UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ROGÉRIO ZOLIN BERTECHINI

ANÁLISE ELÁSTICA LINEAR DE PLACAS DELGADAS ISOTRÓPICAS UTILIZANDO O ELEMENTO FINITO TRIANGULAR

CAMPO MOURÃO

2022

ROGÉRIO ZOLIN BERTECHINI

ANÁLISE ELÁSTICA LINEAR DE PLACAS DELGADAS ISOTRÓPICAS UTILIZANDO O ELEMENTO FINITO TRIANGULAR

Linear elastic analysis of isotropic thin plates by triangular finite element

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Civil ao Curso Superior de Engenharia Civil do Departamento Acadêmico de Construção Civil da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Leandro Waidemam

CAMPO MOURÃO

2022

ROGÉRIO ZOLIN BERTECHINI

ANÁLISE LINEAR ELÁSTICA DE PLACAS DELGADAS ISOTRÓPICAS UTILIZANDO O ELEMENTO FINITO TRIANGULAR

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Civil ao Curso Superior de Engenharia Civil do Departamento Acadêmico de Construção Civil da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Leandro Waidemam

Data de aprovação: 17 de Novembro de 2022

Leandro Waidemam
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

---

Jorge Luís Nunes de Góes
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

---

Marcelo Rodrigo Carreira
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

CAMPO MOURÃO
2022

Aos meus pais Pedro e Dolores (in memoriam),
por todos os momentos de ausência.

AGRADECIMENTOS

Agradeço primeiramente à Deus, que jamais me abandonou, mesmo em meio à todas as dificuldades que se fizeram presentes no decorrer de minha vida, e por me dar saúde para continuar seguindo em frente.

Agradeço também à minha mãe Dolores Aparecida Zolin Bertechini (*in memoriam*) por depositar sua confiança na minha capacidade (quando por muitas vezes, já nem eu mesmo confiava), além de ter sido minha mais leal amizade, bem como um dos meus maiores exemplos de caráter, juntamente com meu pai.

Ao meu pai Pedro Bertechini, por me ensinar a perseverar e a continuar seguindo em frente, mantendo a calma. E, ainda, por fornecer todo o apoio necessário para que eu não desistisse dos meus objetivos no decorrer desta graduação.

Aos demais familiares, pelo incentivo e apoio.

Ao meu orientador, Prof. Dr. Leandro Waidemam, o responsável por despertar em mim a curiosidade pela análise de estruturas e pelo Método dos Elementos Finitos. Agradeço pela grande contribuição na minha formação acadêmica, pela amizade, pela paciência e sobretudo pela excelente orientação neste trabalho.

Ao Prof. Dr. Jorge Luís Nunes de Góes, tutor do PET Civil, programa que fiz parte durante um breve período, porém de grande relevância em meu desenvolvimento acadêmico e pessoal. Agradeço por seus conselhos, conhecimentos transmitidos e confiança.

Ao Prof. Dr. Wellington José Corrêa, por ser sempre solícito quando precisei de ajuda com o LaTeX e com as formalidades matemáticas.

A todos os professores do curso de engenharia civil da UTFPR-CM pelos conhecimentos passados ao longo desses anos e ao grupo PET Civil pelo acolhimento nessa incrível família e pelos bons momentos vividos, os quais jamais esquecerei.

À todas as amizades construídas no decorrer de minha vida, podendo citar: Ana Caroline, Bruno, Gabriel, Henrique, Igor, Jaqueline, Maibuk, Maria Fernanda, Milena e Willyan. E, em especial à Laysa Samara e Ana Paula Vieira por estarem presentes tanto nos momentos bons quanto nos não tão bons assim, e à Guilherme Zarski, para mim exemplo de determinação e perseverança.

E, por fim, a todos que direta ou indiretamente fizeram parte da minha vida acadêmica e contribuíram para que este trabalho se tornasse possível.

“Faça o que puder, com o que tiver, onde estiver.”

- Theodore Roosevelt

RESUMO

O presente trabalho tem como objetivo apresentar uma formulação matemática embasada no Método dos Elementos Finitos (MEF) capaz de realizar a análise linear de placas delgadas. As placas em questão são discretizadas fazendo-se o uso do elemento finito triangular contando com três graus de liberdade em cada um dos três nós do elemento, culminando em um total de nove graus de liberdade por elemento, sendo o campo de deslocamentos no sentido perpendicular ao plano médio da placa aproximado por um polinômio cúbico contendo nove termos. De modo a tornar válida a formulação apresentada, foi implementado um algoritmo computacional no *software* MATLAB® capaz de realizar tal análise para diversas condições de carregamento e vinculação. Ao final, elaborou-se e executou-se alguns exemplos, os quais foram submetidos à uma análise quantitativa e qualitativa comparando-os com resultados fornecidos pelas tabelas de Bares e Czérny, pelo *software* SCIA Engineer, bem como por outros autores.

Palavras-chave: Método dos Elementos Finitos; análise linear de estruturas; placas delgadas.

ABSTRACT

The present work has the purpose of presenting a mathematical formulation based on Finite Element Method (FEM) able to perform thin plates linear analysis. The concerned plates are discretized using the triangular finite element with three degrees of freedom in each one of the three nodes, culminating in a total of nine degrees of freedom per element, and the displacement field on the perpendicular direction of the medium plan of the plate is approximated by a cubic polynomial containing nine terms. In order to make valid the presented formulation, a computational algorithm has been implemented in MATLAB® software, able to perform such analysis for several load and boundary conditions. At the end, it was elaborated and performed some examples, which where submitted at a qualitative and quantitative analysis comparing them with the results provided by the Bares and Czérny tables, SCIA Engineer, as well as by other authors.

Keywords: Finite Element Method; linear analysis of structures; thin plates.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DCL	Diagrama de corpo livre
EDP	Equação diferencial parcial
ELS	Estado limite de serviço
ELU	Estado limite último
MATLAB®	*Matrix Laboratory*
MEF	Método dos Elementos finitos
PTV	Princípio dos Trabalhos Virtuais

LISTA DE SÍMBOLOS

$\{\sigma\}$	Vetor que contém as tensões
σ	Tensão normal
τ	Tensão de cisalhamento
$\{\epsilon\}$	Vetor que contém as deformações
ϵ	Deformação normal
γ	Deformação angular
Q	Força cortante
M	Momento
t	Espessura da placa
u	Deslocamento linear na direção x
v	Deslocamento linear na direção y
w	Função que fornece a deflexão da placa na direção z
θ	Deslocamento angular no plano $x-y$
φ	Deslocamentos angulares nos planos $x-z$ e $y-z$
$[E]$	Matriz constitutiva
E	Módulo de elasticidade longitudinal (Módulo de Young)
ν	Coeficiente de Poisson
p	Carga distribuída na área da placa
D	Rigidez da placa à flexão
U_e	Trabalho externo exercido por uma força
U_i	Energia interna de deformação
A	Área da placa
R	Reação vertical na borda da placa
$[\phi]$	Matriz que contém as funções de forma
$[L]$	Matriz que contém os operadores diferenciais
$[k]$	Matriz de rigidez do elemento
$\{d\}$	Vetor de deslocamentos nodais
ξ_1	Primeira coordenada homogênea
ξ_2	Segunda coordenada homogênea
ξ_3	Terceira coordenada homogênea
$[\psi]$	Matriz contendo as coordenadas homogêneas do polinômio aproximador
$\{\alpha\}$	Vetor contendo as constantes do polinômio aproximador

SUMÁRIO

1 INTRODUÇÃO

A análise estrutural é de suma importância no âmbito da construção civil, pois a mesma possibilita a realização de uma estrutura viável economicamente, e sobretudo segura para seus devidos fins. Nos projetos estruturais deve-se garantir que a estrutura permaneça íntegra durante sua vida útil prevista em norma, para tal é necessário assegurar dois estados limites, o ELU (Estado Limite Último), que garante que a estrutura não sofrerá colapso, e o ELS (Estado Limite de Serviço), que é imposto basicamente para garantir que a estrutura não apresente deformações excessivas, não comprometendo a usabilidade e conforto por parte dos usuários.

Em uma estrutura geralmente são usados vários tipos de elementos estruturais, sendo os mais comuns os pilares, as vigas, as estacas, os blocos de fundação, as sapatas, e as lajes, que por sua vez, podem ser subdivididas em treliçadas, nervuradas ou maciças, sendo a última configuração objetos de estudo deste trabalho.

Pela definição de Bastos (2021) as lajes são elementos planos, bidimensionais e de superfície, nos quais duas dimensões predominam sobre a terceira, sendo esta a espessura. Nesses elementos as cargas atuam de maneira perpendicular à sua superfície, distribuídas em uma área, linearmente, ou de forma pontual, cargas estas que podem ser transmitidas para as vigas, ou de forma menos comum, diretamente aos pilares.

As lajes maciças são estudadas por meio das teorias de placas, sendo subdivididas em placas espessas ou delgadas, com grandes ou pequenas deformações. A escolha de qual teoria aplicar se deve à geometria do problema que se quer resolver, no entanto, algumas teorias são mais complexas que outras, necessitando de considerações adicionais.

Todas as teorias de placas, entretanto, envolvem a resolução de equações diferenciais parciais, que torna o processo trabalhoso e muitas vezes inviável de forma analítica, fazendo com que, no passado, fossem utilizadas séries de Fourier para aproximar as soluções. Todavia, o processo para esse tipo de resolução é bastante complexo e demorado.

Devido à tal complexidade, com o aprimoramento dos computadores, foram desenvolvidos métodos numéricos capazes de encontrar as soluções dessas equações diferenciais de forma muito mais rápida, o que trouxe grande evolução para a engenharia estrutural.

A análise estrutural utilizada nos dias atuais possui quatro níveis de abstração, sendo eles a estrutura real, ou seja, a estrutura em sua integridade, conforme será quando for construída; o modelo estrutural, que é um diagrama composto por linhas e planos, indicando as forças, momentos e outros agentes físicos agindo nele, conhecido como diagrama de corpo livre (DCL); o modelo discreto, fruto do processo chamado de discretização, que pode ser realizado de diversas maneiras a depender do método de análise que se pretende utilizar; e, por último, o modelo computacional, que se trata da implementação computacional do método de análise desejado (MARTHA, 2010).

Um dos métodos numéricos mais utilizados na análise estrutural atualmente é o Método dos Elementos Finitos (MEF), presente em praticamente todos os *softwares* comerciais voltados à esse fim, tendo superado métodos como o de Rayleigh-Ritz, de Galerkin, diferenças finitas e resíduos ponderados. O MEF se sobressai aos dois primeiros devido à dificuldade na obtenção de funções aproximadoras que estes apresentam, além do fato de que essas funções tinham que ter ordem elevada, tornando os cálculos demasiadamente árduos ou em algumas circunstâncias até mesmo impraticáveis (ASSAN, 2003).

O MEF iniciou-se sendo utilizado para a análise de estruturas reticuladas, não obstante o mesmo se mostrou também extremamente útil na análise de estruturas contínuas em duas ou três dimensões, bem como em problemas contendo não-linearidades físicas, inclusive fora desse campo, tais como engenharia geotécnica, interações fluido-mecânicas e análises de fluxo térmico e hidráulico (VAZ, 2011).

Tendo em vista este cenário, o presente trabalho apresenta uma formulação do MEF direcionada à análise linear de placas delgadas isotrópicas.

1.1 Objetivos

Os objetivos geral e específicos do trabalho são apresentados nas seções 1.1.1 e 1.1.2.

1.1.1 Objetivo Geral

Apresentar a formulação matemática e um código computacional para a análise estática linear de placas delgadas via MEF usando o elemento finito triangular.

1.1.2 Objetivos específicos

- Deduzir, por meio da teoria das placas delgadas, a equação diferencial parcial que descreve o fenômeno físico da deflexão de uma placa retangular, e as condições de contorno intrínsecas ao problema;
- Descrever o método dos elementos finitos aplicado ao problema, considerando, para isso, o elemento finito triangular;
- Desenvolver um código implementado no *software* MATLAB® fundamentado no MEF para a solução do problema proposto;
- Comparar os resultados obtidos com o código desenvolvido com as tabelas clássicas, com outros autores e com o *software* comercial SCIA Engineer, de modo a verificar o correto desenvolvimento e implementação do código computacional.

1.2 Justificativa

As placas estão contidas em uma infinidade de construções, isso se deve ao fato de que seu modo de transferir esforços é extremamente útil tendo em vista os propósitos do ser humano. Diniz et al (2013) diz que elas são as principais responsáveis por receber cargas diretamente, sendo variáveis ou permanentes, e as transmitem para as vigas de bordo que, por sua vez transferem-nas aos pilares.

Em um único pavimento de um edifício, por exemplo, pode-se ter várias lajes maciças, não se limitando, todavia, a esse tipo de edificação, visto que elas estão presentes também em pontes, viadutos, passarelas suspensas, barracões industriais, etc.

Existe um interesse especial em saber-se o quanto uma placa irá defletir, pois uma deflexão exagerada desencadeará desconforto e insegurança aos habitantes da edificação. A ABNT NBR 6118:2014 impõe uma limitação com base nessas premissas, o que é chamado de estado limite de serviço (ELS), onde são levados em consideração a aceitabilidade sensorial das pessoas, efeitos em elementos não estruturais, tais como paredes e forros que possam sofrer danos com a movimentação da placa, e efeitos nos próprios elementos estruturais.

Entretanto, o interesse não se limita na deflexão, há ainda a necessidade da obtenção das tensões atuantes na placa e das reações de apoio da mesma, a fim de que se possa dimensionar a mesma e seus apoios para que apresente segurança aos ocupantes e também de modo que o custo de execução não seja maior que o necessário.

O avanço dos computadores permitiu que cálculos estimados por meio de métodos executados manualmente, como expansão em séries, fossem substituídos por métodos numéricos como o MEF, que quando implementado computacionalmente com uma precisão adequada fornece resultados muito próximos dos exatos.

Segundo Cook et al (2002), os erros são intrínsecos ao MEF, exceto nos casos onde o problema é simples a ponto de não necessitar do uso do método. No entanto, uma série de fatores podem reduzir o erro envolvido nos cálculos, tais como o tipo, o tamanho e a forma dos elementos, o tamanho da malha, e ainda quais condições de contorno e restrições são impostas ao mesmo.

Além disso, o presente trabalho visa também ser uma fonte de aprendizado, tendo em vista que muitos engenheiros que se propõem a utilizar programas computacionais baseados em elementos finitos acabam por ter dificuldades, dado que a aprendizagem é feita sem a base conceitual necessária, já que buscam simplesmente aprender a usar os softwares e não o MEF (ALVES FILHO, 2000).

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas as teorias referentes ao assunto abordado no trabalho, tais como as teorias da elasticidade bidimensional e de placas delgadas e o Método dos Elementos Finitos.

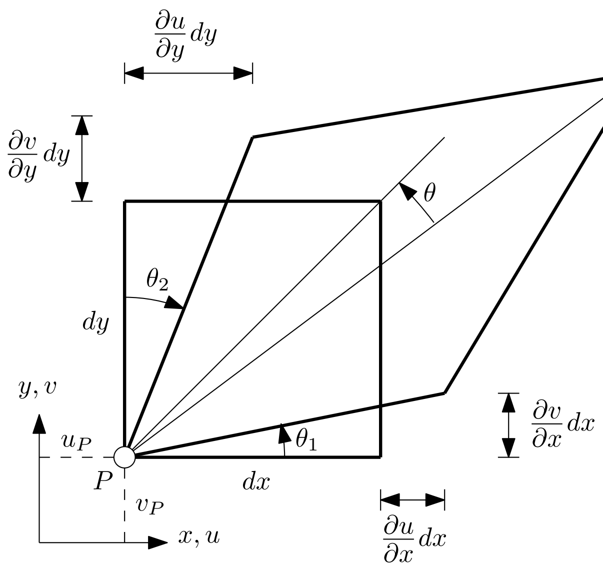
2.1 Teoria da elasticidade bidimensional

Como as placas são elementos em que duas direções predominam sobre uma terceira, parte-se da análise da teoria da elasticidade bidimensional.

2.1.1 Relações deformação-deslocamento

Na análise das deformações por meio teoria da elasticidade bidimensional no plano $x-y$, considera-se que os deslocamentos ao longo do eixo z são desprezíveis, e que as seções transversais ao plano deformam todas da mesma maneira, dessa forma para sua análise é tomado um elemento infinitesimal genérico no plano $x-y$, e analisa-se o campo de deslocamentos $u(x,y)$ e $v(x,y)$, nas direções x e y, respectivamente (OÑATE, 2009). O elemento analisado se encontra na Figura 1.

Figura 1 – Deformação de um elemento plano infinitesimal



Fonte: Adaptado de Oñate (2009, p. 120)

Sabe-se da resistência dos materiais, que a deformação normal é dada pela equação:

$$\epsilon = \frac{\ell - \ell_0}{\ell_0} \tag{1}$$

Onde,

ℓ: Comprimento final;

ℓ_0: Comprimento inicial.

Assim, analisando o elemento infinitesimal apresentado na Figura 1, tem-se:

$$\epsilon_x = \frac{dx + \frac{\partial u}{\partial x}dx - dx}{dx} \tag{2}$$

Colocando dx em evidência, obtém-se:

$$\epsilon_x = \frac{\partial u}{\partial x} \tag{3}$$

Por meio de um processo análogo é possível obter:

$$\epsilon_y = \frac{\partial v}{\partial y} \tag{4}$$

Analisando as deformações tangenciais tem-se,

$$\mathrm{tg}(\theta_1) = \frac{\frac{\partial v}{\partial x}dx}{dx + \frac{\partial u}{\partial x}dx} \tag{5}$$

e:

$$\mathrm{tg}(\theta_2) = \frac{\frac{\partial u}{\partial y}dy}{dy + \frac{\partial v}{\partial y}dy} \tag{6}$$

Como assumiu-se que a placa está submetida à pequenas deformações pode-se considerar que,

$$\mathrm{tg}(\theta_1) \approx \theta_1 \tag{7}$$

e que,

$$\mathrm{tg}(\theta_2) \approx \theta_2 \tag{8}$$

o que implica que os termos $\frac{\partial u}{\partial x}$ e $\frac{\partial v}{\partial y}$ são aproximadamente zero devido à geometria do problema. Assim, simplificando os termos dx e dy, respectivamente, as Equações 5 e 6 se tornam:

$$\theta_1 = \frac{\partial v}{\partial x} \tag{9}$$

$$\theta_2 = \frac{\partial u}{\partial y} \tag{10}$$

Não é difícil perceber que a deformação tangencial no elemento diferencial é dada pela Equação 11:

$$\gamma_{xy} = \theta_1 + \theta_2 \tag{11}$$

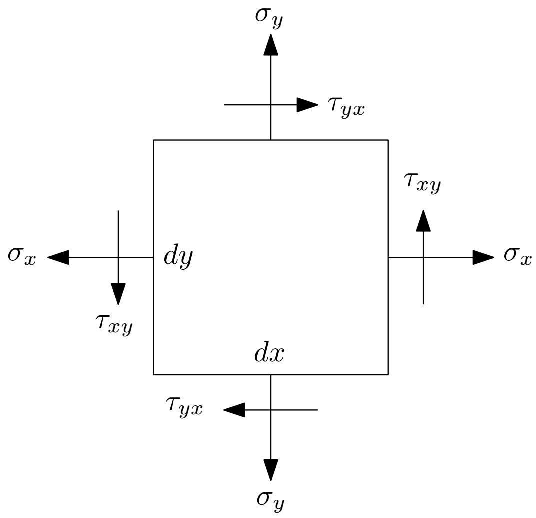
Substituindo os valores das Equações 9 e 10 na Equação 11, tem-se a deformação tangencial.

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \tag{12}$$

2.1.2 Relações tensão-deformação

Para realizar-se a análise das tensões atuantes em uma placa utiliza-se do chamado estado plano de tensões (Figura 2), caracterizado por duas componentes de tensão normal σ_x e σ_y e duas componentes de tensão tangencial $\tau_{xy} = \tau_{yx}$.

Figura 2 – O elemento plano de tensões



Fonte: Adaptado de Oñate (2009, p.121)

Tendo em vista que a aplicação de uma força em um elemento estrutural gera tensões que se refletem em deformações no material, é possível estabelecer relações entre elas, obtidas por meio da Lei de Hooke generalizada. As chamadas relações constitutivas estão dispostas nas Equações 13, 14 e 15, onde E é o módulo de elasticidade longitudinal, ou módulo de Young, e ν é o coeficiente de Poisson, sendo ambos propriedades mecânicas do material da placa, quando a mesma é considerada isotrópica (TIMOSHENKO; GOODIER, 1934).

$$\sigma_x = \frac{E}{1-\nu^2}(\epsilon_x + \nu\epsilon_y) \tag{13}$$

$$\sigma_y = \frac{E}{1-\nu^2}(\epsilon_y + \nu\epsilon_x) \tag{14}$$

$$\tau_{xy} = \frac{E}{2(1+\nu)}\gamma_{xy} \tag{15}$$

Pode-se também representar as Equações 13, 14 e 15 em notação matricial, representada por:

$$\{\sigma\} = [E]\{\epsilon\} \tag{16}$$

onde:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{17}$$

$$[E] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{(1-\nu)}{2} \end{bmatrix} \tag{18}$$

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{19}$$

2.2 Teoria de placas delgadas

De acordo com Timoshenko e Woinowsky-Krieger (1959), as placas são classificadas conforme sua espessura, e as propriedades de flexão das mesmas também dependem dela. Assim há três classes principais de placas: as placas delgadas com pequena deflexão, as placas delgadas com grande deflexão e as placas espessas.

Com base na ampla gama de aplicações, para o presente trabalho adota-se a teoria das placas delgadas com pequenas deflexões.

Sendo assim, Martinelli et al (1986), diz que as placas são consideradas delgadas quando a relação “d/a” fica compreendida entre $1/5$ e $1/100$, onde d é a espessura da placa e a a sua menor dimensão.

Essa teoria assume três hipóteses básicas:

1. O plano médio da placa não sofre deformação quando a placa é submetida à flexão;
2. Segmentos de reta posicionados inicialmente normais ao plano médio da placa permanecem normais ao plano médio da mesma após submetida à flexão;
3. As tensões normais ao plano médio da placa podem ser desconsideradas.

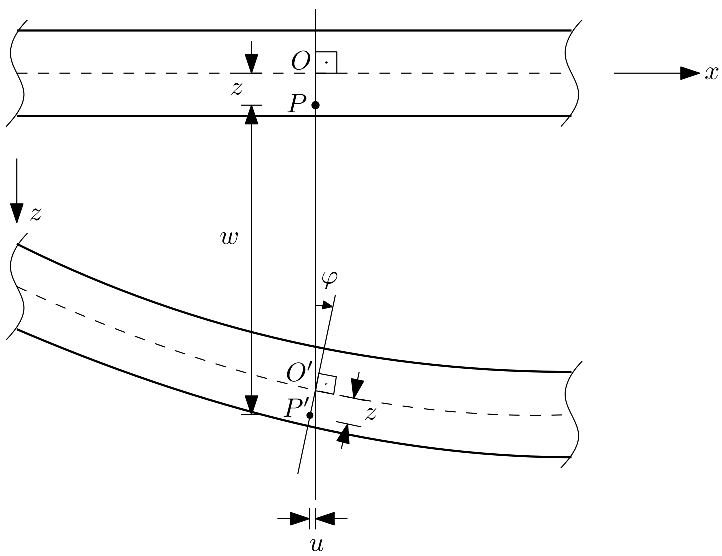
Assumindo essas três hipóteses, o deslocamento pode ser expresso em termos de duas coordenadas apenas, ou seja, se trata de um problema bidimensional.

Sendo que, ainda segundo Timoshenko e Woinowsky-Krieger (1959) a segunda hipótese é equivalente ao se desprezar o efeito das forças cortantes em placas, assim em casos de placas com buracos, por exemplo, a mesma não deve ser desprezada, porém nos outros casos é uma hipótese válida. Da mesma forma a primeira hipótese não se mostra verdadeira no caso em que se tem cargas aplicadas de maneira perpendicular às laterais da placa, pois esses esforços geram tensões e consequentemente deformações em seu plano médio.

2.2.1 Relações deformação-deslocamento aplicadas ao problema de placas delgadas

Como citado anteriormente, assume-se que um segmento de reta inicialmente normal ao plano médio da placa continua normal a esse plano após a mesma sofrer flexão, e ainda que um ponto situado no plano médio da placa sofre deslocamentos apenas no eixo z, não sofrendo nenhum deslocamento nos eixos x e y (WAIDEMAM, 2004). Dessa forma, toma-se um ponto P genérico, pertencente à um segmento de reta normal ao plano médio da placa e situado à uma distância z do mesmo (Figura 3). O ponto sofre deslocamentos u e v na direção x e y respectivamente, assim como um deslocamento w na direção z, o qual é função de x e y e chamado de deflexão.

Figura 3 – Deslocamento de um ponto situado em uma seção normal ao plano da placa



Fonte: Adaptado de Waidemam (2004 p. 51)

Dessa forma, pode-se dizer baseado na definição de derivada, que:

$$\mathrm{tg}(\varphi) = \frac{-u}{z} \tag{20}$$

Organizando a Equação 20 e explicitando u, tem-se:

$$u = -z\frac{\partial w}{\partial x} \tag{21}$$

De maneira análoga, para o plano $y - z$, chega-se a relação:

$$v = -z\frac{\partial w}{\partial y} \tag{22}$$

Substituindo as Equações 21 e 22, nas Equações 3, 4 e 12, respectivamente, obtém-se as seguintes relações:

$$\epsilon_x = \frac{\partial}{\partial x}\left(-z\frac{\partial w}{\partial x}\right) \tag{23}$$

$$\epsilon_y = \frac{\partial}{\partial y}\left(-z\frac{\partial w}{\partial y}\right) \tag{24}$$

$$\gamma_{xy} = \frac{\partial}{\partial x}\left(-z\frac{\partial w}{\partial y}\right) + \frac{\partial}{\partial y}\left(-z\frac{\partial w}{\partial x}\right) \tag{25}$$

que se tornam:

$$\epsilon_x = -z\frac{\partial^2 w}{\partial x^2} \tag{26}$$

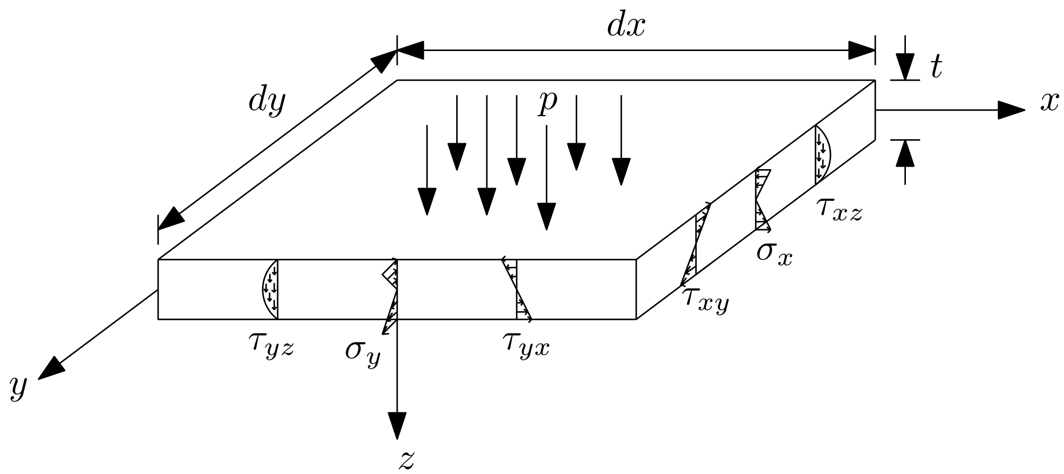
$$\epsilon_y = -z\frac{\partial^2 w}{\partial y^2} \tag{27}$$

$$\gamma_{xy} = -2z\frac{\partial^2 w}{\partial x \partial y} \tag{28}$$

2.2.2 Equilíbrio de uma placa

A Figura 4 mostra uma placa, sobre a qual atua uma carga distribuída p estática. Admite-se que a placa em questão tem uma espessura t, e lados de comprimento dx e dy, respectivamente.

