

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RICARDO AUGUSTO ANDREOTTI

**APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS PARA A  
OBTENÇÃO DA DEFLEXÃO DE PLACAS DELGADAS  
ORTOTRÓPICAS BI-APOIADAS**

CAMPO MOURÃO

2018

RICARDO AUGUSTO ANDREOTTI

**APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS PARA A  
OBTENÇÃO DA DEFLEXÃO DE PLACAS DELGADAS  
ORTOTRÓPICAS BI-APOIADAS**

Trabalho de Conclusão de Curso de Graduação apresentado à Disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior em Engenharia Civil do Departamento Acadêmico de Construção Civil – DACOC – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de bacharel em engenharia civil.

Orientador: Prof. Dr. Adilandri Mércio  
Lobeiro

Coorientador: Prof. Dr. Jorge Luis Nunes de  
Góes

CAMPO MOURÃO

2018



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Câmpus Campo Mourão  
Diretoria de Graduação e Educação Profissional  
Departamento Acadêmico de Construção Civil  
Coordenação de Engenharia Civil



---

## TERMO DE APROVAÇÃO

Trabalho de Conclusão de Curso

**APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS PARA A OBTENÇÃO DA  
DEFLEXÃO DE PLACAS DELGADAS ORTOTRÓPICAS BI-APOIADAS.**

por

**Ricardo Augusto Andreotti**

Este Trabalho de Conclusão de Curso foi apresentado às 16:00h do dia 15 de junho de 2018 como requisito parcial para a obtenção do título de ENGENHEIRO CIVIL, pela Universidade Tecnológica Federal do Paraná. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

**Prof<sup>a</sup>. Dr<sup>a</sup>. Lilian Caroline Xavier**

**Candido**

(UTFPR)

**Prof. Dr. Jeferson Rafael Bueno**

(UTFPR)

**Prof. Dr. Jorge Luis Nunes De Goes**

(UTFPR)

**Co-orientador**

**Prof. Dr. Adilandri Mércio Lobeiro**

(UTFPR)

**Orientador**

Responsável pelo TCC: **Prof. Me. Valdomiro Lubachevski Kurta**

Coordenador do Curso de Engenharia Civil:

**Prof. Dr. Ronaldo Rigobello**

*A Folha de Aprovação assinada encontra-se na Coordenação do Curso.*

## Agradecimentos

A Deus, por sempre estar comigo nas horas mais complicadas, foi meu grande suporte e razão da perseverança.

Aos meus pais, Vilson e Alessandra, principalmente pela paciência, perseverança, incentivo que eles me ofereceram ao longo desses maravilhosos anos de minha graduação. Sem eles, minha jornada teria sido muito mais complicada. Minha eterna gratidão por tudo que fizeram por mim.

Ao meu querido irmão, Guilherme, agradeço e me desculpo por não estar contigo em todos os momentos de sua vida. Muito obrigado por ser essa pessoa maravilhosa, me considerar um grande amigo e sempre me alegrar em momentos complicados.

Ao Prof. Dr. Adilandri Mércio Lobeiro, pela excelente orientação e amizade. Me acompanhou e desafiou a novos pensamentos e atitudes desde que entrei em seu projeto de extensão. Me ajudou a crescer como estudante, mas principalmente como ser humano.

Ao Prof. Dr. Jorge Luis Nunes de Góes, pela excelente coorientação e suas ótimas ideias, que tornaram este trabalho possível.

Ao corpo docente da UTFPR, por dar condições e suporte ao meu estudo.

A minha grande amiga, Jessica, muito obrigado por ser essa pessoa maravilhosa, mesmo longe de ti ao longo desses anos, sinto que você sempre esteve ao meu lado em todos os momentos.

Aos meus grandes amigos, sentirei muita saudades das nossas risadas, das conversas jogadas fora, dentro e principalmente fora das salas de aula. Vocês tornaram este período da minha vida o melhor até agora. Vocês são figuras importantíssimas para mim. Eu só tenho que agradecer à todos vocês. Muito obrigado.

## Resumo

ANDREOTTI, R. A. **Aplicação do método das diferenças finitas para a obtenção da deformação de placas delgadas ortotrópicas bi-apoiadas**. 188 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Civil, Universidade Tecnológica Federal do Paraná, Campo Mourão, 2018.

Neste trabalho é apresentado a solução numérica, via Método das Diferenças Finitas, da Equação Diferencial Parcial (EDP) de quarta ordem que controla a deflexão de placas delgadas ortotrópicas bi-apoiadas, quando submetidas à um carregamento uniformemente distribuído. Esse método transforma a EDP em um sistema de equações lineares. Para obter a solução desse sistema, implementou-se o método de solução iterativo de Gauss-Seidel. Por fim, para facilitar a leitura e interpretação dos resultados, desenvolveu-se um programa compilado em linguagem *MatLab*<sup>®</sup>, utilizando a ferramenta App Designer. Os resultados obtidos foram comparados com o *software* comercial Robot, que utiliza o Método dos Elementos Finitos. Para fins de comparação, utilizou-se o erro relativo entre os dois resultados, que foram relativamente baixos, de ordem 0,59% e 0,84%, para relações geométricas entre o Comprimento ( $L$ ) e largura ( $b$ ), sendo  $L/b = 1$  e  $L/b = 2$ , respectivamente.

**Palavras-chaves:** Equação Diferencial Parcial, Método das Diferenças Finitas, Placas ortotrópicas, Gauss-Seidel.

## Abstract

ANDREOTTI, R. A. **Application of the finite difference method to obtain the bi-supported orthotropic plate's deflection.** 188 p. Trabalho de Conclusão de Curso (Graduação) – Engenharia Civil, Universidade Tecnológica Federal do Paraná, Campo Mourão, 2018.

In this work we present the numerical solution, by Finite Differences Method, of the fourth-order Partial Differential Equation that controls the bi-supported orthotropic thin plates' deflection when subjected to an uniformly distributed load. This method transforms the partial differential equation into a system of linear equations. In order to obtain the solution of that system, the Gauss-Seidel iterative solution method was implemented. Finally, to facilitate the interpretation and reading of the results, a program compiled in *MatLab*<sup>®</sup>'s language was developed, using its tool App Designer. The obtained results were compared with a commercial software, based on Finite Element Method. For comparison purposes, a geometric relations between length ( $L$ ) and width ( $b$ ) was used to compared both results, being  $L/b = 1$  and  $L/b = 2$ , with error order of 0.59% and 0.84%, respectively. The relative error between the two results were relatively low.

**Keywords:** Finite Differences Method, Finite Element Method, orthotropic plates, Gauss-Seidel.

## Lista de figuras

Figura 1 – Laje maciça bi-apoiada. . . . .	21
Figura 2 – Vigotas pré-moldadas. . . . .	21
Figura 3 – Blocos de enchimento. . . . .	22
Figura 4 – Tabuleiros de concreto. . . . .	22
Figura 5 – Planta, seção transversal e elevação de ponte protendida. . . . .	23
Figura 6 – Ponte protendida formada por vigas de seção T. . . . .	24
Figura 7 – Discretização de um domínio contínuo. . . . .	26
Figura 8 – Posicionamento dos eixos principais. . . . .	29
Figura 9 – Deformação elemento infinitesimal. . . . .	29
Figura 10 – Seção transversal da placa deformada. . . . .	32
Figura 11 – Distribuição de tensão normal no plano transversal. . . . .	34
Figura 12 – Plano transversal na direção $x$ . . . . .	35
Figura 13 – Equilíbrio infinitesimal. . . . .	37
Figura 14 – Domínio discretizado. . . . .	41
Figura 15 – Molécula do MDF para a EDP dada pela equação (54) . . . . .	49
Figura 16 – Posicionamento da placa no plano cartesiano. . . . .	49
Figura 17 – Quantidade de pontos semelhantes . . . . .	50
Figura 18 – Pontos semelhantes em uma placa bi-apoiada . . . . .	53
Figura 19 – Molécula posicionada no ponto $i = 2$ e $j = 1$ . . . . .	54
Figura 20 – Molécula posicionada no ponto $i = 3$ e $j = 1$ . . . . .	55
Figura 21 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = 1$ . . . . .	56
Figura 22 – Molécula posicionada no ponto $i = M - 1$ e $j = 1$ . . . . .	58
Figura 23 – Molécula posicionada no ponto $i = M$ e $j = 1$ . . . . .	59
Figura 24 – Molécula posicionada no ponto $i = 2$ e $j = 2$ . . . . .	60
Figura 25 – Molécula posicionada no ponto $i = 3$ e $j = 2$ . . . . .	61
Figura 26 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = 2$ . . . . .	63
Figura 27 – Molécula posicionada no ponto $i = M - 1$ e $j = 2$ . . . . .	64
Figura 28 – Molécula posicionada no ponto $i = M$ e $j = 2$ . . . . .	65
Figura 29 – Molécula posicionada no ponto $i = 2$ e $j = 3$ . . . . .	66
Figura 30 – Molécula posicionada no ponto $i = 3$ e $j = 3$ . . . . .	68

Figura 31 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = 3$ . . . . .	69
Figura 32 – Molécula posicionada no ponto $i = M - 1$ e $j = 3$ . . . . .	70
Figura 33 – Molécula posicionada no ponto $i = M$ e $j = 3$ . . . . .	71
Figura 34 – Molécula posicionada no ponto $i = 2$ e $j = 4, \dots, M - 2$ . . . . .	72
Figura 35 – Molécula posicionada no ponto $i = 3$ e $j = 4, \dots, M - 2$ . . . . .	73
Figura 36 – Molécula posicionada no ponto $i = M - 1$ e $j = 4, \dots, M - 2$ . . . . .	75
Figura 37 – Molécula posicionada no ponto $i = M$ e $j = 4, \dots, M - 2$ . . . . .	76
Figura 38 – Molécula posicionada no ponto $i = 2$ e $j = N - 1$ . . . . .	77
Figura 39 – Molécula posicionada no ponto $i = 3$ e $j = N - 1$ . . . . .	78
Figura 40 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = N - 1$ . . . . .	79
Figura 41 – Molécula posicionada no ponto $i = M - 1$ e $j = N - 1$ . . . . .	81
Figura 42 – Molécula posicionada no ponto $i = M$ e $j = N - 1$ . . . . .	82
Figura 43 – Molécula posicionada no ponto $i = 2$ e $j = N$ . . . . .	83
Figura 44 – Molécula posicionada no ponto $i = 3$ e $j = N$ . . . . .	84
Figura 45 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = N$ . . . . .	85
Figura 46 – Molécula posicionada no ponto $i = M - 1$ e $j = N$ . . . . .	87
Figura 47 – Molécula posicionada no ponto $i = M$ e $j = N$ . . . . .	88
Figura 48 – Molécula posicionada no ponto $i = 2$ e $j = N + 1$ . . . . .	89
Figura 49 – Molécula posicionada no ponto $i = 3$ e $j = N + 1$ . . . . .	90
Figura 50 – Molécula posicionada no ponto $i = 4, \dots, M - 2$ e $j = N + 1$ . . . . .	91
Figura 51 – Molécula posicionada no ponto $i = M - 1$ e $j = N + 1$ . . . . .	93
Figura 52 – Molécula posicionada no ponto $i = M$ e $j = N + 1$ . . . . .	94
Figura 53 – Página inicial . . . . .	96
Figura 54 – Fluxograma. . . . .	97
Figura 55 – Página das propriedades geométricas. . . . .	98
Figura 56 – Página das propriedades do material isotrópico. . . . .	99
Figura 57 – Página das propriedades do material ortotrópico. . . . .	100
Figura 58 – Página de definições das vinculações. . . . .	101
Figura 59 – Página de definições das vinculações com apresentação do erro. . . . .	102
Figura 60 – Página de definições das vinculações com apresentação da legenda. . . . .	103
Figura 61 – Página dos parâmetros. . . . .	104
Figura 62 – Página para introdução dos carregamentos. . . . .	105
Figura 63 – Página para verificação numérica. . . . .	107



Figura 64 – Página para apresentação do resultado gráfico. . . . .	108
Figura 65 – Página para apresentação do resultado para análise. . . . .	109
Figura 66 – Erro absoluto para uma placa bi-apoiada de $L/b = 1$ . . . . .	112
Figura 67 – Comparação entre MDF e MEF. . . . .	114
Figura 68 – Representação gráfica da solução por MDF, com $M = N = 60$ . . . . .	114
Figura 69 – Representação gráfica dada pelo <i>software Robot 2018</i> . . . . .	115
Figura 70 – Erro absoluto para uma placa bi-apoiada de $L/b = 2$ . . . . .	116
Figura 71 – Comparação entre MDF e MEF. . . . .	118
Figura 72 – Representação gráfica da solução por MDF, com $M = N = 40$ . . . . .	118
Figura 73 – Representação gráfica dada pelo <i>software Robot 2018</i> . . . . .	119

## Lista de Algoritmos

Algoritmo 1 – Definição da matriz dos coeficientes $A$ . . . . .	124
Algoritmo 2 – Critério de Sassenfeld. . . . .	158
Algoritmo 3 – Método de Gauss-Seidel. . . . .	160
Algoritmo 4 – Auxiliar do Algoritmo 1: Escrita_matriz. . . . .	173
Algoritmo 5 – Auxiliar do Algoritmo 1: Escrita_matriz_i. . . . .	174
Algoritmo 6 – Auxiliar do Algoritmo 1: Escrita_matriz_i_N. . . . .	176
Algoritmo 7 – Auxiliar do Algoritmo 1: Escrita_matriz_j. . . . .	178
Algoritmo 8 – Auxiliar do Algoritmo 1: Escrita_matriz_M. . . . .	180
Algoritmo 9 – Auxiliar do Algoritmo 1: Escrita_matriz_M_j. . . . .	182
Algoritmo 10 – Auxiliar do Algoritmo 1: Escrita_matriz_M_N. . . . .	184
Algoritmo 11 – Auxiliar do Algoritmo 1: Escrita_matriz_N. . . . .	186

## Lista de tabelas

Tabela 1 – Verificação da convergência . . . . .	111
Tabela 2 – Comparação entre o MDF e MEF. . . . .	113
Tabela 3 – Verificação da convergência . . . . .	115
Tabela 4 – Comparação entre o MDF e MEF . . . . .	117

## Lista de abreviaturas e siglas

ED	Equação Diferencial
EDO	Equação Diferencial Ordinária
EDP	Equação Diferencial Parcial
ELS	Estados Limites de Serviço
ELU	Estados Limites Últimos
EPS	Poliestireno Expandido
MDF	Método das Diferenças Finitas
MEF	Método dos Elementos Finitos
WVU	West Virginia University

## Lista de símbolos

### Alfabeto grego minúsculo

$\epsilon$	Deformação linear
$\lambda$	Deformação angular
$\nu$	Coefficiente de poisson
$\rho$	Raio espectral
$\sigma$	Tensão normal ao plano
$\tau$	Tensão de cisalhamento
$\theta$	Giro da seção transversal

### Alfabeto romano minúsculo

<b><math>b</math></b>	Matriz dos termos independentes
$a$	Coordenada inicial do intervalo das abscissas
$b$	Coordenada final do intervalo das abscissas
$c$	Coordenada inicial do intervalo das ordenadas
$d$	Coordenada final do intervalo das ordenadas
$ds$	Comprimento inicial infinitesimal
$ds'$	Comprimento final infinitesimal
$h$	Espaçamento nas abscissas
$k$	Espaçamento nas ordenadas
$p$	Carregamento
$r$	Deformação na direção do eixo $x$
$s$	Deformação na direção do eixo $y$
$t$	Espessura da placa

$u$	Deformação na direção do eixo $z$
$\mathbf{x}$	Vetor solução
$z$	Superfície média da placa em movimento em um plano $X, Y, Z$

### **Alfabeto romano maiúsculo**

$A$	Matriz dos coeficientes
$I$	Matriz identidade
$E$	Módulo de Young
$G$	Módulo de elasticidade ao cisalhamento
$L$	Matriz triangular estritamente inferior
$M$	Momento fletor
$M$	Quantidade de subintervalos nas abscissas
$N$	Quantidade de subintervalos nas ordenadas
$Q$	Força cortante
$T$	Matriz iteração
$U$	Matriz triangular estritamente superior
$D$	Rigidez a flexão da placa

## Sumário

<b>1</b>	<b>Introdução</b> . . . . .	16
1.1	<i>Organização do trabalho</i> . . . . .	17
<b>2</b>	<b>Objetivos</b> . . . . .	18
2.1	<i>Objetivo geral</i> . . . . .	18
2.2	<i>Objetivos específicos</i> . . . . .	18
<b>3</b>	<b>Justificativa</b> . . . . .	19
<b>4</b>	<b>Revisão bibliográfica</b> . . . . .	20
4.1	<i>Considerações sobre a revisão bibliográfica</i> . . . . .	27
4.2	<i>Teoria das placas delgadas ortotrópicas equivalentes</i> . . . . .	28
4.2.1	Estados plano de deformações . . . . .	28
4.2.2	Estados plano de tensões . . . . .	32
4.2.3	Equilíbrio de uma seção infinitesimal . . . . .	36
4.2.4	Condições de contorno . . . . .	39
4.3	<i>Métodos das diferenças finitas</i> . . . . .	40
4.3.1	Expansões em série de Taylor . . . . .	41
4.4	<i>Métodos iterativos de solução de sistemas de equações lineares</i> . . . . .	43
4.4.1	Método de Jacobi . . . . .	44
4.4.2	Método de Gauss-Seidel . . . . .	44
4.4.3	Critérios de convergência . . . . .	45
<b>5</b>	<b>Metodologia</b> . . . . .	47
5.1	<i>Considerando <math>i = 2</math> e <math>j = 1</math></i> . . . . .	53
5.2	<i>Considerando <math>i = 3</math> e <math>j = 1</math></i> . . . . .	55
5.3	<i>Considerando <math>i = 4, \dots, M - 2</math> e <math>j = 1</math></i> . . . . .	56
5.4	<i>Considerando <math>i = M - 1</math> e <math>j = 1</math></i> . . . . .	57
5.5	<i>Considerando <math>i = M</math> e <math>j = 1</math></i> . . . . .	59
5.6	<i>Considerando <math>i = 2</math> e <math>j = 2</math></i> . . . . .	60
5.7	<i>Considerando <math>i = 3</math> e <math>j = 2</math></i> . . . . .	61
5.8	<i>Considerando <math>i = 4, \dots, M - 2</math> e <math>j = 2</math></i> . . . . .	62

5.9	Considerando $i = M - 1$ e $j = 2$ . . . . .	64
5.10	Considerando $i = M$ e $j = 2$ . . . . .	65
5.11	Considerando $i = 2$ e $j = 3$ . . . . .	66
5.12	Considerando $i = 3$ e $j = 3$ . . . . .	67
5.13	Considerando $i = 4, \dots, M - 2$ e $j = 3$ . . . . .	68
5.14	Considerando $i = M - 1$ e $j = 3$ . . . . .	70
5.15	Considerando $i = M$ e $j = 3$ . . . . .	71
5.16	Considerando $i = 2$ e $j = 4, \dots, M - 2$ . . . . .	72
5.17	Considerando $i = 3$ e $j = 4, \dots, M - 2$ . . . . .	73
5.18	Considerando $i = M - 1$ e $j = 4, \dots, M - 2$ . . . . .	74
5.19	Considerando $i = M$ e $j = 4, \dots, M - 2$ . . . . .	76
5.20	Considerando $i = 2$ e $j = N - 1$ . . . . .	77
5.21	Considerando $i = 3$ e $j = N - 1$ . . . . .	78
5.22	Considerando $i = 4, \dots, M - 2$ e $j = N - 1$ . . . . .	79
5.23	Considerando $i = M - 1$ e $j = N - 1$ . . . . .	80
5.24	Considerando $i = M$ e $j = N - 1$ . . . . .	82
5.25	Considerando $i = 2$ e $j = N$ . . . . .	83
5.26	Considerando $i = 3$ e $j = N$ . . . . .	84
5.27	Considerando $i = 4, \dots, M - 2$ e $j = N$ . . . . .	85
5.28	Considerando $i = M - 1$ e $j = N$ . . . . .	86
5.29	Considerando $i = M$ e $j = N$ . . . . .	88
5.30	Considerando $i = 2$ e $j = N + 1$ . . . . .	89
5.31	Considerando $i = 3$ e $j = N + 1$ . . . . .	90
5.32	Considerando $i = 4, \dots, M - 2$ e $j = N + 1$ . . . . .	91
5.33	Considerando $i = M - 1$ e $j = N + 1$ . . . . .	92
5.34	Considerando $i = M$ e $j = N + 1$ . . . . .	94
5.35	Interface de cálculo . . . . .	95
<b>6</b>	<b>Verificação da solução numérica . . . . .</b>	<b>110</b>
6.1	Placa bi-apoiada ortotrópica de relação $L/b = 1$ . . . . .	110
6.2	Placa bi-apoiada ortotrópica de relação $L/b = 2$ . . . . .	115
<b>7</b>	<b>Considerações finais e conclusão . . . . .</b>	<b>120</b>



Referências . . . . .	121
-----------------------	-----

**APÊNDICES** **123**

Apêndice A – Algoritmos . . . . .	124
-----------------------------------	-----

Apêndice B – Trabalhos realizados . . . . .	188
---	-----

## 1 Introdução

Há uma grande variedade de fenômenos, que ocorrem nas mais diversas áreas, que podem ser descritos por Equações Diferenciais (ED). De acordo com Zill e Cullen (2001), uma ED pode ser classificada em uma Equação Diferencial Ordinária (EDO), quando possui derivadas de apenas uma variável independente, em uma Equação Diferencial Parcial (EDP), quando possui derivadas parciais de diversas variáveis independentes.

Quase todos os fenômenos físicos podem ser descritos tanto por EDOs tanto quanto por EDPs. Um desses fenômenos é a deflexão de placas, em que é descrito por uma EDP. Esta equação dita as deformações causadas pela aplicação de um carregamento transversal no plano médio dessa classe de elementos estruturais.

Deflexão de placas é um assunto de grande importância para a engenharia civil, pois é comum a existência de elementos estruturais considerados elementos de superfície, ou seja, em que a espessura é pequena ao comparada às outras dimensões, como as lajes de edifícios residenciais e comerciais, bem como os tabuleiros de pontes e viadutos.

Para que haja o dimensionamento de um elemento estrutural, devem ser verificados, segundo Souza (2017), os Estados Limites Últimos (ELU) e os Estados Limites de Serviço (ELS). Os ELU estão relacionados com o colapso total ou parcial da estrutura, associados com a perda de instabilidade, resistência e equilíbrio, que podem comprometer a segurança dos usuários. Os ELS, por sua vez, relacionam-se com as deficiências no desempenho para as condições de utilização, como, por exemplo, deflexões elevadas e vibrações excessivas, que causam problemas de fissurações nos elementos de vedação e desconforto aos ocupantes.

Obter a solução analítica de uma EDP é de elevada complexidade e nem sempre é possível, pois esse tipo de equação depende de uma grande quantidade de variáveis independentes. Por tal motivo, há o emprego de métodos numéricos, que fornecem uma solução aproximada com precisão satisfatória.

O método numérico utilizado neste trabalho é o Método das Diferenças Finitas (MDF) que consiste, basicamente, em transformar o domínio em um conjunto de pontos através da discretização e construir aproximações para as derivadas da ED por meio da expansão em série de Taylor. As aproximações das derivadas são substituídas na ED, transformando-a em um sistema de equações lineares. Dessa forma, é possível aplicar

métodos computacionais para obter a solução dos sistemas de equações lineares, que resultarão na aproximação da solução da ED.

Por fim, escolheu-se o *MatLab*<sup>®</sup> como uma ferramenta para a obtenção da solução numérica, pelas suas facilidades em implementação de funções matemáticas que foram bastante utilizadas.

### 1.1 Organização do trabalho

Este trabalho está organizada em mais dez capítulos mais um apêndice.

- No Capítulo 2 há o objetivo geral deste trabalho assim como os objetivos específicos;
- No Capítulo 3 há a justificativa deste trabalho;
- No Capítulo 4 há o levantamento bibliográfico, assim como as pesquisas que se desenvolveram sobre o tema apresentado neste trabalho;
- No Capítulo 4.2 há a descrição detalhada da teoria de placas delgadas ortotrópicas equivalentes;
- No Capítulo 4.3 descreve-se detalhadamente o MDF;
- No Capítulo 4.4 descreve-se detalhadamente os métodos iterativos de solução de sistemas de equações lineares;
- No Capítulo 5 aplica-se o MDF na EDP que descreve a deflexão de placas delgadas ortotrópicas equivalentes;
- No Capítulo 5.35, há a apresentação da interface de usuário desenvolvida no *software* MatLab;
- No Capítulo 6 é apresentado alguns exemplos numéricos para comprovar o método aqui empregado;
- No Apêndice B há alguns trabalhos publicados sobre o tema aqui apresentado.

## 2 Objetivos

### 2.1 *Objetivo geral*

Desenvolvimento de um *software* computacional para análise da deflexão de placas delgadas ortotrópicas bi-apoiadas via MDF.

### 2.2 *Objetivos específicos*

- Desenvolvimento teórico e numérico para um modelo matemático fundamentado no MDF no estudo de placas delgadas ortotrópicas bi-apoiadas;
- Obter as aproximações das derivadas via série de Taylor;
- Determinar as condições de contorno do problema avaliado;
- Aplicar métodos numéricos para a solução de sistemas lineares;
- Implementar uma rotina de cálculo no *software Matlab*<sup>®</sup> para análise linear das placas com carregamentos e condições de vinculações distintas;
- Processar alguns exemplos de placas bi-apoiadas na interface desenvolvida;
- Verificar a veracidade dos resultados obtidos e compara-los com um *software* comercial que utilizado o Método dos Elementos Finitos (MEF) para obtenção da deformação.

### 3 Justificativa

A justificativa deste trabalho se faz diante da necessidade de estudar placas e suas deflexões não através dos métodos tradicionais provenientes das tabelas de Czérny, Barnes e Timoshenko e sim por meio de métodos numéricos. Os métodos numéricos proporcionam obter as soluções de casos mais específicos. Este estudo é importante na engenharia civil, pelo fato de que esses elementos estruturais possuem a função de receber os carregamentos e distribuí-los para as vigas e pilares.

A deflexão máxima de uma placa é uma das questões a serem verificados quanto aos ELS. Galambos e Ellingwood (1986) informam que as deflexões estão associadas com danificações de elementos não estruturais, tais como os elementos de vedação, janelas e portas, prejudicando sua função pré-definida, assim como o desconforto, tanto visual e de utilização, em parte dos usuários.

Como a deflexão de uma placa pode ser calculada matematicamente, via solução de uma EDP, e conforme Cuminato e Junior (2011) informam que na maioria dos casos das EDPs a solução analítica não está disponível, e é essa a situação da EDP que controla a deflexão de placas, os autores afirmam que a aplicabilidade dos métodos numéricos é de essencial importância pois trazem uma representação da solução em um domínio discreto de algo que é inicialmente dado como contínuo. Transformar algo contínuo em discreto, faz com que o uso de computadores para a obtenção dessas soluções se torne possível.

## 4 Revisão bibliográfica

Há uma grande aplicabilidade de elementos de placas na engenharia civil. Como exemplo, pode-se citar as lajes de edifícios residenciais, comerciais e os tabuleiros de pontes e viadutos. Esses elementos apresentam geometrias e propriedades complexas. Cusens e Pama (1975) informam que esses elementos apresentam materiais com propriedades elásticas e/ou geométricas diferentes em direções ortogonais. Essa característica dá-se o nome de ortotropia.

Um exemplo de material ortotrópico é a madeira, pois devido as condições de sua formação, as retrações, inchamentos, deformações, que acontecem com diferentes intensidades em direções diversas, fazem com que apresentem propriedades elásticas diferentes em planos ortogonais distintos.

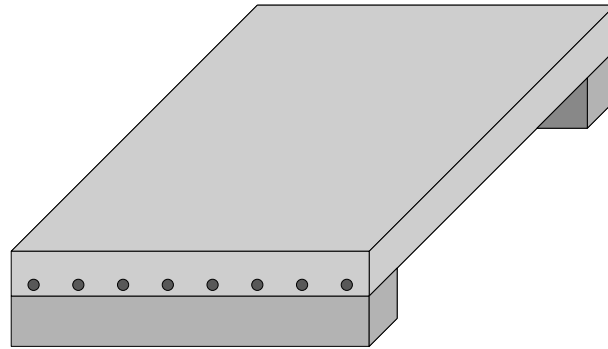
Outro exemplo é o concreto pois, conforme Bono (2008), é um material formado pela mistura de agregado, cimento e areia, apresentando um comportamento variado. Já no início, mesmo antes da aplicação de quaisquer carregamentos, o concreto apresenta microfissuras que vão aumentando em tamanho e quantidade, contribuindo para um material de propriedades não lineares.

As estruturas metálicas, por sua vez, apresentam uma vantagem no uso do aço por ser um material homogêneo com características mecânicas bem-definidas e de simples caracterização (SOUZA, 2017).

O Brasil apresenta uma cultura muito forte na utilização de concreto armado nas edificações, como informa Santos (2006), o concreto é um material estrutural absolutamente hegemônico nas construções das cidades brasileiras. O concreto armado, conforme Bono (2008), é formado por concreto com uma armação em aço. Seu comportamento se deve pela interação entre seus dois componentes, apresentando propriedades não lineares.

As lajes das edificações podem apresentar modelos construtivos diversos, um dos mais utilizados é a laje maciça. Segundo Pinheiro, Muzard e Santos (2010), esse modelo é totalmente preenchido por concreto armado e apoiado em vigas, possuindo geometria retangular nos diferentes planos, representados na Figura 1.

Figura 1 – Laje maciça bi-apoiada.

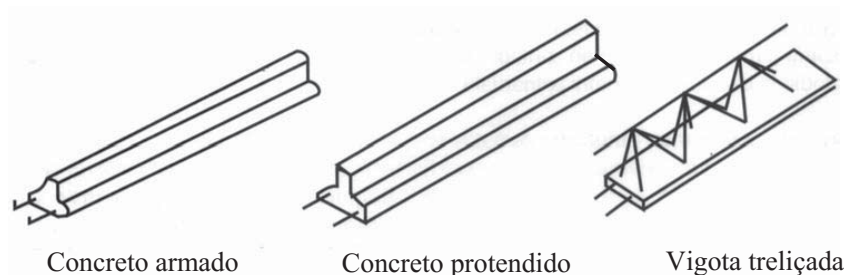


Fonte: Autoria própria (2018).

Um modelo criado para aumentar os vãos e melhorar o aproveitamento do aço e do concreto são as lajes nervuradas. Esse modelo é constituído por um conjunto de vigas que se cruzam solidarizadas pela mesa de compressão. Podem ser moldadas no local ou com nervura pré-moldadas. As moldadas no local são aquelas que são executadas “in loco”, utilizando formas de madeira, polipropileno ou até de metal. Já as pré-moldadas, dispensam a necessidade de formas, pois as vigotas chegam na obra “pré-moldadas”, necessitando apenas de cimbramentos intermediários (PINHEIRO; MUZARD; SANTOS, 2010).

Conforme Franca e Fusco (1997), há três tipos de vigotas pré-moldadas, que estão representados na Figura 2.

Figura 2 – Vigotas pré-moldadas.



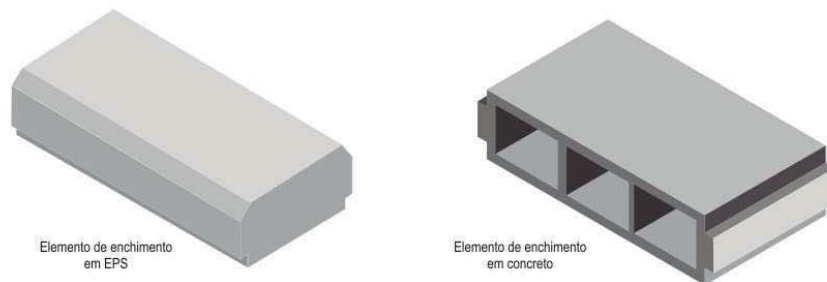
Fonte: Franca e Fusco (1997).

Pinheiro, Muzard e Santos (2010) informam que o formato de lajes nervuradas em vigotas fazem com que diminuam a quantidade de concreto nas áreas tracionadas, tornando possível a utilização de materiais de enchimentos, entre os quais destacam-se: blocos cerâmicos, blocos vazados de concreto e blocos de poliestireno expandido (EPS). Ao utilizar esses materiais, há um alívio no peso próprio.

Os blocos cerâmicos ou de concreto, conforme Pinheiro, Muzard e Santos (2010), são melhores isolantes térmicos do que o concreto maciço. Seu único problema é o elevado

peso específico. Já os blocos de EPS vêm ganhando espaço na execução pelo fato de ser um ótimo isolante térmico-acústico, facilidade de corte, além disso, pelo fato de seu módulo de elasticidade ser baixo, há uma adequada distribuição de cargas, entre outras características.

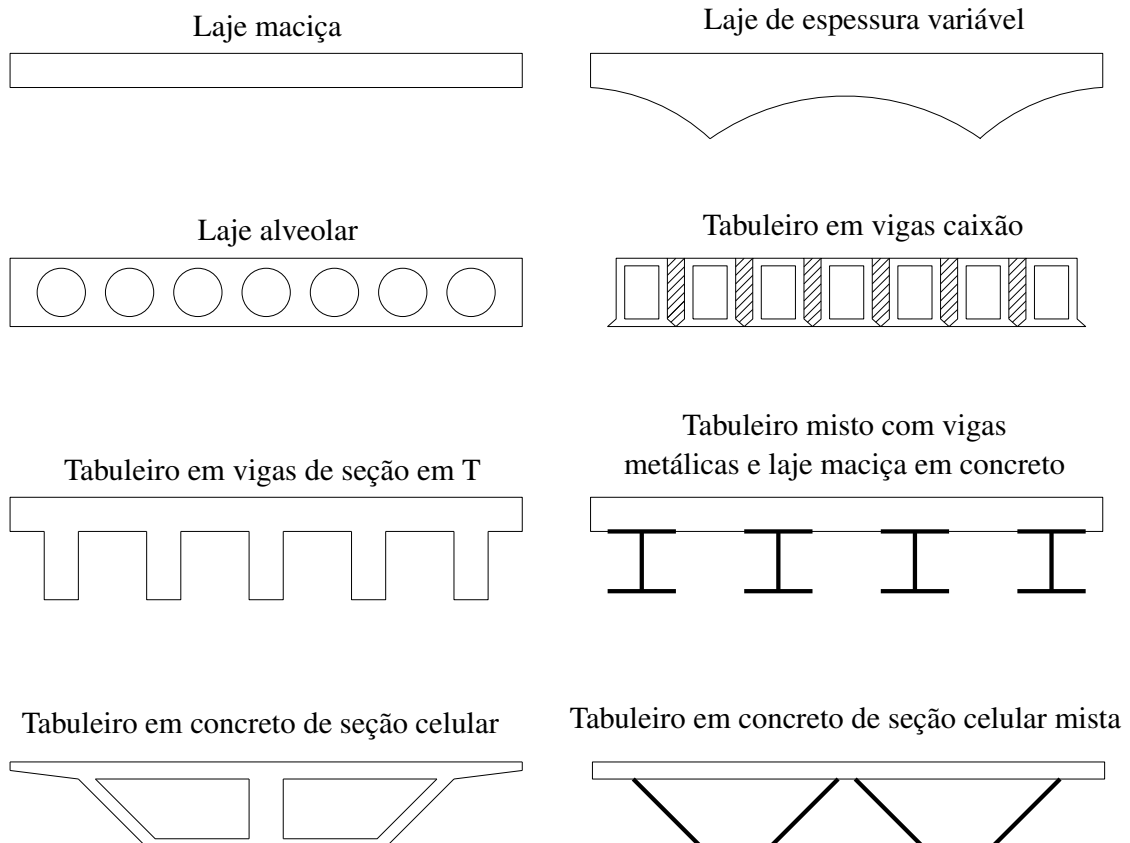
Figura 3 – Blocos de enchimento.



Fonte: Junior (2017).

A construção de pontes e viadutos causam, conforme Cusens e Pama (1975), uma grande repercussão sobre o público. Isso se deve pelo impacto visual que essas construções provocam, além de seus vários benefícios para melhorar a comunicação entre áreas. As principais categorias de tabuleiros de concreto para pontes estão ilustradas na Figura 4.

Figura 4 – Tabuleiros de concreto.



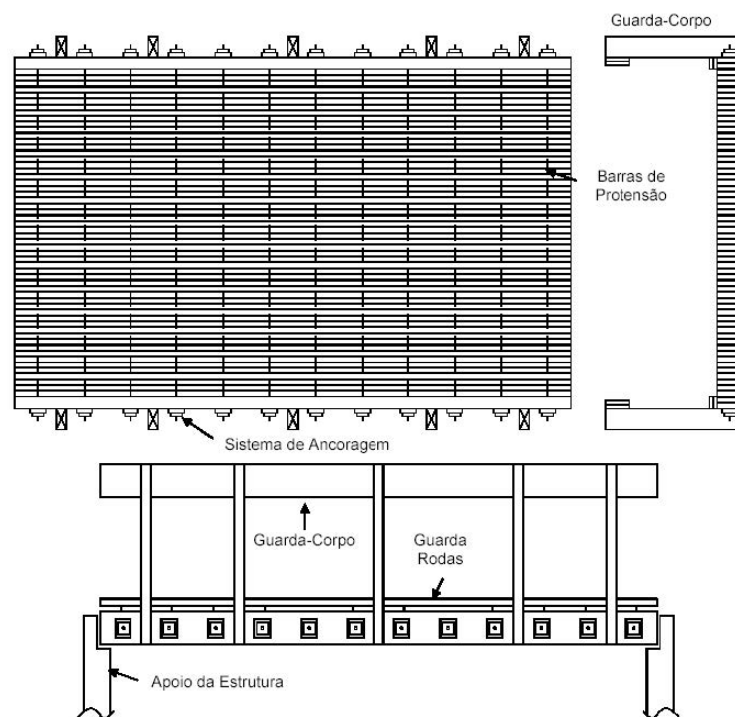
Fonte: Adaptado de Cusens e Pama (1975).



A utilização de madeira no Brasil não é tão forte como a do concreto. Encontra-se, nas estradas e pontes vicinais, estruturas em madeira que, conforme Góes (2005), estão em um estado lastimável, em que o trânsito é dificultado e há o desconforto e insegurança aos usuários. Isso se deve ao fato de que tais estruturas não serem projetadas e construídas por técnicos e construtores especializados em madeira.

Há várias tecnologias em madeira que podem ser aplicadas. Okimoto e Junior (2001) informam que há uma tecnologia na construção de pontes em madeira cujos tabuleiros são constituídos por peças posicionadas ao longo do vão e associadas a um sistema de protensão transversal que as mantém unidas, criando um comportamento estrutural de placa ortotrópica. A representação transversal desse tipo de tabuleiro é dada na Figura 5.

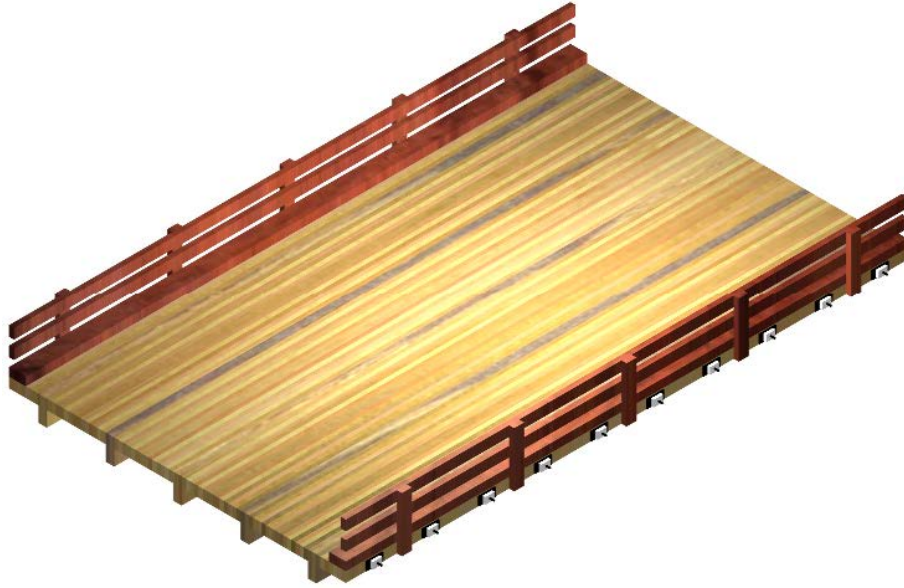
Figura 5 – Planta, seção transversal e elevação de ponte protendida.



Fonte: Okimoto e Junior (2001).

Outra aplicação de tabuleiro de pontes dado no trabalho de Góes (2005), é que pesquisadores da West Virginia University (WVU), para satisfazer a necessidade por vãos maiores, desenvolveram o tipo de estrutura laminada protendida formada por vigas de seção T, ilustrada na Figura 6.

Figura 6 – Ponte protendida formada por vigas de seção T.



Fonte: Góes (2005).

Conforme os exemplos citados acima, há uma variedade de materiais e geometrias para esses elementos de placas. O estudo matemático para a formulação de uma teoria de membranas de placas muito finas, iniciou-se com os trabalhos de Euler (1766) no qual resolveu problemas de vibrações em membranas elásticas retangulares, triangulares e circulares. Seu aluno, Jacces Bernoulli, aplicou a analogia apresentada no trabalho de Euler para vigas, somente com rigidez à flexão (SZILARD, 2004).

A história da teoria de placas finas teve a ajuda da matemática Sophie Germain (1776–1831). Em seu primeiro trabalho sobre a teoria de vibrações em placas ela se baseou em Euler, porém cometeu um erro, apresentando uma EDP errônea dada pela equação (1):

$$\frac{\partial^2 z}{\partial t^2} + \lambda^2 \left( \frac{\partial^6 z}{\partial x^4 \partial y^2} + \frac{\partial^6 z}{\partial x^2 \partial y^4} \right) = 0, \quad (1)$$

em que  $z(x, y, t)$  representa a superfície média da placa em movimento num plano  $X, Y, Z$  cartesiano,  $t$  representa o tempo e  $\lambda$  uma constante física relacionada com a propriedades de vibração da placa. Lagrange, percebeu o erro, obtendo a equação (2), que descreve corretamente a vibração em placas

$$\frac{\partial^2 z}{\partial t^2} + k^2 \left( \frac{\partial^4 z}{\partial x^4} + 2 \frac{\partial^4 z}{\partial x^2 \partial y^2} + \frac{\partial^4 z}{\partial y^4} \right) = 0. \quad (2)$$

Os créditos para L. Navier não podem ser deixados de lado por apresentar corretamente uma EDP que modela a deformação de placas estáticas, ou seja, em que o tempo  $t$ , apresentada pela Equação (2) é nulo. A Equação (3) define, portanto, uma placa

estática com um carregamento lateral  $p(x, y)$ , que possibilitou a introdução de rigorosos métodos matemáticos para análises estruturais para este tipo de elemento. Navier aplicou as hipóteses criadas por Bernoulli, nas quais diz que seções planas permanecerão planas após a deformação, assim,

$$D \left( \frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} \right) = p(x, y), \quad (3)$$

em que  $D$  representa a rigidez a flexão da placa, que é proporcional ao cubo da espessura da mesma e a função  $u(x, y)$  que representa a deformação na superfície média. Esta equação porém, estava limitada para pequenas deformações.

Na Alemanha, a publicação de Kirchhoff “Lectures on Mathematical Physics, Mechanics” (em Alemão) teve um grande impacto acadêmico, pois ocorreu a publicação de uma teoria completa de deformações de placas, baseando-se da mesma forma que Navier e nas hipóteses criadas por Bernoulli para vigas. Sua contribuição para a teoria das placas foi a introdução de forças complementares de contorno, fazendo com que as condições de contorno fossem declaradas em função dos deslocamentos e suas derivadas em função de  $x$  ou  $y$ . Além disso, pode-se citar as inúmeras contribuições de Timoshenko para a criação da teoria, como exemplo a solução de placas circulares com grandes deformações e a formulação de problemas de estabilidades elásticas (SZILARD, 2004).

Como pode-se verificar, os sistemas estruturais são complexos, exigindo modelos sofisticados de análises para o dimensionamento sob carregamentos do projeto. Um dos métodos utilizados, sugeridos por Góes et al. (2006) é o modelo de placa ortotrópica equivalente, que leva em conta as dimensões, formas e materiais utilizados. Esse modelo, também discutido em Cusens e Pama (1975), faz com que a complexidade da ortotropia em questões de material e geometria, são reduzidas a uma placa maciça de propriedades equivalentes.

Outro ponto, abordado em Corrêa (1991), ao considerar que a estrutura está em regime elástico, faz com que apresente uma fácil compreensão das ações e do comportamento estrutural, tornando possível realizar sobreposições de efeitos.

Conforme Fortuna (2000), as EDPs são uma classe de equações que “não possuem uma teoria matemática suficiente desenvolvida para permitir a obtenção de soluções analíticas em condições arbitrárias e de contornos gerais”. Por tal motivo, muitos estudos são realizados com tratamento teórico e/ou experimental. Com o advento dos computadores nos anos 50, outra alternativa de se analisar esses problemas foram com as análises nu-

méricas utilizando métodos computacionais. Porém, vale ressaltar, que essa alternativa não desmotivou o uso de técnicas experimentais e análises teóricas e sim as complementaram, fazendo com que houvesse maiores comprovações entre os resultados teóricos e experimentais, validando ou não o modelo de análise empregado.

Para encontrar computacionalmente a solução das EDPs, deve-se trabalhar em um domínio discretizado, que confere, conforme Neves (2004), no processo de transformar um conjunto infinito de dados em um conjunto finito.

Conforme Neves (2004), uma maneira de discretizar o domínio sem a preocupação de boa distribuição, é considerar um mesmo espaçamento para todos os pontos do domínio, para tanto, tem para a abscissa um intervalo  $[a, b]$  com  $M + 1$  elementos, e para as ordenadas um intervalo  $[c, d]$  com  $N + 1$  elementos, assim, os passos de discretização terão nas abscissas amplitude  $h = (b - a)/M$ , e para as ordenadas amplitude  $k = (d - c)/N$ . Por conseguinte, teremos os pontos com coordenada  $(x_i, y_j)$  para a discretização.

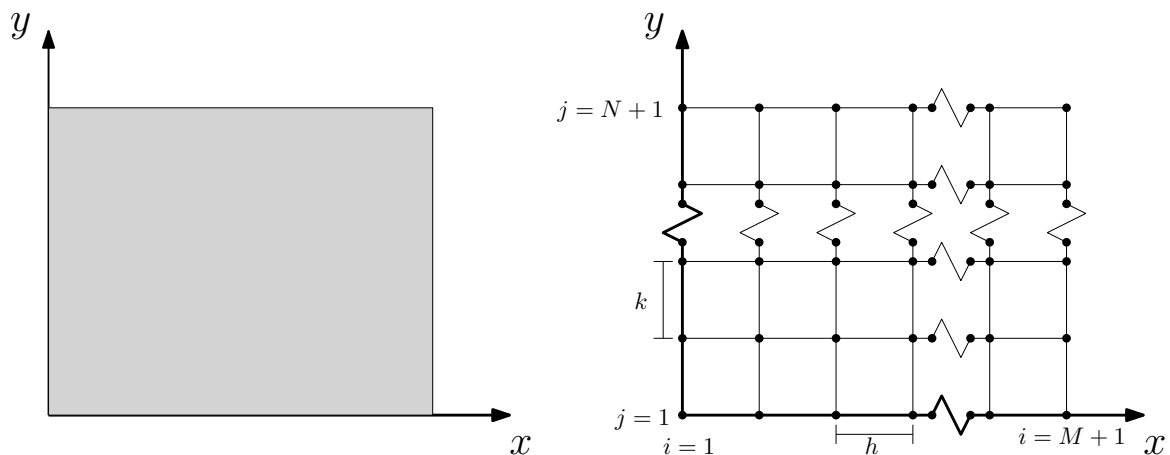
$$x_i = a + (i - 1)h, \quad \forall \quad i = 1, 2, 3, \dots, M + 1,$$

e,

$$y_j = c + (j - 1)k, \quad \forall \quad j = 1, 2, 3, \dots, N + 1.$$

Em outras palavras, conforme Fortuna (2000), ao escolher valores de espaçamento de  $h$  nas abscissas e  $k$  nas ordenadas, ao realizar intersecções de linhas horizontais e verticais nessas distâncias, há a discretização do domínio conforme a Figura 7, em que  $i$  e  $j$  definem as coordenadas de um ponto de  $i$ -ésima coluna e  $j$ -ésima linha, respectivamente.

