



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

CICERO VITOR CHAVES JUNIOR

**RESOLUÇÃO DE PROBLEMAS DE ORDEM SUPERIOR EM PLACAS PELO
MÉTODO DAS DIFERENÇAS FINITAS GENERALIZADAS**

**RECIFE
2018**

CICERO VITOR CHAVES JUNIOR

**RESOLUÇÃO DE PROBLEMAS DE ORDEM SUPERIOR EM PLACAS PELO
MÉTODO DAS DIFERENÇAS FINITAS GENERALIZADAS**

Monografia apresentada à Universidade Federal de Pernambuco para obtenção do título 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

C512r Chaves Junior, Cicero Vitor.
Resolução de problemas de ordem superior em placas pelo método das diferenças finitas generalizadas / Cicero Vitor Chaves Junior – 2018.
51 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. Diferenças finitas generalizadas. 3. Placas.
4. Lagrange. 5. Poisson. 6. Série de Taylor. I. Ribeiro, Paulo Marcelo
Vieira. (Orientador). II. Título.

624 CDD (22. ed.)

UFPE
BCTG/2018-76



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: Cicero Vitor Chaves Junior

BANCA EXAMINADORA:

Orientador: Paulo Marcelo Vieira Ribeiro

Examinador 1: José Jeferson do Rêgo Silva

Examinador 2: Augusto César Albuquerque Ferreira

TÍTULO DO TRABALHO DE CONCLUSÃO DE CURSO: RESOLUÇÃO DE PROBLEMAS DE ORDEM SUPERIOR EM PLACAS PELO MÉTODO DAS DIFERENÇAS FINITAS GENERALIZADAS.

LOCAL: CTG – UFPE, Sala .

DATA: 08/02/2018 **HORÁRIO DE INÍCIO:** .

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: _____ (deixar 'Exame Final', quando for o caso).

1) () aprovado(s) (nota $\geq 7,0$), 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 pontuar revisões).

O trabalho com nota no seguinte intervalo, **$3,0 \leq \text{nota} < 7,0$** , será reapresentado, gerando-se uma nota 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, 08 de Fevereiro de 2018

Orientador:

Avaliador 1:

Avaliador 2:

Candidato:

Dedico este trabalho aos meus pais Cicero e Cirlene, demais familiares e amigos pela paciência e ajuda ao longo desta graduação.

AGRADECIMENTOS

A meus pais Cicero Vitor Chaves e Cirlene Martins de Menezes Chaves que proveveram meu sustento durante todo o período do curso e me apoiaram sempre.

A minha irmã Camila Martins de Menezes Chaves por estar sempre comigo.

Ao professor Paulo Marcelo por ter me orientado no TCC e ter me proposto este tema.

A todos os amigos que me acompanharam no curso incentivando e tornando o tempo da faculdade sempre bem-humorado.

RESUMO

Neste trabalho é apresentado o método das diferenças finitas generalizadas (MDFG) como uma alternativa para a resolução de equações diferenciais parciais (EDPs), mais propriamente dito a equação de Germain Lagrange, e é verificada sua consistência e convergência em alguns casos. Este método consiste na aproximação das derivadas infinitesimais parciais pela expansão em série de Taylor. É utilizado um artifício para a resolução da equação de Germain Lagrange dividindo-a em duas equações de Poisson. Através do MATLAB é criado um código computacional para a resolução da equação de Germain Lagrange utilizando como condições de contorno o bordo simplesmente apoiado ou engastado, a discretização do domínio é feita pelo GiD. É demonstrado através de alguns exemplos a convergência do método e sua aplicabilidade para problemas em placas de domínios irregulares e compara-se a solução dada pelo método com a solução obtida através do método das diferenças finitas (MDF) e através do método dos elementos finitos (MEF).

Palavras-chave: Diferenças finitas generalizadas. Placas. Lagrange. Poisson. Série de Taylor.

ABSTRACT

In this work the generalized finite difference method (GFDM) is presented as an alternative for the resolution of partial differential equations (PDEs), more precisely the Germain Lagrange equation, and its consistency and convergence are verified in some cases. This method consists of the approximation of the partial infinitesimal derivatives by the Taylor series expansion. An artifice is used to solve the Germain Lagrange equation by dividing it into two Poisson equations. Through MATLAB a computational code is created for the resolution of the Germain Lagrange equation using as boundary conditions the simply supported or set edge, the discretization of the domain is done by GiD. It is demonstrated by some examples the convergence of the method and its applicability to problems in plates of irregular domains and it compares the solution given by the method with the solution obtained through the finite difference method (FDM) and through the finite element method (FEM).

Keywords: Generalized finite differences. Plates. Lagrange. Poisson. Taylor series.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Esforços no plano médio do elemento de placa..... | 15 |
| Figura 2 - Arranjo dos 12 pontos capturados pelo pivô | 18 |
| Figura 3 - Arranjo dos 4 pontos capturados pelo pivô | 19 |
| Figura 4 - Exemplo de malha gerada pelo GiD | 25 |
| Figura 5 - Código de espelhamento de pontos no MATLAB..... | 25 |
| Figura 6 - Trecho do código da montagem da matriz A para os pontos internos..... | 26 |
| Figura 7 - Trecho do código da montagem da matriz A para os pontos de contorno..... | 26 |
| Figura 8 - Trecho do código da montagem da matriz AA e vetor b..... | 27 |
| Figura 9 - Malha gerada pelo GiD para o MDFG | 28 |
| Figura 10 – Deslocamentos (m) pelo MDFG (todos os bordos simplesmente apoiados) | 29 |
| Figura 11 – Deslocamentos (m) pelo MDF – vista lateral (todos os bordos simplesmente apoiados)..... | 29 |
| Figura 12 – Deslocamentos (m) pelo MDFG (todos os bordos engastados)..... | 30 |
| Figura 13 – Deslocamentos (m) pelo MDF – vista lateral (todos os bordos engastados) | 30 |
| Figura 14 – Malha de 0,06 m..... | 32 |
| Figura 15 – Deslocamentos (m) para a malha de 0,06 m | 32 |
| Figura 16 – Momentos (kNm/m) na direção X para a malha de 0,06 m..... | 33 |
| Figura 17 – Momentos (kNm/m) na direção Y para a malha de 0,06 m..... | 33 |
| Figura 18 – Seção transversal ($y = 0,75\text{m}$) para a malha de 0,06 m..... | 34 |
| Figura 19 – Gráfico: erro X distância mínima entre os pontos da malha..... | 34 |
| Figura 20 – Malha de 0,05m..... | 35 |
| Figura 21 – Deslocamentos (m) com a malha de 0,05m | 36 |
| Figura 22 – Malha de 0,1m..... | 36 |
| Figura 23 – Deslocamentos (m) com a malha de 0,1m | 37 |
| Figura 24 – Seção transversal ($y = 0,0\text{ m}$) para a malha | 37 |
| Figura 25 – Malha de 0,2m (MDFG) | 38 |
| Figura 26 – Malha de 0,2m (MEF)..... | 38 |
| Figura 27 – Deslocamentos (m) pelo MDFG | 39 |
| Figura 28 – Deslocamentos (m) pelo MEF | 39 |
| Figura 29 – Malha de 0,09m gerada para o MDFG..... | 40 |

| | |
|---|----|
| Figura 30 - Malha de 0,1m gerada para o MEF..... | 41 |
| Figura 31 – Deslocamentos (m) de placa U com bordos internos engastados e o resto simplesmente apoiados (MDFG)..... | 41 |
| Figura 32 - Deslocamentos (m) de placa U com bordos internos engastados e o resto simplesmente apoiados (MEF) | 41 |
| Figura 33 – Deslocamentos (m) de placa U com os bordos simplesmente apoiados (MDFG) | 42 |
| Figura 34 – Deslocamentos (m) de placa U com os bordos simplesmente apoiados (MEF)... | 42 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Erro entre o MDFG e o MDF | 31 |
| Tabela 2 – Resultados obtidos pelo MDFG para o exemplo..... | 31 |
| Tabela 3 – Comparação dos deslocamentos entre o MDFG e MEF no ponto $P(-0,78;1,49)$... | 40 |

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO..... | 12 |
| 1.1 | Justificativa e motivação..... | 12 |
| 1.2 | Objetivos gerais e específicos..... | 13 |
| 2 | REFERENCIAL TEÓRICO..... | 14 |
| 2.1 | Apresentação do problema de placa..... | 14 |
| 2.2 | Método das diferenças finitas..... | 16 |
| 2.3 | Método das diferenças finitas generalizadas..... | 19 |
| 2.4 | Estratégia para a resolução do problema de placa..... | 20 |
| 2.5 | Condições de contorno..... | 22 |
| 3 | ASPECTOS COMPUTACIONAIS..... | 24 |
| 3.1 | Programas utilizados..... | 24 |
| 3.2 | Pré-processamento..... | 24 |
| 3.3 | Desenvolvimento do método..... | 25 |
| 4 | APLICAÇÕES E RESULTADOS..... | 28 |
| 4.1 | MDFG e MDF..... | 28 |
| 4.2 | MDFG e Solução Exata..... | 31 |
| 4.3 | Placa circular..... | 35 |
| 4.4 | MDFG e MEF (poligonal qualquer) | 38 |
| 4.5 | MDFG e MEF (placa em U) | 40 |
| 5 | CONCLUSÃO..... | 43 |
| | REFERÊNCIAS..... | 44 |
| | ANEXO A..... | 45 |
| | ANEXO B..... | 50 |
| | ANEXO C..... | 51 |

1 INTRODUÇÃO

Muitos fenômenos na engenharia podem ser descritos por meio de equações diferenciais parciais (EDPs). Resolver essas equações por métodos analíticos para geometrias quaisquer é muito complexo. O método dos elementos finitos (MEF) é o método mais usado para resolver esses tipos de problemas de forma aproximada, porém em problemas com domínios complexos esse método gera muito custo computacional na parte do pré-processamento. Outra opção seria o método das diferenças finitas (MDF), o empecilho deste é que quando se trabalha em duas dimensões se deve usar malhas regulares e isto se torna um problema quando se deseja trabalhar com contornos irregulares. Por isso uma boa opção é utilizar uma extensão do MDF que é o método das diferenças finitas generalizadas (MDFG), que por não se trabalhar com malhas estruturadas não possui um problema com geometrias complexas no contorno. Assim como o MDF, o MDFG aproxima as derivadas infinitesimais com a expansão em série de Taylor, o que traz a vantagem de se ter uma estimativa para o erro.

A desvantagem deste método é o trabalho que se tem para localizar os pontos. Como a malha não é estruturada, não há uma relação inicial entre os pontos então deve-se criar uma rotina para o cálculo das distâncias entre os pontos.

A discretização do domínio é feita por um programa de pré-processamento para evitar problemas de singularidade. Perrone e Kao (1975) mostraram vários problemas de singularidade ocasionados no MDFG e a posição dos pontos do domínio que os causavam, indicando pra sua solução uma estratégia de captura de pontos vizinhos. Neste trabalho é usado apenas a estratégia de se pegar os pontos mais próximos confiando ao GiD, programa de pré-processamento, a disposição dos pontos.

1.1 Justificativa e motivação

As equações diferenciais parciais de ordem elevada se adequam a maior parte dos problemas de engenharia civil. Comumente para a resolução desses problemas é usado o método dos elementos finitos, mas este método causa um grande desgaste computacional, dependendo do domínio do problema, devido ao pré-processamento. O método das diferenças finitas generalizadas, que é uma extensão do método das diferenças finitas, não possui esse empecilho por não possuir malha estruturada.

1.2 Objetivos gerais e específicos

Este trabalho com o uso do MDFG, desenvolve códigos no MATLAB que busquem boa eficiência para a resolução da equação diferencial parcial de Lagrange com diferentes formas de domínio, diferentes condições de contorno.

Como objetivos específicos, pode-se listar:

- Desenvolver código para a resolução de EDPs de ordem elevada com diferentes condições de contorno utilizando o MDFG;
- Comparar os resultados com soluções exatas, com o MEF e o com MDF;
- Analisar convergência e o bom funcionamento do método.

