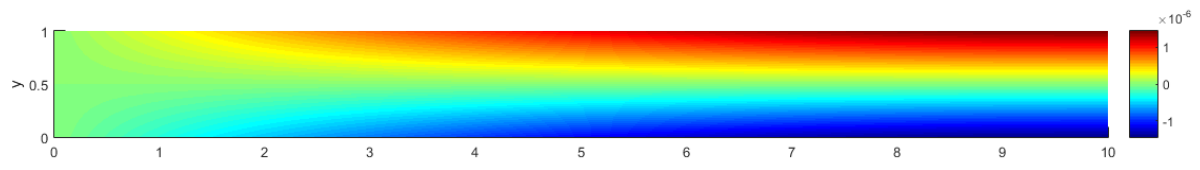
Os deslocamentos na direção x e y são mostrados nas Fig. 40 e 41. A distribuição tensões σ_x , σ_y e τ_{xy} são apresentadas nas Fig. 42, 43 e 44, respectivamente.

Figura 40- Distribuição de deslocamentos em y da viga engastada em balanço.



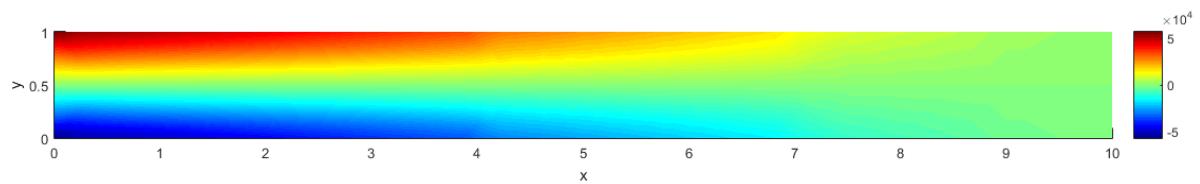
Fonte: A Autora.

Figura 41- Distribuição de deslocamentos em x da viga engastada em balanço.



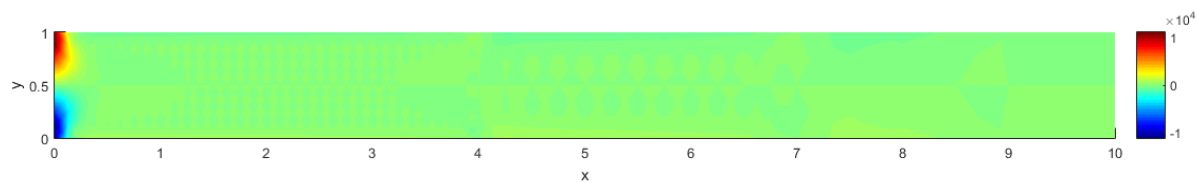
Fonte: A Autora.

Figura 42- Distribuição de σ_x da viga engastada em balanço.



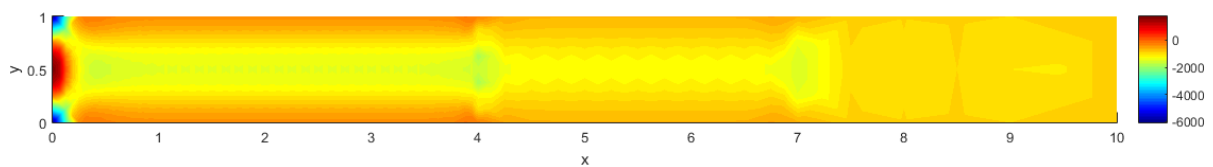
Fonte: A Autora.

Figura 43- Distribuição de σ_y da viga engastada em balanço.



Fonte: A Autora.

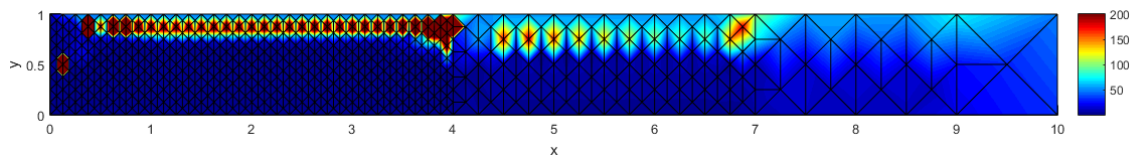
Figura 44- Distribuição de τ_{xy} da viga engastada em balanço.



Fonte: A Autora.

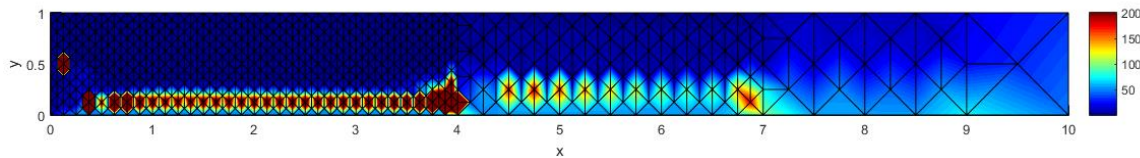
Os erros das tensões foram avaliados com relação às tensões normais máximas e mínimas, Eq. (26) e (27). A partir do comparativo entre uma malha muito refinada, com 10240 elementos, e a malha gerada pelo refinamento adaptativo, com 1259 elementos. Observando a distribuição dos erros percentuais, Fig. 45 e 46, e o histograma do erro percentual das tensões normais máximas e mínimas, Fig.47, percebe-se que a grande maioria dos nós possuem erro percentual menor que 20%. Portanto, apesar de alguns nós apresentarem altos erros percentuais, a solução como um todo convergiu.

Figura 45- Distribuição do erro percentual de tensões normais mínimas.



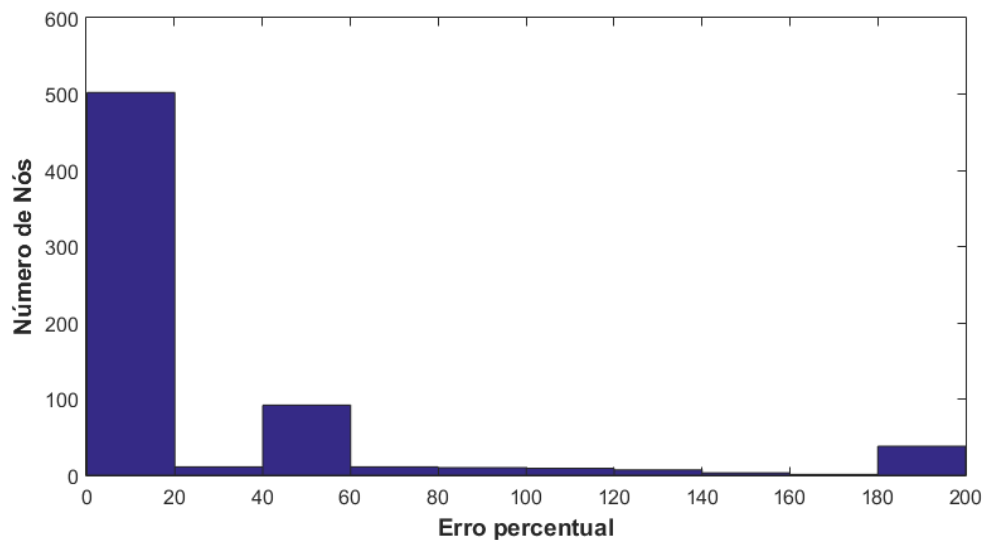
Fonte: A Autora.

Figura 46- Distribuição do erro percentual de tensões normais máximas.



Fonte: A Autora.

Figura 47- Histograma da distribuição do erro percentual das tensões normais.

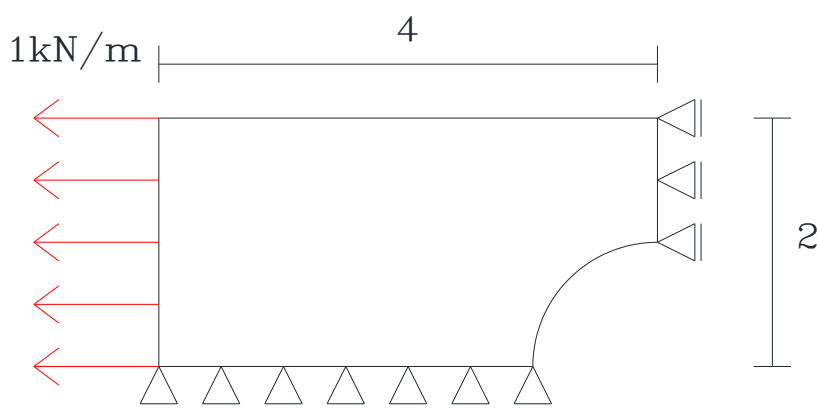


Fonte: A Autora.

4.2.2 Chapa com furo

A chapa sob tração analisada possui 8m de comprimento por 4 metros de altura e um furo central de 1 m de raio. No entanto, fez-se uso da simetria do problema analisando-se apenas um quarto da placa. O problema tornou-se, então, uma chapa de 4m de comprimento, 2m de altura e um quarto de círculo de 1m de raio como furo, Fig. 48. A força de tração aplicada na face lateral esquerda é de 1kN/m. Para garantir a simetria do problema, foram inseridos apoios do primeiro gênero na face inferior da chapa, restringindo o movimento em y, e na face lateral direita, restringindo o movimento em x. O material utilizado foi o aço, com módulo de elasticidade de 210 GPa e coeficiente de Poisson de 0,3.

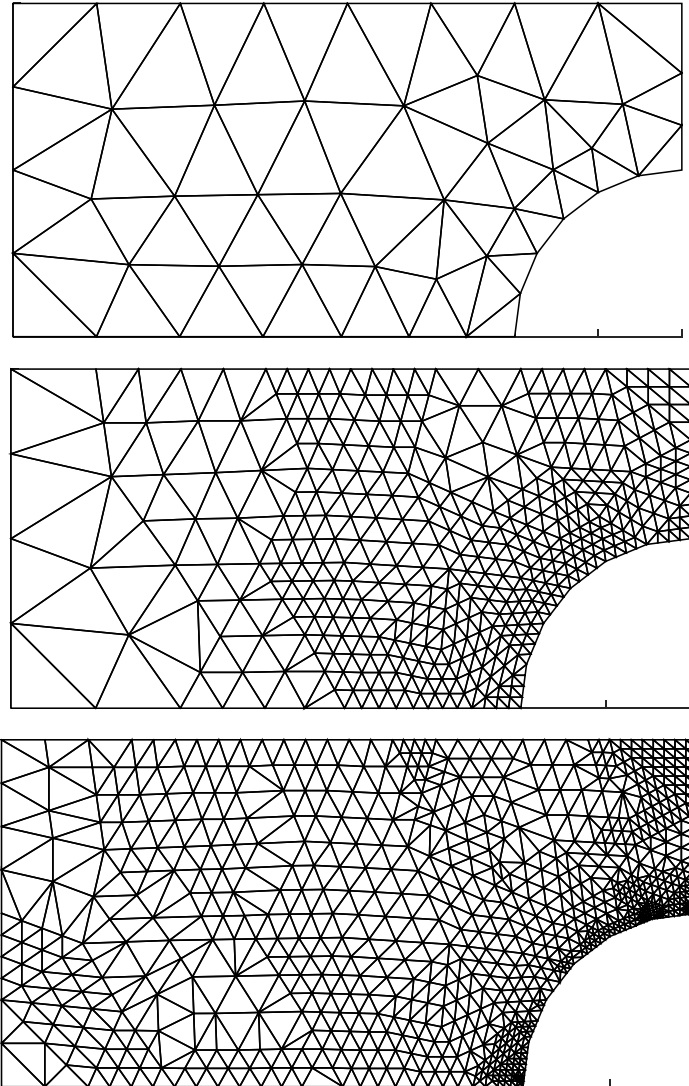
Figura 48- Apresentação do problema da chapa com furo.



Fonte: A Autora.

A malha inicial do problema foi gerada pelo GiD, Fig. 49. O tamanho mínimo dos elementos foi escolhido de maneira que a malha inicial fosse pobre, mas representasse bem o domínio, já que os nós do contorno da malha inicial são utilizados para definição do domínio e geração das demais malhas. Para a análise do problema, adotou-se para o domínio global um erro relativo percentual em energia admissível de 4% e como critério elementar um erro relativo percentual em energia admissível de 5%. Os resultados obtidos para cada malha gerada durante o processo de refinamento e os parâmetros que definem a qualidade deste processo estão descritos na Tabela 7. O deslocamento apresentado na tabela é o máximo, que ocorre na extremidade superior da face direita do domínio.

Figura 49- Evolução da malha da chapa com furo.



Fonte: A Autora.

Tabela 7 - Resultados do refinamento adaptativo da chapa com furo.

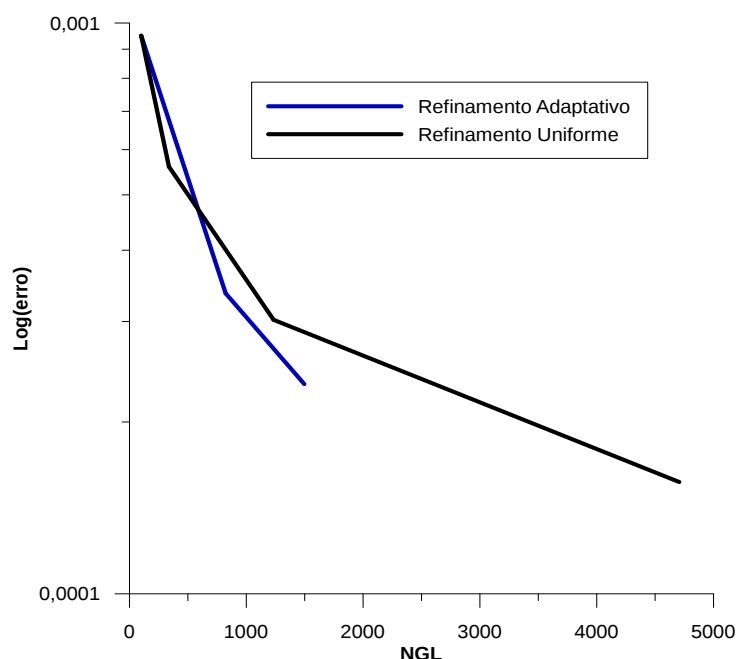
Malha	Número de elementos	NGL	$\ e\ $ (10^{-3})	θ	η (%)	Desl. y (máx) (10^{-8}m)
1	70	100	0,950	1.012	13,07	-1,042
2	741	824	0,336	1.015	4,60	-1,204
3	1496	1496	0,233	0,998	3,19	-1,220

Tabela 8 - Resultados do refinamento uniforme da chapa com furo.