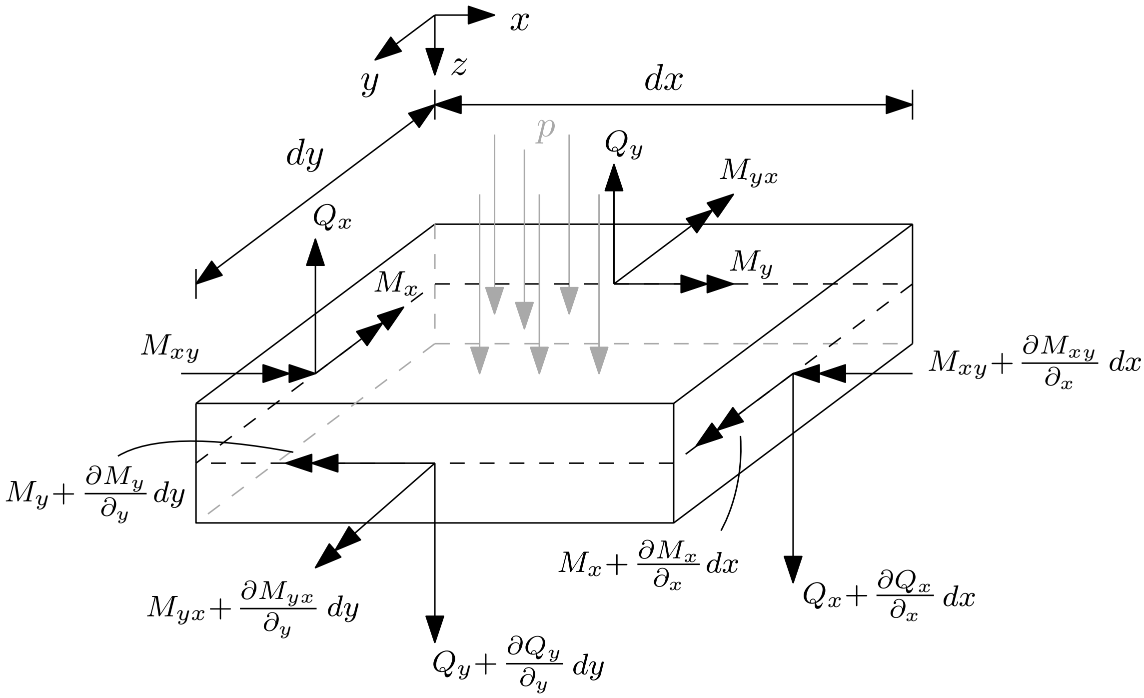
Figura 4 – Placa submetida à um carregamento e as devidas tensões atuantes



Fonte: Adaptado de Waidemam (2004, p. 53)

Analisando-se a Figura 4, pode-se observar que são gerados três tipos de tensões na placa. As tensões σ_x e σ_y, são responsáveis pelos momentos fletores M_x e M_y respectivamente, da mesma forma τ_{xy} e τ_{yx} geram os momentos torçores M_{xy} e M_{yx}, ambas variando linearmente em relação à z. Por fim, tem-se as tensões τ_{yz} e τ_{xz}, que variam quadraticamente em relação ao eixo z e acarretam nas respectivas forças cortantes Q_y e Q_x. Os esforços oriundos das tensões atuantes podem ser observados na Figura 5.

Figura 5 – Esforços atuantes em uma placa submetida a um carregamento distribuído



Fonte: Adaptado de Leal (2015, p. 27)

Sabe-se da resistência dos materiais que:

$$F = \int \sigma dA \tag{29}$$

Sabe-se também da mecânica que a intensidade do momento é dada por:

$$M = F \cdot s \tag{30}$$

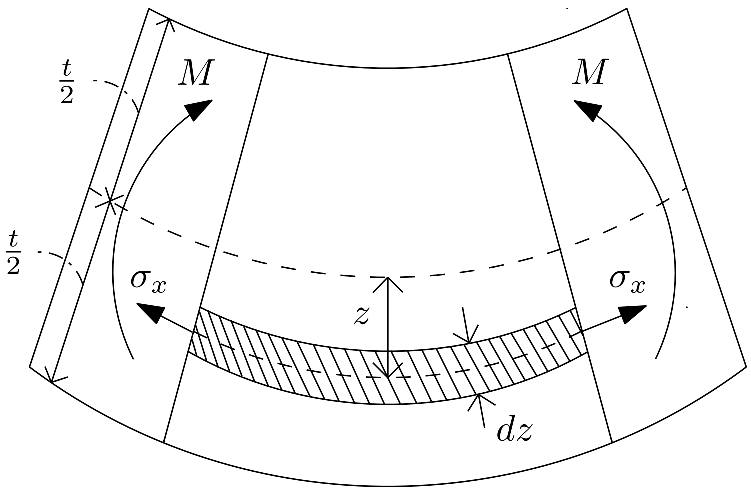
onde s é a distância entre o ponto de cálculo do momento e a linha de ação da força, tomada de forma perpendicular.

Aplicando-se a Equação 29 na Equação 30, tem-se:

$$M = \int \sigma s dA \tag{31}$$

Assim, tomando-se um pedaço da seção transversal da placa (Figura 6), pode-se integrar as tensões em relação à um elemento diferencial dz encontrado à uma distância z do plano médio da mesma e obter-se os momentos por unidade de comprimento.

Figura 6 – Representação gráfica da integral de momento fletor



Fonte: Adaptado de Timoshenko (1959, p. 5)

Dessa forma tem-se os seguintes momentos:

$$M_x = \int_{-t/2}^{t/2} \sigma_x z dz \tag{32}$$

$$M_y = \int_{-t/2}^{t/2} \sigma_y z dz \tag{33}$$

$$M_{xy} = \int_{-t/2}^{t/2} \tau_{xy} z dz \tag{34}$$

$$M_{yx} = \int_{-t/2}^{t/2} \tau_{xy} z dz \tag{35}$$

sendo que, de acordo com Costa (1986), devido às tensões $\tau_{xy} = \tau_{yx}$:

$$M_{yx} = -M_{xy} \tag{36}$$

Além dos momentos, pode-se obter de maneira análoga as forças cortantes:

$$Q_x = \int_{-t/2}^{t/2} \tau_{xz} dz \tag{37}$$

$$Q_y = \int_{-t/2}^{t/2} \tau_{yz} dz \tag{38}$$

Os esforços atuantes na placa carregada podem ser representados conforme a Figura 5, e a partir disso pode-se realizar o equilíbrio de esforços na placa.

Realizando o equilíbrio de forças na direção "z", tem-se:

$$\left(Q_x + \frac{\partial Q_x}{\partial x} \cdot dx\right) dy + \left(Q_y + \frac{\partial Q_y}{\partial y} \cdot dy\right) dx - Q_y \cdot dx - Q_x \cdot dy + p \cdot dy \cdot dx = 0 \tag{39}$$

Colocando os termos em evidência e dividindo todos eles por $dy \cdot dx$, é obtido:

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} = -p \tag{40}$$

Fazendo o equilíbrio de momentos em torno de "x":

$$\begin{aligned} M_y \cdot dx - \left(M_y + \frac{\partial M_y}{\partial y} \cdot dy\right) dx - M_{xy} \cdot dy + \left(M_{xy} + \frac{\partial M_{xy}}{\partial x} \cdot dx\right) dy - Q_x \cdot dy \cdot \frac{dy}{2} \\ + \left(Q_x + \frac{\partial Q_x}{\partial x} \cdot dx\right) dy \cdot \frac{dy}{2} + \left(Q_y + \frac{\partial Q_y}{\partial y} \cdot dy\right) dx \cdot dy + p \cdot dx \cdot dy \cdot \frac{dy}{2} = 0 \end{aligned} \tag{41}$$

Os termos que contém o produto de mais de dois elementos infinitesimais podem ser desprezados, pois se tornam demasiadamente pequenos em comparação com o restante dos termos. Ainda, fazendo de maneira análoga ao que foi realizado com a equação de equilíbrio de forças, tem-se:

$$\frac{\partial M_y}{\partial y} \cdot dy \cdot dx - \frac{\partial M_{xy}}{\partial x} \cdot dy \cdot dx = Q_y \cdot dy \cdot dx \tag{42}$$

Por fim, dividindo todos os termos por $dy \cdot dx$, tem-se:

$$\frac{\partial M_y}{\partial y} - \frac{\partial M_{xy}}{\partial x} = Q_y \tag{43}$$

Finalmente, efetuando-se o equilíbrio de momentos em torno de "y":

$$\begin{aligned} - M_x \cdot dy + \left(M_x + \frac{\partial M_x}{\partial x} \cdot dx\right) dy - M_{xy} \cdot dx + \left(M_{xy} + \frac{\partial M_{xy}}{\partial y} \cdot dy\right) dx + Q_y \cdot dx \cdot \frac{dx}{2} \\ - \left(Q_y + \frac{\partial Q_y}{\partial y} \cdot dy\right) dx \cdot \frac{dx}{2} - \left(Q_x + \frac{\partial Q_x}{\partial x} \cdot dx\right) dy \cdot dx - p \cdot dy \cdot dx \cdot \frac{dx}{2} = 0 \end{aligned} \tag{44}$$

Fazendo de maneira similar ao realizado na seção anterior, tem-se:

$$\frac{\partial M_x}{\partial x} \cdot dy \cdot dx + \frac{\partial M_{xy}}{\partial y} \cdot dy \cdot dx = Q_x \cdot dy \cdot dx \tag{45}$$

e, por fim:

$$\frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y} = Q_x \tag{46}$$

Substituindo as Equações 43 e 46 na Equação 40, tem-se:

$$\frac{\partial}{\partial x}\left(\frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y}\right) + \frac{\partial}{\partial y}\left(\frac{\partial M_y}{\partial y} - \frac{\partial M_{xy}}{\partial x}\right) = -p \tag{47}$$

Realizando as operações necessárias e levando em consideração que $M_{yx} = -M_{xy}$, tem-se:

$$\frac{\partial^2 M_x}{\partial x^2} - 2 \cdot \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} = -p \tag{48}$$

Substituindo-se as Equações 26, 27 e 28 nas Equações 13, 14 e 15 tem-se:

$$\sigma_x = -\frac{Ez}{1-\nu^2}\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) \tag{49}$$

$$\sigma_y = -\frac{Ez}{1-\nu^2}\left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right) \tag{50}$$

$$\tau_{xy} = -\frac{Ez}{(1-\nu)}\left(\frac{\partial^2 w}{\partial x \partial y}\right) \tag{51}$$

Para obter-se os momentos, basta substituir as Equações 49, 50 e 51 nas Equações 32, 33 e 34 respectivamente e resolver as integrais. Dessa forma:

$$M_x = -\frac{Et^3}{12(1-v^2)}\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) \tag{52}$$

$$M_y = -\frac{Et^3}{12(1-v^2)}\left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right) \tag{53}$$

$$M_{yx} = -\frac{Et^3}{12(1+v)}\left(\frac{\partial^2 w}{\partial x \partial y}\right) \tag{54}$$

Como já dito anteriormente, $M_{yx} = -M_{xy}$, assim M_{xy} fica:

$$M_{xy} = \frac{Et^3}{12(1+v)}\left(\frac{\partial^2 w}{\partial x \partial y}\right) \tag{55}$$

Pode-se definir:

$$D = \frac{Et^3}{12(1-v^2)} \tag{56}$$

Desse modo, as Equações 52, 53, 54 e 55 podem ser escritas na forma:

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) \tag{57}$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right) \tag{58}$$

$$M_{yx} = -D(1+\nu)\left(\frac{\partial^2 w}{\partial x \partial y}\right) \tag{59}$$

$$M_{xy} = D(1+\nu)\left(\frac{\partial^2 w}{\partial x \partial y}\right) \tag{60}$$

Substituindo-se as Equações 57, 58, e 60 na Equação 47, é obtido:

$$\frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\left(-D\left(\frac{\partial^2 w}{\partial x^2}+\nu\frac{\partial^2 w}{\partial y^2}\right)\right)\right]-$$
$$-2\cdot\frac{\partial}{\partial x}\left[\frac{\partial}{\partial y}\left(D(1+\nu)\left(\frac{\partial^2 w}{\partial x\partial y}\right)\right)\right]+ \tag{61}$$
$$+\frac{\partial}{\partial y}\left[\frac{\partial}{\partial y}\left(-D\left(\frac{\partial^2 w}{\partial y^2}+\nu\frac{\partial^2 w}{\partial x^2}\right)\right)\right]=-p$$

que após realizadas as devidas simplificações se torna:

$$\frac{\partial^4 w}{\partial x^4}+2\cdot\frac{\partial^4 w}{\partial x^2\partial y^2}+\frac{\partial^4 w}{\partial y^4}=\frac{p}{D} \tag{62}$$

a qual é a equação que descreve a deflexão w da placa em termos de suas coordenadas x e y, sendo uma EDP de quarta ordem, não homogênea e linear, apresentando coeficientes constantes. E que, de acordo com Ventsel e Krauthammer (2001) foi obtida primeiramente por Lagrange em 1811.

2.2.3 Condições de contorno

Como acontece com a maioria das equações diferenciais, para se resolver a EDP obtida com a teoria de placas delgadas (Equação 62) é necessário que se imponha condições de contorno, ou seja, valores conhecidos da função $w(x,y)$ ou de suas derivadas em pontos com coordenadas x e y conhecidas.

2.2.3.1 Borda simplesmente apoiada

Sabe-se que em uma borda apoiada a translação na direção do eixo z é impedida e os momentos são nulos, por ser uma vinculação do tipo rótula, assim considerando-se uma placa retangular com a lateral paralela ao eixo y simplesmente apoiada, sem a presença de momentos fletores aplicados, onde $x=a$, tem-se:

$$w=0 \tag{63}$$

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) = 0 \tag{64}$$

2.2.3.2 Borda engastada

Na borda engastada, ambos os movimentos, de translação e rotação são impedidos, dessa forma, caso a borda paralela ao eixo "y" seja engastada, para $x = a$ tem-se:

$$w = 0 \tag{65}$$

$$\frac{\partial w}{\partial x} = 0 \tag{66}$$

Ainda, nesse caso, a reação na borda fica:

$$R_x = Q_x \tag{67}$$

2.2.3.3 Borda livre

Quando não há vínculos na borda da placa, não se tem momento fletor, momento torçor, e nem esforços cortantes, dessa forma:

$$M_x = 0; \qquad M_y = 0; \qquad M_{xy} = 0; \qquad Q_x = 0; \qquad Q_y = 0; \tag{68}$$

Coforme Timoshenko e Woinowsky-Krieger (1959) para a resolução da Equação 62 são necessárias somente duas condições de contorno. Dessa forma pode-se combinar a força cortante Q_x com o momento torçor $\frac{\partial M_{xy}}{\partial y}$ e obter:

$$Q_x - \frac{\partial M_{xy}}{\partial y} = 0 \tag{69}$$

Substituindo os valores de Q_x e de $\frac{\partial M_{xy}}{\partial y}$ na Equação 69 e realizando algumas manipulações algébricas chega-se em:

$$\frac{\partial^3 w}{\partial x^3} + (2-\nu)\frac{\partial^3 w}{\partial x \partial y^2} = 0 \tag{70}$$

Por fim, para o momento fletor tem-se:

$$\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} = 0 \tag{71}$$

2.3 Método dos Elementos Finitos (MEF)

Sabe-se que, na resistência dos materiais contida nos cursos de graduação, geralmente é aprendido a análise de vigas, onde todas as fórmulas são obtidas através da solução analítica da equação diferencial que rege o fenômeno. No entanto, isso é possível graças ao problema ser unidimensional, o que não ocorre quando se trata de elementos de placa, que por sua vez são bidimensionais, fazendo com que a equação diferencial que governa os fenômenos físicos relacionados as mesmas seja muito mais complexa que a de vigas (ALVES FILHO, 2013).

Alves Filho (2013) diz ainda que tal complexidade se resume na obtenção da solução exata apenas para casos muito específicos, com certas condições de contorno, ou casos em que há um grande número de simplificações, o que se torna inviável na prática pois os problemas do mundo real ficam demasiadamente distantes do modelo adotado. Azevedo (2003) afirma também que muitas vezes se recorria às séries de Fourier para resolver esses problemas, porém da mesma forma as resoluções eram limitadas à casos os quais a geometria era simples, por conta da complexidade dos cálculos envolvidos.

Para sanar esse problema foi desenvolvido o Método dos Elementos Finitos (MEF), que serve tanto para análise de estruturas reticuladas, como vigas e pórticos, como para análise de meios contínuos como é o caso das placas (AZEVEDO, 2003).

Primeiramente o MEF foi desenvolvido por Walter Ritz em 1909, para aplicá-lo em problemas da mecânica dos sólidos deformáveis. Porém, nesse método o funcional de energia era aproximado por funções com coeficientes a serem determinados. No entanto, Richard Courant, em 1943 introduziu funções especiais definidas sobre regiões triangulares no método, fazendo com que o mesmo tivesse mais chances de sucesso, pois as incógnitas, que foram determinadas, são os valores nos pontos nodais das regiões triangulares (CAMPOS, 2006).

Após as contribuições de Courant, Argyris e Kelsey introduziram definitivamente a formulação matricial do MEF, quando aplicaram o método na análise da fuselagem e asas de aviões (ASSAN, 2003).

A formulação do MEF como é conhecida atualmente foi estabelecida por Turner, Clough, Martin e Topp, no ano de 1956, sendo que o nome Método dos Elementos Finitos foi proposto por Clough (ASSAN, 2003). Campos (2006) também afirma que a primeira vez que o método surgiu com esse nome foi com o trabalho de Clough em 1960, intitulado *The finite element method in plane stress analysis*.

Assan (2003) salienta que o MEF é conhecido desde meados da década de 50, entretanto, seu uso em projetos nas diversas áreas, além da área estrutural só foi consolidado após a evolução dos computadores, e que hoje o método está presente em diversos softwares comerciais e agregado no dia a dia dos engenheiros.

O MEF é como outros métodos numéricos, uma técnica que utiliza da discretização de um domínio, contínuo ou não, de modo que se tenha vários domínios (elementos) de dimensões finitas, daí seu nome, com forma variando dependendo da sua aplicação, e interligados por um número pequeno de pontos chamados nós, os quais podem estar presentes em maior ou menor número nos elementos, também dependendo de para qual problema se queira aplicar o método (SORIANO, 2002).

Segundo Vaz (2011), para a formulação do MEF podem ser usados o Princípio da Mínima Energia Potencial Total, o Método dos Resíduos Ponderados e o Princípio dos Deslocamentos Virtuais, que faz parte do PTV, o qual será o utilizado neste trabalho.

2.3.1 Princípio dos Trabalhos Virtuais (PTV)

Quando um corpo é submetido a uma ou mais forças, são geradas tensões em seu interior que se refletem em deformações no mesmo, esse é o princípio básico da resistência dos materiais, também conhecida como mecânica dos sólidos deformáveis (HIBBELER, 2010).

Admitindo uma força aplicada no exterior de um corpo, sabe-se que a mesma causa um certo deslocamento no corpo. Diz-se que a força externa transferiu energia ao corpo sendo este denominado trabalho externo. À energia absorvida pelo corpo em função das deformações apresentadas, dá-se o nome de energia interna de deformação ou trabalho interno. No entanto, no Princípio dos Trabalhos Virtuais (PTV), existem duas condições possíveis: a primeira diz que há forças virtuais que geram tensões virtuais, já a segunda afirma que há deslocamentos virtuais que se refletem em deformações virtuais. No caso aqui abordado os deslocamentos e deformações são virtuais.

Por fim, se estabelece ainda, pelo princípio da conservação de energia, que os trabalhos virtuais externo e interno devem ser equivalentes, como mostra a Equação 72.

$$U_i^* = U_e^* \tag{72}$$

Sabe-se que o trabalho virtual externo é tido como o produto de força por deslocamento. Já o trabalho virtual interno é tido como a integral do produto entre as tensões e deformações no volume do corpo. Como neste trabalho foi considerado uma carga distribuída na área da placa, surge o termo que contém a integral de área no trabalho causado por forças externas. Dessa forma, a Equação 72 fica:

$$\int_V \sigma\epsilon^* \, dV = Fd^* + \int_A w^* p \, dA \tag{73}$$

onde,

ϵ^*: Deformação virtual interna da placa;

σ: Tensão real interna atuante na placa;

d^*: Deslocamento virtual externo da placa;

F: Força externa atuante na placa;

w^*: Deslocamento virtual perpendicular ao plano da placa;

p: Força distribuída na área da placa.

Em função do problema bidimensional discretizado tratado nesse trabalho, a Equação 73 precisa ser escrita na forma vetorial, se tornando:

$$\int_V \{\epsilon^*\}^T \{\sigma\} \, dV = \{d^*\}^T \{F\} + \int_A w^* p \, dA \tag{74}$$

onde,

$\{\epsilon^*\}^T$: Vetor transposto de deformações virtuais internas no elemento;

$\{\sigma\}$: Vetor de tensões reais atuantes no elemento;

$\{d^*\}^T$: Vetor transposto de deslocamentos virtuais externos nodais;

$\{F\}$: Vetor de forças nodais atuantes no elemento;

É preciso relacionar os deslocamentos de cada elemento com o deslocamento de cada nó presente no mesmo, para tal utiliza-se as chamadas funções de forma, que são funções de aproximação, ou seja, por meio delas é possível obter uma aproximação do deslocamento do elemento com base nos deslocamentos dos nós. Esse procedimento pode ser representado na forma matricial:

$$w = [\phi]\{d\} \tag{75}$$

onde, w é o deslocamento do elemento, $[\phi]$ é a matriz de funções de forma e $\{d\}$ são os deslocamentos nodais.

A mesma pode ser escrita na sua forma transposta, ficando:

$$w = \{d\}^T[\phi]^T \tag{76}$$

Fazendo as Equações 3, 4 e 12 na forma matricial, tem-se:

$$\{\epsilon\} = [L]\{w\} \tag{77}$$

onde o vetor $\{\epsilon\}$ é composto pelas seguintes relações fornecidas pela teoria de placas:

$$\epsilon_x = -z\frac{\partial^2 w}{\partial x^2}$$

$$\epsilon_y = -z\frac{\partial^2 w}{\partial y^2}$$

$$\gamma_{xy} = -2z\frac{\partial^2 w}{\partial x \partial y}$$

podendo assim ser escrito na forma:

$$\left\{ \begin{array}{c} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{array} \right\} = -z \cdot \left[\begin{array}{c} \dfrac{\partial^2}{\partial x^2} \\ \dfrac{\partial^2}{\partial y^2} \\ 2\dfrac{\partial^2}{\partial x \partial y} \end{array} \right] \cdot \left\{ \begin{array}{c} w \end{array} \right\}$$

sendo a matriz $[L]$ dada por:

$$[L] \quad = \quad -z \cdot \left[\begin{array}{c} \dfrac{\partial^2}{\partial x^2} \\ \dfrac{\partial^2}{\partial y^2} \\ 2\dfrac{\partial^2}{\partial x \partial y} \end{array} \right] \tag{78}$$

Tendo isso em vista pode-se substituir a Equação 75 na Equação 77, e obter-se:

$$\{\epsilon\} = [L][\phi]\{d\} \tag{79}$$

Multiplicando a Equação 78 pela matriz de funções de forma, tem-se:

$$[B] = [L][\phi] \tag{80}$$

Assim pode-se relacionar o vetor de deformações expresso na Equação 19, com o vetor de deslocamentos nodais:

$$\{\epsilon\} = [B]\{d\} \tag{81}$$

De maneira similar, pode-se representar a expressão na forma transposta:

$$\{\epsilon^*\}^T = \{d^*\}^T[B]^T \tag{82}$$

Substituindo a Equação 81 na Equação 16, obtém-se:

$$\{\sigma\} \quad = \quad [E][B]\{d\} \tag{83}$$

Substituindo as Equações 76, 82 e 83 na Equação 74, tem-se:

$$\int_V \{d^*\}^T[B]^T[E][B]\{d\}\,dV = \{d^*\}^T\{F\} + \int_A \{d^*\}^T[\phi]^T p\,dA \tag{84}$$

Como os deslocamentos nodais, tanto reais quanto virtuais, são constantes e não nulos, os mesmos podem ser retirados das integrais, se tornando:

$$\{d^*\}^T\left(\int_V [B]^T[E][B]\,dV\right)\{d\} = \{d^*\}^T\{F\} + \{d^*\}^T\int_A [\phi]^T p\,dA \tag{85}$$

Colocando o termo $\{d^*\}^T$ em evidência do lado direito da Equação 85 tem-se:

$$\{d^*\}^T\left(\int_V [B]^T[E][B]\,dV\right)\{d\} = \{d^*\}^T\left(\{F\} + \int_A [\phi]^T p\,dA\right) \tag{86}$$

Simplificando a equação resta:

$$\left(\int_V [B]^T[E][B]\,dV\right)\{d\} = \{F\} + \int_A [\phi]^T p\,dA \tag{87}$$

Na forma abreviada a equação fica:

$$[k]\{d\} = \{f\} \tag{88}$$

que é a equação fundamental do MEF, onde:

$$[k] = \int_V [B]^T[E][B]dV \tag{89}$$

é chamada de matriz de rigidez, sendo específica para cada elemento, e:

$$\{f\} = \{F\} + \int_A [\phi]^T p\,dA \tag{90}$$

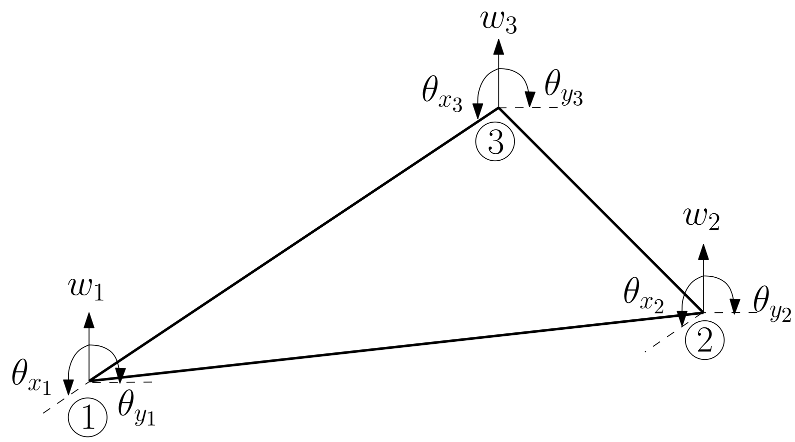
é o vetor de forças nodais equivalentes para cada elemento.

2.3.2 Elemento finito triangular

No método dos elementos finitos podem ser usados diferentes tipos de elementos, os quais são escolhidos de acordo com a geometria do problema que se queira resolver. Para problemas planos, destacam-se o elemento triangular e o elemento retangular, tendo o triangular a particularidade de contornar mais precisamente superfícies circulares ou curvas.

Para o presente trabalho, foi escolhido o elemento triangular de 3 nós (Figura 7), o qual possui três graus de liberdade em cada ponto nodal, sendo uma translação e duas rotações. A translação é denotada por w, enquanto que a rotação em torno de x é expressa como θ_x e a rotação em torno de y como θ_y.