2 REFERENCIAL TEÓRICO

2.1 Apresentação do problema de placa

Seja uma placa em que as cargas que atuam sobre ela são normais a sua superfície, e que as deformações verticais são pequenas em relação a sua espessura. Tendo o plano médio de um elemento desta placa como está representado na Figura 1 e que existe uma carga uniforme q que atua sobre a placa, temos as seguintes equações de equilíbrio (TIMOSHENKO & WOINOWSKY-KRIEGER, 1970):

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} + q = 0; \quad (1)$$

$$\frac{\partial M_{xy}}{\partial x} - \frac{\partial M_y}{\partial y} + Q_y = 0; \quad (2)$$

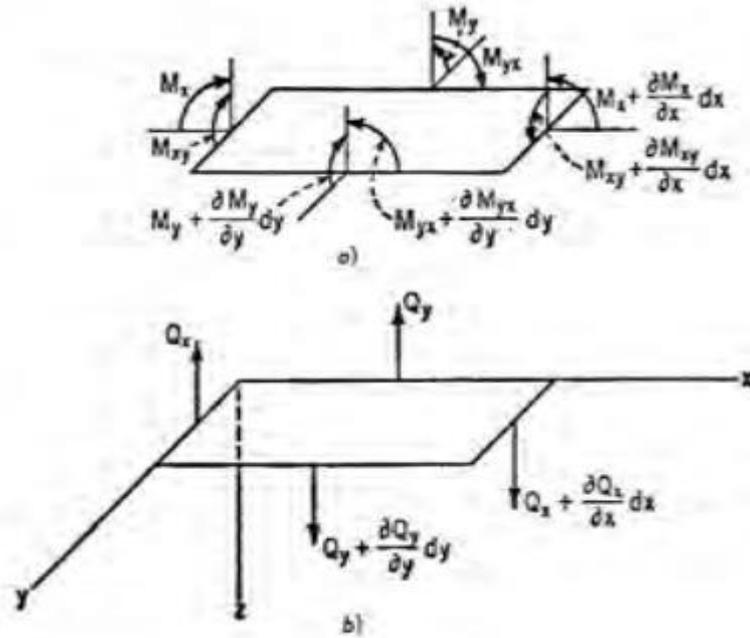
$$\frac{\partial M_{yx}}{\partial y} + \frac{\partial M_x}{\partial x} - Q_x = 0. \quad (3)$$

Sendo q a carga transversal uniforme aplicada sobre a placa, Q_x e Q_y os esforços cortantes nas direções x e y respectivamente, M_x e M_y os momentos fletores nas direções x e y respectivamente, e M_{xy} e M_{yx} os momentos torsores.

Reorganizando as Equações (2) e (3) e substituindo na equação (1) e sabendo que $M_{xy} = -M_{yx}$, ficamos com:

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_y}{\partial y^2} - 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} = -q. \quad (4)$$

Figura 1 - Esforços no plano médio do elemento de placa



Fonte: Timoshenko & Woinowsky-Krieger (1970)

Tendo em vista que pra placas de espessura muito pequena em relação a suas outras duas dimensões, desprezar os efeitos dos esforços cortantes sobre a flexão causam pequenos erros na obtenção das deformações verticais. Temos para flexão pura:

$$M_x = -D \left(\frac{\partial^2 \omega}{\partial x^2} + \nu \frac{\partial^2 \omega}{\partial y^2} \right); \quad (5)$$

$$M_y = -D \left(\frac{\partial^2 \omega}{\partial y^2} + \nu \frac{\partial^2 \omega}{\partial x^2} \right); \quad (6)$$

$$M_{xy} = D(1 - \nu) \frac{\partial^2 \omega}{\partial x \partial y}; \quad (7)$$

Substituindo as Equações (5), (6) e (7) na Equação (4) e reorganizando, ficamos com:

$$\frac{\partial^4 \omega}{\partial x^4} + 2 \frac{\partial^4 \omega}{\partial x^2 \partial y^2} + \frac{\partial^4 \omega}{\partial y^4} = \frac{q}{D}; \quad (8)$$

$$D = \frac{Eh^3}{12(1-\nu^2)}. \quad (9)$$

Onde:

ω – deformação vertical;

q – carga uniforme aplicada sobre a placa;

D – rigidez da placa;

E – módulo de elasticidade;

h – espessura da placa;

ν – coeficiente de Poisson.

Vamos supor agora que os bordos da placa sejam paralelos aos eixos x e y , e vamos tomar o bordo paralelo ao eixo y . De acordo com Timoshenko e Woinowsky-Krieger (1970) podemos ter tais condições de bordo:

Se este bordo for engastado as deformações verticais ao longo do bordo são nulas e a placa não poderá girar livremente sobre a linha do bordo, ou seja, as condições de contorno são:

$$\omega = 0; \frac{\partial \omega}{\partial x} = 0.$$

Agora se o bordo for simplesmente apoiado a deformação vertical ao longo do bordo continuará sendo nula e agora ela poderá girar livremente sobre a linha do bordo, mas os momentos M_x serão nulos neste bordo:

$$\omega = 0; M_x = 0.$$

Se o bordo for considerado livre ele deformará verticalmente e girará ao redor do bordo, porém os momentos M_x e os esforços cortantes Q_x serão nulos:

$$Q_x = 0; M_x = 0.$$

2.2 Método das diferenças finitas

O método das diferenças finitas consiste em aproximar uma equação diferencial em uma equação de diferenças finitas, ou seja, aproximar as derivadas infinitesimais por diferenças finitas discretas.

As fórmulas de diferenças finitas progressiva, regressiva e central, bem como muitas outras fórmulas usadas para calcular derivadas de forma aproximada, podem ser deduzidas a partir da expansão em série de Taylor. Essas fórmulas fornecem uma estimativa da derivada em um ponto usando valores de pontos em sua vizinhança. O número de pontos usados nos cálculos varia com a fórmula, e os pontos podem estar à frente, atrás ou em ambos os lados do ponto

onde se calcula a derivada. Uma vantagem do uso da expansão em série de Taylor na dedução das fórmulas está no fato de que ela também fornece uma estimativa do erro de truncamento presente na aproximação (GILAT & SUBRAMANIAM, 2008).

Seja $f(x, y)$ uma função de classe C^4 no domínio e h o espaçamento em relação ao eixo x entre dois pontos, temos que a aproximação pelo polinômio de Taylor em uma direção será:

$$f(x + h, y) = f(x, y) + \frac{\partial f}{\partial x}(x, y)h + \frac{\partial^2 f}{\partial x^2}(x, y)\frac{h^2}{2} + \frac{\partial^3 f}{\partial x^3}(x, y)\frac{h^3}{6} + \frac{\partial^4 f}{\partial x^4}(x, y)\frac{h^4}{24}; \quad (15)$$

$$f(x - h, y) = f(x, y) - \frac{\partial f}{\partial x}(x, y)h + \frac{\partial^2 f}{\partial x^2}(x, y)\frac{h^2}{2} - \frac{\partial^3 f}{\partial x^3}(x, y)\frac{h^3}{6} + \frac{\partial^4 f}{\partial x^4}(x, y)\frac{h^4}{24}; \quad (16)$$

O último termo de cada equação é o resíduo onde ξ_1 é um valor entre $x+h$ e x , e ξ_2 um valor entre x e $x-h$. Somando as Equações (15) e (16) e reorganizando obteremos a aproximação para a segunda derivada em relação a x :

$$\frac{\partial^2 f}{\partial x^2}(x, y) = \frac{f(x+h, y) - 2f(x, y) + f(x-h, y)}{h^2} + o(h^2); \quad (17)$$

Que é a diferença finita central de segunda ordem com três pontos, onde $o(h^2)$ é o erro de truncamento que vale $\frac{\partial^4 f}{\partial x^4}(\xi, y)\frac{h^2}{12}$, com ξ sendo um valor entre $x+h$ e $x-h$. Analogamente em relação ao eixo y , onde k é o espaçamento no eixo y entre os pontos:

$$\frac{\partial^2 f}{\partial y^2}(x, y) = \frac{f(x, y+k) - 2f(x, y) + f(x, y-k)}{k^2} + o(h^2); \quad (18)$$

Para as derivadas de quarta ordem em uma direção temos:

$$\frac{\partial^4 f}{\partial x^4}(x, y) \approx \frac{f(x+2h, y) - 4f(x+h, y) + 6f(x, y) - 4f(x-h, y) + f(x-2h, y)}{h^4}; \quad (19)$$

$$\frac{\partial^4 f}{\partial y^4}(x, y) \approx \frac{f(x, y+2k) - 4f(x, y+k) + 6f(x, y) - 4f(x, y-k) + f(x, y-2k)}{k^4}; \quad (20)$$

E para as derivadas cruzadas de ordem quatro:

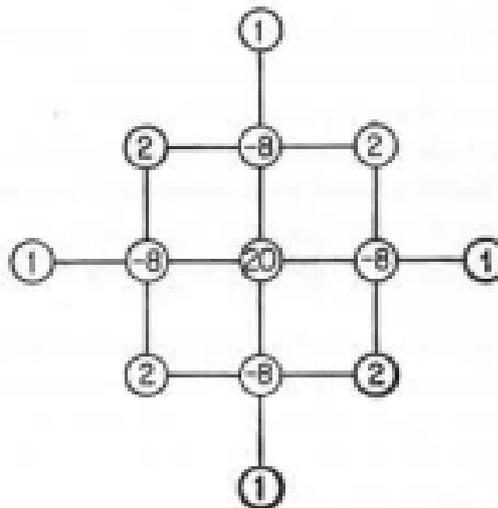
$$\frac{\partial^4 f}{\partial x^2 \partial y^2}(x, y) \approx \frac{4f(x, y) - 2(f(x+h, y) + f(x-h, y) + f(x, y+k) + f(x, y-k)) + f(x+h, y+k) + f(x-h, y-k) + f(x+h, y-k) + f(x-h, y+k)}{h^2 k^2}; \quad (21)$$

Fazendo $h = k = l$, aplicando as diferenças finitas à Equação (8) de Lagrange (problema de placa) e utilizando as notações i e j para o ponto central, temos (TIMOSHENKO & WOINOWSKY-KRIEGER, 1970):

$$20w_{i,j} - 8(w_{i-1,j} + w_{i+1,j} + w_{i,j-1} + w_{i,j+1}) + 2(w_{i+1,j+1} + w_{i+1,j-1} + w_{i-1,j+1} + w_{i-1,j-1}) + w_{i+2,j} + w_{i-2,j} + w_{i,j+2} + w_{i,j-2} = \frac{q_z}{D} l^4; \quad (22)$$

Ou seja, cada ponto pivô busca outros 12 pontos ao seu redor, o arranjo deve ficar como o ilustrado na Figura 3. Por se trabalhar com uma malha ortogonal, o método tem grande desvantagem ao se trabalhar com domínios irregulares. Pois como a malha pode não satisfazer a geometria requerida isso pode acarretar em erros associados a discretização do domínio. Quanto mais irregular for o contorno maior será o erro. Devido a esse problema com o MDF iremos trabalhar com o MDFG.

Figura 2 - Arranjo dos 12 pontos capturados pelo pivô



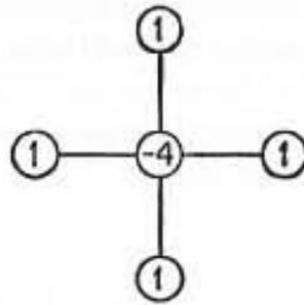
Fonte: Timoshenko & Woinowsky-Krieger (1970)

Para a equação de Laplace a resolução por diferenças finitas seria:

$$T_{i,j} = \frac{T_{i+1,j} + T_{i,j+1} + T_{i-1,j} + T_{i,j-1}}{4}; \quad (23)$$

Com cada ponto pivô utilizando os valores de quatro vizinhos como ilustra a Figura 4.

Figura 3 - Arranjo dos 4 pontos capturados pelo pivô



Fonte: Timoshenko & Woinowsky-Krieger (1970)

2.3 Método das diferenças finitas generalizadas

Do mesmo modo que o MDF o MDFG aproxima o valor das derivadas a partir da série de Taylor, só que em vez de possuir grids regulares ele trabalha com grids irregulares e os pontos desse grid possuem espaçamentos variados. Dada a expansão de Taylor de quarta ordem:

$$\begin{aligned} f(x+h, y+k) \approx & f(x, y) + \frac{\partial f}{\partial x}(x, y)h + \frac{\partial f}{\partial y}(x, y)k + \frac{\partial^2 f}{\partial x^2}(x, y)\frac{h^2}{2} + \frac{\partial^2 f}{\partial x \partial y}(x, y)hk + \\ & \frac{\partial^2 f}{\partial y^2}(x, y)\frac{k^2}{2} + \frac{\partial^3 f}{\partial x^3}(x, y)\frac{h^3}{6} + \frac{\partial^3 f}{\partial x^2 \partial y}(x, y)\frac{h^2 k}{2} + \frac{\partial^3 f}{\partial x \partial y^2}(x, y)\frac{h k^2}{2} + \frac{\partial^3 f}{\partial y^3}(x, y)\frac{k^3}{6} + \\ & \frac{\partial^4 f}{\partial x^4}(x, y)\frac{h^4}{24} + \frac{\partial^4 f}{\partial x^3 \partial y}(x, y)\frac{h^3 k}{6} + \frac{\partial^4 f}{\partial x^2 \partial y^2}(x, y)\frac{h^2 k^2}{4} + \frac{\partial^4 f}{\partial x \partial y^3}(x, y)\frac{h k^3}{6} + \frac{\partial^4 f}{\partial y^4}(x, y)\frac{k^4}{24}; \quad (24) \end{aligned}$$

A partir desta expansão podemos aproximar as derivadas de um ponto com valores de 14 vizinhos. Colocando em notação matricial, sendo i o número da linha, cada linha da matriz A_1 fica da forma:

$$\left[h_i \quad k_i \quad \frac{h_i^2}{2} \quad h_i k_i \quad \frac{k_i^2}{2} \quad \frac{h_i^3}{6} \quad \frac{h_i^2 k_i}{2} \quad \frac{h_i k_i^2}{2} \quad \frac{k_i^3}{6} \quad \frac{h_i^4}{24} \quad \frac{h_i^3 k_i}{6} \quad \frac{h_i^2 k_i^2}{2} \quad \frac{h_i k_i^3}{6} \quad \frac{k_i^4}{24} \right]$$

Cada linha do vetor B recebe restante do termo designado a cada coluna da matriz acima, e cada linha do vetor com os valores da função nos pontos vizinhos F_i recebe o seguinte termo:

$$f(x + h_i, y + k_i)$$

E cada linha do vetor com os valores da função no ponto central F_0 recebe:

$$f(x, y)$$

Desta forma a equação matricial fica:

$$A_1 B = F_i - F_0; \quad (25)$$

Com isto temos que:

$$B = A_1^{-1} F_i - A_1^{-1} F_0; \quad (26)$$

A ideia principal do método é substituir a derivada que está na linha i do vetor B pela linha i do segundo termo da equação acima. Se este processo for realizado para cada nó do domínio, um sistema de equações lineares é obtido, onde as incógnitas são os valores F_i . Ao resolver este sistema de equações, os valores aproximados da função nos nós do domínio são obtidos (Benito et al., 2001).