Malha	Número de elementos	NGL	$\ e\ $ (10^{-3})	η (%)	Desl. y (máx) (10^{-8}m)
1	70	100	0,950	13,07	-1,042
2	280	338	0,560	7,68	-1,162
3	1120	1234	0,302	4,13	-1,209
4	4480	4706	0,157	2,14	-1,226
5	17920	18370	0,082	1,12	-1,230

Pelas tabelas acima, observa-se que o erro percentual admissível e o deslocamento da extremidade superior direita possuem uma boa convergência com um número muito inferior de elementos em comparação com o refino uniforme. O gráfico do Erro na Norma em Energia ($\|e\|$) versus Número de Graus de Liberdade (NGL), Fig. 50, evidencia a eficiência do adaptativo em alcançar bons resultados a um menor custo computacional. Conforme esperado, as curvas tendem a tangenciar o eixo x, sendo a curva referente ao refinamento adaptativo aquela que se aproxima mais rapidamente deste eixo. O índice de efetividade, θ , tendeu a 1, indicando que estimador de erro utilizado é de boa qualidade.

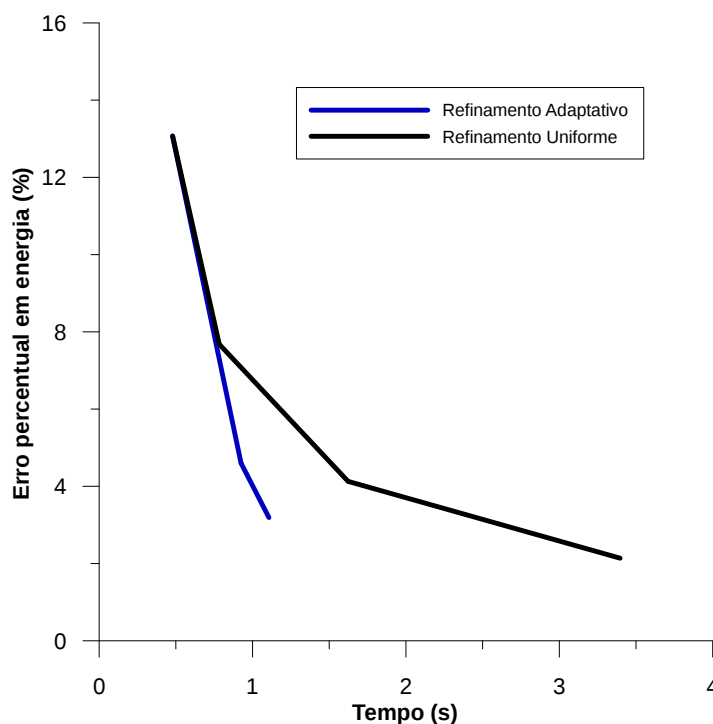
Figura 50- Erro na norma em energia versus NGL.



Fonte: A Autora.

O gráfico da análise do esforço computacional, Fig. 51, evidencia, mais uma vez, a eficiência do programa em realizar às análises a um baixo custo computacional. Foi necessário menos de 2 segundos para que a análise utilizando o refinamento adaptativo atingisse a convergência e fosse finalizada.

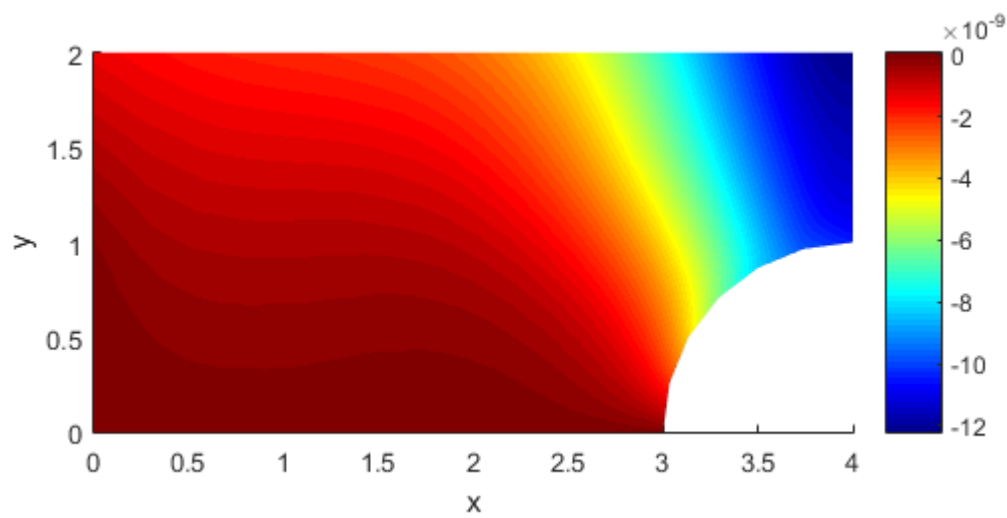
Figura 51- Análise do esforço computacional.



Fonte: A Autora.

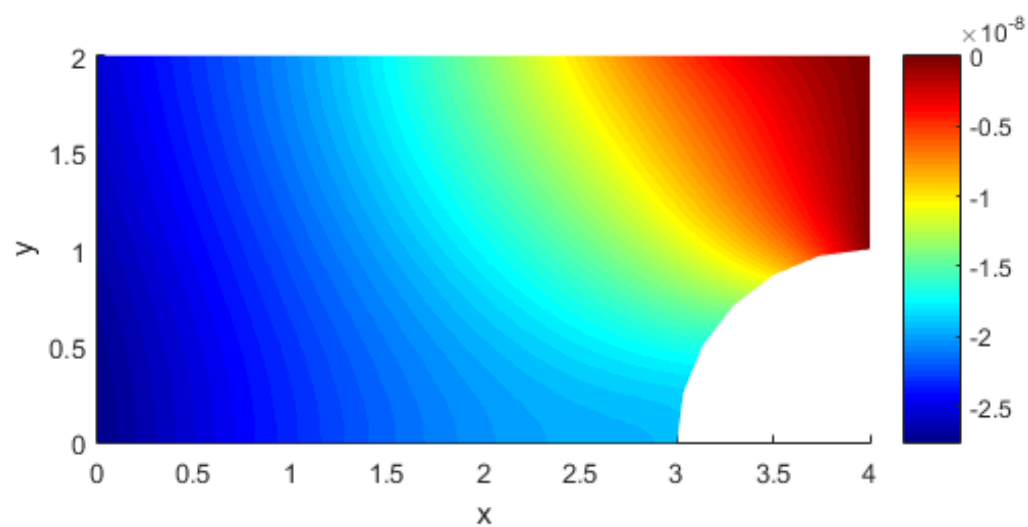
Para validação da tendência de refinamento da malha final foi feita uma comparação com malhas de outros trabalhos que mostram que o refinamento caminha para a região próxima ao furo. Foi apresentado anteriormente um exemplo de refinamento adaptativo realizado pelo COMSOL – Software de Elementos Finitos comercial – que mostra esta tendência, Fig. 8. Os deslocamentos na direção y e x são mostrados nas Fig. 52 e 53. A distribuição tensões σ_x , σ_y e τ_{xy} são apresentadas nas Fig. 54, 55 e 56, respectivamente.

Figura 52- Distribuição de deslocamentos em y da chapa com furo.

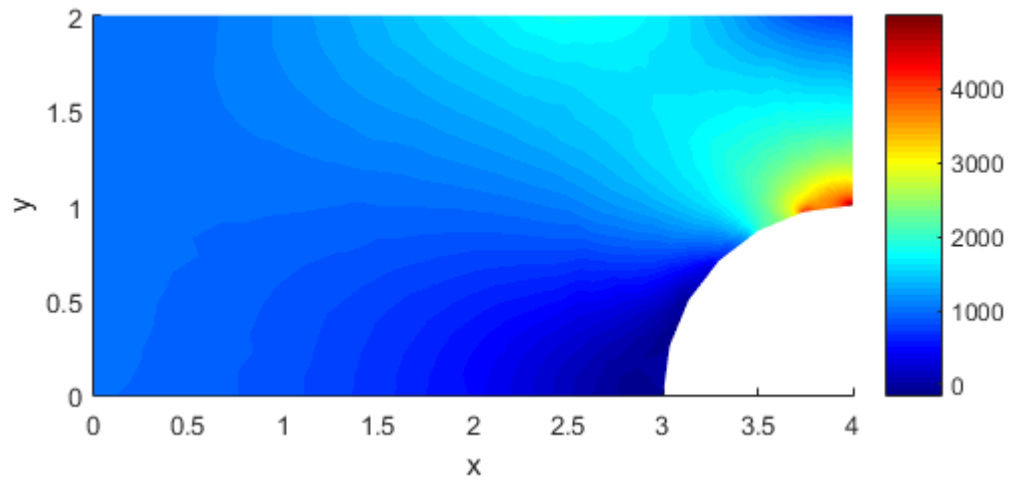


Fonte: A Autora.

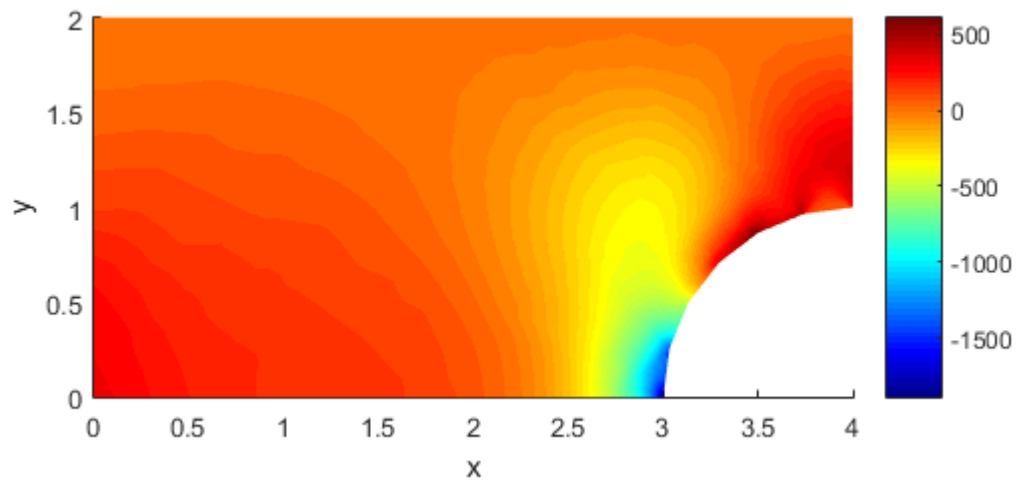
Figura 53- Distribuição de deslocamentos em x da chapa com furo.



Fonte: A Autora.

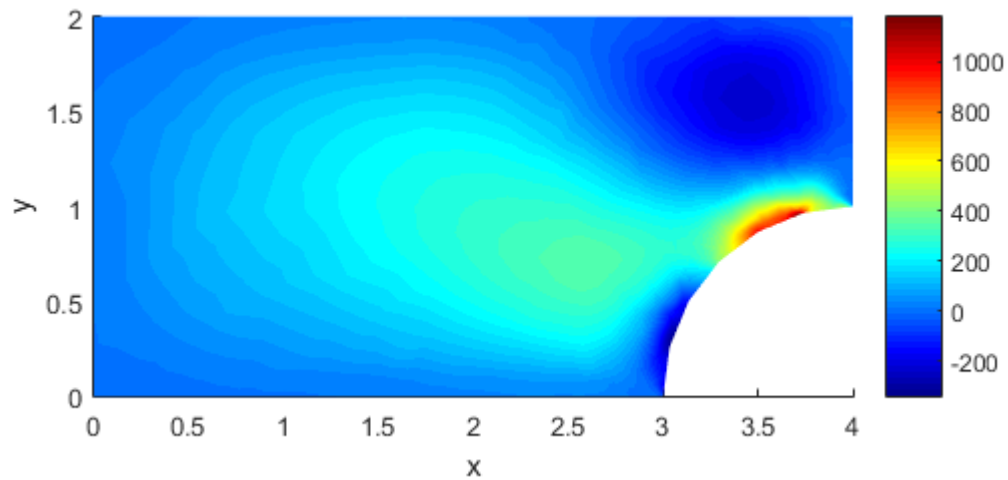
Figura 54- Distribuição de σ_x da chapa com furo.

Fonte: A Autora.

Figura 55- Distribuição de σ_y da chapa com furo.

Fonte: A Autora.

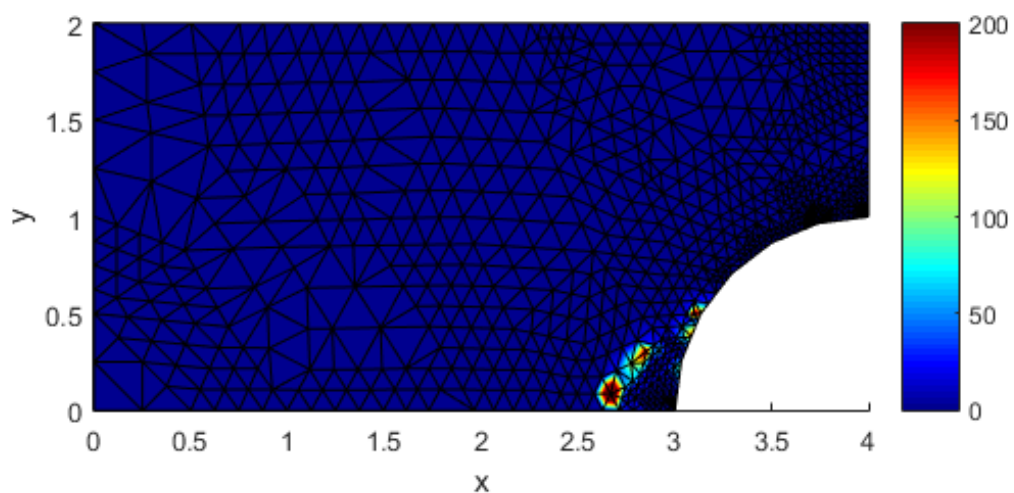
Figura 56- Distribuição de τ_{xy} da chapa com furo.