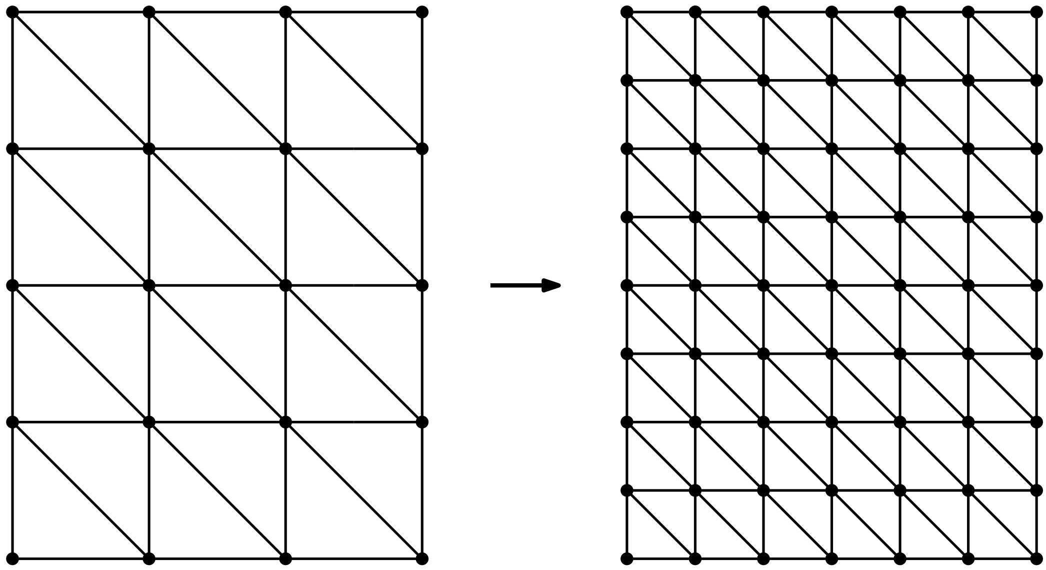
Figura 7 – Elemento triangular e seus graus de liberdade



Fonte: Adaptado de Waidemam (2004, p. 75)

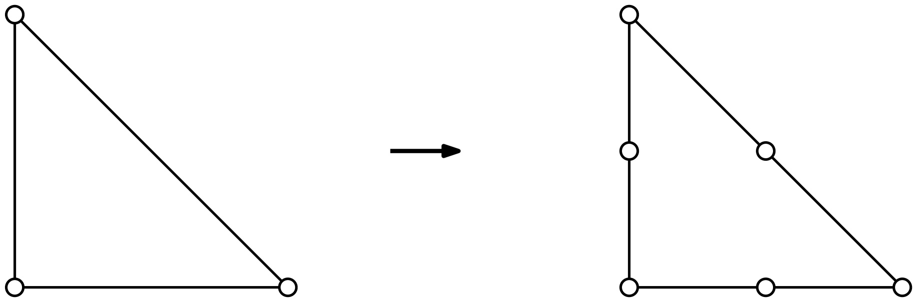
Existem fundamentalmente dois modos de se aumentar a precisão do método. A primeira maneira diz respeito a usar mais elementos de menores dimensões cada, mantendo inalterado o número de nós no elemento (Figura 8). Esta técnica é conhecida como refinamento de malha. Já a outra consiste em usar mais nós em um mesmo elemento (Figura 9), o que aumentaria o número de termos no polinômio empregado nas funções de forma. Neste trabalho, deve-se empregar a técnica do refinamento de malha para se obter maior precisão nos resultados desejados.

Figura 8 – Refinamento de malha



Fonte: Adaptado de Leal (2015, p. 49)

Figura 9 – Elemento com maior número de nós



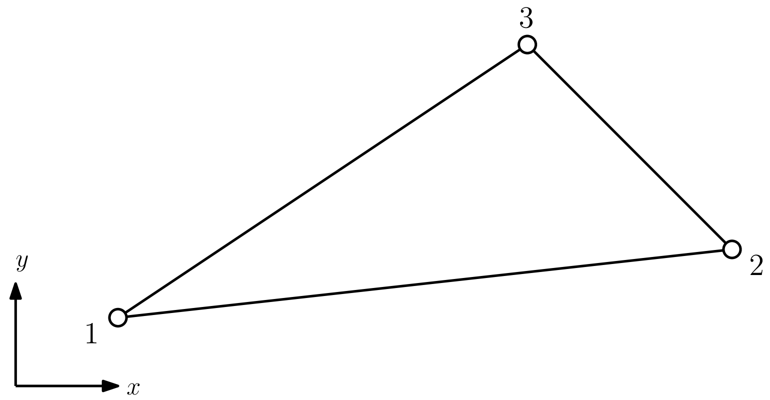
Fonte: Adaptado de Leal (2015, p. 49)

Vale salientar no entanto, que o método possui limitações, sendo que a medida que aumenta-se o número de elementos o custo computacional envolvido também aumenta, tornando o processo muito demorado ou até mesmo inviável, já que exige um maior poder de processamento por parte da máquina.

2.3.2.1 Relação geral entre coordenadas

Considere-se o elemento triangular genérico presente na Figura 10.

Figura 10 – Elemento triangular genérico



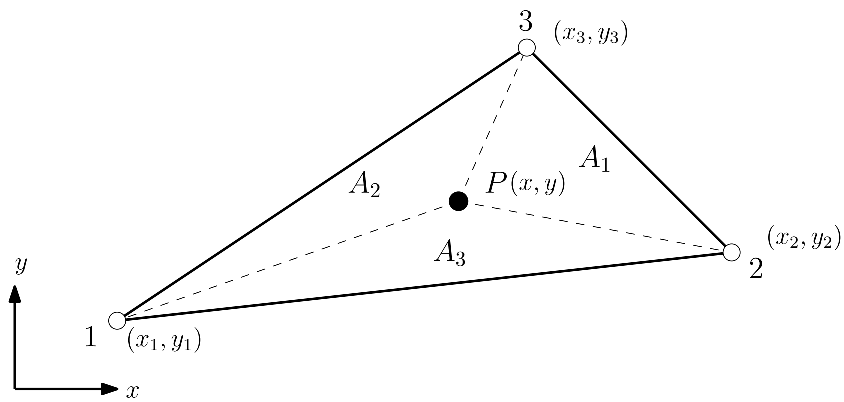
Fonte: Adaptado de Zienkiewickz e Taylor (2000, p. 180)

O elemento em questão possui três nós, numerados de maneira que se tenha as coordenadas (x, y) equivalentes à cada um deles. Com base nisso, sabe-se que por meio da geometria analítica é possível calcular a área de um triângulo resolvendo o determinante expresso na Equação 91.

$$2A_{\Delta} = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{91}$$

A Figura 11 ilustra um elemento triangular no qual admite-se internamente um ponto genérico P com coordenadas (x, y).

Figura 11 – Ponto P admitido em uma coordenada (x, y) do elemento triangular



Fonte: Adaptado de Zienkiewickz e Taylor (2000, p. 180)

Pode-se obter um novo conjunto de coordenadas homogêneas dividindo-se a área de cada triângulo formado pelas retas que ligam o ponto P a cada uma das extremidades pela área do triângulo total. Tomando como exemplo a área A_1 tem-se:

$$\xi_1 = \frac{A_1}{A_\Delta} \tag{92}$$

Desse modo:

$$2\,A_1 = \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{93}$$

Calculando o determinante, tem-se:

$$A_1 = \frac{x_2y_3 - x_3y_2 + (y_2 - y_3)x + (x_3 - x_2)y}{2} \tag{94}$$

Fazendo:

$$\begin{aligned} a_1 &= x_2y_3 - x_3y_2 \\ b_1 &= y_2 - y_3 \\ c_1 &= x_3 - x_2 \end{aligned}$$

A Equação 94 fica:

$$A_1 = \frac{a_1 + b_1x + c_1y}{2} \tag{95}$$

Substituindo a Equação 95 na Equação 92, obtém-se,

$$\xi_1 = \frac{1}{2A_\Delta}(a_1 + b_1x + c_1y) \tag{96}$$

onde:

$$2A_\Delta = x_2y_3 + x_1y_2 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3 \tag{97}$$

O mesmo pode ser feito com as áreas A_2 e A_3, para obter-se as coordenadas homogêneas ξ_2 e ξ_3. Ao realizar-se o cálculo dos determinantes correspondentes às outras áreas, percebe-se que os índices variam de maneira cíclica. Para tal, as equações podem ser representadas de maneira genérica:

$$\begin{aligned} a_i &= x_j y_k - x_k y_j \\ b_i &= y_j - y_k \\ c_i &= x_k - x_j \end{aligned} \tag{98}$$

com a variação dos índices podendo ser representada por:

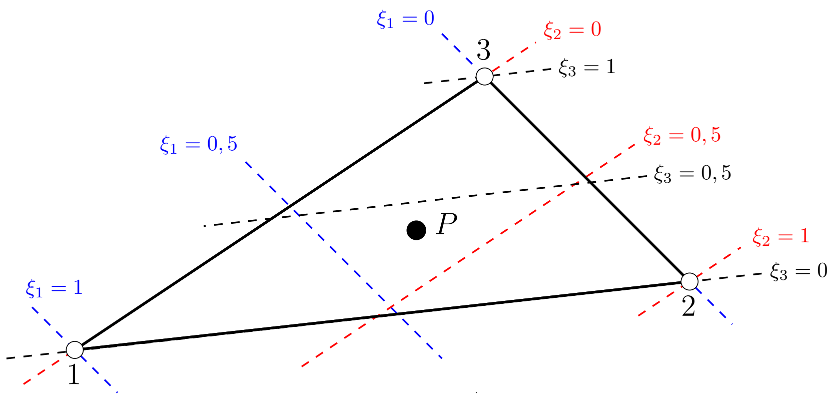
$$\begin{cases} P/i = 1 \rightarrow j = 2, k = 3 \\ P/i = 2 \rightarrow j = 3, k = 1 \\ P/i = 3 \rightarrow j = 1, k = 2 \end{cases}$$

Desse modo, obtém-se a seguinte relação entre coordenadas:

$$\xi_i = \frac{1}{2A_\Delta}(a_i + b_i x + c_i y) \tag{99}$$

A Figura 12 ilustra os valores de cada coordenada homogênea ao longo das faces e nos vértices do elemento triangular.

Figura 12 – Valores de cada coordenada ao longo do elemento triangular



Fonte: Adaptado de Zienkiewickz e Taylor (2000, p. 180)

2.3.2.2 Funções de forma

Venâncio Filho (1975) apud Waidemam (2004), estabelece que o campo de deslocamentos para o elemento triangular utilizado é caracterizado pelo seguinte polinômio em termos das coordenadas (x, y):

$$w = \left[\begin{array}{cccccccccc} 1 & x & y & x^2 & y^2 & xy & x^3 & x^2y & xy^2 & y^3 \end{array}\right] \cdot \{\alpha\} \tag{100}$$

sendo que $\{\alpha\}$ é um vetor com dez constantes.

Como o elemento utilizado tem apenas nove graus de liberdade é necessário que o polinômio tenha o mesmo número de termos, desse modo um termo entre x^2y e xy^2 devia ser eliminado.

No entanto, a retirada de um termo acarretaria na perda de simetria do polinômio, problema contornado combinando os termos acima descritos resultando em $(x^2y + xy^2)$. Porém, ao combinar tais termos, quando os eixos x e y coincidissem com os lados do elemento, comprometeria o desenvolvimento do mesmo, o que não ocorre quando se usa coordenadas homogêneas.

Sendo assim, Venâncio Filho (1975) apud Waidemam (2004) apresenta o seguinte polinômio para representar o campo de deslocamentos em termos de coordenadas homogêneas (ξ_1, ξ_2, ξ_3):

$$w = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 & \xi_1\xi_2 & \xi_2\xi_3 & \xi_2\xi_1 & (\xi_1\xi_2^2 - \xi_1^2\xi_2) & (\xi_2\xi_3^2 - \xi_2^2\xi_3) & (\xi_3\xi_1^2 - \xi_3^2\xi_1) \end{bmatrix} \cdot \{\alpha\} \quad (101)$$

onde $\{\alpha\}$ é um vetor composto por nove constantes.

A Equação 101 é um polinômio de 3º grau incompleto, no qual é ausente o termo $\xi_1\xi_2\xi_3$, sendo o mesmo responsável por deslocamentos nulos nos lados do elemento, e dessa forma, associado ao deslocamento de um nó interno.

O campo de deslocamentos (Equação 101) pode também ser escrito na forma:

$$\begin{aligned} w(\xi_1, \xi_2, \xi_3) = \alpha_0\xi_1 + \alpha_1\xi_2 + \alpha_2\xi_3 + \alpha_3\xi_1\xi_2 + \alpha_4\xi_2\xi_3 + \alpha_5\xi_2\xi_1 + \\ + \alpha_6(\xi_1\xi_2^2 - \xi_1^2\xi_2) + \alpha_7(\xi_2\xi_3^2 - \xi_2^2\xi_3) + \alpha_8(\xi_3\xi_1^2 - \xi_3^2\xi_1) \end{aligned} \quad (102)$$

Ou ainda, matricialmente:

$$w = [\psi] \cdot \{\alpha\} \quad (103)$$

onde:

$$[\psi] = \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 & \xi_1\xi_2 & \xi_2\xi_3 & \xi_2\xi_1 & (\xi_1\xi_2^2 - \xi_1^2\xi_2) & (\xi_2\xi_3^2 - \xi_2^2\xi_3) & (\xi_3\xi_1^2 - \xi_3^2\xi_1) \end{bmatrix} \quad (104)$$

e:

$$\{\alpha\}^T = \{\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6 \quad \alpha_7 \quad \alpha_8\} \quad (105)$$

Além do polinômio de deslocamentos, precisa-se também dos polinômios referentes aos giros nas direções x e y. Desse modo, o giro em torno de x pode ser escrito como:

$$\theta_x(\xi_1,\xi_2,\xi_3) = \frac{\partial w(\xi_1,\xi_2,\xi_3)}{\partial y} = \frac{\partial w}{\partial \xi_1}\cdot\frac{\partial \xi_1}{\partial y} + \frac{\partial w}{\partial \xi_2}\cdot\frac{\partial \xi_2}{\partial y} + \frac{\partial w}{\partial \xi_3}\cdot\frac{\partial \xi_3}{\partial y} \tag{106}$$

Realizando-se as derivadas percebe-se que o polinômio fica:

$$\begin{aligned}\theta_x(\xi_1,\xi_2,\xi_3) = \frac{1}{2A}\{&c_1\alpha_0 + c_2\alpha_1 + c_3\alpha_2 + (\xi_2 c_1 + \xi_1 c_2)\alpha_3 + (\xi_3 c_2 + \xi_2 c_3)\alpha_4 + \\ &+ (\xi_3 c_1 + \xi_1 c_3)\alpha_5 + \left[(\xi_2^2 - 2\xi_1\xi_2)c_1 + (2\xi_1\xi_2 - \xi_1^2)c_2\right]\alpha_6 + \\ &+ \left[(\xi_3^2 - 2\xi_2\xi_3)c_2 + (2\xi_2\xi_3 - \xi_2^2)c_3\right]\alpha_7 + \\ &+ \left[(2\xi_3\xi_1 - \xi_3^2)c_1 + (\xi_1^2 - 2\xi_3\xi_1)c_3\right]\alpha_8\}\end{aligned} \tag{107}$$

Analogamente tem-se:

$$\begin{aligned}\theta_y(\xi_1,\xi_2,\xi_3) = \frac{1}{2A}\{&b_1\alpha_0 + b_2\alpha_1 + b_3\alpha_2 + (\xi_2 b_1 + \xi_1 b_2)\alpha_3 + (\xi_3 b_2 + \xi_2 b_3)\alpha_4 + \\ &+ (\xi_3 b_1 + \xi_1 b_3)\alpha_5 + \left[(\xi_2^2 - 2\xi_1\xi_2)b_1 + (2\xi_1\xi_2 - \xi_1^2)b_2\right]\alpha_6 + \\ &+ \left[(\xi_3^2 - 2\xi_2\xi_3)b_2 + (2\xi_2\xi_3 - \xi_2^2)b_3\right]\alpha_7 + \\ &+ \left[(2\xi_3\xi_1 - \xi_3^2)b_1 + (\xi_1^2 - 2\xi_3\xi_1)b_3\right]\alpha_8\}\end{aligned} \tag{108}$$

De maneira que se pode reuni-los na forma matricial:

$$\begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \begin{bmatrix} \varphi \end{bmatrix} \cdot \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{Bmatrix} \tag{109}$$

onde $[\varphi]$ é uma matriz de três linhas e nove colunas com os coeficientes que multiplicam α presentes nas Equações 102, 107 e 108.

Ainda pode-se escrever a Equação 109 na forma simplificada:

$$\{\delta\} = [\varphi] \cdot \{\alpha\}$$

Ao substituir-se as coordenadas nodais na Equação 109 tem-se:

$\Rightarrow$ Para o nó 1 onde $\xi_1 = 1$ e $\xi_2 = \xi_3 = 0$:

$$[\varphi_1] = \begin{bmatrix} 2A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_2 & 0 & c_3 & -c_2 & 0 & c_3 \\ b_1 & b_2 & b_3 & b_2 & 0 & b_3 & -b_2 & 0 & b_3 \end{bmatrix}$$

$\Rightarrow$ Para o nó 2 onde $\xi_2 = 1$ e $\xi_1 = \xi_3 = 0$:

$$[\varphi_2] = \begin{bmatrix} 0 & 2A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_1 & c_3 & 0 & c_1 & -c_3 & 0 \\ b_1 & b_2 & b_3 & b_1 & b_3 & 0 & b_1 & -b_3 & 0 \end{bmatrix}$$

$\Rightarrow$ Para o nó 3 onde $\xi_3 = 1$ e $\xi_1 = \xi_2 = 0$:

$$[\varphi_3] = \begin{bmatrix} 0 & 0 & 2A & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & 0 & c_2 & c_1 & 0 & c_2 & -c_1 \\ b_1 & b_2 & b_3 & 0 & b_2 & b_1 & 0 & b_2 & -b_1 \end{bmatrix}$$

Combinando as matrizes referentes aos três pontos nodais, pode-se escrever:

$$\begin{Bmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} 2A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_2 & 0 & c_3 & -c_2 & 0 & c_3 \\ b_1 & b_2 & b_3 & b_2 & 0 & b_3 & -b_2 & 0 & b_3 \\ 0 & 2A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_1 & c_3 & 0 & c_1 & -c_3 & 0 \\ b_1 & b_2 & b_3 & b_1 & b_3 & 0 & b_1 & -b_3 & 0 \\ 0 & 0 & 2A & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & 0 & c_2 & c_1 & 0 & c_2 & -c_1 \\ b_1 & b_2 & b_3 & 0 & b_2 & b_1 & 0 & b_2 & -b_1 \end{bmatrix} \cdot \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{Bmatrix}$$

Que pode ser expresso como:

$$\{d\} = [A] \cdot \{\alpha\}$$

Isolando-se $\{\alpha\}$:

$$\{\alpha\} = [A]^{-1} \cdot \{d\} \tag{110}$$

no qual:

$$[A]^{-1} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & -b_3 & c_3 & 0 & b_3 & -c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -b_1 & c_1 & 0 & b_1 & -c_1 \\ 0 & b_2 & -c_2 & 0 & 0 & 0 & 0 & -b_2 & c_2 \\ -2 & b_3 & -c_3 & 2 & b_3 & 0-c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & b_1 & -c_1 & 2 & b_1 & -c_1 \\ 2 & b_2 & -c_2 & 0 & 0 & 0 & -2 & b_2 & -c_2 \end{bmatrix}$$

Substituindo-se a Equação 110 na Equação 103:

$$w = [\psi] \cdot [A]^{-1} \cdot \{d\} \tag{111}$$

obtém-se:

$$w = [\phi] \cdot \{d\} \tag{112}$$

onde em $[\phi]$ estão contidas as nove funções de ξ_1, ξ_2 e ξ_3 descritas abaixo:

$$[\phi] = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 & \phi_9 \end{bmatrix} \tag{113}$$

no qual:

$$
\begin{aligned}
\phi_1 &= \xi_1 - (\xi_1\xi_2^2 - \xi_1^2\xi_2) + (\xi_3\xi_1^2 - \xi_3^2\xi_1)\\
\phi_2 &= \frac{1}{2}\left[-b_3\xi_1\xi_2 + b_2\xi_3\xi_1 + b_3(\xi_1\xi_2^2 - \xi_1^2\xi_2) + b_2(\xi_3\xi_1^2 - \xi_3^2\xi_1)\right]\\
\phi_3 &= \frac{1}{2}\left[c_3\xi_1\xi_2 - c_2\xi_3\xi_1 - c_3(\xi_1\xi_2^2 - \xi_1^2\xi_2) - c_2(\xi_3\xi_1^2 - \xi_3^2\xi_1)\right]\\
\phi_4 &= \xi_2 + (\xi_1\xi_2^2 - \xi_1^2\xi_2) - (\xi_2\xi_3^2 - \xi_2^2\xi_3)\\
\phi_5 &= \frac{1}{2}\left[b_3\xi_1\xi_2 - b_1\xi_2\xi_3 + b_3(\xi_1\xi_2^2 - \xi_1^2\xi_2) + b_1(\xi_2\xi_3^2 - \xi_2^2\xi_3)\right]\\
\phi_6 &= \frac{1}{2}\left[-c_3\xi_1\xi_2 + c_1\xi_2\xi_3 - c_3(\xi_1\xi_2^2 - \xi_1^2\xi_2) - c_1(\xi_2\xi_3^2 - \xi_2^2\xi_3)\right]
\end{aligned}
$$

$$
\begin{aligned}
\phi_7 &= \xi_3 + (\xi_2\xi_3^2 - \xi_2^2\xi_3) - (\xi_3\xi_1^2 - \xi_3^2\xi_1)\\
\phi_8 &= \frac{1}{2}\left[b_1\xi_2\xi_3 - b_2\xi_3\xi_1 + b_1(\xi_2\xi_3^2 - \xi_2^2\xi_3) + b_2(\xi_3\xi_1^2 - \xi_3^2\xi_1)\right]\\
\phi_9 &= \frac{1}{2}\left[-c_1\xi_2\xi_3 + c_2\xi_3\xi_1 - c_1(\xi_2\xi_3^2 - \xi_2^2\xi_3) - c_2(\xi_3\xi_1^2 - \xi_3^2\xi_1)\right]
\end{aligned}
$$

sendo todos funções de ξ_1, ξ_2 e ξ_3.

2.3.2.3 Matriz de rigidez

Sabe-se que a matriz de rigidez de cada elemento é dada pela Equação 89:

$$[k] = \int_V [B]^T[D][B]dV$$

e que tem-se a relação expressa pela Equação 77:

$$\{\epsilon\} = [L]\{w\}$$

que pode ser escrita em termos das coordenadas homogêneas, onde L se torna:

$$[L] = \frac{-z}{4A^2} \cdot \begin{bmatrix} b_1^2 \frac{\partial^2}{\partial \xi_1^2} + b_2^2 \frac{\partial^2}{\partial \xi_2^2} + b_3^2 \frac{\partial^2}{\partial \xi_3^2} + 2b_1 b_2 \frac{\partial^2}{\partial \xi_1 \partial \xi_2} + 2b_1 b_3 \frac{\partial^2}{\partial \xi_1 \partial \xi_3} + 2b_2 b_3 \frac{\partial^2}{\partial \xi_2 \partial \xi_3} \\ c_1^2 \frac{\partial^2}{\partial \xi_1^2} + c_2^2 \frac{\partial^2}{\partial \xi_2^2} + c_3^2 \frac{\partial^2}{\partial \xi_3^2} + 2c_1 c_2 \frac{\partial^2}{\partial \xi_1 \partial \xi_2} + 2c_1 c_3 \frac{\partial^2}{\partial \xi_1 \partial \xi_3} + 2c_2 c_3 \frac{\partial^2}{\partial \xi_2 \partial \xi_3} \\ 2\left(b_1 c_1 \frac{\partial^2}{\partial \xi_1^2} + b_2 c_2 \frac{\partial^2}{\partial \xi_2^2} + b_3 c_3 \frac{\partial^2}{\partial \xi_3^2} + (b_1 c_2 + b_2 c_1) \frac{\partial^2}{\partial \xi_1 \partial \xi_2} + (b_1 c_3 + b_3 c_1) \frac{\partial^2}{\partial \xi_1 \partial \xi_3} + (b_2 c_3 + b_3 c_2) \frac{\partial^2}{\partial \xi_2 \partial \xi_3} \right) \end{bmatrix}$$

ou:

$$[L] = \frac{-z}{4A^2} \cdot [l]$$

Conforme estabelecido pela Equação 80:

$$[B] = [L][\phi]$$

Assim, pode-se substituí-la na Equação 89:

$$[k] = \int_V ([L][\phi])^T [E] ([L][\phi]) \, dV$$

$$\begin{aligned} [k] &= \int_V \frac{-z}{4A^2} ([l][\phi])^T [E] \left(\frac{-z}{4A^2} \right) ([l][\phi]) \, dV \\ [k] &= \frac{1}{16A^4} \int_A \int_{-\frac{t}{2}}^{\frac{t}{2}} z^2 ([l][\phi])^T [E] ([l][\phi]) \, dz dA \\ [k] &= \left(\frac{1}{16A^4} \right) \left(\frac{t^3}{12} \right) \int_A ([l][\phi])^T [E] ([l][\phi]) \, dA \end{aligned}$$

podendo-se generalizar como:

$$[k] = \left(\frac{1}{16A^4} \right) \left(\frac{t^3}{12} \right) \int_A f(\xi_1, \xi_2, \xi_3) \, dA \tag{114}$$

Segundo Brebbia e Connor (1973) apud Waidemam (2004), a integral da

Equação 114 pode ser resolvida fazendo as seguintes substituições:

$$\xi_3 = 1 - \xi_1 - \xi_2$$
$$dA = 2A \; d\xi_1 \; d\xi_2$$

Assim, a integral se torna:

$$[k] = \left(\frac{1}{16A^4}\right)\left(\frac{t^3}{12}\right) 2A \int_0^1 \int_0^{(1-\xi_2)} f(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 \tag{115}$$

No presente trabalho, a integral presente na Equação 115 será resolvida utilizando-se dos recursos do MATLAB®, dispensando, assim, a apresentação da matriz de rigidez de forma explícita.

2.3.2.4 Vetor de cargas nodais equivalentes

O método dos elementos finitos, como se sabe, pressupõe o domínio do problema discretizado em uma malha composta por nós e elementos. Assim, no caso de uma carga distribuída é preciso calcular a carga equivalente em cada nó, ou seja, cada nó recebe uma parcela dessa carga, o que é realizado por meio do vetor de cargas nodais equivalentes:

$$[f_d] = \int_A [\phi]^T \, p \, dA \tag{116}$$

que pode ser resolvida utilizando-se da mesma substituição de variáveis empregada na solução da integral da equação 114. Nota-se que como p é uma carga constante, a mesma pode ser retirada da integral da equação 116.

$$[f_d] = 2A \cdot p \int_0^1 \int_0^{(1-\xi_2)} f(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 \tag{117}$$

Como $[\phi]$ é um vetor que contém nove elementos, o resultado da integral tbm é um vetor com nove elementos, sendo contido nesse vetor valores de força aplicada na direção z, momento fletor em torno de x e momento fletor em torno de y para cada um dos três nós do elemento triangular.

3 ASPECTOS COMPUTACIONAIS

Este trabalho se constituiu da elaboração de um código computacional implementado no *software* MATLAB®, capaz de calcular as deflexões e os esforços internos atuantes nas placas.

A entrada de dados se dá no próprio software, onde são fornecidos os valores das propriedades geométricas da placa e das propriedades mecânicas do material que a compõe, bem como a intensidade e localização dos carregamentos a serem aplicados na mesma.