2.4 Estratégia para a resolução do problema de placa

Na a resolução de um problema de placa de quarta ordem pelo MDFG é necessário que cada ponto pivô pegue 14 vizinhos em sua célula, e quando se trabalha com muitos pontos podem aparecer vários problemas de condicionamento por haverem pontos alinhados, ou muito distantes do pivô, ou com pontos pivô que acabem pegando os mesmos pontos como vizinhos. Para eliminar este mal condicionamento pode-se resolver esse problema de quarta ordem

separando-o em dois problemas de segunda ordem. Segundo Ugural (2010) a Equação (8) pode ser substituída por:

$$\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2} = -q; \quad (27)$$

$$\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} = -\frac{M}{D}; \quad (28)$$

Sendo:

$$M = \frac{M_x + M_y}{(1+\nu)} = -D \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \quad (29)$$

Ou seja, o ponto pivô agora precisa pegar para sua célula apenas 5 pontos vizinhos. E desta forma colocando na equação (26) para este caso temos:

$$\begin{bmatrix} \frac{\partial f_0}{\partial x} \\ \frac{\partial f_0}{\partial y} \\ \frac{\partial^2 f_0}{\partial x^2} \\ \frac{\partial^2 f_0}{\partial x \partial y} \\ \frac{\partial^2 f_0}{\partial y^2} \end{bmatrix} = \begin{bmatrix} h_1 & k_1 & \frac{h_1^2}{2} & \frac{h_1 k_1}{1} & \frac{k_1^2}{2} \\ h_2 & k_2 & \frac{h_2^2}{2} & \frac{h_2 k_2}{1} & \frac{k_2^2}{2} \\ h_3 & k_3 & \frac{h_3^2}{2} & \frac{h_3 k_3}{1} & \frac{k_3^2}{2} \\ h_4 & k_4 & \frac{h_4^2}{2} & \frac{h_4 k_4}{1} & \frac{k_4^2}{2} \\ h_5 & k_5 & \frac{h_5^2}{2} & \frac{h_5 k_5}{1} & \frac{k_5^2}{2} \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} - \begin{bmatrix} h_1 & k_1 & \frac{h_1^2}{2} & \frac{h_1 k_1}{1} & \frac{k_1^2}{2} \\ h_2 & k_2 & \frac{h_2^2}{2} & \frac{h_2 k_2}{1} & \frac{k_2^2}{2} \\ h_3 & k_3 & \frac{h_3^2}{2} & \frac{h_3 k_3}{1} & \frac{k_3^2}{2} \\ h_4 & k_4 & \frac{h_4^2}{2} & \frac{h_4 k_4}{1} & \frac{k_4^2}{2} \\ h_5 & k_5 & \frac{h_5^2}{2} & \frac{h_5 k_5}{1} & \frac{k_5^2}{2} \end{bmatrix}^{-1} \begin{bmatrix} f_0 \\ f_0 \\ f_0 \\ f_0 \\ f_0 \end{bmatrix}; \quad (30)$$

Substituindo as linhas 3 e 5 desse sistema nas equações (27) e (28) formamos dois outros sistemas que segundo Ugural (2010) podem ser resolvidos separadamente para o caso em que os bordos sejam todos simplesmente apoiados e simultaneamente quando os bordos tiverem quaisquer condições de apoio.

2.5 Condições de contorno

Para condições de contorno de Dirichlet, ou seja, quando é dado o valor da função no ponto de contorno se faz uma simples substituição de dos valores conhecidos. Porém quando o valor conhecido é da derivada e não da função, condições de contorno de Neumann, é necessário a criação de pontos virtuais fora da placa. No caso do bordo simplesmente apoiado temos apenas condições de contorno de Dirichlet nas duas equações. Mas para o caso de bordo engastado temos condições de Dirichlet e de Neumann.

As condições para um bordo engastado paralelo ao eixo y são: $\frac{\partial \omega}{\partial x} = 0$; $\omega = 0$. Utilizando a fórmula de diferenças finitas para diferença central com dois pontos teremos:

$$\frac{\partial \omega(x_i, y)}{\partial x} = \frac{\omega(x_{i+1}, y) - \omega(x_{i-1}, y)}{x_{i+1} - x_{i-1}} = 0; \quad (31)$$

Reorganizando temos:

$$\omega(x_{i+1}, y) = \omega(x_{i-1}, y); \quad (32)$$

Mesmo quando o bordo for simplesmente apoiado são usados pontos virtuais. Neste caso teremos como condições de contorno: $\omega = 0$; $M_x = -D \left(\frac{\partial^2 \omega}{\partial x^2} + \nu \frac{\partial^2 \omega}{\partial y^2} \right) = 0$. Como o bordo é paralelo ao eixo y e temos que $\frac{\partial^2 \omega}{\partial y^2} = 0$, logo, $\frac{\partial^2 \omega}{\partial x^2} = 0$. Usando a fórmula de diferenças finitas para diferença central com três pontos teremos:

$$\frac{\partial^2 \omega(x_i, y)}{\partial x^2} = \frac{\omega(x_{i+1}, y) - 2\omega(x_i, y) + \omega(x_{i-1}, y)}{(x_{i+1} - x_{i-1})^2}; \quad (33)$$

Reorganizando temos:

$$\omega(x_{i+1}, y) = -\omega(x_{i-1}, y); \quad (34)$$

Ou seja, em um bordo engastado um ponto virtual terá o mesmo deslocamento que um ponto interno que em relação ao bordo seja um reflexo do ponto virtual e em um bordo

simplesmente apoiado um ponto virtual terá o deslocamento contrário do ponto interno que é o seu reflexo em relação ao bordo.

Quando o pivô é um ponto do contorno ele pega como vizinhos pontos internos da placa e pontos virtuais externos a placa, como artifício foi usado pontos espelhados a cada bordo como pontos virtuais. Segundo Timoshenko e Woinowsky-Krieger (1970): se o bordo for engastado o ponto virtual terá o mesmo deslocamento que o ponto interno do qual for espelhado, e se for simplesmente apoiado seu deslocamento será igual ao ponto do ponto que foi espelhado só que com o sentido inverso.

3 ASPECTOS COMPUTACIONAIS

3.1 Programas utilizados

O código computacional foi desenvolvido no MATLAB (que abrevia MATRix LABoratory – Laboratório de matrizes), pois é um programa de computador otimizado para implementar cálculos científicos e de engenharia e possui uma biblioteca de funções que torna mais fácil a programação, incluindo suas ferramentas gráficas que facilitam a visualização dos resultados.

A malha não estruturada de entrada no código desenvolvido no MATLAB é gerada pelo programa GiD que é um pré e pós processador universal, fácil de usar para simulações numéricas.

3.2 Pré-processamento

No programa GiD é criada a malha não estruturada a partir dos vértices do contorno inserido e com o espaçamento mínimo que se queira. Com isso é possível a criação de qualquer polígono. Como o programa é uma ferramenta de pré-processamento ele gera uma malha que no decorrer do código no MATLAB gere pouco ou nenhum problema de condicionamento.

A entrada no código será feita com os pontos vértices do polígono que forma a placa, o menor espaçamento que deve ter entre os pontos, as condições de contorno, a espessura da placa, a carga uniforme que atua sobre a placa e os pontos internos e de contorno gerados pelo GiD.

Com um código feito no MATLAB é feito o espelhamento dos pontos internos em relação a cada bordo da placa, esses pontos espelhados são utilizados como pontos virtuais a serem pegos pelos pontos do contorno como vizinhos.

Figura 4 Exemplo de malha gerada pelo GiD

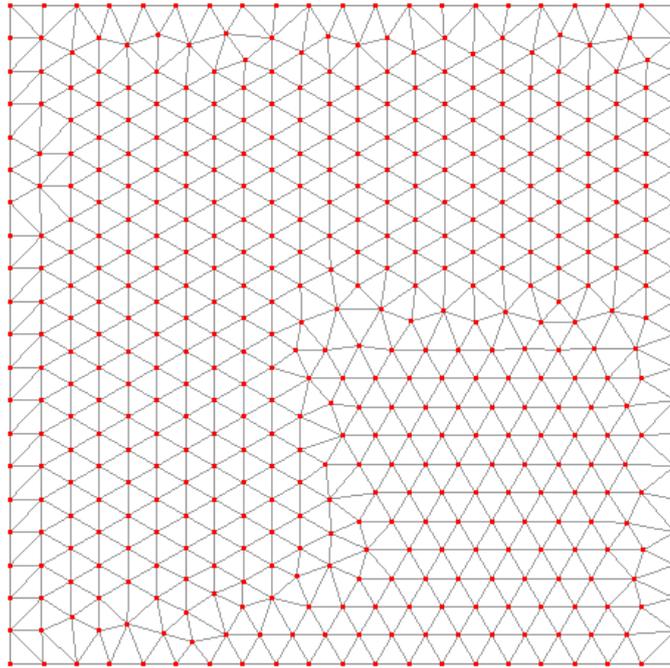


Figura 5 - Código de espelhamento de pontos no MATLAB

```

function Ref=Esp(Seg,P)
%% Retorna os pontos que são o reflexo dos pontos P pelo seguimento Seg.
%Seg - Cada linha é um ponto do seguimento espelhador;
%P - Cada linha é um ponto a ser espelhado.

[a,b]=size(P); %a - Número de pontos a serem espelhados;
Ref=zeros(a,b);
Mi=Seg(2,:)-Seg(1,:);
Mip=[Mi(2),-Mi(1)];
for i=1:a;
    xy=[-Mi(1),Mip(1);-Mi(2),Mip(2)]\([Seg(1,1);Seg(1,2)]-[P(i,1);P(i,2)]);
    Ref(i,:)=P(i,:)+Mip*xy(2)*2;
end

```

3.3 Desenvolvimento do método

Após a etapa de pré-processamento é feita a montagem da matriz $A1$ para cada ponto pivô, para isso foi criado um código chamado `Distancia1` para achar os 5 pontos mais próximos de um pivô que é um ponto interno e outro chamado `Distancia2` para achar os 5 pontos mais próximos de um pivô que é um ponto de contorno.

Com as linhas 3 e 5 da inversa da matriz $A1$ é montada uma matriz genérica A com npt (número de pontos total) linhas e npt colunas que vai ser modificada para formar outras duas

matrizes, uma para o sistema da Equação (27) e outra para o sistema da Equação (28). Sendo np o número de pontos internos e npc o número de pontos de contorno.

Figura 6 - Trecho do código da montagem da matriz A para os pontos internos

```
A11=inv(A1);
A(i,i)=-sum(A11(3,:)+A11(5,:));
for j=1:5;
    lugar=ceil(Dist(j,2)/np)-1;
    pos=Dist(j,2)-lugar*np;
    if lugar==0; A(i,pos)=A(i,pos)+A11(3,j)+A11(5,j);
    else A(i,pos+np)=A(i,pos+np)+A11(3,j)+A11(5,j);
    end
end
```

Figura 7 - Trecho do código da montagem da matriz A para os pontos de contorno

```
A11=inv(A1);
A(np+i,np+i)=-sum(A11(3,:)+A11(5,:));
for j=1:5;
    lugar=ceil(Dist(j,2)/np)-1;
    pos=Dist(j,2)-lugar*np;
    if lugar==0; A(np+i,pos)=A(np+i,pos)+A11(3,j)+A11(5,j);
    elseif lugar>=1 && lugar<=n1;
        if CA(lugar)==1; A(np+i,pos)=A(np+i,pos)+A11(3,j)+A11(5,j);
        elseif CA(lugar)==2; A(np+i,pos)=A(np+i,pos)-A11(3,j)-A11(5,j);
        end
    else A(np+i,pos+np)=A(np+i,pos+np)+A11(3,j)+A11(5,j);
    end
end
```

A partir da matriz A é construída a matriz A_m para o sistema da Equação (27). Neste sistema apenas os pontos internos são pivô, logo, as linhas da matriz A destinadas aos pontos de contorno como pivô são eliminadas para compor a matriz A_m .

Já para o sistema da Equação (28) é montada a matriz A_w a partir da matriz A. Neste sistema, como as condições de contorno para o código só podem ser engastado ou simplesmente apoiados, o deslocamento no contorno é zero, logo, as colunas destinadas aos pontos de contorno são eliminadas para compor a matriz A_w .