Figura 7 – Discretização de um domínio contínuo.



Fonte: Adaptado de Fortuna (2000).

Com o intuito de obter a solução de uma EDP e ser possível implementar computacionalmente, Fortuna (2000) afirma que deve-se “representar os diferenciais da EDP por expressões algébricas, ou seja, discretizar a EDP”.

Pode-se citar uma variedade de trabalhos que ajudaram no desenvolvimento teórico e prático em questões de análises estruturais em função da teoria de deformações de placas delgadas. Os métodos mais conhecidos e empregados na comunidade acadêmica são os MDF e MEF. Como exemplo, Leal (2015), em que implementou-se computacionalmente uma formulação baseada no MEF para realizar uma análise elástico-linear de placas isotrópicas.

Júnior (2008) elaborou um programa que visa demonstrar o lado teórico de lajes maciças, obtendo esforços, dimensionamento e detalhes da estrutura de concreto armado, assim como ter um respaldo educacional, dando uma visão geral do cálculo de placas. Para a formulação computacional, ele utilizou o MDF.

No trabalho realizado por Ghods e Mir (2014), em que ele investiga a performance do MDF em análises de placas retangulares.

Em Roknuzzaman et al. (2015), o MDF foi utilizado para analisar uma placa retangular com uma abertura. Esse problema pode ser verificado para o dimensionamento de lajes com aberturas para passagem de tubulações, entre outros motivos. O autor discute que o MDF é o mais antigo, em especial para análise de placas, pela facilidade de se realizar uma análise em um domínio retangular.

#### *4.1 Considerações sobre a revisão bibliográfica*

Há uma gama de exemplos de placas ortotópicas na engenharia civil, tanto para o desenvolvimento de pontes, viadutos, edifícios residenciais e comerciais. Os modelos empregados nessas edificações apresentam geometrias complexas, por tal motivo, suas análises foram simplificadas com o uso da equação de placa delgada ortotrópica equivalente.

Com os avanços tecnológicos, as análises desses elementos estruturais ocorrem por meio dos métodos numéricos. Os mais utilizados no meio acadêmico são o MEF e MDF. Conforme Gray e Schnurr (1975), o MEF é mais eficiente e mais preciso que o MDF, porém, o MDF é mais simples de ser implementado.

O MDF se mostrou eficaz no trabalho feito por Ghods e Mir (2014) para a solução de placas retangulares finas, por este motivo, para a formulação deste trabalho, aplicado a

teoria baseada em MDF para solucionar a ED que modela placas delgadas ortotrópicas equivalentes.

Conforme Fortuna (2000), deve-se levar em consideração que a solução numérica não será exata devido a vários fatores, que podem ser devidos ao erro de truncamento para as aproximações das derivadas nas EDPs via série de Taylor, ao arredondamento feitos pelo computador e pelas aproximações das condições de contorno do problema.

#### 4.2 Teoria das placas delgadas ortotrópicas equivalentes

Para o estabelecimento da teoria de placas ortotrópicas equivalentes, primeiramente adota-se certas hipóteses, como a de que a placa delgada, ou seja, o deslocamento  $u$  causado pelo carregamento deve ser menor que sua espessura. Tal teoria é semelhante a apresentada em Timoshenko e Woinowsky-Krieger (1959), com certas modificações abordadas em Cusens e Pama (1975).

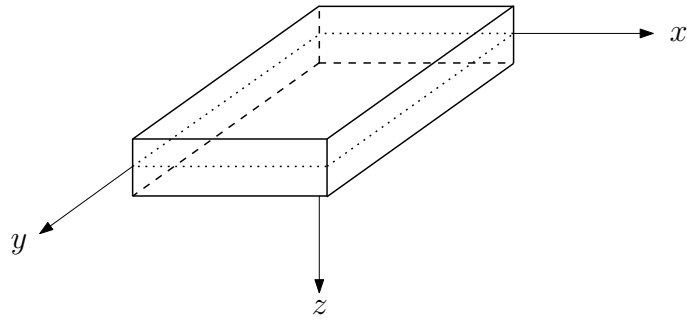
1. O material da placa é elástico linear, homogêneo e ortotrópico;
2. Não há deformação no plano médio da placa;
3. As linhas normais à superfície média permanecem normais à superfície fletida;
4. As tensões normais que atuam perpendicular á superfície média podem ser desprezadas;
5. Regime de pequenas deformações.

O elemento de placa estará posicionado sobre outros elementos estruturais indeformáveis, tais como as vigas e pilares. Submetido um carregamento uniformemente distribuído por toda a sua extensão.

##### 4.2.1 Estados plano de deformações

Inicialmente, da mesma forma realizada em Timoshenko e Woinowsky-Krieger (1959), posiciona-se uma placa retangular, com seções transversais também retangulares de dimensões conhecidas e com uma espessura  $t$ . O plano  $xy$  partirá do plano transversal central da placa, com o eixo  $y$  coincidindo com o lado longitudinal e o eixo  $z$  positivo para baixo, conforme a Figura 8.

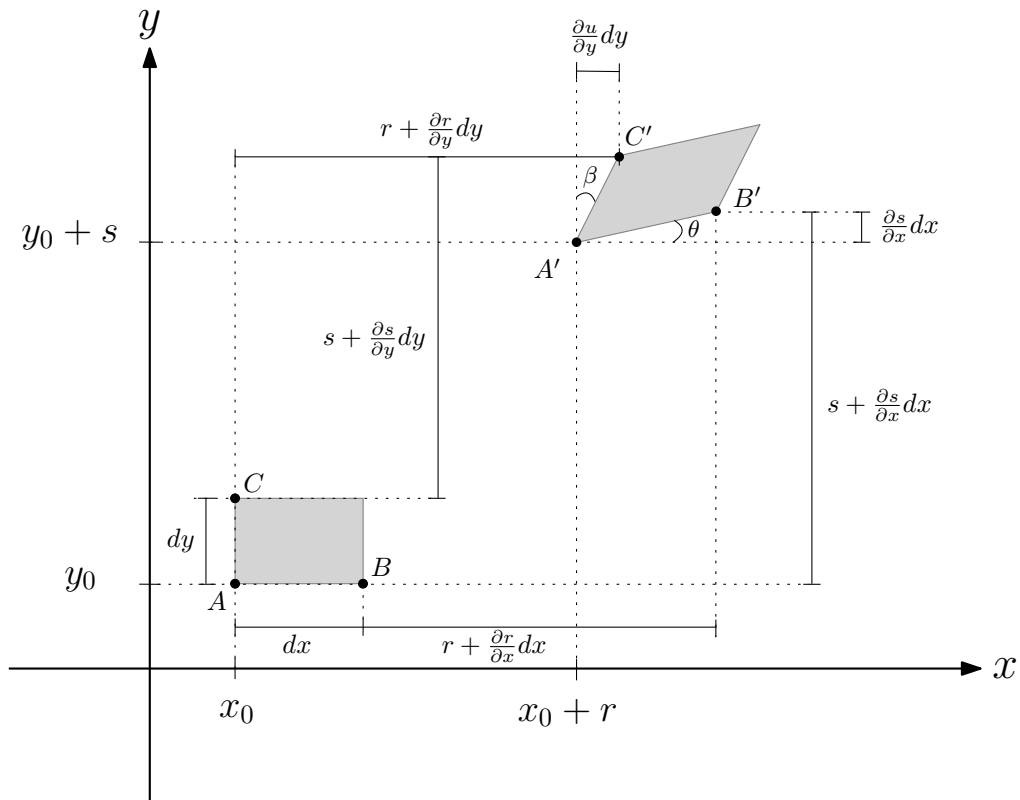
Figura 8 – Posicionamento dos eixos principais.



Fonte: Autoria própria (2018).

O mesmo procedimento é tomado em Villaça e Garcia (1998). Ao solicitar a placa com um carregamento  $p(x, y)$ , tal elemento sofrerá uma deformação rígida pela equação de deformação  $u(x, y)$ . Ao analisar um elemento infinitesimal, tem-se a representação na Figura 9 no plano  $xy$ .

Figura 9 – Deformação elemento infinitesimal.



Fonte: Adaptado de Villaça e Garcia (1998).

Como pode-se verificar, o elemento acima sofreu uma mudança de posição, dimensão e forma. Essa modificação é definida como deformação. A relação entre o alongamento que sofreu com o comprimento inicial é denominado deformação linear  $\epsilon$ ,

$$\epsilon = \frac{ds' - ds}{ds}, \quad (4)$$

sendo  $ds$  o comprimento inicial,  $ds'$  o comprimento após o carregamento.

A relação entre o ângulo perpendicular original entre dois pontos com o ângulo  $\alpha$ , formado após a deformação do corpo, é denominado como deformação angular  $\lambda$ ,

$$\lambda = \frac{\pi}{2} - \alpha. \quad (5)$$

De acordo com a Figura 9, pode se verificar que o elemento sofreu uma mudança de posicionamento devido as deformações  $r$  e  $s$  que são, respectivamente, as deformações nas direções dos eixos  $x$  e  $y$ .

Os novos comprimentos do elemento,  $A'B'$  e  $A'C'$ , podem ser relacionados com os comprimentos originais  $AB$  e  $AC$  da seguinte maneira:

$$dx + r + \frac{\partial r}{\partial x} dx = r + A'B' \cos(\theta) \quad (6)$$

e

$$dy + s + \frac{\partial s}{\partial y} dy = s + A'C' \cos(\beta), \quad (7)$$

em que  $\cos(\theta) \approx 1$  e  $\cos(\beta) \approx 1$  pelo fato de estar em um regime de pequenas deformações.

Devido ao item 5, pode-se escrever as equações (6) e (7) da forma que se segue

$$A'B' = dx + \frac{\partial r}{\partial x} dx \quad (8)$$

e

$$A'C' = dy + \frac{\partial s}{\partial y} dy, \quad (9)$$

Da equação (4), pode-se realizar as seguinte relações entre as deformações lineares no plano

$$\epsilon_x = \frac{\partial r}{\partial x} \quad (10)$$

e

$$\epsilon_y = \frac{\partial s}{\partial y}. \quad (11)$$

A deformação angular, equação (5), no plano, Figura 9, será:

$$\lambda_{xy} = \theta + \beta. \quad (12)$$



Pela consideração de pequenas deformações,  $\text{sen}(\theta) \approx \theta$  e  $\text{sen}(\beta) \approx \beta$ , desta forma, ao observar a Figura 9, tem-se

$$\text{sen}(\theta) \approx \theta = \frac{\frac{\partial s}{\partial x} dx}{A'B'} \quad (13)$$

e

$$\text{sen}(\beta) \approx \beta = \frac{\frac{\partial r}{\partial y} dy}{A'C'}, \quad (14)$$

e ao relacionar as equações (8) e (9) com as equações (10) e (11), é possível escrever as equações (13) e (14), da seguinte forma

$$\theta = \frac{\frac{\partial s}{\partial x} dx}{dx(1 + \epsilon_x)} \quad (15)$$

e

$$\beta = \frac{\frac{\partial r}{\partial y} dy}{dy(1 + \epsilon_y)}, \quad (16)$$

e por saber que  $\epsilon_x$  e  $\epsilon_y$  são estritamente menores do que a unidade, tem-se

$$\theta = \frac{\partial s}{\partial x} \quad (17)$$

e

$$\beta = \frac{\partial r}{\partial y}. \quad (18)$$

Desta forma, a deformação angular do elemento, dada pela equação (12), pode ser escrita como

$$\lambda_{xy} = \frac{\partial s}{\partial x} + \frac{\partial r}{\partial y}. \quad (19)$$

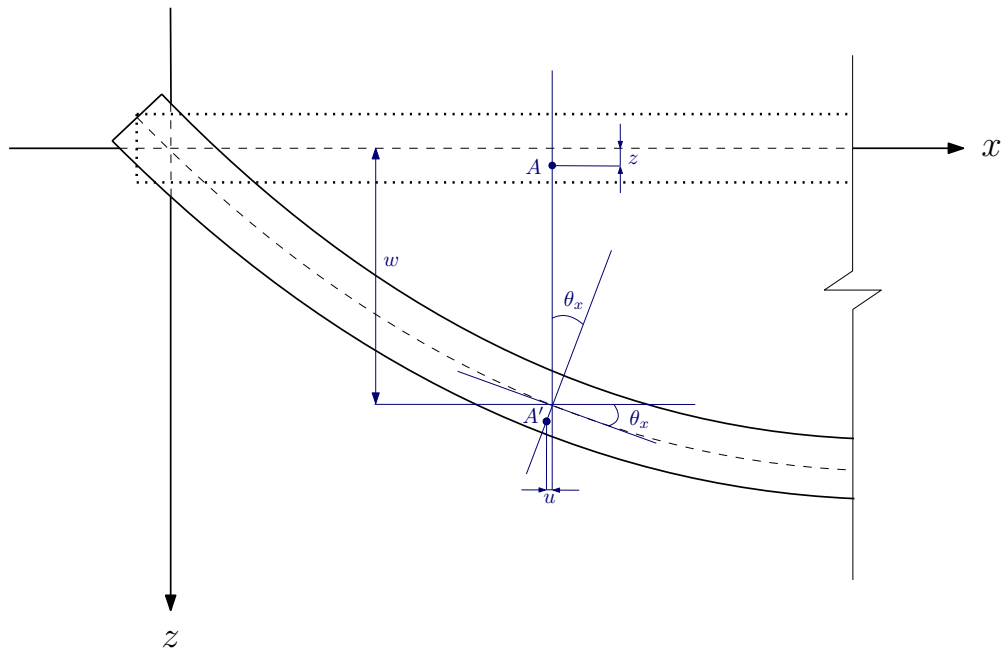
A mesma análise é feita para os outros planos, obtendo as seguintes relações entre as deformações e deslocamentos

$$\epsilon_x = \frac{\partial r}{\partial x}, \quad \epsilon_y = \frac{\partial s}{\partial y}, \quad \epsilon_z = \frac{\partial u}{\partial z}, \quad (20)$$

$$\lambda_{xy} = \frac{\partial s}{\partial x} + \frac{\partial r}{\partial y}, \quad \lambda_{xz} = \frac{\partial r}{\partial z} + \frac{\partial u}{\partial x} \quad e \quad \lambda_{yz} = \frac{\partial s}{\partial z} + \frac{\partial u}{\partial y}.$$

Conforme Villaça e Garcia (1998), ao analisar a placa deformada transversalmente, conforme Figura 10, é possível encontrar uma relação das deformações lineares e angulares em função dos deslocamentos  $u(x, y)$ .

Figura 10 – Seção transversal da placa deformada.



Fonte: Adaptado de Villaça e Garcia (1998).

Observando o ponto  $A$  da imagem, no plano  $xz$ , verifica-se que ele possui um deslocamento  $r$  na direção  $x$  e um deslocamento  $u$  na direção  $z$ . Ao escrever  $r$  em função do giro da seção  $\theta_x$ , lembrando que a seção permanecerá indeformada, tem-se

$$r = -z\theta_x = -z \frac{\partial u}{\partial x}. \quad (21)$$

Da mesma forma, pegando novamente o ponto  $A$ , no plano  $yz$ , verifica-se que

$$s = -z\theta_y = -z \frac{\partial u}{\partial y}. \quad (22)$$

Substituindo (21) e (22) nas Equações (10), (11) e (19), observando uma relação entre as deformações lineares-angulares com os deslocamentos,

$$\epsilon_x = -z \frac{\partial^2 u}{\partial x^2}, \quad \epsilon_y = -z \frac{\partial^2 u}{\partial y^2}, \quad \lambda_{xy} = -2z \frac{\partial^2 u}{\partial x \partial y}, \quad (23)$$

lembrando que  $\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial y \partial x}$  para manter a continuidade.

#### 4.2.2 Estados plano de tensões

Como considerou-se que  $p$  cause um estado de tensão linear elástico, conforme Hibbeler (2003), as Leis de Hooke generalizadas são verdadeiras dentro dessas condições.

Elas ditam as deformações ao longo do eixo  $x$  ( $\epsilon_x$ ) e as deformações ao longo do eixo  $y$  ( $\epsilon_y$ ), em termos de tensões, dados por,

$$\epsilon_x = \frac{\sigma_x}{E_x} - \frac{\nu_y \sigma_y}{E_y} \quad (24)$$

e

$$\epsilon_y = \frac{\sigma_y}{E_y} - \frac{\nu_x \sigma_x}{E_x}, \quad (25)$$

onde  $E_x$  e  $E_y$  são os módulos de Young do material,  $\nu_x$  e  $\nu_y$  são os coeficientes de Poisson, nos planos  $x$  e  $y$  respectivamente.

Ao isolar  $\sigma_y$  da equação (25), tem-se

$$\sigma_y = \epsilon_y E_y + \frac{\nu_x \sigma_x E_y}{E_x}. \quad (26)$$

Ao substituir a equação (26) na equação (24), encontra-se a relação de tensão  $\sigma_x$  em função das deformações  $\epsilon_x$  e  $\epsilon_y$ , ou seja,

$$\sigma_x = \frac{E_x}{(1 - \nu_x \nu_y)} [\epsilon_x + \nu_y \epsilon_y]. \quad (27)$$

Procedendo da mesma forma, encontra-se

$$\sigma_y = \frac{E_y}{(1 - \nu_x \nu_y)} [\epsilon_y + \nu_x \epsilon_x]. \quad (28)$$

sendo  $\sigma_x$  e  $\sigma_y$  as tensões normais ao plano  $x$  e  $y$ , respectivamente.

A lei de Hooke para cisalhamento conforme Hibbeler (2003), é definida por (29),

$$\tau_{xy} = G \lambda_{xy}, \quad (29)$$

onde  $\tau_{xy}$  é a tensão de cisalhamento no plano  $xy$  e  $G$  é o módulo de elasticidade ao cisalhamento.

Das equações (23), (27), (28) e (29), tem-se

$$\sigma_x = -\frac{E_x z}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right], \quad (30)$$

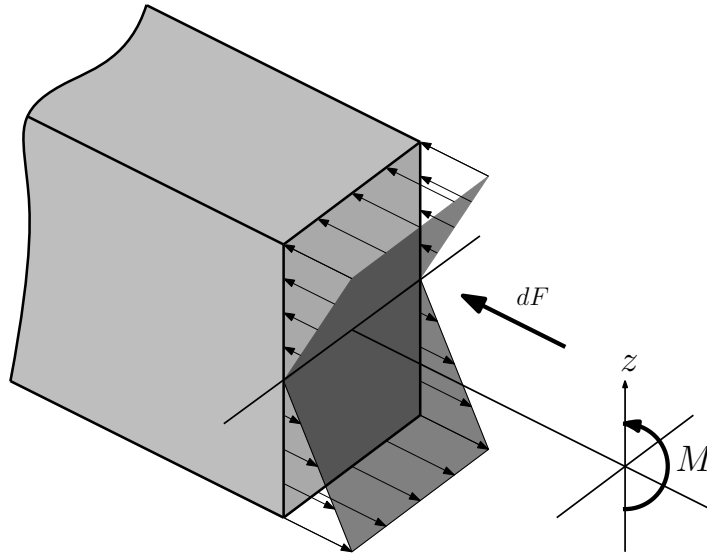
$$\sigma_y = -\frac{E_y z}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial y^2} + \nu_x \frac{\partial^2 u}{\partial x^2} \right] \quad (31)$$

e

$$\tau_{xy} = -2Gz \frac{\partial^2 u}{\partial x \partial y}. \quad (32)$$

Como dito nas considerações, o momento fletor  $M$  é tal que as tensões normais na seção permanecem no regime elástico, visto que a variação de tensão ao longo da seção é linear partindo da linha neutra, conforme representado na Figura 11.

Figura 11 – Distribuição de tensão normal no plano transversal.



Fonte: Autoria própria (2018).

As tensões na seção podem ser obtidas a partir do momento  $M$  quando ele for igual ao momento produzido pela distribuição de tensão em torno do eixo neutro, ou seja, a somatória de momentos gerados pela variação de  $dF$  ao longo de  $z$ , deverá ser igual ao momento causado, assim

$$M = \sum z dF. \quad (33)$$

Como tensão é força sobre área,  $\sigma = F/A$ , tem-se que  $dF = dA \cdot \sigma$ , daí

$$M = \int_A z \sigma dA. \quad (34)$$

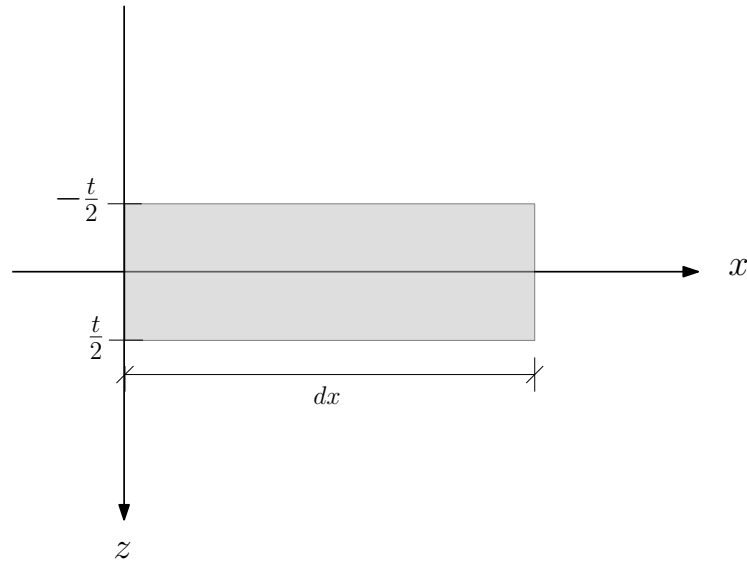
Substituindo as equações (30) e (31) na equação (34), têm-se as seguintes situações,

$$M_x = \int_A -\frac{E_x z^2}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] dA \quad (35)$$

e

$$M_y = \int_A -\frac{E_y z^2}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial y^2} + \nu_x \frac{\partial^2 u}{\partial x^2} \right] dA. \quad (36)$$

Primeiramente, resolve-se a equação (35). Como é uma integral de área, são necessários os limites de integração que serão obtidos a partir da Figura 12.

Figura 12 – Plano transversal na direção  $x$ .

Fonte: Autoria própria (2018).

Para o eixo  $x$ , os limites de integração serão de 0 até  $dx$  e para o eixo  $z$  de  $t/2$  até  $-t/2$ ,

$$\begin{aligned}
 M_x &= \int_0^{dx} \int_{t/2}^{-t/2} -\frac{E_x z^2}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] dz dx \\
 &= -\frac{E_x}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \int_0^{dx} \int_{t/2}^{-t/2} z^2 dz dx \\
 &= -\frac{E_x}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \int_0^{dx} \frac{z^3}{3} \Big|_{t/2}^{-t/2} dx \\
 &= -\frac{E_x}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \int_0^{dx} \frac{1}{3} \left( \frac{t^3}{2^3} - \left( -\frac{t^3}{2^3} \right) \right) dx \\
 &= -\frac{E_x}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \int_0^{dx} \frac{t^3}{12} dx \\
 &= -\frac{E_x}{(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \frac{t^3}{12} dx,
 \end{aligned} \tag{37}$$

portanto,

$$M_x = -\frac{E_x t^3 dx}{12(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right]. \tag{38}$$

Analogamente,

$$M_y = -\frac{E_y t^3 dy}{12(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial y^2} + \nu_x \frac{\partial^2 u}{\partial x^2} \right]. \tag{39}$$

Para  $M_{xy}$ , utiliza-se os mesmos parâmetros de integração definidos na Figura 12, conforme mostra a equação (40),

$$\begin{aligned}
 M_{xy} &= \int_0^{dx} \int_{t/2}^{-t/2} -2Gz^2 \frac{\partial^2 u}{\partial x \partial y} dz dx \\
 &= -2G \frac{\partial^2 u}{\partial x \partial y} \int_0^{dx} \int_{t/2}^{-t/2} z^2 dz dx \\
 &= -2G \frac{\partial^2 u}{\partial x \partial y} \int_0^{dx} \frac{z^3}{3} \Big|_{-t/2}^{t/2} dx \\
 &= -G \frac{\partial^2 u}{\partial x \partial y} \int_0^{dx} \frac{t^3}{6} dx \\
 &= -G dx \frac{t^3}{6} \frac{\partial^2 u}{\partial x \partial y},
 \end{aligned} \tag{40}$$

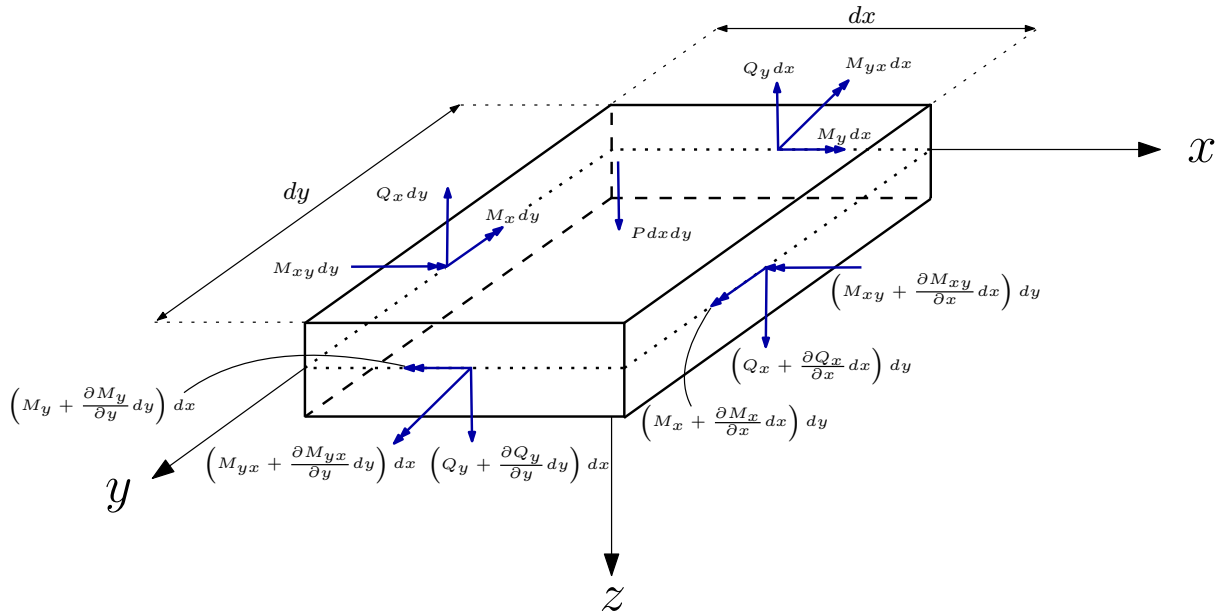
portanto,  $M_{xy}$  pode ser expresso por:

$$M_{xy} = -G dx \frac{t^3}{6} \frac{\partial^2 u}{\partial x \partial y}. \tag{41}$$

#### 4.2.3 Equilíbrio de uma seção infinitesimal

Como as funções cortante ( $Q$ ) e momento ( $M$ ) são contínuas no plano, ao analisar as forças no elemento infinitesimal, obtém-se a representação dada na Figura 13.

Figura 13 – Equilíbrio infinitesimal.



Fonte: Adaptado de Villaça e Garcia (1998).

A equações de equilíbrio no elemento infinitesimal acima das forças verticais, é dada por

$$-p dx dy - Q_x dy - Q_y dx + \left( Q_x + \frac{\partial Q_x}{\partial x} dx \right) dy + \left( Q_y + \frac{\partial Q_y}{\partial y} dy \right) dx = 0, \quad (42)$$

e ao reorganizar os termos, obtém-se

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} = -p. \quad (43)$$

Da mesma forma, a equações de equilíbrio do elemento dos momentos fletores em torno ao eixo  $y$ , pode ser escrita por

$$\begin{aligned} & -p \frac{dy dx^2}{2} - M_x dy + Q_y \frac{dx^2}{2} - M_{yx} dx - \left( Q_x + \frac{\partial Q_x}{\partial x} dx \right) dy dx \\ & + \left( M_x + \frac{\partial M_x}{\partial x} dx \right) dy - \left( Q_y + \frac{\partial Q_y}{\partial y} dy \right) \frac{dx^2}{2} + \left( M_{yx} + \frac{\partial M_{yx}}{\partial y} dy \right) dx = 0, \end{aligned} \quad (44)$$

e ao reorganizar os termos, obtém-se

$$\frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} = Q_x + p \frac{dx}{2} + \frac{\partial Q_y}{\partial y} \frac{dx}{2} + \frac{\partial Q_x}{\partial x} dx, \quad (45)$$

Como as parcelas que acompanham  $dx$  são estritamente pequenas, tais termos podem ser desconsiderados, a equação (45) pode ser escrita

$$\frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} = Q_x. \quad (46)$$

Por fim, a equação de equilíbrio do elemento dos momentos fletores em torno ao eixo  $x$ , pode ser escrita como

$$\begin{aligned} & p \frac{dydx^2}{2} + M_y dx - Q_x \frac{dy^2}{2} + M_{xy} dy + \left( Q_x + \frac{\partial Q_x}{\partial x} dx \right) \frac{dy^2}{2} \\ & - \left( M_y + \frac{\partial M_y}{\partial y} dy \right) dx + \left( Q_y + \frac{\partial Q_y}{\partial y} dy \right) dx - \left( M_{xy} + \frac{\partial M_{xy}}{\partial x} dx \right) dy = 0, \end{aligned} \quad (47)$$

e ao reorganizar os termos, encontra-se

$$\frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} = Q_y + p \frac{dy}{2} + \frac{\partial Q_x}{\partial x} \frac{dy}{2} + \frac{\partial Q_y}{\partial y} dy, \quad (48)$$

Como as parcelas que acompanham  $dy$  são estritamente pequenas, tais serão desconsiderados, daí

$$\frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} = Q_y. \quad (49)$$

Ao substituir as equações (46) e (49) na equação (43), obtém-se

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{yx}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} + \frac{\partial^2 M_{xy}}{\partial y \partial x} = -p. \quad (50)$$

Como os momentos são contínuos ao longo da placa,  $\frac{\partial^2 M_{yx}}{\partial x \partial y} = \frac{\partial^2 M_{xy}}{\partial y \partial x}$ , assim

$$\frac{\partial^2 M_x}{\partial x^2} + 2 \frac{\partial^2 M_{yx}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} = -p. \quad (51)$$

Ao substituir as relações (38), (39) e (41), na equação (51), obtém-se

$$\begin{aligned} & \frac{\partial^2}{\partial x^2} \left( -\frac{E_x t^3 dx}{12(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right] \right) \\ & + 2 \frac{\partial^2}{\partial x \partial y} \left( -G dx \frac{t^3}{6} \frac{\partial^2 u}{\partial x \partial y} \right) \\ & + \frac{\partial^2}{\partial y^2} \left( -\frac{E_y t^3 dy}{12(1 - \nu_x \nu_y)} \left[ \frac{\partial^2 u}{\partial y^2} + \nu_x \frac{\partial^2 u}{\partial x^2} \right] \right) = -p. \end{aligned} \quad (52)$$

Ao denotar, conforme Cusens e Pama (1975),  $D_x = E_x t^3 dx / 12(1 - \nu_x \nu_y)$ ,  $D_1 = E_x t^3 \nu_y dx / 12(1 - \nu_x \nu_y)$ ,  $D_y = E_y t^3 dy / 12(1 - \nu_x \nu_y)$  e  $D_2 = E_y t^3 \nu_x dy / 12(1 - \nu_x \nu_y)$  e  $D_{xy} = G dx t^3 / 6$  em que a rigidez à flexão é definida por  $D_x$  e  $D_y$ , a rigidez ao acoplamento é definido por  $D_1$  e  $D_2$ , e a rigidez à torção é definido por  $D_{xy}$ , encontra-se

$$D_x \frac{\partial^4 u}{\partial x^4} + (D_1 + 2D_{xy} + D_2) \frac{\partial^4 u}{\partial x^2 \partial y^2} + D_y \frac{\partial^4 u}{\partial y^4} = p \quad (53)$$



Nomeando  $2H = (D_1 + 2D_{xy} + D_2)$ , a EDP governante de uma placa ortotrópica em função dos deslocamentos é dada na equação (54).

$$D_x \frac{\partial^4 u}{\partial x^4} + 2H \frac{\partial^4 u}{\partial x^2 \partial y^2} + D_y \frac{\partial^4 u}{\partial y^4} = p(x, y). \quad (54)$$

#### 4.2.4 Condições de contorno

Para que seja possível a solução da equação (54), é necessário conhecer certos valores. As placas são elementos estruturais planos com carregamentos ortogonais utilizados em diversas aplicações estruturais, com diferentes geometrias e condições de vinculações.

Para o caso de edifícios é comum as placas apresentarem bordas engastadas e/ou apoiadas, ou ainda bordas livres, no caso de sacadas. Para as pontes em placas, é comum duas bordas paralelas estarem simplesmente apoiadas e as outras duas livres. Assim, há uma grande variedade de condições de contorno.

Para este trabalho, será considerado somente as condições de contorno de tabuleiros de pontes que, em sua maioria, são bi-apoiados.

##### • Bordas apoiadas

Ao considerar que uma borda está apoiada, fixando  $x = a$ , o momento fletor em torno de  $x$  será nulo assim como o valor de sua deflexão, ou seja,

$$M_x = \frac{\partial^2 u}{\partial x^2}(a, y) = 0 \quad \text{e} \quad u(a, y) = 0. \quad (55)$$

Analogamente, se a borda em  $y$  está apoiada,  $y = c$ , o momento fletor em torno de  $y$  será nulo, assim como a deflexão,

$$M_y = \frac{\partial^2 u}{\partial y^2}(x, c) = 0 \quad \text{e} \quad u(x, c) = 0. \quad (56)$$

##### • Bordas livres

Ao considerar que uma borda está livre,  $x = a$ , o momento fletor em torno de  $x$  será nulo assim como o valor da força cortante, ou seja,

$$M_x = \frac{\partial^2 u}{\partial x^2}(a, y) = 0 \quad \text{e} \quad Q_x = \frac{\partial^3 u}{\partial x^3}(a, y) = 0. \quad (57)$$

De forma análoga, se esta borda  $y = c$ , o momento fletor em torno de  $y$  será nulo assim como o valor da força cortante, ou seja,

$$M_y = \frac{\partial^2 u}{\partial y^2}(x, c) = 0 \quad \text{e} \quad Q_y = \frac{\partial^3 u}{\partial y^3}(x, c) = 0. \quad (58)$$

### 4.3 Métodos das diferenças finitas

O MDF consiste, conforme Burden e Faires (2010), em solucionar um problema de contorno substituindo as derivadas por uma equação de diferenças, denominadas quocientes de diferenças. Esses quocientes de diferenças, conforme Fortuna (2000), consiste em aproximações do operador contínuo diferencial ( $\frac{\partial}{\partial x}$ ) por uma aproximação discreta a partir de valores de um número finito de pontos. Em outras palavras, a ideia geral do MDF consiste na discretização do domínio e a substituição das derivadas presentes na ED por aproximações envolvendo valores numéricos da função.

Sendo assim, o primeiro passo para aplicar o MDF é realizar a discretização do domínio, para tanto, conforme Neves (2004), para realizar a discretização sem a preocupação de boa distribuição, é considerado um mesmo espaçamentos para todos os pontos do domínio. Sendo  $x$  e  $y$  variáveis independentes, seleciona-se números inteiros  $M > 0$  e  $N > 0$  e divide o intervalo  $[a, b]$  em  $M$  subintervalos, assim como divide o intervalo  $[c, d]$  em  $N$  subintervalos, de mesmo comprimento. Os passos de discretização terão espaçamento nas abscissas  $h = (b - a)/M$ , e espaçamento para as ordenadas  $k = (d - c)/N$ . Por conseguinte, teremos os pontos com coordenada  $(x_i, y_j)$  para a discretização, sendo

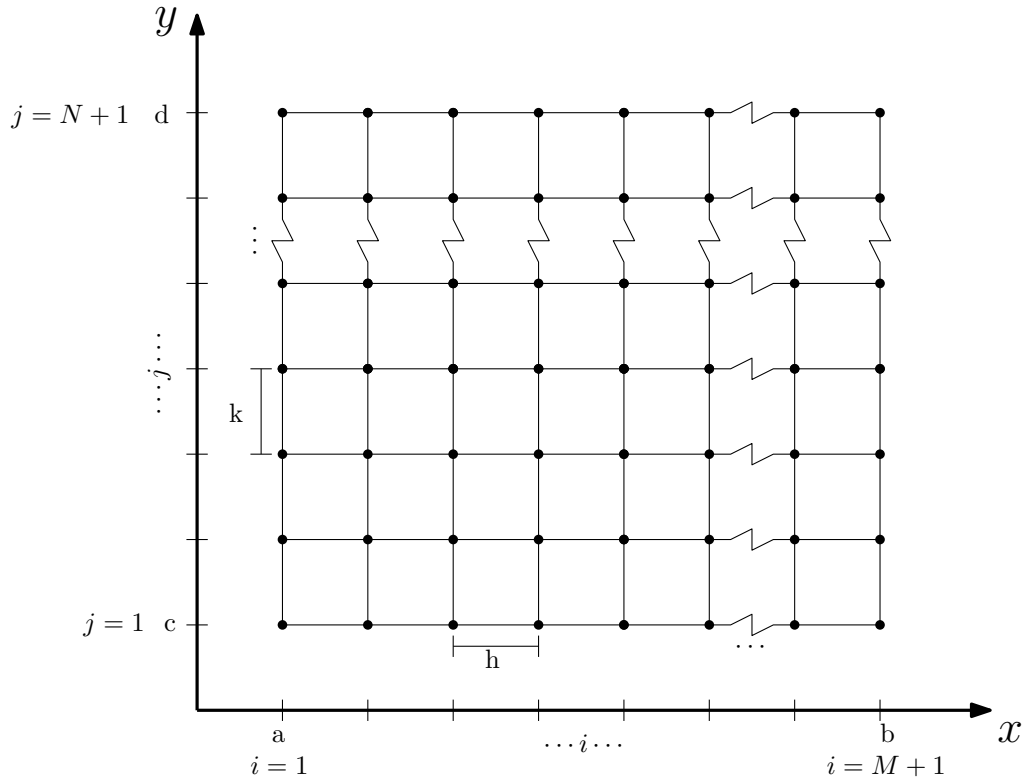
$$x_i = a + (i - 1)h, \quad \forall \quad i = 1, 2, 3, \dots, M + 1$$

e

$$y_j = c + (j - 1)k, \quad \forall \quad j = 1, 2, 3, \dots, N + 1.$$

As extremidades desses subintervalos são os pontos da malha, onde  $x_1 = a$ ,  $x_{M+1} = b$ ,  $y_1 = c$  e  $y_{N+1} = d$ , conforme a Figura 14.

Figura 14 – Domínio discretizado.



Fonte: Autoria própria (2018).

#### 4.3.1 Expansões em série de Taylor

Como estamos interessados em obter uma expressão que relacione a variação de  $u$  com incrementos de  $x$  e  $y$ , simultaneamente, devemos utilizar a expansão de série de Taylor de função de duas variáveis, onde  $u$  é uma função contínua e que apresenta derivadas parciais contínuas até ordem  $n$  na região de interesse, com  $n > 0$ .

Segundo Cuminato e Junior (2011), as aproximações serão dadas no formato

$$y^{(q)} = F(x, h) + E(x, h), \quad (59)$$

em que  $F(x, h)$  é um quociente de diferenças para aproximação da derivada de ordem  $q$  de uma função  $y$  com erro de truncamento  $E(x, h)$ . Diz-se que o termo  $F(x, h)$  é de ordem  $p$  se  $E(x, y) = h^p R(x)$ , onde  $R(x)$  não depende de  $h$ . Nesse caso usa a notação  $E(x, h) = O(h^p)$ . Essa notação significa que  $\lim_{h \rightarrow 0} E(x, h)/h^p$  é uma constante finita.

A derivada de maior ordem utilizada nos cálculos, dado por

$$u(x_0 + h, y_0 + k) = \sum_{p=0}^{\infty} \left[ \frac{1}{p!} \left[ \sum_{q=0}^p \left[ \frac{p!}{q!(p-q)!} h^q k^{p-q} \frac{\partial^p f}{\partial x^q \partial y^{p-q}}(x_0, y_0) \right] \right] \right] \quad (60)$$

Após algumas manipulações algébricas, a combinação adequada das expansões de  $u(x_i \pm h, y_j - k)$  e  $u(x_i \pm h, y_j + k)$ , até termos de ordem  $n$ , obtém-se as fórmulas de diferenças centradas

$$\frac{\partial u}{\partial x}(x_0, y_0) = \frac{u(x_0 + h, y_0) - u(x_0 - h, y_0)}{2h} + O(h^2) \quad (61)$$

$$\frac{\partial u}{\partial y}(x_i, y_j) = \frac{u(x_0, y_0 + k) - u(x_0, y_0 - k)}{2k} + O(k^2), \quad (62)$$

$$\frac{\partial^2 u}{\partial x^2}(x_0, y_0) = \frac{u(x_0 + h, y_0) - 2u(x_0, y_0) + u(x_0 - h, y_0)}{h^2} + O(h^2), \quad (63)$$

$$\frac{\partial^2 u}{\partial y^2}(x_0, y_0) = \frac{u(x_0, y_0 + k) - 2u(x_0, y_0) + u(x_0, y_0 - k)}{k^2} + O(k^2), \quad (64)$$

$$\frac{\partial^2 u}{\partial x \partial y}(x_0, y_0) = \frac{u(x_0 + h, y_0 + k) + u(x_0 - h, y_0 - k) - u(x_0 - h, y_0 + k) - u(x_0 + h, y_0 - k)}{4hk} + O(h^2 + k^2), \quad (65)$$

$$\frac{\partial^3 u}{\partial x^3}(x_i, y_j) = \frac{u(x_0 + 2h, y_0) - u(x_0 - 2h, y_0) - 2u(x_0 + h, y_0) + 2u(x_0 - h, y_0)}{2h^3} + O(h^2), \quad (66)$$

$$\frac{\partial^3 u}{\partial y^3}(x_i, y_j) = \frac{u(x_0, y_0 + 2k) - u(x_0, y_0 - 2k) - 2u(x_0, y_0 + k) + 2u(x_0, y_0 - k)}{2k^3} + O(k^2), \quad (67)$$

$$\frac{\partial^4 u}{\partial x^4}(x_i, y_j) = \frac{u(x_0 + 2h, y_0) + u(x_0 - 2h, y_0) - 4u(x_0 + h, y_0) - 4u(x_0 - h, y_0) + 6u(x_0, y_0)}{h^4} + O(h^2), \quad (68)$$

$$\frac{\partial^4 u}{\partial y^4}(x_i, y_j) = \frac{u(x_0, y_0 + 2k) + u(x_0, y_0 - 2k) - 4u(x_0, y_0 + k) - 4u(x_0, y_0 - k) + 6u(x_0, y_0)}{k^4} + O(k^2), \quad (69)$$

e

$$\frac{\partial^4 u}{\partial x^2 \partial y^2}(x_i, y_j) = \frac{u(x_0 + h, y_0 + k) + u(x_0 + h, y_0 - k) + u(x_0 - h, y_0 + k) + u(x_0 - h, y_0 - k) - 2u(x_0 + h, y_0) - 2u(x_0 - h, y_0) - 2u(x_0, y_0 + k) - 2u(x_0, y_0 - k)}{h^2 k^2} + O(h^2 + k^2). \quad (70)$$

Após obter esses quocientes de diferenças e substituir nas EDs, de acordo com Burden e Faires (2010), ao variar o ponto em torno do qual a expansão é feita obtém-se um sistema de equações algébricas, da forma

$$\mathbf{Ax} = \mathbf{b}, \quad (71)$$

em que  $A$  é uma matriz dos coeficientes,  $\mathbf{b}$  é a matriz coluna dos termos independentes e  $\mathbf{x}$  é a matriz coluna dos termos desconhecidos. Para a obtenção da solução desse sistema, há uma variedade de métodos a serem empregados, que serão explicados no Capítulo 4.4.

#### 4.4 Métodos iterativos de solução de sistemas de equações lineares

Conforme foi definido, após substituir os quocientes de diferenças na EDP, um sistema de equações lineares é formado, sendo

$$\sum_{j=1}^n a_{i,j}x_j = b_i, \quad 1 \leq i \leq n \quad (72)$$

ou ainda da forma matricial,

$$\mathbf{Ax} = \mathbf{b}, \quad (73)$$

em que  $A = [a_{i,j}]$ , uma matriz quadrada de ordem  $n$  pertencente aos  $IR^n$ ,  $\mathbf{b}$  é uma vetor coluna também pertencente aos  $IR^n$ . Conforme define Varga (2009), o vetor solução  $\mathbf{x}$  é único e existente em  $IR^n$  se e somente se  $A$  for não singular. De acordo com Burden e Faires (2010), uma matriz é dita não singular quando  $A^{-1}$  existe e que  $AA^{-1} = A^{-1}A = I$ , sendo  $I$  a matriz identidade. Supondo que  $A$  seja não singular e que seus termos da diagonal principal  $[a_{i,i}]$  sejam diferentes de zero, pode-se definir o vetor  $\mathbf{x}$  explicitamente da seguinte maneira

$$\mathbf{x} = A^{-1}\mathbf{b}. \quad (74)$$

Pode-se expressar a matriz  $A$  como sendo a soma

$$A = D - L - U \quad (75)$$

em que  $D = \text{diag}\{a_{1,1}; a_{2,2}; a_{3,3}; \dots; a_{n,n}\}$ ,  $L$  é uma matriz triangular estritamente inferior e  $U$  é uma matriz triangular estritamente superior. As matrizes  $L$  e  $U$  possuem como entrada os valores negativos da matriz  $A$ , sendo respectivamente, os termos inferiores e

superiores à diagonal principal. Com a equação (75) é possível escrever a equação (73) da seguinte maneira

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}, \quad (76)$$

A seguir, serão descritos os métodos numéricos de Jacobi e Gauss-Seidel.

#### 4.4.1 Método de Jacobi

Conforme Ortega (1932), o método de Jacobi é o método iterativo mais simples. Sendo que os termos da matriz  $D$  são diferentes de zero, ou seja

$$a_{i,i} \neq 0 \quad \forall \quad i = 1, \dots, n$$

pode-se escrever a equação (76)

$$a_{ii}x_i = - \sum_{j=1; j \neq i}^n a_{ij}x_j + b_i, \quad \forall \quad i = 1, \dots, n \quad (77)$$

Assumindo que o vetor iteração  $\mathbf{x}^k$  foi computado, têm-se que a próxima iteração  $\mathbf{x}^{k+1}$  será

$$a_{ii}x_i^{k+1} = - \sum_{j=1; j \neq i}^n a_{ij}x_j^k + b_i, \quad i = 1, \dots, n \quad \text{com } k \geq 0, \quad (78)$$

ou ainda,

$$x_i^{k+1} = \frac{1}{a_{i,i}} \left( - \sum_{j=1; j \neq i}^n a_{ij}x_j^k \right) + \frac{b_i}{a_{i,i}}, \quad i = 1, \dots, n \quad \text{com } k \geq 0, \quad (79)$$

em notação matricial, têm-se

$$\mathbf{x}^{k+1} = D^{-1}(L + U)\mathbf{x}^k + D^{-1}\mathbf{b}, \quad \text{com } k \geq 0 \quad (80)$$

pode-se denominar  $T_J = D^{-1}(L + U)$  e  $c_J = D^{-1}\mathbf{b}$ , em que

$$\mathbf{x}^{k+1} = T_J\mathbf{x}^k + c_J, \quad \text{com } k \geq 0, \quad (81)$$

sendo  $T_J$  a matriz de iteração do método de Jacobi.

#### 4.4.2 Método de Gauss-Seidel

Examinando o método de Jacobi dado pela equação (79), pode-se verificar que é necessário salvar os termos  $\mathbf{x}^k$  enquanto se computa as componentes do vetor  $\mathbf{x}^{k+1}$ . Logo,

seria interessante utilizar os termos já computados  $x_i^{k+1}$  das componentes  $x_i$  do vetor solução  $\mathbf{x}$  em todas as subseqüentes iterações (VARGA, 2009).

Dessa maneira, conforme segue em Burden e Faires (2010), têm-se a seguinte equação

$$x_i^{k+1} = \frac{1}{a_{ii}} \left[ - \sum_{j=1}^{i-1} (a_{ij}x_j^{k+1}) - \sum_{j=i+1}^n (a_{ij}x_j^k) + b_i \right], \quad \text{com } k \geq 0, \quad (82)$$

para todo  $i = 1, 2, \dots, n$ , é a modificação feita da equação (79), denominada método iterativo de Gauss-Seidel.