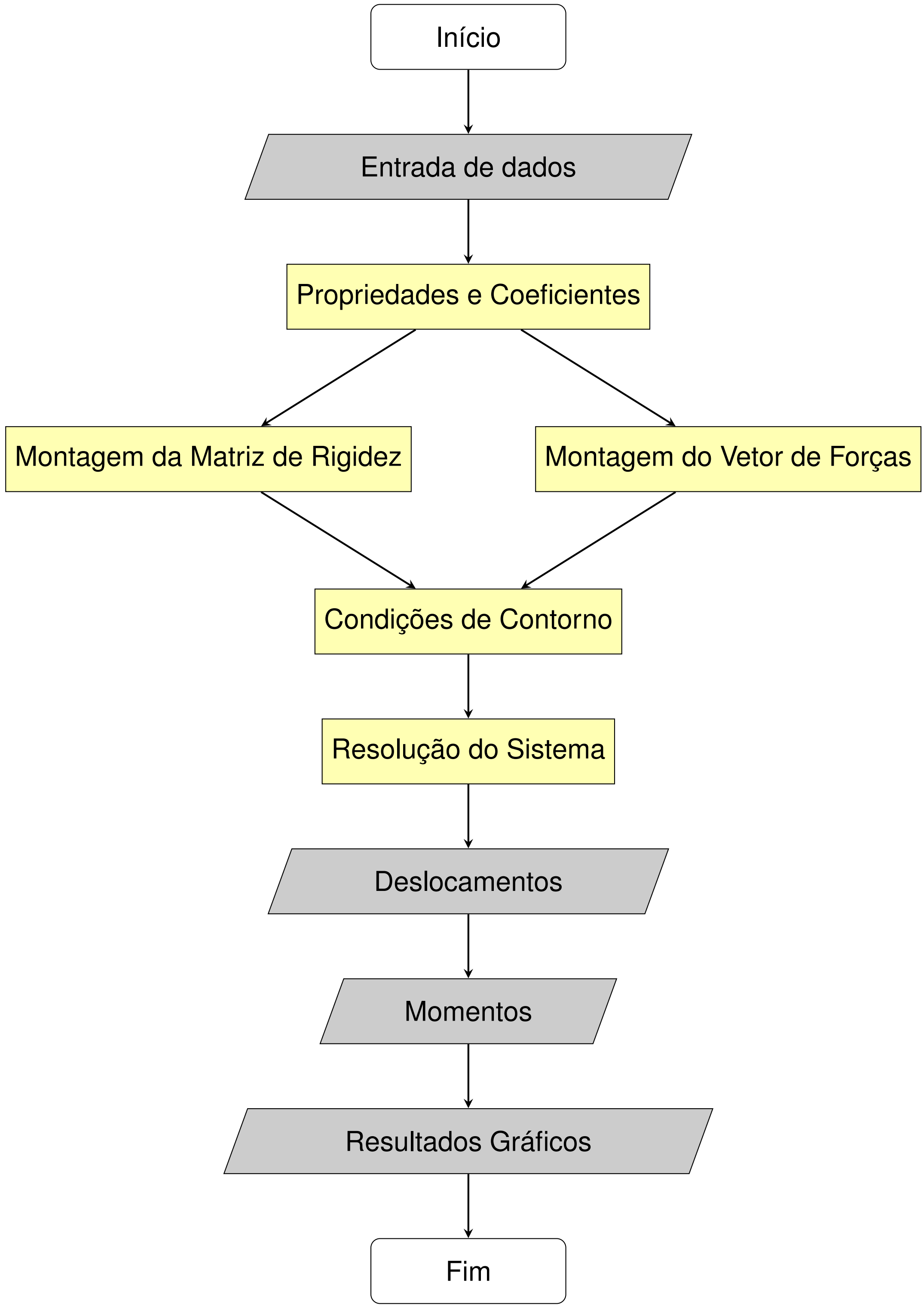
Nas próximas etapas, o algoritmo calcula e monta as matrizes de rigidez locais e global da estrutura, o vetor de forças global, contendo as forças distribuídas e as nodais equivalentes provenientes do carregamento distribuído, sendo por fim aplicadas as condições de contorno tanto na matriz de rigidez, quanto no vetor de forças globais.

Por último, o código resolve o sistema linear de equações para obter os deslocamentos e giros nodais e calcula os esforços internos de momentos fletores e torçores atuantes na placa, apresentando os gráficos contendo esses valores para cada nó da estrutura.

3.1 Esquema geral de cálculo

Para que se tenha um melhor entendimento do processo realizado no código computacional foi elaborado o fluxograma da Figura 14, onde, de forma simplificada, cada processo se refere à uma etapa do cálculo realizada no código.

Figura 13 – Fluxograma referente ao processo de cálculo



Fonte: Autoria própria (2022)

3.2 Etapas

3.2.1 Entrada de dados

No algoritmo criado, quando executado, existem campos de entrada, na linha de comando do MATLAB®, onde podem ser fornecidos valores de carga distribuída, módulo de Young, coeficiente de Poisson, espessura da placa, dimensões da placa nas direções x e y, e a quantidade de nós a ser dividida cada uma dessas dimensões, sendo assim montada uma malha regular por meio do próprio algoritmo.

A carga concentrada precisa ser posicionada no coordenada correta da placa e isso nem sempre condiz com coordenadas de nós, sendo assim foi elaborado uma condicional que compara a coordenada da carga com a coordenada de todos os nós da estrutura, de modo a posicioná-la corretamente, sendo que quando a coordenada não coincide, o código informa o erro ao usuário, solicita a sua correção de modo que o código seja executado novamente. O processo pode ser repetido para quantas cargas concentradas o usuário queira inserir na estrutura, bastando fornecer o número de cargas.

Após esse passo foram também criados campos de entrada para que se pudesse inserir as condições de contorno presentes na placa.

3.2.2 Propriedades e coeficientes

A partir dos dados armazenados são calculadas a área para cada elemento e a matriz constitutiva do material, bem como os coeficientes utilizados na mudança de coordenadas, nas funções de forma e em suas derivadas (Equações 98).

3.2.3 Montagem da matriz de rigidez

Nessa parte do algoritmo são montadas a matriz de rigidez local utilizando-se da equação 115, com a qual é realizada uma integração em termos das coordenadas homogêneas, e a matriz de rigidez global, que é basicamente uma sobreposição das matrizes de rigidez local de cada elemento, formando uma única matriz para a estrutura inteira.

Cabe salientar, no entanto, que tanto a matriz de rigidez local quanto a matriz de rigidez global são simétricas e que, além disso, não admitem inversa, ou seja, são singulares, o que é alterado somente ao se aplicar as condições de contorno na matriz de rigidez global, tornando-a inversível.

3.2.4 Montagem do vetor de forças

Como já mencionado anteriormente, para a montagem do vetor de forças distribuídas é utilizada a equação 117, no entanto tem-se também o vetor de cargas concentradas e, como após a integração das cargas distribuídas tem-se somente cargas concentradas nos nós, os mesmos podem ser somados, ficando:

$$\{f\} = \{f_d\} + \{f_c\} \tag{118}$$

onde,

$\{f_d\}$: Vetor de forças distribuídas;

$\{f_c\}$: Vetor de forças concentradas;

3.2.5 Condições de contorno

Conforme dito na seção 3.2.3 deste trabalho, a matriz de rigidez pode ser invertida somente se forem aplicadas à ela as condições de contorno do problema, que são restrições impostas em termos de giro e deslocamento em pontos da placa.

Para isso, foi criada uma rotina que tem como função definir as condições de contorno nas bordas da placa, na qual o código pede ao usuário se a placa está engastada ou apoiada, no qual por meio de comandos de zero e um o usuário diz se não ou se sim, respectivamente.

3.2.6 Resolução do sistema

Para obter-se o vetor de deslocamentos, é necessário que seja resolvido o sistema linear (Equação 88). Para resolvê-lo, basta inverter a matriz de rigidez e multiplicá-la pelo vetor de forças.

3.2.7 Deslocamentos

Como declarado na seção anterior, a resolução do sistema fornece em um primeiro momento o vetor de deslocamentos em todos os pontos da placa analisada.

3.2.8 Momentos

Para obter-se os esforços internos de momento, é necessário que se tenha o vetor de deslocamentos, pois de acordo com as equações 57, 58, 60 e 113, w precisa ser derivado duas vezes em relação às duas variáveis e em relação à cada uma delas de maneira cruzada, e nota-se que w depende de d, que é o vetor de deslocamentos. Assim, é possível obter os momentos para cada um dos três nós de cada elemento e, como alguns nós são comuns a dois ou mais elementos, deve-se fazer uma média dos valores encontrados para os referidos nós.

3.2.9 Resultados gráficos

Com os valores de deslocamentos e momentos obtidos anteriormente, é possível que sejam traçados gráficos bidimensionais de momentos fletores e torçores, bem como uma superfície que representa a placa em seu estado deformado.

O código fonte implementado está disponível no Apêndice A do trabalho.

4 ANÁLISE NUMÉRICA

4.1 Generalidades

Esta seção tem como objetivo explicitar os resultados obtidos por meio de simulações numéricas realizadas no código implementado. A validação dos resultados foi realizada mediante outras formas de cálculo existentes, como as tabelas de Bares (CARVALHO; FIGUEIREDO FILHO, 2014) e Czérny (ZENZEN, 2012) e usando o software SCIA Engineer, e comparando o resultado com outros autores.

Para melhor entendimento os resultados são apresentados de maneira elucidativa na forma de tabelas e gráficos.

4.1.1 Uso de tabelas

As tabelas de lajes são os métodos que se tinha para calcular esforços internos e deflexões nas mesmas antes do desenvolvimento de métodos como o MEF. A tabela de Bares, por exemplo, é advinda de uma expansão em séries de Fourier, já a tabela de Czérny foi realizada a partir da teoria da elasticidade (ZENZEN, 2012).

Originalmente as tabelas foram construídas com valores do coeficiente de Poisson distintos dos usados na ABNT NBR 6118:2014. Assim, para que pudessem ser utilizadas em estruturas de concreto armado, precisaram ser adaptadas para um ν de 0,2.

Carvalho e Figueiredo Filho (2014) adaptaram as tabelas de Bares para a norma vigente. Já as tabelas de Czérny foram adaptadas no livro Beton-Kalender de 1976 conforme consta em Zenzen (2012). As tabelas em questão estão apresentadas na forma de anexo.

4.1.2 Erro relativo

Uma maneira de determinar a magnitude dos erros é utilizando-se do erro relativo, dado pela equação:

$$e(\%) = \frac{V_o - V_c}{V_c} \cdot 100 \tag{119}$$

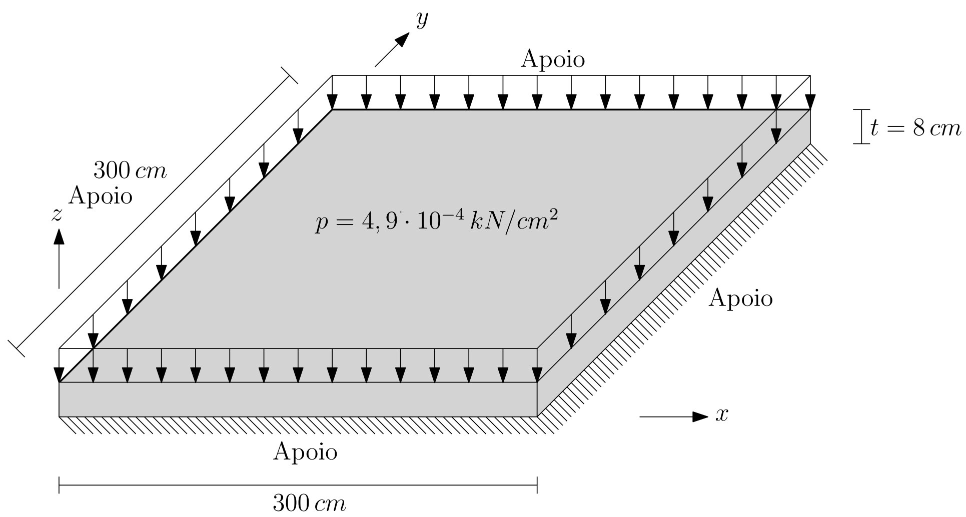
onde V_o se refere ao valor obtido pelo código implementado e V_c aos valores de comparação obtidos por meio das tabelas, outros *softwares* ou outros trabalhos.

4.2 Exemplo 1 - Placa quadrada submetida a carregamento distribuído uniforme

O primeiro exemplo se resume em uma placa quadrada com uma carga distribuída uniforme aplicada sobre a mesma, onde, no caso, foi simulada uma carga comum à edificações residenciais, juntamente com o peso próprio de uma laje maciça. O módulo de Young utilizado foi de 2400 kN/cm^2 e o coeficiente de Poisson de 0,2.

A placa (Figura 14) foi calculada por meio do algoritmo implementado com 8, 72, 200, 800, 1800, e 3200 elementos, e também por meio da tabela de Bares e de Czérny.

Figura 14 – Placa quadrada com todos os bordos simplesmente apoiados submetida a carregamento uniformemente distribuído



Fonte: Autoria própria (2022)

A Tabela 1 mostra a comparação entre os resultados obtidos através do código implementado, utilizando uma malha composta por 3200 elementos, com os fornecidos pelas tabelas de Bares e de Czérny. Nota-se que os valores de M_x e M_y são iguais no ponto central da placa, isso se deve ao fato da placa ser simétrica, tanto em termos de dimensões quanto em termos de vinculações e ainda, ao carregamento ser uniforme.

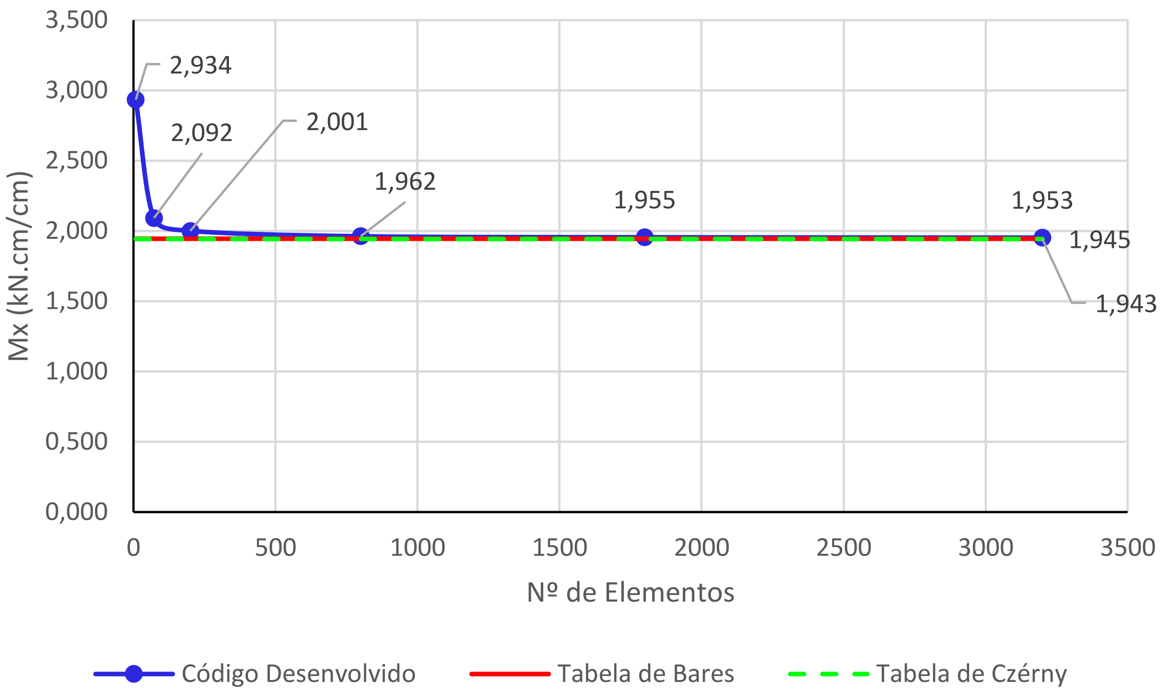
Tabela 1 – Comparação entre resultados de momentos e deslocamentos obtidos para o ponto central da placa, através do código implementado e das tabelas de Bares e Czérny

	$M_x(\frac{kN.cm}{cm})$	$M_y(\frac{kN.cm}{cm})$	$w(cm)$
Código (MEF)	1,952559	1,952559	-0,151275
Tabela de Bares	1,944810	1,944810	-0,150840
Erro relativo $(\%)$	0,398428	0,398428	0,288274
Tabela de Czérny	1,942731	1,942731	-0,150934
Erro relativo $(\%)$	0,505855	0,505855	0,226095

Fonte: Autoria própria (2022)

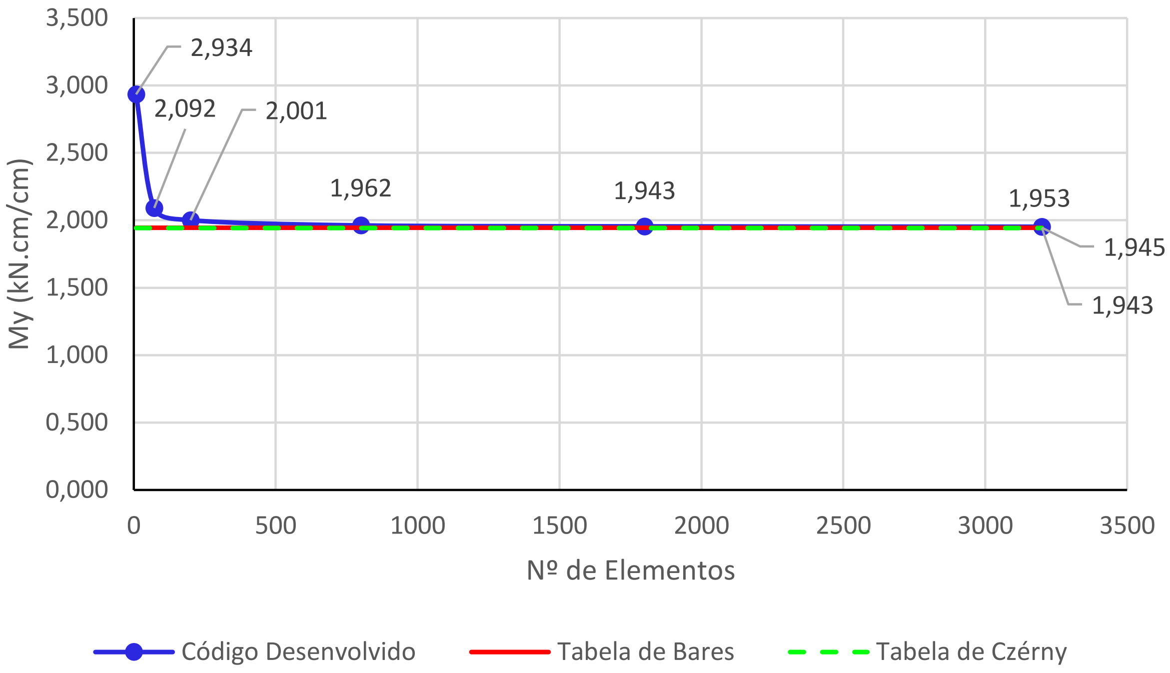
Os gráficos das Figuras 15, 16 e 17 mostram a convergência dos resultados às soluções por meio das tabelas de Bares e Czérny ao passo que se aumenta o número de elementos.

Figura 15 – Convergência do momento M_x



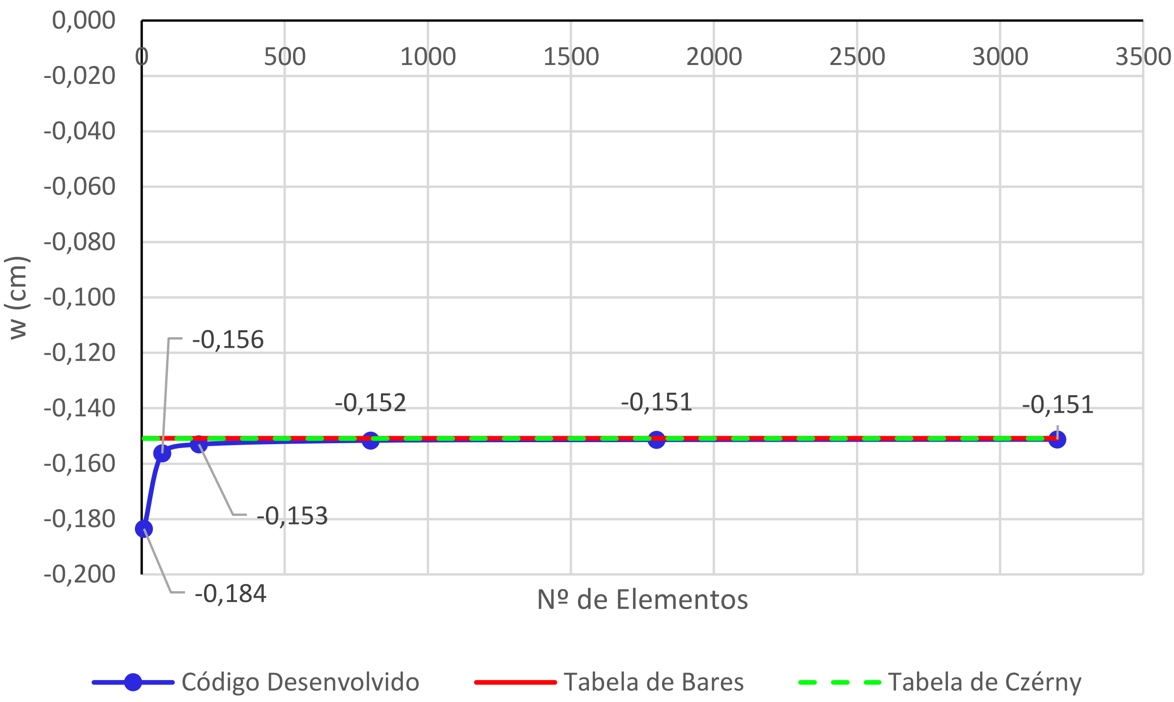
Fonte: Autoria própria (2022)

Figura 16 – Convergência do momento M_y



Fonte: Autoria própria (2022)

Figura 17 – Convergência dos deslocamentos

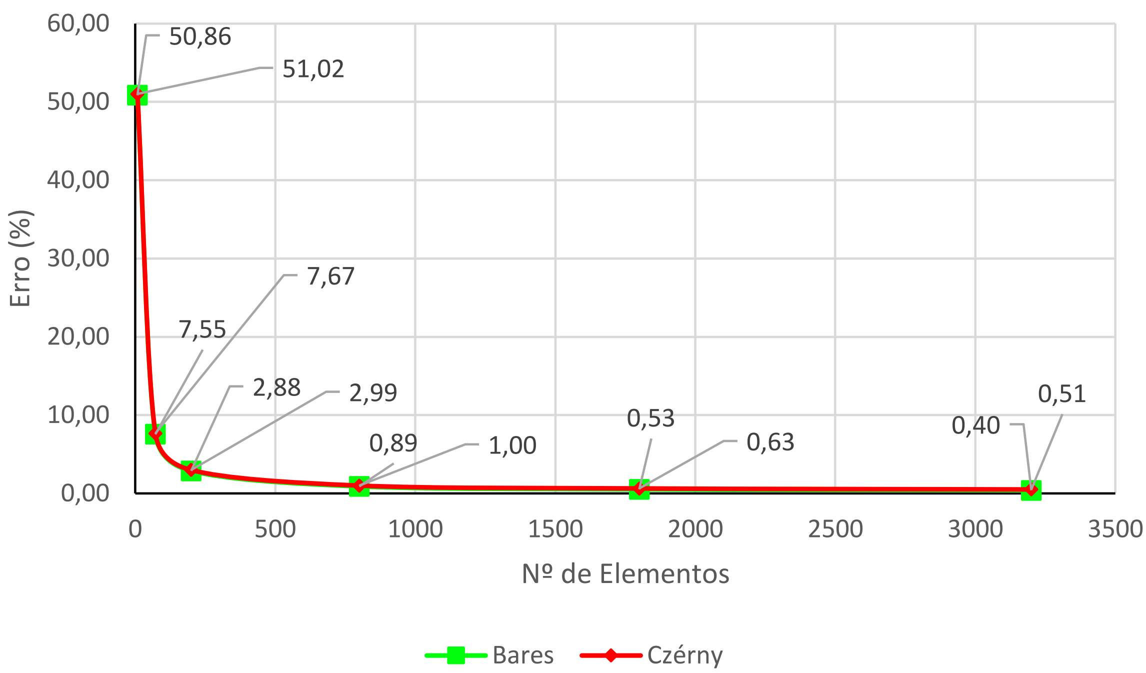


Fonte: Autoria própria (2022)

Percebe-se que a medida que aumenta-se o número de elementos o erro é reduzido, chegando o resultado a valores extremamente próximos aos das tabelas. O erro se dá pelo fato dos cálculos serem realizados por meios aproximados e diferentes entre si.

No gráfico da figura 18 pode-se observar a ordem de grandeza da redução dos erros para os momentos fletores conforme se aumenta o número de elementos na análise. Nesse gráfico, os valores apresentados à esquerda referem-se, sempre, aos valores obtidos por meio da comparação com a tabela de Bares, e os apresentados à direita, com os obtidos pela comparação com a tabela de Czérny, respectivamente.

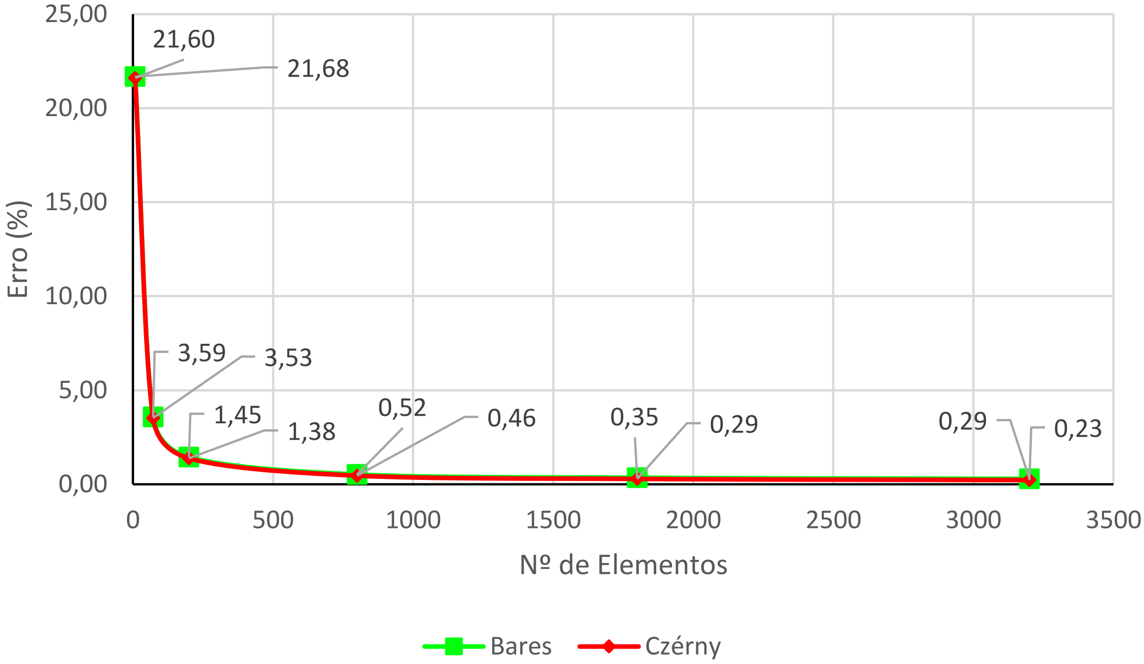
Figura 18 – Convergência do erro percentual relativo para os momentos fletores



Fonte: Autoria própria (2022)

A Figura 19, por sua vez, traz um gráfico similar, porém nesse pode-se constatar a convergência dos erros para os deslocamentos.

Figura 19 – Convergência do erro percentual relativo para os deslocamentos

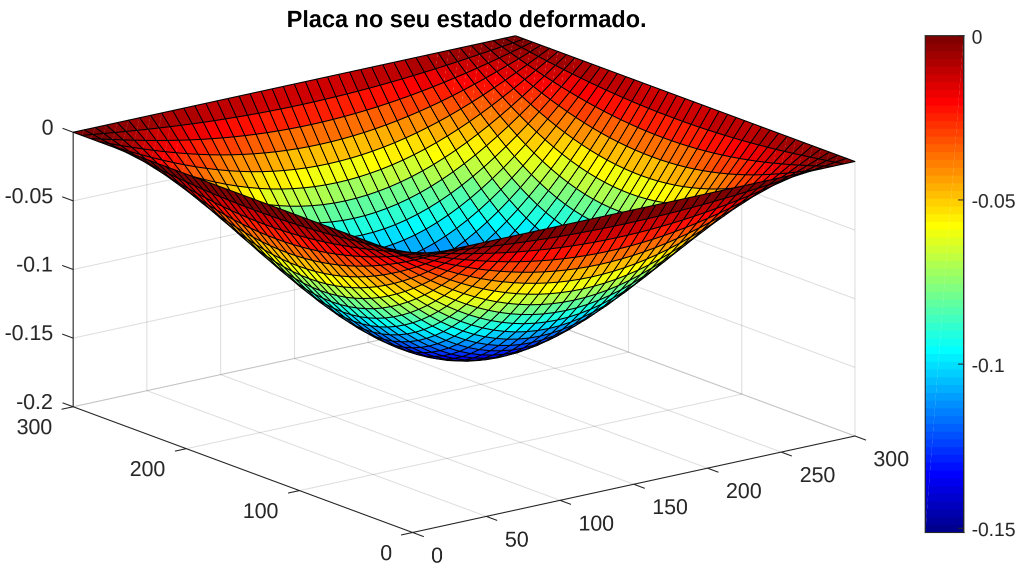


Fonte: Autoria própria (2022)

Nota-se que os resultados de deslocamento tendem a convergir mais rapidamente que resultados de momento fletor. Isso se dá por conta da redução de grau do polinômio interpolador, pois, quando deriva-se o mesmo para obter-se os momentos fletores, o polinômio originalmente de terceiro grau passa a ser linear, o que reduz a precisão envolvida.