Para a resolução simultânea do problema são construídos uma outra matriz quadrada AA com dimensão $npt + np$ e um vetor b de tamanho $(1, npt + np)$ descrito na Figura 9.

Figura 8 - Trecho do código da montagem da matriz AA e vetor b

```
%% Criação do vetor b para o problema AA * Mw = b  
b=[-ones(np,1)*q;zeros(npt,1)];  
-----  
%% Criação da matriz AA para o problema AA * Mw = b  
AA=[Am zeros(np);zeros(npt) Aw];  
v=.2; E=1; D=E*H^3/12/(1-v^2);%Propriedades do material  
for i=1:npt; AA(i+np,i)=1/D; end  
-----
```

Como resposta é dado o vetor Mw, onde os elementos de 1 a npt são a resolução do sistema da Equação (27) e os elementos restantes são os deslocamentos da placa.

4 APLICAÇÕES E RESULTADOS

4.1 MDFG e MDF

O exemplo a seguir é de uma placa retangular com comprimento igual a 2,0 m, largura igual a 1,0 m, 10,0 cm de espessura. O modulo de elasticidade da placa usado foi 22,33 Mpa e coeficiente de Poisson foi 0,2.

Para o MDFG foi usado um espaçamento mínimo entre os pontos de 0,1 m; o que gerou uma malha com 185 pontos internos e 60 pontos de contorno. Para o MDF foi usado uma malha com os espaçamentos vertical e horizontal de 0,1 m; o que gerou uma malha de 171 pontos internos e 60 pontos de contorno.

O problema foi resolvido para as situações de todos os bordos simplesmente apoiados e todos os bordos engastados com a carga transversal uniformemente distribuída de 1,0 kN/m².

Figura 9 - Malha gerada pelo GiD para o MDFG

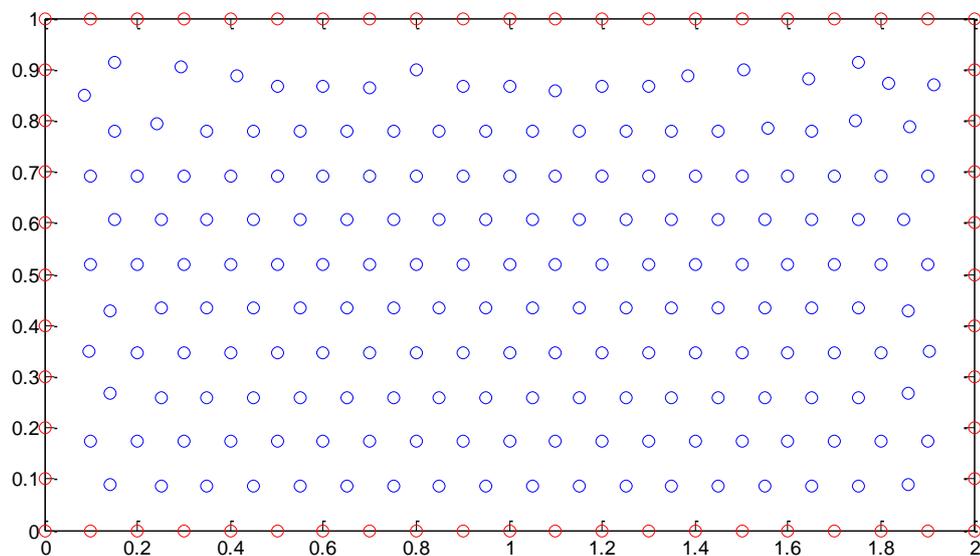


Figura 10 – Deslocamentos (m) pelo MDFG (todos os bordos simplesmente apoiados)

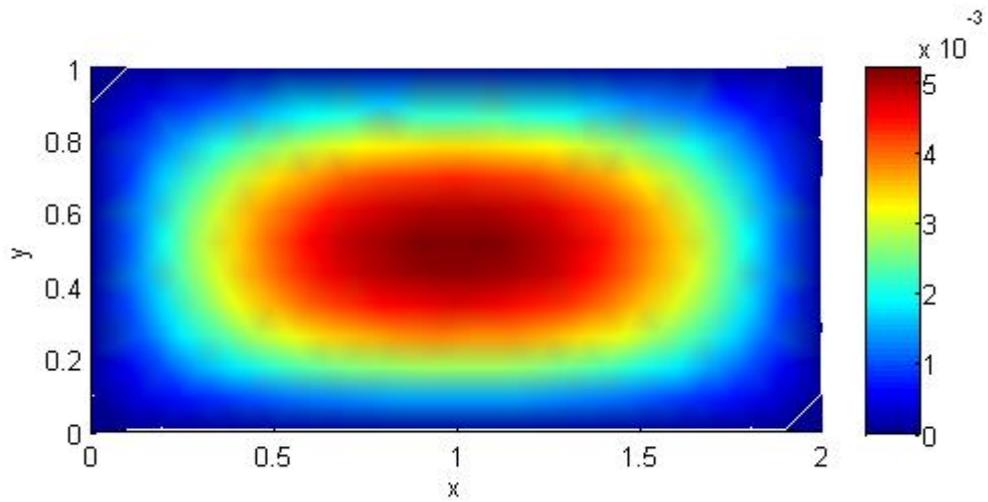


Figura 11 – Deslocamentos (m) pelo MDF – vista lateral (todos os bordos simplesmente apoiados)

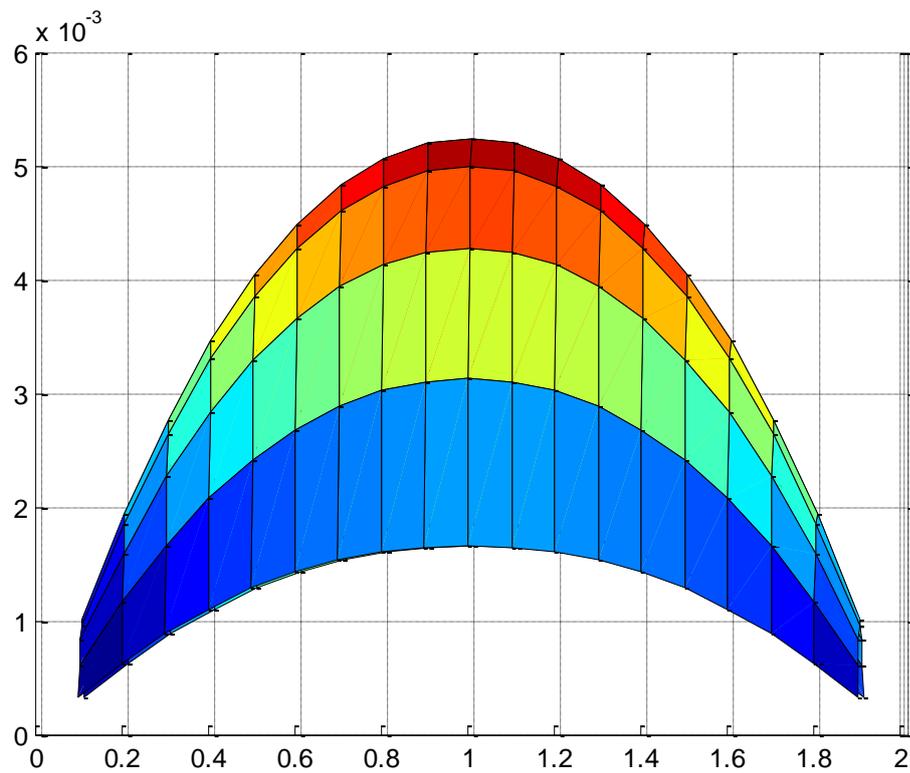


Figura 12 – Deslocamentos (m) pelo MDFG (todos os bordos engastados)

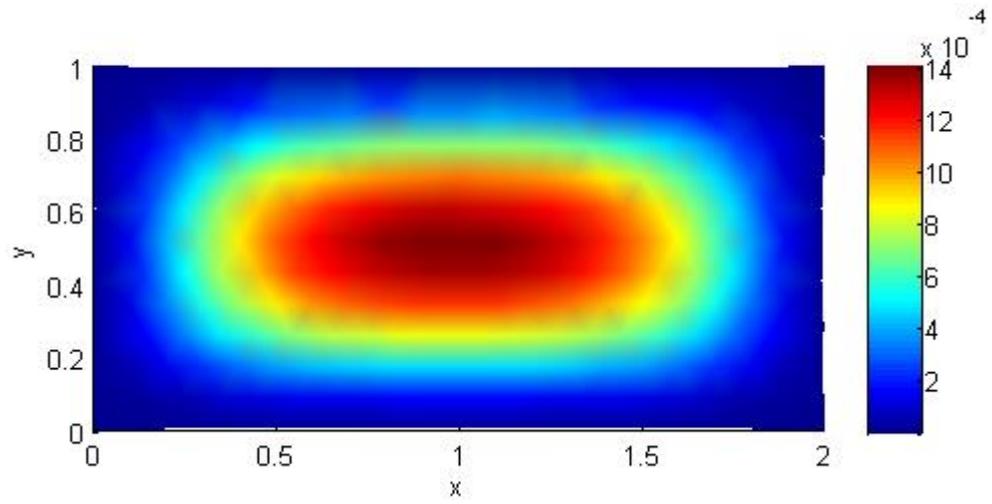
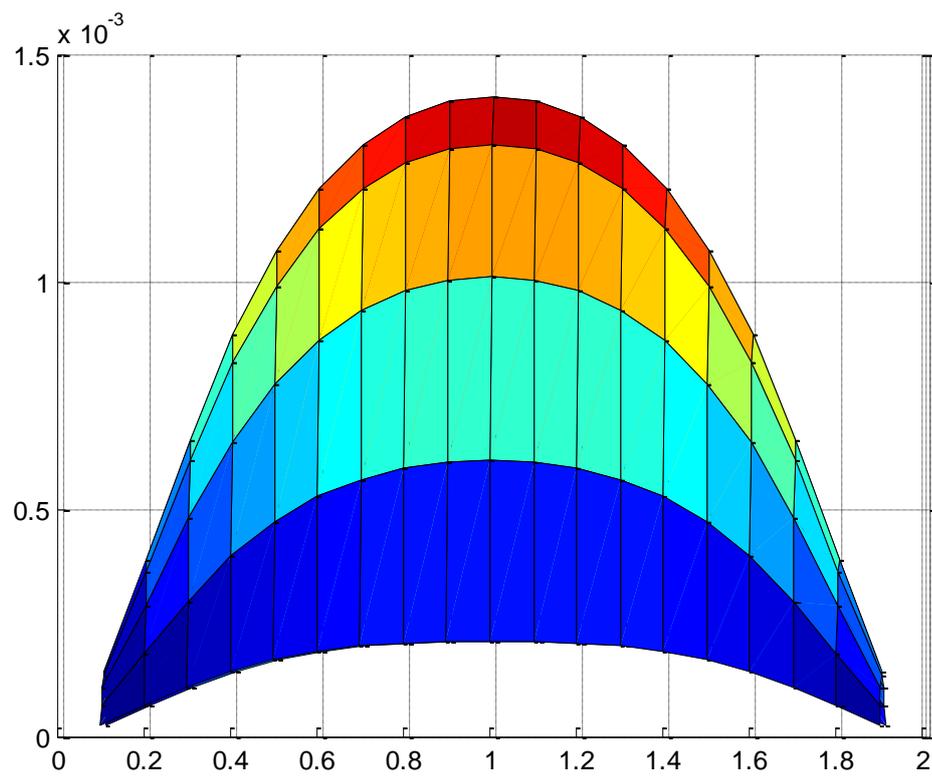


Figura 13 – Deslocamentos (m) pelo MDF – vista lateral (todos os bordos engastados)



Comparando os deslocamentos no centro do vão:

Tabela 1 – Erro entre o MDFG e o MDF

| | wc - MDFG (m) | wc - MDF (m) | diferença |
|----------------|---------------|--------------|-----------|
| Todo Engastado | 0,00138928 | 0,0013317 | 4% |
| Todo Apoiado | 0,00518651 | 0,0052299 | 1% |

As pequenas diferenças demonstram que as soluções dadas pelos dois métodos estão bastante próximas, quanto mais pontos forem utilizados menor será o erro entre eles. O próximo exemplo será comparando o MDFG com uma solução exata para dar maior consistência aos resultados obtidos pelo MDFG

4.2 MDFG e Solução Exata

Neste exemplo foi usado uma placa retangular com comprimento igual a 2,0 m, largura igual a 1,5 m e 10,0 cm de espessura. O módulo de elasticidade da placa usado foi 22,33 MPa e coeficiente de Poisson foi 0,2.

O MDFG foi usado com espaçamento mínimo entre os pontos de 0,1 m a 0,06 m; o que gerou uma quantidade de pontos mostrados na Tabela 2. A solução exata a qual foi comparado o método foi calculada por Araújo (2010).

O problema foi resolvido para a situação em que o bordo esquerdo esta engastado e os outros 3 bordos simplesmente apoiados com a carga transversal uniformemente distribuída de 1,0 kN/m². Foram comparados os deslocamentos centrais, os momentos centrais nas direções x e y e o momento no engaste em x. Os resultados são mostrados na Tabela 2.