Com a equação (82) não é necessário arquivar simultaneamente as duas aproximações  $x_i^{k+1}$  e  $x_i^k$  no processo computacional. Pode-se escrever tal equação da forma matricial

$$(D - L)\mathbf{x}^{k+1} = U\mathbf{x}^k + \mathbf{b}, \quad \text{com } k \geq 0, \quad (83)$$

ou ainda

$$\mathbf{x}^{k+1} = (D - L)^{-1}U\mathbf{x}^k + (D - L)^{-1}\mathbf{b}, \quad \text{com } k \geq 0, \quad (84)$$

denominando  $T_{gs} = (D - L)^{-1}U$  e  $c_{gs} = (D - L)^{-1}\mathbf{b}$ , tem-se

$$\mathbf{x}^{k+1} = T_{gs}\mathbf{x}^k + c_{gs}, \quad \text{com } k \geq 0, \quad (85)$$

em que  $T_{gs}$  é a matriz iteração do método de Gauss-Seidel.

#### 4.4.3 Critérios de convergência

As iterações dadas pelas equações (81) e (85), podem ser escritas da forma geral

$$\mathbf{x}^{k+1} = T\mathbf{x}^k + c, \quad k \geq 0, \quad (86)$$

em que  $T$  é a matriz de iteração e  $c$  a matriz dos termos independentes. Conforme Varga (2009), a equação  $\mathbf{x}^* = H\mathbf{x}^* + c$ , sendo  $\mathbf{x}^*$  a solução única, é verdadeira se e somente se  $A\mathbf{x}^* = b$ .

Em Ortega (1932), para que o método iterativo convirja para qualquer valor inicial de entrada  $\mathbf{x}^0$ ,  $\rho(T) < 1$ . O termo  $\rho(T)$  diz respeito ao raio espectral da matriz de iteração  $T$ , em que

$$\rho(T) = \max\{|\lambda_i|\} \quad (87)$$

em que  $\lambda$  é o vetor de autovalores da matriz  $T$ .

A afirmação descrita anteriormente é verdadeira pois, assumindo que  $T \in L(\mathbb{R}^n)$  e que a equação  $\mathbf{x} = T\mathbf{x} + c$  possui a solução única  $\mathbf{x}^*$ , subtraindo  $\mathbf{x}^* = T\mathbf{x}^* + c$  da equação (86), têm-se

$$\mathbf{x}^{k+1} - \mathbf{x}^* = T(\mathbf{x}^k - \mathbf{x}^*) = \dots = T^{k+1}(\mathbf{x}^0 - \mathbf{x}^*) \quad (88)$$

sendo que na última iteração, o  $\lim_{k \rightarrow \infty}(\mathbf{x}^k - \mathbf{x}^*) = 0$ . Para que se utilize qualquer valor de  $\mathbf{x}^0$ , é suficiente que  $\lim_{k \rightarrow \infty}(T^k) = 0$  (ORTEGA, 1932).

Conforme Franco (2006), para que o método de Gauss-Seidel convirja, é suficiente que o critério de Sassenfeld seja satisfeito, isto é, se:

$$\max_{1 \leq i \leq n} \beta_i \leq 1 \quad (89)$$

onde  $\beta_i$  são calculados por recorrência através de

$$\beta_i = \sum_{j=1}^{i-1} |a_{ij}^*| \beta_j + \sum_{j=i+1}^n |a_{ij}^*|. \quad (90)$$



## 5 Metodologia

A solução numérica da equação de deflexão de placas obtida no Capítulo 4.2, será obtida utilizando o método numérico descrito no Capítulo 4.3. Para tanto, retoma-se a equação (54), ou seja,:

$$D_x \frac{\partial^4 u}{\partial x^4}(x, y) + 2H \frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y) + D_y \frac{\partial^4 u}{\partial y^4}(x, y) = p(x, y).$$

em  $\Omega = \{(x, y) | a < x < b, c < y < d\}$ .

Conforme descrito no Capítulo 4.3, para aplicação do método, o primeiro passo é realizar a discretização do domínio, ou seja, é necessário selecionar os números inteiros  $M$  e  $N$  e definir os tamanhos de passo  $h = (b - a)/M$  e  $k = (d - c)/N$ . Em seguida, deve-se realizar a partição do intervalo  $[a, b]$  em  $M$  parte iguais de largura  $h$  e do intervalo  $[c, d]$  em  $N$  parte iguais de largura  $k$ .

$$x_i = a + (i - 1)h, \quad \forall \quad i = 1, 2, 3, \dots, M + 1$$

e

$$y_j = c + (j - 1)k, \quad \forall \quad j = 1, 2, 3, \dots, N + 1.$$

As extremidades desses subintervalos são os pontos da malha, onde  $x_1 = a$ ,  $x_{M+1} = b$ ,  $y_1 = c$  e  $y_{N+1} = d$ , conforme a Figura 14.

Como próximo passo, deve-se realizar as aproximações das derivadas conforme demonstrando no Capítulo 4.3, e realiza-se a substituição das equações (68), (70) e (69) na equação (54), assim

$$\begin{aligned} & D_x \left[ \frac{u_{i+2,j} + u_{i-2,j} - 4u_{i+1,j} - 4u_{i-1,j} + 6u_{i,j}}{h^4} - \frac{h^2}{3} \frac{\partial^6 u}{\partial x^6}(\xi_{21}, y_j) \right] \\ & + 2H \left[ \frac{u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}}{h^2 k^2} \right. \\ & \quad \left. - \frac{2u_{i+1,j} - 2u_{i-1,j} - 2u_{i,j+1} - 2u_{i,j-1}}{h^2 k^2} \right. \\ & \quad \left. - k^2 \frac{\partial^6 u}{\partial x^2 \partial y^4}(\xi_{21}, \eta_8) - \frac{h^2}{3} \frac{\partial^6 u}{\partial x^4 \partial y^2}(\xi_{21}, \eta_8) \right] \\ & + D_y \left[ \frac{u_{i,j+2} + u_{i,j-2} - 4u_{i,j+1} - 4u_{i,j-1} + 6u_{i,j}}{k^4} - \frac{k^2}{3} \frac{\partial^6 u}{\partial y^6}(x_i, \eta_{22}) \right] = p(x, y), \end{aligned} \quad (91)$$

ao juntar os termos em comum, chega-se na seguinte equação

$$\begin{aligned}
& \frac{D_x}{h^4} u_{i+2,j} + \frac{D_x}{h^4} u_{i-2,j} - 4 \left( \frac{D_x}{h^4} + \frac{H}{h^2 k^2} \right) u_{i+1,j} \\
& - 4 \left( \frac{D_x}{h^4} + \frac{H}{h^2 k^2} \right) u_{i-1,j} + \left( \frac{6D_x}{h^4} + \frac{8H}{h^2 k^2} + \frac{6D_y}{k^4} \right) u_{i,j} \\
& + \frac{2H}{h^2 k^2} u_{i+1,j+1} + \frac{2H}{h^2 k^2} u_{i+1,j-1} + \frac{2H}{h^2 k^2} u_{i-1,j+1} \\
& + \frac{2H}{h^2 k^2} u_{i-1,j-1} - 4 \left( \frac{H}{h^2 k^2} + \frac{D_y}{k^4} \right) u_{i,j+1} - 4 \left( \frac{H}{h^2 k^2} + \frac{D_y}{k^4} \right) u_{i,j-1} \\
& + \frac{D_y}{k^4} u_{i,j+2} + \frac{D_y}{k^4} u_{i,j-2} - D_x \frac{h^2}{3} \frac{\partial^6 u}{\partial x^6} (\xi_{21}, y_j) \\
& - 2Hk^2 \frac{\partial^6 u}{\partial x^2 \partial y^4} (\xi_{21}, \eta_8) - 2H \frac{h^2}{3} \frac{\partial^6 u}{\partial x^4 \partial y^2} (\xi_{21}, \eta_8) - D_y \frac{k^2}{3} \frac{\partial^6 u}{\partial y^6} (x_i, \eta_{22}) = p(x, y)
\end{aligned} \tag{92}$$

Agora se faz necessário substituir os valores da equação  $u(x_i, y_j)$ , pelos valores  $w_{i,j}$ , de tal maneira que  $u(x_i, y_j) \approx w_{i,j}$ . Isso possibilita que os termos do erro sejam desprezados, que será somente avaliado após os valores de  $w_{i,j}$  forem obtidos nos pontos considerados. Ao multiplicar ambos os lado da equação (92) por  $h^4 k^4$ , tem-se

$$\begin{aligned}
& D_x k^4 w_{i+2,j} + D_x k^4 w_{i-2,j} - 4(D_x k^4 + H h^2 k^2) w_{i+1,j} \\
& - 4(D_x k^4 + H h^2 k^2) w_{i-1,j} + (6D_x k^4 + 8H h^2 k^2 + 6D_y h^4) w_{i,j} \\
& + 2H h^2 k^2 w_{i+1,j+1} + 2H h^2 k^2 w_{i+1,j-1} + 2H h^2 k^2 w_{i-1,j+1} \\
& + 2H h^2 k^2 w_{i-1,j-1} - 4(D_y h^4 + H h^2 k^2) w_{i,j+1} \\
& - 4(D_y h^4 + H h^2 k^2) w_{i,j-1} + D_y h^4 w_{i,j+2} + D_y h^4 w_{i,j-2} = h^4 k^4 p(x, y)
\end{aligned} \tag{93}$$

denominando  $A = D_x k^4 x k^4$ ,  $B = -4(D_x k^4 + H h^2 k^2)$ ,  $C = (6D_x k^4 + 8H h^2 k^2 + 6D_y h^4)$ ,  $D = 2H h^2 k^2$ ,  $E = -4(D_y h^4 + H h^2 k^2)$ ,  $F = D_y h^4$ , a equação (93) pode ser escrita da seguinte forma

$$\begin{aligned}
& Aw_{i+2,j} + Aw_{i-2,j} + Bw_{i+1,j} + Bw_{i-1,j} \\
& + Cw_{i,j} + Dw_{i+1,j+1} + Dw_{i+1,j-1} + Dw_{i-1,j+1} \\
& + Dw_{i-1,j-1} + Ew_{i,j+1} + Ew_{i,j-1} + Fw_{i,j+2} + Fw_{i,j-2} = h^4 k^4 p(x, y)
\end{aligned} \tag{94}$$

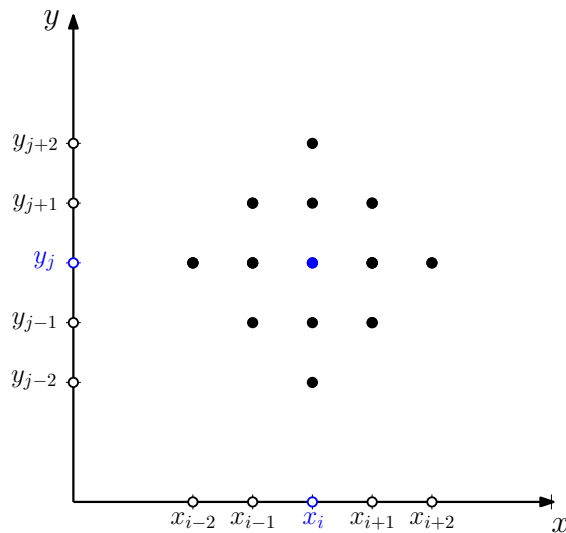
A equação (94) é denominada equação de diferenças da equação (54), para todo  $i = 1, \dots, M + 1$  e  $j = 1, \dots, N + 1$ .

A equação (94) envolve aproximações a  $u(x_i, y_j)$  nos pontos

$$\begin{aligned}
& (x_i, y_{j+2}), \\
& (x_{i-1}, y_{j+1}), (x_i, y_{j+1}), (x_{i+1}, y_{j+1}), \\
& (x_{i-2}, y_j), (x_{i-1}, y_j), (x_i, y_j), (x_{i+1}, y_j), (x_{i+2}, y_j), \\
& (x_{i-1}, y_{j-1}), (x_i, y_{j-1}), (x_{i+1}, y_{j-1}), \\
& (x_i, y_{j-2}),
\end{aligned} \tag{95}$$

Ao localizar estes pontos na malha, percebe-se que para cada aproximação, utiliza-se pontos de uma região apresentada na Figura 15, tendo como centro o ponto  $(x_i, y_j)$ .

Figura 15 – Molécula do MDF para a EDP dada pela equação (54)

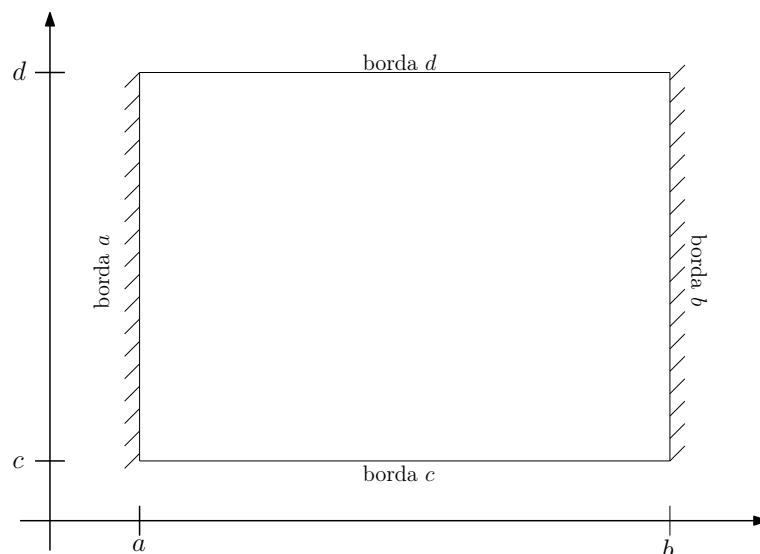


Fonte: Autoria própria (2018).

Primeiramente, denomina-se a borda com os pontos  $(x_1, y_j)$  como sendo a borda  $a$  e os pontos  $(x_{M+1}, y_j)$  como sendo a borda  $b$ , para todo  $j = 1, \dots, N + 1$ , assim como para todos os pontos de  $(x_i, y_1)$  como sendo a borda  $c$  e os pontos de  $(x_i, y_{N+1})$  como sendo a borda  $d$ , para todo  $i = 1, \dots, M + 1$ .

Considera-se que as bordas  $a$  e  $b$  estão apoiadas, assim como as bordas  $c$  e  $d$  estão livres, conforme a Figura 16.

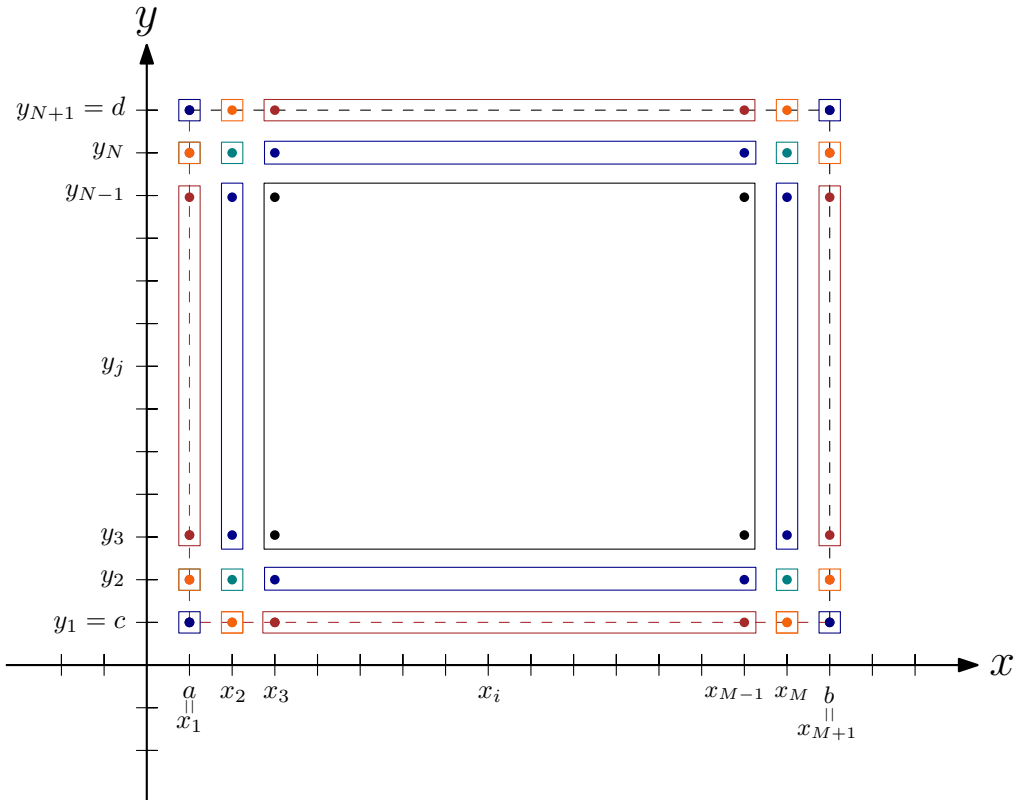
Figura 16 – Posicionamento da placa no plano cartesiano.



Fonte: Autoria própria (2018).

Com a molécula que está representada pela Figura 15, percorre-se todos os pontos do domínio. Ao realizar tal ação, verifica-se que há certas regiões em que a molécula necessitará de pontos não pertencentes ao domínio  $\Omega$ , denominados pontos fantasmas. Na Figura 17 há a representação dos pontos semelhantes na quantidade de pontos fantasmas por diferentes colorações.

Figura 17 – Quantidade de pontos semelhantes



Fonte: Autoria própria (2018).

A Figura 17 dispõe um sistema de  $(M + 1).(N + 1)$  equações, porém com  $(M + 5).(N + 5) - 12$  incógnitas, sendo assim um sistema indeterminado. Para que esse sistema torne-se possível e o problema dos pontos fantasmas seja corrigido, as condições de contorno de placas bi-apoiadas apresentadas no Capítulo 4.2 devem ser introduzidas, de modo que acrescente mais equações ao problema.

Pelo fato da borda  $a$  estar apoiada, a deflexão nesta borda é nula, ou seja,

$$u(a, y) = 0, \quad (96)$$

ainda que o momento fletor  $M_x$  é nulo, ou seja,

$$M_x = \frac{\partial^2 u}{\partial x^2}(a, y) = 0. \quad (97)$$

Ao aplicar a expansões em série de Taylor e aproximar a derivada segunda conforme a equação (63), onde  $u(a, y) \approx w_{1,j}$ , tem-se que

$$w_{1,j} = 0 \quad (98)$$

e

$$M_x = \frac{w_{2,j} - 2w_{1,j} + w_{0,j}}{h^2} = 0. \quad (99)$$

Devido a equação (98), pode-se escrever a equação (99) da seguinte maneira:

$$w_{0,j} = -w_{2,j}. \quad (100)$$

Da mesma forma que a borda  $a$ , a borda  $b$  está apoiada, sendo assim, a deflexão nesta borda também é nula,

$$u(b, y) = 0, \quad (101)$$

e o momento fletor  $M_x$  é nulo, ou seja,

$$M_x = \frac{\partial^2 u}{\partial x^2}(b, y) = 0. \quad (102)$$

Ao aplicar a expansões em série de Taylor e aproximar a derivada segunda conforme descrito pela equação (63), além que  $u(b, y) \approx w_{M+1,j}$ , tem-se que

$$w_{M+1,j} = 0 \quad (103)$$

e

$$M_x = \frac{w_{M+2,j} - 2w_{M+1,j} + w_{M,j}}{h^2} = 0. \quad (104)$$

Devido a equação (103), pode-se escrever a equação (104) da seguinte maneira:

$$w_{M+2,j} = -w_{M,j}. \quad (105)$$

Sendo que a borda  $c$  está livre, a deflexão em seus pontos não serão nulos, porém, conforme descrito no Capítulo 4.2,

$$M_y = \frac{\partial^2 u}{\partial y^2}(x, c) = 0 \quad \text{e} \quad Q_y = \frac{\partial^3 u}{\partial y^3}(x, c) = 0. \quad (106)$$

Ao aplicar a expansões em série de Taylor e aproximar as derivadas conforme descrito nas equações (64) e (67), onde  $u(x, c) \approx w_{j,1}$ , tem-se que

$$\frac{w_{i,2} - 2w_{i,1} + w_{i,0}}{k^2} = 0 \quad (107)$$

e

$$\frac{w_{i,3} - w_{i,-1} - 2w_{i,2} + 2w_{i,0}}{2k^3} = 0, \quad (108)$$

ao isolar o termo  $w_{i,0}$  da equação (107), e substituir o mesmo na equação (108), tem-se

$$w_{i,0} = 2w_{i,1} - w_{i,2} \quad (109)$$

e

$$\begin{aligned} w_{i,3} - w_{i,-1} - 2w_{i,2} + 2w_{i,0} &= 0 \\ w_{i,3} - w_{i,-1} - 2w_{i,2} + 2(2w_{i,1} - w_{i,2}) &= 0 \\ w_{i,3} - w_{i,-1} - 4w_{i,2} + 4w_{i,1} &= 0, \end{aligned} \quad (110)$$

isola-se o termo  $w_{i,-1}$ , encontra-se

$$w_{i,-1} = w_{i,3} - 4w_{i,2} + 4w_{i,1}. \quad (111)$$

Conforme descrito para a borda  $c$ , a borda  $d$  também está livre,

$$M_y = \frac{\partial^2 u}{\partial y^2}(x, d) = 0 \quad \text{e} \quad Q_y = \frac{\partial^3 u}{\partial y^3}(x, d) = 0, \quad (112)$$

ao aplicar as expansões em série de Taylor e aproximar as derivadas conforme descrito nas equações (64) e (67), além que  $u(x, d) \approx w_{j,N+1}$ , temos que

$$M_y = \frac{w_{i,N+2} - 2w_{i,N+1} + w_{i,N}}{k^2} = 0 \quad (113)$$

e

$$Q_y = \frac{w_{i,N+3} - w_{i,N-1} - 2w_{i,N+2} + 2w_{i,N}}{2k^3} = 0. \quad (114)$$

Ao isolar o termo  $w_{i,N+2}$  da equação (113), e substituir o mesmo na equação (114), tem-se

$$w_{i,N+2} = 2w_{i,N+1} - w_{i,N} \quad (115)$$

e

$$\begin{aligned} w_{i,N+3} - w_{i,N-1} - 2w_{i,N+2} + 2w_{i,N} &= 0 \\ w_{i,N+3} - w_{i,N-1} - 2(2w_{i,N+1} - w_{i,N}) + 2w_{i,N} &= 0 \\ w_{i,N+3} - w_{i,N-1} - 4w_{i,N+1} + 4w_{i,N} &= 0. \end{aligned} \quad (116)$$

Ao isolar o termo  $w_{i,N+3}$ , obtém-se

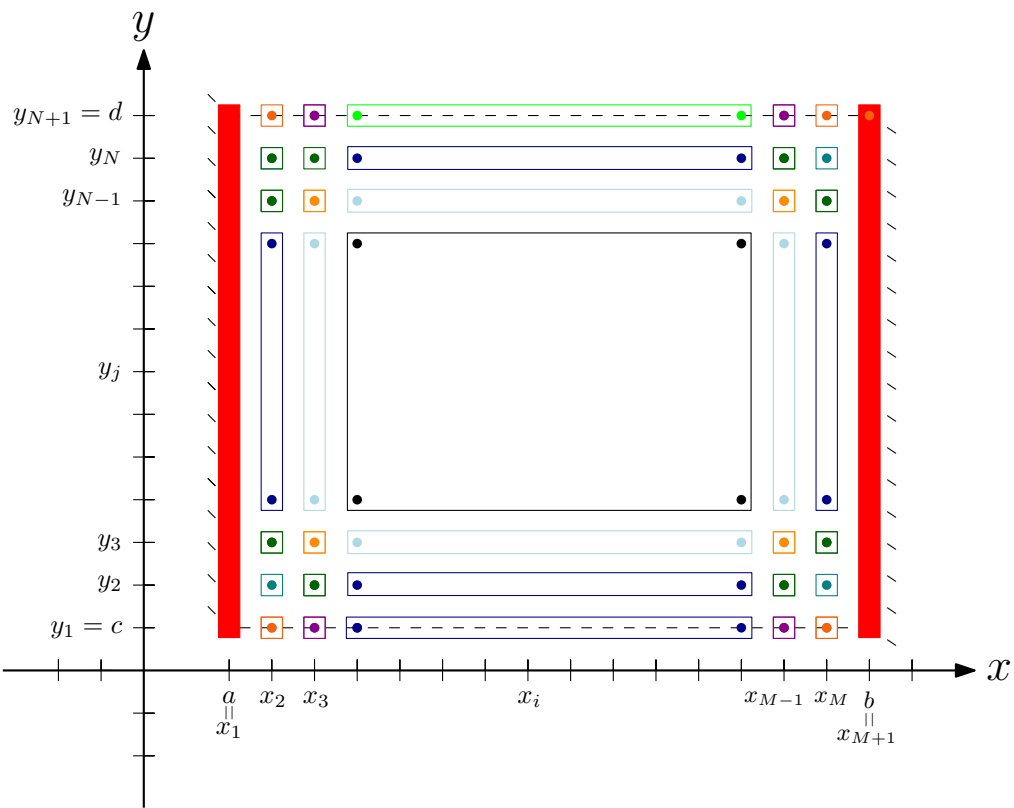
$$w_{i,N+3} = w_{i,N-1} + 4w_{i,N+1} - 4w_{i,N}. \quad (117)$$

Após obter essas equações auxiliares, é necessário aplica-las no sistema para torna-lo possível de se obter a solução. Como que, para aplicar o MDF, o domínio foi discretizado,

será encontrado uma equação para cada ponto representado na malha da Figura 18. Tais pontos foram divididos de acordo com a quantidade de pontos fantasmas e pontos de bordas, que eles apresentavam.

Os pontos representados em vermelhos na Figura 18 são os que a deflexão é nula devido a condição de contorno de borda apoiada, por tal motivo, ao retirar tais pontos do sistema, o número de incógnitas diminui, passando a ser  $(N + 5).(M + 1)$ , porém com  $(M - 1).(N + 1)$  equações, sendo assim, ainda um sistema indeterminado.

Figura 18 – Pontos semelhantes em uma placa bi-apoiada

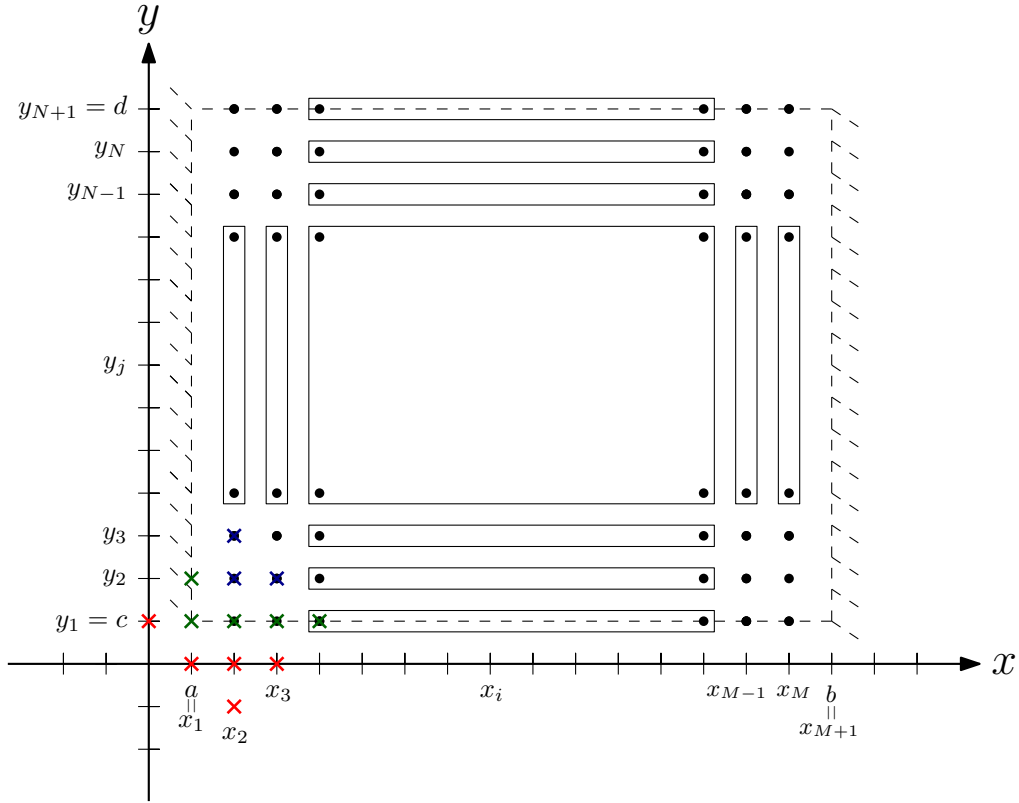


Fonte: Autoria própria (2018).

Pela Figura 18, ainda é possível verificar que as incógnitas estão contidas na região em que  $2 \leq i \leq M - 1$  e  $1 \leq j \leq N + 1$ . Sendo assim, as condições de contorno serão aplicadas nos seguintes pontos:

### 5.1 Considerando $i = 2$ e $j = 1$

Ao considerar  $i = 2$  e  $j = 1$  e posicionar molécula no domínio  $\Omega$ , percebe-se que há cinco pontos fora do mesmo, conforme apresenta a Figura 19.

Figura 19 – Molécula posicionada no ponto  $i = 2$  e  $j = 1$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = 1$  na equação (94),

$$\begin{aligned}
 & Aw_{4,1} + Aw_{0,1} + Bw_{3,1} + Bw_{1,1} \\
 & + Cw_{2,1} + Dw_{3,2} + Dw_{3,0} + Dw_{1,2} \\
 & + Dw_{1,0} + Ew_{2,2} + Ew_{2,0} + Fw_{2,3} + Fw_{2,-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{118}$$

A molécula irá ser influenciada pelas condições de contorno que as bordas  $a$  e  $c$  estão submetidas, sendo assim, as equações (109) e (111) referentes à borda  $c$  são verdadeiras, assim como as equações (98) e (100) referentes à borda  $a$ , portanto, têm-se

$$\begin{aligned}
 & Aw_{4,1} + A(-w_{2,1}) + Bw_{3,1} + Cw_{2,1} + Dw_{3,2} + D(2w_{3,1} - w_{3,2}) \\
 & + Ew_{2,2} + E(2w_{2,1} - w_{2,2}) + Fw_{2,3} + F(w_{2,3} - 4w_{2,2} + 4w_{2,1}) = h^4 k^4 p(x, y),
 \end{aligned} \tag{119}$$

reorganizando os termos,

$$\begin{aligned}
 & Aw_{4,1} + (B + 2D)w_{3,1} + (C - A + 2E + 4F)w_{2,1} \\
 & + 2Fw_{2,3} - 4Fw_{2,2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{120}$$

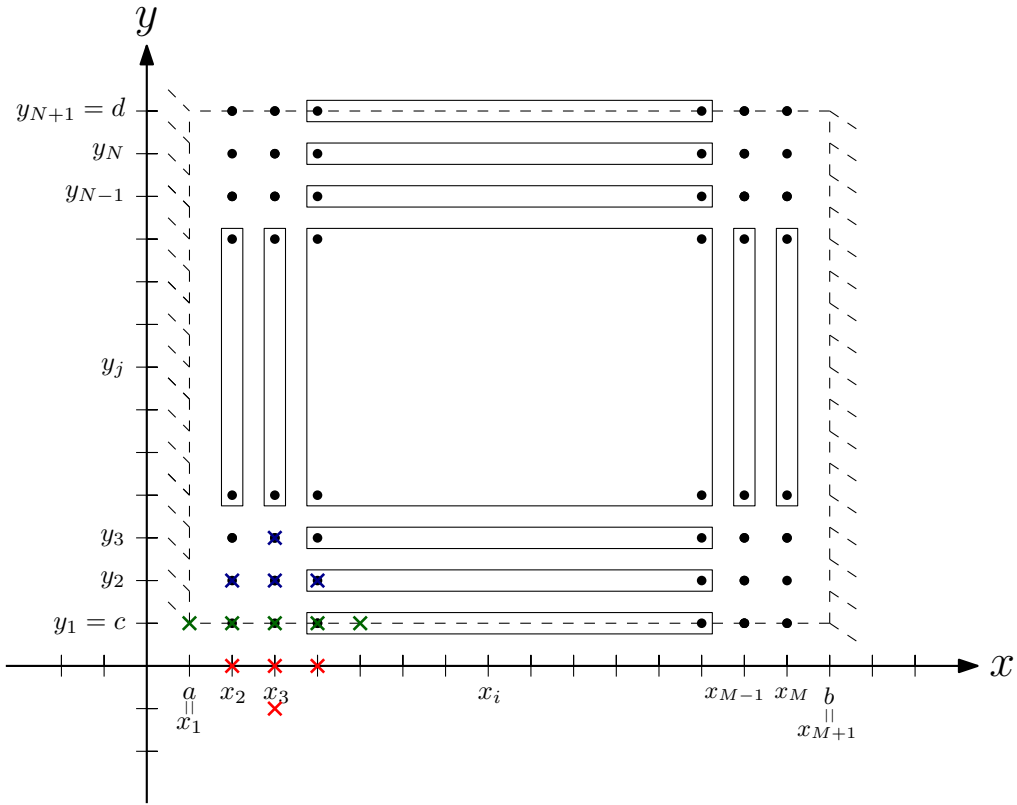
em que a equação (120) representa a aproximação para o ponto  $u(x_2, y_1)$ .



## 5.2 Considerando $i = 3$ e $j = 1$

Ao considerar  $i = 3$  e  $j = 1$  e posicionar a molécula no domínio  $\Omega$ , percebe-se que há quatro pontos fora do mesmo, conforme apresenta a Figura 20.

Figura 20 – Molécula posicionada no ponto  $i = 3$  e  $j = 1$



Fonte: Autoria própria (2018).

Observa-se este fato também se nota ao aplicar  $i = 3$  e  $j = 1$  na equação (94),

$$\begin{aligned}
 & Aw_{5,1} + Aw_{1,1} + Bw_{4,1} + Bw_{2,1} \\
 & + Cw_{3,1} + Dw_{4,2} + Dw_{4,0} + Dw_{2,2} \\
 & + Dw_{2,0} + Ew_{3,2} + Ew_{3,0} + Fw_{3,3} + Fw_{3,-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{121}$$

Novamente, a molécula irá ser influenciada pelas condições de contorno que as bordas  $a$  e  $c$  estão submetidas, sendo assim,

$$\begin{aligned}
 & Aw_{5,1} + Bw_{4,1} + Bw_{2,1} \\
 & + Cw_{3,1} + Dw_{4,2} + D(2w_{4,1} - w_{4,2}) + Dw_{2,2} \\
 & + D(2w_{2,1} - w_{2,2}) + Ew_{3,2} + E(2w_{3,1} - w_{3,2}) \\
 & + Fw_{3,3} + F(w_{3,3} - 4w_{3,2} + 4w_{3,1}) = h^4 k^4 p(x, y)
 \end{aligned} \tag{122}$$

reorganizando os termos,

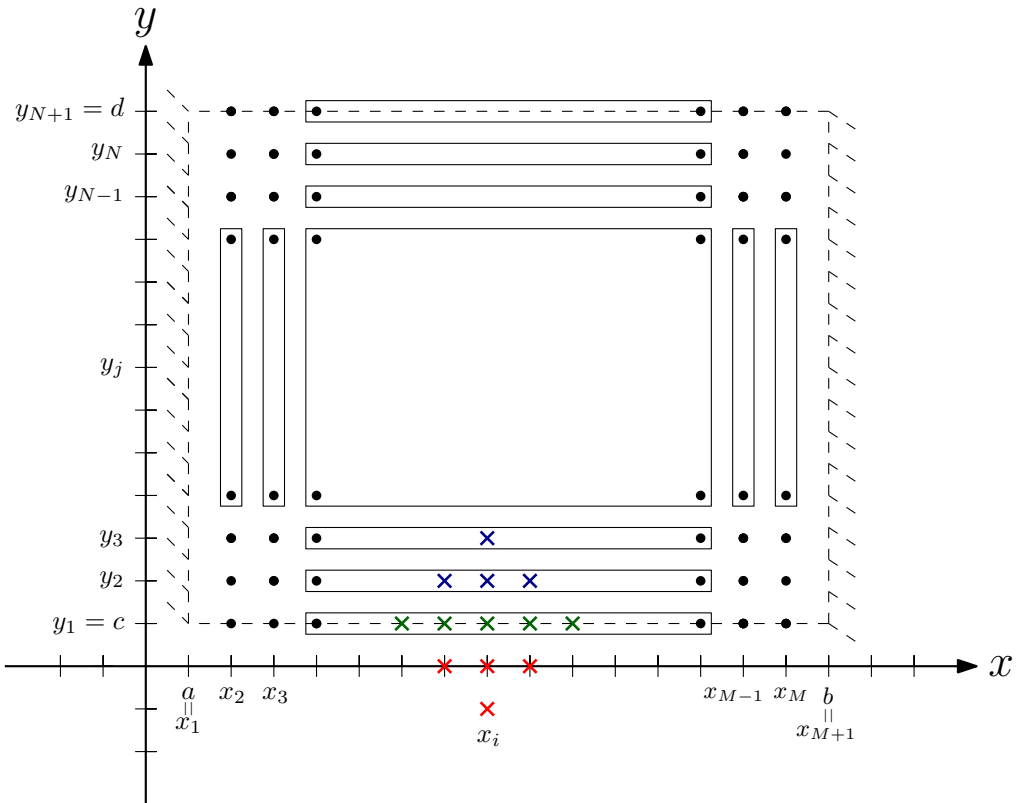
$$\begin{aligned} & Aw_{5,1} + (B + 2D)w_{4,1} + (B + 2D)w_{2,1} \\ & + (C + 2E + 4F)w_{3,1} + 2Fw_{3,3} - 4Fw_{3,2} = h^4k^4p(x, y) \end{aligned} \quad (123)$$

tem-se que a equação (123) representa a aproximação para o ponto  $u(x_3, y_1)$ .

### 5.3 Considerando $i = 4, \dots, M - 2$ e $j = 1$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = 1$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que esses apresentaram quatro pontos fora do mesmo, conforme apresenta a Figura 21.

Figura 21 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = 1$  na equação (94),

$$\begin{aligned} & Aw_{i+2,1} + Aw_{i-2,1} + Bw_{i+1,1} + Bw_{i-1,1} \\ & + Cw_{i,1} + Dw_{i+1,2} + Dw_{i+1,0} + Dw_{i-1,2} \\ & + Dw_{i-1,0} + Ew_{i,2} + Ew_{i,0} + Fw_{i,3} + Fw_{i,-1} = h^4k^4p(x, y), \end{aligned} \quad (124)$$

Desta vez, a molécula irá ser influenciada somente pelas condições de contorno que a borda  $c$  está submetida, sendo assim as equações (109) e (111) referentes á borda  $c$  são verdadeiras. Aplicando-as na equação (124), tem-se

$$\begin{aligned}
& Aw_{i+2,1} + Aw_{i-2,1} + Bw_{i+1,1} + Bw_{i-1,1} \\
& + Cw_{i,1} + Dw_{i+1,2} + D(2w_{i+1,1} - w_{i+1,2}) + Dw_{i-1,2} \\
& + D(2w_{i-1,1} - w_{i-1,2}) + Ew_{i,2} + E(2w_{i,1} - w_{i,2}) \\
& + Fw_{i,3} + F(w_{i,3} - 4w_{i,2} + 4w_{i,1}) = h^4 k^4 p(x, y),
\end{aligned} \tag{125}$$

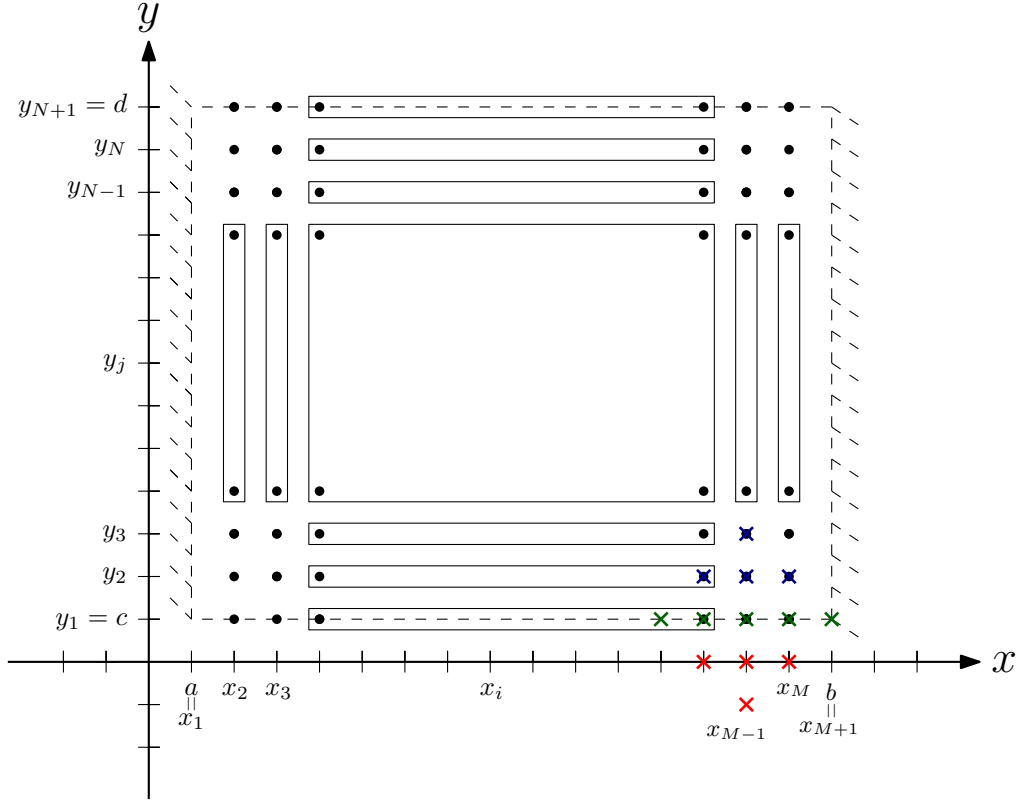
reorganizando os termos,

$$\begin{aligned}
& Aw_{i+2,1} + Aw_{i-2,1} + (B + 2D)w_{i+1,1} + (B + 2D)w_{i-1,1} \\
& + (C + 2E + 4F)w_{i,1} + 2Fw_{i,3} - 4Fw_{i,2} = h^4 k^4 p(x, y),
\end{aligned} \tag{126}$$

em que a equação (126) representa a aproximação os pontos  $u(x_i, y_1)$  para  $i = 4, \dots, M - 2$ .

#### 5.4 Considerando $i = M - 1$ e $j = 1$

Ao considerar  $i = M - 1$  e  $j = 1$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que esses apresentaram quatro pontos fora do mesmo, conforme apresenta a Figura 22.

Figura 22 – Molécula posicionada no ponto  $i = M - 1$  e  $j = 1$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,1} + Aw_{M-3,1} + Bw_{M,1} + Bw_{M-2,1} \\
 & + Cw_{M-1,1} + Dw_{M,2} + Dw_{M,0} + Dw_{M-2,2} \\
 & + Dw_{M-2,0} + Ew_{M-1,2} + Ew_{M-1,0} + Fw_{M-1,3} + Fw_{M-1,-1} = h^4 k^4 p(x, y).
 \end{aligned} \tag{127}$$

Neste ponto, a molécula será influenciada pelas condições de contorno que as bordas  $b$  e  $c$  estão submetidas, sendo assim, as equações (109) e (111) referentes à borda  $c$  são verdadeiras, assim como as equações (103) e (105) referentes à borda  $b$ . Aplicando-as na equação (127), tem-se

$$\begin{aligned}
 & Aw_{M-3,1} + Bw_{M,1} + Bw_{M-2,1} \\
 & + Cw_{M-1,1} + Dw_{M,2} + D(2w_{M,1} - w_{M,2}) + Dw_{M-2,2} \\
 & + D(2w_{M-2,1} - w_{M-2,2}) + Ew_{M-1,2} + E(2w_{M-1,1} - w_{M-1,2}) \\
 & + Fw_{M-1,3} + F(w_{M-1,3} - 4w_{M-1,2} + 4w_{M-1,1}) = h^4 k^4 p(x, y),
 \end{aligned} \tag{128}$$

reorganizando os termos,

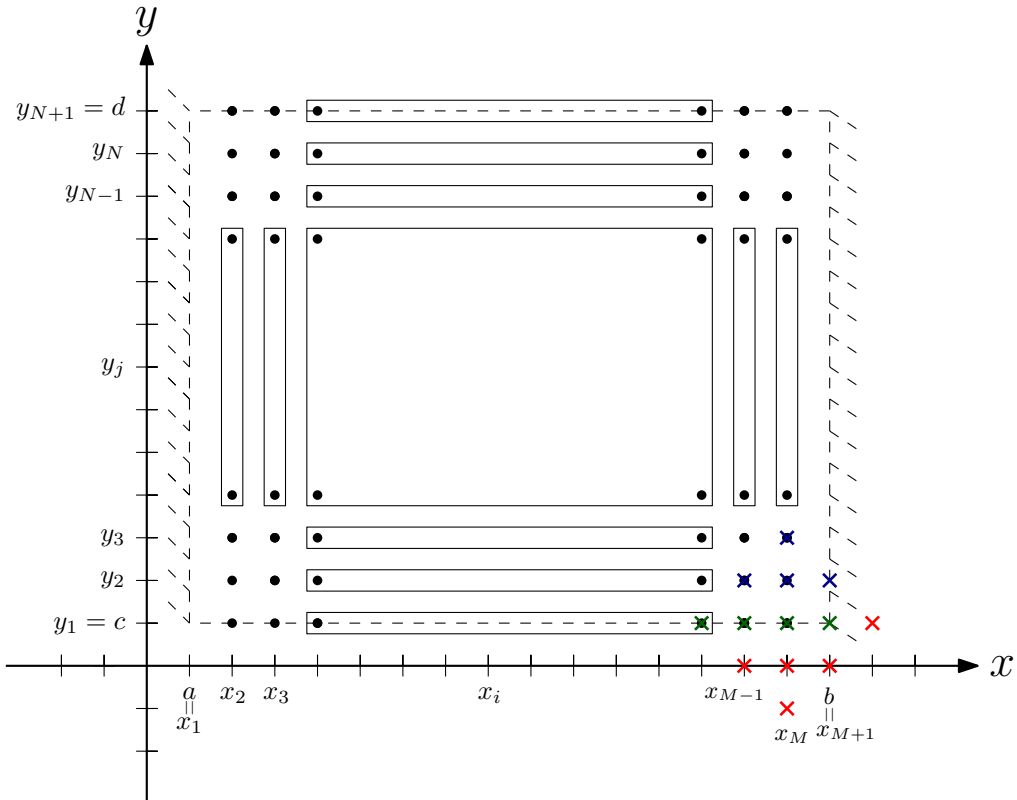
$$\begin{aligned}
 & Aw_{M-3,1} + (B + 2D)w_{M,1} + (B + 2D)w_{M-2,1} \\
 & + (C + 2E + 4F)w_{M-1,1} + 2Fw_{M-1,3} - 4Fw_{M-1,2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{129}$$

em que a equação (129) representa a aproximação do ponto  $u(x_{M-1}, y_1)$ .

### 5.5 Considerando $i = M$ e $j = 1$

Ao considerar  $i = M$  e  $j = 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta cinco pontos fora do mesmo, conforme apresenta a Figura 23.