Fonte: A Autora.

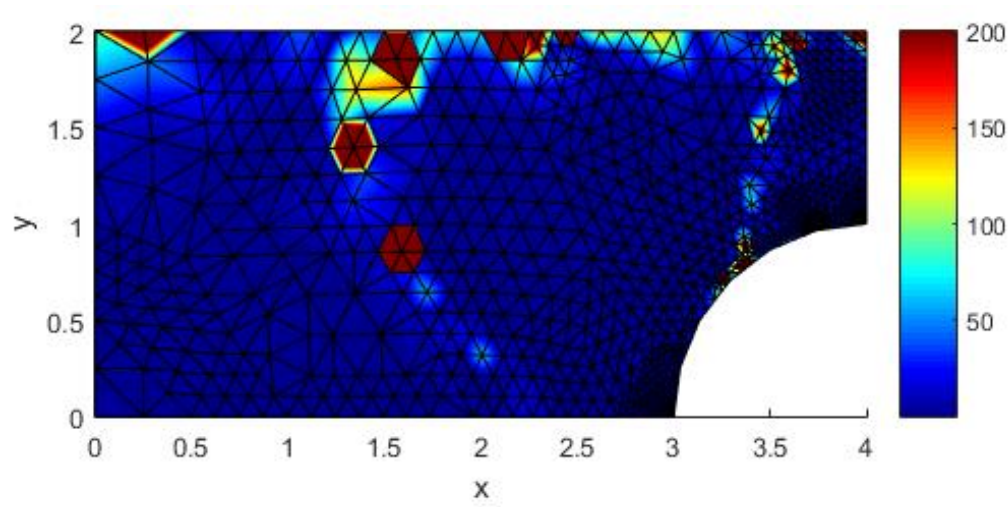
Os erros das tensões foram avaliados com relação às tensões normais máximas e mínimas, Eq. (26) e (27). A partir do comparativo entre uma malha muito refinada, com 17290 elementos, e a malha gerada pelo refinamento adaptativo, com 1496 elementos. Observando a distribuição dos erros percentuais, Fig. 57 e 58, e os histogramas do erro percentual, Fig. 59 e 60, percebe-se que cerca de 94% dos nós possuem erro percentual menor que 20%. Portanto, de maneira geral os valores das tensões convergiram, obtendo-se resultados satisfatórios.

Figura 57- Distribuição do erro percentual de tensões normais máximas.



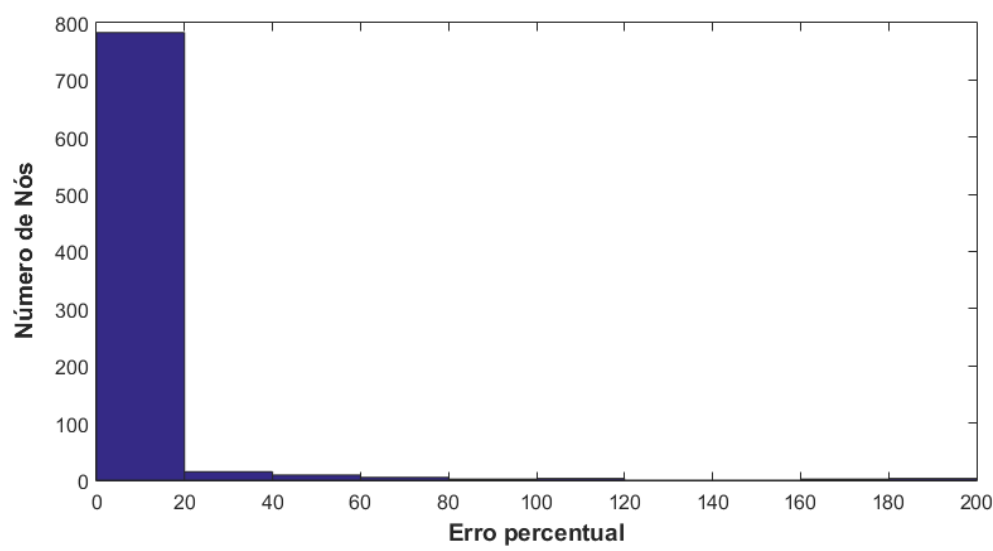
Fonte: A Autora.

Figura 58- Distribuição do erro percentual de tensões normais mínimas.



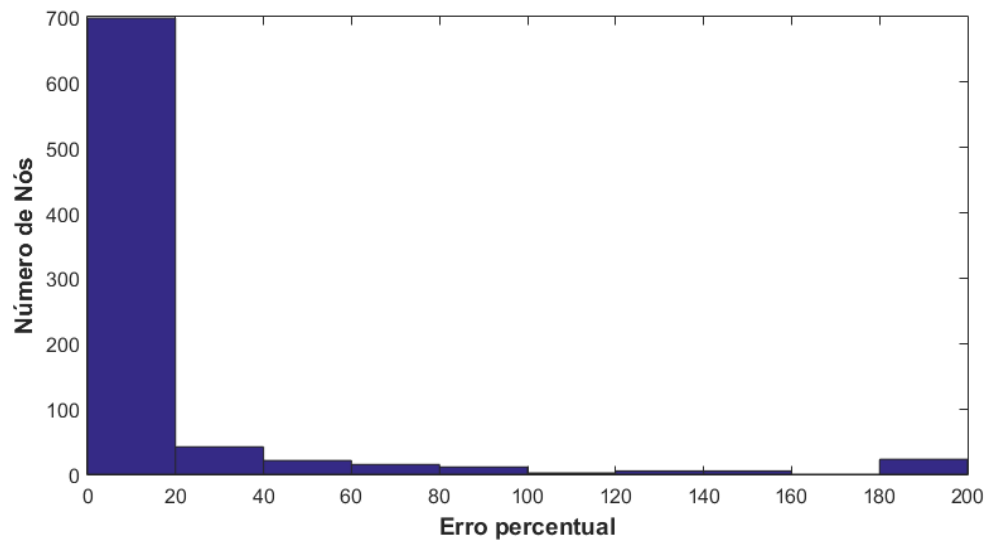
Fonte: A Autora.

Figura 59- Histograma do erro percentual de tensões normais máximas.



Fonte: A Autora.

Figura 60- Histograma do erro percentual de tensões normais mínimas.



Fonte: A Autora.

5 CONCLUSÕES

Os exemplos apresentados e validados confirmam o funcionamento do programa desenvolvido para solução de diversos problemas bidimensionais descritos pela equação de Poisson e de Elasticidade Plana. As estratégias de adaptação e o estimador de erro a posteriori baseado na norma do erro em energia e na recuperação do gradiente da solução pelo método MNS, mostraram-se eficientes. O erro relativo percentual admissível foi atingido com um número muito menor de graus de liberdade que o refinamento uniforme em todas as análises, conforme evidenciado nos gráficos do Erro na Norma em Energia ($\|e\|$) versus Número de Graus de Liberdade (NGL).

Os gráficos da análise do esforço computacional mostraram a rapidez com que a solução numérica é obtida, a exemplo do problema da viga engastada em balanço que teve sua análise finalizada em menos de 2 segundos, atingindo a convergência na primeira malha adaptativa. Além disso, o comparativo do esforço computacional do refinamento adaptativo em relação ao uniforme evidencia, mais uma vez, a convergência mais rápida do adaptativo a um menor custo computacional.

Em todos os casos apresentados o refinamento caminhou na direção esperada, quando comparado a trabalhos disponíveis na literatura, refinando os locais onde haviam singularidade e maior gradiente. Quanto a eficiência do estimador de erro adotado, verificou-se que o índice de efetividade aproximou-se de 1 em todas as análises, indicando que a estimativa do erro converge para o erro verdadeiro.

Quanto a qualidade da malha, observou-se que quando o refinamento partiu de uma malha inicial mais regular e simétrica, ao final do processo de refinamento adaptativo os elementos eram mais regulares e apresentavam menores distorções. Além disso, o critério de desnível unitário interelementos juntamente com o critério de regularização da malha promoveram a continuidade da distribuição dos níveis de refinamento determinados pela análise de erro. Eles evitam, por exemplo, que um elemento de grau 3 tenha como vizinho um elemento de grau 1, pois isso geraria elementos muito irregulares que, numa próxima iteração, passariam essa característica para os seus possíveis elementos filhos.

Os critérios de desnível unitário e a limitação do grau máximo de refinamento por etapa contribuíram para evitar o refinamento excessivo da malha em regiões de singularidade. Outra medida adotada com respeito à presença de singularidades, foi a limitação do tamanho mínimo dos elementos da malha.

Os procedimentos apresentados são compatíveis para o desenvolvimento de programas em outras áreas da análise numérica e ao desenvolvimento da capacidade de desrefinamento em regiões que apresentam erro abaixo da tolerância desejada.

5. 1 Sugestões para trabalhos futuros

Embora o corrente trabalho tenha atingido os objetivos propostos, determinados aspectos podem ser aperfeiçoados e outros campos de análise podem ser explorados. Dentre os quais, pode-se sugerir:

- Utilizar um gerador de malha para a realização dos processos adaptativos, permitindo uma maior liberdade na manipulação da geração da malha e no controle da qualidade dos elementos.

- Considerar o domínio inicial para as adaptações da malha. Ao utilizar a malha inicial para as adaptações executadas, perde-se a informação original do domínio do problema, estando a qualidade da malha final adaptada sujeita à malha inicial. O custo computacional de conservar a geometria é maior, mas este é um fator de grande importância, principalmente para domínios irregulares.

- Executar o desrefinamento da malha, otimizando o processo de refinamento, além de permitir a análise transiente. Para tanto, é interessante que seja utilizada uma estrutura de dados em árvore, conforme proposto por Araújo (2015).

- Estender a outros tipos de malhas, como de elementos quadrangulares e malhas mistas, neste caso, seria necessário o desenvolvimento de uma ferramenta de análise de erros para malhas em MEF mistas.

- Ampliação da análise para outros estimadores de erro, como o SPR e o REP.

- Emprego de estimativas de erro em processos adaptativos anisotrópicos, que levariam em conta os parâmetros direcionais do erro, entre outros fatores diferentes dos analisados pelo estimador utilizado neste trabalho.

REFERÊNCIAS

- ARAÚJO, F. S. **Procedimentos Adaptativos na simulação de reservatórios de petróleo em volumes finitos**. Dissertação de mestrado, UFPE. Recife, 2004. 109 p.
- ASSAN, A. E. **Método dos Elementos Finitos: Primeiros Passos**. Ed. Unicamp, Brasil, 298p, 1999.
- BANK, R. E; SHERMAN, A. H. An Adaptive, Multi-Level Method for Elliptic Boundary Value Problems. **Symposium on Reservoir Simulation held in Denver**, Colorado. Computing, vol. 26, p. 91-105. 1981.
- CASTELLAZZI, G.; de MIRANDA, S.; UBERTINI, F. Adaptivity based on the recovery by compatibility in patches. **Finite Elements in Analysis and Design**, v. 46, p. 379–390, 2010.
- CERVENKA, J. **Grid refinement – adaptive meshes**. Red-Green-Refinement. 1996. Disponível em: <http://www.iue.tuwien.ac.at/phd/cervenka/node14.html>. Acesso em 22 de julho de 2018.
- CIALERT, P. G. **The Finite Element Method for Elliptic Problems**. Paris: Ed. Elsevier. 530 p. 2002.
- COMSOL. Comsol Blog. **Meshing Considerations for Linear Static Problems**. 2017. Disponível em: <https://br.comsol.com/blogs/meshing-considerations-linear-static-problems/>. Acesso em: 29 julho 2017.
- COOK, R. D.; MALKUS, D. S.; PLESHA, M. E.; WITT, R. J. **Concepts and Applications of Finite Element Analysis**. USA: 1. ed. John Wiley and Sons, 2002.
- FANCELLO, E.A; NOVOTNY, A. A. Um refinamento h, p e hp adaptativo na análise de flexão de placas semi-espessas. **Revista internacional de métodos numéricos para cálculo y Diseño em Ingeniería**. Vol. 14, n. 1, p. 25-48, 1998.
- GERE, J. M. **Mechanics of Materials**, 5th ed., Brooks/Cole Publishers, Pacific Grove, CA, 2001.
- HINTON, E.; CAMPBELL, J. S. Local and global smoothing of discontinuous finite element functions using a least squares method. **International Journal for Numerical Methods in Engineering**. Vol.8, p. 461-480, 1974.
- KRISHNAMOORTHY, C. S.; UMESH, K. Rajeshirke. **Adaptive Mesh Refinement for Two-Dimensional Finite Element Stress Analysis**. Computers & Structures, Great Britain, 1993. Vol. 48, p. 121-133.
- KWON, Y. W & BANG, H., 1997. **The Finite Element Method Using Matlab**. CRC
- LINS, R. M. **Estimador de Erro a Posteriori Baseado em Recuperação do Gradiente para o Método dos Elementos Finitos Generalizados**. 2011. 94 p. Mestrado (Engenharia de Estruturas)- USP, São Carlos/SP, 2011.
- LYRA, P. R. M. **Uma estratégia de refinamento auto-adaptativo versão-h do Método dos Elementos Finitos Aplicada a Problemas Bidimensionais Regidos pela Equação de Campo**. Dissertação de Mestrado – Engenharia Civil / Estruturas – COPPE/UFRJ. 1998
- LOGAN, D. L. **A First Course in the Finite Element Method**. Cengage, 808 p. 2007.
- ONÃTE, E.; BUGEDA, G. A study of mesh optimality criteria in adaptive finite element analysis. **Engineering Computations**, v. 10, p. 307–321, 1993.

PASSOS, J. P. **Desenvolvimento e implementação de um procedimento adaptativo para elementos finitos triangulares baseado em estimador de erro a posteriori**. Dissertação de mestrado – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012.