Tabela 2 – Resultados obtidos pelo MDFG para o exemplo

| dmin (m) | nº pontos internos | nº pontos de contorno | wc (m) | Mx (kNm/m) | My (kNm/m) | Mxe (kNm/m) |
|----------|--------------------|-----------------------|------------|------------|------------|-------------|
| 0,1 | 288 | 70 | 0,01375715 | 0,093035 | 0,12305215 | -0,23222308 |
| 0,09 | 387 | 78 | 0,01375864 | 0,09334519 | 0,12329002 | -0,23437156 |
| 0,08 | 487 | 88 | 0,01367805 | 0,09275069 | 0,12251093 | -0,23643587 |
| 0,07 | 651 | 100 | 0,01371451 | 0,09360031 | 0,12289976 | -0,23672429 |
| 0,06 | 869 | 116 | 0,0137551 | 0,09333473 | 0,1233127 | -0,23691235 |

Para a malha com a espessura mínima de 0,06 m os resultados obtidos estão nas Figuras de 17 a 21.

Figura 14 – Malha de 0,06 m

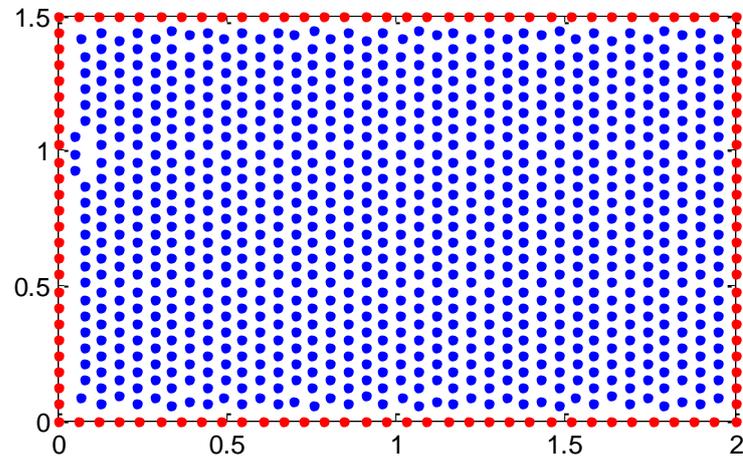


Figura 15 – Deslocamentos (m) para a malha de 0,06 m

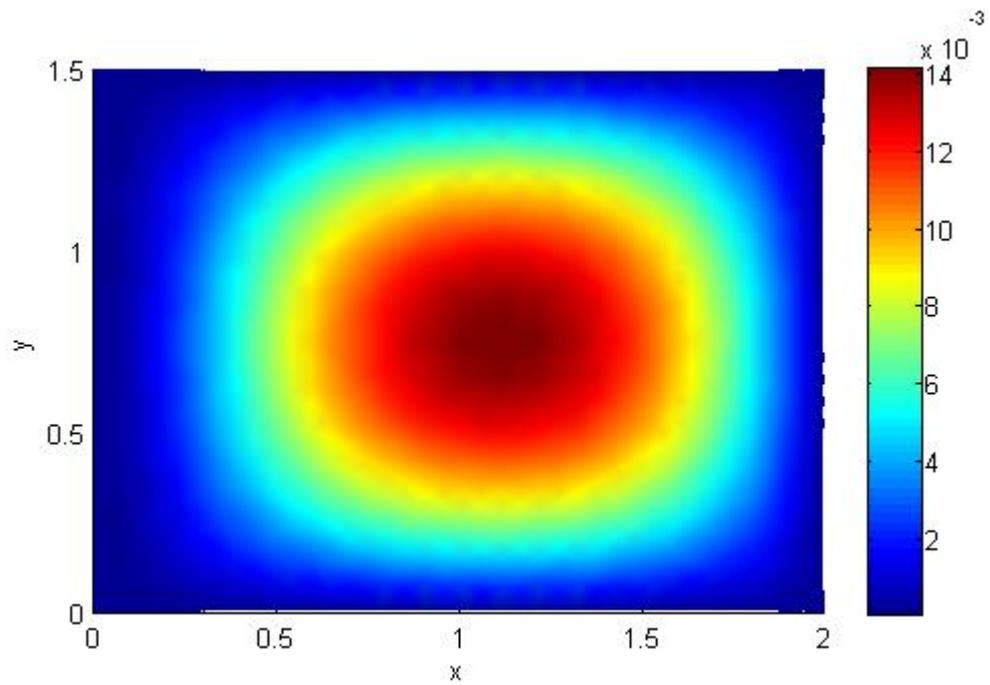


Figura 16 – Momentos (kNm/m) na direção X para a malha de 0,06 m

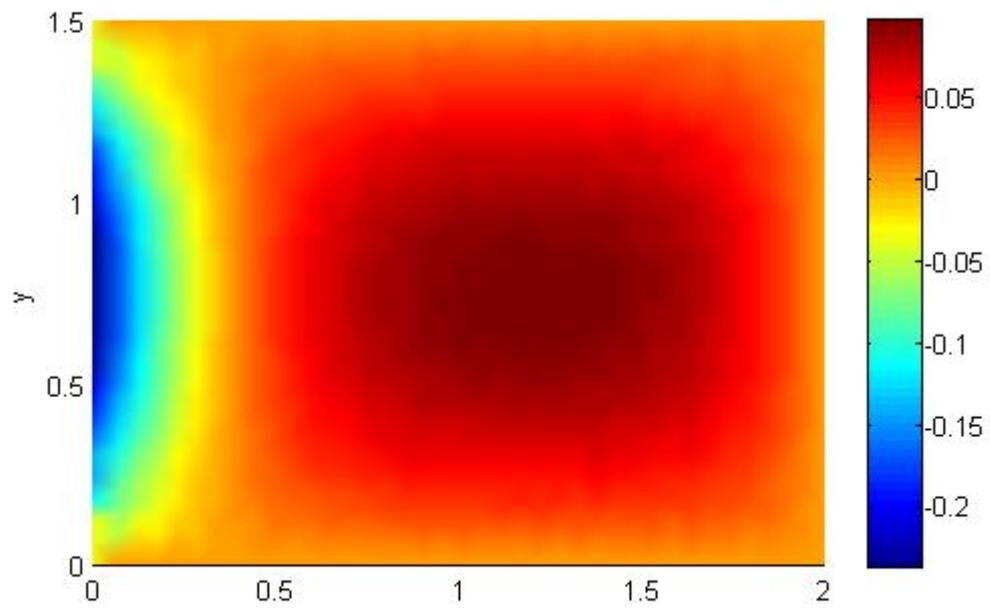


Figura 17– Momentos (kNm/m) na direção Y para a malha de 0,06 m

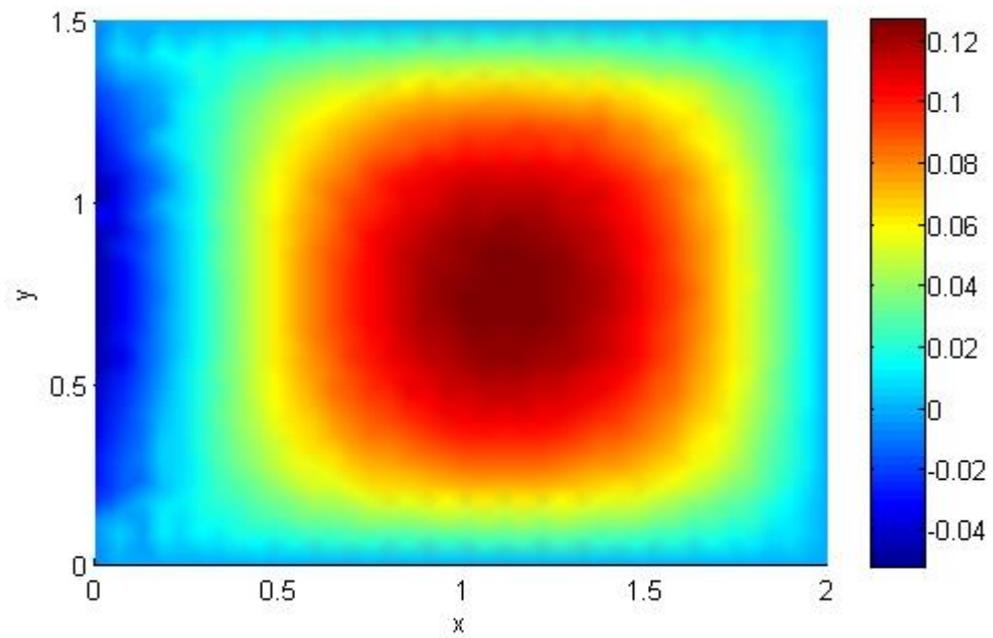
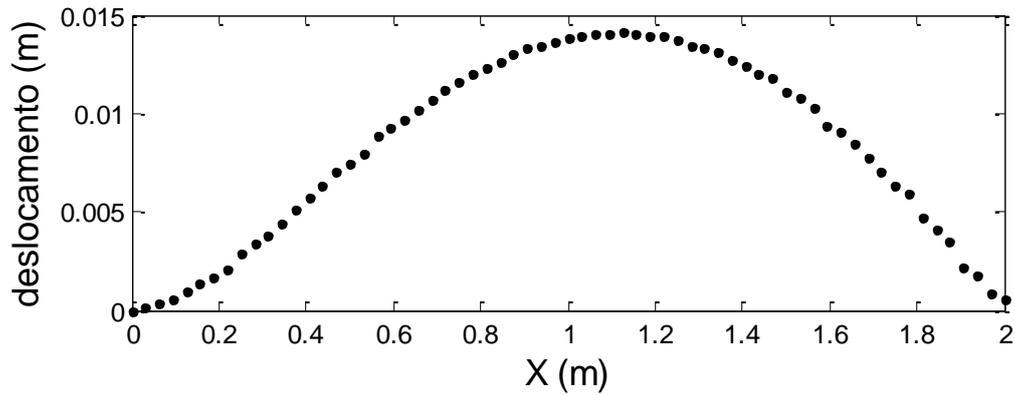
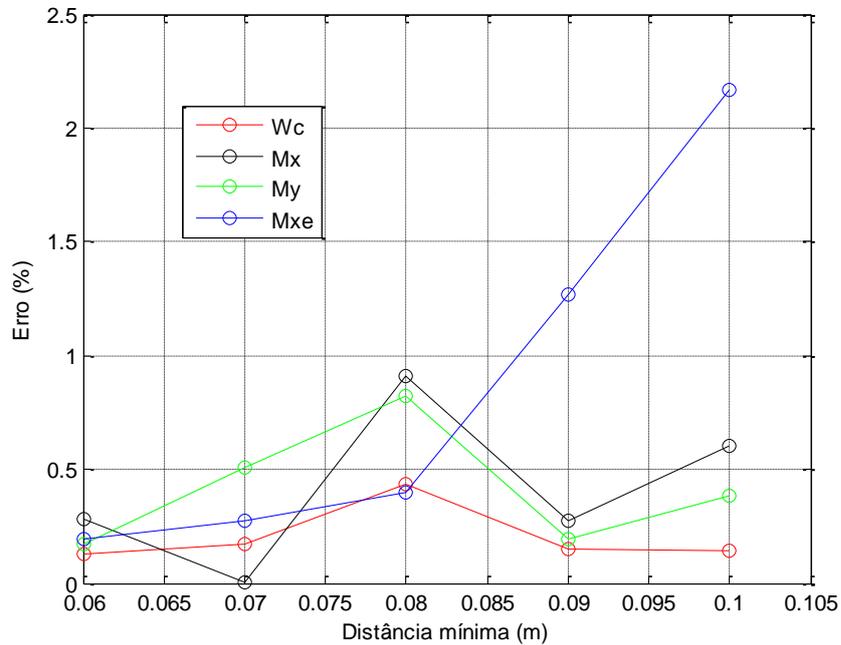


Figura 18 – Seção transversal ($y = 0,75\text{m}$) para a malha de $0,06\text{ m}$



Em comparação com a resposta exata a Figura 22 mostra os erros em função da distância mínima entre os pontos da malha. Percebe-se que quando vai se diminuindo a distância mínima entre os pontos o maior erro dos quatro parâmetros verificados vai diminuindo comprovando sua convergência.

Figura 19 – Gráfico: erro X distância mínima entre os pontos da malha



4.3 Placa circular

Os exemplos anteriores dão consistência ao MDFG comparando sua solução à solução por MDF e a solução exata. Os exemplos a seguir demonstram o funcionamento do método para diferentes tipos de contorno.

Para o próximo exemplo foi usada uma placa circular com raio de 0,5m; espessura de 0,1m e contorno simplesmente apoiado. O módulo de elasticidade usado foi de 22,33 MPa e coeficiente de Poisson de 0,2. A carga distribuída aplicada transversalmente a placa foi de $1,0\text{kN/m}^2$.

As Figuras 23 e 25 mostram a disposição das malhas com os espaçamentos mínimos entre os pontos de 0,05m e 0,1m respectivamente. Já nas Figuras 24 e 26 dá para ver os deslocamentos calculados pelo método para estas malhas.