Figura 23 – Molécula posicionada no ponto  $i = M$  e  $j = 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,1} + Aw_{M-2,1} + Bw_{M+1,1} + Bw_{M-1,1} \\
 & + Cw_{M,1} + Dw_{M+1,2} + Dw_{M+1,0} + Dw_{M-1,2} \\
 & + Dw_{M-1,0} + Ew_{M,2} + Ew_{M,0} + Fw_{M,3} + Fw_{M,-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{130}$$

Novamente a molécula será influenciada pelas condições de contorno que as bordas  $b$  e  $c$  estão submetidas, assim, tem-se que

$$\begin{aligned}
 & A(-w_{M,2}) + Aw_{M-2,1} + Bw_{M-1,1} \\
 & + Cw_{M,1} + Dw_{M-1,2} \\
 & + D(2w_{M-1,1} - w_{M-1,2}) + Ew_{M,2} + E(2w_{M,1} - w_{M,2}) \\
 & + Fw_{M,3} + F(w_{M,3} - 4w_{M,2} + 4w_{M,1}) = h^4 k^4 p(x, y),
 \end{aligned} \tag{131}$$

reorganizando os termos,

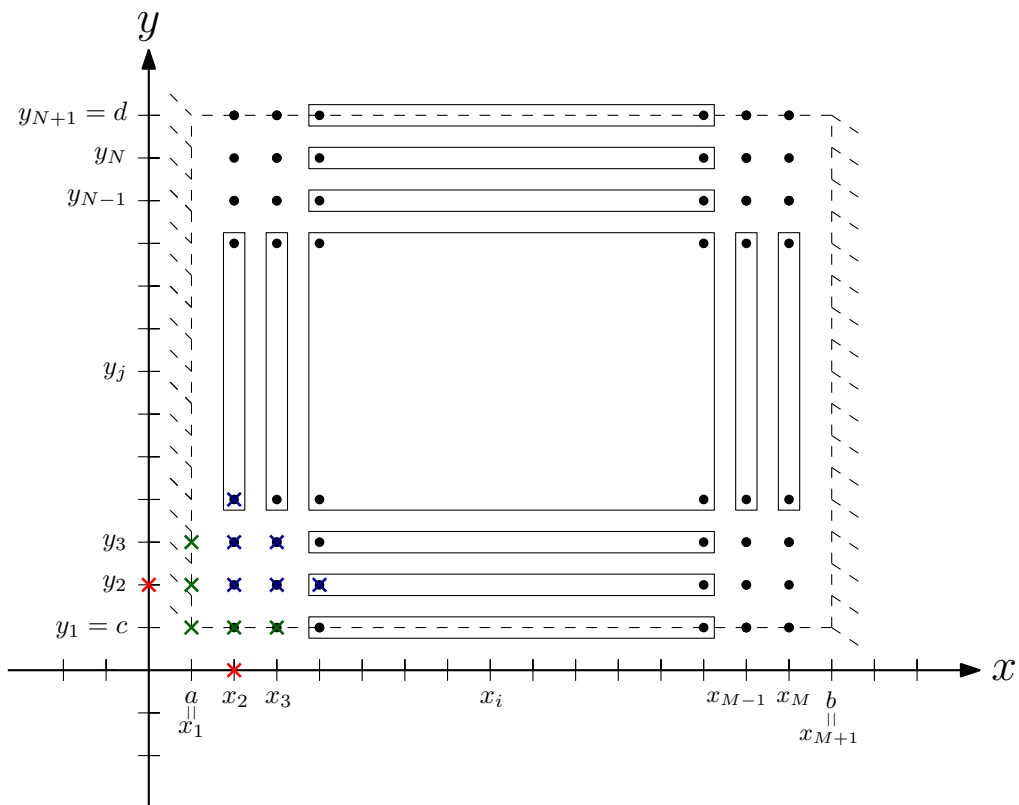
$$\begin{aligned}
 & +Aw_{M-2,1} + (B + 2D)w_{M-1,1} \\
 & +(C - A + 2E + 4F)w_{M,1} + 2Fw_{M,3} - 4Fw_{M,2} = h^4k^4p(x, y),
 \end{aligned} \tag{132}$$

em que a equação (132) representa a aproximação do ponto  $u(x_M, y_1)$ .

### 5.6 Considerando $i = 2$ e $j = 2$

Ao considerar  $i = 2$  e  $j = 2$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos fora do mesmo, conforme apresenta a Figura 24.

Figura 24 – Molécula posicionada no ponto  $i = 2$  e  $j = 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = 2$  na equação (94),

$$\begin{aligned}
 & Aw_{4,2} + Aw_{0,2} + Bw_{3,2} + Bw_{1,2} \\
 & +Cw_{2,2} + Dw_{3,3} + Dw_{3,1} + Dw_{1,3} \\
 & +Dw_{1,1} + Ew_{2,3} + Ew_{2,1} + Fw_{2,4} + Fw_{2,0} = h^4k^4p(x, y),
 \end{aligned} \tag{133}$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $c$ , por tal motivo as equações (98) e (100) referentes à borda  $a$  são verdadeiras, assim como as equações (109) e (111) referentes à borda  $c$ . Aplicando-as na equação (133), tem-se

$$\begin{aligned} & Aw_{4,2} + A(-w_{2,2}) + Bw_{3,2} \\ & + Cw_{2,2} + Dw_{3,3} + Dw_{3,1} \\ & + Ew_{2,3} + Ew_{2,1} + Fw_{2,4} + F(2w_{2,1} - w_{2,2}) = h^4 k^4 p(x, y), \end{aligned} \quad (134)$$

reorganizando os termos,

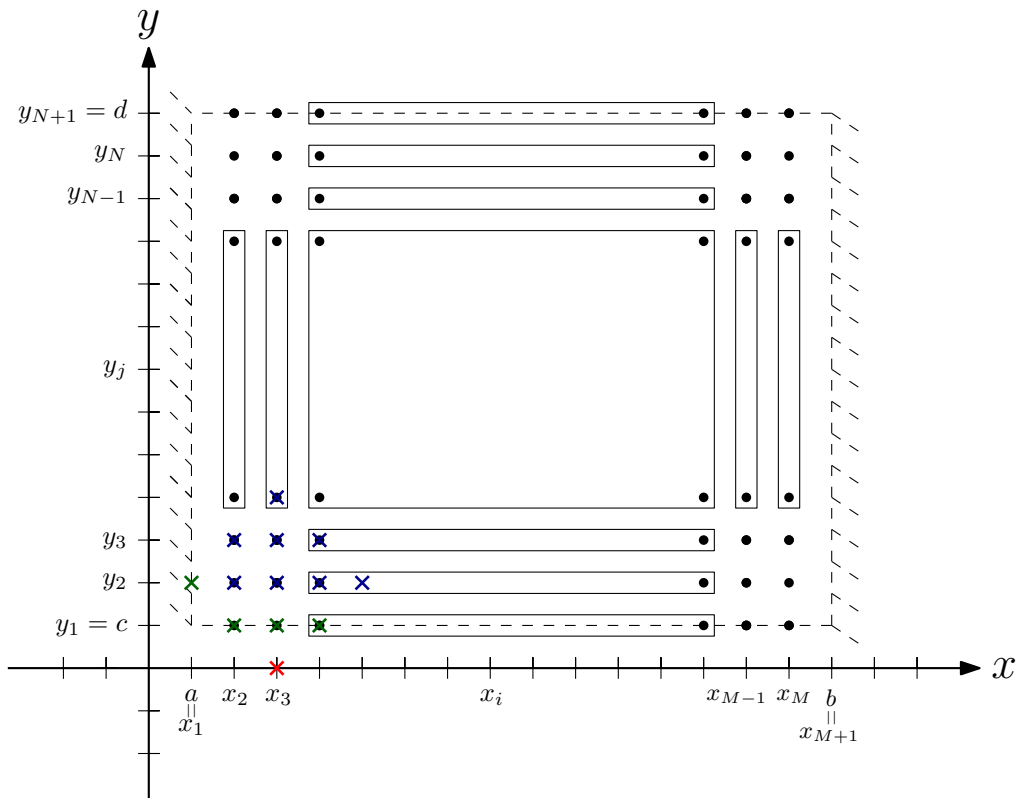
$$\begin{aligned} & Aw_{4,2} + Bw_{3,2} + (C - A - F)w_{2,2} \\ & + Dw_{3,3} + Dw_{3,1} + Ew_{2,3} + (E + 2F)w_{2,1} + Fw_{2,4} = h^4 k^4 p(x, y), \end{aligned} \quad (135)$$

em que a equação (135) representa a aproximação do ponto  $u(x_2, y_2)$ .

### 5.7 Considerando $i = 3$ e $j = 2$

Ao considerar  $i = 3$  e  $j = 2$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um único ponto fora do mesmo, conforme apresenta a Figura 25.

Figura 25 – Molécula posicionada no ponto  $i = 3$  e  $j = 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 3$  e  $j = 2$  na equação (94),

$$\begin{aligned}
& Aw_{5,2} + Aw_{1,2} + Bw_{4,2} + Bw_{2,2} \\
& + Cw_{3,2} + Dw_{4,3} + Dw_{4,1} + Dw_{2,3} \\
& + Dw_{2,1} + Ew_{3,3} + Ew_{3,1} + Fw_{3,4} + Fw_{3,0} = h^4 k^4 p(x, y),
\end{aligned} \tag{136}$$

A molécula, da mesma maneira, pega pontos da borda  $a$  e da borda  $c$ , sendo assim, tem-se

$$\begin{aligned}
& Aw_{5,2} + Bw_{4,2} + Bw_{2,2} \\
& + Cw_{3,2} + Dw_{4,3} + Dw_{4,1} + Dw_{2,3} \\
& + Dw_{2,1} + Ew_{3,3} + Ew_{3,1} + Fw_{3,4} + F(2w_{3,1} - w_{3,2}) = h^4 k^4 p(x, y),
\end{aligned} \tag{137}$$

reorganizando os termos,

$$\begin{aligned}
& Aw_{5,2} + Bw_{4,2} + Bw_{2,2} \\
& + (C - F)w_{3,2} + Dw_{4,3} + Dw_{4,1} + Dw_{2,3} \\
& + Dw_{2,1} + Ew_{3,3} + (E + 2F)w_{3,1} + Fw_{3,4} = h^4 k^4 p(x, y),
\end{aligned} \tag{138}$$

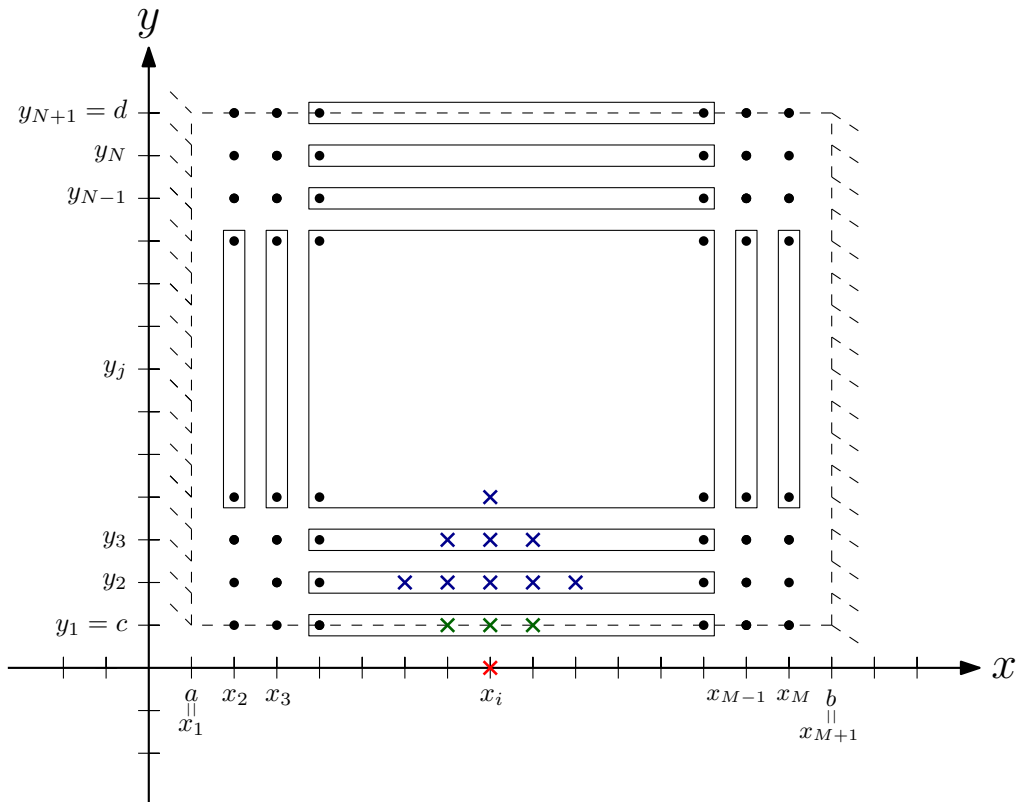
em que a equação (138) representa a aproximação do ponto  $u(x_3, y_2)$ .

### 5.8 Considerando $i = 4, \dots, M - 2$ e $j = 2$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = 2$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam somente um ponto fora do mesmo, conforme apresenta a Figura 26.



Figura 26 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = 2$  na equação (94),

$$\begin{aligned}
 & Aw_{i+2,2} + Aw_{i-2,2} + Bw_{i+1,2} + Bw_{i-1,2} \\
 & + Cw_{i,2} + Dw_{i+1,3} + Dw_{i+1,1} + Dw_{i-1,3} \\
 & + Dw_{i-1,1} + Ew_{i,3} + Ew_{i,1} + Fw_{i,4} + Fw_{i,0} = h^4 k^4 p(x, y),
 \end{aligned} \tag{139}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $c$  está submetida, por este motivo, as equações (107) e (109) referentes á borda  $c$  são verdadeiras. Aplicando-as na equação (139),

$$\begin{aligned}
 & Aw_{i+2,2} + Aw_{i-2,2} + Bw_{i+1,2} + Bw_{i-1,2} \\
 & + Cw_{i,2} + Dw_{i+1,3} + Dw_{i+1,1} + Dw_{i-1,3} \\
 & + Dw_{i-1,1} + Ew_{i,3} + Ew_{i,1} + Fw_{i,4} + F(2w_{i,1} - w_{i,2}) = h^4 k^4 p(x, y)
 \end{aligned} \tag{140}$$

reorganizando os termos,

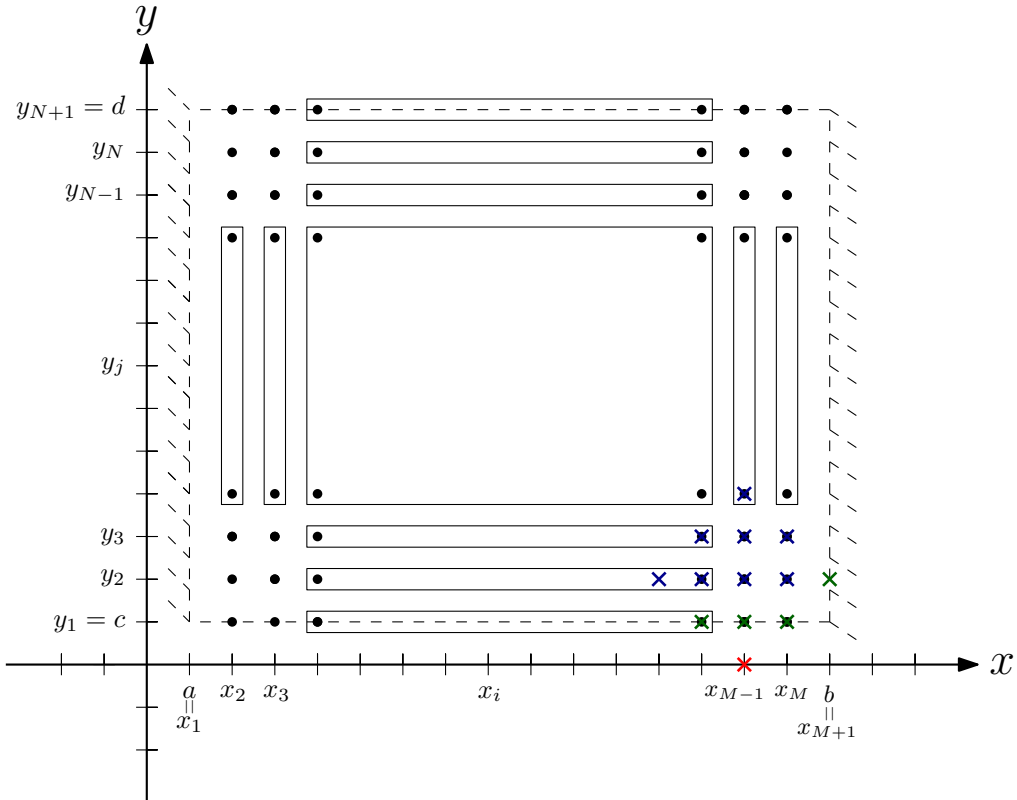
$$\begin{aligned}
 & Aw_{i+2,2} + Aw_{i-2,2} + Bw_{i+1,2} + Bw_{i-1,2} \\
 & + (C - F)w_{i,2} + Dw_{i+1,3} + Dw_{i+1,1} + Dw_{i-1,3} \\
 & + Dw_{i-1,1} + Ew_{i,3} + (E + 2F)w_{i,1} + Fw_{i,4} = h^4 k^4 p(x, y)
 \end{aligned} \tag{141}$$

em que a equação (141) representa as aproximações dos ponto  $u(x_i, y_2)$  para  $i = 4, \dots, M - 2$ .

### 5.9 Considerando $i = M - 1$ e $j = 2$

Ao considerar  $i = M - 1$  e  $j = 2$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 27.

Figura 27 – Molécula posicionada no ponto  $i = M - 1$  e  $j = 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = 2$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,2} + Aw_{M-3,2} + Bw_{M,2} + Bw_{M-2,2} \\
 & + Cw_{M-1,2} + Dw_{M,3} + Dw_{M,1} + Dw_{M-2,3} \\
 & + Dw_{M-2,1} + Ew_{M-1,3} + Ew_{M-1,1} + Fw_{M-1,4} + Fw_{M-1,0} = h^4 k^4 p(x, y),
 \end{aligned} \tag{142}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $c$ , por este motivo, as equações (103) e (105) referentes à borda  $b$  são verdadeiras, assim como as equações (109) e (111) referentes à borda  $c$ . Aplicando-as na equação (142),

$$\begin{aligned}
 & Aw_{M-3,2} + Bw_{M,2} + Bw_{M-2,2} \\
 & + Cw_{M-1,2} + Dw_{M,3} + Dw_{M,1} + Dw_{M-2,3} \\
 & + Dw_{M-2,1} + Ew_{M-1,3} + Ew_{M-1,1} \\
 & + Fw_{M-1,4} + F(2w_{M-1,1} - w_{M-1,2}) = h^4 k^4 p(x, y),
 \end{aligned} \tag{143}$$

reorganizando os termos,

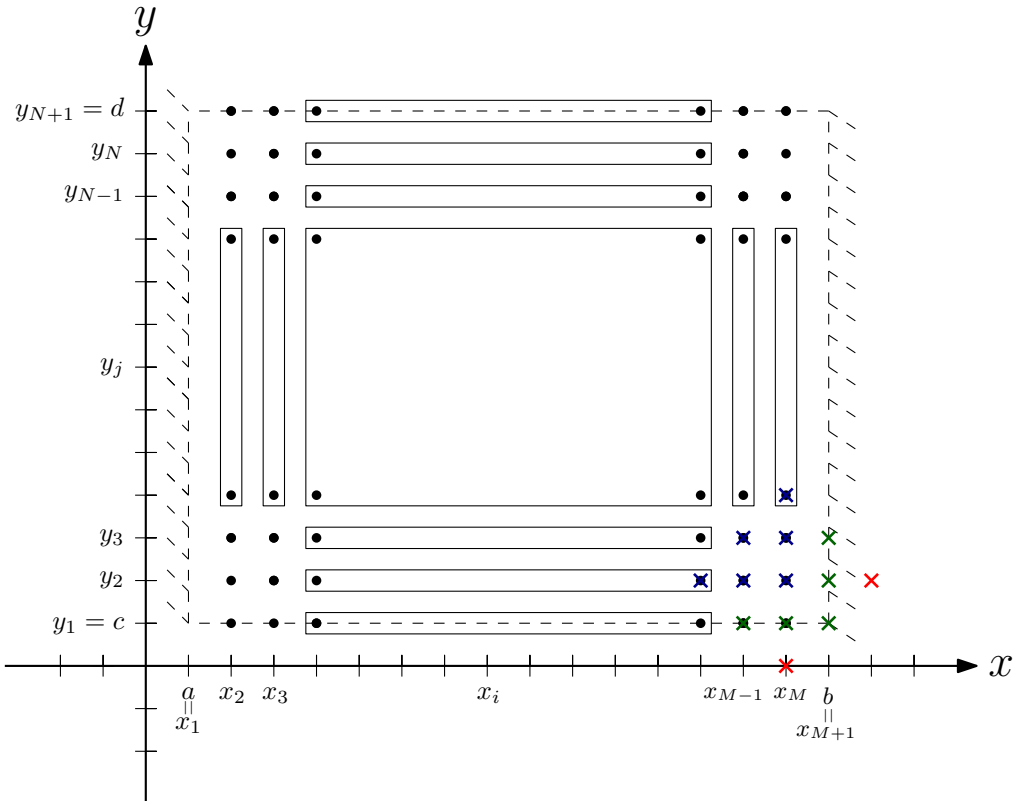
$$\begin{aligned}
 & Aw_{M-3,2} + Bw_{M,2} + Bw_{M-2,2} \\
 & + (C - F)w_{M-1,2} + Dw_{M,3} + Dw_{M,1} + Dw_{M-2,3} \\
 & + Dw_{M-2,1} + Ew_{M-1,3} + (E + 2F)w_{M-1,1} + Fw_{M-1,4} = h^4 k^4 p(x, y),
 \end{aligned} \tag{144}$$

em que a equação (144) representa a aproximação do ponto  $u(x_{M-1}, y_2)$ .

### 5.10 Considerando $i = M$ e $j = 2$

Ao considerar  $i = M$  e  $j = 2$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos fora do mesmo, conforme apresenta a Figura 28.

Figura 28 – Molécula posicionada no ponto  $i = M$  e  $j = 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = 2$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,2} + Aw_{M-2,2} + Bw_{M+1,2} + Bw_{M-1,2} \\
 & + Cw_{M,2} + Dw_{M+1,3} + Dw_{M+1,1} + Dw_{M-1,3} \\
 & + Dw_{M-1,1} + Ew_{M,3} + Ew_{M,1} + Fw_{M,4} + Fw_{M,0} = h^4 k^4 p(x, y),
 \end{aligned} \tag{145}$$

A molécula, neste caso, também estará sofrendo influência das condições de contorno da borda  $b$  e da borda  $c$ , assim, tem-se

$$\begin{aligned} & A(-w_{M,2}) + Aw_{M-2,2} + Bw_{M-1,2} + Cw_{M,2} + Dw_{M-1,3} \\ & + Dw_{M-1,1} + Ew_{M,3} + Ew_{M,1} + Fw_{M,4} + F(2w_{M,1} - w_{M,2}) = h^4 k^4 p(x, y), \end{aligned} \quad (146)$$

reorganizando os termos,

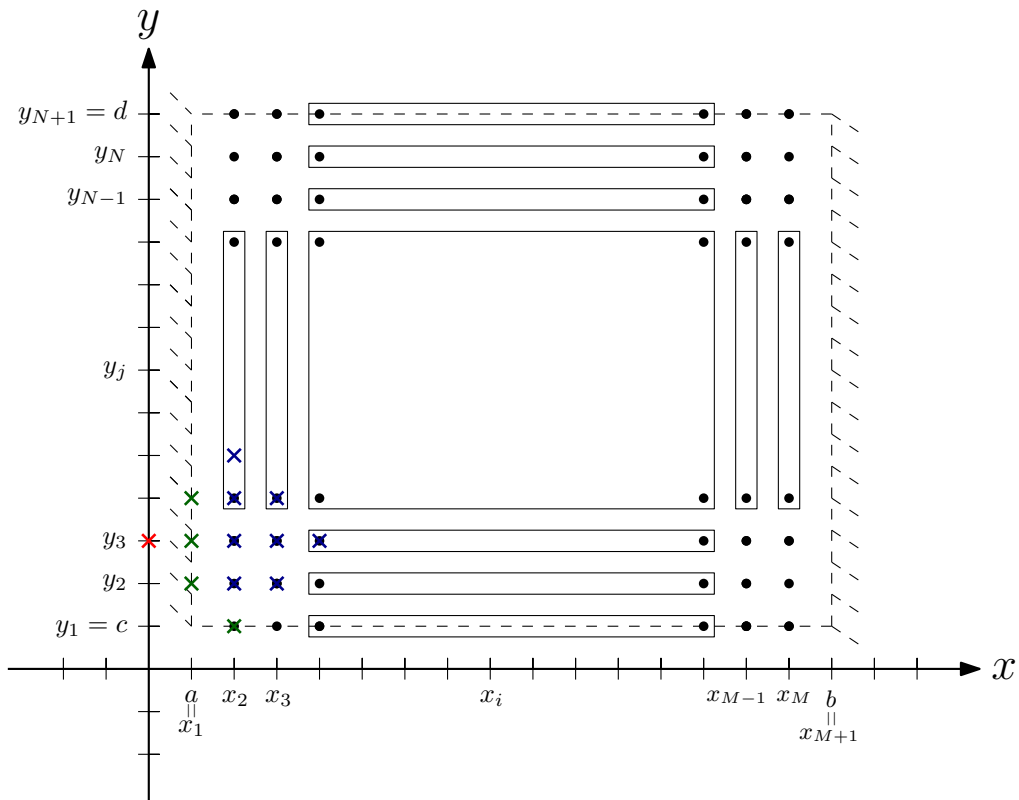
$$\begin{aligned} & Aw_{M-2,2} + Bw_{M-1,2} + (C - A - F)w_{M,2} + Dw_{M-1,3} \\ & + Dw_{M-1,1} + Ew_{M,3} + (E + 2F)w_{M,1} + Fw_{M,4} = h^4 k^4 p(x, y), \end{aligned} \quad (147)$$

em que a equação (147) representa a aproximação do ponto  $u(x_M, y_2)$ .

### 5.11 Considerando $i = 2$ e $j = 3$

Ao considerar  $i = 2$  e  $j = 3$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 29.

Figura 29 – Molécula posicionada no ponto  $i = 2$  e  $j = 3$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = 3$  na equação (94),

$$\begin{aligned}
& Aw_{4,3} + Aw_{0,3} + Bw_{3,3} + Bw_{1,3} \\
& + Cw_{2,3} + Dw_{3,4} + Dw_{3,2} + Dw_{1,4} \\
& + Dw_{1,2} + Ew_{2,4} + Ew_{2,2} + Fw_{2,5} + Fw_{2,1} = h^4 k^4 p(x, y),
\end{aligned} \tag{148}$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $c$ , por este motivo, as equações (98) e (100) referentes á borda  $a$  são verdadeiras, assim como as equações (109) e (111) referentes à borda  $c$ . Aplicando-as na equação (148), tem-se

$$\begin{aligned}
& Aw_{4,3} + A(-w_{2,3}) + Bw_{3,3} \\
& + Cw_{2,3} + Dw_{3,4} + Dw_{3,2} \\
& + Ew_{2,4} + Ew_{2,2} + Fw_{2,5} + Fw_{2,1} = h^4 k^4 p(x, y),
\end{aligned} \tag{149}$$

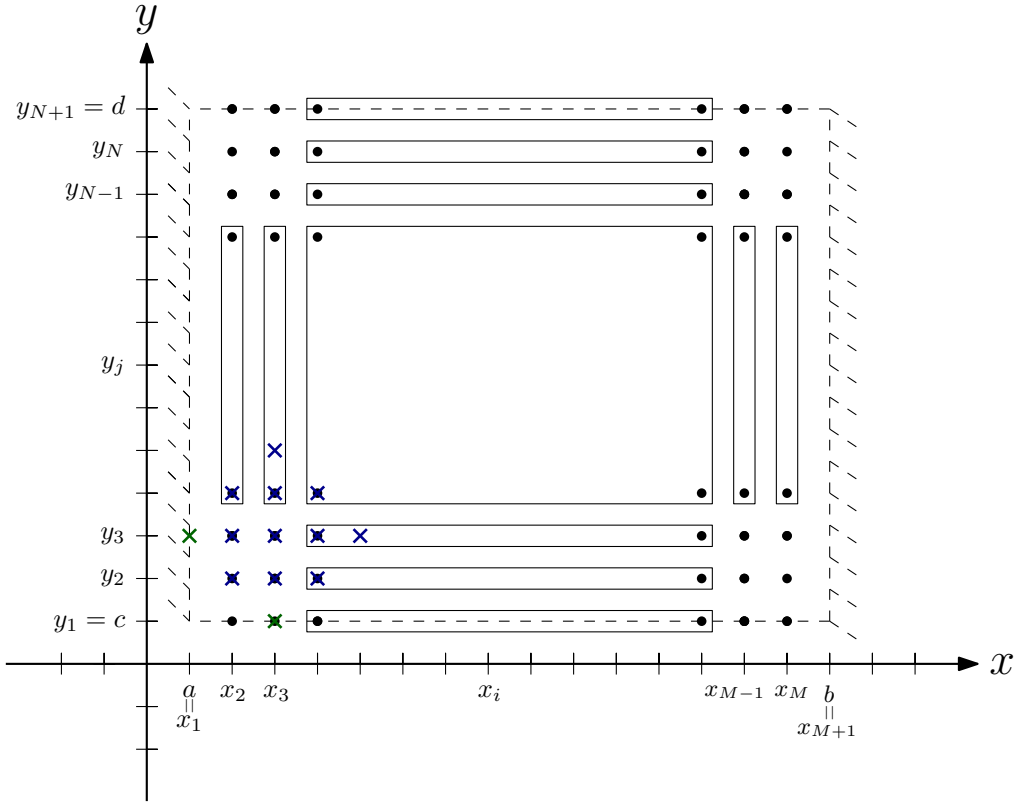
reorganizando os termos,

$$\begin{aligned}
& Aw_{4,3} + Bw_{3,3} + (C - A)w_{2,3} + Dw_{3,4} \\
& + Dw_{3,2} + Ew_{2,4} + Ew_{2,2} + Fw_{2,5} + Fw_{2,1} = h^4 k^4 p(x, y),
\end{aligned} \tag{150}$$

em que a equação (150) representa a aproximação do ponto  $u(x_2, y_3)$ .

### 5.12 Considerando $i = 3$ e $j = 3$

Ao considerar  $i = 3$  e  $j = 3$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta nenhum pontos fora do mesmo, porém apresenta dois pontos em contato com a borda, conforme apresenta a Figura 30.

Figura 30 – Molécula posicionada no ponto  $i = 3$  e  $j = 3$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 3$  e  $j = 3$  na equação (94),

$$\begin{aligned}
 & Aw_{5,3} + Aw_{1,3} + Bw_{4,3} + Bw_{2,3} \\
 & + Cw_{3,3} + Dw_{4,4} + Dw_{4,2} + Dw_{2,4} \\
 & + Dw_{2,2} + Ew_{3,4} + Ew_{3,2} + Fw_{3,5} + Fw_{3,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{151}$$

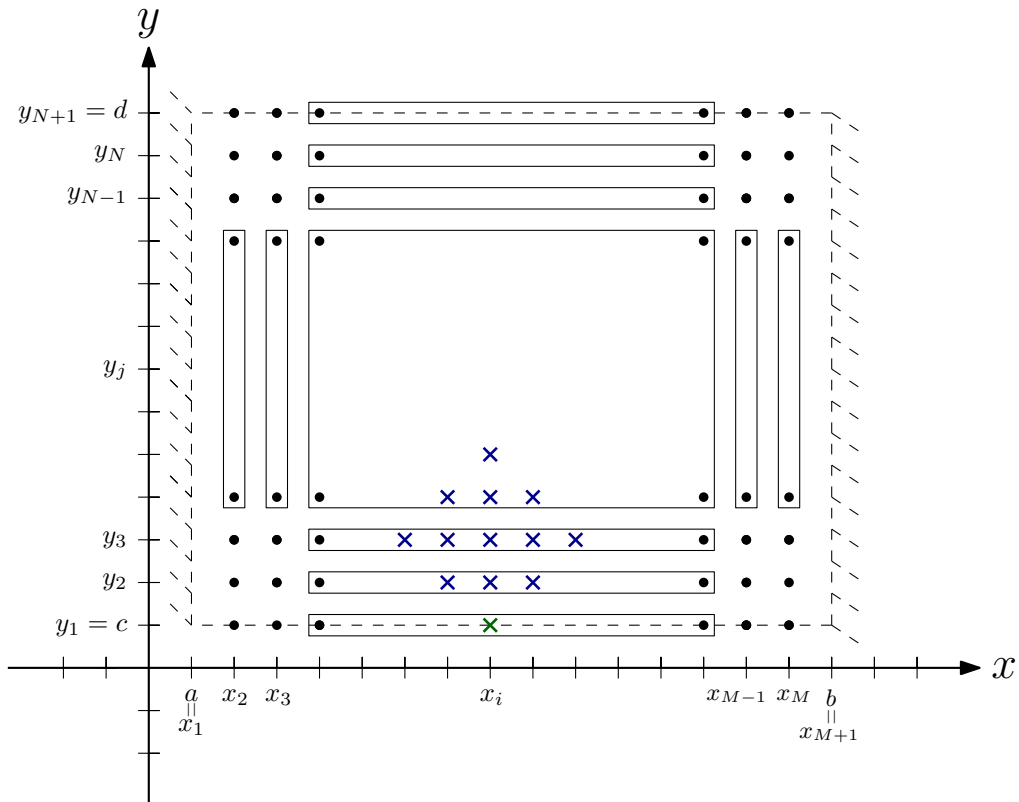
A molécula, neste caso, também possui pontos da borda  $a$  e da borda  $c$ , assim,

$$\begin{aligned}
 & Aw_{5,3} + Bw_{4,3} + Bw_{2,3} \\
 & + Cw_{3,3} + Dw_{4,4} + Dw_{4,2} + Dw_{2,4} \\
 & + Dw_{2,2} + Ew_{3,4} + Ew_{3,2} + Fw_{3,5} + Fw_{3,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{152}$$

em que a equação (152) representa a aproximação do ponto  $u(x_3, y_3)$ .

### 5.13 Considerando $i = 4, \dots, M - 2$ e $j = 3$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = 3$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam nenhum ponto fora do mesmo, porém apresentam, conforme apresenta a Figura 31, um ponto em contato com a borda  $c$ .

Figura 31 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = 3$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = 3$  na equação (94),

$$\begin{aligned}
 & Aw_{i+2,3} + Aw_{i-2,3} + Bw_{i+1,3} + Bw_{i-1,3} \\
 & + Cw_{i,3} + Dw_{i+1,4} + Dw_{i+1,2} + Dw_{i-1,4} \\
 & + Dw_{i-1,2} + Ew_{i,4} + Ew_{i,2} + Fw_{i,5} + Fw_{i,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{153}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $c$  está submetida, por este motivo,

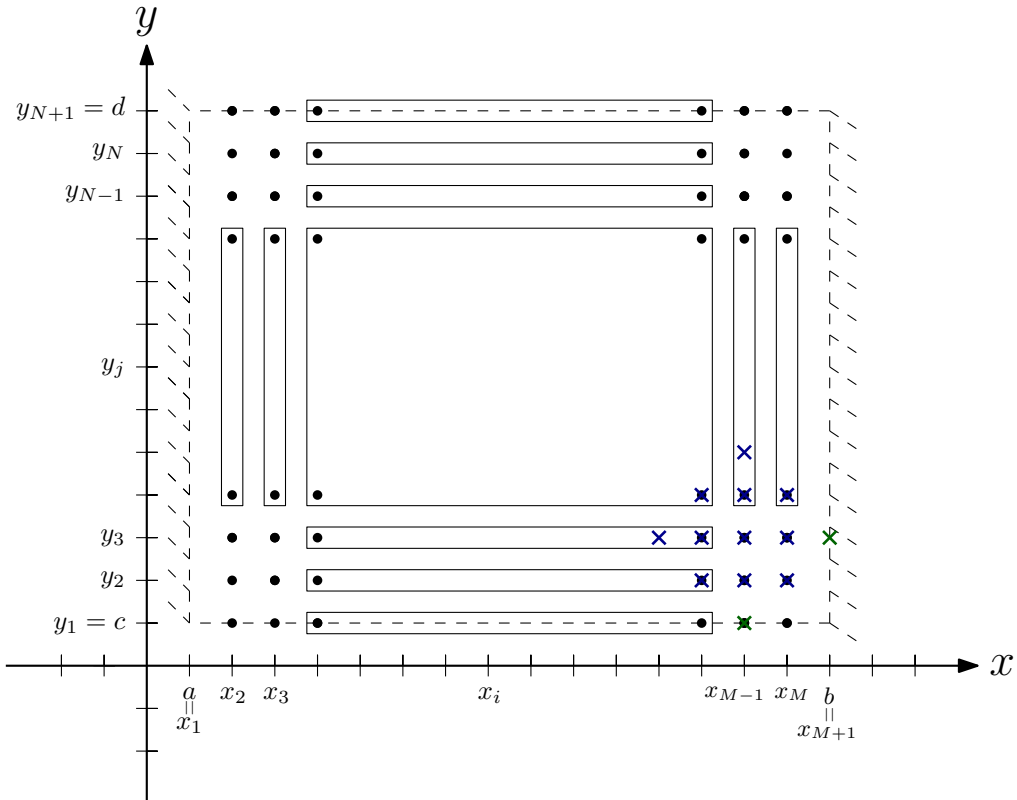
$$\begin{aligned}
 & Aw_{i+2,3} + Aw_{i-2,3} + Bw_{i+1,3} + Bw_{i-1,3} \\
 & + Cw_{i,3} + Dw_{i+1,4} + Dw_{i+1,2} + Dw_{i-1,4} \\
 & + Dw_{i-1,2} + Ew_{i,4} + Ew_{i,2} + Fw_{i,5} + Fw_{i,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{154}$$

em que a equação (154) representa a aproximação dos pontos  $u(x_i, y_3)$  para todo  $i = 4, \dots, M - 2$ .

5.14 Considerando  $i = M - 1$  e  $j = 3$

Ao considerar  $i = M - 1$  e  $j = 3$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta nenhum pontos fora do mesmo, porém, conforme apresenta a Figura 32, há dois pontos de borda.

Figura 32 – Molécula posicionada no ponto  $i = M - 1$  e  $j = 3$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = 3$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,3} + Aw_{M-3,3} + Bw_{M,3} + Bw_{M-2,3} \\
 & + Cw_{M-1,3} + Dw_{M,4} + Dw_{M,2} + Dw_{M-2,4} \\
 & + Dw_{M-2,2} + Ew_{M-1,4} + Ew_{M-1,2} + Fw_{M-1,5} + Fw_{M-1,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{155}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $c$ , tem-se

$$\begin{aligned}
 & Aw_{M-3,3} + Bw_{M,3} + Bw_{M-2,3} \\
 & + Cw_{M-1,3} + Dw_{M,4} + Dw_{M,2} + Dw_{M-2,4} \\
 & + Dw_{M-2,2} + Ew_{M-1,4} + Ew_{M-1,2} + Fw_{M-1,5} + Fw_{M-1,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{156}$$

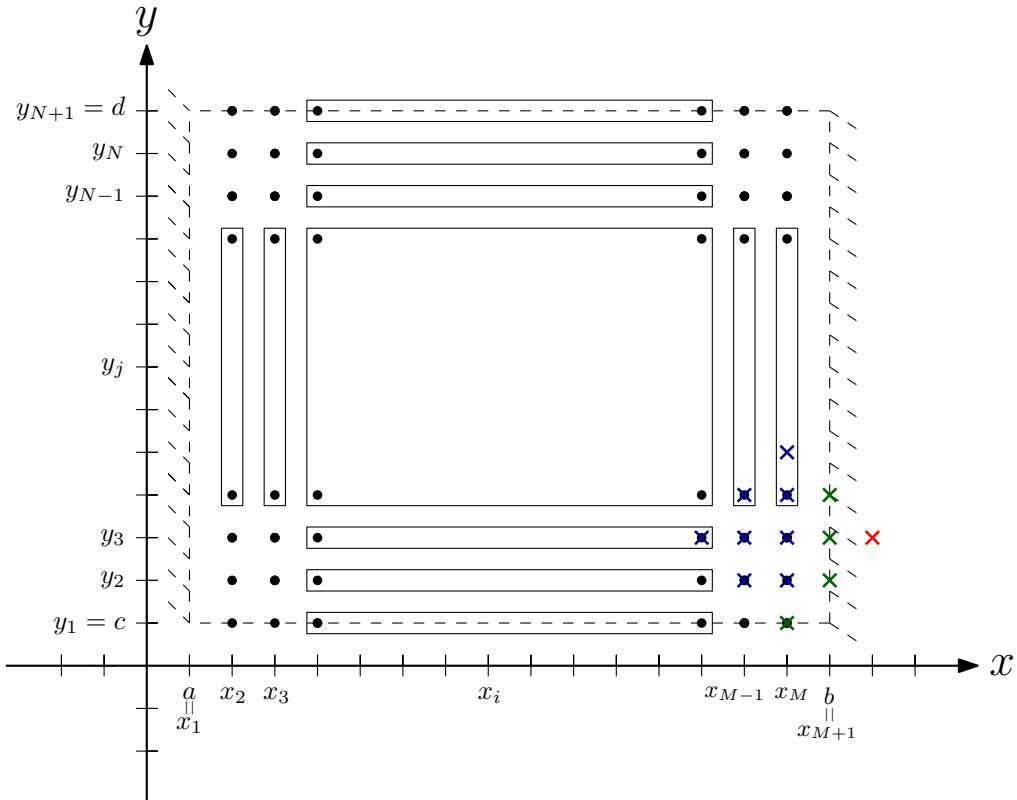
em que a equação (156) representa a aproximação do ponto  $u(x_{M-1}, y_3)$ .



5.15 Considerando  $i = M$  e  $j = 3$

Ao considerar  $i = M$  e  $j = 3$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 33.

Figura 33 – Molécula posicionada no ponto  $i = M$  e  $j = 3$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = 3$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,3} + Aw_{M-2,3} + Bw_{M+1,3} + Bw_{M-1,3} \\
 & + Cw_{M,3} + Dw_{M+1,4} + Dw_{M+1,2} + Dw_{M-1,4} \\
 & + Dw_{M-1,2} + Ew_{M,4} + Ew_{M,2} + Fw_{M,5} + Fw_{M,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{157}$$

A molécula, neste caso, também pega pontos da borda  $b$  e da borda  $c$ , por tal motivo, tem-se

$$\begin{aligned}
 & A(-w_{M,3}) + Aw_{M-2,3} + Bw_{M-1,3} \\
 & + Cw_{M,3} + Dw_{M-1,4} \\
 & + Dw_{M-1,2} + Ew_{M,4} + Ew_{M,2} + Fw_{M,5} + Fw_{M,1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{158}$$

reorganizando os termos,

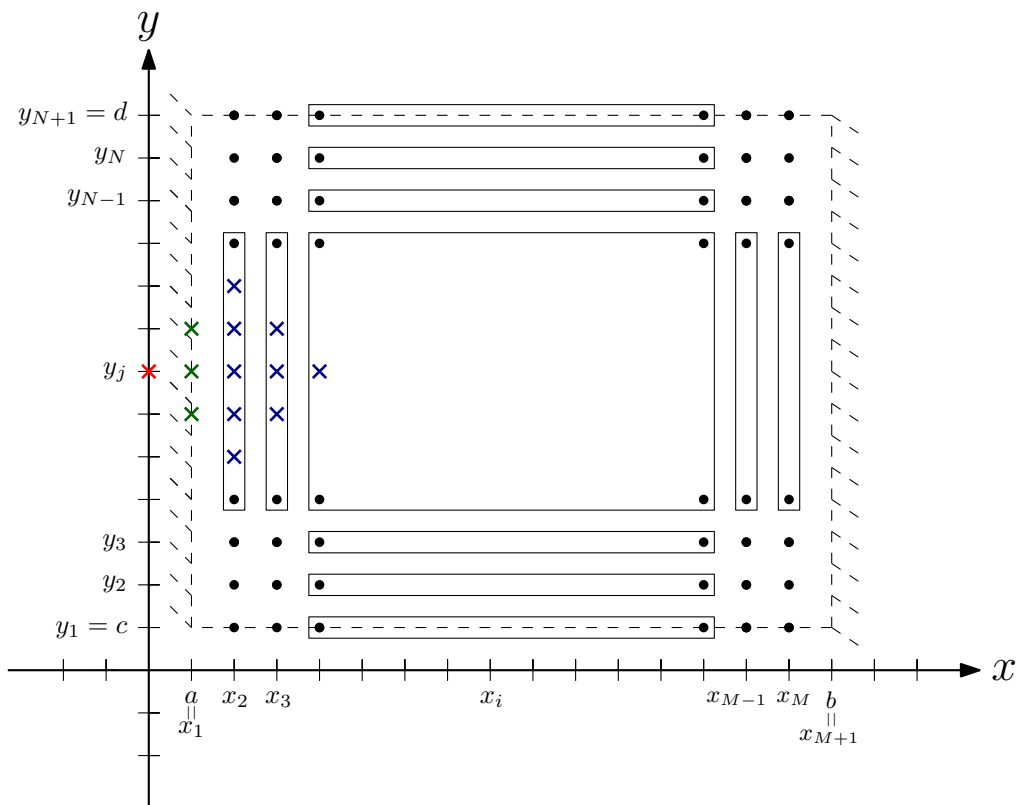
$$\begin{aligned} Aw_{M-2,3} + Bw_{M-1,3} + (C - A)w_{M,3} + Dw_{M-1,4} \\ + Dw_{M-1,2} + Ew_{M,4} + Ew_{M,2} + Fw_{M,5} + Fw_{M,1} = h^4 k^4 p(x, y), \end{aligned} \quad (159)$$

tem-se que a equação (159) representa a aproximação do ponto  $u(x_M, y_3)$ .

### 5.16 Considerando $i = 2$ e $j = 4, \dots, M - 2$

Ao considerar  $i = 2$  e  $j = 4, \dots, M - 2$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam um ponto fora do mesmo, conforme apresenta a Figura 34.

Figura 34 – Molécula posicionada no ponto  $i = 2$  e  $j = 4, \dots, M - 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = 4, \dots, M - 2$  na equação (94),

$$\begin{aligned} Aw_{4,j} + Aw_{0,j} + Bw_{3,j} + Bw_{1,j} \\ + Cw_{2,j} + Dw_{3,j+1} + Dw_{3,j-1} + Dw_{1,j+1} \\ + Dw_{1,j-1} + Ew_{2,j+1} + Ew_{2,j-1} + Fw_{2,j+2} + Fw_{2,j-2} = h^4 k^4 p(x, y), \end{aligned} \quad (160)$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $a$  está submetida, por este motivo, tem-se

$$\begin{aligned} & Aw_{4,j} + A(-w_{2,j}) + Bw_{3,j} \\ & + Cw_{2,j} + Dw_{3,j+1} + Dw_{3,j-1} \\ & + Ew_{2,j+1} + Ew_{2,j-1} + Fw_{2,j+2} + Fw_{2,j-2} = h^4 k^4 p(x, y), \end{aligned} \quad (161)$$

reorganizando os termos,

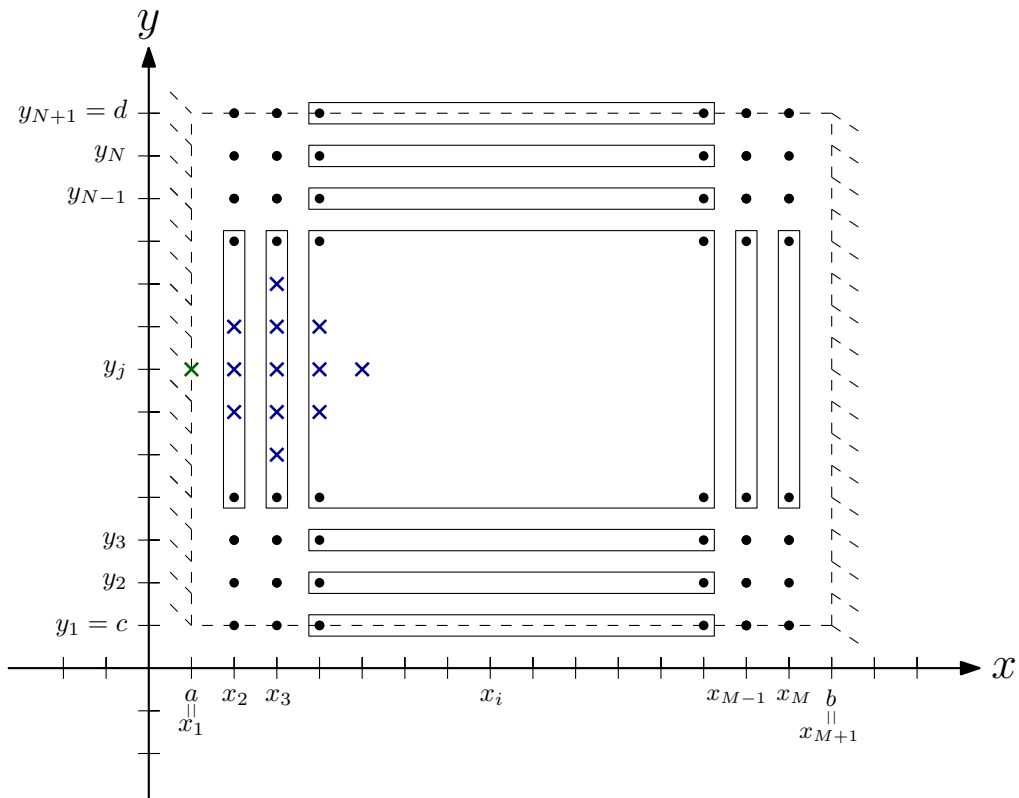
$$\begin{aligned} & Aw_{4,j} + Bw_{3,j} + (C - A)w_{2,j} + Dw_{3,j+1} + Dw_{3,j-1} \\ & + Ew_{2,j+1} + Ew_{2,j-1} + Fw_{2,j+2} + Fw_{2,j-2} = h^4 k^4 p(x, y), \end{aligned} \quad (162)$$

em que a equação (162) representa a aproximação dos pontos  $u(x_2, y_j)$  para todo  $j = 4, \dots, M - 2$ .