REDDY, J. N. **An Introduction to the Finite Element Method**. New York: 3. ed. Mc Graw Hill, 2006.

RIVARA, M. C. Algorithms for Refining Triangular Grids Suitable for Adaptive and Multigrid Techniques. **International Journal for Numerical Methods in Engineering**, vol. 20, p. 745-756. 1984.

ROARK, R. J.; BUDYNAS, R. G.; YOUNG, W. C. **Roark's formulas for stress and strain**. McGraw-Hill, 852 p. 2002.

SAMPAIO, P. A. B.; LYRA, P. R. M.; MORGAN, K.; WEATHERILL, N. P. Petrov-Galerkin Solutions of Incompressible Navier-Stokes Equations in Primitive Variables with Adaptive Remeshing. **Computer Methods Applied Mechanics and Engineering**, 106, 146-178. 1993.

SILVA, J. **Análise de estimadores de erro a posteriori aplicados ao método dos elementos finitos utilizando refino h-adaptativo**. Dissertação de Mestrado, UFPR. Curitiba, 2015. 149 p.

SOUZA, R. M. **O método dos elementos finitos aplicado ao problema de condução de calor**. Apostila, UFPA. Pará, 2003. 40 p.

TAYLOR, R. L; ZIENKIEWICZ, O.C.; ZHU, J.Z. **The Finite Element Method: It's Basis and Fundamentals**. Butterworth-Heinemann, 2005.

TIMOSHENKO, S.; GOODIER, J. N. **Theory of Elasticity**. McGraw-hill Book Company, 1951.

TIMOSHENKO, S.; WOINOWSKY-KRIEGER, S. **Theory of Plates and Shells**. Mcgraw-hill. 1989.

ZIENKIEWICZ, O. C.; TAYLOR, R. L. **The Finite Element Method**. Butterworth-Heinemann. 2000.

ZIENKIEWICZ, O. C. & ZHU, J. Z. A simple error estimator and adaptive procedure for practical engeneering analysis. **International Journal for Numerical Methods in Engineering**, vol. 24, p. 333-357, 1987.

APÊNDICE A – INTEGRAÇÃO NUMÉRICA

A integral da Eq. (13), que se refere à fonte interna e resulta num vetor de fluxos, foi calculada por integração numérica. Considerando o elemento finito triangular com coordenadas homogêneas, pela quadratura de Gauss-Legendre, pode-se apresentar o cálculo da integral sobre esse domínio como (ASSAN, 1999):

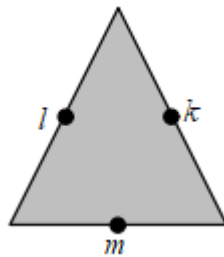
$$\int_A g(x, y) dA = \sum_{i=1}^n f(\xi_1, \xi_2, \xi_3) \omega_i \det J \quad (83)$$

Onde ω_i são os pesos (*Hammer*) para $-1 \leq \xi_i \leq 1$ ($i=1,2,3$) e J é a matriz Jacobiana dada por:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial x}{\partial \xi_2} \\ \frac{\partial y}{\partial \xi_1} & \frac{\partial y}{\partial \xi_2} \end{bmatrix} = \begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix} \quad (84)$$

Neste trabalho, para o cálculo da integral citada, foram utilizados 3 pontos de Hammer, Fig. 61. A Tabela 9 indica os pontos e pesos para o grau quadrático. Uma tabela completa está disponível em ASSAN (1999).

Figura 61- Pontos de integração para os elementos triangulares planos.



Fonte: Adaptado de ASSAN (1999).

Tabela 9 - Pontos e pesos (Hammer).

n (grau)	Pontos	ξ_1	ξ_2	ξ_3	ω_i
3 (quadrático)	l	0	1/2	1/2	1/6
	m	1/2	0	1/2	
	k	1/2	1/2	0	

As variáveis x e y da função a ser integrada são substituídas por:

$$\begin{aligned} x &= x_1 \xi_1 + x_2 \xi_2 + x_3 \xi_3 \\ y &= y_1 \xi_1 + y_2 \xi_2 + y_3 \xi_3 \end{aligned} \tag{85}$$

Onde (x_i, y_i) são as coordenadas dos nós do triângulo.

APÊNDICE B – CÓDIGO PARA O REFINAMENTO H-ADAPTATIVO APLICADO À ELASTICIDADE PLANA – *ELT2D*

```
% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% REFINAMENTO H-ADAPTATIVO APLICADO À ELASTICIDADE PLANA

%% Inicialização
clc
clear all
close all

format long

%% 1. DADOS DE ENTRADA

%1.1 PROPRIEDADES DO MATERIAL

modE=210*10^9; %Módulo de Elasticidade do material (Pa)
esp=1; % Espessura (admitida como unitária)
vp=0.3; % Coeficiente de Poisson
Xb=0; %densidade de peso na direção x (= densidade * g) (kN/m³)
Yb=7860*9.81; %densidade de peso na direção y (= densidade * g) (kN/m³)

%1.2 DELIMITAÇÃO GEOMÉTRICA DO CONTORNO

vertices=[6,0,4,4,3,3,0,2,2,1,1,0,0];%O valor da primeira posição indica o
número de arestas do polígono. Em sequência vem as coordenadas x e depois
as coordenadas y dos vértices
%OBS1: se o domínio do problema não for um polígono, digite as coordenadas
dos vértices que descreve o polígono que mais se aproxima da região, de
maneira que esses vértices delimitem os limites superiores e inferiores das
condições de contorno
%OBS2: se a condição de contorno ou o carregamento não se aplicar a todo o
lado do polígono, o usuário deve inserir neste vetor as coordenadas de um
nó delimitante

%1.3 CONDIÇÕES DE CONTORNO

v1=[2,5]; % número do vértice (origem) que contém a condição de contorno
(pertencente ao lado). Ex.: se a restrição ocorre no lado que contém o
segundo vértice e o terceiro, então insira 2 neste vetor e 3 no vetor v2.
v2=[3,6]; % número do vértice (fim) que contém a condição de contorno
(pertencente ao lado)
restx=[1,0]; % 1- indica que a condição de contorno aplicada apresenta
restrição de deslocamento em x. restx(i) se refere ao comprimento definido
por v1(i)<->v2(i)
resty=[0,1]; % 1- indica a condição de contorno aplicada apresenta
restrição de deslocamento em y. restx(i) se refere ao comprimento definido
por v1(i)<->v2(i)

%1.4 FORÇAS EXTERNAS UNIFORMEMENTE DISTRIBUÍDAS

vfp1=[1];%vértice (origem) sobre o qual agem forças externas (pertencente
ao lado)
vfp2=[6];%vértice (origem) sobre o qual agem forças externas (pertencente
ao lado)
```

```

F_ext_x=[-1000]; %vetor que contém as forças externas na direção x
(ordenado pela mesma sequência de lados do vetor que contém os vértices)
F_ext_y=[0]; %vetor que contém as forças externas na direção y (ordenado
pela mesma sequência de lados do vetor que contém os vértices)

%1.5 DADOS P/ O REFINAMENTO
Grmax=2; %grau de refinamento máximo permitido por etapa
d_min=10^-4; %usuário determina a dimensão mínima do elemento (isto evita o
refinamento excessivo em singularidades)
nadam1=0.05; %erro relativo percentual admissível global (valor entre 0 e
1)
nadam2=0.05; %erro relativo percentual admissível elementar (valor entre 0
e 1)

%% 1.6 FORÇAS DE CORPO

prompt= 'Considerar o peso próprio da estrutura? 0-não, 1-sim';
p_proprio=input(prompt);

%% 2. MALHA INICIAL (IMPORTADA DO GiD)

%Inserir matriz de coordenadas do GiD
c=[];

%Inserir matriz de conectividades do GiD
tri=[];

%Inserir vetor com as coordenadas x dos nós do contorno
xv=[];

%Inserir vetor com as coordenadas y dos nós do contorno
yv=[];

%% 3. OPERAÇÕES GERAIS

tri(:,1)=[]; %remove a primeira coluna da matriz de conectividades
importada do GiD
c(:,4)=[]; %remove a última linha da matriz de coordenadas importada do GiD
x=c(:,2); %coordenadas x dos nós
y=c(:,3); %coordenadas y dos nós
totais=c(:,1); % total de pontos da malha (contorno + internos)
p=0;
cont1=1;

while p<1

D=(modE/(1-(vp^2)))*[1 vp 0; vp 1 0; 0 0 ((1-vp)/2)]; %matriz de
propriedades do material (Estado Plano de Tensão)
nn=size(c,1); % número de nós
nel=size(tri,1); % número de elementos
ngl=nn*2; % número de graus de liberdade

%% 3.1 APLICAÇÃO DAS CONDIÇÕES DE CONTORNO E DAS FORÇAS EXTERNAS

[contor,valor_cont,Fs]=contornoecargaext2(c,ngl,v1,v2,restx,resty,vfp1,vfp2
,F_ext_x,F_ext_y,vertices,tri,esp);

```

```
%% 3.2 CÁLCULO DAS MATRIZES DE RIGIDEZ ELEMENTAR E MONTAGEM DA MATRIZ DE
RIGIDEZ GLOBAL
```

```
KG = kglobal(ngl,nel,tri,c,D,esp);
```

```
%% 3.3 SOLUÇÃO DO SISTEMA DE EQUAÇÕES
```

```
if p_proprio==1
Fe= Forças(c,tri,nel,esp,nn,Xb,Yb); %Cálculo das forças de corpo, quando
forem consideradas na análise
else
    Fe=zeros(ngl,1);
end
```

```
[u,deslx,desly]=solucaomef(contor,valor_cont,ngl,KG,Fe,Fs); %Solução do
sistema de equações (MEF)
```

```
%% 3.4 CÁLCULO DOS GRADIENTES
```

```
[G,Deform,a]=gradiente(u,D,tri,c); %Fornece o gradiente de todos os
elementos e a área
```

```
%% 3.5 MÉTODO DA MÉDIA NODAL SIMPLES
```

```
sigmax=[];
sgmay=[];
gamxy=[];
[erro,Gmed]= mmedianodal(G,D,nn,nel,tri,a);
sigmax=Gmed(1,:);
sgmay=Gmed(2,:);
gamxy=Gmed(3,:);
```

```
%% 3.6 CRITÉRIO DE CONVERGÊNCIA DO PROCESSO DE REFINAMENTO
```

```
[p,Grauref]=criterio_convergencia(erro,nel,nadml,nadm2,a,D,Deform,d_min,Grm
ax);
```

```
%% 3.7 DESNÍVEL UNITÁRIO E REGULARIZAÇÃO DA MALHA
```

```
if p<1
[Gref]=nivelrefnovo(tri,Grauref,nel,c);
end
```

```
%% 3.8 SUBDIVISÃO DOS ELEMENTOS
```

```
if p<1
[c]=subdivide(c,tri,Gref);
end
```

```
%% 3.9 REGENERAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO
```

```
tri=eliminatriangulo(c,xv,yv);
cont1=cont1+1;
```

```
end
```

```
%% 4 PLOTAGEM DOS RESULTADOS
```

```
pos_processamento(c,desly,tri); %deslocamentos em y
pos_processamento(c,deslx,tri); %deslocamentos em x
pos_processamento(c,sigmax,tri); %Tensões em x
pos_processamento(c,sigmay,tri); %Tensões em y
pos_processamento(c,gamxy,tri); %Tensões cisalhantes
```

B.1 APLICAÇÃO DAS CONDIÇÕES DE CONTORNO E DAS FORÇAS EXTERNAS

```
function[contorno1,valor_cont,fs]=contornoecargaext2(c,ngl,v1,v2,restx,rest
y,vfp1,vfp2,F_ext_x,F_ext_y,vertices,tri,esp)
```

```
% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% APLICAÇÃO DAS CONDIÇÕES DE CONTORNO E DAS FORÇAS EXTERNAS

% DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
% esp - espessura (admitida como unitária)
% F_ext_x - vetor que contém as forças externas na direção x
% F_ext_y - vetor que contém as forças externas na direção y
% ngl - número de graus de liberdade
% restx - [0 ou 1], 1 - indica que a condição de contorno aplicada
apresenta restrição de deslocamento em x
% resty - [0 ou 1], 1 - indica que a condição de contorno aplicada
apresenta restrição de deslocamento em y
% tri - matriz de elementos (conectividades)
% v1 - número do vértice (origem) que contém a condição de contorno
(pertencente ao lado)
% v2 - número do vértice (fim) que contém a condição de contorno
(pertencente ao lado)
% vertices - sequência das coordenadas x e y do vértice do polígono

% DADOS DE SAÍDA:
% contorno1 - vetor com as posições das condições de contorno
% valor_cont - vetor com as condições de contorno
% fs - vetor de forças externas

lados=vertices(1); %armazena a quantidade de lados do polígono
n=lados + 1;
g=size(vertices,2);
x1=vertices(2:n); %armazena as coordenadas x do vértice do polígono
y1=vertices(n+1:g); %armazena as coordenadas y do vértice do polígono
ntri=size(tri,1); %número de elementos da malha

cx=[];cy=[];j=[];contorno=[];sc=[];contor=[];vc1=[];

cx=c(:,2); %coordenadas x
cy=c(:,3); %coordenadas y
x=cx;
y=cy;

%% APLICAÇÃO DAS CONDIÇÕES DE CONTORNO

nlcont=size(v1,2); %número de lados
contorno1=[];
```

```

for i=1:nlcont

xv=[x1(v1(i)),x1(v2(i)),x1(v1(i))];
yv=[y1(v1(i)),y1(v2(i)),y1(v1(i))];

cont1=[];
cont2=[];
cont=[];
[in]=inpolygon(cx,cy,xv,yv); %verifica se os pontos estão contidos na
aresta
contor= find(in==1); % in=1 indica que o ponto é de contorno
contorno=contor; %armazena os pontos do contorno
contor=[]; %esvazia a variável contor, que será preenchida com os pontos do
contorno do próximo lado a ser calculado
if restx(i)==1
    cont2=2*contorno-1;
end
if resty(i)==1
    cont1=2*contorno;
end

cont=[cont2; cont1];
contorno1=cat(1,contorno1,cont);
contorno=[];
end

tam=size(contorno1,1);
valor_cont=zeros(tam,1); %para condição de contorno nula, isto é deslx e
desly iguais a zero

%% FORÇAS EXTERNAS

nladosfex=size(vfp1);
nlfext=nladosfex(2);
contor2=[]; contorno2=[];

for i=1:nlfext

xv=[x1(vfp1(i)),x1(vfp2(i)),x1(vfp1(i))];
yv=[y1(vfp1(i)),y1(vfp2(i)),y1(vfp1(i))];

[in]=inpolygon(cx,cy,xv,yv); %verifica se os pontos estão contidos na
aresta
contor2= find(in==1); % in=1 indica que o ponto é de contorno
%contor2= contor2+lados; %corrigindo o "nome" dos pontos, já que retirei os
vértices
contorno2=cat(1,contorno2,contor2); %armazena os pontos do contorno (nós)
contor2=[]; %esvazia a variável contor, que será preenchida com os pontos
do contorno do próximo lado a ser calculado
end

pontos_t=size(contorno2,1);
p_t=pontos_t-1;

fsx=[];
fsy=[];
dist=zeros(1,p_t);
distancia=zeros(1,p_t);