Figura 20 – Malha de 0,05m

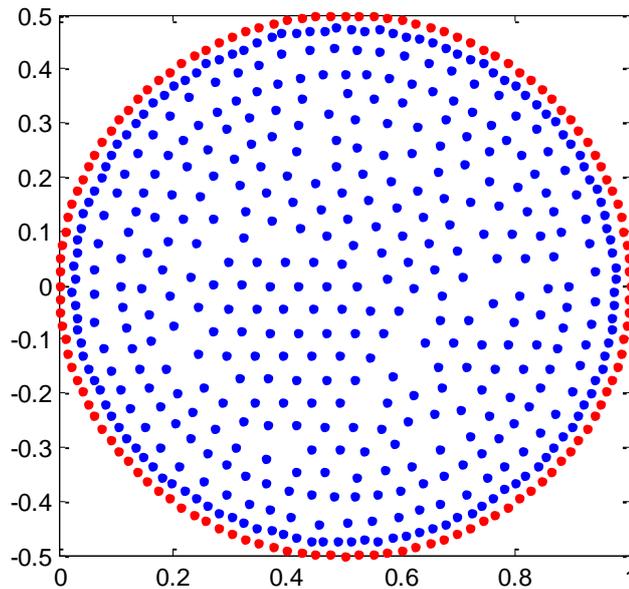


Figura 21 – Deslocamentos (m) com a malha de 0,05m

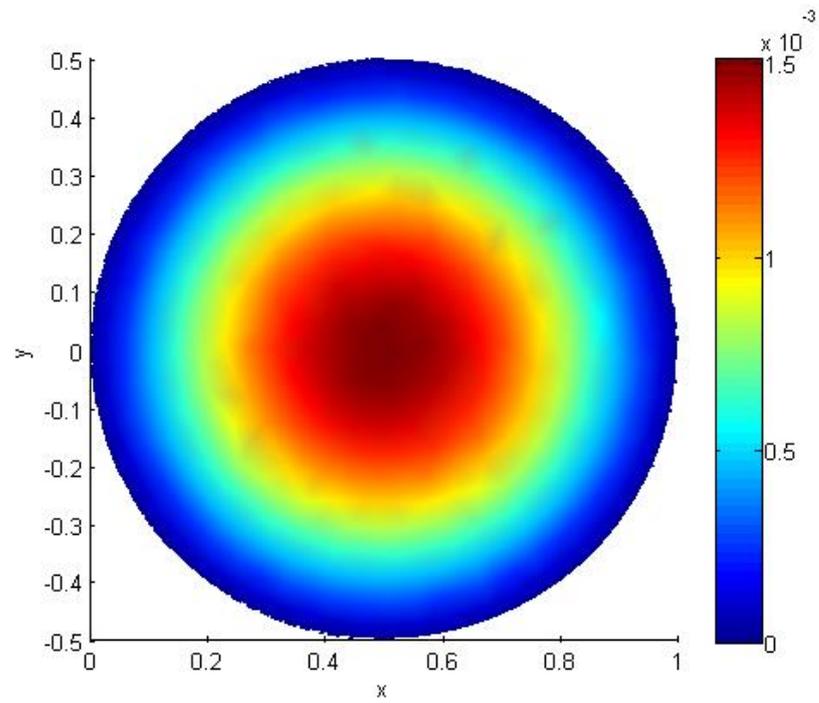


Figura 22 – Malha de 0,1m

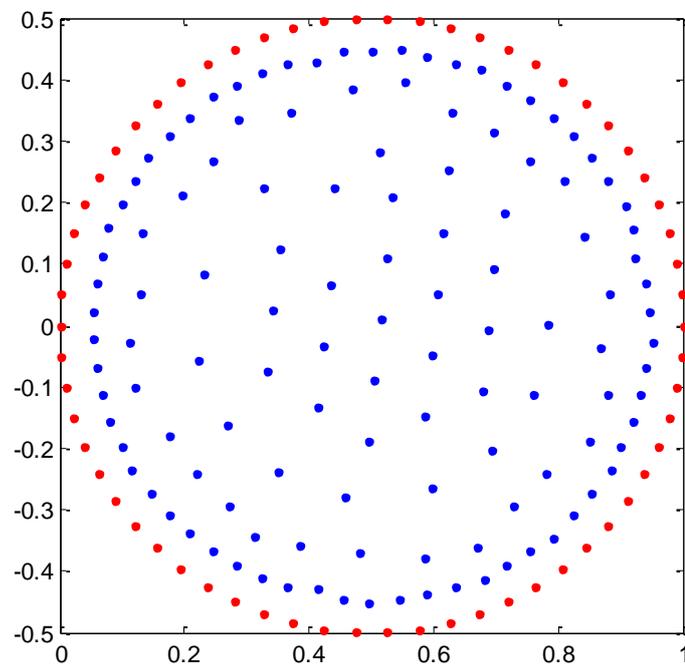
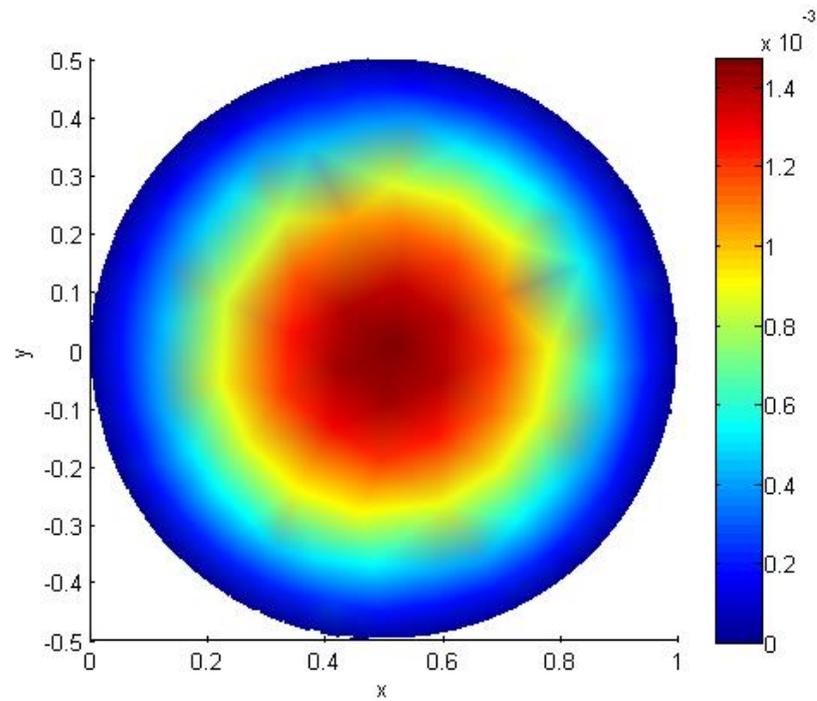
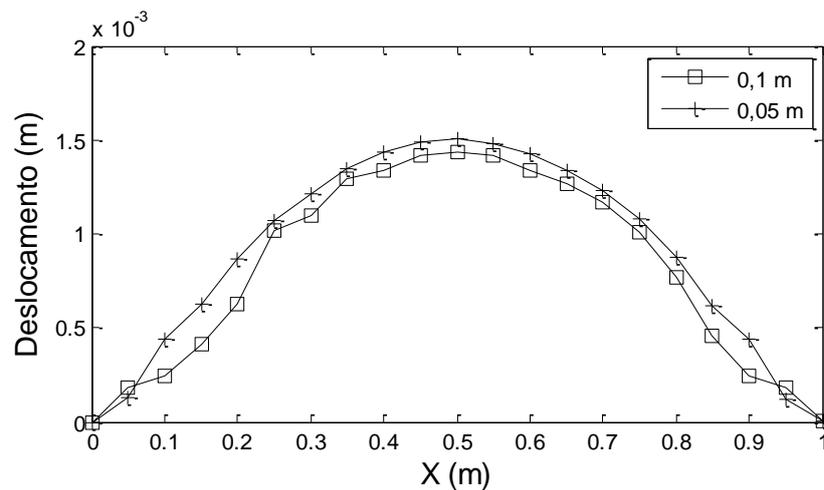


Figura 23 – Deslocamentos (m) com a malha de 0,1m



A Figura 27 mostra um corte transversal paralelo o eixo X, nela é possível ver que apesar do deslocamento central ser bem próximo para as duas malhas a distribuição dos deslocamentos ao longo da placa melhora bastante ao aumentar a quantidade de pontos do domínio.

Figura 24 – Seção transversal ($y = 0,0$ m) para a malha

4.4 MDFG e MEF (poligonal qualquer)

Neste exemplo foi usada uma placa poligonal conforme mostrado na Figura 28 com espessura de 0,1m e com uma malha de 0,2m tanto para o MDFG quanto para o MEF. O módulo de elasticidade usado foi de 22,33 MPa e o coeficiente de Poisson foi de 0,2. A carga transversal distribuída foi de 1kN/m^2 .

Na Figura 28 é possível ver que as condições de contorno utilizadas foram engastado para os bordos em vermelho e simplesmente apoiado para os contornos em preto, os deslocamentos são mostrados nas Figura 30 e 31, os momentos na direção X nas Figuras 32 e 33, e os momentos na direção Y nas figuras 34 e 35.

Figura 25 – Malha de 0,2m (MDFG)

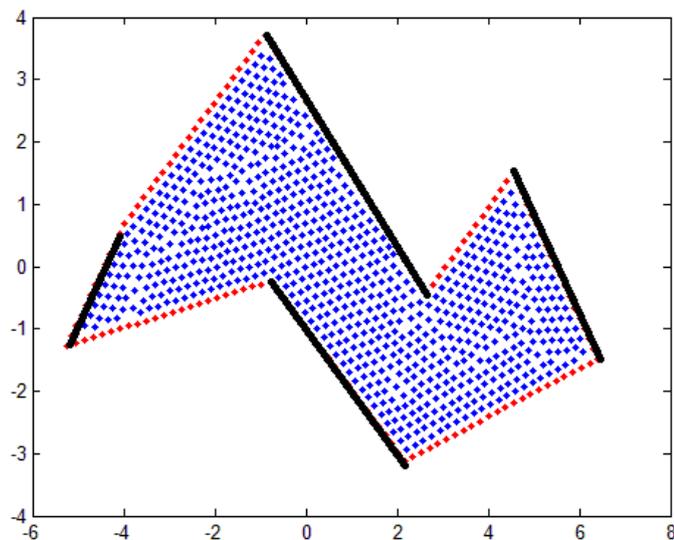


Figura 26 – Malha de 0,2m (MEF)

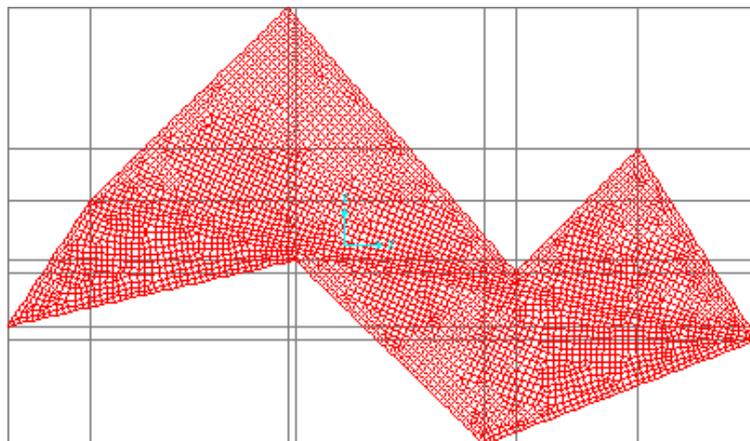


Figura 27 – Deslocamentos (m) pelo MDFG

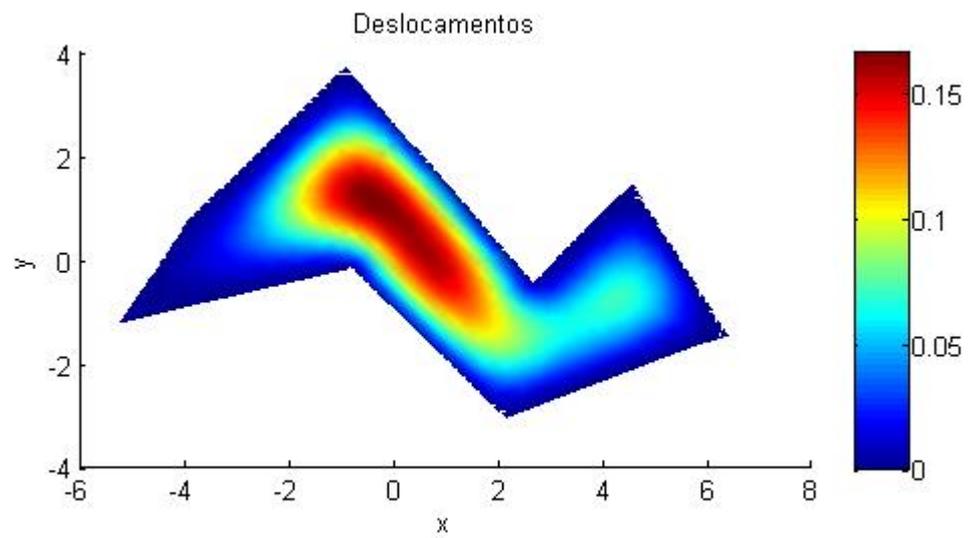
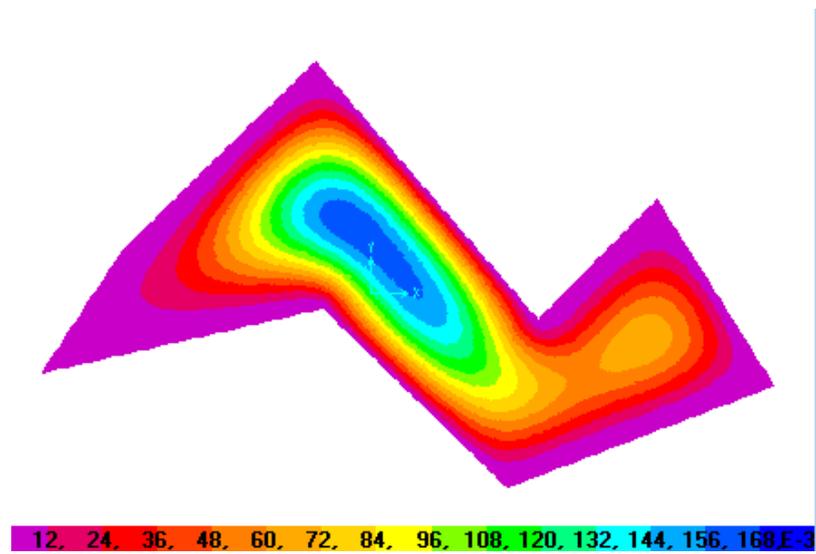


Figura 28 – Deslocamentos (m) pelo MEF



Comparando os deslocamentos calculados pelos dois métodos é possível notar visualmente que possuem soluções muito próximas. Analisando em um ponto P (-0,78;1,49) pode-se ver pela Tabela 3 que o erro entre os dois métodos é muito pequeno, o que comprova ainda mais a proximidade de suas soluções.