### 5.17 Considerando $i = 3$ e $j = 4, \dots, M - 2$

Ao considerar  $i = 3$  e  $j = 4, \dots, M - 2$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam um ponto de borda, conforme apresenta a Figura 35.

Figura 35 – Molécula posicionada no ponto  $i = 3$  e  $j = 4, \dots, M - 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 3$  e  $j = 4, \dots, M - 2$  na equação (94),

$$\begin{aligned}
 & Aw_{5,j} + Aw_{1,j} + Bw_{4,j} + Bw_{2,j} \\
 & + Cw_{3,j} + Dw_{4,j+1} + Dw_{4,j-1} + Dw_{2,j+1} \\
 & + Dw_{2,j-1} + Ew_{3,j+1} + Ew_{3,j-1} + Fw_{3,j+2} + Fw_{3,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{163}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $a$  está submetida, por este motivo, tem-se

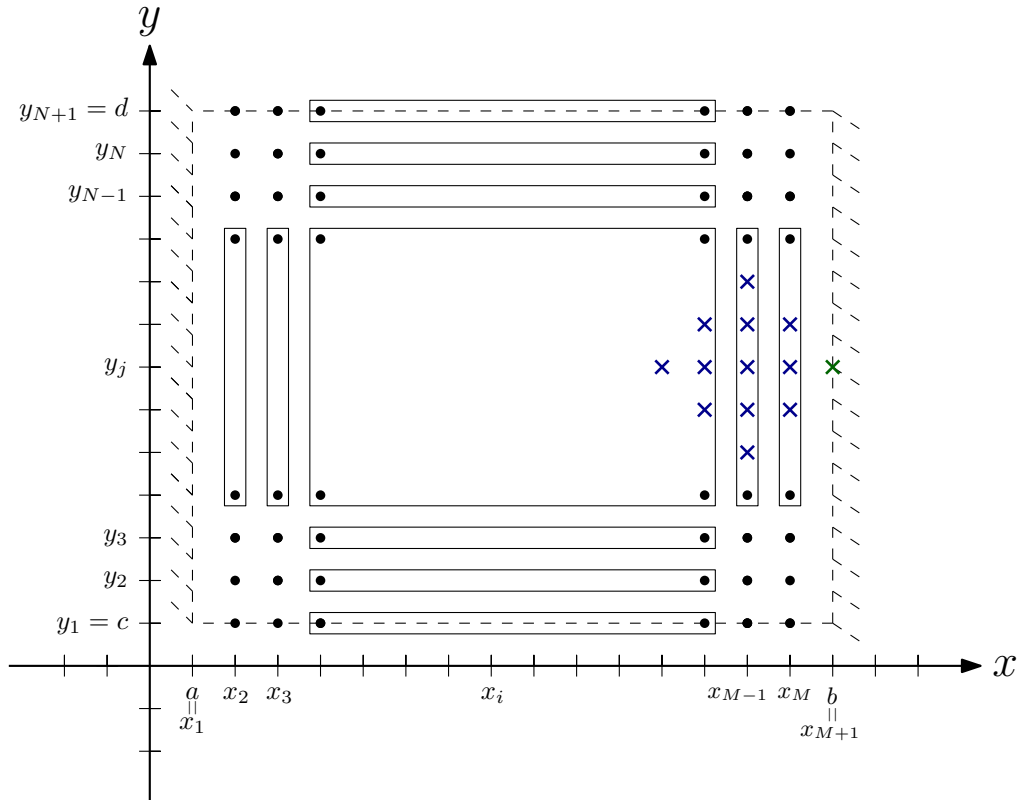
$$\begin{aligned}
 & Aw_{5,j} + Bw_{4,j} + Bw_{2,j} \\
 & + Cw_{3,j} + Dw_{4,j+1} + Dw_{4,j-1} + Dw_{2,j+1} \\
 & + Dw_{2,j-1} + Ew_{3,j+1} + Ew_{3,j-1} + Fw_{3,j+2} + Fw_{3,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{164}$$

em que a equação (164) representa a aproximação dos pontos  $u(x_3, y_j)$  para todo  $j = 4, \dots, M - 2$ .

#### 5.18 Considerando $i = M - 1$ e $j = 4, \dots, M - 2$

Ao considerar  $i = M - 1$  e  $j = 4, \dots, M - 2$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam um ponto de borda, conforme apresenta a Figura 36.

Figura 36 – Molécula posicionada no ponto  $i = M - 1$  e  $j = 4, \dots, M - 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = 4, \dots, M - 2$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,j} + Aw_{M-3,j} + Bw_{M,j} + Bw_{M-2,j} \\
 & + Cw_{M-1,j} + Dw_{M,j+1} + Dw_{M,j-1} + Dw_{M-2,j+1} + Dw_{M-2,j-1} \\
 & + Ew_{M-1,j+1} + Ew_{M-1,j-1} + Fw_{M-1,j+2} + Fw_{M-1,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{165}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $b$  está submetida, por este motivo, tem-se

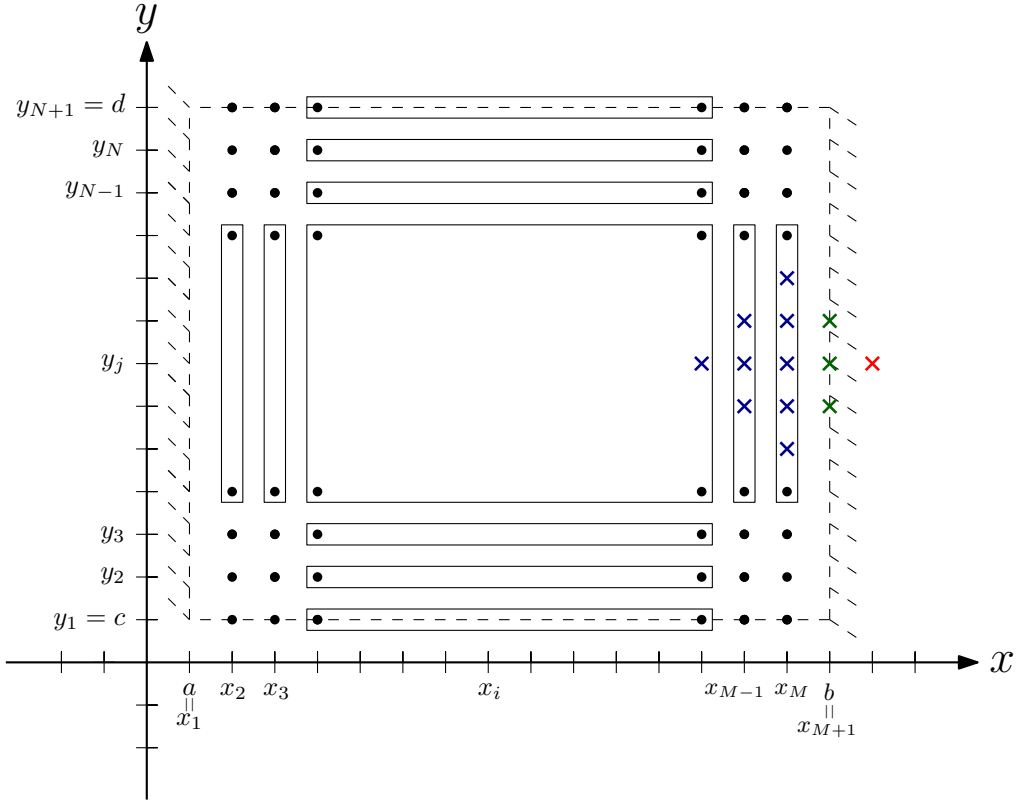
$$\begin{aligned}
 & Aw_{M-3,j} + Bw_{M,j} + Bw_{M-2,j} + Cw_{M-1,j} \\
 & + Dw_{M,j+1} + Dw_{M,j-1} + Dw_{M-2,j+1} + Dw_{M-2,j-1} \\
 & + Ew_{M-1,j+1} + Ew_{M-1,j-1} + Fw_{M-1,j+2} + Fw_{M-1,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{166}$$

em que a equação (166) representa a aproximação dos pontos  $u(x_{M-1}, y_j)$  para todo  $j = 4, \dots, M - 2$ .

5.19 Considerando  $i = M$  e  $j = 4, \dots, M - 2$

Ao considerar  $i = M$  e  $j = 4, \dots, M - 2$  e posicionar a molécula neste conjunto de pontos pertencentes ao domínio  $\Omega$ , percebe-se que eles apresentam somente um ponto fora do mesmo, conforme apresenta a Figura 37.

Figura 37 – Molécula posicionada no ponto  $i = M$  e  $j = 4, \dots, M - 2$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = 4, \dots, M - 2$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,j} + Aw_{M-2,j} + Bw_{M+1,j} + Bw_{M-1,j} \\
 & + Cw_{M,j} + Dw_{M+1,j+1} + Dw_{M+1,j-1} + Dw_{M-1,j+1} \\
 & + Dw_{M-1,j-1} + Ew_{M,j+1} + Ew_{M,j-1} + Fw_{M,j+2} + Fw_{M,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{167}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $b$  está submetida, por este motivo, tem-se

$$\begin{aligned}
 & A(-w_{M,j}) + Aw_{M-2,j} + Bw_{M-1,j} + Cw_{M,j} + Dw_{M-1,j+1} \\
 & + Dw_{M-1,j-1} + Ew_{M,j+1} + Ew_{M,j-1} + Fw_{M,j+2} + Fw_{M,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{168}$$

reorganizando os termos,

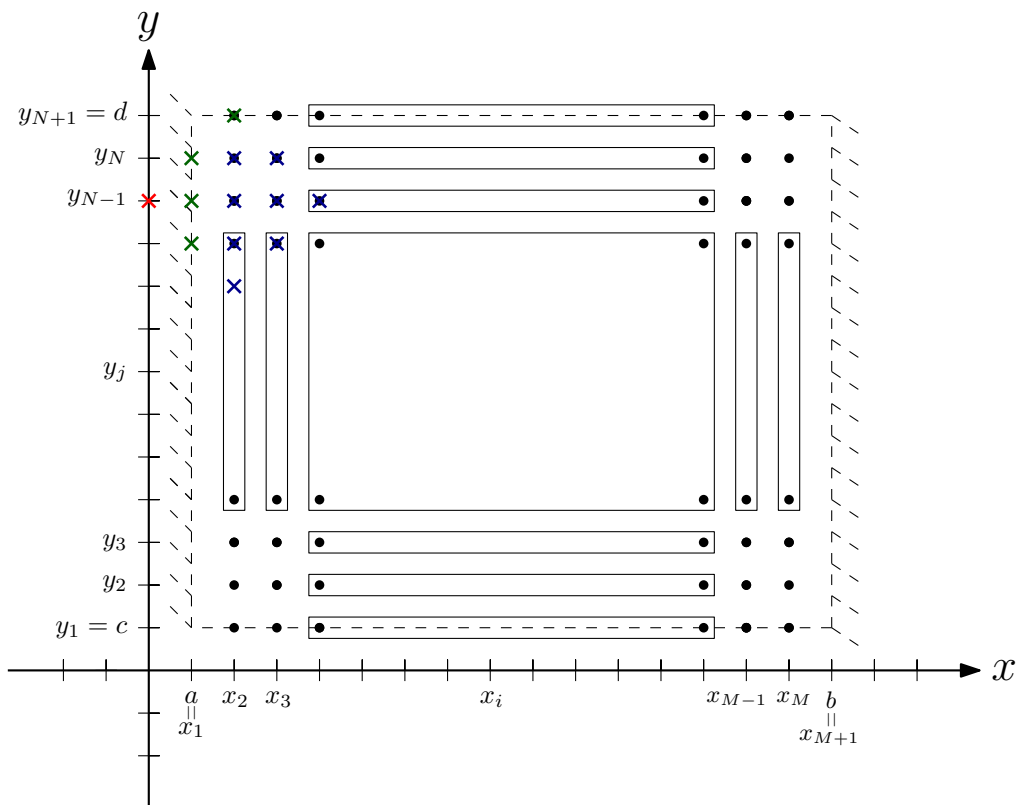
$$\begin{aligned}
 & Aw_{M-2,j} + Bw_{M-1,j} + (C - A)w_{M,j} + Dw_{M-1,j+1} \\
 & + Dw_{M-1,j-1} + Ew_{M,j+1} + Ew_{M,j-1} + Fw_{M,j+2} + Fw_{M,j-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{169}$$

em que a equação (169) representa a aproximação dos pontos  $u(x_M, y_j)$  para todo  $j = 4, \dots, M - 2$ .

### 5.20 Considerando $i = 2$ e $j = N - 1$

Ao considerar  $i = 2$  e  $j = N - 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 38.

Figura 38 – Molécula posicionada no ponto  $i = 2$  e  $j = N - 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = N - 1$  na equação (94),

$$\begin{aligned}
 & Aw_{4,N-1} + Aw_{0,N-1} + Bw_{3,N-1} + Bw_{1,N-1} \\
 & + Cw_{2,N-1} + Dw_{3,N} + Dw_{3,N-2} + Dw_{1,N} \\
 & + Dw_{1,N-2} + Ew_{2,N} + Ew_{2,N-2} + Fw_{2,N+1} + Fw_{2,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{170}$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $d$ , por este motivo, tem-se

$$\begin{aligned} & Aw_{4,N-1} + A(-w_{2,N-1}) + Bw_{3,N-1} \\ & + Cw_{2,N-1} + Dw_{3,N} + Dw_{3,N-2} \\ & + Ew_{2,N} + Ew_{2,N-2} + Fw_{2,N-3} + Fw_{2,N+1} = h^4 k^4 p(x, y), \end{aligned} \quad (171)$$

reorganizando os termos,

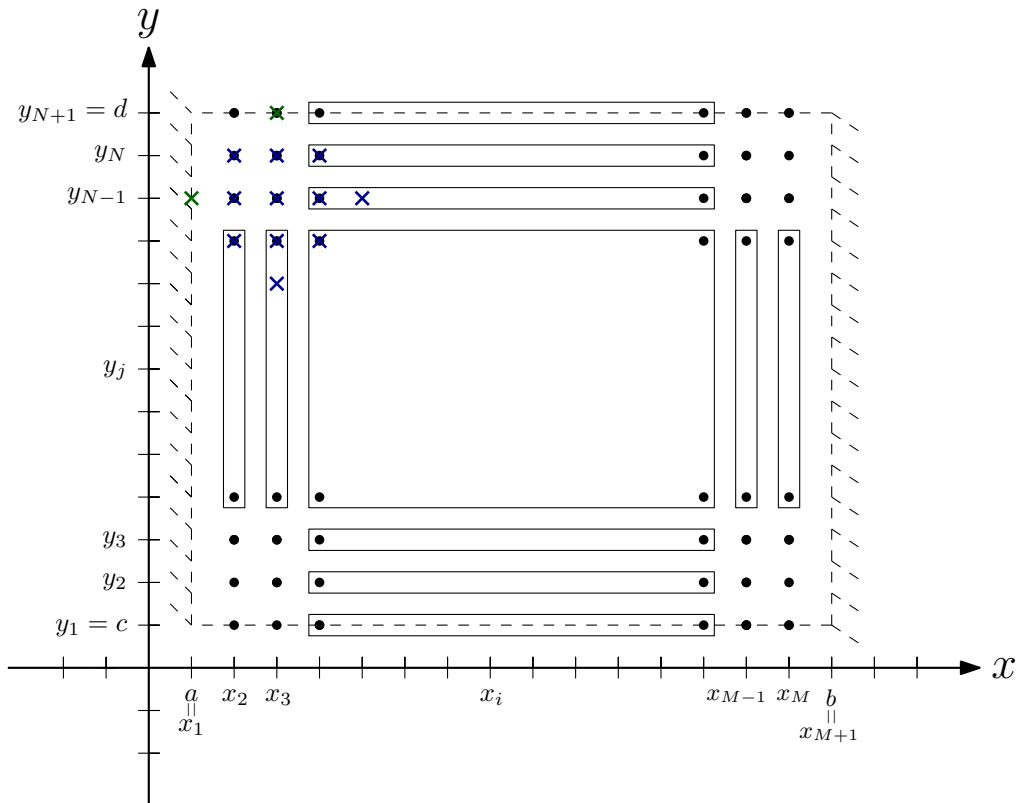
$$\begin{aligned} & Aw_{4,N-1} + Bw_{3,N-1} + (C - A)w_{2,N-1} + Dw_{3,N} \\ & + Dw_{3,N-2} + Ew_{2,N} + Ew_{2,N-2} + Fw_{2,N-3} + Fw_{2,N+1} = h^4 k^4 p(x, y), \end{aligned} \quad (172)$$

em que a equação (172) representa a aproximação do ponto  $u(x_2, y_{N-1})$ .

### 5.21 Considerando $i = 3$ e $j = N - 1$

Ao considerar  $i = 3$  e  $j = N - 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos de borda, conforme apresenta a Figura 39.

Figura 39 – Molécula posicionada no ponto  $i = 3$  e  $j = N - 1$



Fonte: Autoria própria (2018).



Observa-se este fato também ao aplicar  $i = 3$  e  $j = N - 1$  na equação (94),

$$\begin{aligned}
 & Aw_{5,N-1} + Aw_{1,N-1} + Bw_{4,N-1} + Bw_{2,N-1} \\
 & + Cw_{3,N-1} + Dw_{4,N} + Dw_{4,N-2} + Dw_{2,N} \\
 & + Dw_{2,N-2} + Ew_{3,N} + Ew_{3,N-2} + Fw_{3,N+1} + Fw_{3,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{173}$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $d$ , por este motivo,

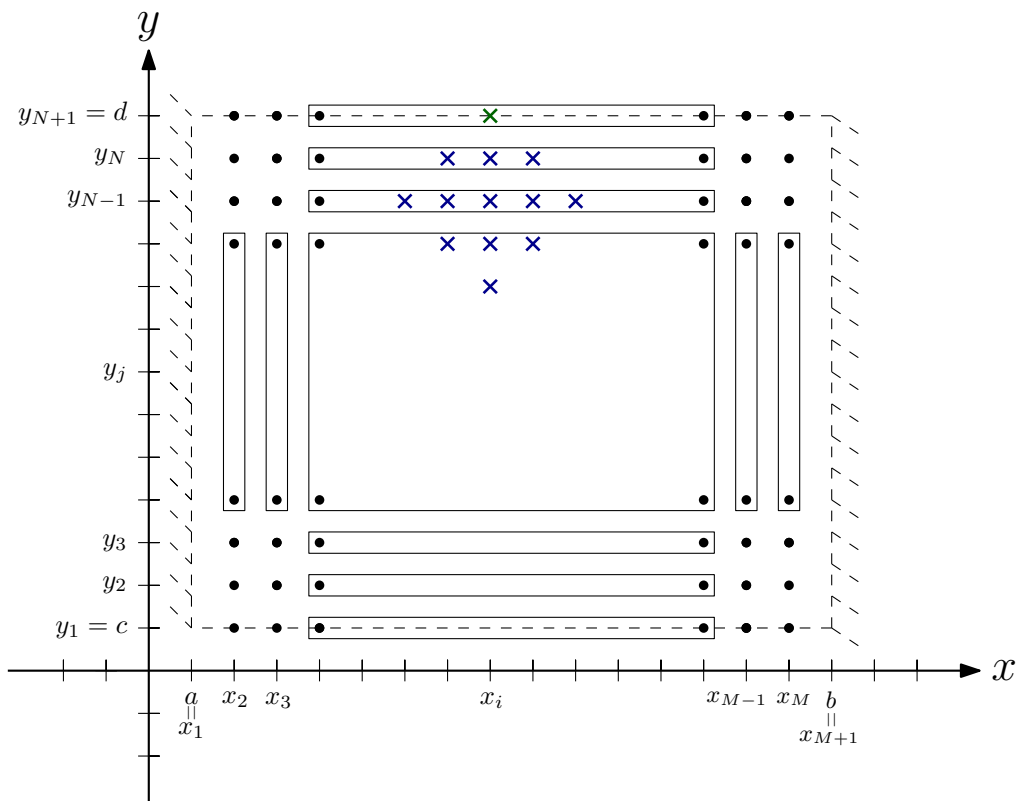
$$\begin{aligned}
 & Aw_{5,N-1} + Bw_{4,N-1} + Bw_{2,N-1} \\
 & + Cw_{3,N-1} + Dw_{4,N} + Dw_{4,N-2} + Dw_{2,N} \\
 & + Dw_{2,N-2} + Ew_{3,N} + Ew_{3,N-2} + Fw_{3,N-3} + Fw_{3,N+1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{174}$$

em que a equação (174) representa a aproximação do ponto  $u(x_3, y_{N-1})$ .

### 5.22 Considerando $i = 4, \dots, M - 2$ e $j = N - 1$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = N - 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto de borda, conforme apresenta a Figura 40.

Figura 40 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = N - 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = N - 1$  na equação (94),

$$\begin{aligned}
 & Aw_{i+2,N-1} + Aw_{i-2,N-1} + Bw_{i+1,N-1} + Bw_{i-1,N-1} \\
 & \quad + Cw_{i,N-1} + Dw_{i+1,N} + Dw_{i+1,N-2} + Dw_{i-1,N} \\
 & + Dw_{i-1,N-2} + Ew_{i,N} + Ew_{i,N-2} + Fw_{i,N+1} + Fw_{i,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{175}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $d$  está submetida, por este motivo,

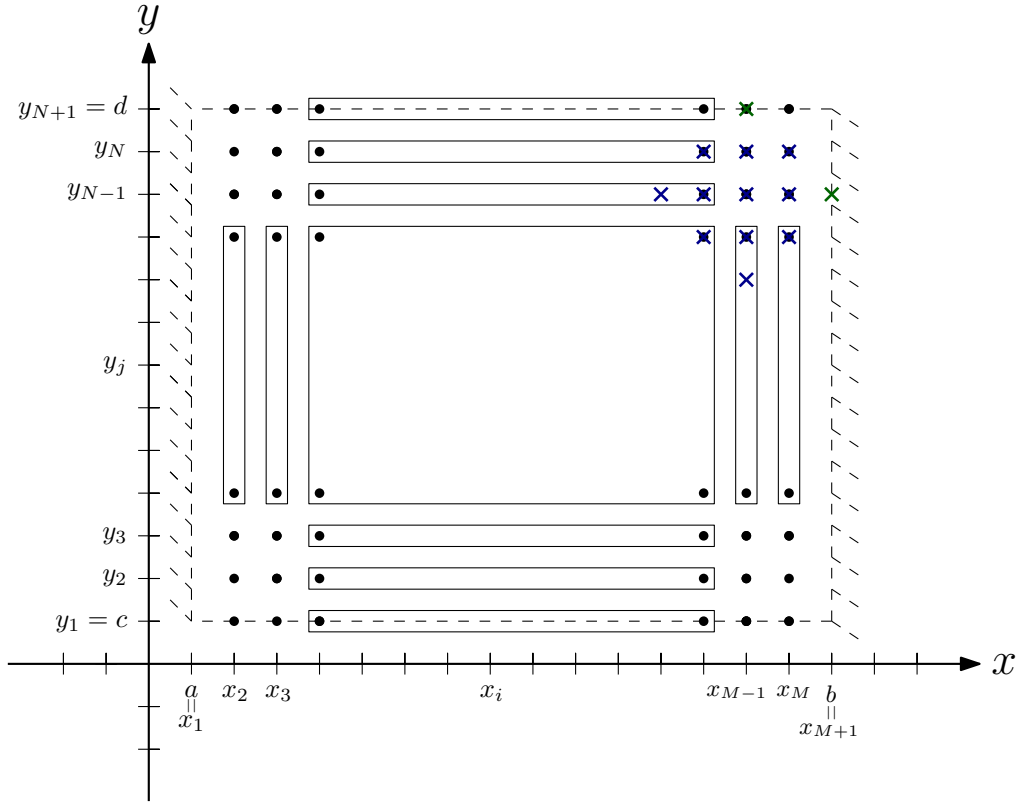
$$\begin{aligned}
 & Aw_{i+2,N-1} + Aw_{i-2,N-1} + Bw_{i+1,N-1} + Bw_{i-1,N-1} \\
 & \quad + Cw_{i,N-1} + Dw_{i+1,N} + Dw_{i+1,N-2} + Dw_{i-1,N} \\
 & + Dw_{i-1,N-2} + Ew_{i,N} + Ew_{i,N-2} + Fw_{i,N+1} + Fw_{i,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{176}$$

em que a equação (176) representa a aproximação dos pontos  $u(x_i, y_{N-1})$  para  $i = 4, \dots, M - 2$ .

### 5.23 Considerando $i = M - 1$ e $j = N - 1$

Ao considerar  $i = M - 1$  e  $j = N - 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos de borda, conforme apresenta a Figura 41.

Figura 41 – Molécula posicionada no ponto  $i = M - 1$  e  $j = N - 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = N - 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,N-1} + Aw_{M-3,N-1} + Bw_{M,N-1} \\
 & + Bw_{M-2,N-1} + Cw_{M-1,N-1} + Dw_{M,N} \\
 & + Dw_{M,N-2} + Dw_{M-2,N} + Dw_{M-2,N-2} \\
 & + Ew_{M-1,N} + Ew_{M-1,N-2} + Fw_{M-1,N+1} + Fw_{M-1,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{177}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $d$ , por este motivo,

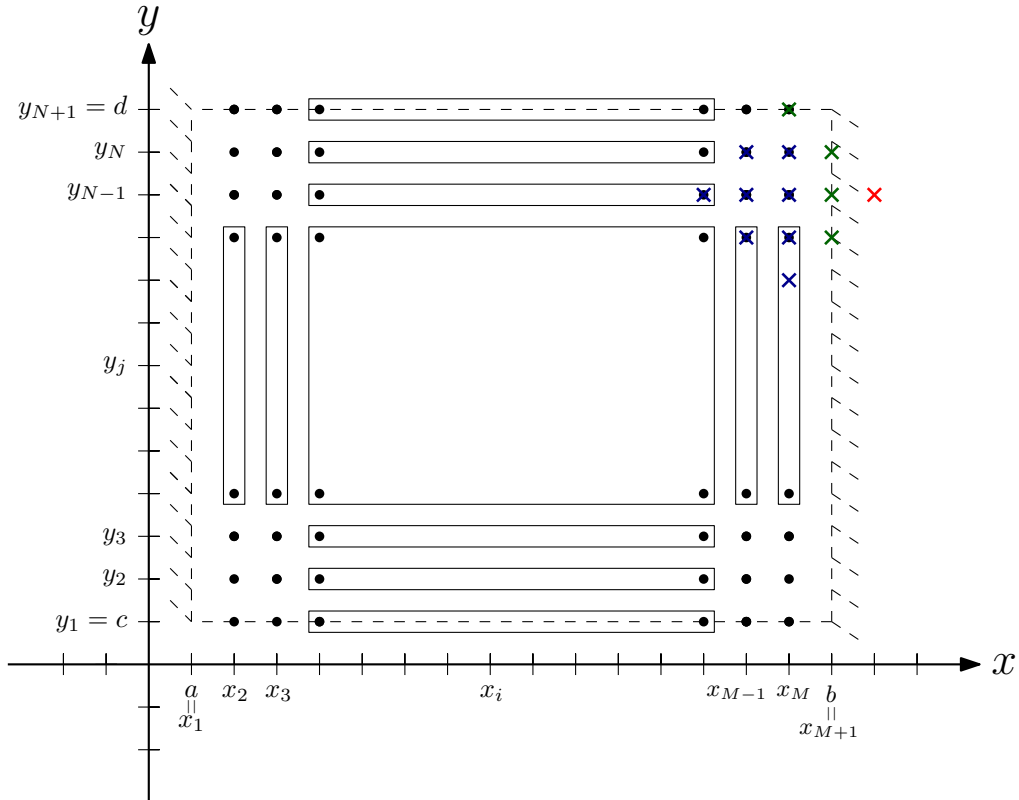
$$\begin{aligned}
 & Aw_{M-3,N-1} + Bw_{M,N-1} + Bw_{M-2,N-1} \\
 & + Cw_{M-1,N-1} + Dw_{M,N} + Dw_{M,N-2} \\
 & + Dw_{M-2,N} + Dw_{M-2,N-2} + Ew_{M-1,N} \\
 & + Ew_{M-1,N-2} + Fw_{M-1,N+1} + Fw_{M-1,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{178}$$

em que a equação (178) representa a aproximação do ponto  $u(x_{M-1}, y_{N-1})$ .

5.24 Considerando  $i = M$  e  $j = N - 1$

Ao considerar  $i = M$  e  $j = N - 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 42.

Figura 42 – Molécula posicionada no ponto  $i = M$  e  $j = N - 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = N - 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,N-1} + Aw_{M-2,N-1} + Bw_{M+1,N-1} + Bw_{M-1,N-1} \\
 & + Cw_{M,N-1} + Dw_{M+1,N} + Dw_{M+1,N-2} + Dw_{M-1,N} \\
 & + Dw_{M-1,N-2} + Ew_{M,N} + Ew_{M,N-2} + Fw_{M,N+1} + Fw_{M,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{179}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $d$ , por este motivo,

$$\begin{aligned}
 & A(-w_{M,N-1}) + Aw_{M-2,N-1} + Bw_{M-1,N-1} \\
 & + Cw_{M,N-1} + Dw_{M-1,N} + Dw_{M-1,N-2} \\
 & + Ew_{M,N} + Ew_{M,N-2} + Fw_{M,N-3} = h^4 k^4 p(x, y),
 \end{aligned} \tag{180}$$

reorganizando os termos,

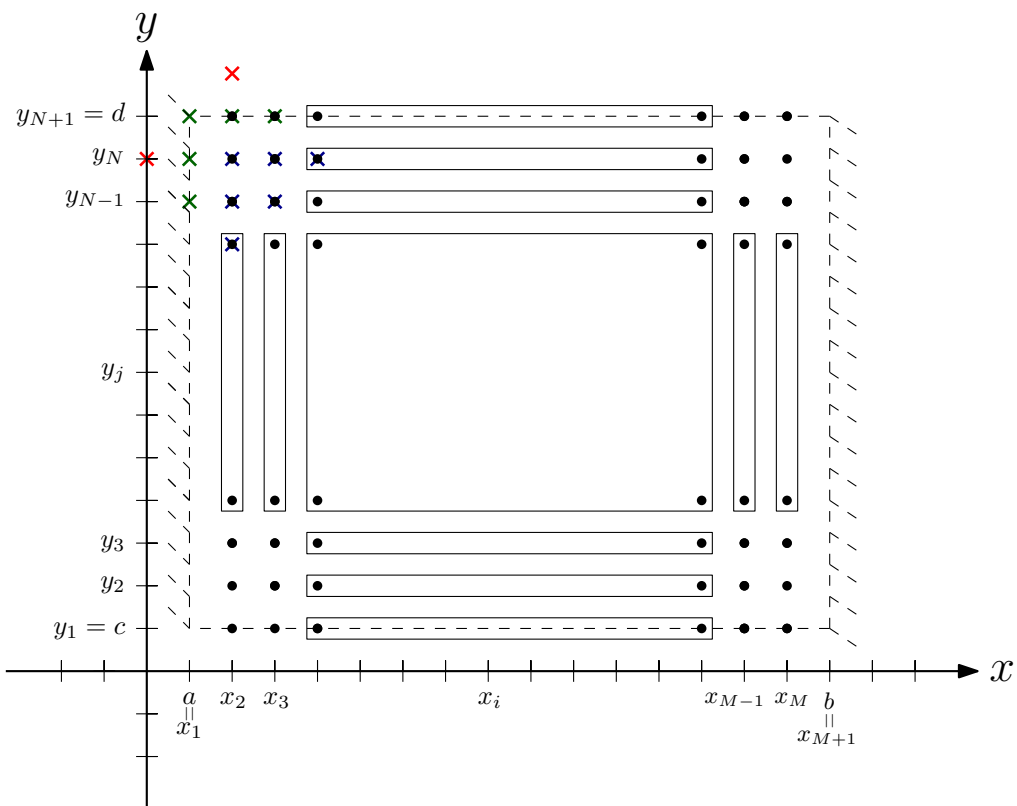
$$\begin{aligned}
 & Aw_{M-2,N-1} + Bw_{M-1,N-1} + (C - A)w_{M,N-1} + Dw_{M-1,N} \\
 & + Dw_{M-1,N-2} + Ew_{M,N} + Ew_{M,N-2} + Fw_{M,N-3} + Fw_{M,N+1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{181}$$

em que a equação (181) representa a aproximação do ponto  $u(x_M, y_{N-1})$ .

### 5.25 Considerando $i = 2$ e $j = N$

Ao considerar  $i = 2$  e  $j = N$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos fora do mesmo, conforme apresenta a Figura 43.

Figura 43 – Molécula posicionada no ponto  $i = 2$  e  $j = N$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = N$  na equação (94),

$$\begin{aligned}
 & Aw_{4,N} + Aw_{0,N} + Bw_{3,N} + Bw_{1,N} \\
 & + Cw_{2,N} + Dw_{3,N+1} + Dw_{3,N-1} + Dw_{1,N+1} \\
 & + Dw_{1,N-1} + Ew_{2,N+1} + Ew_{2,N-1} + Fw_{2,N+2} + Fw_{2,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{182}$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $d$ , por este motivo,

$$\begin{aligned}
 & Aw_{4,N} + A(-w_{2,N}) + Bw_{3,N} \\
 & + Cw_{2,N} + Dw_{3,N+1} + Dw_{3,N-1} \\
 & + Ew_{2,N+1} + Ew_{2,N-1} + F(2w_{2,N+1} - w_{2,N}) + Fw_{2,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{183}$$

reorganizando os termos,

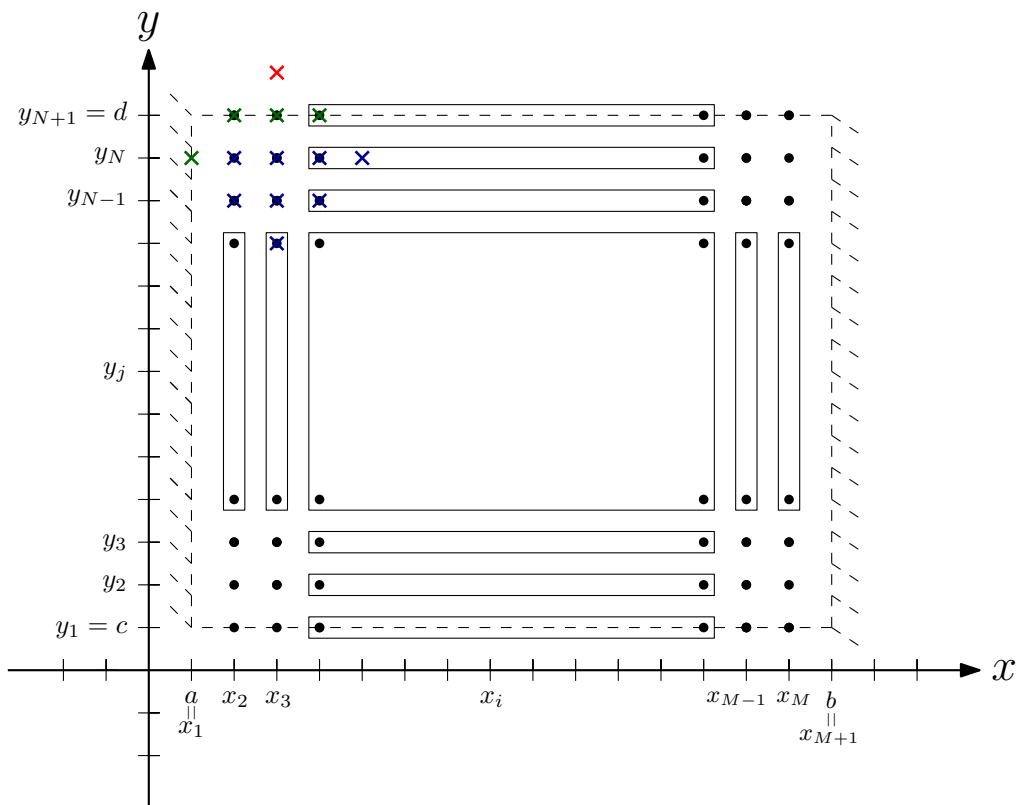
$$Aw_{4,N} + Bw_{3,N} + (C - A - F)w_{2,N} + Dw_{3,N+1} + Dw_{3,N-1} + (E + 2F)w_{2,N+1} + Ew_{2,N-1} + Fw_{2,N-2} = h^4 k^4 p(x, y), \quad (184)$$

em que a equação (184) representa a aproximação do ponto  $u(x_2, y_N)$ .

### 5.26 Considerando $i = 3$ e $j = N$

Ao considerar  $i = 3$  e  $j = N$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 44.

Figura 44 – Molécula posicionada no ponto  $i = 3$  e  $j = N$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 3$  e  $j = N$  na equação (94),

$$Aw_{5,N} + Aw_{1,N} + Bw_{4,N} + Bw_{2,N} + Cw_{3,N} + Dw_{4,N+1} + Dw_{4,N-1} + Dw_{2,N+1} + Dw_{2,N-1} + Ew_{3,N+1} + Ew_{3,N-1} + Fw_{3,N+2} + Fw_{3,N-2} = h^4 k^4 p(x, y), \quad (185)$$

A molécula, neste caso, pega pontos da borda  $a$  e da borda  $d$ , por este motivo,

$$\begin{aligned}
 & Aw_{5,N} + Bw_{4,N} + Bw_{2,N} \\
 & + Cw_{3,N} + Dw_{4,N+1} + Dw_{4,N-1} \\
 & + Dw_{2,N+1} + Dw_{2,N-1} + Ew_{3,N+1} \\
 & + Ew_{3,N-1} + F(2w_{3,N+1} - w_{3,N}) + Fw_{3,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{186}$$

reorganizando os termos,

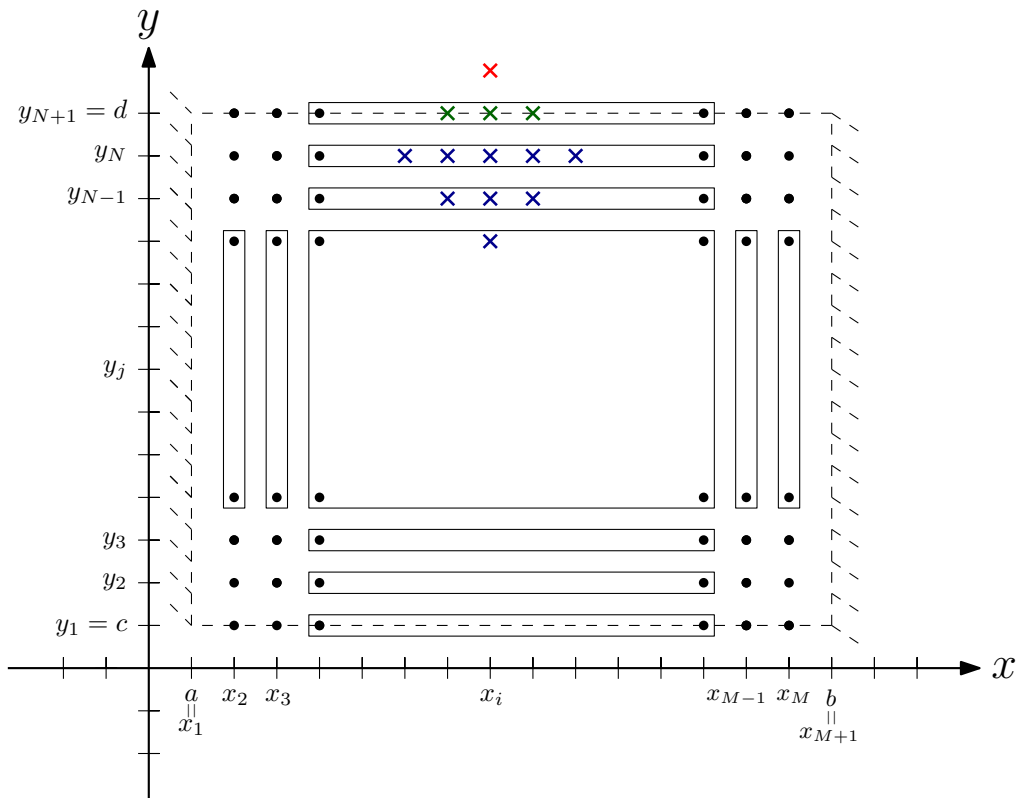
$$\begin{aligned}
 & Aw_{5,N} + Bw_{4,N} + Bw_{2,N} \\
 & + (C - F)w_{3,N} + Dw_{4,N+1} + Dw_{4,N-1} + Dw_{2,N+1} \\
 & + Dw_{2,N-1} + (E + 2F)w_{3,N+1} + Ew_{3,N-1} + Fw_{3,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{187}$$

em que a equação (187) representa a aproximação do ponto  $u(x_3, y_N)$ .

### 5.27 Considerando $i = 4, \dots, M - 2$ e $j = N$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = N$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 45.

Figura 45 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = N$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = N$  na equação (94),

$$\begin{aligned}
& Aw_{i+2,N} + Aw_{i-2,N} + Bw_{i+1,N} + Bw_{i-1,N} \\
& + Cw_{i,N} + Dw_{i+1,N+1} + Dw_{i+1,N-1} + Dw_{i-1,N+1} \\
& + Dw_{i-1,N-1} + Ew_{i,N+1} + Ew_{i,N-1} + Fw_{i,N+2} + Fw_{i,N-2} = h^4 k^4 p(x, y),
\end{aligned} \tag{188}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $d$  está submetida, por este motivo,

$$\begin{aligned}
& Aw_{i+2,N} + Aw_{i-2,N} + Bw_{i+1,N} \\
& + Bw_{i-1,N} + Cw_{i,N} + Dw_{i+1,N+1} \\
& + Dw_{i+1,N-1} + Dw_{i-1,N+1} + Dw_{i-1,N-1} \\
& + Ew_{i,N+1} + Ew_{i,N-1} + F(2w_{i,N+1} - w_{i,N}) + Fw_{i,N-2} = h^4 k^4 p(x, y),
\end{aligned} \tag{189}$$

reorganizando os termos,

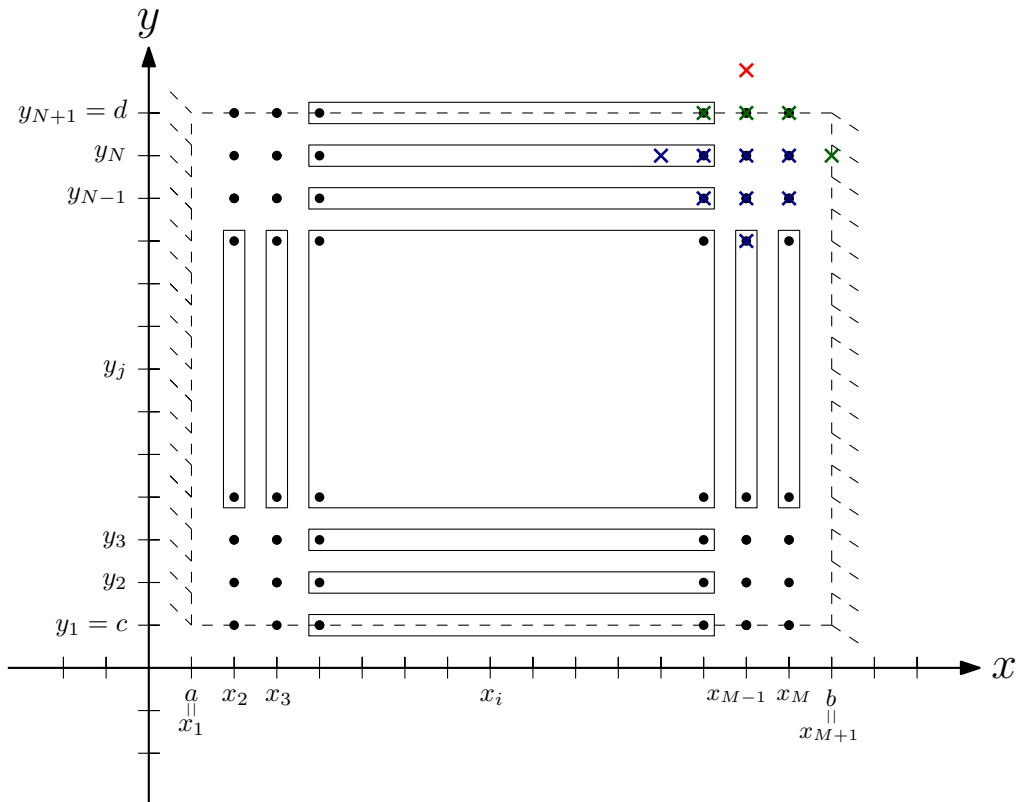
$$\begin{aligned}
& Aw_{i+2,N} + Aw_{i-2,N} + Bw_{i+1,N} + Bw_{i-1,N} \\
& + (C - F)w_{i,N} + Dw_{i+1,N+1} + Dw_{i+1,N-1} + Dw_{i-1,N+1} \\
& + Dw_{i-1,N-1} + (E + 2F)w_{i,N+1} + Ew_{i,N-1} + Fw_{i,N-2} = h^4 k^4 p(x, y),
\end{aligned} \tag{190}$$

tem-se que a equação (190) representa a aproximação dos pontos  $u(x_i, y_N)$  para  $i = 4, \dots, M - 2$ .

### 5.28 Considerando $i = M - 1$ e $j = N$

Ao considerar  $i = M - 1$  e  $j = N$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta um ponto fora do mesmo, conforme apresenta a Figura 46.