```

```

for i=1:p_t

dist(i)=sqrt((x(contorno2(1))-x(contorno2(i+1)))^2+(y(contorno2(1))-
y(contorno2(i+1)))^2);

end

[b,Or]=sort(dist);
Or=Or+1;
g=size(Or,2);
n=g+1;
guia=zeros(n,1);
guia(1)=contorno2(1);
guia(2:n,1)=contorno2(Or);

cont11=2*guia;
cont22=cont11-1;

distancia(1)= sqrt((x(contorno2(1))-
x(contorno2(Or(1))))^2+(y(contorno2(1))-y(contorno2(Or(1))))^2);
for i=2:p_t

    distancia(i)=sqrt((x(contorno2(Or(i-1))))-
x(contorno2(Or(i))))^2+(y(contorno2(Or(i-1)))-y(contorno2(Or(i))))^2);

end

for i=1:p_t
fsx(i,:)= [F_ext_x(1)*distancia(i)*esp/2;F_ext_x(1)*distancia(i)*esp/2];
%estão nas posições cont2
fsy(i,:)= [F_ext_y(1)*distancia(i)*esp/2;F_ext_y(1)*distancia(i)*esp/2];
%estão nas posições cont1

end
fs_x=[];
fs_x(1)=fsx(1,1);
fs_x(p_t+1)=fsx(p_t,2);
fs_y=[];
fs_y(1)=fsy(1,1);
fs_y(p_t+1)=fsy(p_t,2);

for i=2:p_t
    fs_x(i)= fs_x((i-1),2)+fs_x(i,1);
    fs_y(i)= fs_y((i-1),2)+fs_y(i,1);
end

fs=zeros(1,ngl);
fs(cont22)=fs_x;
fs(cont11)=fs_y;
fs=fs'; %vetor de forças externas

end

```

B.2 CÁLCULO DAS MATRIZES DE RIGIDEZ ELEMENTAR E MONTAGEM DA MATRIZ DE RIGIDEZ GLOBAL

```
function KG = kglobal(ngl,nel,tri,c,D,esp)
```

```

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% MATRIZ DE RIGIDEZ

% DADOS DE ENTRADA:
% c - matriz: [nÓ, coord x, coord y]
% esp - espessura (admitida como unitária)
% D - matriz de propriedades do material
% nel - número de elementos
% ngl - número de graus de liberdade
% tri - matriz de conectividades

% DADO DE SAÍDA:
% KG - matriz de rigidez global (matriz esparsa)

l=[];m=[];v=[];
KG=[]; Ke=[];

% loop nos elementos para cálculo das matrizes de rigidez elementar e
montagem da matriz de rigidez global
for i=1:nel
    Ke=[];
    % nós dos elementos
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);

    % coordenadas
    x1=c(no1,2);
    y1=c(no1,3);

    x2=c(no2,2);
    y2=c(no2,3);

    x3=c(no3,2);
    y3=c(no3,3);

    % área do triângulo
    A=0.5*(x1*y2-x1*y3+x2*y3-x2*y1+x3*y1-x3*y2);

    B=(1/(2*A))*[y2-y3 0 y3-y1 0 y1-y2 0;
                 0 x3-x2 0 x1-x3 0 x2-x1;
                 x3-x2 y2-y3 x1-x3 y3-y1 x2-x1 y1-y2];
    Bt=B';
    Ke=esp*A*Bt*D*B; % matriz de rigidez elementar

    % graus de liberdade gl's (correspondência na global)
    n1=no1*2;
    n2=no2*2;
    n3=no3*2;
    gl=[];
    gl=[n1-1 n1 n2-1 n2 n3-1 n3];
    % loop com introdução de [Ke] em [KG]
    for j=1:6
        for k=1:6

```

```

        linha=gl(j);
        coluna=gl(k);
        valor=Ke(j,k);

        l=[l linha];
        m=[m coluna];
        v=[v valor];
    end
end
end
% matriz de rigidez global
KG=sparse(l,m,v,ngl,ngl);

end

```

B.3 CÁLCULO DAS FORÇAS DE CORPO

```

function [Fe]= Forces(c,tri,nel,esp,nn,Xb,Yb)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DAS FORÇAS DE CORPO

% DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
% esp - espessura (admitida como unitária)
% nel - número de elementos
% nn - número de nós
% tri - matriz de conectividades
% Xb - %densidade de peso na direção x (= densidade * g) (kN/m³)
% Yb - %densidade de peso na direção y (= densidade * g) (kN/m³)

% DADO DE SAÍDA:
% Fe - vetor de forças de corpo

arm=size(tri);
fb=zeros(arm(1),arm(2));
Fe=[];

%cálculo das forças de corpo em cada elemento
for i=1:nel

% nós dos elementos
no1=tri(i,1);
no2=tri(i,2);
no3=tri(i,3);
% coordenadas
cx=[c(no1,2) c(no2,2) c(no3,2)];
cy=[c(no1,3) c(no2,3) c(no3,3)];

A2=cx(1)*(cy(2)-cy(3))+cx(3)*(cy(1)-cy(2))+cx(2)*(cy(3)-cy(1));
A=A2/2;

X= [Xb; Yb; Xb; Yb; Xb; Yb];
fbe(i,:)= (A*esp/3)*X;
end

```

```

%montagem do vetor de forças de corpo (soma das contribuições nodais de
cada
%elemento)
for i=1:nn
    n=0;
    rows=[];
    col=[];
    [rows,col]=find(tri==i);
    n=size(rows,1);
    fb=zeros(1,2);
    for j=1:n
        if col(j)==1
            cl=1; clf=2;
        end
        if col(j)==2
            cl=3; clf=4;
        end
        if col(j)==3
            cl=5; clf=6;
        end
        fb(1,:)=fbe(rows(j),cl:clf)+fb(1,:);
    end
    Fe=cat(2,Fe,fb);
end
Fe=Fe'; %vetor de forças de corpo
end

```

B.4 SOLUÇÃO DO SISTEMA DE EQUAÇÕES - MEF

```

function [u,deslx,desly]=solucaomef(contor,valor_cont,ngl,KG,Fe,Fs)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% SOLUÇÃO MEF

% DADOS DE ENTRADA:
% c - matriz: [nÓ, coord x, coord y]
% contornol - vetor com as posições das condições de contorno
% Fe - vetor de forças de corpo
% Fs - vetor de forças externas
% KG - matriz de rigidez global (matriz esparsa)
% ngl - número de graus de liberdade
% valor_cont - vetor com as condições de contorno

% DADOS DE SAÍDA:
% u - vetor solução incluindo todos os graus de liberdade
% deslx - vetor com os valores dos deslocamentos x
% desly - vetor com os valores dos deslocamnetos y

KGR=KG; % acumula a matriz global antes da eliminação de linhas e
colunas
KGR(:,contor)=[]; % elimina colunas na matriz restringida
KGR(contor,:)=[]; % elimina linhas na matriz restringida
contor;
tamanhoK=size(KGR);
u=zeros(ngl,1); % inicializa um vetor de soluções nulo
u(contor)=valor_cont;% armazena no vetor de soluções as condições de
contorno de Dirichlet

```

```

FP=[];
FP=-KG*u; % vetor de contribuições dos valores prescritos
FP=Fs+FP+Fe; % vetor de força final (fluxo + prescritas)
FP(contor,:)=[];% elimina linhas correspondes ao contorno

sol=KGR\FP; % solução do sistema de equações após restrições

posi=1:1:ngl;
internos=setdiff(posi,contor);% recupera pontos internos

u(internos)=sol; %vetor solução com todos os graus de liberdade
nn=ngl/2;
deslx=zeros(nn,1);
desly=zeros(nn,1);
j=1;
for i=1:nn

    deslx(i)=u(j); %deslocamentos em x
    desly(i)=u(j+1); %deslocamentos em y
    j=j+2;
end
end

```

B.5 CÁLCULO DOS GRADIENTES

```

function [G,Deform,a]=gradiente(u,D,tri,c)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% GRADIENTE

% DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
% D - matriz de propriedades do material
% tri - matriz de conectividades
% u - vetor solução incluindo todos os graus de liberdade

% DADOS DE SAÍDA:
% a - vetor que contém as áreas dos elementos
% Deform - Matriz com as deformações (x, y e cizalhante)
% G - matriz de gradientes

nel=size(tri,1); %número de elementos
G=[];
a=zeros(1,nel);
Deform=[];
% loop para o cálculo do gradiente me cada elemento
for i=1:nel
    %nós dos elementos
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);
    %coordenadas
    x=[c(no1,2) c(no2,2) c(no3,2)];
    y=[c(no1,3) c(no2,3) c(no3,3)];
    A= (x(1)*y(2))-(x(1)*y(3))+(x(2)*y(3))-(x(2)*y(1))+(x(3)*y(1))-(x(3)*y(2));

```

```

B=(1/A)*[(y(2)-y(3)) 0 (y(3)-y(1)) 0 (y(1)-y(2)) 0; 0 (x(3)-x(2)) 0 (x(1)-
x(3)) 0 (x(2)-x(1)); (x(3)-x(2)) (y(2)-y(3)) (x(1)-x(3)) (y(3)-y(1)) (x(2)-
x(1)) (y(1)-y(2))];
a(i)=A/2;
n1=no1*2;
n2=no2*2;
n3=no3*2;
gl=[n1-1 n1 n2-1 n2 n3-1 n3];
sol=[];
sol=u(gl); % vetor coluna com as soluções dos nós do elemento
grad=[]; Es=[];
grad=D*B*sol;
Es=B*sol;
Deform=cat(2,Deform,Es);
G=cat(2,G,grad); % matriz de gradiente

end

```

B.6 MÉTODO MÉDIA NODAL SIMPLES (MNS)

```

function [erro,Gmed]= mmedianodal(G,D,nn,nel,tri,a)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% RECUPERAÇÃO DO CAMPO SUAVIZADO PELO MÉTODO MÉDIA NODAL SIMPLES (MNS)

% DADOS DE ENTRADA:
% a - vetor que contém as áreas dos elementos
% D - matriz de propriedades do material
% G - matriz de gradientes
% nel - número de elementos
% nn - número de nós
% tri - matriz de conectividades

% DADOS DE SAÍDA:
% erro - vetor que contém o erro na norma em energia dos elementos da malha
% Gmed - Gradientes nodais suavizados

erro=[];
somag=zeros(3,1);
Gmed=[];

% Cálculo dos gradientes nodais suavizados
for i=1:nn
    n=0;
    [rows,col]=find(tri==i); %procura pelo nó nas linhas e colunas da
matriz de conectividade dos elementos da malha
    n=size(rows,1);
    for i=1:n
        somag=G(:,rows(i))+somag; %soma dos gradientes dos elementos que contém
o nó
    end
    somag=somag/n; %média nodal
    Gmed=cat(2,Gmed,somag);
    somag=[];
    somag=zeros(3,1);
end

```

```

% Campo suavizado
Gsuav=[];
for i=1:nel
    gsv=[];
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);
    gsv=(Gmed(:,no1)+Gmed(:,no2)+Gmed(:,no3))/3;
    Gsuav=cat(2,Gsuav,gsv);
end

% Estimativa do erro
for i=1:nel
    A=(Gsuav(:,i)- G(:,i));
    At=A';
    cte=At*(inv(D))*A;
    erro(i)=cte*a(i); %vetor que contém o erro na norma em energia dos
    elementos da malha
end
end

```

B.7 CRITÉRIO DE CONVERGÊNCIA E DEFINIÇÃO DO GRAU DE REFINAMENTO DOS ELEMENTOS

```

function[p,Grauref]=criterioconvergencia(erro,nel,nadm1,nadm2,a,D,Deform,d_
min,Grmax)

```

```

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CRITÉRIO DE CONVERGÊNCIA E DEFINIÇÃO DO GRAU DE REFINAMENTO DOS ELEMENTOS

```

```

% DADOS DE ENTRADA:
% a - vetor que contém as áreas dos elementos
% D - matriz de propriedades do material
% Deform - Matriz com as deformações (x, y e cisalhante)
% d_min - dimensão mínima permitida p/ os elementos
% Grmax - grau de refinamento máximo permitido por etapa
% nadam1 - erro relativo percentual admissível global (valor entre 0 e 1)
% nadam2 - erro relativo percentual admissível elementar (valor entre 0 e
1)
% nel - número de elementos

```

```

% DADOS DE SAÍDA:
% p - indica se haverá um novo refinamento ou se a convergência foi
atingida
% Grauref - vetor que contém o grau de refinamento dos elementos

```

```

tgr=[];
p=0; et=0;
umeft=0; T=[]; tref=[]; el=[]; t=[]; aux=[];tr=[];
a_min=d_min^2/2;

```

```

for i=1:nel
    A=Deform(:,i);
    At=A';
    cte=At*D*A;
    umef=cte*a(i); %energia acumulada em um elemento