Tabela 3 – Comparação dos deslocamentos entre o MDFG e MEF no ponto P(-0,78;1,49)

| | MEF | MDFG | erro |
|------------------|--------|--------|------|
| Deslocamento (m) | 0,1531 | 0,1553 | 1,4% |

4.5 MDFG e MEF (placa em U)

No exemplo a seguir foi usada uma placa em U como é possível ver nas Figuras 36 e 37. Foi usada uma malha de 0,09m para o MDFG e de 0,1m para o MEF. Para o MEF foi utilizado o SAP 2000 versão 14.2.2. A espessura da placa é de 0,1m. o módulo de elasticidade usado foi de 22,33 MPa e o coeficiente de Poisson de 0,2. A carga transversal distribuída foi de 1kN/m^2 .

A placa foi analisada como se fosse uma garagem de um edifício a ser engastado ou não no mesmo, logo, ela foi resolvida destas duas formas. Nas Figuras 38 e 39 é possível ver o comportamento da placa caso fosse engastada no edifício, pelo MDFG e pelo MEF respectivamente. E na Figuras 40 e 41 com a placa não engastada no edifício.

Figura 29 – Malha de 0,09m gerada para o MDFG

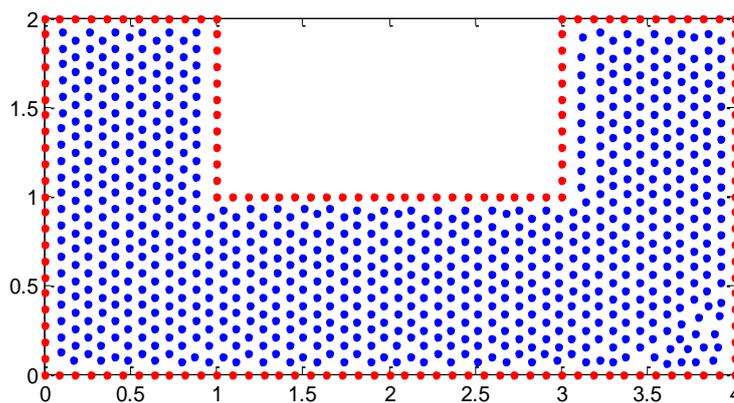


Figura 30 - Malha de 0,1m gerada para o MEF

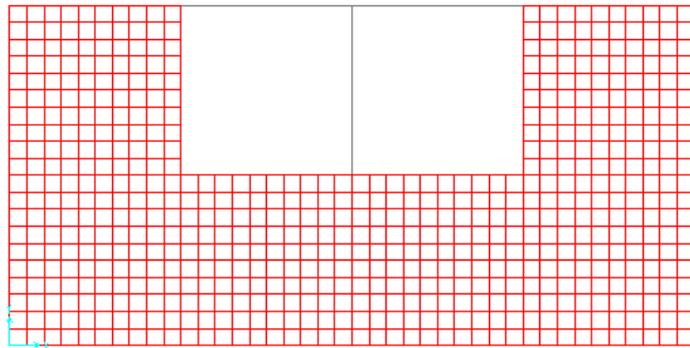


Figura 31 – Deslocamentos (m) de placa U com bordos internos engastados e o resto simplesmente apoiados (MDFG)

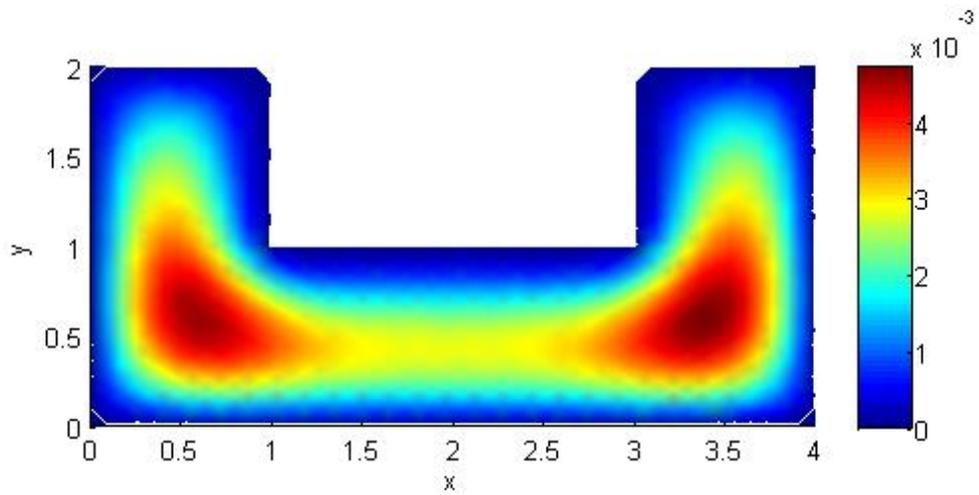


Figura 32- Deslocamentos (m) de placa U com bordos internos engastados e o resto simplesmente apoiados (MEF)

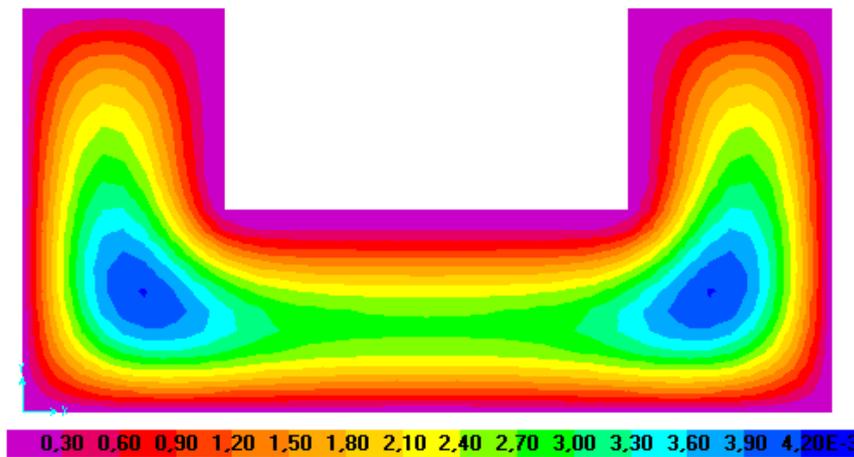


Figura 33 – Deslocamentos (m) de placa U com os bordos simplesmente apoiados (MDFG)

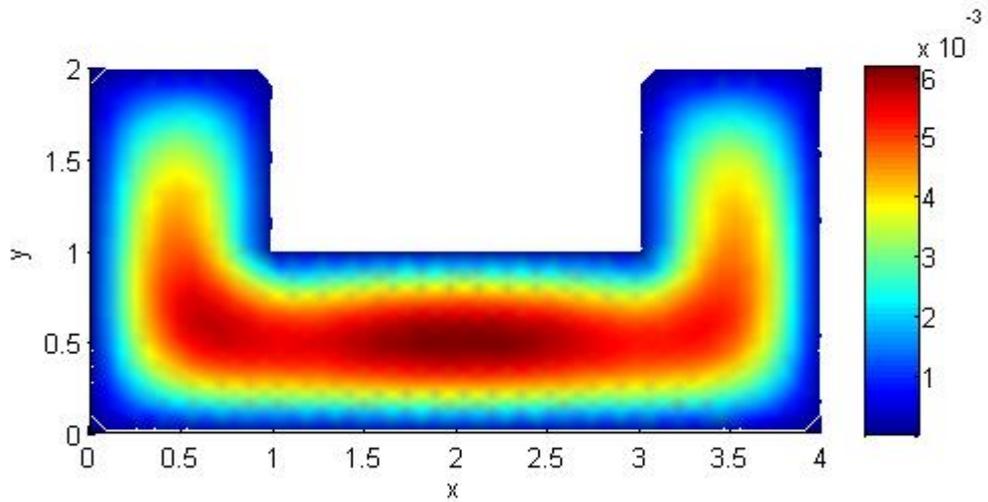
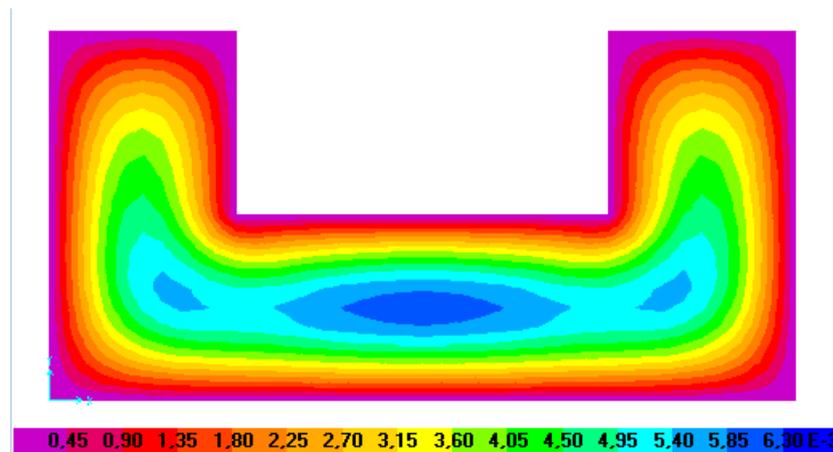


Figura 34 – Deslocamentos (m) de placa U com os bordos simplesmente apoiados (MEF)



Como se pode ver os resultados para os dois métodos deram muito próximos para ambos os casos o que confirma a convergência do MDFG para a mesma solução que o MEF, comprovando assim sua eficiência.

5 CONCLUSÃO

Os dois primeiros e os dois últimos exemplos validados confirmam o bom funcionamento e convergência do método para a análise de diversos problemas independente do domínio apresentado, tanto para o cálculo de deslocamentos quanto para o cálculo dos momentos. A utilização do GiD para geração da malha não estruturada maximiza a quantidade de pontos com uma mesma distância mínima entre os pontos da malha e diminui os problemas de singularidade devido a melhor disposição dos pontos na malha.

O artifício utilizado de resolver duas equações de Poisson em vez de resolver a equação de Lagrange também diminui os problemas de singularidades devido ao fato de para cada pivô necessitar de menos vizinhos. E a resolução simultânea das duas equações de Laplace permite se trabalhar como outras condições de contorno além de simplesmente apoiado.

Existe muito espaço para futuras melhorias, seja no código ou na abordagem do problema. Se propõe aqui a adição da condição de contorno livre, o aumento no grau da expansão de Taylor para a melhoria da solução com menos pontos no domínio, a utilização de métodos de refinamento para a melhoria da solução e um estudo comparativo de convergência entre estes métodos de refinamento.

REFERÊNCIAS

- ARAÚJO, José Milton de. **Curso De Concreto Armado**. 3. ed. Rio Grande: Dunas, 2010. 395 p. v. 2.
- BENITO, J. J.; URENA, F.; GAVETE, L. **Influence of several factors in the generalized finite difference method**, Madrid - Spain. Applied Mathematical Modelling, vol. 25, p. 1039-1053, 2001.
- CHAPRA, Steven C.; CANALE, Raymond P. **Métodos numéricos para engenharia**. 5. ed. Porto Alegre: AMGH, 2008. 808 p.
- FERREIRA, Augusto César Albuquerque. **Método sem malha com diferenças finitas generalizadas para solução de equações diferenciais parciais**. 2016. 61 f. TCC (Graduação) - Curso de Engenharia Civil, Universidade Federal de Pernambuco, Recife, 2016.
- GILAT, Amos; SUBRAMANIAM, Vish. Fórmulas de diferenças finitas usando a expansão em série de Taylor. In: GILAT, Amos; SUBRAMANIAM, Vish. **Métodos Numéricos para Engenheiros e Cientistas: Uma Introdução com Aplicações Usando o MATLAB**. Ni: Bookman, 2008. p. 260-269.
- PERRONE, N.; KAO, R. **A general finite difference method for arbitrary meshes**, Washington, DC – USA. Catholic University Of America, v. 5, p. 45-58, 1975.
- TIMOSHENKO, Stephen P.; WOINOWSKY-KRIEGER, S.. **Teoría de placas y láminas**. Ni: Urmo, 1970. 621 p.
- UGURAL, Ansel C. **Stresses in Beams, Plates, and Shells**. 3. ed. Boca Raton: CRC Press, 2010. 574 p.