Por fim, na Figura 20, tem-se a configuração deformada da placa analisada.

Figura 20 – Configuração deformada da placa (valores em cm)

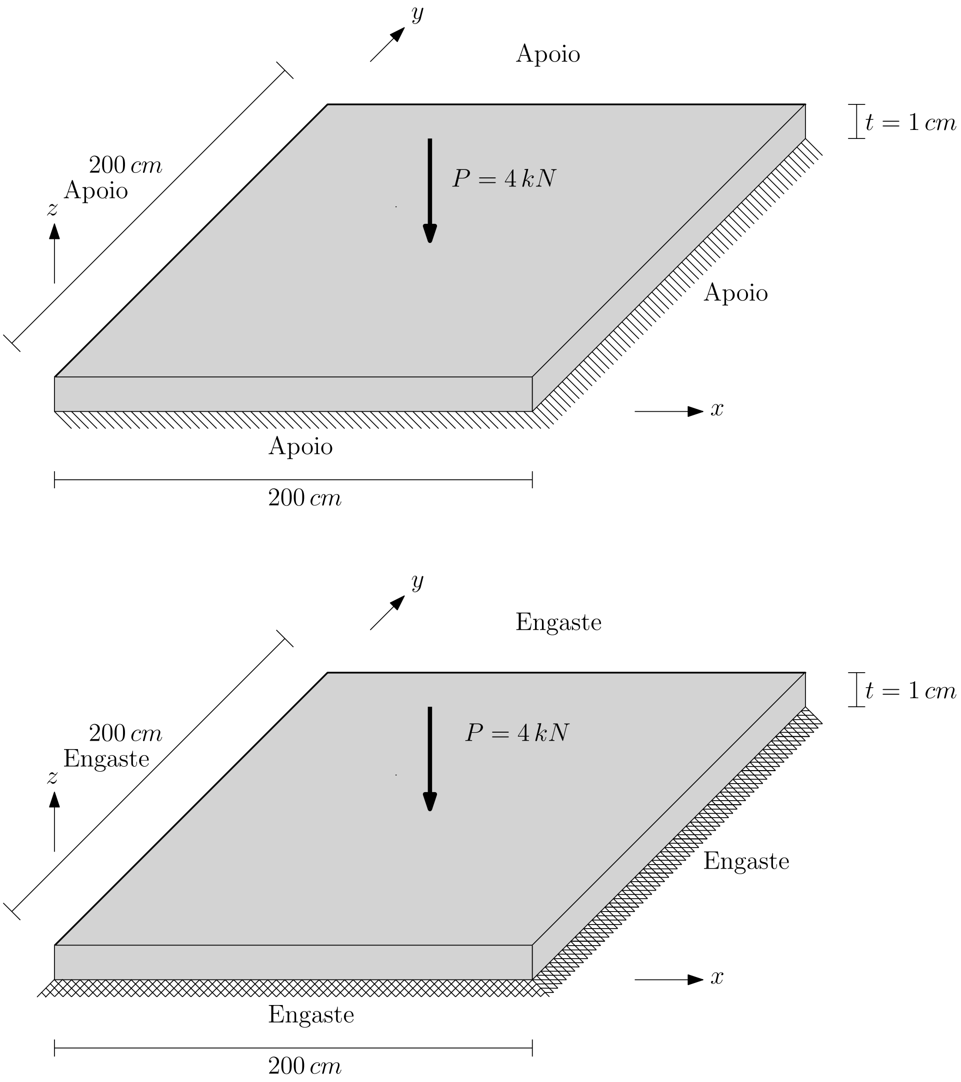


Fonte: Autoria própria (2022)

4.3 Exemplo 2 - Placa quadrada submetida a carregamento aplicado no centro

O segundo exemplo se trata de uma placa quadrada com uma força aplicada em seu centro, o módulo de Young nesse caso é de 10000 kN/cm^2 e o coeficiente de Poisson de 0,3. A placa em questão foi calculada com duas diferentes vinculações, a primeira com todas as bordas apoiadas e a segunda com todas as bordas engastadas. Mais informações podem ser encontradas na Figura 21.

Figura 21 – Placa quadrada com carga pontual aplicada no centro



Fonte: Autoria própria (2022)

A Tabela 2 mostra os deslocamentos do ponto central da placa para ambas as situações de vinculações, para diferentes quantidades de elementos, e também traz os valores fornecidos por Leal (2015), que utilizou de um código em elementos finitos usando uma malha de elementos retangulares, compostos por quatro nós, contendo também três graus de liberdade em cada nó. Além disso, pode-se verificar o erro relativo obtido com cada valor de flecha calculado quando comparados com a solução do autor mencionado.

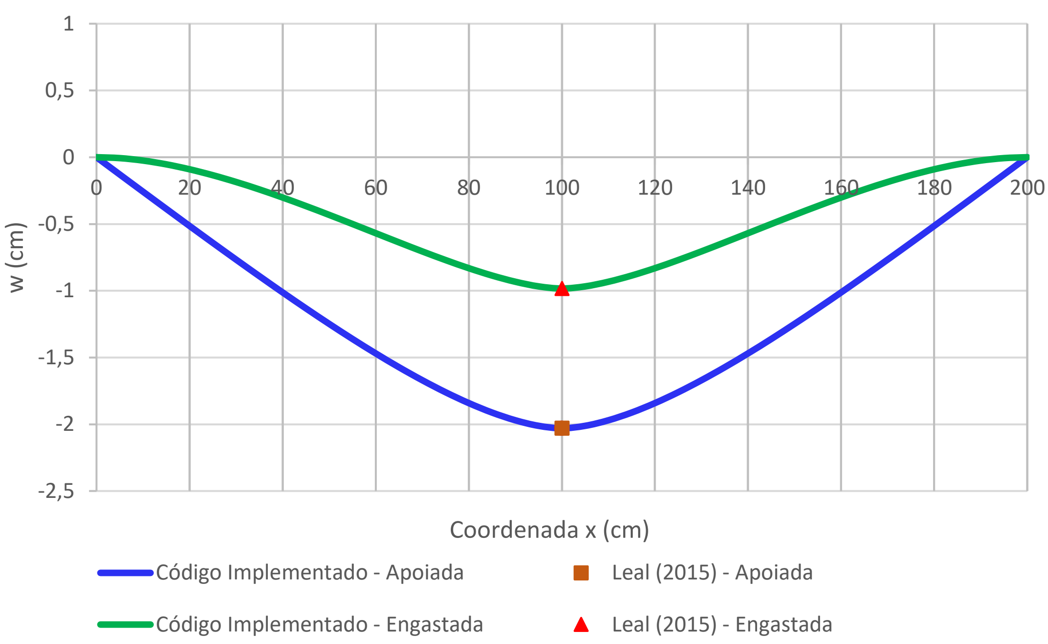
Tabela 2 – Comparação dos resultados obtidos através do código implementado com os obtidos por Leal (2015)

	Nº de elementos	$w(cm)$ Apoiada	Erro relativo $(\%)$	$w(cm)$ Engastada	Erro relativo $(\%)$
Código (MEF)	32	-2,177980	7,329	-1,070273	8,917
	200	-2,060237	1,527	-1,009137	2,695
	800	-2,037060	0,385	-0,989699	0,717
	1800	-2,031890	0,130	-0,985111	0,250
	3200	-2,029896	0,032	-0,983306	0,067
Leal (2015)	1600	-2,029246	-	-0,982651	-

Fonte: Autoria própria (2022)

Pode-se comparar os resultados também de maneira gráfica, por meio da Figura 22, que compara os valores obtidos por meio do código desenvolvido com os valores encontrados em Leal (2015) para as duas condições de vinculação estabelecidas. Por meio do gráfico, também é possível observar a configuração deformada da seção crítica da placa (coordenada y = 100 cm).

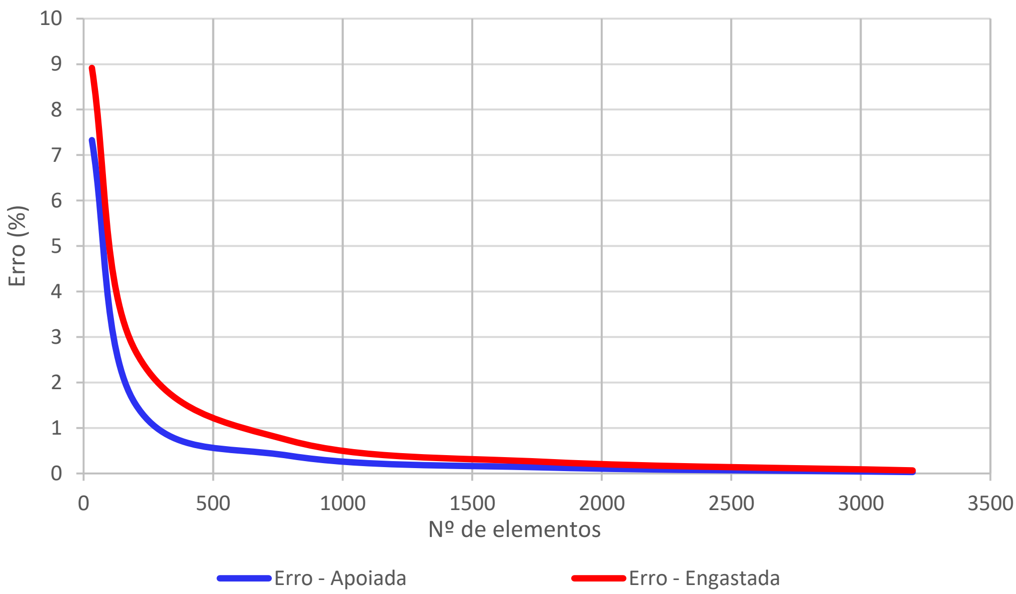
Figura 22 – Comparação da deflexão das placas nas configurações apoiada e engastada com os valores obtidos de Leal (2015)



Fonte: Autoria própria (2022)

Os erros envolvidos no cálculo com os diferentes números de elementos fazendo uso das duas condições de contorno distintas também foram representados de maneira gráfica (Figura 23) para que se possa observar as diferenças entre os mesmos. Pode-se constatar que os erros com a placa engastada foram sempre maiores, entretanto as diferenças não foram significativas.

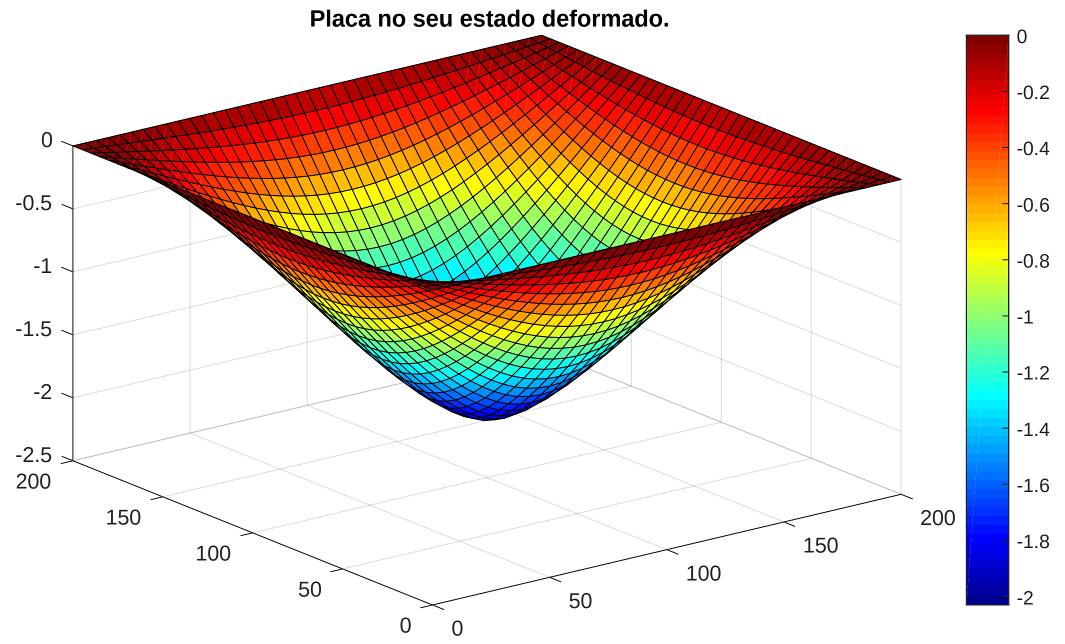
Figura 23 – Convergência do erro percentual para a deflexão da placa



Fonte: Autoria própria (2022)

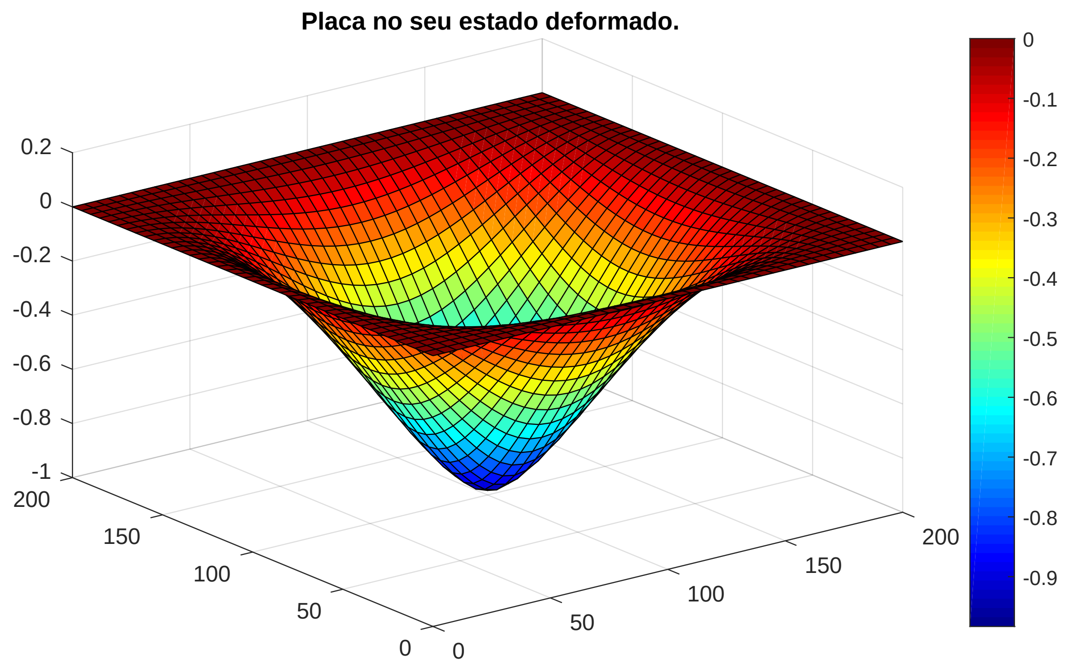
As Figuras 24 e 25 trazem as configurações deformadas para a placa do exemplo em questão para as condições de vinculação apoiada e engastada, respectivamente. Percebe-se que na placa engastada os giros são nulos nas extremidades.

Figura 24 – Configuração deformada da placa apoiada em todas as bordas (valores em cm)



Fonte: Autoria própria (2022)

Figura 25 – Configuração deformada da placa engastada em todas as bordas (valores em cm)



Fonte: Autoria própria (2022)

A partir da análise realizada percebe-se que ao se alterar as condições de vinculação com a mesma carga pontual aplicada o resultado de deflexão diverge em cerca de 50 $\%$.

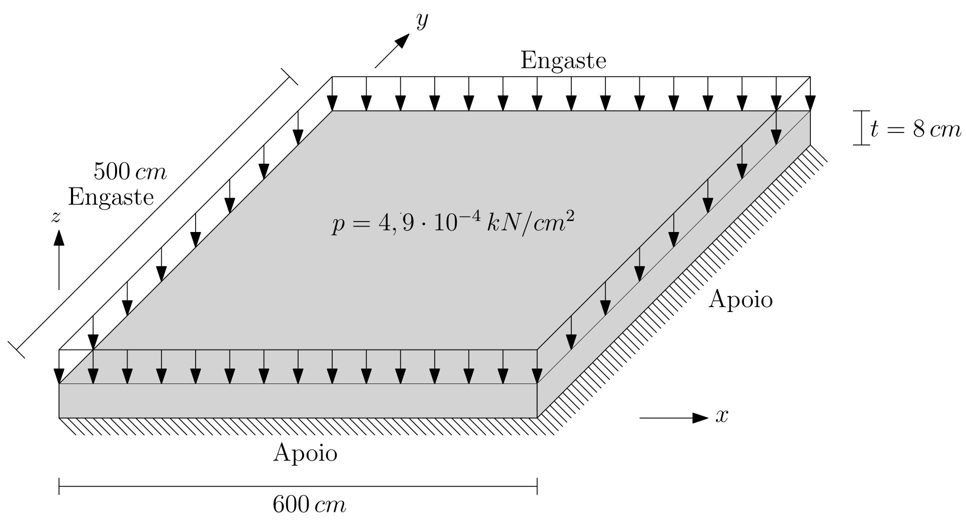
Em relação à comparação com os resultados fornecidos por Leal (2015), pode-se notar que o código implementado com elementos triangulares demora mais à convergir, tendo em vista que o resultado fornecido pelo referido autor alcança a devida precisão fazendo o uso de 1600 elementos apenas, enquanto que o resultado obtido pelo algoritmo implementado somente alcança a mesma precisão com 3200 elementos.

Entretanto, ambos os tipos de elementos, retangulares ou triangulares alcançam resultados similares, com erros percentuais tendendo à zero, o que reforça a correta implementação do MEF com o uso de elementos triangulares neste trabalho.

4.4 Exemplo 3 - Placa retangular submetida a carregamento distribuído uniforme

O presente exemplo consta de uma placa retangular (Figura 26) submetida ao mesmo carregamento do Exemplo 1, porém com dois de seus bordos engastados e dois apoiados, de modo a simular a continuidade da laje sobre vigas.

Figura 26 – Placa retangular com dois bordos simplesmente apoiados e dois bordos engastados



Fonte: Autoria própria (2022)

Em uma primeira tentativa, a placa em questão foi calculada fazendo-se o uso de 3200 elementos, utilizando-se a mesma quantidade de elementos em cada uma das duas dimensões, e os resultados comparados com um modelo implementado no SCIA Engineer utilizando 3000 elementos. A Tabela 3 expressa os resultados obtidos.

Tabela 3 – Comparação entre resultados obtidos através do código implementado e do software SCIA Engineer para a mesma quantidade de elementos em ambos os lados

	Código (MEF)	**SCIA**	**Erro relativo** $(\%)$
$M_x(kN.cm/cm)$	3,696002	3,700	-0,108
$M'_x(kN.cm/cm)$	-8,984439	-9,330	-3,704
$M_y(kN.cm/cm)$	5,030336	5,040	-0,192
$M'_y(kN.cm/cm)$	-10,612839	-10,600	0,121
$w(cm)$	-0,871191	-0,875	-0,435

Fonte: Autoria própria (2022)

Tendo em vista a ordem de grandeza maior obtida para o erro referente à um dos momentos negativos, o exemplo foi recalculado por meio do código implementado com 6000 elementos, porém com um número maior de elementos em sua face maior, e também por meio do software SCIA engineer com 3000 elementos.

A Tabela 4 compara os valores obtidos através do código implementado e do SCIA Engineer. Os valores de M_x e M_y se referem aos momentos que tracionam as fibras inferiores da placa na seção mais solicitada, neste caso não sendo em seu centro devido às condições de vinculação da mesma.Enquanto que os valores de M'_x e M'_y se referem aos momentos que tracionam as fibras superiores da mesma, atuando nas extremidades dos bordos engastados. Os gráficos das Figuras 30, 32, 34 e 35 mostram com maior clareza as coordenadas onde se situam esses esforços.

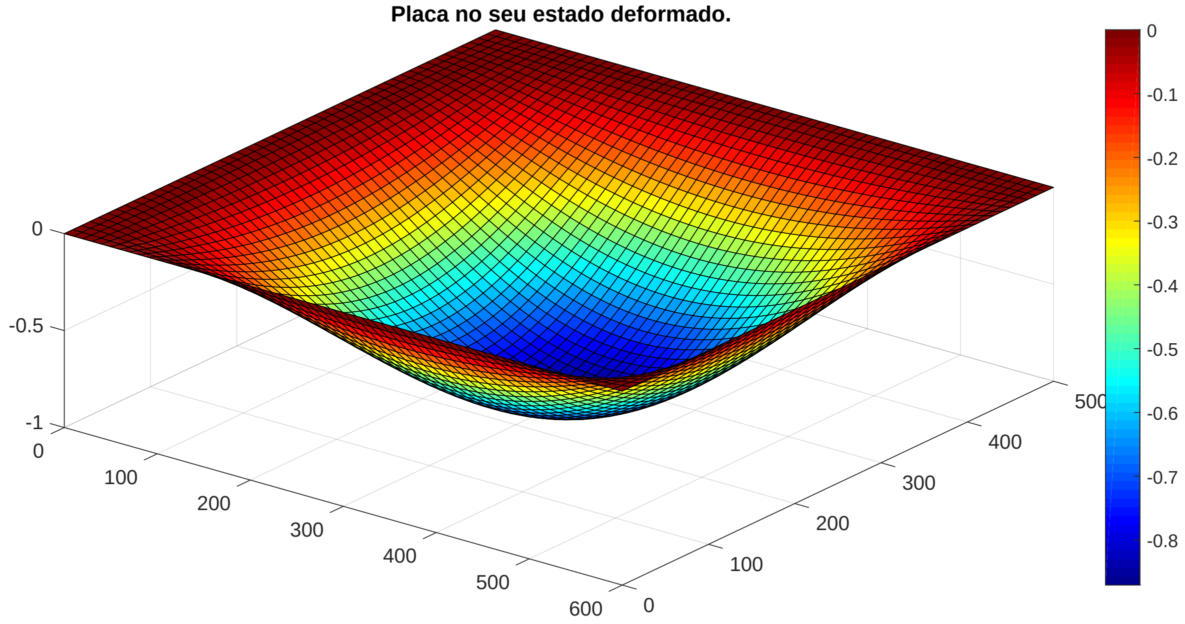
Tabela 4 – Comparação entre resultados obtidos através do código implementado e do software SCIA Engineer para uma maior quantidade de elementos na maior dimensão da placa

	Código (MEF)	**SCIA**	**Erro relativo** $(\%)$
$M_x(kN.cm/cm)$	3,691333	3,700	-0,234
$M'_x(kN.cm/cm)$	-9,226976	-9,330	-1,104
$M_y(kN.cm/cm)$	5,029922	5,040	-0,200
$M'_y(kN.cm/cm)$	-10,481505	-10,600	-1,118
$w(cm)$	-0,870702	-0,875	-0,491

Fonte: Autoria própria (2022)

Na Figura 27 pode-se observar a configuração deformada da placa em questão obtida com o código implementado.

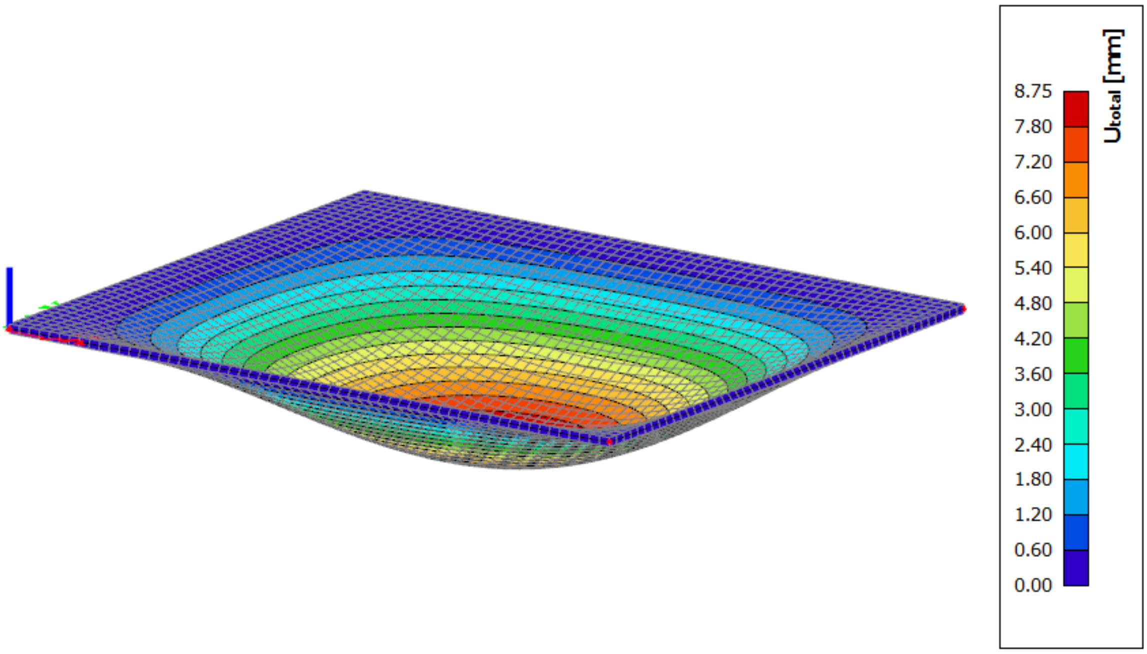
Figura 27 – Configuração deformada da placa obtida por meio do código implementado (valores em cm)



Fonte: Autoria própria (2022)

A Figura 28, por sua vez, mostra a configuração deformada obtida no *software* SCIA Engineer.

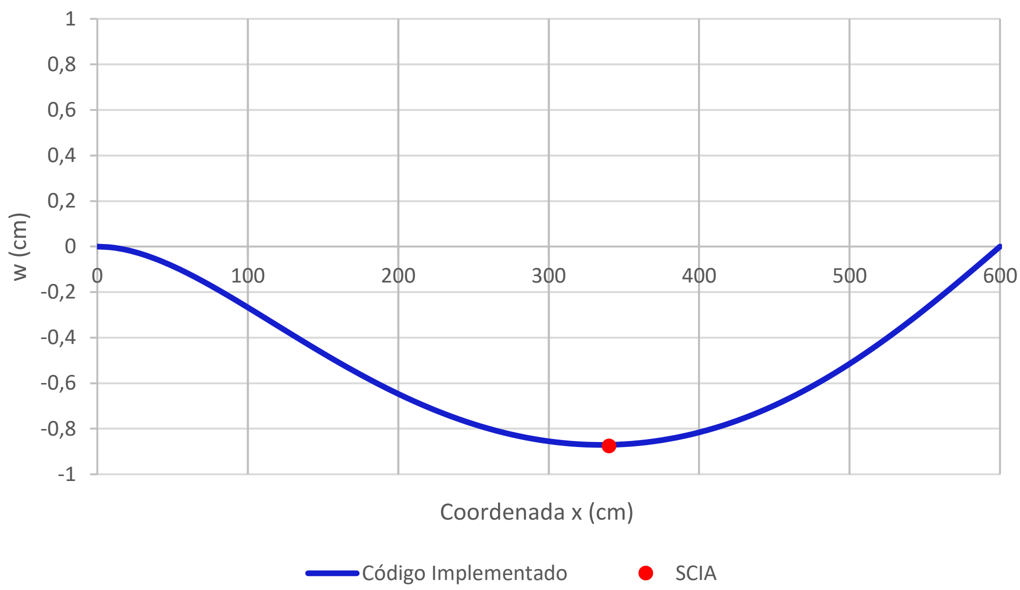
Figura 28 – Configuração deformada da placa obtida por meio do SCIA Engineer



Fonte: SCIA Engineer (2022)

O gráfico da Figura 29 apresenta uma comparação entre os deslocamentos obtidos por meio do algoritmo implementado e o deslocamento máximo obtido no SCIA, na seção crítica da placa.