```

```

umeft=umef+umeft; %energia total acumulada pelo MEF
et=et+erro(i); %erro total na norma em energia
end
energ_u=umeft+et; %energia total acumulada no domínio

normae=(sqrt(et)/sqrt(energ_u));
e_maxelem=nadm2*sqrt(energ_u)/sqrt(nel) %erro máximo permitido p/ os
elementos da malha (critério de equidistribuição do erro)
j=1;
cont=0;

if normae>nadml
    cont=1;
    for i=1:nel
        d0=sqrt(2*a(i));
        d1=d0*e_maxelem/sqrt(erro(i));
        Gr=(log10(d0/d1))/log10(2); %grau de refinamento do elemento
        inteiro=floor(Gr);
        decimal=Gr-inteiro;

        if decimal<0.25 %critério de arredondamento do valor de Gr
            Gr=inteiro;
        else
            Gr=inteiro+1; %critério de arredondamento do valor de Gr
        end

        if Gr>Grmax %limitação do grau de refinamento dos elementos ao grau
maximo permitido por etapa (Grmax)
            Gr=Grmax;
        end
        if Gr<0 %para que não haja desrefinamento
            Gr=0;
        end

        if a(i)<a_min %limitação do tamanho mínimo dos elementos da malha
            Gr=0;
        end

        Grauref(i)=Gr;

    end
end

if cont==0
    p=1;
    Grauref=[];
end
end

```

B.8 DESNÍVEL UNITÁRIO E REGULARIZAÇÃO DA MALHA

```

function [Grauref]=nivelrefnovo(tri,Grauref,nel,c)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% DESNÍVEL UNITÁRIO E REGULARIZAÇÃO DA MALHA

```

```

% DADOS DE ENTRADA:
% c - matriz: [nÓ, coord x, coord y]
% Grauref - vetor que contém o grau de refinamento dos elementos
% nel - número de elementos
% tri - matriz de conectividades

% DADOS DE SAÍDA:
% Grauref - vetor que contém o grau de refinamento dos elementos

x=c(:,2);
y=c(:,3);
nt=triangulation(tri,x,y);
viz=neighbors(nt);
auxunit=0;
auxreg=0;
verif=[];

%Desnível unitário do grau de refinamento dos elementos
for i=1:nel

    vizinhos=viz(i,:);
    vetor=isnan(vizinhos);
    cont1=1;
    for m=1:3 %loop para retirada dos NaN, deixar apenas os triângulos vizinhos
    if vetor(m)==0
    vizelem(cont1)=vizinhos(m);
    cont1=cont1+1;
    end
    end
    cont=0;
    Gref=Grauref(vizelem);
    Graumax=max(Gref);
    if Grauref(i)<(Graumax-1) %verificação do desnível unitário
        Grauref(i)=Graumax-1;
        verifica=vizelem; %salva os vizinhos do elemento que sofreu
        modificação do Grau de refinamento para que sejam verificados novamente.
        verif=cat(2,verif,verifica);
        cont=cont+1;
        auxunit=auxunit+1;
    end

    n1=size(vizelem,2);
    aux=0;
    for k=1:n1 %verificação do critério de regularização da malha
        if Grauref(i)<Gref(k)
            aux=aux+1;
        end
    end
    if aux>1 %aplicação do critério de regularização da malha
        novoGrau=max(Gref);
        Grauref(i)=novoGrau;
        if cont==0
            verifica=vizelem; %salva os vizinhos do elemento que sofreu
            modificação do Grau de refinamento para que sejam verificados novamente.
            verif=cat(2,verif,verifica);
            auxreg=auxreg+1;
        end
    end
end
end

```

```

tam=[];
vizelem=[];
verif=unique(verif)
tam=size(verif,2);
if tam>0 %verificação do desnível unitários nos vizinhos dos elementos que
tiveram seu grau de refinamento alterado
    for k=1:tam
        vizinhos=viz(verif(k),:);
        vetor=isnan(vizinhos);
    end
    cont1=1;
    for m=1:3 %loop para retirada dos NaN, deixar apenas os triângulos vizinhos
        if vetor(m)==0
            vizelem(cont1)=vizinhos(m);
            cont1=cont1+1;
        end
    end
    end
    Gref=Grauref(vizelem);
    Graumax=max(Gref);
    if Grauref(verif(k))<(Graumax-1) %verificação do desnível unitário
        Grauref(verif(k))=Graumax-1;
    end

    n1=size(vizelem,2);
    aux=0;
    for l=1:n1 %verificação do critério de regularização da malha
        if Grauref(verif(k))<Gref(l)
            aux=aux+1;
        end
    end
    if aux>1
        novoGrau=max(Gref); %aplicação do critério de regularização da
malha
        Grauref(verif(k))=novoGrau;
    end
end
end
end
end

```

B.9 SUBDIVISÃO DOS ELEMENTOS

```

function [c,tri]=subdivide(c,tri,Gref)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% SUBDIVISÃO DOS ELEMENTOS

% DADOS DE ENTRADA:
% c - matriz: [nÓ, coord x, coord y]
% Gref - vetor que contém o grau de refinamento dos elementos
% tri - matriz de conectividades

% DADOS DE SAÍDA:
% c - vetor que contém o grau de refinamento dos elementos
% tri - matriz de conectividades

ntri=size(tri,1); % number of elements
nn=size(c,1); % number of nodes
x=c(:,2); %coordenadas x

```

```

y=c(:,3); %coordenadas y
j=1;
vet=[];
pt=[];
pmed=[];
pontomed=[];
cmed=[];
aux_x1=zeros(1,2);aux_x2=zeros(1,2);aux_x3=zeros(1,2);
aux_y1=zeros(1,2);aux_y2=zeros(1,2);aux_y3=zeros(1,2);

%loop para subdivisão
elem_filhos1=zeros(8,3);

for w=1:ntri
    Gr=Gref(w);
    if Gr>0
        %Coordenadas do triangulo original
        pt=tri(w,:);
        x1= [x(pt(1)) x(pt(2)) x(pt(3))];
        y1= [y(pt(1)) y(pt(2)) y(pt(3))];
        x1(4)=x1(1);
        y1(4)=y1(1);

        %Primeira subdivisão (Gr=1)

        for i=1:3
            cmed(i,1)=mean([x1(i) x1(i+1)]);
            cmed(i,2)=mean([y1(i) y1(i+1)]);

            if i==3
                cmed(i,3)=1;
                cmed(i,4)=3;
            else
                cmed(i,3)=i;
                cmed(i,4)=i+1;
            end
        end
        %armazenado as coordenadas dos novos nós
        cd=cmed(:,1:2);
        pontomed=cat(1,pontomed,cd);
        j1=1;j2=1; j3=1;

        %gerando uma matriz de conectividades entre os elementos filhos
        if Gr>1
            elem_filhos=zeros(8,3);
            for i=1:3
                if cmed(i,3)==1 || cmed(i,4)==1
                    aux_x1(j1)=cd(i,1);
                    aux_y1(j1)=cd(i,2);
                    j1=j1+1;
                end
                if cmed(i,3)==2 || cmed(i,4)==2
                    aux_x2(j2)=cd(i,1);
                    aux_y2(j2)=cd(i,2);
                    j2=j2+1;
                end
                if cmed(i,3)==3 || cmed(i,4)==3
                    aux_x3(j3)=cd(i,1);
                    aux_y3(j3)=cd(i,2);
                    j3=j3+1;
                end
            end
        end
    end
end

```

```

end
end
    cd=cd';
elem_filhos(1,:)=[x1(1) aux_x1];
elem_filhos(2,:)=[y1(1) aux_y1];
elem_filhos(3,:)=[x1(2) aux_x2];
elem_filhos(4,:)=[y1(2) aux_y2];
elem_filhos(5,:)=[x1(3) aux_x3];
elem_filhos(6,:)=[y1(3) aux_y3];
elem_filhos(7,:)=cd(1,:);
elem_filhos(8,:)=cd(2,:);
end

j1=1;j2=1;j3=1;
cont=1; cd=[];
%aquí começa as subdivisões a partir de grau 2
if Gr>1
    for m=1:4
        coordx=elem_filhos(cont,:);
        coordy=elem_filhos(cont+1,:);
        coordx(4)=elem_filhos(cont,1);
        coordy(4)=elem_filhos(cont+1,1);
        cont=cont+2;
        cmed=[];
        for j=1:3
            cmed(j,1)=mean([coordx(j) coordx(j+1)]);
            cmed(j,2)=mean([coordy(j) coordy(j+1)]);
            if j==3
                cmed(j,3)=1;
                cmed(j,4)=3;
            else
                cmed(j,3)=j;
                cmed(j,4)=j+1;
            end
        end
    end

    %armazenamento das novas coordenadas
    cd=cmed(:,1:2);
    pontomed=cat(1,pontomed,cd);
    j1=1;j2=1; j3=1;
    if Gr==3
        for i=1:3
            if cmed(i,3)==1 || cmed(i,4)==1
                aux_x1(j1)=cd(i,1);
                aux_y1(j1)=cd(i,2);
                j1=j1+1;
            end
            if cmed(i,3)==2 || cmed(i,4)==2
                aux_x2(j2)=cd(i,1);
                aux_y2(j2)=cd(i,2);
                j2=j2+1;
            end
            if cmed(i,3)==3 || cmed(i,4)==3
                aux_x3(j3)=cd(i,1);
                aux_y3(j3)=cd(i,2);
                j3=j3+1;
            end
        end
    end
    cd=cd';
    elem_filhos1(1,:)=[coordx(1) aux_x1];

```

```

        elem_filhos1(2,:)=[coordx(1) aux_y1];
        elem_filhos1(3,:)=[coordx(2) aux_x2];
        elem_filhos1(4,:)=[coordx(2) aux_y2];
        elem_filhos1(5,:)=[coordx(3) aux_x3];
        elem_filhos1(6,:)=[coordx(3) aux_y3];
        elem_filhos1(7,:)=cd(1,:);
        elem_filhos1(8,:)=cd(2,:);
        elem_filhos=cat(2,elem_filhos,elem_filhos1);
    end
end

end

k=4;
cmed=[];
%subdivisões para elementos de grau 3
if Gr==3
    for l=1:4
        elem_f=elem_filhos(:,k:k+2);
        k=k+3;
        cont=1;
        for i=1:4
            coordx=elem_f(cont,:);
            coordy=elem_f(cont+1,:);
            coordx(4)=elem_f(cont,1);
            coordy(4)=elem_f(cont+1,1);
            cont=cont+2;
            for j=1:3
                cmed(j,1)=mean([coordx(j) coordx(j+1)]);
                cmed(j,2)=mean([coordy(j) coordy(j+1)]);
            end
            cd=cmed;
            pontomed=cat(1,pontomed,cd);
        end
    end
end

end

end

pontoresul=unique(pontomed,'rows');
d=size(pontoresul,1);
pmed=zeros(d,3);
contador=(nn+1):(nn+d);
pmed(:,2)=pontoresul(:,1);
pmed(:,3)=pontoresul(:,2);
pmed(:,1)=contador;
%novas coordenadas
c=cat(1,c,pmed);
x=c(:,2);
y=c(:,3);
end

```

B. 10 TRIANGULARIZAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO

```

function tri=eliminatriangulo(c,xv,yv)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE

```

```

% AUTOR: MYRELLA VIEIRA CABRAL
% TRIANGULARIZAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO

% DADOS DE ENTRADA:
% c - matriz: [nÓ, coord x, coord y]
% xv - coordenadas x dos nós do contorno
% yv - coordenadas y dos nós do contorno

%DADO DE SAÍDA:
% tri - matriz de conectividades

x=c(:,2);      % valores de x nodais
y=c(:,3);      % valores de y nodais

npt=size(c,1); % número total de nós
tri=delaunay(x,y); % triangularização
ntri=size(tri,1); % número total de triângulos

% loop para avaliação do cg dos triângulos
for i=1:ntri
    pontos=tri(i,:);
    coordx=c(pontos,2);
    coordy=c(pontos,3);
    xcg(i,1)=mean(coordx);
    ycg(i,1)=mean(coordy);
end

cg=[xcg ycg];
% verifica se o cg dos triângulos é interno ao polígono
in=inpolygon(xcg,ycg,xv,yv);

% triângulos inválidos
indices=find(in==0);
% remove triângulos da triangularização inicial
tri(indices,:)=[];

end

```

APÊNDICE C – CÓDIGO PARA O REFINAMENTO H-ADAPTATIVO APLICADO A PROBLEMAS DESCRITOS PELA EQUAÇÃO DE POISSON – *PoissonG*

Observação: As funções *criterioconvergencia.m*, *nivelrefnovo.m* e *subdivide.m* foram apresentadas no APÊNDICE B.

```
% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% REFINAMENTO H-ADAPTATIVO APLICADO A PROBLEMAS DESCRITOS PELA EQUAÇÃO DE
% POISSON

%% Inicialização
clc
clear all
close all
format long

%% 1. DADOS DE ENTRADA

%1.1 DEFINIÇÃO DO DOMÍNIO:

%Para um círculo, deve ser atribuído o valor 1 à primeira linha, a segunda e
a terceira linha devem conter as coordenadas x e y do centro, respectivamente.
A quarta linha deve conter o raio do círculo.
%Para um polígono, deve ser atribuído o valor 2 à primeira linha, a segunda
linha deve conter o número, n, de segmentos de linha do contorno do polígono,
isto é, o número de arestas. As próximas n linhas seguintes devem conter as
coordenadas x dos vértices. Na sequência, próximas n linhas, devem ser
inseridas as coordenadas y dos vértices.
% Para um retângulo, deve ser atribuído o valor 3. E as demais linhas possuem
o mesmo formato do polígono. Neste caso, n é igual a 4.
% Para uma elipse, deve ser atribuído o valor 4 à primeira linha. A segunda
e a terceira linhas devem conter as coordenadas x e y do centro,
respectivamente. A quarta e a quinta linhas devem conter os semi eixos da
elipse. O ângulo rotacional (em radianos) da elipse deve ser armazenado na
sexta linha.

R1 = [3,4,0,1,1,0,1,1,0,0]'; %exemplo de um domínio retangular
vertices= R1;
vertices(1)=[];

%1.2 DADOS DO MATERIAL

E=200*10^3; %módulo de elasticidade (p/ o problema de flexão de placas)
te=0.2; %espessura (p/ o problema de flexão de placas)
vp=0.3; %coeficiente de poisson (p/o problema de flexão de placas)
k=1;% coef presente na Eq. de Poisson ( p/ temperatura corresponde ao
coeficiente de condutividade térmica do material)

%1.3 DADOS P/ O REFINAMENTO
hmax=0.5; %dimensão máxima dos elementos da malha inicial
Grmax=2; %grau de refinamento máximo permitido por etapa
d_min=10^-4; %usuário determina a dimensão mínima do elemento (isto evita o
refinamento excessivo em singularidades)
```

```
nadm1=0.04; %erro relativo percentual admissível global (valor entre 0 e 1)
nadm2=0.05; %erro relativo percentual admissível elementar (valor entre 0 e 1)
```