ANEXO A

Esse anexo contém um código computacional desenvolvido no MATLAB para a resolução do problema de placa separando-o em dois problemas de segunda ordem pelo método das diferenças finitas generalizadas.

```
function wc=T3Lap2e(xv,yv,dmin,H,q,CA,x,y,xc,yc)
%% Programa q calcula deslocamentos em placas com bordos engastados ou
apoiados
% w - deslocamento vertical em cada ponto da placa calculados pelo código;
% xv, yv - coordenadas dos vértices;
% dmin - distância mínima entre dois pontos;
% q - carga transversal
% CA - condições de apoio de cada bordo (1-engastado|2-simplesmente
% apoiado);
close all

%% Geração da malha
[x,y,xc,yc]=Gpic(xv,yv,dmin);
np=length(x) %n° de pontos internos
npc=length(xc)%n° de pontos no contorno
npt=np+npc;%n° de pontos totais
nl=length(xv);%n° de lados da placa
for i=1:nl;
    if i~=nl;
ref{i}=Esp([xv(i),yv(i);xv(i+1),yv(i+1)], [transpose(x),transpose(y)]);
        else ref{i}=Esp([xv(i),yv(i);xv(1),yv(1)], [transpose(x),transpose(y)]);
        end
    end
%% Criação da matriz A para o problema A * w = b
A1=zeros(5);
A=zeros(npt);
for i=1:np;
    Dist=Distancial(i,x,y,xc,yc,dmin,5,ref);
    %Dist=Quadrante(i,x,y,xc,yc,dmin,5);
    %figure
    %plot(x(i),y(i),'*k');
    %hold on
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0; h=x(i)-x(pos);k=y(i)-y(pos);
%plot(x(pos),y(pos),'vr');
        elseif lugar>=1 && lugar<=nl; h=x(i)-ref{lugar}(pos,1);k=y(i)-
ref{lugar}(pos,2); %plot(ref{lugar}(pos,1),ref{lugar}(pos,2),'vb');
        else h=x(i)-xc(pos);k=y(i)-yc(pos); %plot(xc(pos),yc(pos),'<g');
        end
        A1(j,:)=[h k h^2/2 h*k k^2/2];
    end
    %hold off
    A11=inv(A1);
    A(i,i)=-sum(A11(3,:)+A11(5,:));
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
```

```

        pos=Dist(j,2)-lugar*np;
        if lugar==0; A(i,pos)=A(i,pos)+A11(3,j)+A11(5,j);
        else A(i,pos+np)=A(i,pos+np)+A11(3,j)+A11(5,j);
        end
    end
end
for i=1:npc;
    Dist=Distancia2(i,x,y,xc,yc,dmin,5,ref);
    %Dist=Quadrante(i,x,y,xc,yc,dmin,5);
    %figure
    %plot(xc(i),yc(i),'*k');
    %hold on
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0; h=xc(i)-x(pos);k=yc(i)-y(pos);
        %plot(x(pos),y(pos),'vr');
        elseif lugar>=1 && lugar<=nl; h=xc(i)-ref{lugar}(pos,1);k=yc(i)-
        ref{lugar}(pos,2); %plot(ref{lugar}(pos,1),ref{lugar}(pos,2),'vb');
        else h=xc(i)-xc(pos);k=yc(i)-yc(pos); %plot(xc(pos),yc(pos),'<g');
        end
        A1(j,:)=[h k h^2/2 h*k k^2/2];
    end
    %hold off
    A11=inv(A1);
    A(np+i,np+i)=-sum(A11(3,:)+A11(5,:));
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0; A(np+i,pos)=A(np+i,pos)+A11(3,j)+A11(5,j);
        elseif lugar>=1 && lugar<=nl;
            if CA(lugar)==1; A(np+i,pos)=A(np+i,pos)+A11(3,j)+A11(5,j);
            elseif CA(lugar)==2; A(np+i,pos)=A(np+i,pos)-A11(3,j)-A11(5,j);
            end
        else A(np+i,pos+np)=A(np+i,pos+np)+A11(3,j)+A11(5,j);
        end
    end
end
Am=A(1:end-npc,:);
Aw=A(:,1:end-npc);
%% Criação do vetor b para o problema AA * Mw = b
b=[-ones(np,1)*q;zeros(npt,1)];
%% Criação da matriz AA para o problema AA * Mw = b
AA=[Am zeros(np);zeros(npt) Aw];
v=.2; E=22330; D=E*H^3/12/(1-v^2);%Propriedades do material
for i=1:npt; AA(i+np,i)=1/D; end
%% Resolução do problema A * w = c
Mw=AA\b;
M=Mw(1:npt);
w=Mw(1+npt:end);
%% Impressão dos resultados
xp=[x xc];
yp=[y yc];
wp=[w;zeros(npc,1)];
Mp=M;
%figure
%trisurf(delaunay(xp,yp),xp,yp, Tp);

```

```

%% Momentos Mx e My
AM=zeros(5);
Mx=zeros(npt,1); My=zeros(npt,1);
for i=1:np;
    Dist=Distancia1(i,x,y,xc,yc,dmin,5,ref);
    %Dist=Quadrante(i,x,y,xc,yc,dmin,5);
    %figure
    %plot(x(i),y(i),'*k');
    %hold on
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0; h=x(i)-x(pos);k=y(i)-y(pos);
%plot(x(pos),y(pos),'vr');
        else h=x(i)-xc(pos);k=y(i)-yc(pos); %plot(xc(pos),yc(pos),'<g');
        end
        AM(j,:)=[h k h^2/2 h*k k^2/2];
    end
    %hold off
    AM1=inv(AM);
    Mx(i)=wp(i)*sum(AM1(3,:)+v*AM1(5,:));
    My(i)=wp(i)*sum(v*AM1(3,:)+AM1(5,:));
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0;
            Mx(i)=Mx(i)-wp(pos)*(AM1(3,j)+v*AM1(5,j));
            My(i)=My(i)-wp(pos)*(v*AM1(3,j)+AM1(5,j));
        else
            Mx(i)=Mx(i)-wp(pos+np)*(AM1(3,j)+v*AM1(5,j));
            My(i)=My(i)-wp(pos+np)*(v*AM1(3,j)+AM1(5,j));
        end
    end
end
end

for i=1:np;
    Dist=Distancia2(i,x,y,xc,yc,dmin,5,ref);
    %Dist=Quadrante(i,x,y,xc,yc,dmin,5);
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0; h=xc(i)-x(pos);k=yc(i)-y(pos);
        elseif lugar>=1 && lugar<=nl; h=xc(i)-ref{lugar}(pos,1);k=yc(i)-
ref{lugar}(pos,2);
        else h=xc(i)-xc(pos);k=yc(i)-yc(pos);
        end
        AM(j,:)=[h k h^2/2 h*k k^2/2];
    end
    AM1=inv(AM);
    Mx(np+i)=wp(np+i)*sum(AM1(3,:)+v*AM1(5,:));
    My(np+i)=wp(np+i)*sum(v*AM1(3,:)+AM1(5,:));
    for j=1:5;
        lugar=ceil(Dist(j,2)/np)-1;
        pos=Dist(j,2)-lugar*np;
        if lugar==0;
            Mx(np+i)=Mx(np+i)-wp(pos)*(AM1(3,j)+v*AM1(5,j));
            My(np+i)=My(np+i)-wp(pos)*(v*AM1(3,j)+AM1(5,j));
        elseif lugar>=1 && lugar<=nl;
            if CA(lugar)==1;
                Mx(np+i)=Mx(np+i)-wp(pos)*(AM1(3,j)+v*AM1(5,j));
            end
        end
    end
end

```



```

% carrega imagem e opções
figure
tri=delaunay(xp,yp);
plots=trisurf(tri,xp,yp,wp,'edgecolor','k','facecolor','interp'); % plota
xyz
grid off
xlabel('x')
ylabel('y')
view(0,90)
colorbar
set(plots,'LineStyle','none')
daspect([1 1 1])
title('Deslocamentos');

%% plot Mx

% carrega imagem e opções
figure
tri=delaunay(xp,yp);
plots=trisurf(tri,xp,yp,Mx,'edgecolor','k','facecolor','interp'); % plota
xyz
grid off
xlabel('x')
ylabel('y')
view(0,90)
colorbar
set(plots,'LineStyle','none')
daspect([1 1 1])
title('Momentos na direção X');
%% plot My

% carrega imagem e opções
figure
tri=delaunay(xp,yp);
plots=trisurf(tri,xp,yp,My,'edgecolor','k','facecolor','interp'); % plota
xyz
grid off
xlabel('x')
ylabel('y')
view(0,90)
colorbar
set(plots,'LineStyle','none')
daspect([1 1 1])
title('Momentos na direção Y');
end

```

ANEXO B

Esse anexo contém um código computacional desenvolvido no MATLAB para a aplicação do critério da distância na seleção dos pontos vizinhos quando o ponto pivô é um ponto interno. Ele é usado pelo código do anexo A.

```
function Dist=Distancial(i,x,y,xc,yc,dmin,n,varargin)
%Função que aplica o critério das distâncias nos pontos internos
%i - posição do ponto pivô
%x,y - coordenadas dos pontos internos
%xc,yc - coordenadas dos pontos de contorno
%dmin - distância mínima entre os pontos da malha
%n - quantidade de pontos necessários

Dist=[1000000 0];
np=length(x);
npc=length(xc);
a=3.5;
if nargin-7>0;
    ref=varargin;
    nl=length(ref{1});
else nl=0;
end
for j=1:np;
    if x(j)<=x(i)+a*dmin && x(j)>=x(i)-a*dmin && y(j)<=y(i)+a*dmin &&
y(j)>=y(i)-a*dmin && i~=j;
        Dist=[Dist;sqrt((x(i)-x(j))^2+(y(i)-y(j))^2) j];
    end
end
for j=1:npc;
    if xc(j)<=x(i)+a*dmin && xc(j)>=x(i)-a*dmin && yc(j)<=y(i)+a*dmin &&
yc(j)>=y(i)-a*dmin;
        Dist=[Dist;sqrt((x(i)-xc(j))^2+(y(i)-yc(j))^2) j+(nl+1)*np];
    end
end
Dist=sortrows(Dist);
Dist=Dist(1:n,:);
end
```

ANEXO C

Esse anexo contém um código computacional desenvolvido no MATLAB para a aplicação do critério da distância na seleção dos pontos vizinhos quando o ponto pivô é um ponto de contorno. Ele é usado pelo código do anexo A.

```
function Dist=Distancia2(i,x,y,xc,yc,dmin,n,varargin)
%Função que aplica o critério das distâncias nos pontos de contorno
%i - posição do ponto pivô
%x,y - coordenadas dos pontos internos
%xc,yc - coordenadas dos pontos de contorno
%n - quantidade de pontos necessários
Dist=[1000000 0];
pos=[];
np=length(x);
npc=length(xc);
a=3.5;
if nargin-7>0;
    ref=varargin;
    nl=length(ref{1});
else nl=0;
end
for j=1:np;
    if x(j)<=xc(i)+a*dmin && x(j)>=xc(i)-a*dmin && y(j)<=yc(i)+a*dmin &&
y(j)>=yc(i)-a*dmin;
        Dist=[Dist;sqrt((xc(i)-x(j))^2+(yc(i)-y(j))^2) j];
        pos=[pos j];
    end
end
Distc=[1000000 0];
for j=1:npc;
    if xc(j)<=xc(i)+a*dmin && xc(j)>=xc(i)-a*dmin && yc(j)<=yc(i)+a*dmin &&
yc(j)>=yc(i)-a*dmin && i~=j;
        Distc=[Distc;sqrt((xc(i)-xc(j))^2+(yc(i)-yc(j))^2) j+(nl+1)*np];
    end
end
Distc=sortrows(Distc);
Dist=[Dist;Distc(1:2,:)];
tD=length(pos);
if nl>0;
    for k=1:nl;
        for j=1:tD;
            loc=pos(j);
            if ref{1}{k}(loc,1)<=xc(i)+a*dmin && ref{1}{k}(loc,1)>=xc(i)-
a*dmin && ref{1}{k}(loc,2)<=yc(i)+a*dmin && ref{1}{k}(loc,2)>=yc(i)-a*dmin;
                Dist=[Dist;sqrt((xc(i)-ref{1}{k}(loc,1))^2+(yc(i)-
ref{1}{k}(loc,2))^2) loc+k*np];
            end
        end
    end
end
Dist=sortrows(Dist);
Dist=Dist(1:n,:);
end
```