Figura 46 – Molécula posicionada no ponto  $i = M - 1$  e  $j = N$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = N$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,N} + Aw_{M-3,N} + Bw_{M,N} \\
 & + Bw_{M-2,N} + Cw_{M-1,N} + Dw_{M,N+1} \\
 & + Dw_{M,N-1} + Dw_{M-2,N+1} + Dw_{M-2,N-1} \\
 & + Ew_{M-1,N+1} + Ew_{M-1,N-1} + Fw_{M-1,N+2} + Fw_{M-1,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{191}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $d$ , por este motivo,

$$\begin{aligned}
 & Aw_{M-3,N} + Bw_{M,N} + Bw_{M-2,N} \\
 & + Cw_{M-1,N} + Dw_{M,N+1} + Dw_{M,N-1} + Dw_{M-2,N+1} \\
 & + Dw_{M-2,N-1} + Ew_{M-1,N+1} + Ew_{M-1,N-1} \\
 & + F(2w_{M-1,N+1} - w_{M-1,N}) + Fw_{M-1,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{192}$$

reorganizando os termos,

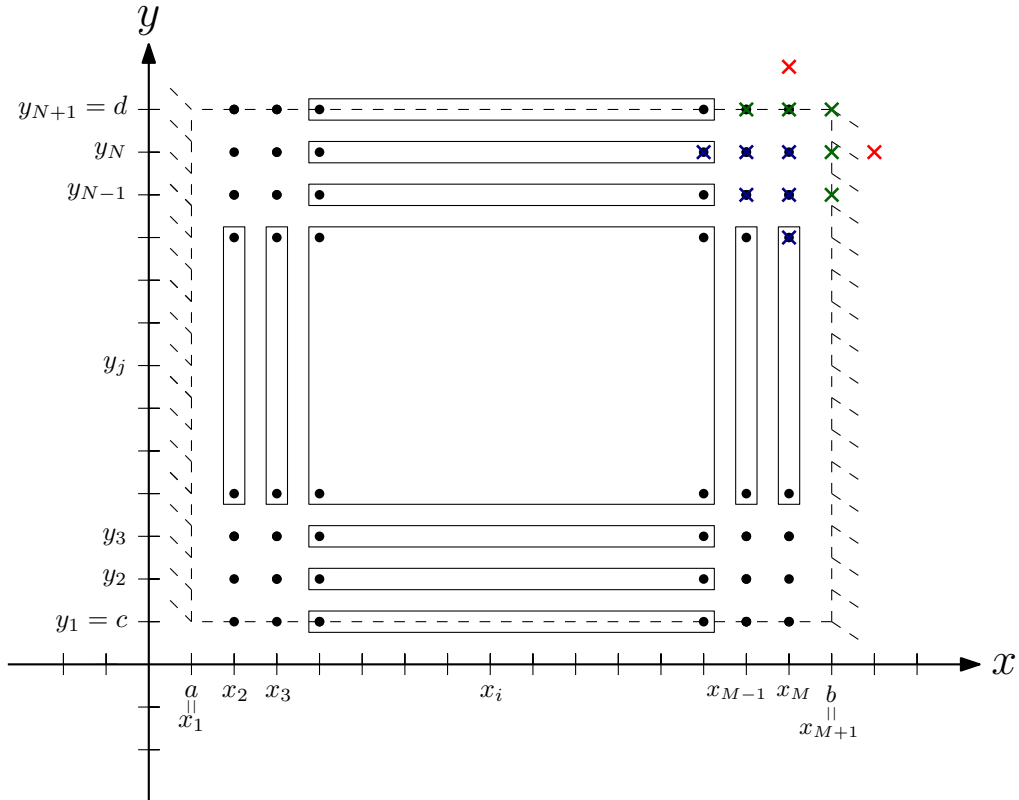
$$\begin{aligned}
 & Aw_{M-3,N} + Bw_{M,N} + Bw_{M-2,N} \\
 & + (C - F)w_{M-1,N} + Dw_{M,N+1} + Dw_{M,N-1} \\
 & + Dw_{M-2,N+1} + Dw_{M-2,N-1} + (E + 2F)w_{M-1,N+1} \\
 & + Ew_{M-1,N-1} + Fw_{M-1,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{193}$$

em que a equação (193) representa a aproximação do ponto  $u(x_{M-1}, y_N)$ .

5.29 Considerando  $i = M$  e  $j = N$

Ao considerar  $i = M$  e  $j = N$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta dois pontos fora do mesmo, conforme apresenta a Figura 47.

Figura 47 – Molécula posicionada no ponto  $i = M$  e  $j = N$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = N$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,N} + Aw_{M-2,N} + Bw_{M+1,N} + Bw_{M-1,N} \\
 & + Cw_{M,N} + Dw_{M+1,N+1} + Dw_{M+1,N-1} + Dw_{M-1,N+1} \\
 & + Dw_{M-1,N-1} + Ew_{M,N+1} + Ew_{M,N-1} + Fw_{M,N+2} + Fw_{M,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{194}$$

A molécula, neste caso, pega pontos da borda  $b$  e da borda  $d$ , por este motivo,

$$\begin{aligned}
 & A(-w_{M,N}) + Aw_{M-2,N} + Bw_{M-1,N} + Cw_{M,N} + Dw_{M-1,N+1} \\
 & + Dw_{M-1,N-1} + Ew_{M,N+1} + Ew_{M,N-1} \\
 & + F(2w_{M,N+1} - w_{M,N}) + Fw_{M,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{195}$$

reorganizando os termos,

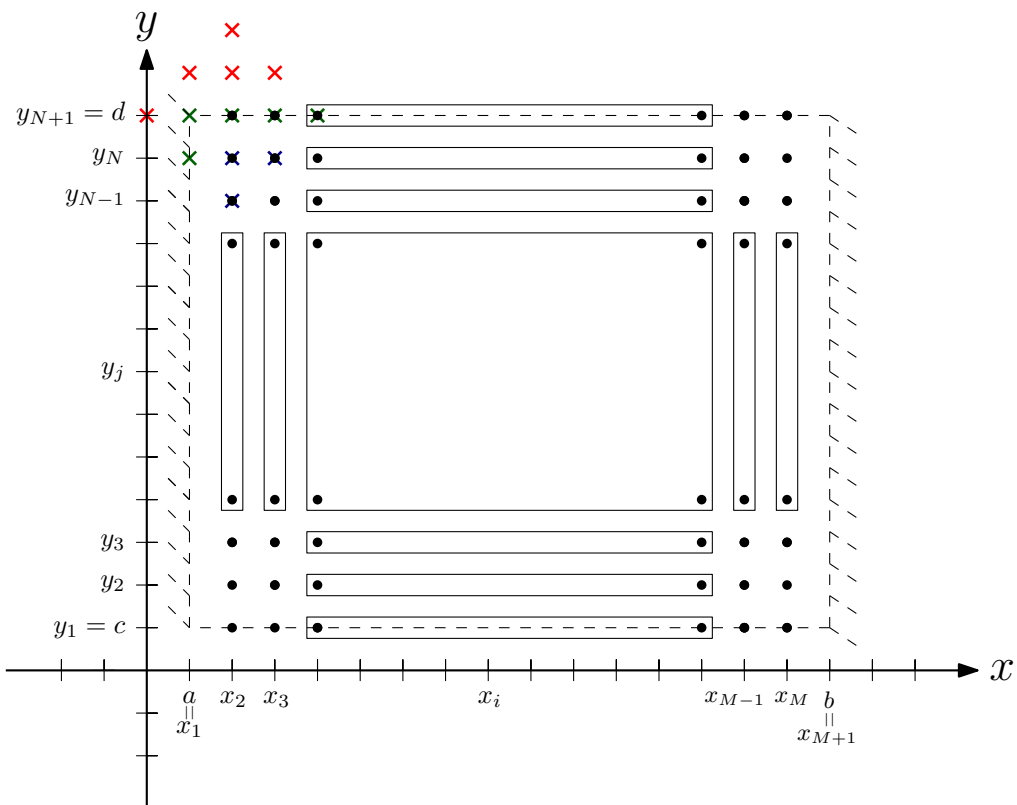
$$\begin{aligned}
 & Aw_{M-2,N} + Bw_{M-1,N} + (C - A - F)w_{M,N} + Dw_{M-1,N+1} \\
 & + Dw_{M-1,N-1} + (E + 2F)w_{M,N+1} + Ew_{M,N-1} + Fw_{M,N-2} = h^4 k^4 p(x, y),
 \end{aligned} \tag{196}$$

em que a equação (196) representa a aproximação do ponto  $u(x_M, y_N)$ .

### 5.30 Considerando $i = 2$ e $j = N + 1$

Ao considerar  $i = 2$  e  $j = N + 1$  e posicionar a molécula no domínio  $\Omega$ , percebe-se que há cinco pontos fora do mesmo, conforme apresenta a Figura 48.

Figura 48 – Molécula posicionada no ponto  $i = 2$  e  $j = N + 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 2$  e  $j = N + 1$  na equação (94),

$$\begin{aligned}
 & Aw_{4,N+1} + Aw_{0,N+1} + Bw_{3,N+1} + Bw_{1,N+1} \\
 & + Cw_{2,N+1} + Dw_{3,N+2} + Dw_{3,N} + Dw_{1,N+2} \\
 & + Dw_{1,N} + Ew_{2,N+2} + Ew_{2,N} + Fw_{2,N+3} + Fw_{2,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{197}$$

Novamente, a molécula irá ser influenciada pelas condições de contorno que as bordas  $a$  e  $d$  estão submetidas, sendo assim,

$$\begin{aligned}
 & Aw_{4,N+1} + A(-w_{2,N+1}) + Bw_{3,N+1} \\
 & + Cw_{2,N+1} + D(2w_{3,N+1} - w_{3,N}) + Dw_{3,N} \\
 & + E(2w_{2,N+1} - w_{2,N}) + Ew_{2,N} \\
 & + F(w_{2,N-1} + 4w_{2,N+1} - 4w_{2,N}) + Fw_{2,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{198}$$

reorganizando os termos,

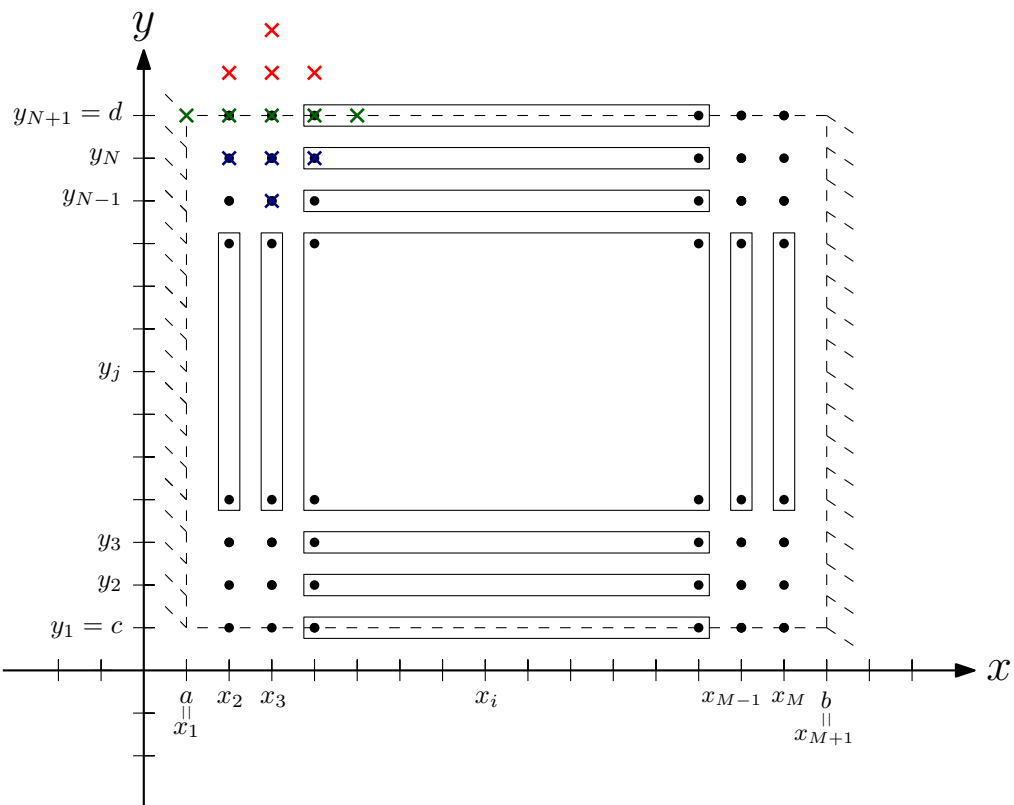
$$Aw_{4,N+1} + (B + 2D)w_{3,N+1} + (C - A + 2E + 4F)w_{2,N+1} + 2Fw_{2,N-1} - 4Fw_{2,N} = h^4k^4p(x, y), \quad (199)$$

em que a equação (199) representa a aproximação do ponto  $u(x_2, y_{N+1})$ .

### 5.31 Considerando $i = 3$ e $j = N + 1$

Ao considerar  $i = 3$  e  $j = N + 1$  e posicionar a molécula no domínio  $\Omega$ , percebe-se que há quatro pontos fora do mesmo, conforme apresenta a Figura 49.

Figura 49 – Molécula posicionada no ponto  $i = 3$  e  $j = N + 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = 3$  e  $j = N + 1$  na equação (94),

$$Aw_{5,N+1} + Aw_{1,N+1} + Bw_{4,N+1} + Bw_{2,N+1} + Cw_{4,N+1} + Dw_{4,N+2} + Dw_{4,N} + Dw_{2,N+2} + Dw_{2,N} + Ew_{3,N+2} + Ew_{3,N} + Fw_{3,N+3} + Fw_{3,N-1} = h^4k^4p(x, y), \quad (200)$$

Novamente, a molécula irá ser influenciada pelas condições de contorno que as bordas  $a$  e  $d$  estão submetidas, sendo assim,

$$\begin{aligned}
 & Aw_{5,N+1} + Bw_{4,N+1} + Bw_{2,N+1} + Cw_{4,N+1} \\
 & + D(2w_{4,N+1} - w_{4,N}) + Dw_{4,N} + D(2w_{2,N+1} - w_{2,N}) \\
 & + Dw_{2,N} + E(2w_{3,N+1} - w_{3,N}) + Ew_{3,N} \\
 & + F(w_{3,N-1} + 4w_{3,N+1} - 4w_{3,N}) + Fw_{3,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{201}$$

reorganizando os termos,

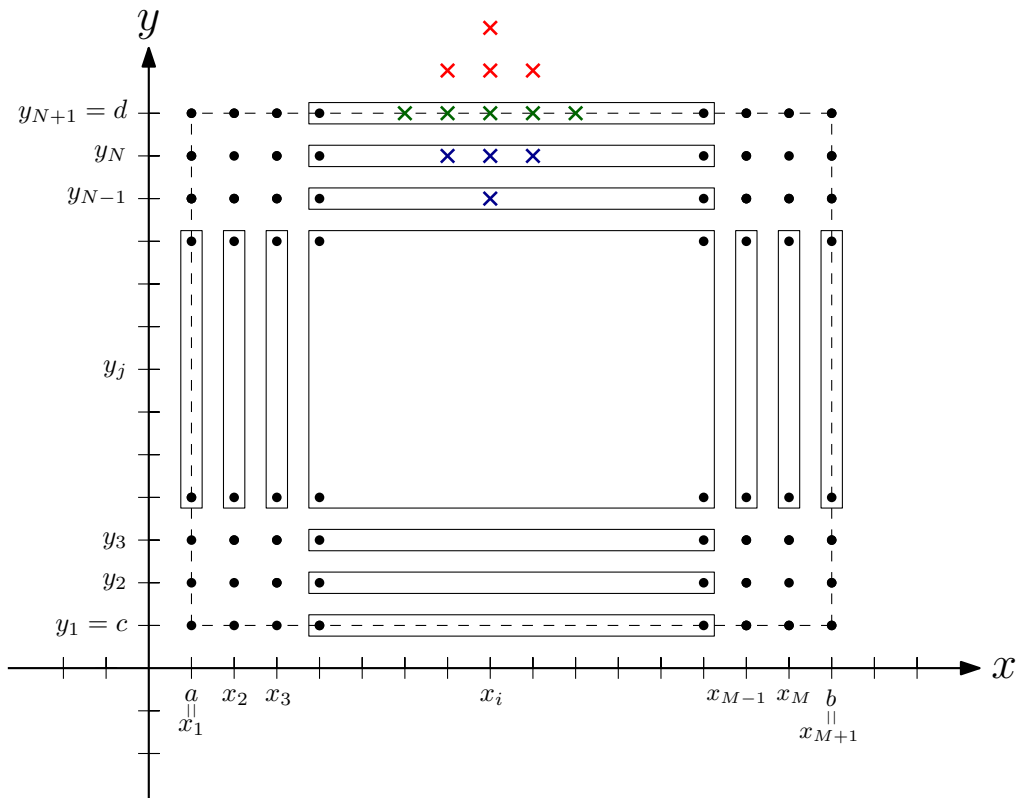
$$\begin{aligned}
 & Aw_{5,N+1} + (B + 2D)w_{4,N+1} + (B + 2D)w_{2,N+1} \\
 & + (C + 2E + 4F)w_{4,N+1} + 2Fw_{3,N-1} - 4Fw_{3,N} = h^4 k^4 p(x, y),
 \end{aligned} \tag{202}$$

em que a equação (202) representa a aproximação do ponto  $u(x_3, y_{N+1})$ .

### 5.32 Considerando $i = 4, \dots, M - 2$ e $j = N + 1$

Ao considerar  $i = 4, \dots, M - 2$  e  $j = N + 1$  e posicionar a molécula neste ponto pertencente ao domínio  $\Omega$ , percebe-se que apresenta quatro pontos fora do mesmo, conforme apresenta a Figura 50.

Figura 50 – Molécula posicionada no ponto  $i = 4, \dots, M - 2$  e  $j = N + 1$



Observa-se este fato também ao aplicar  $i = 4, \dots, M - 2$  e  $j = N + 1$  na equação (94),

$$\begin{aligned}
& Aw_{i+2,N+1} + Aw_{i-2,N+1} + Bw_{i+1,N+1} + Bw_{i-1,N+1} \\
& + Cw_{i,N+1} + Dw_{i+1,N+2} + Dw_{i+1,N} + Dw_{i-1,N+2} \\
& + Dw_{i-1,N} + Ew_{i,N+2} + Ew_{i,N} + Fw_{i,N+3} + Fw_{i,N-1} = h^4 k^4 p(x, y),
\end{aligned} \tag{203}$$

A molécula, neste caso, sofrerá interferência somente das condições que a borda  $d$  está submetida, por este motivo,

$$\begin{aligned}
& Aw_{i+2,N+1} + Aw_{i-2,N+1} + Bw_{i+1,N+1} + Bw_{i-1,N+1} \\
& + Cw_{i,N+1} + D(2w_{i+1,N+1} - w_{i+1,N}) + Dw_{i+1,N} \\
& + D(2w_{i-1,N+1} - w_{i-1,N}) + Dw_{i-1,N} \\
& + E(2w_{i,N+1} - w_{i,N}) + Ew_{i,N} \\
& + F(w_{i,N-1} + 4w_{i,N+1} - 4w_{i,N}) + Fw_{i,N-1} = h^4 k^4 p(x, y),
\end{aligned} \tag{204}$$

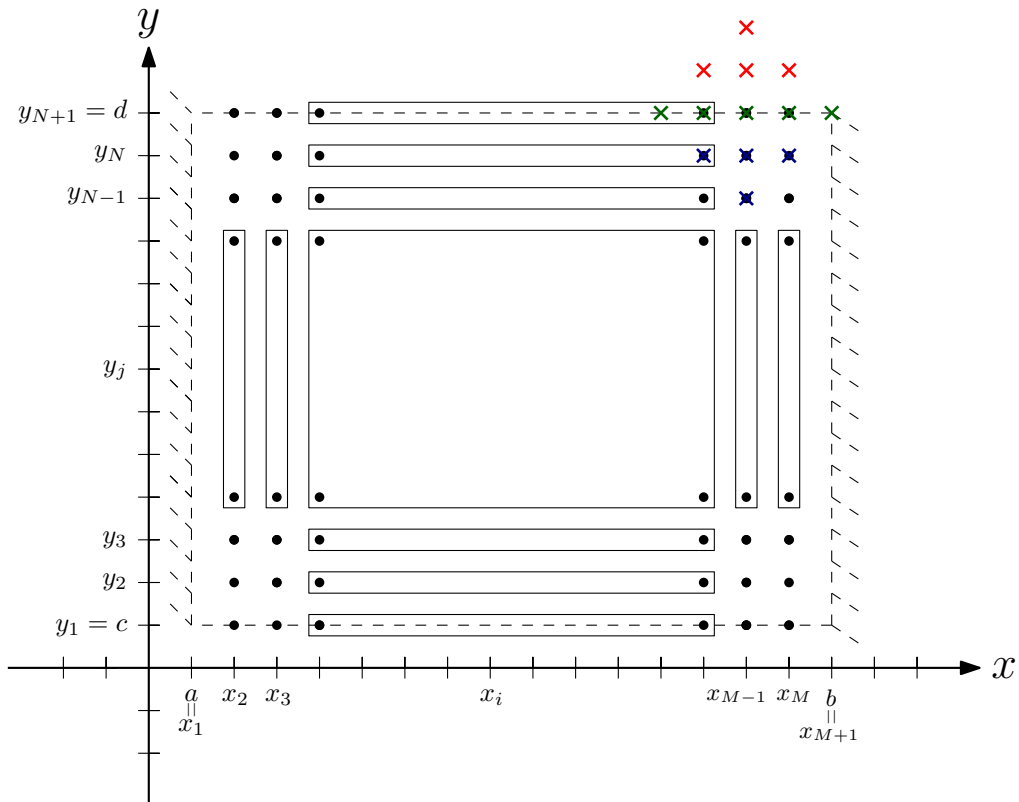
reorganizando os termos,

$$\begin{aligned}
& Aw_{i+2,N+1} + Aw_{i-2,N+1} + (B + 2D)w_{i+1,N+1} \\
& + (B + 2D)w_{i-1,N+1} + (C + 2E + 4F)w_{i,N+1} \\
& + 2Fw_{i,N-1} - 4Fw_{i,N} = h^4 k^4 p(x, y),
\end{aligned} \tag{205}$$

em que a equação (205) representa a aproximação dos pontos  $u(x_i, y_{N+1})$  para  $i = 4, \dots, M - 2$ .

### 5.33 Considerando $i = M - 1$ e $j = N + 1$

Ao considerar  $i = M - 1$  e  $j = N + 1$  e posicionar a molécula no domínio  $\Omega$ , percebe-se que há quatro pontos fora do mesmo, conforme apresenta a Figura 51.

Figura 51 – Molécula posicionada no ponto  $i = M - 1$  e  $j = N + 1$ 

Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M - 1$  e  $j = N + 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+1,N+1} + Aw_{M-3,N+1} + Bw_{M,N+1} \\
 & + Bw_{M-2,N+1} + Cw_{M-1,N+1} + Dw_{M,N+2} \\
 & + Dw_{M,N} + Dw_{M-2,N+2} + Dw_{M-2,N} \\
 & + Ew_{M-1,N+2} + Ew_{M-1,N} + Fw_{M-1,N+3} + Fw_{M-1,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{206}$$

Novamente, a molécula irá ser influenciada pelas condições de contorno que as bordas  $b$  e  $d$  estão submetidas, sendo assim,

$$\begin{aligned}
 & Aw_{M-3,N+1} + Bw_{M,N+1} + Bw_{M-2,N+1} + Cw_{M-1,N+1} \\
 & + D(2w_{M,N+1} - w_{M,N}) + Dw_{M,N} \\
 & + D(2w_{M-2,N+1} - w_{M-2,N}) + Dw_{M-2,N} \\
 & + E(2w_{M-1,N+1} - w_{M-1,N}) + Ew_{M-1,N} \\
 & + F(w_{M-1,N-1} + 4w_{M-1,N+1} - 4w_{M-1,N}) + Fw_{M-1,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{207}$$

reorganizando os termos,

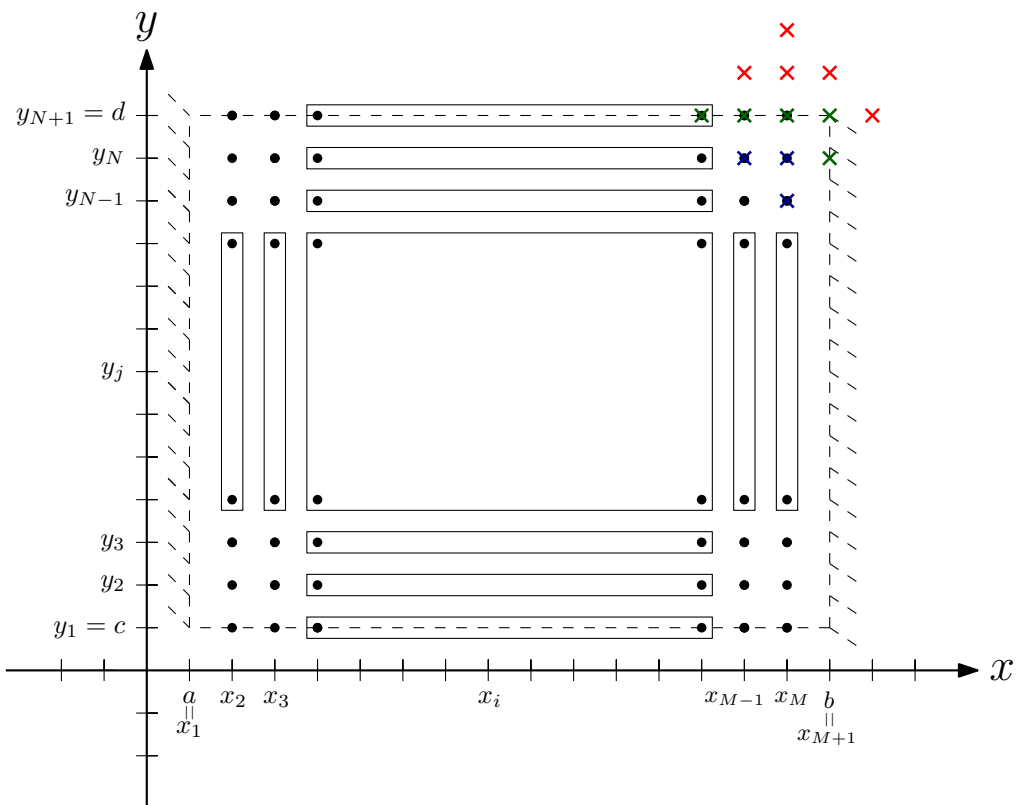
$$\begin{aligned}
 & Aw_{M-3,N+1} + (B + 2D)w_{M,N+1} \\
 & + (B + 2D)w_{M-2,N+1} + (C + 2E + 4F)w_{M-1,N+1} \\
 & + 2F(w_{M-1,N-1} - 4Fw_{M-1,N}) = h^4 k^4 p(x, y),
 \end{aligned} \tag{208}$$

em que a equação (208) representa a aproximação do ponto  $u(x_{M-1}, y_{N+1})$ .

### 5.34 Considerando $i = M$ e $j = N + 1$

Ao considerar  $i = M$  e  $j = N + 1$  e posicionar a molécula no domínio  $\Omega$ , percebe-se que há cinco pontos fora do mesmo, conforme apresenta a Figura 52.

Figura 52 – Molécula posicionada no ponto  $i = M$  e  $j = N + 1$



Fonte: Autoria própria (2018).

Observa-se este fato também ao aplicar  $i = M$  e  $j = N + 1$  na equação (94),

$$\begin{aligned}
 & Aw_{M+2,N+1} + Aw_{M-2,N+1} + Bw_{M+1,N+1} + Bw_{M-1,N+1} \\
 & + Cw_{M,N+1} + Dw_{M+1,N+2} + Dw_{M+1,N} + Dw_{M-1,N+2} \\
 & + Dw_{M-1,N} + Ew_{M,N+2} + Ew_{M,N} + Fw_{M,N+3} + Fw_{M,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{209}$$

Novamente, a molécula irá ser influenciada pelas condições de contorno que as bordas  $b$  e  $d$  estão submetidas, sendo assim,

$$\begin{aligned}
 & A(-w_{M,N+1}) + Aw_{M-2,N+1} + Bw_{M-1,N+1} + Cw_{M,N+1} \\
 & + D(2w_{M-1,N+1} - w_{M-1,N}) + Dw_{M-1,N} \\
 & + E(2w_{M,N+1} - w_{M,N}) + Ew_{M,N} \\
 & + F(w_{M,N-1} + 4w_{M,N+1} - 4w_{M,N}) + Fw_{M,N-1} = h^4 k^4 p(x, y),
 \end{aligned} \tag{210}$$



reorganizando os termos,

$$Aw_{M-2,N+1} + (B + 2D)w_{M-1,N+1} + (C - A + 2E + 4F)w_{M,N+1} + 2Fw_{M,N-1} - 4Fw_{M,N} = h^4k^4p(x, y), \quad (211)$$

em que a equação (211) representa a aproximação do ponto  $u(x_M, y_{N+1})$ .

Com tais equações, o sistema passou a ter  $(M - 1).(N + 1)$  equações com  $(M - 1).(N + 1)$  incógnitas, tornando-se um sistema possível e determinado.

### 5.35 Interface de cálculo

Com a finalidade de revisar a teoria apresentada neste trabalho, verificando as etapas e conceitos, e implementar diversos casos de placas, foi desenvolvido um programa desenvolvido em linguagem *MatLab*<sup>®</sup>, utilizando sua ferramenta App Designer, disponível na versão R2017b.

Este programa tem como objetivo ser simples e de fácil utilização permitindo o cálculo da deflexão de vários tipos de placas ortotrópicas equivalentes. Os conceitos iniciais do programa são os seguintes:

- Facilidade da interpretação e leitura dos resultados por parte do usuário;
- Geração dos pontos de resultado conforme a malha pré-definida pelo usuário;
- Visualização dos pontos da malha discretizada;
- Possibilidade de realizar combinações dos carregamentos implementados utilizando os coeficientes de ponderação que o usuário definiu;
- Visualização da estrutura deformada em 3 dimensões;
- Facilidade de análise dos resultados;

Para demonstração da interface, será considerada um estudo de caso de uma placa ortotrópica bi-apoiada seguindo os seguintes dados de entrada:

$$\begin{aligned} \text{Largura } (b) &= 6000 \text{ mm;} \\ \text{Comprimento } (L) &= 6000 \text{ mm;} \\ \text{Espessura } (t) &= 200 \text{ mm;} \\ \text{Módulo de Young } (E) &= 20000 \text{ MPa;} \\ \text{Módulo de elasticidade ao cisalhamento } (G) &= 8695,65 \text{ MPa;} \\ \text{Coeficiente de Poisson } (\nu) &= 0,15, \end{aligned}$$

carga uniformemente distribuída  $p = 25 \text{ kN/m}^2$ .

O programa foi executado em Windows 10, com a seguinte página inicial:

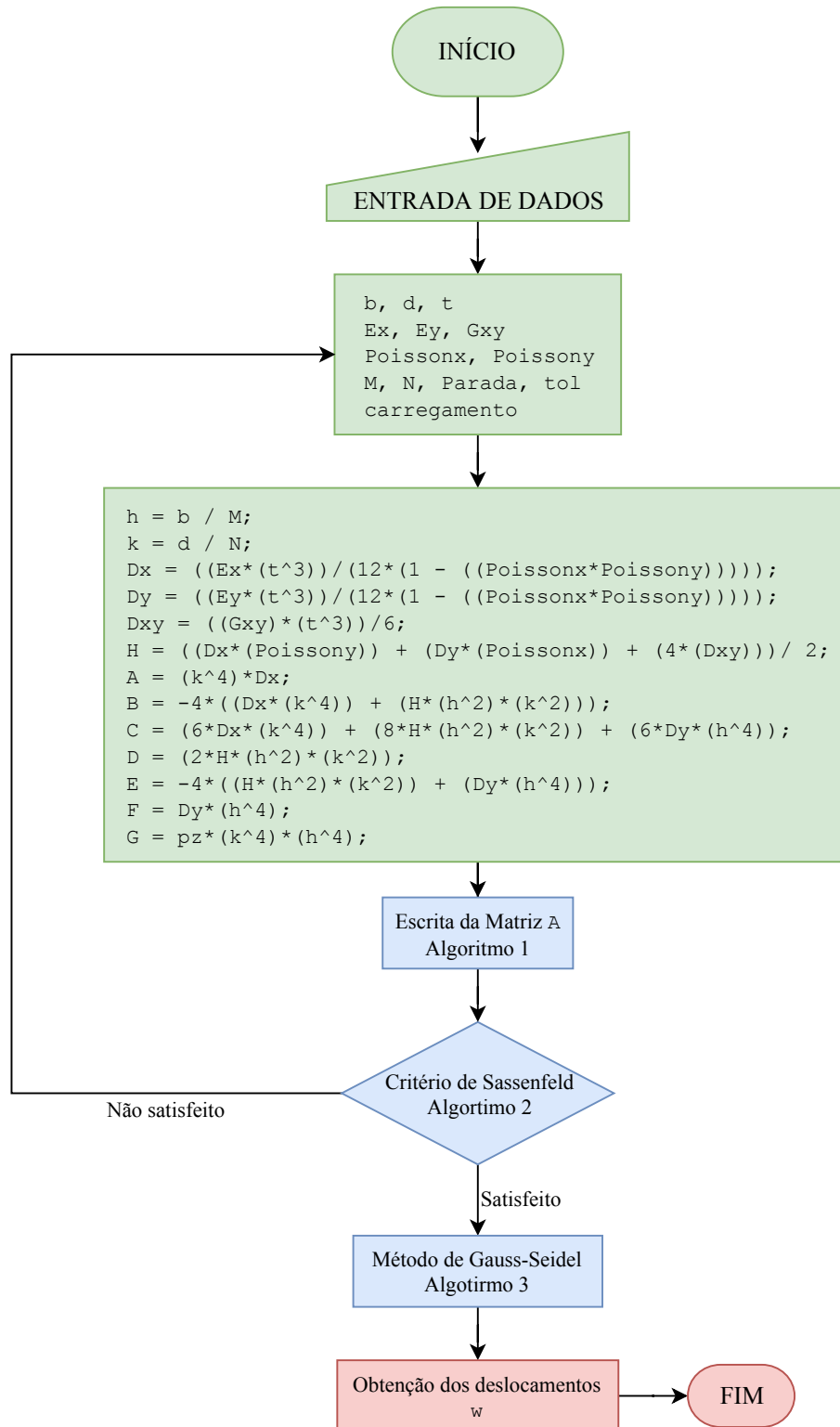
Figura 53 – Página inicial



Fonte: Autoria própria (2018).

A página inicial, conforme mostra a Figura 53, o usuário observará uma pequena descrição do que o programa irá realizar. O fluxograma ilustrado na Figura 54, apresenta o funcionamento geral da interface, em que a parte em verde representa a parte inicial do programa, que engloba a entrada de dados e definição, por parte do usuário, os valores das variáveis. A parte em azul o processamento dos Algoritmos do Anexo A e a parte em vermelho a apresentação dos resultados.

Figura 54 – Fluxograma.



Fonte: Autoria própria (2018).

Para sair desta página e continuar é necessário pressionar o botão “Iniciar”, que leva para a tela representada pela Figura 55. Aqui, inicia-se o processo de “Entrada de Dados” do fluxograma da Figura 54.

Figura 55 – Página das propriedades geométricas.

Deformação de Placas Delgadas Ortotrópicas

< Propriedades Geométricas Propriedades do Material Vinculações Parâmetros C >

### Propriedades

#### Propriedades Geométricas

Largura  
 mm

Comprimento  
 mm

Espessura  
 mm

Comprimento

Largura

< Anterior Próximo > ?

Fonte: Autoria própria (2018).

Conforme demonstra a Figura 55, as condições de contorno começarão a ser implementadas. Nesta figura, nota-se uma limitação do programa, que funciona somente com placas retangulares de espessura constante.

Nesta fase, o usuário deve preencher os espaços para que o botão de “Próximo” seja liberado, se qualquer um desses espaços for igual a zero, o usuário não conseguirá prosseguir.

Ao preencher todos os campos, o usuário é levado para a terceira tela, que é definida pela Figura 56.

Figura 56 – Página das propriedades do material isotrópico.

Deformação de Placas Delgadas Ortótropicas

< Propriedades Geométricas Propriedades do Material Vinculações Parâmetros C >

### Propriedades

#### Propriedades do Material

Isotrópico ▼

#### Módulo de Young

20000.0000 Mpa      Direção y      0.0000 Mpa

#### Módulo de Cisalhamento

8695.6500 Mpa

#### Coeficiente de Poisson

0.150      Direção y      0.000

< Anterior      Próximo >      ?

Fonte: Autoria própria (2018).

Nela, as propriedades do material da placa serão introduzidos na memória do programa. Primeiramente, o usuário terá que definir se ele quer realizar uma análise isotrópica ou ortotrópica. A primeira opção já está aparente, definida na Figura 56 e nela as propriedades  $E_x$  e  $E_y$  são iguais, assim como  $\nu_x$  e  $\nu_y$ . Na segunda opção, mais campos aparecem, conforme apresenta na Figura 57.

Figura 57 – Página das propriedades do material ortotrópico.

Deformação de Placas Delgadas Ortótropicas

< Início Propriedades Geométricas **Propriedades do Material** Vinculações Parâmetros >

## Propriedades

### Propriedades do Material

Ortotrópico ▼

#### Módulo de Young

Direção x  Mpa

Direção y  Mpa

#### Módulo de Cisalhamento

Plano xy  Mpa

#### Coeficiente de Poisson

Direção x

Direção y

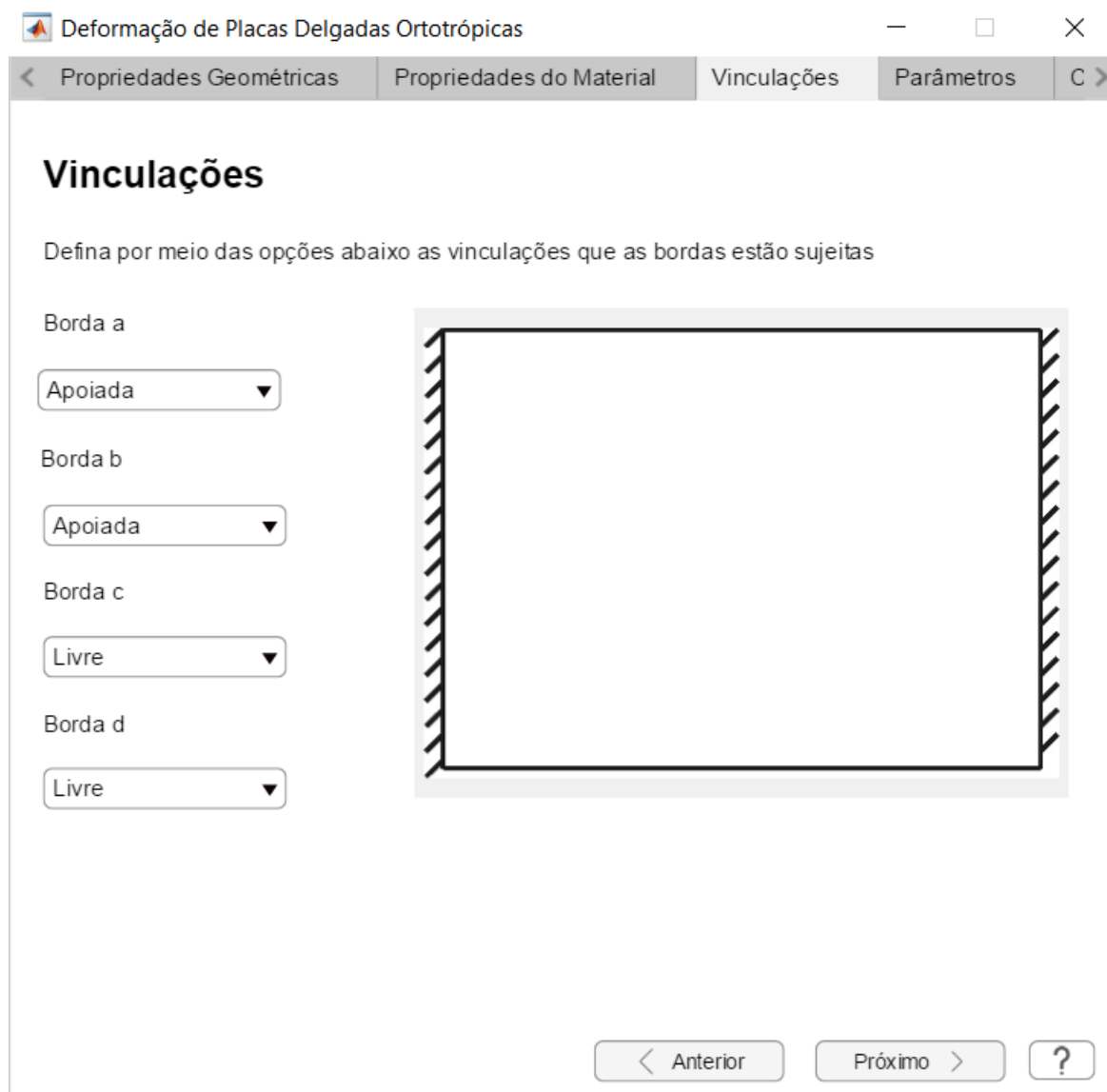
< Anterior Próximo > ?

Fonte: Autoria própria (2018).

Assim, até que todos os campos sejam preenchidos, o botão de “Próximo” não estará disponível.

Na quarta tela o usuário definirá as vinculações que a placa está submetida, cujas opções são: apoiada, engastada e livre, conforme se observa na Figura 58, porém, para realização de análises, esta versão da interface possui somente o caso de placas bi-apoiadas. Sendo a placa retangular, as quatro bordas foram nomeadas de  $a$ ,  $b$ ,  $c$  e  $d$ , seguindo a nomenclatura apresentada no Capítulo 5.

Figura 58 – Página de definições das vinculações.



Fonte: Autoria própria (2018).

Caso o usuário, por engano, coloque alguma opção em que as vinculações não sejam estáveis, o botão de "Próximo" ficará indisponível e uma mensagem aparecerá alertando sobre erro, conforme apresentado na Figura 59.

Figura 59 – Página de definições das vinculações com apresentação do erro.



Fonte: Autoria própria (2018).

Para introduzir as vinculações e tornar o programa mais intuitivo, pensou-se na mudança do desenho ao lado das opções, que mudariam conforme o usuário escolhe qual tipo de vinculação a borda possui. Para que não haja nenhuma dúvida em relação as representações, ao pressionar o botão do ponto de interrogação, posicionado na parte inferior da janela na Figura 59, o usuário se deparará com uma legenda, conforme apresentado na Figura 60.



Figura 60 – Página de definições das vinculações com apresentação da legenda.



Fonte: Autoria própria (2018).

Na quinta tela, o usuário entrará com os parâmetros necessário para aplicação do MDF. Primeiramente, terá que introduzir o número  $M$ , apresentado no Capítulo 4.3, que representa o número de subdivisões nas abscisas. O próximo passo é definir o número  $N$ , que diz respeito ao número de subdivisões nas ordenadas. A cada entrada desses dois termos, o gráfico ao lado apresentará linhas horizontais e verticais, demonstrando o plano  $\Omega$  discretizado, com  $a = b = 0$ , ou seja, fixado na origem, conforme demonstra a Figura 61.

Figura 61 – Página dos parâmetros.

Deformação de Placas Delgadas Ortotrópicas

< Propriedades Geométricas Propriedades do Material Vinculações Parâmetros C >

## Parâmetros

### Discretização do Domínio

Número de subdivisões para x

Número de subdivisões para y

### Critérios de Parada

Numero Máximo de Iterações

Erro de truncamento



< Anterior Próximo > ?

Fonte: Autoria própria (2018).

Após as entradas para a discretização do domínio será necessário introduzir os critérios de parada, que são os números de iterações e o erro de truncamento. Quando o método numérico não conseguir atingir o erro de truncamento definido pelo usuário, o número de iterações será utilizado como critério de parada para que o código não fique dentro de um *loop* infinito.

A sexta e última tela para entrada de dados, servirá para definir os carregamentos solicitantes. Na Figura 62 é possível verificar que na sexta janela, o usuário poderá definir o tipo de carregamento, cuja única opção possível, para essa versão, são os carregamentos uniformemente distribuídos.

Figura 62 – Página para introdução dos carregamentos.

Defina o tipo de carregamento solicitante

Unif. Distribuido

Módulo

0.0000 N/mm<sup>2</sup>

Coeficiente de Ponderação

0.00

Retirar + Adicionar

item	Modo do carregamento	Módulo	Coeficiente
1	Unif. Distribuido	0.0250	-1

< Anterior Próximo > ?

Fonte: Autoria própria (2018).

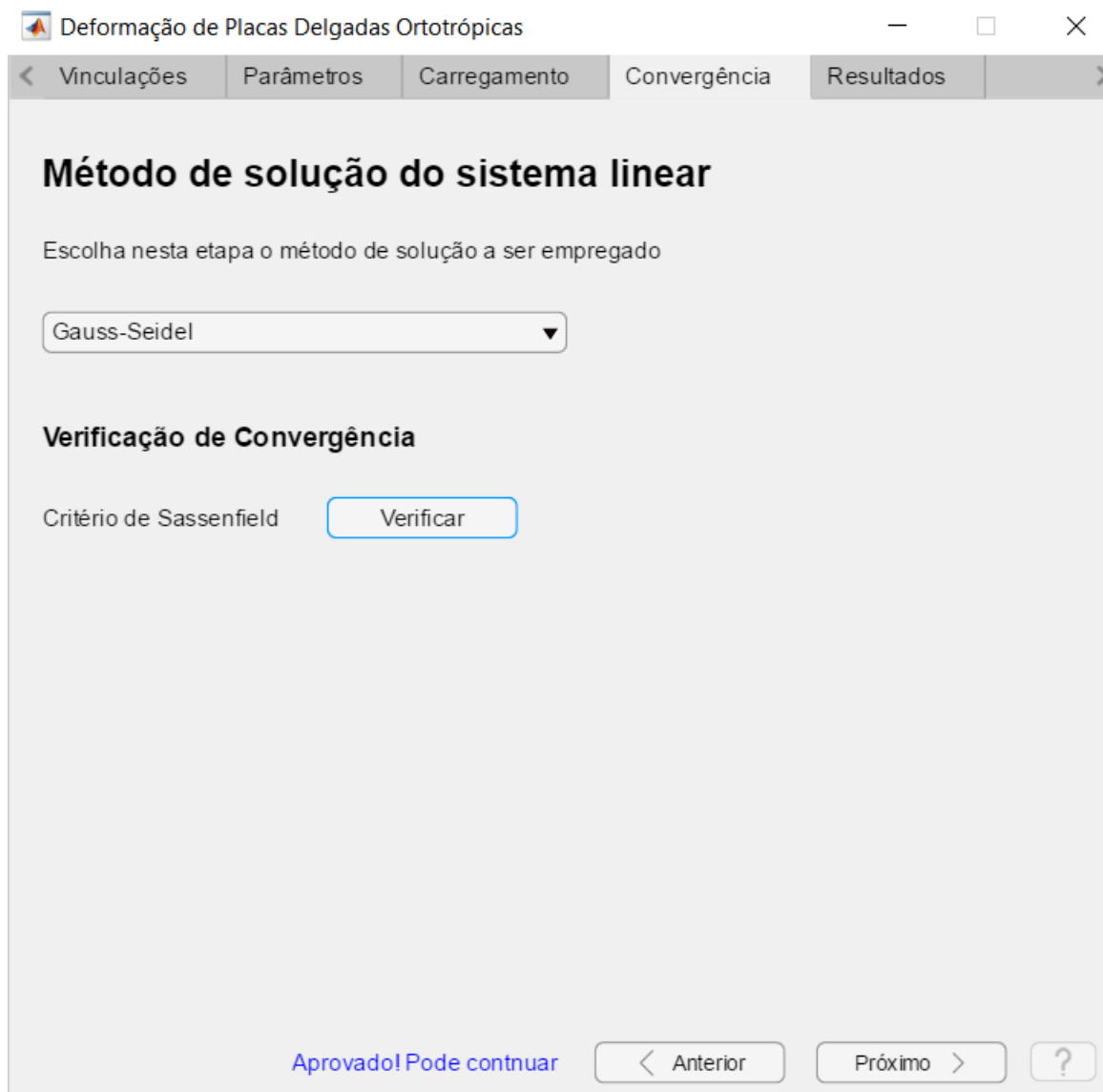
Também neste programa, é possível realizar combinações dos carregamentos. Para isso, terá que ser introduzido um coeficiente de ponderação, que será multiplicado pelo módulo do carregamento. Para que estes parâmetros sejam introduzidos na memória do programa, o usuário deve pressionar o botão “Adicionar”, permitindo que outros valores de carregamentos e de coeficientes sejam adicionados.

Os diferentes carregamentos estarão expostos na tabela apresentada na Figura 62. O usuário pode modificar, por causa de eventuais erros, a terceira e quarta coluna, que são respectivamente referentes ao módulo e ao coeficiente de ponderação dos diferentes carregamentos.

A partir deste ponto, todos os dados de entrada foram definidos, finalizando a parte em verde e iniciando a parte em azul do fluxograma. A próxima janela levará o usuário a realizar as verificações numéricas necessárias. Para tanto, ao pressionar o botão “Verificar” da Figura 63, o programa executa o Algoritmo 1, no Anexo A, que gerará a matriz dos coeficientes  $A$  do sistema de equações lineares, por conta do MDF.

Este passo é necessário por conta que a obtenção da solução do sistema será pelo método de Gauss-Seidel, sendo assim, o critério de Sassenfeld deve ser satisfeito, conforme descrito no Capítulo 4.4. Para tanto, a interface, em conjunto da formação da matriz dos coeficientes  $A$ , rodará o Algoritmo 2. Se aprovado, uma mensagem irá aparecer, conforme Figura 63, liberando o botão de “Próximo”, caso contrário, o botão permanecerá inativo e a mensagem “Negado! sistema não convergente” aparecerá.