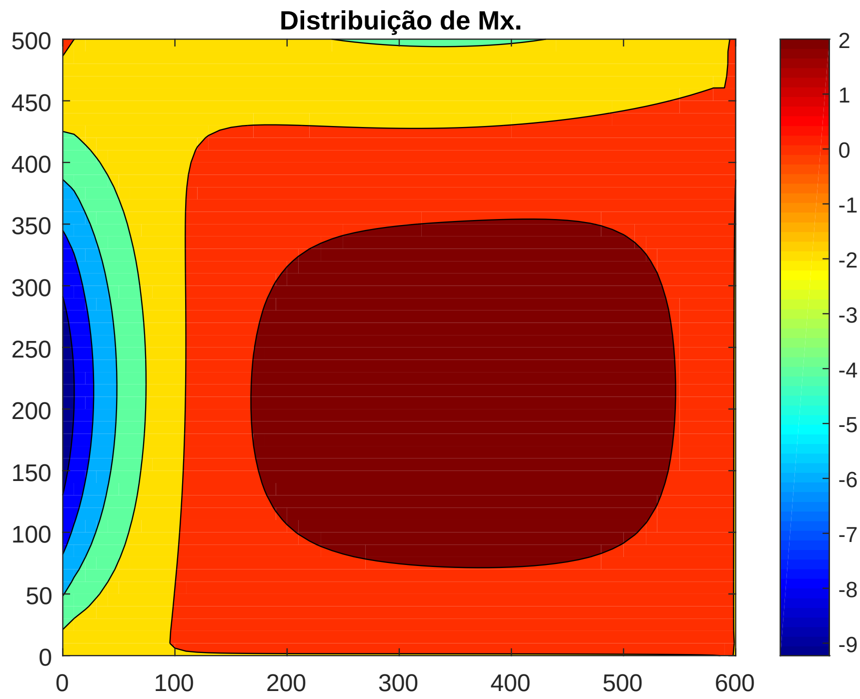
Figura 29 – Comparação entre os deslocamentos obtidos via código implementado e via SCIA Engineer, na seção y = 220 cm



Fonte: Autoria própria (2022)

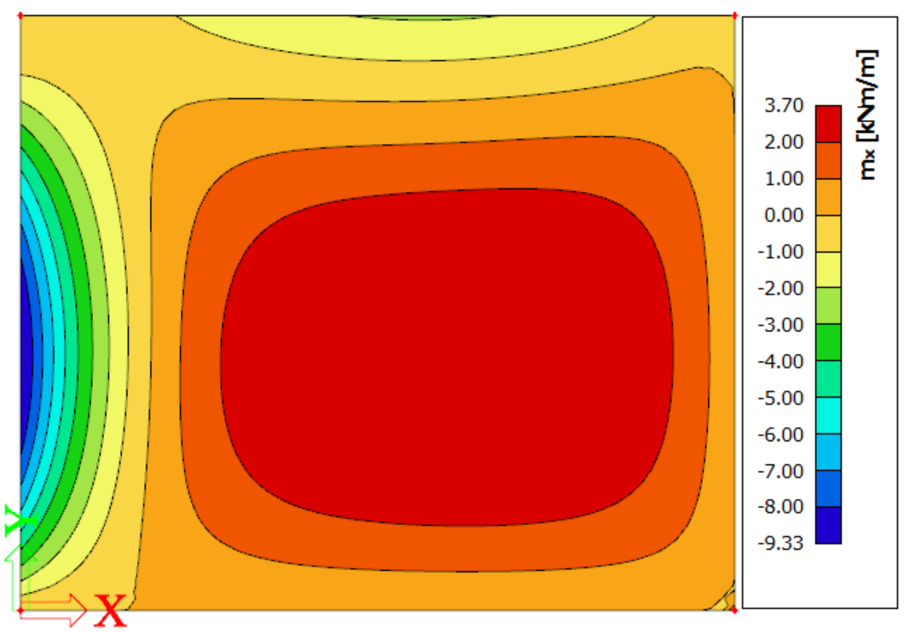
Também foram obtidos os gráficos das Figuras 30 e 31, onde pode-se observar a distribuição de momentos M_x na placa analisada. Os gráficos foram obtidos por meio do código implementado e do SCIA Engineer, respectivamente. As unidades dos momentos dos gráficos gerados pelo algoritmo implementado são $kN.cm/cm$.

Figura 30 – Distribuição de M_x obtida por meio do código implementado (dimensões da placa em cm)



Fonte: Autoria própria (2022)

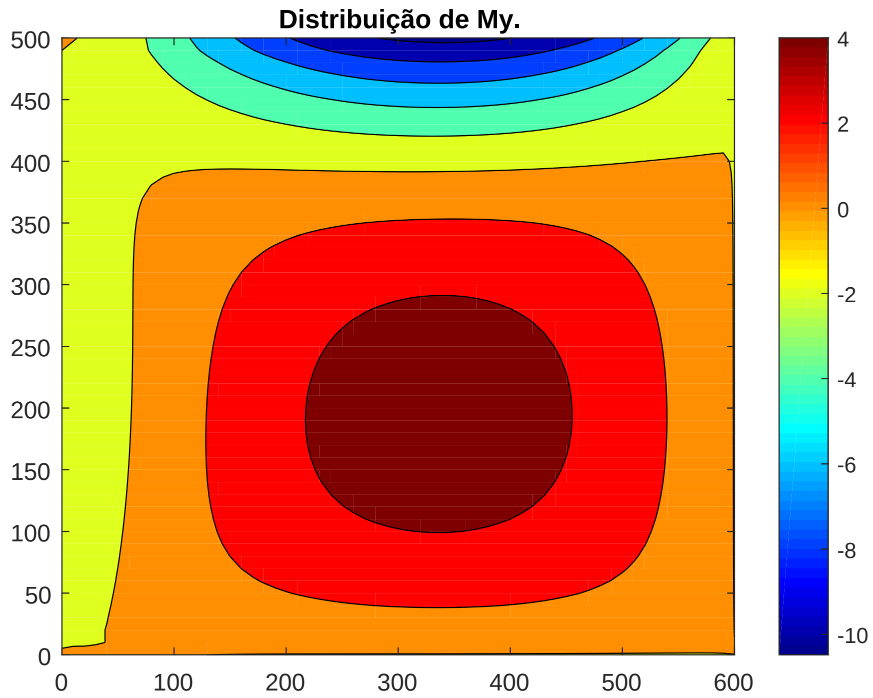
Figura 31 – Distribuição de M_x obtida por meio do SCIA Engineer



Fonte: SCIA Engineer (2022)

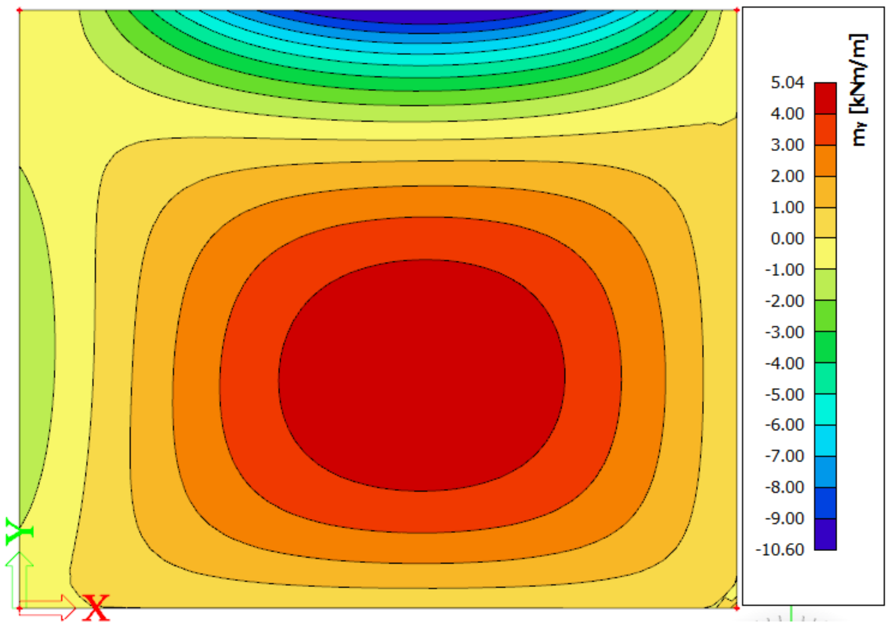
Da mesma forma, foram obtidos também os gráficos de momento para os momentos M_y (Figuras 32 e 33).

Figura 32 – Distribuição de M_y obtida por meio do código implementado (dimensões da placa em cm)



Fonte: Autoria própria (2022)

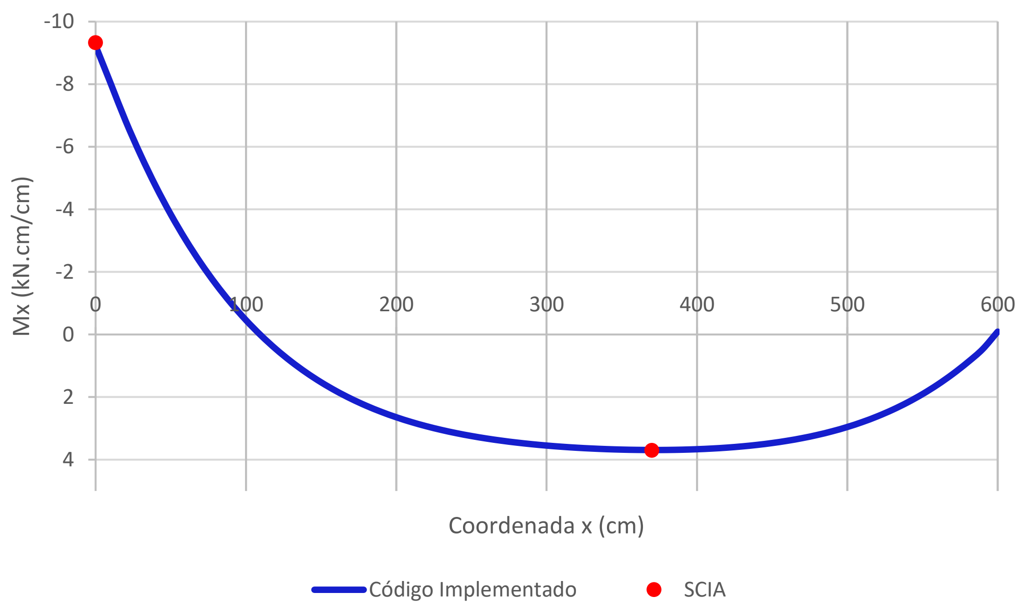
Figura 33 – Distribuição de M_y obtida por meio do SCIA Engineer .



Fonte: SCIA Engineer (2022)

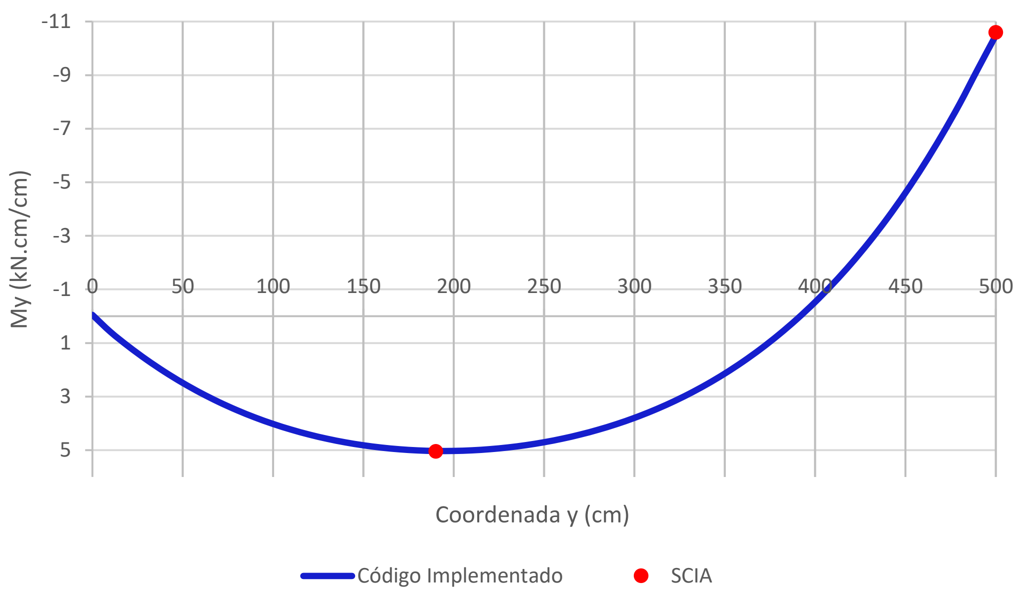
Para melhor visualização foram elaborados ainda gráficos comparativos entre os momentos obtidos pelo algoritmo implementado e os valores de momentos máximos fornecidos pelo SCIA. Os momentos foram tomados na seção mais crítica da placa nas direções x e y e posicionados no gráfico na forma de diagramas (Figuras 34 e 35).

Figura 34 – Momentos fletores M_x obtidos na seção y = 220 cm da placa por meio do código implementado e do SCIA Engineer



Fonte: Autoria própria (2022)

Figura 35 – Momentos fletores M_y obtidos na seção x = 350 cm da placa por meio do código implementado e do SCIA Engineer



Fonte: Autoria própria (2022)

Com a comparação de dados realizada por meio dos gráficos e tabelas anteriormente apresentados pode-se perceber que os resultados variaram muito pouco, em torno de 1,1 $\%$, em relação aos dados obtidos por meio do *software* SCIA Engineer. Dessa forma, é verificada mais uma vez a adequada implementação do algoritmo baseado no MEF, mesmo em situações nas quais as vinculações e dimensões da placa não são todas iguais.

O percentual de erro nesse caso se deve basicamente ao fato de no SCIA serem utilizados elementos retangulares, que possuem um nó e consequentemente 3 graus de liberdade por elemento a mais, o que culmina em um polinômio interpolador com mais termos, fornecendo uma maior precisão.

Quando comparados aos resultados obtidos na primeira tentativa, alguns resultados ficaram até mais distantes que os obtidos anteriormente, no entanto, em uma visão geral, na segunda tentativa os resultados apresentaram erros mais homogêneos, não apresentando um erro muito maior que os outros, como é o caso da primeira tentativa. Isso ocorre provavelmente por conta dos elementos possuírem dimensões maiores em um dos lados, na primeira tentativa, o que faz com que fiquem distorcidos. Além disso, foi notado que utilizando-se a segunda configuração, a solução demorou mais para convergir, assim fez-se o uso de um maior número de elementos na respectiva análise.

4.5 Exemplo 4 - Placas simplesmente apoiadas genéricas com diferentes relações entre suas dimensões.

Por fim, neste último exemplo, foram utilizadas três diferentes placas, com uma carga q uniformemente distribuída sobre as mesmas, módulo de young E, espessura t e coeficiente de Poisson 0,2. Entretanto, cada uma com relações entre o lado maior (ℓ_y) e o lado menor (ℓ_x) da placa, distintas, onde os valores podem ser comparados com os presentes nas tabelas de Bares e Czérny.

Como se sabe, após um certo valor da relação ℓ_y/ℓ_x, as tabelas apresentam dados constantes para os coeficientes, apresentando a relação como "∞", o qual sugere que após esse certo valor, não há alterações significativas nas expressões utilizadas para cálculo de deslocamentos e esforços. Assim, de modo a ter-se um valor de comparação, a primeira relação de ℓ_y/ℓ_x tomada foi uma para a qual os valores dos coeficientes utilizados nas equações fornecidas pelas tabelas eram conhecidos, ao passo que os outros dois valores da relação se encontram após o ponto mencionado no início do parágrafo.

A Tabela 5 mostra os coeficientes obtidos com o código implementado e os presentes nas tabelas de Bares e Czérny, com base na relação estabelecida no parágrafo anterior.

Tabela 5 – Comparação entre valores dos coeficientes das tabelas de Bares e Czérny com os valores obtidos por meio do código implementado para diferentes valores da relação ℓ_y/ℓ_x

	α			α_2		
ℓ_y/ℓ_x	**Tabela de Bares**	**Código (MEF)**	**Erro relativo** $(\%)$	**Tabela de Czérny**	**Código (MEF)**	**Erro relativo** $(\%)$
2,00	11,68	11,72	0,342	8,6	8,53	-0,814
5,00	15,35	14,96	-2,541	6,7	6,69	-0,149
10,00	15,35	15,01	-2,215	6,7	6,66	-0,597

Fonte: Autoria própria (2022)

Como pode-se notar, a variação dos valores dos coeficientes obtidos utilizando o código implementado foi bastante reduzida, com alguns valores se aproximando mais com os valores da tabela de Bares e alguns outros com os da tabela de Czérny. Ademais, o resultado obtido também mostra que as tabelas fornecem valores razoáveis quando a relação ℓ_y/ℓ_x é superior aos valores contidos nas mesmas.

Com base nessas premissas, obtém-se mais uma confirmação de que o código foi corretamente implementado e fornece resultados confiáveis, até mesmo em situações de placas retangulares em que um dos seus lados tem ordem de grandeza muito superior ao outro. Entretanto, caso fosse necessário, as malhas poderiam ser ainda mais refinadas o que certamente forneceria um resultado mais próximo das soluções apresentadas nas tabelas.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo fundamental a implementação computacional de uma formulação baseada no Método dos Elementos Finitos direcionada a análise linear de placas isotrópicas consideradas delgadas, buscando determinar os deslocamentos e esforços internos de momento fletores atuando nas mesmas.

A formulação utilizada exigiu que fosse realizada uma breve apresentação da teoria de placas delgadas juntamente com as hipóteses de Kirchhoff. O Princípio dos Trabalhos Virtuais foi utilizado para se determinar a matriz de rigidez de cada elemento, que foi obtida fazendo-se o uso de um polinômio interpolador de terceiro grau contendo nove termos para aproximar o campo de deslocamentos presente nos mesmos, bem como o uso de coordenadas triplas homogêneas.

Tendo em posse a teoria, deu-se início à programação de um código computacional implementado no *software* MATLAB®.

Uma vez implementado o algoritmo, partiu-se para a simulação numérica de quatro exemplos de placas submetidas a diferentes tipos de carregamentos e vinculações, para que se pudesse demonstrar a veracidade dos resultados obtidos. Os resultados foram comparadas por meio de tabelas e gráficos, sendo assim de mais fácil interpretação.

Alguns erros puderam ser notados quando os valores obtidos por meio do algoritmo foram comparados com valores fornecidos por outros meios, como as tabelas de Bares, Czérny e programas comerciais, como o SCIA Engineer. Foi notado também, que quando a placa é retangular e usa-se o mesmo número de elementos em ambas as direções, fazendo com que os elementos consequentemente se tornem maiores em uma dimensão que em outra, os resultados tendem a ser menos precisos em uma delas. Assim é necessária uma correção, sendo essa a aplicação de um maior número de elementos na maior dimensão, fazendo dessa forma com que a malha fique mais regular, ou seja, com dimensões semelhantes em ambas as direções. De maneira geral, entretanto, os erros se mostraram bastante reduzidos, estando dentro do esperado e assim denotando a correta implementação do código computacional.

Como sugestão para futuros trabalhos, pode-se utilizar o MEF para fazer a análise linear de cascas, que são aplicáveis em projeto de silos, por exemplo, ou ainda realizar a análise de placas com bordas curvas, as quais o elemento finito triangular seria especialmente aplicável.

REFERÊNCIAS

ALVES FILHO, A. **Elementos finitos:** A base da tecnologia CAE. 6. ed. São Paulo: Érica, 2013.

ALVES FILHO, A. **Elementos finitos:** A base da tecnologia CAE. São Paulo: Érica, 2000.

ASSAN, A. E. **Método dos elementos finitos:** primeiros passos. 2. ed. Campinas: Editora da Unicamp, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6118:** Projeto de estruturas de concreto: procedimento. Rio de Janeiro: ABNT, 2014.

AZEVEDO, A. F. M. **Método dos elementos finitos**. 1. ed. Porto: Faculdade de Engenharia do Porto, 2003. Disponível em: http://www.alvaroazevedo.com/publications/books/livro_mef_aa_1ed/doc/livro_mef_aa.pdf. Acesso em 28 de março de 2022.

BASTOS, P. S. S. **Lajes de concreto armado**. Bauru: Unesp, 2021. Disponível em: https://wwwp.feb.unesp.br/pbastos/concreto1/Lajes.pdf. Acesso em 12 de abril de 2022.

CAMPOS, M. D. O método de elementos finitos aplicado à simulação numérica de escoamentos de fluidos. *In*: III Bienal da SBM - IME/UFG. 2006, Goiânia. **Anais [...]** Goiânia: IME/UFG, 2006.

CARVALHO, R. C.; FIGUEIREDO FILHO, J. R. **Cálculo e detalhamento de estruturas usuais de concreto armado:** segundo a NBR 6118:2014. 4. ed. São Carlos: Edufscar, 2014.

COOK, R. D.; MALKUS, D. S.; PLESHA, M. E.; WITT, R. J. **Concepts and applications of finite element analysis.** 4. ed. Hoboken: John Wiley & Sons, Inc., 2002.

COSTA, H. B. **Elementos finitos (via resíduos ponderados) na resolução do problema de segunda ordem das placas.** 1986. Tese (Doutorado em Engenharia) - Escola Politécnica da Universidade de São Paulo. São Paulo, 1986.

DINIZ, A. L. S. M.; SANTOS, A. P. S.; ANDRADE, D. D. A.; SILVEIRA, J. C.; ALVARES, P. M. **Lajes de concreto - tipos, aplicações e vantagens**. 2013. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) Universidade Anhembi Morumbi. São Paulo, 2013.

HIBBELER, R. C. **Resistência dos materiais**. 7. ed. São Paulo: Pearson Prentice Hall, 2010.

LEAL, C. E. F. **Formulação do método dos elementos finitos para a análise elástica linear de placas delgadas**. 2015. Trabalho de Conclusão de Curso (Curso de Engenharia Civil) Universidade Tecnológica Federal do Paraná. Campo Mourão, 2015.

MARTHA, L. F. C. R. **Análise de estruturas:** conceitos e métodos básicos. Rio de Janeiro: Elsevier, 2010.

MARTINELLI, D. A. O.; MONTANARI, I.; SAVASSI, W. **Placas Elásticas:** equações gerais e placas retangulares. Conceituação teórica, métodos das diferenças finitas e elementos finitos. São Carlos: EESC, 1986. 114p. Publicação 026/86.

OÑATE, E. **Structural analysis with the finite element method:** linear statics. v. 1. Barcelona: Springer, 2009.

SORIANO, H. L. **Método de elementos finitos em análise de estruturas**. São Paulo: Editora da Universidade de São Paulo, 2002.

TIMOSHENKO, S.; GOODIER, J. N. **Theory of elasticity**. New York: McGraw-Hill Book Company Inc., 1934.

TIMOSHENKO, S.; WOINOWSKY-KRIEGER, S. **Theory of plates and shells**. 2nd edition. New York: McGraw-Hill Book Company Inc., 1959.

VAZ, L. E. **Método dos elementos finitos em análise de estruturas**. Rio de Janeiro: Elsevier, 2011.

VENTSEL, E.; KRAUTHAMMER, T. **Thin plates and shells:** theory, analysis and applications. New York: Marcel Dekker Inc., 2001.

WAIDEMAM, L. **Análise dinâmica de placas delgadas utilizando elementos finitos triangulares e retangulares**. 2004. Dissertação (Mestrado em Engenharia) - Programa de Pós Graduação em Engenharia Civil, Universidade Estadual Paulista Júlio de Mesquita Filho. Ilha Solteira, 2004.

ZENZEN, A. **Comparação de métodos de cálculo para determinação dos momentos fletores em lajes de concreto armado**. 2012. Trabalho de Conclusão de Curso (Curso de Graduação em Engenharia Civil) Universidade Regional do Noroeste do Estado do Rio Grande do Sul. Ijuí, 2012.

ZIENKIEWICKZ, O. C.; TAYLOR, R. L. **The finite element method:** the basis. 5. ed. v. 1. Woburn: ButterWorth-Heinemann, 2000.