%1.4 FUNÇÃO QUE DESCREVE O PROBLEMA

```
tn=0; %indica se há solução real ou não: 1- contorno descrito por uma
função; 0- contorno descrito por um vetor;
```

```
prompt= 'Qual o valor da fonte interna? ';
Q=input(prompt);
```

```
% As condições de contorno podem ser descritas por uma função, um escalar
% ou um vetor
%f=@(x,y) [100+2*(1+y)/(x^2+(1+y)^2)]; %Inserir entre os parênteses a
%função que descreve as condições de contorno
f=0; % Para contorno descrito por um valor escalar, basta igualar f a este
valor. Como neste exemplo, onde f=0.
%f=[100,0,0,0] % Para contorno descrito por um vetor, isto é, quando cada
%lado apresenta valores escalares diferentes. Colocar a sequência de
%valores aos lados correspondentes
vert=[0,0,0,0]; % condições de contorno dos vértices, apenas para o caso
que o contorno é descrito por um vetor
```

% 1.5 PERGUNTA AO USUÁRIO QUAL O PROBLEMA QUE DESEJA ANALISAR (FLEXÃO DE PLACAS, TORÇÃO OU TEMPERATURA)

```
prompt = 'Problema de: 1- flexão de placas, 2-torção ou 3-temperatura ';
snf=input(prompt);
```

%% 2. GERAÇÃO DA MALHA INICIAL

```
model = createpde;
```

```
gm = [R1];
sf = 'R1';
```

```
ns = char('R1');
ns = ns';
g = decsg(gm,sf,ns);
```

```
pg=geometryFromEdges(model,g)
figure
pdegplot(model,'EdgeLabels','on')
axis equal
xlim([-1.1,1.1])
[p,e,t] = initmesh(g,'hmax',hmax);
```

```
c=[];
x=p(1,:); %coordenada x de todos os nós
y=p(2,:); %coordenada y de todos os nós
x=x'; %transpondo para vetor coluna
y=y'; %transpondo para vetor coluna
```

```
m=size(p,2); %cada coluna da matriz p contém um nó e a numeração
corresponde com a da condição de contorno
totais=1:1:m;
c(:,1)=totais';
```

```

c(:,2)=x;
c(:,3)=y;
x=c(:,2); %coordenadas x dos nós
y=c(:,3); %coordenadas y dos nós

%% 3. OPERAÇÕES GERAIS

t(4,:)=[];
tri=t'
p=0;
cont=1;
K=k*[1 0; 0 1];

while p<1

nn=size(c,1);      % número de nós
nel=size(tri,1);    % número de elementos
ngl=nn*1;           % número de graus de liberdade

%% 3.1 APLICAÇÃO DAS CONDIÇÕES DE CONTORNO

[contorno,sc]=contornoedge2(c,f,vert,tn,vertices,tri) %retorna o vetor com
os nós de contorno e o vetor "sc" com as condições de contorno de cada nó
do contorno

%% 3.2 CÁLCULO DAS MATRIZES DE RIGIDEZ ELEMENTAR E MONTAGEM DA MATRIZ DE
RIGIDEZ GLOBAL

KG = kglobal(ngl,nel,tri,c);

%% 3.3 CÁLCULO DO VETOR DE FLUXO

Fe=Fonte_interna(Q,c,tri,nel,nn); %Fluxo de calor interno

%% 3.4 SOLUÇÃO DO SISTEMA DE EQUAÇÕES - PROBLEMA DE VALOR PRESCRITO

u=solucaoPoisson(contorno,sc,c,ngl,k,KG,Fe); %Vetor de soluções incluindo
todos os graus de liberdade

%% 3.5 RESOLUÇÃO DA SEGUNDA EQ. DIFERENCIAL - DESLOCAMENTOS (APENAS PARA O
PROBLEMA DE FLEXÃO DE PLACAS)

if snf==1
D=(E*te^3)/(12*(1-vp^2)); %rigidez à flexão da placa
We=deslocamentos(u,D,c,tri,nel,nn);
u2=solucaodesl(contorno,sc,c,ngl,KG,We);
end

%% 3.6 CÁLCULO DOS GRADIENTES

[G,a]=gradiente(u,tri,c); %Fornece o gradiente de todos os elementos e a
área

%% 3.7 MÉTODO DA MÉDIA NODAL SIMPLES

[erro,et,Gsuav,Gmed]= mmedianodal(G,k,nn,nel,tri,a);

```

```

%% 3.8 CÁLCULO DA RIGIDEZ TORCIONAL

if snf==2
[T]=ftorca(u,tri,c);
torque(cont)=T;
end

%% 3.9 CRITÉRIO DE CONVERGÊNCIA DO PROCESSO DE REFINAMENTO

[p,Grauref]=criterioconvergencia(erro,nel,nadm1,nadm2,a,K,G,d_min,Grmax);

%% 3.10 DESNÍVEL UNITÁRIO E REGULARIZAÇÃO DA MALHA

if p<1
[Gref]=nivelrefnovo(tri,Grauref,nel,c);
end

%% 3.11 SUBDIVISÃO DOS ELEMENTOS

if p<1
[c]=subdivide(c,tri,Gref);
end
t=[];

%% 3.12 REGENERAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO

[tri]=eliminatric(vertices);
cont=cont+1;
end

%% 4. PLOTAGEM DOS RESULTADOS
if snf==1
u=u2;
end
if snf==2
X = sprintf('A rigidez torcional é %d',T);
disp(X)
end
pos_processamento(c,u,tri);

```

C.1 APLICAÇÃO DAS CONDIÇÕES DE CONTORNO

```

function [contorno,sc]=contornoedge2(c,f,vert,tn,vertices,tri)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% APLICAÇÃO DAS CONDIÇÕES DE CONTORNO

%DADOS DE ENTRADA:
%c - matriz de coordenadas dos nós da malha
%f - vetor ou função que define as condições de contorno
%vert - vetor com os valores correspondentes aos vértices do polígono(apenas
%deve ser preenchido quando o contorno é definido por um vetor de
%escalares)
%tn - indica se as condições de contorno são definidas por uma função ou
%por um vetor ( tn=0 -> f é um vetor ou um escalar; tn=1 -> f é um função)
%vertices - [número de lados, coord x dos vértices, coord y dos vértices]

```

```

%tri - matriz de conectividades

%DADOS DE SAÍDA:
%contorno - nós do contorno
%sc - valores das condições de contorno

lados=vertices(1); %armazena a quantidade de lados do polígono
n=lados + 1;
g=size(vertices,1);
x1=vertices(2:n); %armazena as coordenadas x do vértice do polígono
y1=vertices(n+1:g); %armazena as coordenadas y do vértice do polígono
ntri=size(tri,1); %número de elementos da malha

%inicializa algumas variáveis
cx=[];cy=[];j=[];contorno=[];sc=[];contor=[];vcl=[];

lb=size(f,2); % tamanho de f (função ou vetor de contorno)
cx=c(:,2); %coordenadas x
cy=c(:,3); %coordenadas y
cx(1:lados)=[]; %retira as coordenadas x dos vértices
cy(1:lados)=[]; %retira as coordenadas y dos vértices

%loop para o cálculo das condições de contorno: Verifica-se para todas as
%arestas se todos os pontos pertencem a eles, exceto os do vértice, que
%serão inseridos posteriormente.
for i=1:lados

if i==lados
    xv=[x1(i),x1(1),x1(i)];
    yv=[y1(i),y1(1),y1(i)];
else
    xv=[x1(i),x1(i+1),x1(i)];
    yv=[y1(i),y1(i+1),y1(i)];
end

[in]=inpolygon(cx,cy,xv,yv); %verifica se os pontos estão contidos na
aresta
contor= find(in==1); % in=1 indica que o ponto é de contorno
contorno=cat(1,contorno,contor); %armazena os pontos do contorno

k=size(contor,1);
v=[];
if tn==0 && lb>1 %Caso em que o contorno é definido por um vetor, onde
cada lado possui um valor escalar diferente
v(1:1:k)=f(i); %atribui a cada ponto do contorno seu valor correspondente
sc=cat(2,sc,v); %armazena em um vetor os valores correspondentes aos pontos
de contorno
end
contor=[]; %esvazia a variável contor, que será preenchida com os pontos do
contorno do próximo lado a ser calculado
end

contorno=contorno+lados; %corrigindo o "nome" dos pontos, já que retirei os
vértices
cc=1:1:lados; %vértices
cc=cc';
contorno=cat(1,contorno,cc); %armazenando os vértices no vetor de contorno

if lb>1

```

```

sc=cat(2,sc,vert); %armazenando os valores correspondentes aos vértices no
vetor de contorno
end

tam=size(contorno,1); %tamando do vetor que contém os pontos de contorno
if lb==1 %tamanho do vetor f
    if tn==0 %tn=0 indica que o contorno é definido por um vetor e não por
uma função
        sc(1:1:tam)=f; %valores do contorno
    else %quaundo os valores dos pontos do contorno são defindos por uma
função
        for i=1:tam
            sc(i)=f(c(contorno(i),2),c(contorno(i),3)) %valores do contorno
        end
    end
end
contorno=contorno'; %transformando em vetor linha

end

```

C.2 CÁLCULO DAS MATRIZES DE RIGIDEZ ELEMENTAR E MONTAGEM DA MATRIZ DE RIGIDEZ GLOBAL

```

function KG = kglobal(ngl,nel,tri,c)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DAS MATRIZES DE RIGIDEZ ELEMENTAR E MONTAGEM DA MATRIZ DE RIGIDEZ
% GLOBAL

%DADOS DE ENTRADA:
%c - matriz de coordenadas dos nós da malha
%nel - número de elementos
%ngl - número de graus de liberdade
%tri - matriz de conectividades

%DADO DE SAÍDA:
%KG - matriz de rigidez global (esparsa)

l=[];m=[];v=[];
KG=[];

for i=1:nel

    % nós do elemento
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);

    % coordenadas
    x1=c(no1,2);
    y1=c(no1,3);

    x2=c(no2,2);

```

```

y2=c(no2,3);

x3=c(no3,2);
y3=c(no3,3);

% area of the triangle(A)

xx=[x1 x2 x3];
yy=[y1 y2 y3];

A=polyarea(xx,yy);

% matriz de rigidez elementar

k11=(1/(4*A))*(x3-x2)^2+(y2-y3)^2;
k12=(1/(4*A))*((x3-x2)*(x1-x3)+(y2-y3)*(y3-y1));
k13=(1/(4*A))*((x3-x2)*(x2-x1)+(y2-y3)*(y1-y2));
k21=k12;
k22=(1/(4*A))*((x1-x3)^2+(y3-y1)^2);
k23=(1/(4*A))*((x1-x3)*(x2-x1)+(y3-y1)*(y1-y2));
k31=k13;
k32=k23;
k33=(1/(4*A))*((x2-x1)^2+(y1-y2)^2);

Ke=[k11 k12 k13; k21 k22 k23; k31 k32 k33];

% graus de liberdade (relacionados à matriz de rigidez global)

gl(1)=no1;
gl(2)=no2;
gl(3)=no3;

% loop para inserção de [Ke] em [KG]

for j=1:3
    for k=1:3

        linha=gl(j);
        coluna=gl(k);
        valor=Ke(j,k);
        l=[l linha];
        m=[m coluna];
        v=[v valor];

    end
end

end

% Matriz de rigidez global (esparsa)
KG=sparse(l,m,v,nl,nl);

end

```

C.3 CÁLCULO DO VETOR DE FLUXO

```
function [Fe]= Fonte_interna(Q,c,tri,nel,nn)
```

```

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DO VETOR DE FLUXOS

%DADOS DE ENTRADA:
%c - matriz de coordenadas dos nós da malha
%Q - fonte interna
%nel - número de elementos
%nn - número de nós
%tri - matriz de conectividades

% DADO DE SAÍDA:
% Fe - vetor de fluxos

syms x y
w=1/6;
Fe=[];
r=[];
E=[0 1/2 1/2;1/2 0 1/2;1/2 1/2 0];
for i=1:nel

% nós do elemento
no1=tri(i,1);
no2=tri(i,2);
no3=tri(i,3);
% coordenadas
cx=[c(no1,2) c(no2,2) c(no3,2)];
cy=[c(no1,3) c(no2,3) c(no3,3)];

f1=0;f2=0;f3=0;

% integração numérica
for j=1:3

xt=cx(1)*E(j,1)+cx(2)*E(j,2)+cx(3)*E(j,3);
yt=cy(1)*E(j,1)+cy(2)*E(j,2)+cy(3)*E(j,3);
g=subs(Q,x,xt);
f=subs(g,y,yt);
A2=cx(1)*(cy(2)-cy(3))+cx(3)*(cy(1)-cy(2))+cx(2)*(cy(3)-cy(1));
% funções de forma do elemento triangular linear
cte=(1/(A2));
H1=cte*((cx(2)*cy(3))-(cx(3)*cy(2))+(cy(2)-cy(3))*xt+(cx(3)-cx(2))*yt);
H2=cte*((cx(3)*cy(1))-(cx(1)*cy(3))+(cy(3)-cy(1))*xt+(cx(1)-cx(3))*yt);
H3=cte*((cx(1)*cy(2))-(cx(2)*cy(1))+(cy(1)-cy(2))*xt+(cx(2)-cx(1))*yt);

% funções de integração
f1=H1*f+f1;
f2=H2*f+f2;
f3=H3*f+f3;
end
ff1=A2*w*f1;
ff2=A2*w*f2;
ff3=A2*w*f3;
r(i,:)=[ff1 ff2 ff3];
end

%armazenamento do vetor de fluxos
for i=1:nn