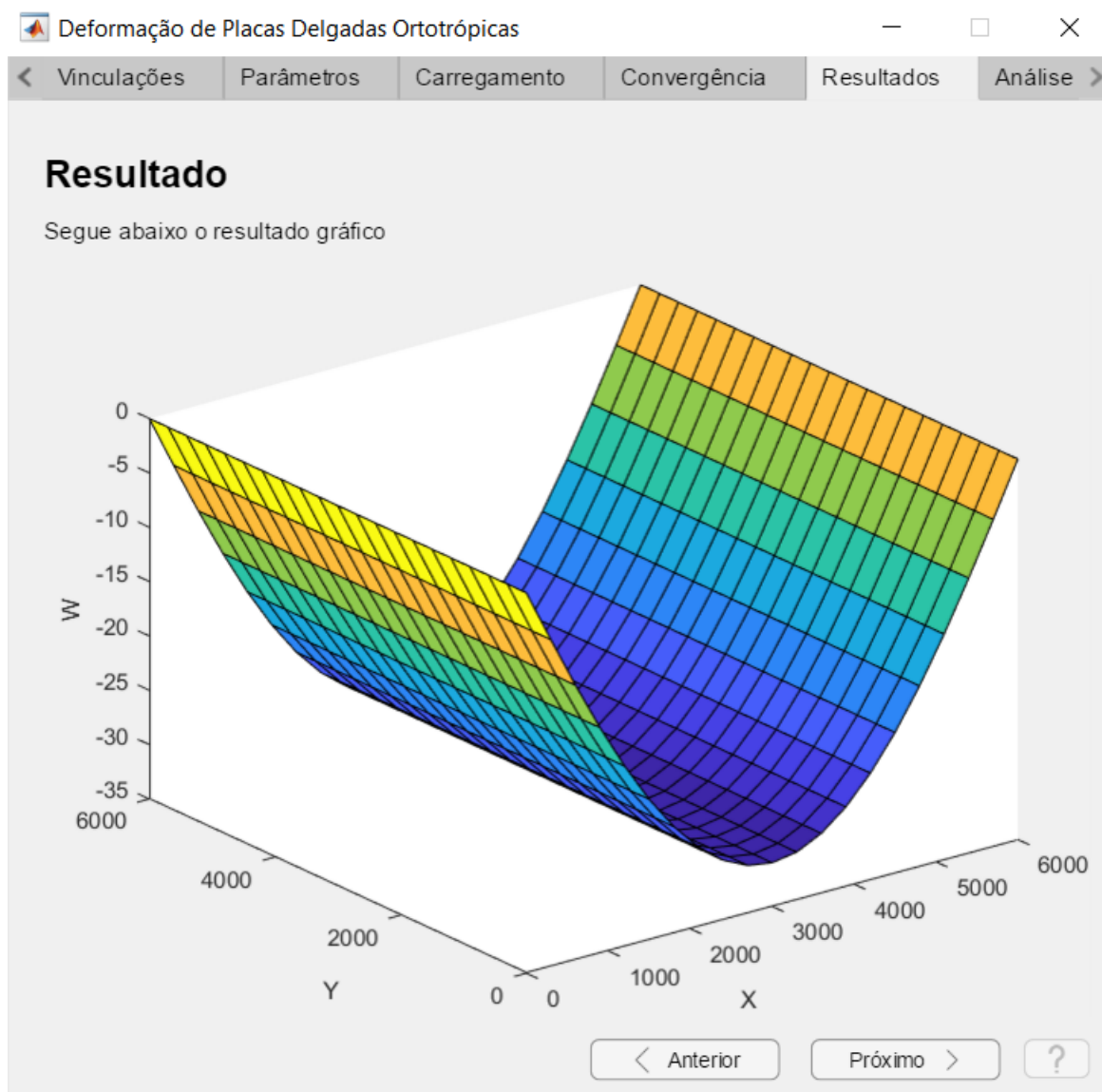
Figura 63 – Página para verificação numérica.



Fonte: Autoria própria (2018).

Ao pressionar o botão de “Próximo”, conforme apresentado na Figura 63, uma mensagem será apresentada ao usuário dizendo que o programa está solucionando o sistema conforme os dados de entrada, utilizando o Algoritmo 3. Após calculado, uma oitava janela abrirá, contendo a solução gráfica em 3D, conforme se vê na Figura 64, representando a parte em vermelho do fluxograma da Figura 54.

Figura 64 – Página para apresentação do resultado gráfico.



Fonte: Autoria própria (2018).

Para uma análise rápida, foi organizado no programa uma nona página que pode ser acessada ao pressionar o botão “Próximo” da oitava janela. Nela, o usuário poderá pesquisar a deflexão nos pontos de discretização escolhendo as coordenadas desse ponto. Para tanto, há uma lista que descreve as coordenadas dos pontos de discretização em  $x$  e uma outra lista para  $y$ . Ao pressionar o botão “Pesquisar”, a resposta aparecerá no texto logo abaixo deste botão. Nesta página ainda, o programa disponibiliza os maiores e menores valores de deflexão, com suas devidas coordenadas.

O que foi descrito acima pode ser visualizado na Figura 65.

Figura 65 – Página para apresentação do resultado para análise.

Deformação de Placas Delgadas Ortotrópicas

< nculações Parâmetros Carregamento Convergência Resultados **Análise** >

## Análise

Pesquise o valor da deflexão no ponto

Direção x: 3000

Direção y: 3000

O valor de deflexão no ponto de escolha é: **-30.991** mm

### Deflexão máxima e mínima

O valor de deflexão máxima positiva é: **0** mm

No ponto: **0** mm

**0** mm

O valor de deflexão máxima negativa é: **-30.9906** mm

No ponto: **3000** mm

**6000** mm

Fonte: Autoria própria (2018).

## 6 Verificação da solução numérica

A interface de usuário apresentado no Capítulo 5.35, utiliza o MDF para resolver numericamente a equação de placas delgadas ortotrópicas equivalentes apresentada no Capítulo 4.2. Com o intuito de verificar se a análise numérica apresenta resultados satisfatórios, alguns exemplos numéricos foram analisados utilizando o *software* comercial de elementos finitos, *Robot*, e os resultados obtidos foram comparados com os resultados apresentados pela interface.

Para tanto, foi escolhidas duas placas com relações geométricas distintas, sendo que a primeira uma relação de  $L/b = 1$  e a segunda  $L/b = 2$ , sendo  $L$  o comprimento e  $b$  a largura da placa. Foi realizada esta escolha pois, conforme Góes (2005), grande parte das pontes protendidas de madeira apresentam uma geometria em planta contida no intervalo  $1 \leq L/b \leq 2$ .

Como forma de mensurar a eficiência das aproximações será utilizado o erro relativo que é dado pela Equação (212)

$$e(\%) = \frac{(V_0 - V_t)}{V_t} \times 100 \quad (212)$$

em que  $V_0$  é o valor obtido pela interface,  $V_t$  é o valor de comparação, obtido via *Robot*.

### 6.1 Placa bi-apoiada ortotrópica de relação $L/b = 1$

Considerado uma placa ortotrópica seguindo os seguintes dados de entrada:

$$\begin{aligned} \text{Largura } (b) &= 6000 \text{ mm;} \\ \text{Comprimento } (L) &= 6000 \text{ mm;} \\ \text{Espessura} &= 200 \text{ mm;} \\ E_x = E_y &= 20000 \text{ MPa;} \\ G_{xy} &= 8695,65 \text{ MPa;} \\ \nu_{xy} = \nu_{xz} = \nu_{yz} &= 0,15. \end{aligned}$$

carregada com uma carga uniformemente distribuída  $p = 0.025 \text{ N/mm}^2$ .

O sistema gerado pelo MDF apresentado no Capítulo 5 será solucionado pelo método de Gauss-Seidel, para tanto, o Critério de Sassenfeld deve ser satisfeito, conforme descrito no Capítulo 4.4. A matriz dos coeficientes  $A$  deve ser conhecida para aplicação do critério, por tal motivo, houve a implementação do Algoritmo 1, que está no Apêndice A.



A matriz dos coeficientes  $A$  no Algoritmo 1 é dado pela matriz  $W$ . Para implementação do critério de Sassenfeld e verificar se a matriz converge para o método de Gauss-Seidel, o Algoritmo 2 no Apêndice A, foi desenvolvido. Desta maneira, a Tabela 1, apresenta os valores máximos de  $\beta$  obtidos pelo Algoritmo 2 e a deflexão máxima obtida pelo Algoritmo 3, para diferentes discretizações.

Na tabela 1, nas suas duas primeiras colunas, encontra-se as coordenadas, dadas em metros, dos valores de deflexões máximas nas diferentes discretizações. Em sua terceira e quarta coluna, há os valores de subdivisões  $M$  e  $N$  para as abscisas e ordenadas, respectivamente. Na sua quinta coluna, há a quantidade de pontos que cada discretização gera no domínio  $\Omega$ , seguindo a equação  $(M - 1) \times (N + 1)$ . Na sexta coluna há os valores de máximos de  $\beta$ , na sétima coluna há os valores de deflexão máxima, e por fim, na oitava coluna, o erro relativo, dado pela equação (212).

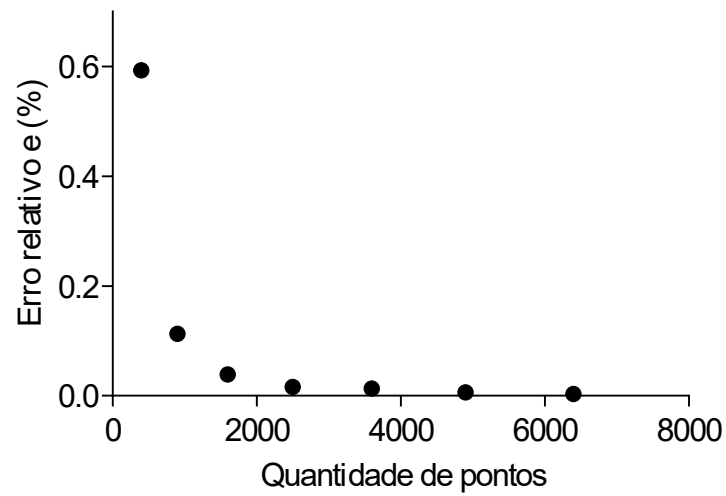
Tabela 1 – Verificação da convergência

$x$ (m)	$y$ (m)	$M$	$N$	Quantidade de pontos	$\max \beta_i$	$w_{ij}$ (mm)	$e(\%)$
3,0	3,0	10	10	99	0,8744	-31,1760	-
3,0	3,0	20	20	399	0,8744	-30,9910	0,593
3,0	3,0	30	30	899	0,8744	-30,9560	0,113
3,0	3,0	40	40	1599	0,8744	-30,9440	0,039
3,0	3,0	50	50	2499	0,8744	-30,9390	0,016
3,0	3,0	60	60	3599	0,8744	-30,9350	0,013
3,0	3,0	70	70	4899	0,8744	-30,9330	0,006
3,0	3,0	80	80	6399	0,8744	-30,9321	0,003

Fonte: Autoria própria (2018).

Ao analisar a Tabela 1, percebe-se que para maiores quantidade de pontos da malha discretizada, o erro relativo  $e$  tende a zero. Esse comportamento se torna mais claro ao analisar o gráfico representado pela Figura 66.

Figura 66 – Erro absoluto para uma placa bi-apoiada de  $L/b = 1$ .



Fonte: Autoria própria (2018).

Para fins de comparação com o *software* Robot 2018, escolheu-se  $M = N = 60$ . Na Tabela 2 há a comparação entre os dois resultados para tal malha. Nela, pode-se observar na terceira e quarta coluna, os valores de deflexão obtidos, utilizando o MDF e MEF, respectivamente, assim como que na primeira e segunda coluna suas respectivas coordenadas. Na quinta, última coluna, há o erro relativo, dado pela equação (212), comparando os resultados dados pela terceira e quarta coluna.

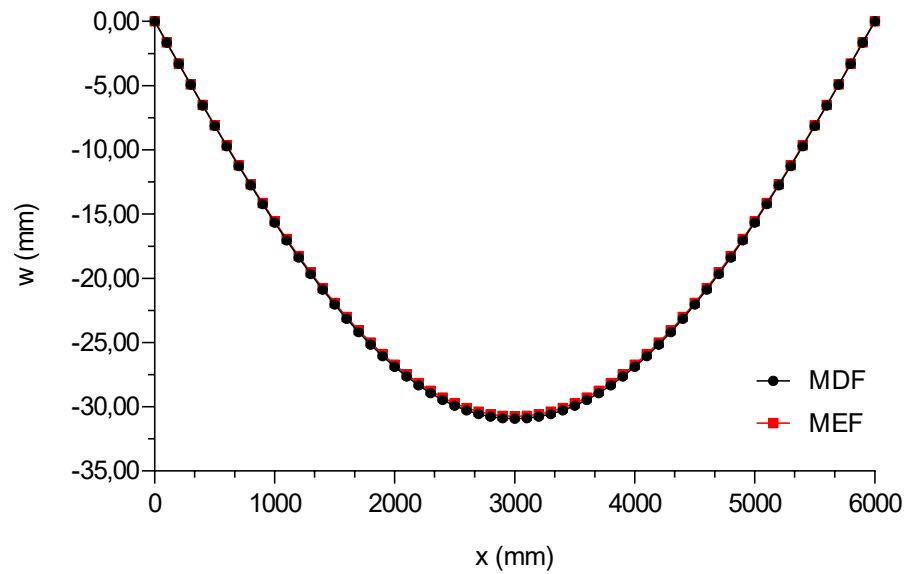
Tabela 2 – Comparação entre o MDF e MEF.

x (m)	y (m)	$w_{ij}$ (mm)		e(%)
		MEF	MDF	
0,0 = 6,0	3,0	0,000	0,000	-
1,0 = 5,9	3,0	-1,639	-1,649	0,592
2,0 = 5,8	3,0	-3,273	-3,293	0,593
3,0 = 5,7	3,0	-4,897	-4,926	0,592
4,0 = 5,6	3,0	-6,505	-6,543	0,592
5,0 = 5,5	3,0	-8,092	-8,140	0,591
6,0 = 5,4	3,0	-9,655	-9,712	0,591
7,0 = 5,3	3,0	-11,187	-11,254	0,595
8,0 = 5,2	3,0	-12,687	-12,761	0,587
9,0 = 5,1	3,0	-14,148	-14,231	0,589
1,0 = 5,0	3,0	-15,567	-15,659	0,590
1,1 = 4,9	3,0	-16,941	-17,041	0,590
1,2 = 4,8	3,0	-18,266	-18,373	0,587
1,3 = 4,7	3,0	-19,538	-19,653	0,589
1,4 = 4,6	3,0	-20,755	-20,877	0,590
1,5 = 4,5	3,0	-21,913	-22,042	0,591
1,6 = 4,4	3,0	-23,009	-23,145	0,592
1,7 = 4,3	3,0	-24,041	-24,183	0,590
1,8 = 4,2	3,0	-25,007	-25,154	0,588
1,9 = 4,1	3,0	-25,904	-26,057	0,592
2,0 = 4,0	3,0	-26,730	-26,887	0,589
2,1 = 3,9	3,0	-27,483	-27,645	0,591
2,2 = 3,8	3,0	-28,161	-28,327	0,590
2,3 = 3,7	3,0	-28,763	-28,933	0,591
2,4 = 3,6	3,0	-29,288	-29,460	0,589
2,5 = 3,5	3,0	-29,734	-29,909	0,590
2,6 = 3,4	3,0	-30,100	-30,277	0,589
2,7 = 3,3	3,0	-30,386	-30,565	0,591
2,8 = 3,2	3,0	-30,590	-30,771	0,592
2,9 = 3,1	3,0	-30,713	-30,894	0,590
3,0	3,0	-30,754	-30,935	0,589

Fonte: Autoria própria (2018).

Diante da comparação dos resultados da Tabela 2, pode-se concluir que o programa se mostra eficiente na solução deste problema, obtendo uma aproximação satisfatória em relação ao MEF, expressando um erro relativo de ordem  $e = 0,59\%$ . A representação gráfica da Tabela 2 está na Figura 67.

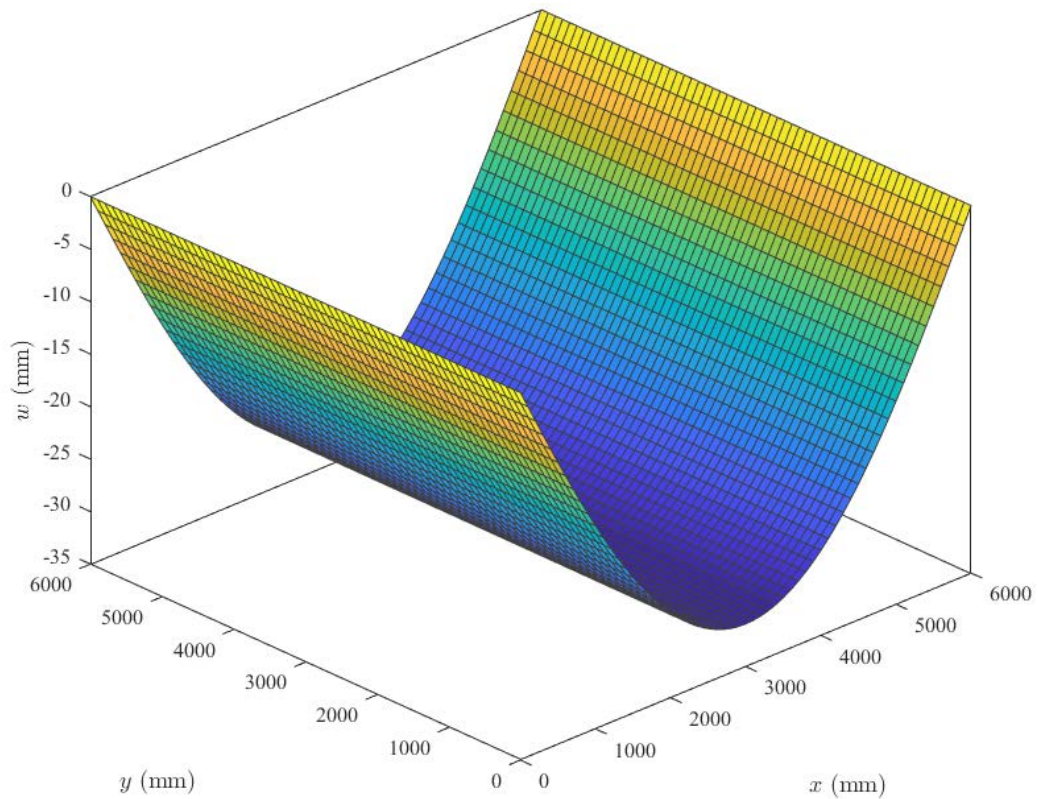
Figura 67 – Comparação entre MDF e MEF.



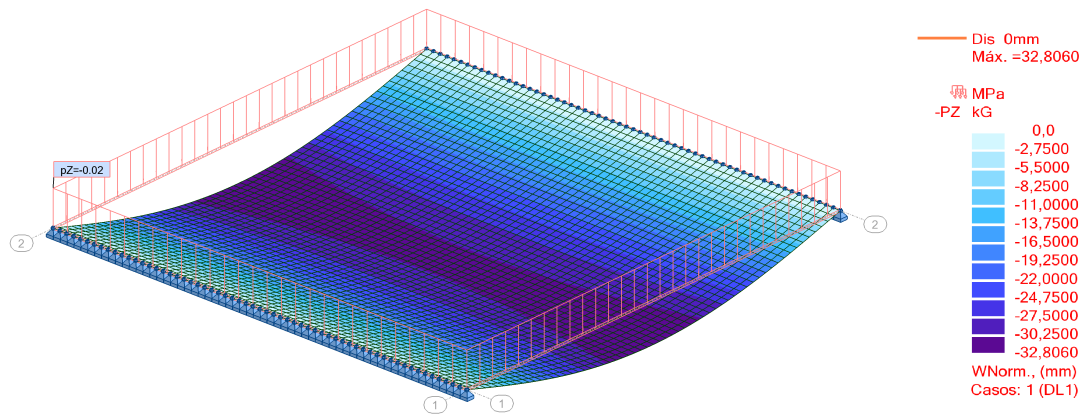
Fonte: Autoria própria (2018).

A Figura 68 apresenta o resultado gráfico do MDF dado pela interface enquanto que a Figura 69 o resultado gráfico dado pelo *software Robot 2018*.

Figura 68 – Representação gráfica da solução por MDF, com  $M = N = 60$ .



Fonte: Autoria própria (2018).

Figura 69 – Representação gráfica dada pelo *software Robot 2018*.

Fonte: Autoria própria (2018).

Vale ressaltar que mesmo que ambos os gráficos apresentem escalas de cores diferentes, os resultados são bastante próximos, conforme se confirma pela Tabela 2.

## 6.2 Placa bi-apoiada ortotrópica de relação $L/b = 2$ .

Considerado uma placa ortotrópica seguindo os seguintes dados de entrada:

$$\begin{aligned}
 \text{Largura } (b) &= 6000 \text{ mm;} \\
 \text{Comprimento } (L) &= 12000 \text{ mm;} \\
 \text{Espessura} &= 400 \text{ mm;} \\
 E_x = E_y &= 20000 \text{ MPa;} \\
 G_{xy} &= 8695,65 \text{ MPa;} \\
 \nu_{xy} = \nu_{xz} = \nu_{yz} &= 0,15.
 \end{aligned}$$

carregada com uma carga uniformemente distribuída  $p = 0.025 \text{ N/mm}^2$ .

Seguindo os mesmos procedimentos descritos em placas com  $L/b = 1$ , a Tabela 3 apresentará os valores máximos de  $\beta$  obtidos pelo Algoritmo 2 e a deflexão máxima obtida pelo Algoritmo 3, para diferentes discretizações.

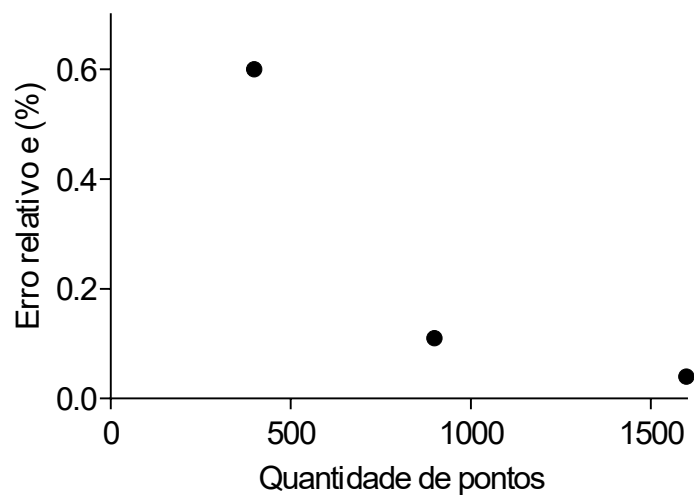
Tabela 3 – Verificação da convergência

$x$ (m)	$y$ (m)	$M$	$N$	Quantidade de pontos	$\max \beta_i$	$w_{ij}$ (mm)	$e(\%)$
3	3	10	10	99	0,9459	-62,3520	-
3	3	20	20	399	0,9459	-61,9810	0,60%
3	3	30	30	899	0,9459	-61,9120	0,11%
3	3	40	40	1599	0,9459	-61,8880	0,04%

Fonte: Autoria própria (2018).

Ao analisar a Tabela 3, há a convergência do erro relativo  $e$  a zero, que pode ser também visto ao analisar o gráfico representado pela Figura 70.

Figura 70 – Erro absoluto para uma placa bi-apoiada de  $L/b = 2$ .



Fonte: Autoria própria (2018).

Para fins de comparação com o *software* Robot 2018, escolheu-se  $M = N = 40$ . Na Tabela 4 há a comparação entre os dois resultados para tal malha.

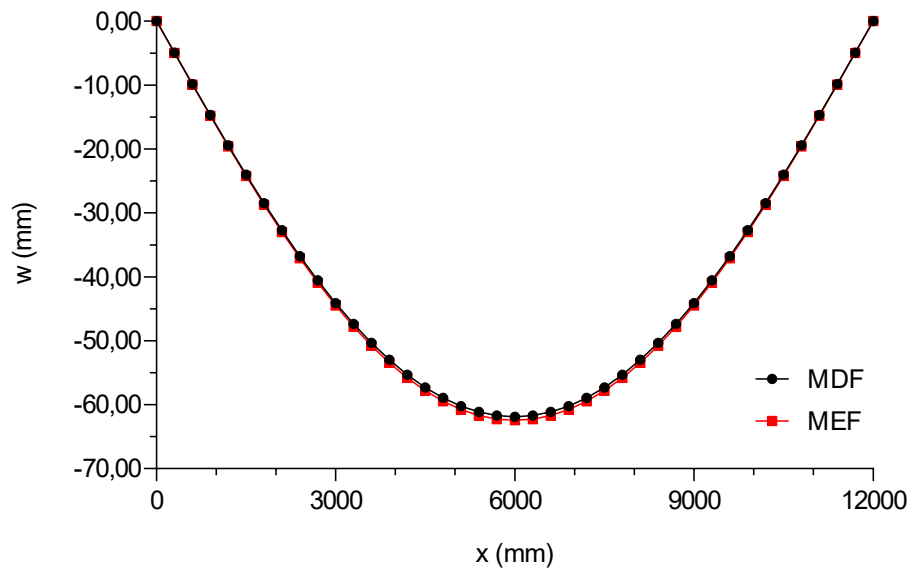
Tabela 4 – Comparação entre o MDF e MEF

x (m)	y (m)	$w_{ij}$ (mm)		e(%)
		MEF	MDF	
0,0 = 12,0	3,0	0	0	-
0,3 = 11,7	3,0	4,9846	4,9455	0,7844
0,6 = 11,4	3,0	9,9337	9,8548	0,7943
0,9 = 11,1	3,0	14,8125	14,694	0,8000
1,2 = 10,8	3,0	19,5881	19,429	0,8122
1,5 = 10,5	3,0	24,2295	24,032	0,8151
1,8 = 10,2	3,0	28,7075	28,471	0,8238
2,1 = 9,9	3,0	32,9948	32,722	0,8268
2,4 = 9,6	3,0	37,0661	36,758	0,8312
2,7 = 9,3	3,0	40,8981	40,557	0,8340
3,0 = 9,0	3,0	44,469	44,097	0,8365
3,3 = 8,7	3,0	47,7592	47,358	0,8400
3,6 = 8,4	3,0	50,7509	50,323	0,8431
3,9 = 8,1	3,0	53,4282	52,977	0,8445
4,2 = 7,8	3,0	55,7769	55,305	0,8460
4,5 = 7,5	3,0	57,785	57,295	0,8480
4,8 = 7,2	3,0	59,442	58,938	0,8479
5,1 = 6,9	3,0	60,7395	60,224	0,8487
5,4 = 6,6	3,0	61,6711	61,147	0,8498
5,7 = 6,3	3,0	62,2319	61,703	0,8499
6,0	3,0	62,4191	61,888	0,8509

Fonte: Autoria própria (2018).

Diante da comparação dos resultados da Tabela 4, pode-se concluir que o programa se mostra eficiente na solução deste problema, obtendo uma aproximação satisfatória em relação ao MEF, expressando um erro relativo de ordem  $e = 0,85\%$ . A representação gráfica da Tabela 4 está na Figura 71.

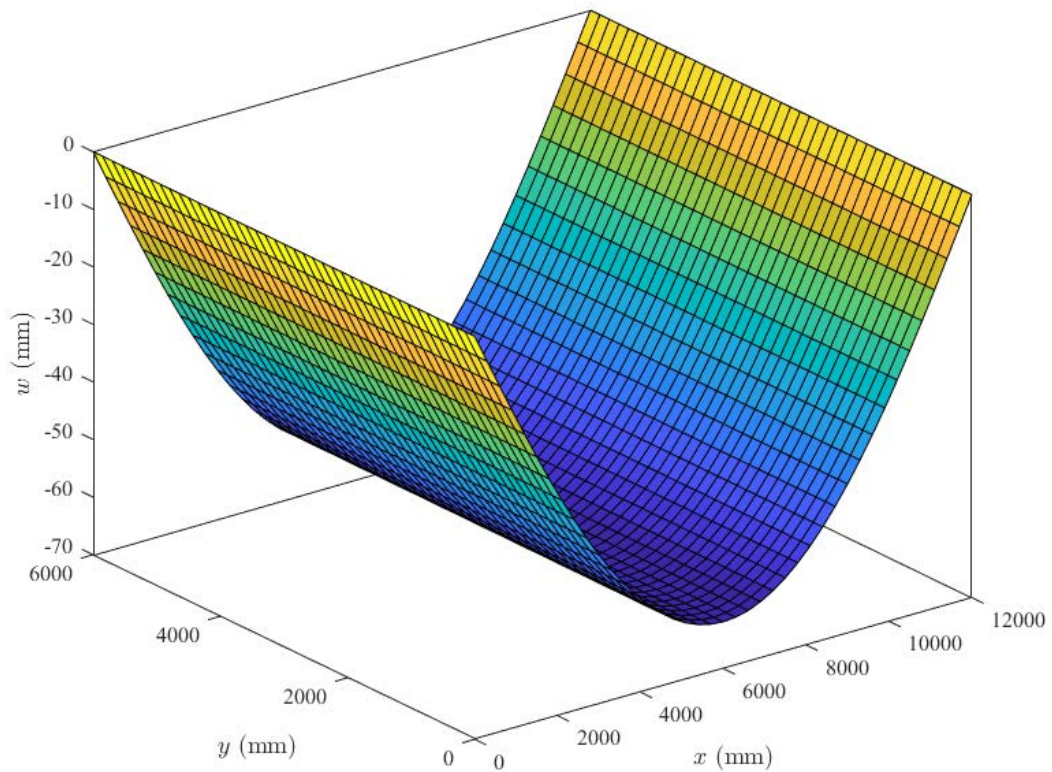
Figura 71 – Comparação entre MDF e MEF.



Fonte: Autoria própria (2018).

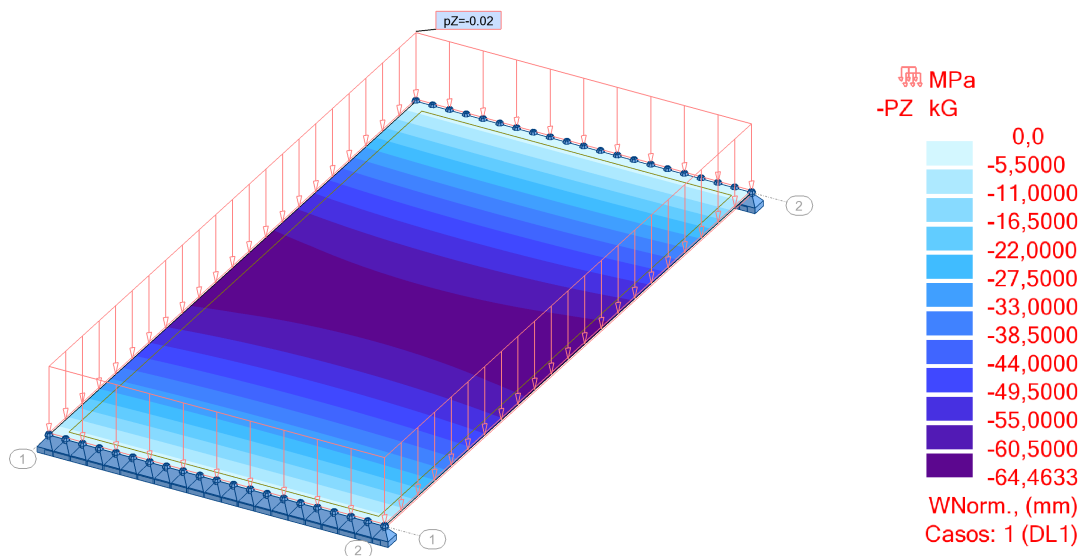
A Figura 72 apresenta o resultado gráfico do MDF dado pela interface enquanto que a Figura 73 o resultado gráfico dado pelo *software Robot 2018*.

Figura 72 – Representação gráfica da solução por MDF, com  $M = N = 40$ .



Fonte: Autoria própria (2018).



Figura 73 – Representação gráfica dada pelo *software Robot 2018*.

Fonte: Autoria própria (2018).

Vale ressaltar que mesmo que ambos os gráficos apresentem escalas de cores diferentes, os resultados são bastante próximos, conforme se confirma na Tabela 4.

## 7 Considerações finais e conclusão

Neste trabalho, foram realizados estudos teóricos sobre a deflexão de placas ortotrópicas, com o objetivo principal, a apresentação e implementação computacional de uma formulação baseada no MDF.

Para tanto, foi levado em consideração os quocientes de diferenças, o erro de truncamento, os critérios de convergência, a Teoria de placas delgadas ortotrópicas e os demais conceitos tratados ao longo deste trabalho.

Após ter encontrado um modelo matemático para deflexão de placas baseado no MDF, foi elaborado a interface baseada em linguagem *Matlab*<sup>®</sup> para a solução do sistema linear e análise dos resultados.

Em seguida, foram abordados alguns exemplos de geometrias distintas e comparados com a modelagem feita no *software Robot*. Alguns erros relativos puderam ser observados, porém, o modelo de solução proposto para o problema é válido pelo fato de que a ordem deste é relativamente baixo, assim os resultados são adequados e aceitáveis.

Por fim, como sugestão para a elaboração de trabalhos futuros, pelos excelentes resultados que o método, aqui empregado, obteve, pode-se citar o emprego de diferentes carregamentos e vinculações, para a criação de uma interface mais completa.

## Referências

- BONO, G. F. F. *Modelos constitutivos para análise tridimensional de estruturas de concreto armado através do método dos elementos finitos*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, abr. 2008.
- BURDEN, R.; FAIRES, J. *Numerical Analysis*. 9. ed. [S.l.]: Cengage Learning, 2010. ISBN 978-0-538-7351-9.
- CORRÊA, M. R. S. *Aperfeiçoamento de modelos usualmente empregados no projeto de sistemas estruturais de edifícios*. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 1991.
- CUMINATO, J. A.; JUNIOR, M. M. *Discretização de Equações Diferenciais Parciais: Técnica de Diferenças Finitas*. São Carlos: Universidade Estadual Paulista, 2011.
- CUSENS, A.; PAMA, R. *Bridge Deck Analysis*. [S.l.]: Wiley-Interscience, 1975. ISBN 0-471-18998-7.
- EULER, L. De motu vibratorio tympanorum. *Novi Commentarii academiae scientiarum Petropolitanae* 10, p. 243–260, 1766.
- FORTUNA, A. d. O. *Técnicas Computacionais para Dinâmica dos Fluidos: Conceitos Básicos e Aplicações*. São Paulo: Editora da Universidade de São Paulo, 2000. (II). ISBN 85-314-0526-2.
- FRANCA, A. B. M. d. F.; FUSCO, P. B. *As lajes nervuradas na moderna construção de edifícios*. São Paulo: AFALA & ABRAPEX, 1997.
- FRANCO, N. B. *Cálculo Numérico*. 1. ed. São Paulo: [s.n.], 2006.
- GALAMBOS, T. V.; ELLINGWOOD, B. Serviceability limit states: Deflection. *American Journal of Civil Engineering*, v. 112, p. 67–84, jan. 1986.
- GHODS, A.; MIR, M. Analysis of rectangular thin plates by using finite difference method. *Journal of Novel Applied Sciences*, p. 260–267, mar. 2014. ISSN 2322-5149.
- GÓES, J. L. N. *Análise de Pontes de Madeira Protendidas Transversalmente com Seção Caixa*. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 2005.
- GÓES, L. N. d. et al. Programa para análise de placas ortotrópicas. *Minerva - Pesquisa & Tecnologia*, v. 3, n. 1, p. 9–21, 2006.
- GRAY, W. H.; SCHNURR, N. A comparison of the finite element and finite difference methods for the analysis of steady two dimensional heat conduction problems. *Computer Methods in Applied Mechanics and Engineering*, v. 6, p. 243–245, jan. 1975.
- HIBBELER, R. C. *Resistência dos Materiais*. 5. ed. São Paulo: Pearson Prentice Hall, 2003. ISBN 85-87918-67-2.
- JUNIOR, D. H. *Sistema treliçado espacial: uma solução de vanguarda*. Pelotas, 2017.

- JÚNIOR, S. D. C. *Elaboração de um programa computacional para o cálculo e dimensionamento de lajes maciças*. 2008. Trabalho de Conclusão de Curso (Graduação) – Engenharia Civil, Universidade Federal de Juiz de Fora, 2008.
- LEAL, C. E. *Formulação do Método dos Elementos Finitos para Análise Elástica Linear de Placas Delgadas*. 2015. Trabalho de Conclusão de Curso (Graduação) – Engenharia Civil, Universidade Tecnológica Federal do Paraná. Campo Mourão, 2015.
- NEVES, A. G. M. *Introdução às equações diferenciais parciais através da discretização*. Salvador: [s.n.], 2004.
- OKIMOTO, F. S. O.; JUNIOR, C. C. Pontes protendidas de madeira. *Cadernos de Engenharia de Estruturas*, São Carlos, n. 18, p. 25–48, 2001.
- ORTEGA, J. M. *Numerical Analysis: a second course*. [S.l.]: Society for Industrial and Applied Mathematics, 1932. v. 3. ISBN 0-89871-250-5.
- PINHEIRO, L. M.; MUZARD, C. D.; SANTOS, S. P. *Fundamentos do concreto e projeto de edifícios*. São Carlos: Universidade de São Paulo, 2010.
- ROKNUZZAMAN et al. Analysis of rectangular plate with opening by finite difference method. *American Journal of Civil Engineering and Architecture*, v. 3, n. 5, p. 165–173, 2015.
- SANTOS, R. E. d. A cultura do concreto armado no brasil: educação e deseducação dos produtores do espaço construído. *Anais do IV Congresso Brasileiro de História da Educação*, Porto Alegre, 2006.
- SOUZA, A. C. d. *Dimensionamento de elementos e ligações em estruturas em aço*. São Carlos: EdUFSCar, 2017. ISBN 978-85-7600-467-7.
- SZILARD, R. *Theories and applications of plate analysis*. [S.l.]: John Wiley & Sons, Inc., 2004. ISBN 0-471-42989-9.
- TIMOSHENKO, S.; WOINOWSKY-KRIEGER, S. *Theory of plates and shells*. [S.l.]: McGraw-Hill Book Company, 1959. (Engineering societies monographs). ISBN 0-07-064779-8.
- VARGA, R. S. *Matrix Iterativa Analysis*. [S.l.]: Springer Series in Computational Mathematics, 2009. ISBN 0179-3632.
- VILLAÇA, S. F.; GARCIA, L. F. T. *Introdução à Teoria da Elasticidade*. 3. ed. Rio de Janeiro: COPPE, 1998. v. 1. ISBN 85-285-0025-X.
- ZILL, D. G.; CULLEN, M. R. *Equações Diferenciais*. 3. ed. São Paulo: Person Education do Brasil, 2001. v. 1. ISBN 85-346-1291-9.

# Apêndices

## Apêndice A – Algoritmos

### Algoritmo 1 – Definição da matriz dos coeficientes $A$ .

---

```

syms A B C D E F

b = app.bField.Value;
d = app.dField.Value;
t = app.tField.Value;
choicematerial = app.DropDown.Value;

if strcmp(choicematerial , 'Isotropico') == 1
    Ex = app.EditField_4.Value;
    Ey = Ex;
    Gxy = app.EditField_6.Value;
    Poissonx = app.EditField_8.Value;
    Poissony = Poissonx;
elseif strcmp(choicematerial , 'Ortotropico') == 1
    Ex = app.EditField_4.Value;
    Ey = app.EditField_5.Value;
    Gxy = app.EditField_6.Value;
    Poissonx = app.EditField_8.Value;
    Poissony = app.EditField_9.Value;
end

bacc = app.DropDown_2.Value;
bbcc = app.DropDown_3.Value;
bccc = app.DropDown_4.Value;
bdcc = app.DropDown_5.Value;

M = app.EditField_14.Value;
N = app.EditField_11.Value;

Parada = app.EditField_12.Value;
tol = app.EditField_13.Value;

%%
carregamento = get(app.UITable , 'Data');
carregamento(:,2) = [];
carregamento = cell2mat(carregamento);
Modulo = carregamento(:,2);
cp = carregamento(:,3);
item = carregamento(:,1);

```

```

[row, column] = size(item);
i = 1;
pz = 0;
while i <= row
    pza = Modulo(i)*cp(i);
    pz = pza + pz;
    i = i + 1;
end

%constantes que serao utilizadas ao longo do programa
h = b / M ;%passo em x
k = d / N ;%passo em y

Dx = ((Ex*(t^3))/(12*(1 - ((Poissonx*Poissony)))));
Dy = ((Ey*(t^3))/(12*(1 - ((Poissonx*Poissony)))));
Dxy = ((Gxy)*(t^3))/6;
H = ((Dx*(Poissony)) + (Dy*(Poissonx)) + (4*(Dxy)))/ 2;

A = (k^4)*Dx;
B = -4*((Dx*(k^4)) + (H*(h^2)*(k^2)));
C = (6*Dx*(k^4)) + (8*H*(h^2)*(k^2)) + (6*Dy*(h^4));
D = (2*H*(h^2)*(k^2));
E = -4*((H*(h^2)*(k^2)) + (Dy*(h^4)));
F = Dy*(h^4);
G = pz*(k^4)*(h^4);

%Aqui inicia-se a escrita da matriz A
new = [];

if strcmp(bacc, 'Apoiada') == 1
    bacc = 1;
elseif strcmp(bacc, 'Engastada') == 1
    bacc = 2;
elseif strcmp(bacc, 'Livre') == 1
    bacc = 3;
end

if strcmp(bbcc, 'Apoiada') == 1
    bbcc = 1;
elseif strcmp(bbcc, 'Engastada') == 1
    bbcc = 2;
elseif strcmp(bbcc, 'Livre') == 1
    bbcc = 3;
end

```

```

elseif strcmp(bbcc, 'Escolha a Vinculacao') == 1
    bbcc = 4;
end

if strcmp(bccc, 'Apoiada') == 1
    bccc = 1;
elseif strcmp(bccc, 'Engastada') == 1
    bccc = 2;
elseif strcmp(bccc, 'Livre') == 1
    bccc = 3;
elseif strcmp(bccc, 'Escolha a Vinculacao') == 1
    bccc = 4;
end

if strcmp(bdcc, 'Apoiada') == 1
    bdcc = 1;
elseif strcmp(bdcc, 'Engastada') == 1
    bdcc = 2;
elseif strcmp(bdcc, 'Livre') == 1
    bdcc = 3;
elseif strcmp(bdcc, 'Escolha a Vinculacao') == 1
    bdcc = 4;
end

if bacc == 3
    if bbcc == 3
        if bccc == 3
            if bdcc == 3
                set(app.PrximoButton_3, 'Enable', 'off')
                set(app.OpoInvlidaLabel, 'Visible', 'on')
            elseif bdcc == 1
                set(app.PrximoButton_3, 'Enable', 'off')
                set(app.OpoInvlidaLabel, 'Visible', 'on')
            else
                set(app.PrximoButton_3, 'Enable', 'on')
                set(app.OpoInvlidaLabel, 'Visible', 'off')
            end
        elseif bccc == 1
            if bdcc == 3
                set(app.PrximoButton_3, 'Enable', 'off')
                set(app.OpoInvlidaLabel, 'Visible', 'on')
            else
                set(app.PrximoButton_3, 'Enable', 'on')
            end
        end
    end
end

```



```

        set(app.OpoInvlidaLabel, 'Visible', 'off')
    end
else
    set(app.PrximoButton_3, 'Enable', 'on')
    set(app.OpoInvlidaLabel, 'Visible', 'off')
end
elseif bbcc == 1
    if bccc == 3
        if bdcc == 3
            set(app.PrximoButton_3, 'Enable', 'off')
            set(app.OpoInvlidaLabel, 'Visible', 'on')
        else
            set(app.PrximoButton_3, 'Enable', 'on')
            set(app.OpoInvlidaLabel, 'Visible', 'off')
        end
    else
        set(app.PrximoButton_3, 'Enable', 'on')
        set(app.OpoInvlidaLabel, 'Visible', 'off')
    end
else
    set(app.PrximoButton_3, 'Enable', 'on')
    set(app.OpoInvlidaLabel, 'Visible', 'off')
end
elseif bacc == 1
    if bbcc == 3
        if bccc == 3
            if bdcc == 3
                set(app.PrximoButton_3, 'Enable', 'off')
                set(app.OpoInvlidaLabel, 'Visible', 'on')
            else
                set(app.PrximoButton_3, 'Enable', 'on')
                set(app.OpoInvlidaLabel, 'Visible', 'off')
            end
        else
            set(app.PrximoButton_3, 'Enable', 'on')
            set(app.OpoInvlidaLabel, 'Visible', 'off')
        end
    else
        set(app.PrximoButton_3, 'Enable', 'on')
        set(app.OpoInvlidaLabel, 'Visible', 'off')
    end
else
    set(app.PrximoButton_3, 'Enable', 'on')

```

```

        set(app.OpoInvlidaLabel, 'Visible', 'off')
    end

    if bacc == 1
        if bbcc == 1
            if bccc == 1
                if bdcc == 1

                    for j = 2:N
                        for i = 2:M
                            old = new;
                            c = sprintf('%d,%d', i, j);
                            new = [old; c];
                            new = convertCharsToStrings(new);
                        end
                    end
                    tamanho = (M-1)*(N-1);

                elseif bdcc == 2
                    for j = 2:N
                        for i = 2:M
                            old = new;
                            c = sprintf('%d,%d', i, j);
                            new = [old; c];
                            new = convertCharsToStrings(new);
                        end
                    end
                    tamanho = (M-1)*(N-1);
                elseif bdcc == 3

                    for j = 2:N+1
                        for i = 2:M
                            old = new;
                            c = sprintf('%d,%d', i, j);
                            new = [old; c];
                            new = convertCharsToStrings(new);
                        end
                    end
                    tamanho = (M-1)*(N);
                end
            elseif bccc == 2
                if bdcc == 1

```

```

    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);

elseif bdcc == 2

    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);

elseif bdcc == 3

    for j = 2:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);

end

elseif bccc == 3
    if bdcc == 1

        for j = 1:N
            for i = 2:M
                old = new;

```

```

        c = sprintf('%d,%d',i,j);
        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
tamanho = (M-1)*(N);

elseif bdcc == 2

    for j = 1:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);

elseif bdcc == 3

    for j = 1:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N+1);

end
end
elseif bbcc == 2
    if bccc == 1
        if bdcc == 1

            for j = 2:N
                for i = 2:M
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
        end
    end
end

```

```

        end
    end
    tamanho = (M-1)*(N-1);

elseif bdcc == 2
    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);

elseif bdcc == 3

    for j = 2:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);
end

elseif bccc == 2
    if bdcc == 1
        for j = 2:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N-1);
    elseif bdcc == 2
        for j = 2:N
            for i = 2:M
                old = new;

```

```

        c = sprintf('%d,%d',i,j);
        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
tamanho = (M-1)*(N-1);
elseif bdcc == 3
    for j = 2:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);
end
elseif bccc == 3
    if bdcc == 1
        for j = 1:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N);
    elseif bdcc == 2
        for j = 1:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N);
    elseif bdcc == 3
        for j = 1:N+1
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);

```

```

        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
    tamanho = (M-1)*(N+1);
end
end
elseif bbcc == 3
    if bccc == 1
        if bdcc == 1
            for j = 2:N
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N-1);
        elseif bdcc == 2
            for j = 2:N
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N-1);
        elseif bdcc == 3
            for j = 2:N+1
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N);
        end
    elseif bccc == 2
        if bdcc == 1
            for j = 2:N

```

```

        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N-1);
elseif bdcc == 2
    for j = 2:N
        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N-1);
elseif bdcc == 3
    for j = 2:N+1
        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N);
end
elseif bccc == 3
    if bdcc == 1
        for j = 1:N
            for i = 2:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    elseif bdcc == 2
        for j = 1:N
            for i = 2:M+1

```



```

        old = new;
        c = sprintf('%d,%d',i,j);
        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
    tamanho = (M)*(N);
elseif bdcc == 3

    end
end
    tamanho = (M)*(N);
elseif bdcc == 3

    end
end
    tamanho = (M)*(N);
elseif bacc == 2
    if bbcc == 1
        if bccc == 1
            if bdcc == 1

                for j = 2:N
                    for i = 2:M
                        old = new;
                        c = sprintf('%d,%d',i,j);
                        new = [old;c];
                        new = convertCharsToStrings(new);
                    end
                end
                tamanho = (M-1)*(N-1);
            elseif bdcc == 2

                for j = 2:N
                    for i = 2:M
                        old = new;
                        c = sprintf('%d,%d',i,j);
                        new = [old;c];
                        new = convertCharsToStrings(new);
                    end
                end
                tamanho = (M-1)*(N-1);
            elseif bdcc == 3

                for j = 2:N+1