APÊNDICE A - CÓDIGO FONTE DO PROGRAMA (COMENTADO)

```

clear all % Limpa a memoria
clc % Limpa o Command Window

p = 0; %Carga distribuida (em kN/cm^2)
E = 0;   % Modulo de Elasticidade Longitudinal (em kN/cm^2)
pss = 0;  % Coeficiente de Poisson (Adimensional)
t = 0; % Espessura da placa (em cm)

% Entrada:

%****** MONTAGEM DA MALHA - INICIO *******************************

 dimx = input('Qual a dimensao em x da placa?'); % Dimensao da placa em x
 dimy = input('Qual a dimensao em y da placa?'); % Dimensao da placa em y

 nnosx = input('Qual o numero de nos na direcao x da placa?'); % Numero de nos na malha em x
 nnosy = input('Qual o numero de nos na direcao y da placa?'); % Numero de nos na malha em y
 nelx = nnosx-1; % Numero de elementos em x
 nely = nnosy-1; % Numero de elementos em y
 nel = 2*(nnosx-1)*(nnosy-1);% Numero de elementos total
 nnostot = nnosx*nnosy;% Numero de nos total
 nnosel=3; % Numero de nos por elemento (sempre igual a 3)

 dx = dimx/(nnosx-1); % Dimensao do trecho entre cada no em x
 dy = dimy/(nnosy-1); % Dimensao do trecho entre cada no em y


 coordnos = zeros(2,(nnosx*nnosy)); % Cria matriz para conter as coordenadas x e y de cada no
```

```
 conect = zeros (3, 2*(nnosx-1)*(nnosy-1));% Cria matriz para
   conter o numero dos nos que formam cada elemento
 % Primeiro numero e 3, por cada elemento conter 3 nos e a segunda
   coordenada se refere ao numero total de elementos


 %CALCULANDO AS COORDENADAS DOS NOS:
 indice=0;

for i=1:nnosy;
    y = (i-1)*dy; % Valor de y na rodada
    for j=1:nnosx;
        x = (j-1)*dx; % Valor de x na rodada
       indice=indice+1; % Incremento de 1 na variavel indice
       coordnos(1,indice) = x; % Aplica o valor de x obtido na
   coordenada respectiva da matriz coordnos
       coordnos(2,indice) = y; % Aplica o valor de y obtido na
   coordenada respectiva da matriz coordnos
    end
end

indice=0;
for i=1:nnosy-1;

    for j=1:nnosx-1;

       indice1 = j + (i-1)*nnosx;
       indice2 = indice1 + 1;
       indice3 = indice2 + nnosx;
       indice4 = indice1 + nnosx;

       indice=indice+1; % Incremento de 1 na variavel indice
       conect(:,indice) = [indice1; indice3; indice4]; % Insere os
   valores obtidos na matriz conect
```

```

        indice=indice+1; % Incremento de 1 na variavel indice
        conect(:,indice) = [indice1; indice2; indice3]; % Insere os valores obtidos na matriz conect
    end
end
conect = conect'; % Transpoe a matriz para que seja possivel realizar operacoes
coordnos = coordnos'; % Transpoe a matriz para que seja possivel realizar operacoes


coordel = zeros(nel,6); % Cria a matriz coordel (coordenada dos elementos)

for j=1:nel;
        for i=1:3;
    coordel(j,i) = coordnos(conect(j,i),1); % Monta a parte onde os numeros estao na primeira coluna de coordnos
    coordel(j,i+3) = coordnos(conect(j,i),2); % Monta a parte onde os numeros estao na segunda coluna de coordnos
        end
end

nos = conect;
%****** MONTAGEM DA MALHA - FIM ********************************

% CARGA CONCENTRADA:

 ncc = input('Quantas cargas concentradas ha?');
conc = zeros(nnostot,5);

% Aplicando no vetor conc:
for i=1:ncc
```

```
    conc(i,1)=input('Digite a coordenada x da carga concentrada.');
    conc(i,2)=input('Digite a coordenada y da carga concentrada.');
    conc(i,3)=input('Digite o valor da forca concentrada.');
    conc(i,4)=input('Digite o valor do momento em x concentrado.');
    conc(i,5)=input('Digite o valor do momento em y concentrado.');
end

% Calcula o numero de graus de liberdade
ngdltot = 3*nnostot;
K = zeros(ngdltot,ngdltot); % Cria a matriz K (Matriz de rigidez
   global)
F = zeros(ngdltot,1); % Cria o vetor de for as F (Vetor de forcas
   global)
fc = zeros(ngdltot,1); % Cria o vetor de cargas concentradas

% Aplica a carga concentrada nos locais corretos:
 for j=1:ncc; % De 1 ate o numero de cargas concentradas
 for i=1:nnostot;   % De 1 ate o numero de nos total

if conc(j,1)==coordnos(i,1) && conc(j,2)==coordnos(i,2); %Se as
   coordenadas forem iguais para o no existente, aplica a carga
   nesse no
fc(3*i-2)=conc(j,3);
fc(3*i-1)=conc(j,4);
fc(3*i-0)=conc(j,5);
elseif i<ncc
    disp('As coordenadas nao coincidem, altere a malha.'); % Caso a
    coordenada onde a carga foi posicionada n o exista e preciso
   rodar novamente o algoritmo
end
 end
end
```

```
%****** CONDICOES DE CONTORNO - INICIO ************************

cc=zeros(nnostot,4);% Cria a matriz cc (condicoes de contorno)

for i=1:nnostot;
    cc(i,1) = i;
end
eng1=0;
eng2=0;
eng3=0;
eng4=0;
ap1=0;
ap2=0;
ap3=0;
ap4=0;

%COMO FUNCIONA?

% O codigo pede a condicao, se nao for nenhuma das duas pedidas, o
   bordo esta livre

%Condicao de contorno para engaste 1:

eng1 = input('Se os nos da primeira linha de elementos forem
   engastados digite 1, senao digite 0.');
if eng1 == 1
    cc(1:nnosx,2) = 1;
    cc(1:nnosx,3) = 1;
    cc(1:nnosx,4) = 1;
elseif eng1 == 0

%Condicao de contorno para apoio 1:
ap1 = input('Se os nos da primeira linha de elementos forem
```

```
    apoiados digite 1, senao digite 0.');%Linha de elementos de x=1:
    ultimo, y = 0
if ap1 == 1
    cc(1:nnosx,2) = 1; % 2 e relacionado a deflexao
    cc(1:nnosx,3) = 0; % 3 e relacionado ao giro em x
    cc(1:nnosx,4) = 1; % 4 e relacionado ao giro em y
elseif ap1 == 0
    cc(1:nnosx,2) = 0;
    cc(1:nnosx,3) = 0;
    cc(1:nnosx,4) = 0;
    else
    disp('Rode novamente o algoritmo.');
    end
end

 %Condicao de contorno para engaste 2:
eng2 = input('Se os nos da ultima linha de elementos forem
    engastados digite 1, senao digite 0.');
if eng2 == 1
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),2) = 1;
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),3) = 1;
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),4) = 1;
elseif eng2 == 0

%Condicao de contorno para apoio 2:
ap2 = input('Se os nos da ultima linha de elementos forem apoiados
    digite 1, senao digite 0.');%Linha de elementos de x=1:ultimo, y
     = ultimo
if ap2 == 1
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),2) = 1;
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),3) = 0;
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),4) = 1;
elseif ap2 == 0
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),2) = 0;
```

```
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),3) = 0;
    cc((nnosy*nnosx)-(nnosx-1):(nnosy*nnosx),4) = 0;
    else
    disp('Cancele e Rode novamente o algoritmo.');
end
end
%Condicao de contorno para engaste 3:

eng3 = input('Se os nos da primeira coluna de elementos forem
   engastados digite 1, senao digite 0.');
if eng3 == 1
    for i=2:nnosy-1;
    cc(nnosx*i-nelx,2) = 1;
    cc(nnosx*i-nelx,3) = 1;
    cc(nnosx*i-nelx,4) = 1;
    end
      % Se o nos dos cantos inferior e superior forem livres tem-se
    que engastar os nos das extremidades:
    if cc(1,2)==0 | cc((nnosy*nnosx)-(nnosx-1),2)==0
       cc(1,2)=1;
       cc(1,3)=1;
       cc(1,4)=1;
       cc((nnosy*nnosx)-(nnosx-1),2)=1;
       cc((nnosy*nnosx)-(nnosx-1),3)=1;
       cc((nnosy*nnosx)-(nnosx-1),4)=1;
    end


elseif eng3 == 0

% Condicao de contorno para apoio 3:

ap3 = input('Se os nos da primeira coluna de elementos forem
   apoiados digite 1, senao digite 0.');
```

```
if ap3 == 1
    for i=2:nnosy-1;
    cc(nnosx*i-nelx,2) = 1;
    cc(nnosx*i-nelx,3) = 1;
    cc(nnosx*i-nelx,4) = 0;
    end
      % Se o nos dos cantos inferior e superior forem livres temos
   que engastar os nos das extremidades:
    if cc(1,2)==0 | cc((nnosy*nnosx)-(nnosx-1),2)==0
       cc(1,2)=1;
       cc(1,3)=1;
       cc(1,4)=0;
       cc((nnosy*nnosx)-(nnosx-1),2)=1;
       cc((nnosy*nnosx)-(nnosx-1),3)=1;
       cc((nnosy*nnosx)-(nnosx-1),4)=0;
    end
end

elseif ap3 == 0
        for i=2:nnosy-1;
    cc(nnosx*i-nelx,2) = 0;
    cc(nnosx*i-nelx,3) = 0;
    cc(nnosx*i-nelx,4) = 0;
        end
    else
    disp('Cancele e Rode novamente o algoritmo.');
  end


  % Condicao de contorno para engaste 4:
eng4 = input('Se os nos da ultima coluna de elementos forem
   engastados digite 1, senao digite 0.');
if eng4 == 1
    for i=2:nnosy-1;
```

```
    cc(nnosx*i,2) = 1;
    cc(nnosx*i,3) = 1;
    cc(nnosx*i,4) = 1;
    end
    % Se o nos dos cantos inferior e superior forem livres tem-se
   que engastar os nos das extremidades:
    if cc(nnosx,2)==0 | cc(nnosx*nnosy,2)==0
        cc(nnosx,2)=1;
        cc(nnosx,3)=1;
        cc(nnosx,4)=1;
        cc(nnosx*nnosy,2)=1;
        cc(nnosx*nnosy,3)=1;
        cc(nnosx*nnosy,4)=1;
    end


elseif eng4 == 0

 % Condicao de contorno para apoio 4:
ap4 = input('Se os nos da ultima coluna de elementos forem apoiados
     digite 1, senao digite 0.');
if ap4 == 1
    for i=2:nnosy-1;
    cc(nnosx*i,2) = 1;
    cc(nnosx*i,3) = 1;
    cc(nnosx*i,4) = 0;
    end
    % Se o nos dos cantos inferior e superior forem livres tem-se
   que engastar os nos das extremidades:
    if cc(nnosx,2)==0 | cc(nnosx*nnosy,2)==0
        cc(nnosx,2)=1; % Nos do canto (nnosx,0)
        cc(nnosx,3)=1;
        cc(nnosx,4)=0;
        cc(nnosx*nnosy,2)=1;% Nos do canto (nnosx,nnosy)
```

```
        cc(nnosx*nnosy,3)=1;
        cc(nnosx*nnosy,4)=0;
    end


elseif ap4 == 0
        for i=2:nnosy-1;
    cc(nnosx*i,2) = 0;
    cc(nnosx*i,3) = 0;
    cc(nnosx*i,4) = 0;
        end
    else
    disp('Cancele e Rode novamente o algoritmo.');
end
end
%Precisa-se aplicar as restricoes nos nos dos cantos para compensar
    os zeros impostos nas condicionais ap1 e ap2
if ap1 == 1 | ap2 == 1 | ap3 == 1 | ap4 == 1 | eng1==1 | eng2==1 |
   eng3 == 1 | eng4 == 1

        %No do canto (0,0)
        cc(1,2)=1;
        cc(1,3)=1;
        cc(1,4)=1;
        %No do canto (0,nnosy)
        cc((nnosy*nnosx)-(nnosx-1),2)=1;
        cc((nnosy*nnosx)-(nnosx-1),3)=1;
        cc((nnosy*nnosx)-(nnosx-1),4)=1;
        %No do canto (nnosx,0)
        cc(nnosx,2)=1;
        cc(nnosx,3)=1;
        cc(nnosx,4)=1;
        %No do canto (nnosx,nnosy)
        cc(nnosx*nnosy,2)=1;
```

```
        cc(nnosx*nnosy,3)=1;
        cc(nnosx*nnosy,4)=1;

elseif ap1 == 1 & ap2 == 1 & ap3 == 1

        %No do canto (0,0)
        cc(1,2)=1;
        cc(1,3)=1;
        cc(1,4)=1;
        %No do canto (0,nnosy)
        cc((nnosy*nnosx)-(nnosx-1),2)=1;
        cc((nnosy*nnosx)-(nnosx-1),3)=1;
        cc((nnosy*nnosx)-(nnosx-1),4)=1;
        %No do canto (nnosx,0)
        cc(nnosx,2)=1;
        cc(nnosx,3)=1;
        cc(nnosx,4)=1;
        %No do canto (nnosx,nnosy)
        cc(nnosx*nnosy,2)=1;
        cc(nnosx*nnosy,3)=1;
        cc(nnosx*nnosy,4)=1;

elseif ap1 == 1 & ap2 == 1 & ap4 == 1

        %No do canto (0,0)
        cc(1,2)=1;
        cc(1,3)=1;
        cc(1,4)=1;
        %No do canto (0,nnosy)
        cc((nnosy*nnosx)-(nnosx-1),2)=1;
        cc((nnosy*nnosx)-(nnosx-1),3)=1;
        cc((nnosy*nnosx)-(nnosx-1),4)=1;
        %No do canto (nnosx,0)
        cc(nnosx,2)=1;
```

```
        cc(nnosx,3)=1;
        cc(nnosx,4)=1;
        %No do canto (nnosx,nnosy)
        cc(nnosx*nnosy,2)=1;
        cc(nnosx*nnosy,3)=1;
        cc(nnosx*nnosy,4)=1;

       end


%****** CONDICOES DE CONTORNO - FIM *************************


for i=1:nel

%Matriz contendo as areas de cada elemento:

%Calculando os termos da matriz "Ael":

Ael(i) = (coordel(i,2)*coordel(i,6) + coordel(i,1)*coordel(i,5) +
   coordel(i,3)*coordel(i,4)...
    - coordel(i,2)*coordel(i,4) - coordel(i,3)*coordel(i,5) -
   coordel(i,1)*coordel(i,6))/2;

%Matriz constitutiva do material:

EE = (E/(1-pss^2))*[1 pss 0; pss 1 0; 0 0 ((1-pss)/2)];

%Declarando as funcoes de forma:
syms xi1 xi2 xi3

%Calculo dos coeficientes "a", "b" e "c":

    a(1) = coordel(i,2)*coordel(i,6) - coordel(i,3)*coordel(i,5);
```

```
    b(1) = coordel(i,5) - coordel(i,6);
    c(1) = coordel(i,3) - coordel(i,2);

    a(2) = coordel(i,3)*coordel(i,4) - coordel(i,1)*coordel(i,3);
    b(2) = coordel(i,6) - coordel(i,4);
    c(2) = coordel(i,1) - coordel(i,3);

    a(3) = coordel(i,1)*coordel(i,5) - coordel(i,2)*coordel(i,4);
    b(3) = coordel(i,4) - coordel(i,5);
    c(3) = coordel(i,2) - coordel(i,1);

%Criacao das funcoes de phi1 a phi9 para cada elemento

phi1 = (xi1 - (xi1*xi2^2 - xi1^2*xi2)+(xi3*xi1^2 - xi3^2*xi1));
phi2 = (1/2)*(-b(3)*xi1*xi2 + b(2)*xi3*xi1 + b(3)*(xi1*xi2^2 - xi1
    ^2*xi2) + b(2)*(xi3*xi1^2 - xi3^2*xi1));
phi3 = (1/2)*(c(3)*xi1*xi2 - c(2)*xi3*xi1 - c(3)*(xi1*xi2^2 - xi1
    ^2*xi2) - c(2)*(xi3*xi1^2 - xi3^2*xi1));
phi4 = xi2 + (xi1*xi2^2 - xi1^2*xi2) - (xi2*xi3^2 - xi2^2*xi3);
phi5 = (1/2)*(b(3)*xi1*xi2 - b(1)*xi2*xi3 + b(3)*(xi1*xi2^2 - xi1
    ^2*xi2) + b(1)*(xi2*xi3^2 - xi2^2*xi3));
phi6 = (1/2)*(-c(3)*xi1*xi2 + c(1)*xi2*xi3 - c(3)*(xi1*xi2^2 - xi1
    ^2*xi2) - c(1)*(xi2*xi3^2 - xi2^2*xi3));
phi7 = xi3 + (xi2*xi3^2 - xi2^2*xi3) - (xi3*xi1^2 - xi3^2*xi1);
phi8 = (1/2)*(b(1)*xi2*xi3 - b(2)*xi3*xi1 + b(1)*(xi2*xi3^2 - xi2
    ^2*xi3) + b(2)*(xi3*xi1^2 - xi3^2*xi1));
phi9 = (1/2)*(-c(1)*xi2*xi3 + c(2)*xi3*xi1 - c(1)*(xi2*xi3^2 - xi2
    ^2*xi3) - c(2)*(xi3*xi1^2 - xi3^2*xi1));

%Aplicando dados na matriz funcoes de forma:

phi(1,1) = phi1;
phi(1,2) = phi2;
phi(1,3) = phi3;
```

```
phi(1,4) = phi4;
phi(1,5) = phi5;
phi(1,6) = phi6;
phi(1,7) = phi7;
phi(1,8) = phi8;
phi(1,9) = phi9;

%Variaveis m (derivadas parciais) que serao contidas na matriz M (
    produto [l]*[phi]:

%Primeira linha da matriz de derivadas:
    for j=1:9
  m1(j) = b(1)^2*(diff(diff(phi(1,j),xi1),xi1)) + b(2)^2*(diff(diff
    (phi(1,j),xi2),xi2)) +...
  b(3)^2*(diff(diff(phi(1,j),xi3),xi3)) + 2*b(1)*b(2)*(diff(diff(
    phi(1,j),xi1),xi2))+...
  2*b(1)*b(3)*(diff(diff(phi(1,j),xi3),xi1)) + 2*b(2)*b(3)*(diff(
    diff(phi(1,j),xi3),xi2));
    end

%Segunda linha da matriz de derivadas:
    for j=1:9
   m2(j) = c(1)^2*(diff(diff(phi(1,j),xi1),xi1)) + c(2)^2*(diff(
   diff(phi(1,j),xi2),xi2)) +...
   c(3)^2*(diff(diff(phi(1,j),xi3),xi3)) + 2*c(1)*c(2)*(diff(diff(
   phi(1,j),xi1),xi2))+...
   2*c(1)*c(3)*(diff(diff(phi(1,j),xi3),xi1)) + 2*c(2)*c(3)*(diff(
   diff(phi(1,j),xi3),xi2));
    end

%Terceira linha da matriz de derivadas:
      for j=1:9
   m3(j) = 2*(b(1)*c(1)*(diff(diff(phi(1,j),xi1),xi1)) + b(2)*c(2)
   *(diff(diff(phi(1,j),xi2),xi2)) +...
```

```
        b(3)*c(3)*(diff(diff(phi(1,j),xi3),xi3)) + (b(1)*c(2) + b(2)
    *c(1))*(diff(diff(phi(1,j),xi2),xi1)) + ...
        (b(1)*c(3) + b(3)*c(1))*(diff(diff(phi(1,j),xi3),xi1)) + (b
    (2)*c(3) + b(3)*c(2))*(diff(diff(phi(1,j),xi3),xi2)));
      end

%Criacao das matrizes M(3,9) que contem o produto [l]*[phi]:

for j=1:9
    M(1,j) = m1(j);
    M(2,j) = m2(j);
    M(3,j) = m3(j);
end

%Cria uma matriz simbolica generica MT:

MT = sym('MT',[9 3]);

%Transposta de M (Aplica os valores na matriz simbolica generica
    criada):
for j=1:9;
    for k=1:3;

if MT(j,k) == M(k,j);
    MT(j,k) = M(k,j);
elseif MT(j,k) ~= M(k,j);
    MT(j,k) = M(k,j);
end
end
end


%Produto ([l][phi])^T * EE * ([l]*[phi])
%Substituicao de variaveis, para M e MT dependerem apenas de xi1 e
```

```
   xi2:

 M = subs(M, xi3, 1-xi1-xi2);
 MT = subs(MT, xi3, 1-xi1-xi2);

 %Criacao da funcao que depende de xi1 e xi2:

 f_xi1xi2(:,:) = MT(:,:)*EE(:,:)*M(:,:);

%Calculo da integral dupla(Presente no texto):

integral(:,:) = int((int(f_xi1xi2(:,:), 0, 1-xi2)),0,1);

%Converte a matriz integral de fracao para decimal:

 integral = double(integral);

%Calculo do produto antes da integral (Presente no texto):

pp(i) = (1/(16*(Ael(i)^4)))*((t^3)/12)*(2*Ael(i));

%Montagem da Matriz de rigidez para cada elemento:

k = zeros(9,9);

%Primeira linha:
k(1,1) = (pp(i)*integral(1,1));
k(1,2) = (pp(i)*integral(1,2));
k(1,3) = (pp(i)*integral(1,3));
k(1,4) = (pp(i)*integral(1,4));
k(1,5) = (pp(i)*integral(1,5));
k(1,6) = (pp(i)*integral(1,6));
k(1,7) = (pp(i)*integral(1,7));
k(1,8) = (pp(i)*integral(1,8));
```

```
k(1,9) = (pp(i)*integral(1,9));

%Segunda linha:
k(2,1) = (pp(i)*integral(2,1));
k(2,2) = (pp(i)*integral(2,2));
k(2,3) = (pp(i)*integral(2,3));
k(2,4) = (pp(i)*integral(2,4));
k(2,5) = (pp(i)*integral(2,5));
k(2,6) = (pp(i)*integral(2,6));
k(2,7) = (pp(i)*integral(2,7));
k(2,8) = (pp(i)*integral(2,8));
k(2,9) = (pp(i)*integral(2,9));

%Terceira linha:
k(3,1) = (pp(i)*integral(3,1));
k(3,2) = (pp(i)*integral(3,2));
k(3,3) = (pp(i)*integral(3,3));
k(3,4) = (pp(i)*integral(3,4));
k(3,5) = (pp(i)*integral(3,5));
k(3,6) = (pp(i)*integral(3,6));
k(3,7) = (pp(i)*integral(3,7));
k(3,8) = (pp(i)*integral(3,8));
k(3,9) = (pp(i)*integral(3,9));

%Quarta linha:
k(4,1) = (pp(i)*integral(4,1));
k(4,2) = (pp(i)*integral(4,2));
k(4,3) = (pp(i)*integral(4,3));
k(4,4) = (pp(i)*integral(4,4));
k(4,5) = (pp(i)*integral(4,5));
k(4,6) = (pp(i)*integral(4,6));
k(4,7) = (pp(i)*integral(4,7));
k(4,8) = (pp(i)*integral(4,8));
k(4,9) = (pp(i)*integral(4,9));
```

```
%Quinta linha:
k(5,1) = (pp(i)*integral(5,1));
k(5,2) = (pp(i)*integral(5,2));
k(5,3) = (pp(i)*integral(5,3));
k(5,4) = (pp(i)*integral(5,4));
k(5,5) = (pp(i)*integral(5,5));
k(5,6) = (pp(i)*integral(5,6));
k(5,7) = (pp(i)*integral(5,7));
k(5,8) = (pp(i)*integral(5,8));
k(5,9) = (pp(i)*integral(5,9));

%Sexta linha:
k(6,1) = (pp(i)*integral(6,1));
k(6,2) = (pp(i)*integral(6,2));
k(6,3) = (pp(i)*integral(6,3));
k(6,4) = (pp(i)*integral(6,4));
k(6,5) = (pp(i)*integral(6,5));
k(6,6) = (pp(i)*integral(6,6));
k(6,7) = (pp(i)*integral(6,7));
k(6,8) = (pp(i)*integral(6,8));
k(6,9) = (pp(i)*integral(6,9));

%Setima linha:
k(7,1) = (pp(i)*integral(7,1));
k(7,2) = (pp(i)*integral(7,2));
k(7,3) = (pp(i)*integral(7,3));
k(7,4) = (pp(i)*integral(7,4));
k(7,5) = (pp(i)*integral(7,5));
k(7,6) = (pp(i)*integral(7,6));
k(7,7) = (pp(i)*integral(7,7));
k(7,8) = (pp(i)*integral(7,8));
k(7,9) = (pp(i)*integral(7,9));
```

```
%Oitava linha:
k(8,1) = (pp(i)*integral(8,1));
k(8,2) = (pp(i)*integral(8,2));
k(8,3) = (pp(i)*integral(8,3));
k(8,4) = (pp(i)*integral(8,4));
k(8,5) = (pp(i)*integral(8,5));
k(8,6) = (pp(i)*integral(8,6));
k(8,7) = (pp(i)*integral(8,7));
k(8,8) = (pp(i)*integral(8,8));
k(8,9) = (pp(i)*integral(8,9));

%Nona linha:
k(9,1) = (pp(i)*integral(9,1));
k(9,2) = (pp(i)*integral(9,2));
k(9,3) = (pp(i)*integral(9,3));
k(9,4) = (pp(i)*integral(9,4));
k(9,5) = (pp(i)*integral(9,5));
k(9,6) = (pp(i)*integral(9,6));
k(9,7) = (pp(i)*integral(9,7));
k(9,8) = (pp(i)*integral(9,8));
k(9,9) = (pp(i)*integral(9,9));

%Montagem da Matriz de Rigidez Global:

%Aplica os valores de k(9,9)(Matriz de rigidez elementar) em K(
   ngdltot,ngdltot)(Matriz de rigidez global):

  noi = nos(i,1); %Numero do no inicial
  noint = nos(i,2); %Numero do no intermediario
  nof = nos(i,3); %Numero do no final

%Primeira linha
  K(3*noi-2,3*noi-2)     = K(3*noi-2,3*noi-2)     + k(1,1);
  K(3*noi-2,3*noi-1)     = K(3*noi-2,3*noi-1)     + k(1,2);
```

```
 K(3*noi-2,3*noi-0)      = K(3*noi-2,3*noi-0)      + k(1,3);
 K(3*noi-2,3*noint-2)    = K(3*noi-2,3*noint-2)    + k(1,4);
 K(3*noi-2,3*noint-1)    = K(3*noi-2,3*noint-1)    + k(1,5);
 K(3*noi-2,3*noint-0)    = K(3*noi-2,3*noint-0)    + k(1,6);
 K(3*noi-2,3*nof-2)      = K(3*noi-2,3*nof-2)      + k(1,7);
 K(3*noi-2,3*nof-1)      = K(3*noi-2,3*nof-1)      + k(1,8);
 K(3*noi-2,3*nof-0)      = K(3*noi-2,3*nof-0)      + k(1,9);

%Segunda linha
 K(3*noi-1,3*noi-2)      = K(3*noi-1,3*noi-2)      + k(2,1);
 K(3*noi-1,3*noi-1)      = K(3*noi-1,3*noi-1)      + k(2,2);
 K(3*noi-1,3*noi-0)      = K(3*noi-1,3*noi-0)      + k(2,3);
 K(3*noi-1,3*noint-2)    = K(3*noi-1,3*noint-2)    + k(2,4);
 K(3*noi-1,3*noint-1)    = K(3*noi-1,3*noint-1)    + k(2,5);
 K(3*noi-1,3*noint-0)    = K(3*noi-1,3*noint-0)    + k(2,6);
 K(3*noi-1,3*nof-2)      = K(3*noi-1,3*nof-2)      + k(2,7);
 K(3*noi-1,3*nof-1)      = K(3*noi-1,3*nof-1)      + k(2,8);
 K(3*noi-1,3*nof-0)      = K(3*noi-1,3*nof-0)      + k(2,9);

%Terceira linha
 K(3*noi-0,3*noi-2)      = K(3*noi-0,3*noi-2)      + k(3,1);
 K(3*noi-0,3*noi-1)      = K(3*noi-0,3*noi-1)      + k(3,2);
 K(3*noi-0,3*noi-0)      = K(3*noi-0,3*noi-0)      + k(3,3);
 K(3*noi-0,3*noint-2)    = K(3*noi-0,3*noint-2)    + k(3,4);
 K(3*noi-0,3*noint-1)    = K(3*noi-0,3*noint-1)    + k(3,5);
 K(3*noi-0,3*noint-0)    = K(3*noi-0,3*noint-0)    + k(3,6);
 K(3*noi-0,3*nof-2)      = K(3*noi-0,3*nof-2)      + k(3,7);
 K(3*noi-0,3*nof-1)      = K(3*noi-0,3*nof-1)      + k(3,8);
 K(3*noi-0,3*nof-0)      = K(3*noi-0,3*nof-0)      + k(3,9);

%Quarta linha
 K(3*noint-2,3*noi-2)    = K(3*noint-2,3*noi-2)    + k(4,1);
 K(3*noint-2,3*noi-1)    = K(3*noint-2,3*noi-1)    + k(4,2);
 K(3*noint-2,3*noi-0)    = K(3*noint-2,3*noi-0)    + k(4,3);
```

```
K(3*noint-2,3*noint-2) = K(3*noint-2,3*noint-2) + k(4,4);
K(3*noint-2,3*noint-1) = K(3*noint-2,3*noint-1) + k(4,5);
K(3*noint-2,3*noint-0) = K(3*noint-2,3*noint-0) + k(4,6);
K(3*noint-2,3*nof-2)   = K(3*noint-2,3*nof-2)   + k(4,7);
K(3*noint-2,3*nof-1)   = K(3*noint-2,3*nof-1)   + k(4,8);
K(3*noint-2,3*nof-0)   = K(3*noint-2,3*nof-0)   + k(4,9);

%Quinta linha
K(3*noint-1,3*noi-2)   = K(3*noint-1,3*noi-2)   + k(5,1);
K(3*noint-1,3*noi-1)   = K(3*noint-1,3*noi-1)   + k(5,2);
K(3*noint-1,3*noi-0)   = K(3*noint-1,3*noi-0)   + k(5,3);
K(3*noint-1,3*noint-2) = K(3*noint-1,3*noint-2) + k(5,4);
K(3*noint-1,3*noint-1) = K(3*noint-1,3*noint-1) + k(5,5);
K(3*noint-1,3*noint-0) = K(3*noint-1,3*noint-0) + k(5,6);
K(3*noint-1,3*nof-2)   = K(3*noint-1,3*nof-2)   + k(5,7);
K(3*noint-1,3*nof-1)   = K(3*noint-1,3*nof-1)   + k(5,8);
K(3*noint-1,3*nof-0)   = K(3*noint-1,3*nof-0)   + k(5,9);

%Sexta linha
K(3*noint-0,3*noi-2)   = K(3*noint-0,3*noi-2)   + k(6,1);
K(3*noint-0,3*noi-1)   = K(3*noint-0,3*noi-1)   + k(6,2);
K(3*noint-0,3*noi-0)   = K(3*noint-0,3*noi-0)   + k(6,3);
K(3*noint-0,3*noint-2) = K(3*noint-0,3*noint-2) + k(6,4);
K(3*noint-0,3*noint-1) = K(3*noint-0,3*noint-1) + k(6,5);
K(3*noint-0,3*noint-0) = K(3*noint-0,3*noint-0) + k(6,6);
K(3*noint-0,3*nof-2)   = K(3*noint-0,3*nof-2)   + k(6,7);
K(3*noint-0,3*nof-1)   = K(3*noint-0,3*nof-1)   + k(6,8);
K(3*noint-0,3*nof-0)   = K(3*noint-0,3*nof-0)   + k(6,9);

%Setima linha
K(3*nof-2,3*noi-2)     = K(3*nof-2,3*noi-2)     + k(7,1);
K(3*nof-2,3*noi-1)     = K(3*nof-2,3*noi-1)     + k(7,2);
K(3*nof-2,3*noi-0)     = K(3*nof-2,3*noi-0)     + k(7,3);
K(3*nof-2,3*noint-2)   = K(3*nof-2,3*noint-2)   + k(7,4);
```

```
 K(3*nof-2,3*noint-1)   = K(3*nof-2,3*noint-1)   + k(7,5);
 K(3*nof-2,3*noint-0)   = K(3*nof-2,3*noint-0)   + k(7,6);
 K(3*nof-2,3*nof-2)     = K(3*nof-2,3*nof-2)     + k(7,7);
 K(3*nof-2,3*nof-1)     = K(3*nof-2,3*nof-1)     + k(7,8);
 K(3*nof-2,3*nof-0)     = K(3*nof-2,3*nof-0)     + k(7,9);

%Oitava linha
 K(3*nof-1,3*noi-2)     = K(3*nof-1,3*noi-2)     + k(8,1);
 K(3*nof-1,3*noi-1)     = K(3*nof-1,3*noi-1)     + k(8,2);
 K(3*nof-1,3*noi-0)     = K(3*nof-1,3*noi-0)     + k(8,3);
 K(3*nof-1,3*noint-2)   = K(3*nof-1,3*noint-2)   + k(8,4);
 K(3*nof-1,3*noint-1)   = K(3*nof-1,3*noint-1)   + k(8,5);
 K(3*nof-1,3*noint-0)   = K(3*nof-1,3*noint-0)   + k(8,6);
 K(3*nof-1,3*nof-2)     = K(3*nof-1,3*nof-2)     + k(8,7);
 K(3*nof-1,3*nof-1)     = K(3*nof-1,3*nof-1)     + k(8,8);
 K(3*nof-1,3*nof-0)     = K(3*nof-1,3*nof-0)     + k(8,9);

%Nona linha
 K(3*nof-0,3*noi-2)     = K(3*nof-0,3*noi-2)     + k(9,1);
 K(3*nof-0,3*noi-1)     = K(3*nof-0,3*noi-1)     + k(9,2);
 K(3*nof-0,3*noi-0)     = K(3*nof-0,3*noi-0)     + k(9,3);
 K(3*nof-0,3*noint-2)   = K(3*nof-0,3*noint-2)   + k(9,4);
 K(3*nof-0,3*noint-1)   = K(3*nof-0,3*noint-1)   + k(9,5);
 K(3*nof-0,3*noint-0)   = K(3*nof-0,3*noint-0)   + k(9,6);
 K(3*nof-0,3*nof-2)     = K(3*nof-0,3*nof-2)     + k(9,7);
 K(3*nof-0,3*nof-1)     = K(3*nof-0,3*nof-1)     + k(9,8);
 K(3*nof-0,3*nof-0)     = K(3*nof-0,3*nof-0)     + k(9,9);

%VETORES DE FORCAS:

%Calcula a integral para aplicar o carregamento distribuido

%Transpondo phi separadamente