```

```

n=0;
rows=[];
col=[];
[rows,col]=find(tri==i); %procura pelo nó nas linhas e colunas da
matriz de conectividades
n=size(rows,1);
fe=0;
for j=1:n
    fe=r(rows(j),col(j))+fe; %computa o fluxo no nó pela contribuição
de todos os elementos que contém este nó
end
Fe(i)=fe; %vetor de fluxos
end
end

```

C.4 SOLUÇÃO DO SISTEMA DE EQUAÇÕES

```
function [Fe]= Fonte_interna(Q,c,tri,nel,nn)
```

```

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DO VETOR DE FLUXOS

```

```

%DADOS DE ENTRADA:
%c - matriz de coordenadas dos nós da malha
%Q - fonte interna
%nel - número de elementos
%nn - número de nós
%tri - matriz de conectividades

```

```

% DADO DE SAÍDA:
% Fe - vetor de fluxos

```

```

syms x y
w=1/6;
Fe=[];
r=[];
E=[0 1/2 1/2;1/2 0 1/2;1/2 1/2 0];
for i=1:nel

```

```

% nós do elemento
no1=tri(i,1);
no2=tri(i,2);
no3=tri(i,3);
% coordenadas
cx=[c(no1,2) c(no2,2) c(no3,2)];
cy=[c(no1,3) c(no2,3) c(no3,3)];

```

```
f1=0;f2=0;f3=0;
```

```

% integração numérica
for j=1:3

```

```

xt=cx(1)*E(j,1)+cx(2)*E(j,2)+cx(3)*E(j,3);
yt=cy(1)*E(j,1)+cy(2)*E(j,2)+cy(3)*E(j,3);
g=subs(Q,x,xt);
f=subs(g,y,yt);
A2=cx(1)*(cy(2)-cy(3))+cx(3)*(cy(1)-cy(2))+cx(2)*(cy(3)-cy(1));

```

```

% funções de forma do elemento triangular linear
cte=(1/(A2));
H1=cte*((cx(2)*cy(3))-(cx(3)*cy(2))+(cy(2)-cy(3))*xt+(cx(3)-cx(2))*yt);
H2=cte*((cx(3)*cy(1))-(cx(1)*cy(3))+(cy(3)-cy(1))*xt+(cx(1)-cx(3))*yt);
H3=cte*((cx(1)*cy(2))-(cx(2)*cy(1))+(cy(1)-cy(2))*xt+(cx(2)-cx(1))*yt);

% funções de integração
f1=H1*f+f1;
f2=H2*f+f2;
f3=H3*f+f3;
end
ff1=A2*w*f1;
ff2=A2*w*f2;
ff3=A2*w*f3;
r(i,:)=[ff1 ff2 ff3];
end

%armazenamento do vetor de fluxos
for i=1:nn
    n=0;
    rows=[];
    col=[];
    [rows,col]=find(tri==i); %procura pelo nó nas linhas e colunas da
matriz de conectividades
    n=size(rows,1);
    fe=0;
    for j=1:n
        fe=r(rows(j),col(j))+fe; %computa o fluxo no nó pela contribuição
de todos os elementos que contém este nó
    end
    Fe(i)=fe; %vetor de fluxos
End
end

```

C.5 RESOLUÇÃO DA SEGUNDA EQ. DIFERENCIAL - DESLOCAMENTOS (APENAS PARA O PROBLEMA DE FLEXÃO DE PLACAS)

C.5.1 Cálculo da fonte interna para a segunda Eq. diferencial

```

function [Fe]= deslocamentos(M,D,c,tri,nel,nn)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DA FONTE INTERNA P/SEGUNDA EQ. DIFERENCIAL (PROBLEMA DE FLEXÃO DE
PLACAS)

%DADOS DE ENTRADA:
%c - matriz: [nó, coord x, coord y]
%D - rigidez à flexão da placa
%nel - número de elementos
%nn - número de nós
%tri - matriz de conectividades

%DADO DE SAÍDA:
%Fe - vetor de fluxos

syms x y

```

```

w=1/6;
Fe=[];
r=[];

E=[0 1/2 1/2;1/2 0 1/2;1/2 1/2 0];
for i=1:nel

    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);
    cx=[c(no1,2) c(no2,2) c(no3,2)]; %coordenadas x dos nós do elemento
    cy=[c(no1,3) c(no2,3) c(no3,3)]; %coordenadas y dos nós do elemento

    f1=0;
    f2=0;
    f3=0;
    m=mean([M(no1) M(no2) M(no3)]);
    Q=m/D;
    for j=1:3
        xt=cx(1)*E(j,1)+cx(2)*E(j,2)+cx(3)*E(j,3);
        yt=cy(1)*E(j,1)+cy(2)*E(j,2)+cy(3)*E(j,3);
        A2=cx(1)*(cy(2)-cy(3))+cx(3)*(cy(1)-cy(2))+cx(2)*(cy(3)-cy(1));

        %funções de forma do elemento CST
        cte=(1/(A2));
        H1=cte*((cx(2)*cy(3))-(cx(3)*cy(2))+(cy(2)-cy(3))*xt+(cx(3)-cx(2))*yt);
        H2=cte*((cx(3)*cy(1))-(cx(1)*cy(3))+(cy(3)-cy(1))*xt+(cx(1)-cx(3))*yt);
        H3=cte*((cx(1)*cy(2))-(cx(2)*cy(1))+(cy(1)-cy(2))*xt+(cx(2)-cx(1))*yt);

        %funções de integração
        f1=H1*Q+f1;
        f2=H2*Q+f2;
        f3=H3*Q+f3;
    end
    ff1=A2*w*f1;
    ff2=A2*w*f2;
    ff3=A2*w*f3;
    r(i,:)=[ff1 ff2 ff3];
end
for i=1:nn
    n=0;
    rows=[];
    col=[];
    [rows,col]=find(tri==i); %procura na matriz de triângulos da malha as
    linhas e colunas que contém o nó em questão
    n=size(rows,1); %quantidade de elementos que contém o nó em questão
    fe=0;
    for j=1:n
        fe=r(rows(j),col(j))+fe; %cálculo do fluxo em cada elemento
    end
    Fe(i)=fe; %vetor de fluxos
end
end

```

C.5.2 Solução do sistema de equações – determinação dos deslocamentos

```
function u=solucadoesl(contorno,desl,c,ngl,KG,Fe)
```

```
% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
```

```

% AUTOR: MYRELLA VIEIRA CABRAL
% SOLUÇÃO DO SISTEMA DE EQUAÇÕES

%DADOS DE ENTRADA:
%c - matriz de coordenadas dos nós da malha
%contorno - nós dos contorno
%desl - valores das condições de contorno
%Fe - vetor de fluxos
%KG - matriz de rigidez global (esparsa)
%ngl - número de graus de liberdade

%DADO DE SAÍDA:
% u - vetor solução incluindo todos os graus de liberdade

KGR=KG;      % acumula a matriz global antes da eliminação de linhas e
colunas
KGR(:,contorno)=[]; % elimina colunas na matriz restringida
KGR(contorno,:)=[]; % elimina linhas na matriz restringida

F=zeros(ngl,1); % inicializa um vetor de forças nulo (fluxo nulo)
u=zeros(ngl,1); % inicializa um vetor de soluções nulo (temperaturas)
u(contorno)=desl;% armazena no vetor de soluções as condições de contorno
de Dirichlet
d=size(u);

FP=[];
Fe=Fe';
FP=-KG*u; % vetor de contribuições de temperaturas prescritas
FP=F+FP+Fe; % vetor de força final (fluxo + prescritas)
FP(contorno,:)=[];% elimina linhas

% solução do sistema de equações após restrições
sol=KGR\FP;
% recupera pontos internos
internos=setdiff(c(:,1),contorno);
% atribui os valores de soluções ao vetor de temperaturas prescritas
u(internos)=sol;
end

```

C.6 CÁLCULO DOS GRADIENTES

```

function [G,a]=gradiente(u,tri,c)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DOS GRADIENTES

%DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
% tri - matriz de conectividades
% u - vetor solução incluindo todos os graus de liberdade

%OUTPUT DATA:
% G - Matriz de gradiente
% a - vetor que contém as áreas dos elementos

nel=size(tri,1); %número de elementos
G=[];

```

```

a=[];

for i=1:nel
%nós dos elementos
no1=tri(i,1);
no2=tri(i,2);
no3=tri(i,3);
%coordenadas
x=[c(no1,2) c(no2,2) c(no3,2)];
y=[c(no1,3) c(no2,3) c(no3,3)];

sol=[u(no1);u(no2);u(no3)]; %vetor coluna com as soluções dos nodais
A2=x(1)*(y(2)-y(3))+x(3)*(y(1)-y(2))+x(2)*(y(3)-y(1));
A=A2/2;
a(i)=A;

%derivadas das funções de forma em relação à x
d1x=(1/(2*A))*(y(2)-y(3));
d2x=(1/(2*A))*(y(3)-y(1));
d3x=(1/(2*A))*(y(1)-y(2));

%derivadas das funções de forma em relação à y
d1y=(1/(2*A))*(x(3)-x(2));
d2y=(1/(2*A))*(x(1)-x(3));
d3y=(1/(2*A))*(x(2)-x(1));

deriv=[d1x d2x d3x; d1y d2y d3y]; %matriz das derivadas parciais das
funções de forma dos elementos
grad=deriv*sol; %gradientes do elemento
G=cat(2,G,grad); %matriz de gradientes
End

```

C.7 MÉTODO DA MÉDIA NODAL SIMPLES (MNS)

```

function [erro,et,Gsuav,Gmed]= mmedianodal(G,K,nn,nel,tri,a)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% RECUPERAÇÃO DO CAMPO SUAVIZADO PELO MÉTODO MÉDIA NODAL SIMPLES (MNS)

% DADOS DE ENTRADA:
% a - vetor que contém as áreas dos elementos
% k - coef relacionado à Eq. de Poisson
% G - matriz de gradientes
% nel - número de elementos
% nn - número de nós
% tri - matriz de conectividades

% DADOS DE SAÍDA:
% erro - vetor que contém o erro na norma em energia dos elementos da malha
% Gmed - Gradientes nodais suavizados

erro=[];
somag=zeros(2,1);
Gmed=[];
et=0;
% Cálculo dos gradientes nodais suavizados

```

```

for i=1:nn
    n=0;
    [rows,col]=find(tri==i); %procura pelo nó nas linhas e colunas da
matriz de conectividade dos elementos da malha
    n=size(rows,1);
    for i=1:n
        somag=G(:,rows(i))+somag; %soma dos gradientes dos elementos que contém
o nó
    end
    somag=somag/n; %média nodal
    Gmed=cat(2,Gmed,somag);
    somag=zeros(2,1);
end

% campo suavizado
Gsuav=[];
for i=1:nel
    gsv=[];
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);
    gsv=(Gmed(:,no1)+Gmed(:,no2)+Gmed(:,no3))/3;
    Gsuav=cat(2,Gsuav,gsv);

end
% Estimativa do erro
for i=1:nel
    A=(Gsuav(:,i)- G(:,i));
    At=A';
    cte=At*inv(K)*A;
    erro(i)=cte*a(i); %vetor que contém o erro na norma em energia dos
elementos da malha
    et=erro(i)+et;
end

end

```

C. 8 CÁLCULO DA RIGIDEZ TORCIONAL

```

function [T]= ftorcao(u,tri,c)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% CÁLCULO DA RIGIDEZ TORCIONAL

%DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
% tri - matriz de conectividades
% u - vetor solução incluindo todos os graus de liberdade

%DADO DE SAÍDA:
% T - rigidez torcional

n=size(tri,1);
T=0;
x=c(:,2);
y=c(:,3);
torq=[];
i=1

```

```

for i=1:n
    phi=[];
    no1=tri(i,1);
    no2=tri(i,2);
    no3=tri(i,3);
    xt=[c(no1,2) c(no2,2) c(no3,2)]; %coordenadas x dos nós do elemento
    yt=[c(no1,3) c(no2,3) c(no3,3)]; %coordenadas y dos nós do elemento
    B=ones(3);
    B(:,2)=xt;
    B(:,3)=yt;
    A=(1/2)*det(B); %área do triângulo
    phi=mean([u(no1) u(no2) u(no3)]);
    torq(i)=-2*phi*A;
    T=torq(i)+T;
end
end

```

C.9 REGENERAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO

```

function [tri]=eliminatric(c,vertices)

% DEPARTAMENTO DE ENGENHARIA CIVIL - UFPE
% AUTOR: MYRELLA VIEIRA CABRAL
% REGENERAÇÃO DA MALHA E ELIMINAÇÃO DE TRIÂNGULOS FORA DO DOMÍNIO

%DADOS DE ENTRADA:
% c - matriz: [nó, coord x, coord y]
%vertices - [número de lados, coord x dos vértices, coord y dos vértices]
%DADO DE SAÍDA
% tri - nova matriz de conectividades

x=c(:,2); % valores de x nodais
y=c(:,3); % valores de y nodais
tri=delaunay(x,y); % triangularização

lados=vertices(1); %armazena a quantidade de lados do polígono
n=lados + 1;
g=size(vertices,1);
x1=vertices(2:n); %armazena as coordenadas x do vértice do polígono
y1=vertices(n+1:g); %armazena as coordenadas y do vértice do polígono

x11=wrev(x1);
x11=x11';
xv=[x11 x11(1)];
y11=wrev(y1);
y11=y11';
yv=[y11 y11(1)];

ntri=size(tri,1);
x=c(:,2);
y=c(:,3);
% loop para avaliação do cg dos triângulos
for i=1:ntri
    pontos=tri(i,:);
    cx=c(pontos,2);
    cy=c(pontos,3);

```

```

        xcg(i,1)=mean(cx);
        ycg(i,1)=mean(cy);
    end

    cg=[xcg ycg];
    % verifica se o cg dos triângulos é interno ao polígono
    in=inpolygon(xcg,ycg,xv,yv);

    % triângulos inválidos
    indices=find(in==0);
    % remove triângulos da triangularização inicial
    tri(indices,:)=[];

end

```