```

```
 phi = subs(phi, xi3, 1-xi1-xi2);

 phiT = sym('phiT',[9 1]);

 for j=1:9;
    for k=1;

if phiT(j,k) == phi(k,j);
    phiT(j,k) = phi(k,j);
elseif phiT(j,k) ~= phi(k,j);
    phiT(j,k) = phi(k,j);
end
end
 end


%Primeira parte da integral:

ppint2(i) = 2*Ael(i)*p;

%Segunda parte da integral:

int2(:,1) = int((int(phiT(:,1), 0, 1-xi2)), 0, 1);

fd(:,1) = ppint2(i)*int2(:,1);%Multiplica as duas partes da
   integral

%Fazer a sobreposicao dos vetores de forca

%Primeira linha
  F(3*noi-2) = F(3*noi-2) + fd(1);

%Segunda linha
  F(3*noi-1) = F(3*noi-1) + fd(2);
```

```
%Terceira linha
  F(3*noi-0) = F(3*noi-0) + fd(3);

%Quarta linha
  F(3*noint-2) = F(3*noint-2) + fd(4);

%Quinta linha
  F(3*noint-1) = F(3*noint-1) + fd(5);

%Sexta linha
  F(3*noint-0) = F(3*noint-0) + fd(6);

%Setima linha
  F(3*nof-2) = F(3*nof-2) + fd(7);

%Oitava linha
  F(3*nof-1) = F(3*nof-1) + fd(8);

%Nona linha
  F(3*nof-0) = F(3*nof-0) + fd(9);

end

KCC = K ; %Criacao da matriz de rigidez global com condicoes de
    contorno

 for i = 1:nnostot ;

     %Aplica condicoes de contorno no vetor de for as:

     if cc(i,2) == 1; %Aplica as condicoes de contorno de
    deslocamentos
         F(3*i-2)=0 ;
```

```
    end

    if cc(i,3) == 1; %Aplica as condicoes de contorno de giro em x
       F(3*i-1)=0;
    end

    if cc(i,4) == 1; %Aplica as condicoes de contorno de giro em y
       F(3*i-0)=0;
    end

%Aplica condicoes de contorno na matriz de rigidez:

    if cc(i,2) == 1; %Aplica as condicoes de contorno de
  deslocamentos
       for j = 1:ngdltot;
       KCC(3*i-2,j)=0;
       KCC(j,3*i-2)=0;
       end
       KCC(3*i-2,3*i-2)=1;%Aplica o valor 1 na diagonal principal
  referentes aos nos restringidos
    end


    if cc(i,3) == 1; %Aplica as condicoes de contorno de giro em x
       for j = 1:ngdltot;
       KCC(3*i-1,j)=0;
       KCC(j,3*i-1)=0;
       end
       KCC(3*i-1,3*i-1)=1;%Aplica o valor 1 na diagonal principal
  referentes aos nos restringidos
    end


    if cc(i,4) == 1; %Aplica as condicoes de contorno de giro em y
```

```
          for j = 1:ngdltot;
         KCC(3*i-0,j)=0;
         KCC(j,3*i-0)=0;
          end
         KCC(3*i-0,3*i-0)=1; %Aplica o valor 1 na diagonal principal
     referentes aos nos restringidos
      end
 end

 F = F + fc; %Soma as forcas oriundas do carregamento distribuido
    com as concentradas

 U = inv(KCC)*F; %Resolve o sistema de equacoes

 %CRIA A MATRIZ PARA PLOTAR OS DESLOCAMENTOS

k=1;
while k<=nnostot;
    for i=1:nnosy;
        for j=1:nnosx;
            w(i,j)=U(3*k-2);
            k=k+1;
        end
    end
end


%ESFORCOS INTERNOS DE MOMENTO:

%Separando o vetor deslocamento geral em vetores deslocamento para
   cada elemento:

mom = zeros(ngdltot,1);
u=zeros(9,1);
```

```
for i=1:nel
    u(1) = U(3*(conect(i,1))-2);%Coloca na coluna 1 do vetor u a parte do vetor U referente aos nos do elemento 1.
    u(2) = U(3*(conect(i,1))-1);%Coloca na coluna 2 do vetor u a parte do vetor U referente aos nos do elemento 2.
    u(3) = U(3*(conect(i,1))-0);
    u(4) = U(3*(conect(i,2))-2);
    u(5) = U(3*(conect(i,2))-1);
    u(6) = U(3*(conect(i,2))-0);
    u(7) = U(3*(conect(i,3))-2);
    u(8) = U(3*(conect(i,3))-1);
    u(9) = U(3*(conect(i,3))-0);

%Area do elemento:
  Ael(i) = (coordel(i,2)*coordel(i,6) + coordel(i,1)*coordel(i,5) + coordel(i,3)*coordel(i,4)...
    - coordel(i,2)*coordel(i,4) - coordel(i,3)*coordel(i,5) - coordel(i,1)*coordel(i,6))/2;

%coef a, b, c:
    a(1) = coordel(i,2)*coordel(i,6) - coordel(i,3)*coordel(i,5);
    b(1) = coordel(i,5) - coordel(i,6);
    c(1) = coordel(i,3) - coordel(i,2);

    a(2) = coordel(i,3)*coordel(i,4) - coordel(i,1)*coordel(i,3);
    b(2) = coordel(i,6) - coordel(i,4);
    c(2) = coordel(i,1) - coordel(i,3);

    a(3) = coordel(i,1)*coordel(i,5) - coordel(i,2)*coordel(i,4);
    b(3) = coordel(i,4) - coordel(i,5);
    c(3) = coordel(i,2) - coordel(i,1);

%Fun  es phi:

```

```
phi1 = (xi1 - (xi1*xi2^2 - xi1^2*xi2)+(xi3*xi1^2 - xi3^2*xi1));
phi2 = (1/2)*(-b(3)*xi1*xi2 + b(2)*xi3*xi1 + b(3)*(xi1*xi2^2 - xi1
    ^2*xi2) + b(2)*(xi3*xi1^2 - xi3^2*xi1));
phi3 = (1/2)*(c(3)*xi1*xi2 - c(2)*xi3*xi1 - c(3)*(xi1*xi2^2 - xi1
    ^2*xi2) - c(2)*(xi3*xi1^2 - xi3^2*xi1));
phi4 = xi2 + (xi1*xi2^2 - xi1^2*xi2) - (xi2*xi3^2 - xi2^2*xi3);
phi5 = (1/2)*(b(3)*xi1*xi2 - b(1)*xi2*xi3 + b(3)*(xi1*xi2^2 - xi1
    ^2*xi2) + b(1)*(xi2*xi3^2 - xi2^2*xi3));
phi6 = (1/2)*(-c(3)*xi1*xi2 + c(1)*xi2*xi3 - c(3)*(xi1*xi2^2 - xi1
    ^2*xi2) - c(1)*(xi2*xi3^2 - xi2^2*xi3));
phi7 = xi3 + (xi2*xi3^2 - xi2^2*xi3) - (xi3*xi1^2 - xi3^2*xi1);
phi8 = (1/2)*(b(1)*xi2*xi3 - b(2)*xi3*xi1 + b(1)*(xi2*xi3^2 - xi2
    ^2*xi3) + b(2)*(xi3*xi1^2 - xi3^2*xi1));
phi9 = (1/2)*(-c(1)*xi2*xi3 + c(2)*xi3*xi1 - c(1)*(xi2*xi3^2 - xi2
    ^2*xi3) - c(2)*(xi3*xi1^2 - xi3^2*xi1));

%Aplica dados na matriz funcoes de forma:

phi(1,1) = phi1;
phi(1,2) = phi2;
phi(1,3) = phi3;
phi(1,4) = phi4;
phi(1,5) = phi5;
phi(1,6) = phi6;
phi(1,7) = phi7;
phi(1,8) = phi8;
phi(1,9) = phi9;


%Primeira linha da matriz de derivadas:
    for j=1:9
  d1(j) = (b(1)^2*(diff(diff(phi(1,j),xi1),xi1)) + b(2)^2*(diff(
    diff(phi(1,j),xi2),xi2)) +...
  b(3)^2*(diff(diff(phi(1,j),xi3),xi3)) + 2*b(1)*b(2)*(diff(diff(
```

```
   phi(1,j),xi1),xi2))+...
  2*b(1)*b(3)*(diff(diff(phi(1,j),xi3),xi1)) + 2*b(2)*b(3)*(diff(
   diff(phi(1,j),xi3),xi2)));
    end

%Segunda linha da matriz de derivadas:
    for j=1:9
   d2(j) = (c(1)^2*(diff(diff(phi(1,j),xi1),xi1)) + c(2)^2*(diff(
   diff(phi(1,j),xi2),xi2)) +...
   c(3)^2*(diff(diff(phi(1,j),xi3),xi3)) + 2*c(1)*c(2)*(diff(diff(
   phi(1,j),xi1),xi2))+...
   2*c(1)*c(3)*(diff(diff(phi(1,j),xi3),xi1)) + 2*c(2)*c(3)*(diff(
   diff(phi(1,j),xi3),xi2)));
    end


%Terceira linha da matriz de derivadas:
      for j=1:9
   d3(j) = (2*(b(1)*c(1)*(diff(diff(phi(1,j),xi1),xi1)) + b(2)*c(2)
   *(diff(diff(phi(1,j),xi2),xi2)) +...
       b(3)*c(3)*(diff(diff(phi(1,j),xi3),xi3)) + (b(1)*c(2) + b(2)
   *c(1))*(diff(diff(phi(1,j),xi2),xi1)) + ...
       (b(1)*c(3) + b(3)*c(1))*(diff(diff(phi(1,j),xi3),xi1)) + (b
   (2)*c(3) + b(3)*c(2))*(diff(diff(phi(1,j),xi3),xi2))));
      end


%Multiplica as derivadas pelo vetor de deslocamentos:
del2w1 = -(1/(4*Ael(i)^2))*d1*u;

del2w2 = -(1/(4*Ael(i)^2))*d2*u;

del2w3 = -(1/(4*Ael(i)^2))*d3*u;

```

```

%Calcula os momentos:
D = (E*t^3)/(12*(1-pss^2));
Mx = -D*(del2w1 + (pss*del2w2));
My = -D*(del2w2 + (pss*del2w1));
Mxy = -(D/2)*(1+pss)*(del2w3);

%Para o primeiro no:

% xi1 = 1;
% xi2 = 0;
% xi3 = 0;

Mx1 = -D*(del2w1 + pss*del2w2);
My1 = -D*(del2w2 + pss*del2w1);
Mxy1 = -(D/2)*(1+pss)*(del2w3);

Mx1 = subs(Mx1,1); %substitui a variavel por um valor numerico
Mx1 = subs(Mx1,0);
Mx1 = subs(Mx1,0);
My1 = subs(My1,1);
My1 = subs(My1,0);
My1 = subs(My1,0);
Mxy1 = subs(Mxy1,1);
Mxy1 = subs(Mxy1,0);
Mxy1 = subs(Mxy1,0);


Mx1 = double(Mx1); %Transforma o valor obtido em decimal
My1 = double(My1);
Mxy1 = double(Mxy1);

%Para o segundo no:

```

```
% xi1 = 0;
% xi2 = 1;
% xi3 = 0;

Mx2 = -D*(del2w1 + pss*del2w2);
My2 = -D*(del2w2 + pss*del2w1);
Mxy2 = -(D/2)*(1+pss)*(del2w3);

Mx2 = subs(Mx2,0);
Mx2 = subs(Mx2,1);
Mx2 = subs(Mx2,0);
My2 = subs(My2,0);
My2 = subs(My2,1);
My2 = subs(My2,0);
Mxy2 = subs(Mxy2,0);
Mxy2 = subs(Mxy2,1);
Mxy2 = subs(Mxy2,0);

Mx2 = double(Mx2);
My2 = double(My2);
Mxy2 = double(Mxy2);

%Para o terceiro no:

% xi1 = 0;
% xi2 = 0;
% xi3 = 1;

Mx3 = -D*(del2w1 + pss*del2w2);
My3 = -D*(del2w2 + pss*del2w1);
Mxy3 = -(D/2)*(1+pss)*(del2w3);

Mx3 = subs(Mx3,0);
Mx3 = subs(Mx3,0);
```

```
Mx3 = subs(Mx3,1);
My3 = subs(My3,0);
My3 = subs(My3,0);
My3 = subs(My3,1);
Mxy3 = subs(Mxy3,0);
Mxy3 = subs(Mxy3,0);
Mxy3 = subs(Mxy3,1);

Mx3 = double(Mx3);
My3 = double(My3);
Mxy3 = double(Mxy3);

%Aloca os momentos obtidos em vetores mom1=Mx, mom2=My, mom3=Mxy

mom1 = [Mx1; Mx2; Mx3];
mom2 = [My1; My2; My3];
mom3 = [Mxy1; Mxy2; Mxy3];

%Soma os valores de momento obtidos para cada no:
for j=1:3
 mom(3*conect(i,j)-2) = mom(3*conect(i,j)-2) + mom1(j);
 mom(3*conect(i,j)-1) = mom(3*conect(i,j)-1) + mom2(j);
 mom(3*conect(i,j)-0) = mom(3*conect(i,j)-0) + mom3(j);
end
mom;
end

%Cria um vetor para conter os momentos finais:
MOM = zeros(ngdltot,1);

%Faz a media dos momentos obtidos para nos pertencentes a mais de
   um
%elemento:
for i=1:nnostot;
```

```
    N = 0;
    for j=1:nel;

        if i == conect(j,1)|i == conect(j,2)|i == conect(j,3);
            N = N+1;
        end
    end
  %Aplica no vetor criado a media obtida:
  MOM(3*i-2)=mom(3*i-2)/N;
  MOM(3*i-1)=mom(3*i-1)/N;
  MOM(3*i-0)=mom(3*i-0)/N;
end


%CRIA A MATRIZ PARA PLOTAR OS MOMENTOS EM X:
k=1;
while k<=nnostot;
    for i=1:nnosy;
        for j=1:nnosx;
            MOMX(i,j)=MOM(3*k-2);
            k=k+1;
        end
    end
end

%CRIA A MATRIZ PARA PLOTAR OS MOMENTOS EM Y:
k=1;
while k<=nnostot;
    for i=1:nnosy;
        for j=1:nnosx;
            MOMY(i,j)=MOM(3*k-1);
            k=k+1;
        end
    end
```

```
end

%CRIA A MATRIZ PARA PLOTAR OS MOMENTOS XY:
k=1;
while k<=nnostot;
    for i=1:nnosy;
        for j=1:nnosx;
            MOMXY(i,j)=MOM(3*k-0);
            k=k+1;
        end
    end
end

%Cria vetores de dimensoes x e y:

   x = [0:dx:dimx];
   y = [0:dy:dimy];

%Plota os deslocamentos:
figure
[X,Y] = meshgrid(x,y);
 grafdesl=surf(X,Y,w,'FaceAlpha',1);
 colormap(jet);
 title('Placa no seu estado deformado.');
colorbar;

%Plota o Momento em X:
figure
[X,Y] = meshgrid(x,y);
grafmomx = contourf(X,Y,MOMX);
colormap(jet);
title('Distribui  o de Mx.');
colorbar;

```

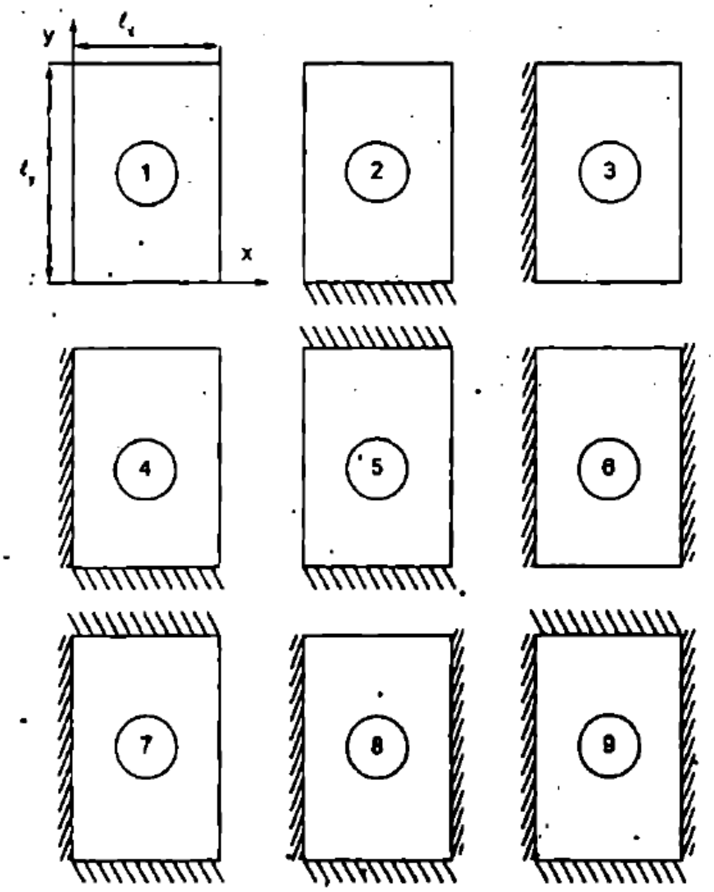
```
%Plota o Momento em X:
figure
[X,Y] = meshgrid(x,y);
grafmomy = contourf(X,Y,MOMY);
colormap(jet);
title('Distribui  o de My.');
colorbar;

%Plota o Momento em XY:
figure
[X,Y] = meshgrid(x,y);
grafmomxy = contourf(X,Y,MOMXY);
colormap(jet);
title('Distribui  o de Mxy.');
colorbar;

%FIM
```

ANEXO A - TABELAS DE BARES E CZÉRNY

Figura 36 – Casos de vinculação - tabela de Bares



Fonte: Carvalho e Figueiredo Filho (2014)

$$f = \frac{p\,\ell_x^4}{E\,h^3} \cdot \frac{\alpha}{100}$$

$$\lambda = \frac{\ell_y}{\ell_x}$$

$$m_x = \mu_x \cdot \frac{p\,\ell_x^2}{100}$$

$$m_y = \mu_y \cdot \frac{p\,\ell_x^2}{100}$$

Figura 37 – Coeficientes α para o cálculo de flechas - tabela de Bares adaptada para ν = 0,2

λ	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6	Caso 7	Caso 8	Caso 9
1,00	4,67	3,20	3,20	2,42	2,21	2,21	1,81	1,81	1,46
1,05	5,17	3,61	3,42	2,67	2,55	2,31	2,04	1,92	1,60
1,10	5,64	4,04	3,63	2,91	2,92	2,41	2,27	2,04	1,74
1,15	6,09	4,47	3,82	3,12	3,29	2,48	2,49	2,14	1,87
1,20	6,52	4,91	4,02	3,34	3,67	2,56	2,72	2,24	1,98
1,25	6,95	5,34	4,18	3,55	4,07	2,63	2,95	2,33	2,10
1,30	7,36	5,77	4,35	3,73	4,48	2,69	3,16	2,42	2,20
1,35	7,76	6,21	4,50	3,92	4,92	2,72	3,36	2,48	2,30
1,40	8,14	6,62	4,65	4,08	5,31	2,75	3,56	2,56	2,37
1,45	8,51	7,02	4,78	4,23	5,73	2,80	3,73	2,62	2,45
1,50	8,87	7,41	4,92	4,38	6,14	2,84	3,91	2,68	2,51
1,55	9,22	7,81	5,00	4,53	6,54	2,86	4,07	2,53	2,57
1,60	9,54	8,17	5,09	4,65	6,93	2,87	4,22	2,87	2,63
1,65	9,86	8,52	5,13	4,77	7,33	2,87	4,37	2,78	2,68
1,70	10,15	8,87	5,17	4,88	7,70	2,88	4,51	2,79	2,72
1,75	10,43	9,19	5,26	4,97	8,06	2,88	4,63	2,81	2,76
1,80	10,71	9,52	5,36	5,07	8,43	2,89	4,75	2,83	2,80
1,85	10,96	9,82	5,43	5,16	8,77	2,89	4,87	2,85	2,83
1,90	11,21	10,11	5,50	5,23	9,08	2,90	4,98	2,87	2,85
1,95	11,44	10,39	5,58	5,31	9,41	2,90	5,08	2,89	2,88
2,00	11,68	10,68	5,66	5,39	9,72	2,91	5,19	2,91	2,91
∞	15,35	15,35	6,38	6,38	15,35	3,07	6,38	3,07	3,07

Fonte: Carvalho e Figueiredo Filho (2014)

Figura 38 – Coeficientes μ_x e μ_y para o cálculo de momentos - tabela de Bares adaptada para ν = 0,2, casos 1, 2 e 3

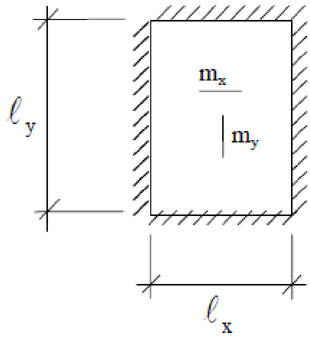
λ	Caso 1		Caso 2			Caso 3		
	μ_x	μ_y	μ_x	μ_y	μ'_y	μ_x	μ'_x	μ_y
1,00	4,41	4,41	3,07	3,94	8,52	3,94	8,52	3,07
1,05	4,80	4,45	3,42	3,78	8,79	4,19	8,91	2,84
1,10	5,18	4,49	3,77	3,90	9,18	4,43	9,30	2,76
1,15	5,56	4,49	4,14	3,97	9,53	4,64	9,63	2,68
1,20	5,90	4,48	4,51	4,05	9,88	4,85	9,95	2,59
1,25	6,27	4,45	4,88	4,10	10,16	5,03	10,22	2,51
1,30	6,60	4,42	5,25	4,15	10,41	5,20	10,48	2,42
1,35	6,93	4,37	5,60	4,18	10,64	5,36	10,71	2,34
1,40	7,25	4,33	5,95	4,21	10,86	5,51	10,92	2,25
1,45	7,55	4,30	6,27	4,19	11,05	5,64	11,10	2,19
1,50	7,86	4,25	6,60	4,18	11,23	5,77	11,27	2,12
1,55	8,12	4,20	6,90	4,17	11,39	5,87	11,42	2,04
1,60	8,34	3,14	7,21	4,14	11,55	5,98	11,55	1,95
1,65	8,62	4,07	7,42	4,12	11,67	6,07	11,67	1,87
1,70	8,86	4,00	7,62	4,09	11,79	6,16	11,80	1,79
1,75	9,06	3,96	7,66	4,05	11,88	6,24	11,92	1,74
1,80	9,27	3,91	7,69	3,99	11,96	6,31	12,04	1,68
1,85	9,45	3,83	8,22	3,97	12,03	6,38	12,14	1,64
1,90	9,63	3,75	8,74	3,94	12,14	6,43	12,24	1,59
1,95	9,77	3,71	8,97	3,88	12,17	6,47	12,29	1,54
2,00	10,00	3,64	9,18	3,80	12,20	6,51	12,34	1,48
∞	12,57	3,77	9,18	3,80	12,20	7,61	12,76	1,48

Fonte: Carvalho e Figueiredo Filho (2014)

Figura 39 – Coeficientes para o cálculo de momentos - tabela de Czérny adaptada para ν = 0,2

TABELA 1 - TIPO 1
Laje com as 4 bordas livremente apoiadas
(carga uniforme)

ℓ_y / ℓ_x	α_x	α_y	β_x	β_y	α_2
1,00	22,7	22,7			21,4
1,05	20,8	22,5			19,4
1,10	19,3	22,3			17,8
1,15	18,1	22,3			16,5
1,20	16,9	22,3			15,4
1,25	15,9	22,4			14,3
1,30	15,2	22,7			13,6
1,35	14,4	22,9			12,9
1,40	13,8	23,1			12,3
1,45	13,2	23,3			11,7
1,50	12,7	23,5			11,2
1,55	12,3	23,5			10,8
1,60	11,9	23,5			10,4
1,65	11,5	23,5			10,1
1,70	11,2	23,5			9,8
1,75	10,8	23,5			9,5
1,80	10,7	23,5			9,3
1,85	10,4	23,5			9,1
1,90	10,2	23,5			8,9
1,95	10,1	23,5			8,7
2,00	9,9	23,5			8,6
>2	8,0	23,5			6,7



$$m_x = \frac{p\ell_x^2}{\alpha_x}$$

$$m_y = \frac{p\ell_x^2}{\alpha_y}$$

$$w_{max} = \frac{p\ell_x^4}{Eh^3\alpha_2}$$

$$\nu = 0,2$$

Beton-Kalender (1976)

Fonte: Zenzen (2012)