```

```

        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);
end
elseif bccc == 2
    if bdcc == 1
        for j = 2:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N-1);
    elseif bdcc == 2
        for j = 2:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N-1);
    elseif bdcc == 3
        for j = 2:N+1
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N);
    end
elseif bccc == 3
    if bdcc == 1

```

```

    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);
elseif bdcc == 2
    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);
elseif bdcc == 3
    for j = 1:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N+1);
end
end
elseif bbcc == 2
    if bccc == 1
        if bdcc == 1
            for j = 2:N
                for i = 2:M
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M-1)*(N-1);

```

```

elseif bdcc == 2
    for j = 2:N
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N-1);
elseif bdcc == 3
    for j = 2:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);
end
elseif bccc == 2
    if bdcc == 1
        for j = 2:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N-1);
    elseif bdcc == 2
        for j = 2:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N-1);
    elseif bdcc == 3

```

```

    for j = 2:N+1
        for i = 2:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M-1)*(N);
end
elseif bccc == 3
    if bdcc == 1
        for j = 1:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N);
    elseif bdcc == 2
        for j = 1:N
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N);
    elseif bdcc == 3
        for j = 1:N+1
            for i = 2:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M-1)*(N+1);
    end
end
end
end

```

```

elseif bccc == 3
    if bccc == 1
        if bdcc == 1
            for j = 2:N
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N-1);
        elseif bdcc == 2
            for j = 2:N
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N-1);
        elseif bdcc == 3
            for j = 2:N+1
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N);
        end
    elseif bccc == 2
        if bdcc == 1
            for j = 2:N
                for i = 2:M+1
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
        end
    end
end

```

```

        tamanho = (M)*(N-1);
elseif bdcc == 2
    for j = 2:N
        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N-1);
elseif bdcc == 3
    for j = 2:N+1
        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N);
end
elseif bccc == 3
    if bdcc == 1
        for j = 1:N
            for i = 2:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    elseif bdcc == 2
        for j = 1:N
            for i = 2:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    end
end
tamanho = (M)*(N);

```

```

elseif bdcc == 3
    for j = 1:N+1
        for i = 2:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N+1);
end
end
elseif bacc == 3
    if bbcc == 1
        if bccc == 1
            if bdcc == 1
                for j = 2:N
                    for i = 1:M
                        old = new;
                        c = sprintf('%d,%d',i,j);
                        new = [old;c];
                        new = convertCharsToStrings(new);
                    end
                end
                tamanho = (M)*(N-1);
            elseif bdcc == 2

                for j = 2:N
                    for i = 1:M
                        old = new;
                        c = sprintf('%d,%d',i,j);
                        new = [old;c];
                        new = convertCharsToStrings(new);
                    end
                end
                tamanho = (M)*(N-1);
            elseif bdcc == 3

                for j = 2:N+1
                    for i = 1:M
                        old = new;
                        c = sprintf('%d,%d',i,j);

```



```

        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
tamanho = (M)*(N);
end
elseif bccc == 2
    if bdcc == 1

        for j = 2:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N-1);
    elseif bdcc == 2

        for j = 2:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N-1);
    elseif bdcc == 3

        for j = 2:N+1
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    end
elseif bccc == 3
    if bdcc == 1

```

```

    for j = 1:N
        for i = 1:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N);
elseif bdcc == 2

    for j = 1:N
        for i = 1:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N);
elseif bdcc == 3

    end
end
elseif bbcc == 2
    if bccc == 1
        if bdcc == 1
            for j = 2:N
                for i = 1:M
                    old = new;
                    c = sprintf('%d,%d',i,j);
                    new = [old;c];
                    new = convertCharsToStrings(new);
                end
            end
            tamanho = (M)*(N-1);

        elseif bdcc == 2
            for j = 2:N
                for i = 1:M
                    old = new;
                    c = sprintf('%d,%d',i,j);

```

```

        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
tamanho = (M)*(N-1);
elseif bdcc == 3
    for j = 2:N+1
        for i = 1:M
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M)*(N);
end
elseif bccc == 2
    if bdcc == 1
        for j = 2:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N-1);
    elseif bdcc == 2
        for j = 2:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N-1);
    elseif bdcc == 3
        for j = 2:N+1
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];

```

```

        new = convertCharsToStrings(new);
    end
end
tamanho = (M)*(N);
end
elseif bccc == 3
    if bdcc == 1
        for j = 1:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    elseif bdcc == 2
        for j = 1:N
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N);
    elseif bdcc == 3
        for j = 1:N+1
            for i = 1:M
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M)*(N+1);
    end
end
elseif bbcc == 3
    if bccc == 1
        if bdcc == 1
            for j = 2:N
                for i = 1:M+1

```

```

        old = new;
        c = sprintf('%d,%d',i,j);
        new = [old;c];
        new = convertCharsToStrings(new);
    end
end
tamanho = (M+1)*(N-1);
elseif bdcc == 2
    for j = 2:N
        for i = 1:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M+1)*(N-1);
elseif bdcc == 3

end
elseif bccc == 2
    if bdcc == 1
        for j = 2:N
            for i = 1:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M+1)*(N-1);
    elseif bdcc == 2
        for j = 2:N
            for i = 1:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M+1)*(N-1);
    elseif bdcc == 3
        for j = 2:N+1

```

```

        for i = 1:M+1
            old = new;
            c = sprintf('%d,%d',i,j);
            new = [old;c];
            new = convertCharsToStrings(new);
        end
    end
    tamanho = (M+1)*(N);
end
elseif bccc == 3
    if bdcc == 1
        IM = imread('1-1-1-a.jpg');
    elseif bdcc == 2
        for j = 1:N
            for i = 1:M+1
                old = new;
                c = sprintf('%d,%d',i,j);
                new = [old;c];
                new = convertCharsToStrings(new);
            end
        end
        tamanho = (M+1)*(N);
    elseif bdcc == 3

    end
end
end
end

W = zeros(tamanho,tamanho);
W = string(W);
p_matriz = zeros(tamanho,1);

for i = 1:tamanho
    p_matriz(i,1) = G;
end

bacc = app.DropDown_2.Value;
bbcc = app.DropDown_3.Value;
bccc = app.DropDown_4.Value;
bdcc = app.DropDown_5.Value;

```

```

%%INICIO DA ESCRITA DA MATRIZ
g = ' - C*W(i,j) - A*W(i+2,j) - A*W(i-2,j) - B*W(i+1,j) - B*W(i-1,j) -
    D*W(i+1,j+1) - D*W(i+1,j-1) - D*W(i-1,j+1) - D*W(i-1,j-1) - E*W(i,j
    +1) - E*W(i,j-1) - F*W(i,j+2) - F*W(i,j-2)';
f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = replace(Position_i, 'i', 's');
Position_j = replace(Position_j, 'j', 'r');
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);

i = 4;
j = 4;
Valor_i = [];
Valor_p_i = [];

while i <= M-2
    Valor_i = [];
    for ui = 1:T_Linha
        old_i = Valor_i;
        i_row = [str2func(['@(s)' vectorize(Position_i(ui))])];
        Valor_i(ui) = i_row(i);
        Valor_i = [old_i; Valor_i(ui)];
        Valor_i = string(Valor_i);
    end
    Valor_p_i = [Valor_p_i, Valor_i];
    i = i + 1;
end
end

```

```

Valor_j = [];
Valor_p_j = [];

while j <= N-2
    Valor_j = [];
    for ui = 1:T_Linha
        old_j = Valor_j;
        j_row = [str2func(['@(r)' vectorize(Position_j(ui))])];
        Valor_j(ui) = j_row(j);
        Valor_j = [old_j; Valor_j(ui)];
        Valor_j = string(Valor_j);

    end

    Valor_p_j = [Valor_p_j, Valor_j];
    j = j + 1;
end
Position_final= [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);

for j = 1:n_j
    for i = 1:n_i
        old = Position_final;
        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];

    end
end

[m_po, n_po] = size(Position_final);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = 4;
j = 4;

for var = 1:n_po
    for j = 4:N-2
        i = 4;
        for i = 4:M-2

```



```

        for p = 1:T_Linha_new
            for q = 1:T_Linha
                if strcmp(Position_final(q,var), new(p,:)) == 1
                    linha_A(p,var) = {Constantes(q,:)};
                end
            end
        end

        for q = 1:T_Linha_new
            if strcmp(linha_A(q,var),"C") == 1
                k = q;
            end
        end

        for q = 1:T_Linha_new
            W(k,q) = {linha_A(q,var)};
        end
        i = i + 1;

    end
    j = j + 1;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C-A+2*E+4*F)*W(2,1) - A*W(4,1) - (B+2*D)*W(3,1) - 2*F*W
        (2,3) - (-4*F)*W(2,2)';
    constante_c = "(C-A+2*E+4*F)";
    Escrita_matriz;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C+2*E+4*F)*W(3,1) - A*W(5,1) - (B+2*D)*W(4,1) - (B+2*D)*W
        (2,1) - 2*F*W(3,3) - (-4*F)*W(3,2)';
    constante_c = "(C+2*E+4*F)";
    Escrita_matriz;
end

if strcmp(bccc , 'Livre') == 1
    g = ' - (C+2*E+4*F)*W(i,1) - A*W(i+2,1) - A*W(i-2,1) - (B+2*D)*W(i
        +1,1) - (B+2*D)*W(i-1,1) - (2*F)*W(i,3) - (-4*F)*W(i,2)';
    constante_c = "(C+2*E+4*F)";
end

```

```

    Escrita_matriz_i;
end

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C+2*E+4*F)*W(M-1,1) - A*W(M-3,1) - (B+2*D)*W(M,1) - (B+2*D)
        *W(M-2,1) - (2*F)*W(M-1,3) - (-4*F)*W(M-1,2)';
    constante_c = "(C+2*E+4*F)";
    Escrita_matriz_M;
end

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C+2*E+4*F-A)*W(M,1) - A*W(M-2,1) - (B+2*D)*W(M-1,1) - (2*F)
        *W(M,3) - (-4*F)*W(M,2)';
    constante_c = "(C+2*E+4*F-A)";
    Escrita_matriz_M;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C-A-F)*W(2,2) - A*W(4,2) - B*W(3,2) - D*W(3,3) - D*W(3,1)
        - E*W(2,3) - (E+2*F)*W(2,1) - F*W(2,4)';
    constante_c = "(C-A-F)";
    Escrita_matriz;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    g = ' - (C-F)*W(3,2) - A*W(5,2) - B*W(4,2) - B*W(2,2) - D*W(4,3) -
        D*W(2,1) - D*W(2,3) - D*W(4,1) - E*W(3,3) - (E+2*F)*W(3,1) - F*W
        (3,4)';
    constante_c = "(C-F)";
    Escrita_matriz;
end

if strcmp(bccc , 'Livre') == 1
    g = ' - (C-F)*W(i,2) - A*W(i+2,2) - A*W(i-2,2) - B*W(i+1,2) - B*W(i
        -1,2) - D*W(i+1,3) - D*W(i-1,1) - D*W(i+1,1) - D*W(i-1,3) - E*W(i
        ,3) - (E+2*F)*W(i,1) - F*W(i,4)';
    constante_c = "(C-F)";
    Escrita_matriz_i;
end

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1

```

```

g = ' - (C-F)*W(M-1,2) - A*W(M-3,2) - B*W(M,2) - B*W(M-2,2) - D*W(M
    ,3) - D*W(M-2,1) - D*W(M,1) - D*W(M-2,3) - E*W(M-1,3) - (E+2*F)*W
    (M-1,1) - F*W(M-1,4)';
constante_c = "(C-F)";
Escrita_matriz_M;
end

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
g = ' - (C-A-F)*W(M,2) - A*W(M-2,2) - B*W(M-1,2) - D*W(M-1,1) - D*W
    (M-1,3) - (E+2*F)*W(M,1) - E*W(M,3) - F*W(M,4)';
constante_c = "(C-A-F)";
Escrita_matriz_M;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
g = ' - (C-A)*W(2,3) - A*W(4,3) - B*W(3,3) - D*W(3,4) - D*W(3,2) -
    E*W(2,4) - E*W(2,2) - F*W(2,5) - F*W(2,1)';
constante_c = "(C-A)";
Escrita_matriz;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
g = ' - C*W(3,3) - A*W(5,3) - B*W(4,3) - B*W(2,3) - D*W(4,4) - D*W
    (2,2) - D*W(4,2) - D*W(2,4) - E*W(3,2) - E*W(3,4) - F*W(3,5) - F*
    W(3,1)';
constante_c = "C";
Escrita_matriz;
end

if strcmp(bccc , 'Livre') == 1
g = ' - C*W(i,3) - A*W(i+2,3) - A*W(i-2,3) - B*W(i+1,3) - B*W(i
    -1,3) - D*W(i+1,4) - D*W(i-1,2) - D*W(i+1,2) - D*W(i-1,4) - E*W(i
    ,2) - E*W(i,4) - F*W(i,5) - F*W(i,1)';
constante_c = "C";
Escrita_matriz_i;
end

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
g = ' - C*W(M-1,3) - A*W(M-3,3) - B*W(M,3) - B*W(M-2,3) - D*W(M,4)
    - D*W(M-2,2) - D*W(M,2) - D*W(M-2,4) - E*W(M-1,2) - E*W(M-1,4) -
    F*W(M-1,5) - F*W(M-1,1)';
constante_c = "C";
Escrita_matriz_M;

```

```

end

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    g = ' - (C-A)*W(M,3) - A*W(M-2,3) - B*W(M-1,3) - D*W(M-1,2) - D*W(M
        -1,4) - E*W(M,2) - E*W(M,4) - F*W(M,5) - F*W(M,1)';
    constante_c = "(C-A)";
    Escrita_matriz_M;
end

if strcmp(bacc , 'Apoiada') == 1
    g = ' - (C-A)*W(2,j) - A*W(4,j) - B*W(3,j) - D*W(3,j+1) - D*W(3,j
        -1) - E*W(2,j+1) - E*W(2,j-1) - F*W(2,j+2) - F*W(2,j-2)';
    constante_c = "(C-A)";
    Escrita_matriz_j;
end

if strcmp(bacc , 'Apoiada') == 1
    g = ' - C*W(3,j) - A*W(5,j) - B*W(4,j) - B*W(2,j) - D*W(4,j+1) - D*
        W(2,j-1) - D*W(2,j+1) - D*W(4,j-1) - E*W(3,j+1) - E*W(3,j-1) - F*
        W(3,j+2) - F*W(3,j-2)';
    constante_c = "C";
    Escrita_matriz_j;
end

if strcmp(bbcc , 'Apoiada') == 1
    g = ' - C*W(M-1,j) - A*W(M-3,j) - B*W(M,j) - B*W(M-2,j) - D*W(M,j
        +1) - D*W(M-2,j-1) - D*W(M-2,j+1) - D*W(M,j-1) - E*W(M-1,j+1) - E
        *W(M-1,j-1) - F*W(M-1,j+2) - F*W(M-1,j-2)';
    constante_c = "C";
    Escrita_matriz_M_j;
end

if strcmp(bbcc , 'Apoiada') == 1
    g = ' - (C-A)*W(M,j) - A*W(M-2,j) - B*W(M-1,j) - D*W(M-1,j-1) - D*W
        (M-1,j+1) - E*W(M,j+1) - E*W(M,j-1) - F*W(M,j+2) - F*W(M,j-2)';
    constante_c = "(C-A)";
    Escrita_matriz_M_j;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    g = ' - (C-A)*W(2,N-1) - A*W(4,N-1) - B*W(3,N-1) - D*W(3,N) - D*W
        (3,N-2) - E*W(2,N) - E*W(2,N-2) - F*W(2,N-3) - F*W(2,N+1)';
    constante_c = "(C-A)";

```

```

    Escrita_matriz_N;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    g = ' - C*W(3,N-1) - A*W(5,N-1) - B*W(4,N-1) - B*W(2,N-1) - D*W(4,N
        ) - D*W(4,N-2) - D*W(2,N) - D*W(2,N-2) - E*W(3,N) - E*W(3,N-2) -
        F*W(3,N-3) - F*W(3,N+1)';
    constante_c = "C";
    Escrita_matriz_N;
end

if strcmp(bdcc , 'Livre') == 1
    g = ' - C*W(i,N-1) - A*W(i+2,N-1) - A*W(i-2,N-1) - B*W(i+1,N-1) - B
        *W(i-1,N-1) - D*W(i+1,N) - D*W(i-1,N-2) - D*W(i-1,N) - D*W(i+1,N
        -2) - E*W(i,N) - E*W(i,N-2) - F*W(i,N-3) - F*W(i,N+1)';
    constante_c = "C";
    Escrita_matriz_i_N;
end

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    g = ' - C*W(M-1,N-1) - A*W(M-3,N-1) - B*W(M,N-1) - B*W(M-2,N-1) - D
        *W(M,N) - D*W(M-2,N-2) - D*W(M-2,N) - D*W(M,N-2) - E*W(M-1,N) - E
        *W(M-1,N-2) - F*W(M-1,N-3) - F*W(M-1,N+1)';
    constante_c = "C";
    Escrita_matriz_M_N;
end

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    g = ' - (C-A)*W(M,N-1) - A*W(M-2,N-1) - B*W(M-1,N-1) - D*W(M-1,N-2)
        - D*W(M-1,N) - E*W(M,N) - E*W(M,N-2) - F*W(M,N-3) - F*W(M,N+1)';
    constante_c = "(C-A)";
    Escrita_matriz_M_N;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    g = ' - (C-A-F)*W(2,N) - A*W(4,N) - B*W(3,N) - D*W(3,N+1) - D*W(3,N
        -1) - (E+2*F)*W(2,N+1) - E*W(2,N-1) - F*W(2,N-2)';
    constante_c = "(C-A-F)";
    Escrita_matriz_N;
end

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1

```

```

g = ' - (C-F)*W(3,N) - A*W(5,N) - B*W(4,N) - B*W(2,N) - D*W(4,N+1)
    - D*W(2,N-1) - D*W(4,N-1) - D*W(2,N+1) - (E+2*F)*W(3,N+1) - E*W
    (3,N-1) - F*W(3,N-2)';
constante_c = "(C-F)";
Escrita_matriz_N;
end

if strcmp(bdcc , 'Livre') == 1
g = ' - (C-F)*W(i,N) - A*W(i+2,N) - A*W(i-2,N) - B*W(i+1,N) - B*W(i
    -1,N) - D*W(i-1,N-1) - D*W(i+1,N-1) - D*W(i-1,N+1) - D*W(i+1,N+1)
    - (E+2*F)*W(i,N+1) - E*W(i,N-1) - F*W(i,N-2)';
constante_c = "(C-F)";
Escrita_matriz_i_N;
end

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
g = ' - (C-F)*W(M-1,N) - A*W(M-3,N) - B*W(M,N) - B*W(M-2,N) - D*W(M
    -2,N-1) - D*W(M,N-1) - D*W(M-2,N+1) - D*W(M,N+1) - (E+2*F)*W(M-1,
    N+1) - E*W(M-1,N-1) - F*W(M-1,N-2)';
constante_c = "(C-F)";
Escrita_matriz_M_N;
end

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
g = ' - (C-A-F)*W(M,N) - A*W(M-2,N) - B*W(M-1,N) - D*W(M-1,N-1) - D
    *W(M-1,N+1) - (E+2*F)*W(M,N+1) - E*W(M,N-1) - F*W(M,N-2)';
constante_c = "(C-A-F)";
Escrita_matriz_M_N;
end

if strcmp(bdcc,'Livre') == 1 && strcmp(bacc,'Apoiada') == 1
g = ' - (C-A+2*E+4*F)*W(2,N+1) - A*W(4,N+1) - (B+2*D)*W(3,N+1) -
    (2*F)*W(2,N-1) - (-4*F)*W(2,N)';
constante_c = "(C-A+2*E+4*F)";
Escrita_matriz_N;
end

if strcmp(bdcc , 'Livre') == 1 && strcmp(bacc , 'Apoiada') == 1
g = ' - (C+2*E+4*F)*W(3,N+1) - A*W(5,N+1) - (B+2*D)*W(4,N+1) - (B
    +2*D)*W(2,N+1) - (2*F)*W(3,N-1) - (-4*F)*W(3,N)';
constante_c = "(C+2*E+4*F)";
Escrita_matriz_N;
end

```

```

if strcmp(bdcc , 'Livre') == 1
    g = ' - (C+2*E+4*F)*W(i,N+1) - A*W(i-2,N+1) - A*W(i+2,N+1) - (B+2*D
        )*W(i+1,N+1) - (B+2*D)*W(i-1,N+1) - (2*F)*W(i,N-1) - (-4*F)*W(i,N
        )';
    constante_c = "(C+2*E+4*F)";
    Escrita_matriz_i_N;
end

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    g = ' - (C+2*E+4*F)*W(M-1,N+1) - A*W(M-3,N+1) - (B+2*D)*W(M,N+1) -
        (B+2*D)*W(M-2,N+1) - (2*F)*W(M-1,N-1) - (-4*F)*W(M-1,N)';
    constante_c = "(C+2*E+4*F)";
    Escrita_matriz_M_N;
end

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    g = ' - (C-A+2*E+4*F)*W(M,N+1) - A*W(M-2,N+1) - (B+2*D)*W(M-1,N+1)
        - (2*F)*W(M,N-1) - (-4*F)*W(M,N)';
    constante_c = "(C-A+2*E+4*F)";
    Escrita_matriz_M_N;
end

for i = 1:tamanho
    for j = 1:tamanho
        W_row = str2func(['@(A,B,C,D,E,F)' vectorize(W(i,j))]) ;
        Valor = W_row(A,B,C,D,E,F);
        W(i,j) = str2double(W(i,j));
        W(i,j) = {Valor};
    end
end

W = str2double(W);

```

---

---

 Algoritmo 2 – Critério de Sassenfeld.
 

---

```

B = ones(tamanho,tamanho);

for i = 1:tamanho
  for j = 1:tamanho
    if i == j
      B(i,j) = 0;
    else
      B(i,j) = 1;
    end
  end
end

k = [];
for i = 1:tamanho
  k_old = k;
  for j = 1:tamanho
    if i == j
      k = W(i,j);
    end
  end
  k = [k_old; k];
end
[m_i, n_i] = size(k);
m_i;
for i = 1:m_i
  W(i,:) = W(i,+)/k(i);
end
for i = 1:tamanho
  for j = 1:tamanho
    if i == j
      k = i;
    end
    if i == 1
      beta = sum(W(i,:)) - W(i,i);
      if j > 1
        B(i,j) = beta;
      end
    end
  end

  if i > 1
    beta = W(i,:)*B(:,i);
    if j > k
      B(i,j) = beta;
    end
  end
end

```



```
        end
    end
end
end
beta = B(:, tamanho);
beta(tamanho) = W(tamanho, :)*B(:, tamanho);
beta = abs(beta);
```

---

Fonte: Aatoria Própria (2018).

---

 Algoritmo 3 – Método de Gauss-Seidel.
 

---

```

cla(app.UIAxes5)
b = app.bField.Value;
d = app.dField.Value;
t = app.tField.Value;
choicematerial = app.DropDown.Value;

if strcmp(choicematerial , 'Isotropico') == 1
    Ex = app.EditField_4.Value;
    Ey = Ex;
    Gxy = app.EditField_6.Value;
    Poissonx = app.EditField_8.Value;
    Poissony = Poissonx;
elseif strcmp(choicematerial , 'Ortotropico') == 1
    Ex = app.EditField_4.Value;
    Ey = app.EditField_5.Value;
    Gxy = app.EditField_6.Value;
    Poissonx = app.EditField_8.Value;
    Poissony = app.EditField_9.Value;
end

bacc = app.DropDown_2.Value;
bbcc = app.DropDown_3.Value;
bccc = app.DropDown_4.Value;
bdcc = app.DropDown_5.Value;

M = app.EditField_14.Value;
N = app.EditField_11.Value;

Parada = app.EditField_12.Value;
tol = app.EditField_13.Value;

%Carregamento

carregamento= get(app.UITable , 'Data');
carregamento(:,2) = [];
carregamento = cell2mat(carregamento);
Modulo = carregamento(:,2);
cp = carregamento(:,3);
item = carregamento(:,1);
[row, column] = size(item);
i = 1;
pz = 0;

```

```

while i <= row
    pza = Modulo(i)*cp(i);
    pz = pza + pz;
    i = i + 1;
end

%constantes que serao utilizadas ao longo do programa
h = b / M ;%passo em x
k = d / N ;%passo em y

Dx = ((Ex*(t^3))/(12*(1 - ((Poissonx*Poissony)))));
Dy = ((Ey*(t^3))/(12*(1 - ((Poissonx*Poissony)))));
Dxy = ((Gxy)*(t^3))/6;
H = ((Dx*(Poissony)) + (Dy*(Poissonx)) + (4*(Dxy)))/ 2;

A = (k^4)*Dx;
B = -4*((Dx*(k^4)) + (H*(h^2)*(k^2)));
C = (6*Dx*(k^4)) + (8*H*(h^2)*(k^2)) + (6*Dy*(h^4));
D = (2*H*(h^2)*(k^2));
E = -4*((H*(h^2)*(k^2)) + (Dy*(h^4)));
F = Dy*(h^4);
G = pz*(k^4)*(h^4);

%consideraremos a matriz W como sendo a nossa matriz de iteracao inicial
,
%ou seja, W como sendo uma matriz de termos nulos

W = zeros(M+1,N+1); %cria a matriz nula MxN

%realizando a discretizacao das coordenadas e ordenadas para plotagem
%discretiza x
xVector = zeros(1, M+1);
i = 1;
for j = 0 : h : b
    xVector(i) = j;
    i = i+1;
end

%discretiza y
i = 1;
yVector = zeros(1, N+1);

for j = 0 : k : d

```

```

yVector(i) = j;
i = i+1;
end
l=1;

while l <= Parada

%Para i = 4,...,M-2 e j = 4,...,M-2
for i = 4:M-2
    for j = 4:N-2
        z = ( - A*W(i+2,j) - A*W(i-2,j) - B*W(i+1,j) - B*W(i-1,j) - D
            *W(i+1,j+1) - D*W(i+1,j-1) - D*W(i-1,j+1) - D*W(i-1,j-1)
            - E*W(i,j+1) - E*W(i,j-1) - F*W(i,j+2) - F*W(i,j-2) + G )/(
            C);
        NORMA = abs(W(i,j) - z);
        W(i,j) = z;
    end
end

% para i = 2 e j = 1

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(4,1) - (B+2*D)*W(3,1) - 2*F*W(2,3) + 4*F*W(2,2) + G)
        /(C - A + 2*E + 4*F);
    if abs( W(2,1) - z) > NORMA
        NORMA = abs(W(2,1) - z);
    end
    W(2,1) = z;
end

%Para i = 3 e j = 1

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(5,1) - (B + 2*D)*W(4,1) - (B + 2*D)*W(2,1) - 2*F*W
        (3,3) + 4*F*W(3,2) + G)/(C + 2*E + 4*F);
    if abs( W(3,1) - z) > NORMA
        NORMA = abs(W(3,1) - z);
    end
    W(3,1) = z;
end

% Para i = 4,...,M-2 e j = 1

```

```

if strcmp(bccc , 'Livre') == 1
    for i = 4:M-2
        z = ( - A*W(i+2,1) - A*W(i-2,1) - (B+2*D)*W(i+1,1) - (B+2*D)
            *W(i-1,1) - 2*F*W(i,3) + 4*F*W(i,2) + G)/(C + 2*E + 4*F);
        if abs( W(i,1) - z) > NORMA
            NORMA = abs(W(i,1) - z);
        end
        W(i,1) = z;
    end
end

%Para i = M-1 e j = 1

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(M-3,1) - (B + 2*D)*W(M,1) - (B + 2*D)*W(M-2,1) - 2*F
        *W(M-1,3) + 4*F*W(M-1,2) + G)/(C + 2*E + 4*F);
    if abs( W(M-1,1) - z) > NORMA
        NORMA = abs(W(M-1,1) - z);
    end
    W(M-1,1) = z;
end

%Para i = M e j = 1

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(M-2,1) - (B+2*D)*W(M-1,1) - 2*F*W(M,3) + 4*F*W(M,2)
        + G)/(C + 2*E + 4*F - A);
    if abs( W(M,1) - z) > NORMA
        NORMA = abs(W(M,1) - z);
    end
    W(M,1) = z;
end

%Para i = 1 e j = 2;

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(4,2) - B*W(3,2) - D*W(3,3) - D*W(3,1) - E*W(2,3) - (
        E + 2*F)*W(2,1) - F*W(2,4) + G)/(C - A - F);
    if abs( W(2,2) - z) > NORMA
        NORMA = abs(W(2,2) - z);
    end
    W(2,2) = z;
end

```

```
%Para i = 3 e j = 2
```

```
if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = ( - A*W(5,2) - B*W(4,2) - B*W(2,2) - D*W(4,3) - D*W(2,1) - D
        *W(2,3) - D*W(4,1) - E*W(3,3) - (E + 2*F)*W(3,1) - F*W(3,4)
        + G)/(C - F);
    if abs( W(3,2) - z) > NORMA
        NORMA = abs(W(3,2) - z);
    end
    W(3,2) = z;
end
```

```
%Para i = 4,...,M-2 e j = 2
```

```
if strcmp(bccc , 'Livre') == 1
    for i = 4:M-2
        z = ( - A*W(i+2,2) - A*W(i-2,2) - B*W(i+1,2) - B*W(i-1,2) -
            D*W(i+1,3) - D*W(i-1,1) - D*W(i+1,1) - D*W(i-1,3) - E*W(i
            ,3) - (E + 2*F)*W(i,1) - F*W(i,4) + G)/(C - F);
        if abs( W(i,2) - z) > NORMA
            NORMA = abs(W(i,2) - z);
        end
        W(i,2) = z;
    end
end
```

```
%Para i = M-1 e j = 2
```

```
if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    z = ( - A*W(M-3,2) - B*W(M,2) - B*W(M-2,2) - D*W(M,3) - D*W(M
        -2,1) - D*W(M,1) - D*W(M-2,3) - E*W(M-1,3) - (E + 2*F)*W(M
        -1,1) - F*W(M-1,4) + G)/(C - F);
    if abs( W(M-1,2) - z) > NORMA
        NORMA = abs(W(M-1,2) - z);
    end
    W(M-1,2) = z;
end
```

```
%Para i = M e j = 2
```

```
if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
```

```

z = (- A*W(M-2,2) - B*W(M-1,2) - D*W(M-1,1) - D*W(M-1,3) - (E +
    2*F)*W(M,1) - E*W(M,3) - F*W(M,4) + G)/(C - A - F);
if abs( W(M,2) - z) > NORMA
    NORMA = abs(W(M,2) - z);
end
W(M,2) = z;
end

%Para i = 2 e j = 3

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = (- A*W(4,3) - B*W(3,3) - D*W(3,4) - D*W(3,2) - E*W(2,4) - E*
        W(2,2) - F*W(2,5) - F*W(2,1) + G)/(C - A);
    if abs( W(2,3) - z) > NORMA
        NORMA = abs(W(2,3) - z);
    end
    W(2,3) = z;
end

%Para i = 3 e j = 3

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bccc , 'Livre') == 1
    z = (- A*W(5,3) - B*W(4,3) - B*W(2,3) - D*W(4,4) - D*W(2,2) - D*
        W(4,2) - D*W(2,4) - E*W(3,2) - E*W(3,4) - F*W(3,5) - F*W(3,1)
        + G)/(C);
    if abs( W(3,3) - z) > NORMA
        NORMA = abs(W(3,3) - z);
    end
    W(3,3) = z;
end

%Para i = 4, ..., M-2 e j = 3

if strcmp(bccc , 'Livre') == 1
    for i = 4:M-2
        z = ( - A*W(i+2,3) - A*W(i-2,3) - B*W(i+1,3) - B*W(i-1,3) -
            D*W(i+1,4) - D*W(i-1,2) - D*W(i+1,2) - D*W(i-1,4) - E*W(i
            ,2) - E*W(i,4) - F*W(i,5) - F*W(i,1) + G)/(C);
        if abs( W(i,3) - z) > NORMA
            NORMA = abs(W(i,3) - z);
        end
        W(i,3) = z;
    end
end

```

```

end
end

%Para i = M-1 e j = 3

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    z = ( - A*W(M-3,3) - B*W(M,3) - B*W(M-2,3) - D*W(M,4) - D*W(M
        -2,2) - D*W(M,2) - D*W(M-2,4) - E*W(M-1,2) - E*W(M-1,4) - F*W
        (M-1,5) - F*W(M-1,1) + G)/(C);
    if abs( W(M-1,3) - z) > NORMA
        NORMA = abs(W(M-1,3) - z);
    end
    W(M-1,3) = z;
end

%Para i = M e j = 3

if strcmp(bccc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    z = ( - A*W(M-2,3) - B*W(M-1,3) - D*W(M-1,2) - D*W(M-1,4) - E*W(
        M,2) - E*W(M,4) - F*W(M,5) - F*W(M,1) + G)/(C - A);
    if abs( W(M,3) - z) > NORMA
        NORMA = abs(W(M,3) - z);
    end
    W(M,3) = z;
end

%Para i = 2 e j = 4,...,N-2

if strcmp(bacc , 'Apoiada') == 1
    for j = 4:N-2
        z = ( - A*W(4,j) - B*W(3,j) - D*W(3,j+1) - D*W(3,j-1) - E*W
            (2,j+1) - E*W(2,j-1) - F*W(2,j+2) - F*W(2,j-2) + G)/(C -
            A);
        if abs( W(2,j) - z) > NORMA
            NORMA = abs(W(2,j) - z);
        end
        W(2,j) = z;
    end
end

%Para i = 3 e j = 4,...,N-2

if strcmp(bacc , 'Apoiada') == 1

```



```

for j = 4:N-2
    z = ( - A*W(5,j) - B*W(4,j) - B*W(2,j) - D*W(4,j+1) - D*W(2,
        j-1) - D*W(2,j+1) - D*W(4,j-1) - E*W(3,j+1) - E*W(3,j-1)
        - F*W(3,j+2) - F*W(3,j-2) + G)/(C);
    if abs( W(3,j) - z) > NORMA
        NORMA = abs(W(3,j) - z);
    end
    W(3,j) = z;
end
end

%Para i = M-1 e j = 4,...,N-2

if strcmp(bbcc , 'Apoiada') == 1
    for j = 4:N-2
        z = ( - A*W(M-3,j) - B*W(M,j) - B*W(M-2,j) - D*W(M,j+1) - D*
            W(M-2,j-1) - D*W(M-2,j+1) - D*W(M,j-1) - E*W(M-1,j+1) - E
            *W(M-1,j-1) - F*W(M-1,j+2) - F*W(M-1,j-2) + G)/(C);
        if abs( W(M-1,j) - z) > NORMA
            NORMA = abs(W(M-1,j) - z);
        end
        W(M-1,j) = z;
    end
end

%Para i = M e j = 4,...,N-2

if strcmp(bbcc , 'Apoiada') == 1
    for j = 4:N-2
        z = ( - A*W(M-2,j) - B*W(M-1,j) - D*W(M-1,j-1) - D*W(M-1,j
            +1) - E*W(M,j+1) - E*W(M,j-1) - F*W(M,j+2) - F*W(M,j-2)
            + G)/(C - A);
        if abs( W(M,j) - z) > NORMA
            NORMA = abs(W(M,j) - z);
        end
        W(M,j) = z;
    end
end

%Para i = 2 e j = N-1

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1

```

```

z = ( - A*W(4,N-1) - B*W(3,N-1) - D*W(3,N) - D*W(3,N-2) - E*W(2,
      N) - E*W(2,N-2) - F*W(2,N-3) - F*W(2,N+1) + G)/(C - A);
if abs( W(2,N-1) - z) > NORMA
    NORMA = abs(W(2,N-1) - z);
end
W(2,N-1) = z;
end

%Para i = 3 e j = N-1

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    z = ( - A*W(5,N-1) - B*W(4,N-1) - B*W(2,N-1) - D*W(4,N) - D*W(4,N
      -2) - D*W(2,N) - D*W(2,N-2) - E*W(3,N) - E*W(3,N-2) - F*W(3,N
      -3) - F*W(3,N+1) + G)/(C);
    if abs( W(3,N-1) - z) > NORMA
        NORMA = abs(W(3,N-1) - z);
    end
    W(3,N-1) = z;
end

%Para i = 4, ..., M-2 e j = N-1

if strcmp(bdcc , 'Livre') == 1
    for i = 4:M-2
        z = ( - A*W(i+2,N-1) - A*W(i-2,N-1) - B*W(i+1,N-1) - B*W(i-1,N
          -1) - D*W(i+1,N) - D*W(i-1,N-2) - D*W(i-1,N) - D*W(i+1,N
          -2) - E*W(i,N) - E*W(i,N-2) - F*W(i,N-3) - F*W(i,N+1) + G
          )/(C);
        if abs( W(i,N-1) - z) > NORMA
            NORMA = abs(W(i,N-1) - z);
        end
        W(i,N-1) = z;
    end
end

%Para i = M-1 e j = N-1

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    z = ( - A*W(M-3,N-1) - B*W(M,N-1) - B*W(M-2,N-1) - D*W(M,N) - D*W
      (M-2,N-2) - D*W(M-2,N) - D*W(M,N-2) - E*W(M-1,N) - E*W(M-1,N
      -2) - F*W(M-1,N-3) - F*W(M-1,N+1) + G)/(C);
    if abs( W(M-1,N-1) - z) > NORMA
        NORMA = abs(W(M-1,N-1) - z);
    end
end

```

```

    end
    W(M-1,N-1) = z;
end

%Para i = M e j = N-1

if strcmp(bdcc , 'Apoiada') == 1 && strcmp(bbcc , 'Livre') == 1
    z = ( - (2*A + B)*W(M+1,N-1) - A*W(M-2,N-1) - B*W(M-1,N-1) - D*W
        (M+1,N) - D*W(M-1,N-2) - D*W(M-1,N) - D*W(M+1,N-2) - E*W(M,N)
        - E*W(M,N-2) - F*W(M,N-3) + G)/(C - A);
    if abs( W(M,N-1) - z) > NORMA
        NORMA = abs(W(M,N-1) - z);
    end
    W(M,N-1) = z;
end

%Para i = 2 e j = N

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    z = ( - A*W(4,N) - B*W(3,N) - D*W(3,N+1) - D*W(3,N-1) - (E + 2*F
        )*W(2,N+1) - E*W(2,N-1) - F*W(2,N-2) + G)/(C - A - F);
    if abs( W(2,N) - z) > NORMA
        NORMA = abs(W(2,N) - z);
    end
    W(2,N) = z;
end

%Para i = 3 e j = N

if strcmp(bacc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    z = ( - A*W(5,N) - B*W(4,N) - B*W(2,N) - D*W(4,N+1) - D*W(2,N-1) -
        D*W(4,N-1) - D*W(2,N+1) - (E + 2*F)*W(3,N+1) - E*W(3,N-1) - F*
        W(3,N-2) + G)/(C - F);
    if abs( W(3,N) - z) > NORMA
        NORMA = abs(W(3,N) - z);
    end
    W(3,N) = z;
end

%Para i = 4,...,M-2 e j = N

if strcmp(bdcc , 'Livre') == 1

```

```

for i = 4:M-2
    z = ( - A*W(i+2,N) - A*W(i-2,N) - B*W(i+1,N) - B*W(i-1,N) -
          D*W(i-1,N-1) - D*W(i+1,N-1) - D*W(i-1,N+1) - D*W(i+1,N+1)
          - (E + 2*F)*W(i,N+1) - E*W(i,N-1) - F*W(i,N-2) + G)/(C -
          F);
    if abs( W(i,N) - z) > NORMA
        NORMA = abs(W(i,N) - z);
    end
    W(i,N) = z;
end
end

%Para i = M-1 e j = N

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    z = ( - A*W(M-3,N) - B*W(M,N) - B*W(M-2,N) - D*W(M-2,N-1) - D*W(M
        ,N-1) - D*W(M-2,N+1) - D*W(M,N+1) - (E + 2*F)*W(M-1,N+1) - E*W(
        M-1,N-1) - F*W(M-1,N-2) + G)/(C - F);
    if abs( W(M-1,N) - z) > NORMA
        NORMA = abs(W(M-1,N) - z);
    end
    W(M-1,N) = z;
end

%Para i = M e j = N

if strcmp(bbcc , 'Apoiada') == 1 && strcmp(bdcc , 'Livre') == 1
    z = ( - A*W(M-2,N) - B*W(M-1,N) - D*W(M-1,N-1) - D*W(M-1,N+1) -
          (2*F + E)*W(M,N+1) - E*W(M,N-1) - F*W(M,N-2) + G)/(C - A - F)
          ;
    if abs( W(M,N) - z) > NORMA
        NORMA = abs(W(M,N) - z);
    end
    W(M,N) = z;
end

%Para i = 2 e j = N+1

if strcmp(bdcc , 'Livre') == 1 && strcmp(bacc , 'Apoiada') == 1
    z = ( - A*W(4,N+1) - (B + 2*D)*W(3,N+1) - 2*F*W(2,N-1) + 4*F*W
          (2,N) + G)/(C - A + 2*E + 4*F);
    if abs( W(2,N+1) - z) > NORMA
        NORMA = abs(W(2,N+1) - z);
    end
end

```

```

    end
    W(2,N+1) = z;
end

%Para i = 3 e j = N+1

if strcmp(bdcc , 'Livre') == 1 && strcmp(bacc , 'Apoiada') == 1
    z = ( - A*W(5,N+1) - (B + 2*D)*W(4,N+1) - (B + 2*D)*W(2,N+1) - 2*
        F*W(3,N-1) + 4*F*W(3,N) + G)/(C + 2*E + 4*F);
    if abs( W(3,N+1) - z) > NORMA
        NORMA = abs(W(3,N+1) - z);
    end
    W(3,N+1) = z;
end

%Para i = 4,...,M-2 e j = N+1

if strcmp(bdcc , 'Livre') == 1
    for i = 4:M-2
        z = (- A*W(i-2,N+1) - A*W(i+2,N+1) - (B + 2*D)*W(i+1,N+1) - (B
            + 2*D)*W(i-1,N+1) - 2*F*W(i,N-1) + 4*F*W(i,N) + G)/(C +
            2*E + 4*F);
        if abs( W(i,N+1) - z) > NORMA
            NORMA = abs(W(i,N+1) - z);
        end
        W(i,N+1) = z;
    end
end

%Para i = M-1 e j = N+1

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1
    z = (- A*W(M-3,N+1) - (B + 2*D)*W(M,N+1) - (B + 2*D)*W(M-2,N+1)
        - 2*F*W(M-1,N-1) + 4*F*W(M-1,N) + G)/(C + 2*E + 4*F);
    if abs( W(M-1,N+1) - z) > NORMA
        NORMA = abs(W(M-1,N+1) - z);
    end
    W(M-1,N+1) = z;
end

%Para i = M e j = N+1

if strcmp(bdcc , 'Livre') == 1 && strcmp(bbcc , 'Apoiada') == 1

```

```
z = ( - A*W(M-2,N+1) - (B + 2*D)*W(M-1,N+1) - 2*F*W(M,N-1) + 4*F
      *W(M,N) + G)/(C - A + 2*E + 4*F);
if abs( W(M,N+1) - z) > NORMA
    NORMA = abs(W(M,N+1) - z);
end
W(M,N+1) = z;
end

if (NORMA <= tol)
    break
end

l = l + 1;
end
dlmwrite('W_solucao.txt', W);
```

---

Fonte: Aatoria Própria (2018).

---

 Algoritmo 4 – Auxiliar do Algoritmo 1: Escrita\_matriz.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = erase(Position, '(');
Position = erase(Position, ')');
Constantes = convertCharsToStrings(Constantes);
Position = convertCharsToStrings(Position);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
linha_A = zeros(tamanho,1);
linha_A = string(linha_A);

Position_final= Position;
[m_po, n_po] = size(Position_final);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);

for var = 1:n_po
    for j = 1:T_Linha_new
        for i = 1:T_Linha
            if strcmp(Position(i,:), new(j,:)) == 1
                linha_A(j,:) = {Constantes(i,:)};
            end
        end
    end
    for q = 1:T_Linha_new
        if strcmp(linha_A(q,var), constante_c) == 1
            k = q;
        end
    end

    for i = 1:T_Linha_new
        W(k,i) = {linha_A(i)};
    end
end

```

---

---

 Algoritmo 5 – Auxiliar do Algoritmo 1: Escrita\_matriz\_i.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = replace(Position_i, 'i', 's');
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = 4;
j = str2double(Position_j(1));
Valor_i = [];
Valor_p_i = [];
while i <= M-2
    Valor_i = [];
    for ui = 1:T_Linha
        old_i = Valor_i;
        i_row = [str2func(['@(s)' vectorize(Position_i(ui))])];
        Valor_i(ui) = i_row(i);
        Valor_i = [old_i; Valor_i(ui)];
        Valor_i = string(Valor_i);
    end
    Valor_p_i = [Valor_p_i, Valor_i];
    i = i + 1;
end
Valor_p_j = Position_j;

Position_final = [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);
for j = 1:n_j
    for i = 1:n_i
        old = Position_final;

```



```

    Valor_i = Valor_p_i(:,i);
    Valor_j = Valor_p_j(:,j);
    Position_final = horzcat(Valor_i,Valor_j);
    Position_final = join(Position_final,',' );
    Position_final = [old,Position_final];
end
end
[m_po, n_po] = size(Position_final);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = 4;
for var = 1:n_po
    for j = str2double(Position_j(1))
        i = 4;
        for i = 4:M-2
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var),constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
            i = i + 1;
        end
    end
end
end

```

---

---

 Algoritmo 6 – Auxiliar do Algoritmo 1: Escrita\_matriz\_i\_N
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = 4;
j = N;
Valor_p_i = [];
while i <= M-2
    Valor_i = [];
    for ui = 1:T_Linha
        old_i = Valor_i;
        i_row = [str2func(['@(i)' vectorize(Position_i(ui))])];
        Valor_i(ui) = i_row(i);
        Valor_i = [old_i; Valor_i(ui)];
        Valor_i = string(Valor_i);

        end

    Valor_p_i = [Valor_p_i, Valor_i];
    i = i + 1;
end
Valor_j = [];
Valor_p_j = [];
for uj = 1:T_Linha
    old_j = Valor_j;
    j_row = [str2func(['@(N)' vectorize(Position_j(uj))])];
    Valor_j(uj) = j_row(j);
    Valor_j = [old_j; Valor_j(uj)];
    Valor_j = string(Valor_j);
end
end

```

```

Valor_p_j = [Valor_p_j, Valor_j];
Position_final= [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);
for j = 1:n_j
    for i = 1:n_i
        old = Position_final;
        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];
    end
end
[m_po, n_po] = size(Position_final);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = 4;
for var = 1:n_po
    for j = N
        i = 4;
        for i = 4:M-2
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var),constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
            i = i + 1;
        end
    end
end
end
end

```

---

 Algoritmo 7 – Auxiliar do Algoritmo 1: Escrita\_matriz\_j.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_j = replace(Position_j, 'j', 'r');
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
j = 4;
Valor_i = [];
Valor_p_i = Position_i;
Valor_j = [];
Valor_p_j = [];

while j <= N-2
    Valor_j = [];
    for ui = 1:T_Linha
        old_j = Valor_j;
        j_row = [str2func(['@(r)' vectorize(Position_j(ui))])];
        Valor_j(ui) = j_row(j);
        Valor_j = [old_j; Valor_j(ui)];
        Valor_j = string(Valor_j);

    end
    Valor_p_j = [Valor_p_j, Valor_j];
    j = j + 1;
end

Position_final = [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);

for j = 1:n_j
    for i = 1:n_i
        old = Position_final;

```

```

        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];
    end
end

[m_po, n_po] = size(Position_final);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
j = 4;
for var = 1:n_po
    for j = 4:N-2
        for i = str2double(Position_i(1))
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var),constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
        end
        j = j + 1;
    end
end
end

```

---

Algoritmo 8 – Auxiliar do Algoritmo 1: Escrita\_matriz\_M.

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = M;
j = str2double(Position_j(1));
Valor_i = [];
Valor_p_i = [];

for ui = 1:T_Linha
    old_i = Valor_i;
    i_row = [str2func(['@(M)' vectorize(Position_i(ui))])];
    Valor_i(ui) = i_row(i);
    Valor_i = [old_i; Valor_i(ui)];
    Valor_i = string(Valor_i);
end

Valor_p_i = [Valor_p_i, Valor_i];
Valor_p_j = Position_j;

Position_final = [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);

for j = 1:n_j
    for i = 1:n_i
        old = Position_final;
        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
    end
end

```

```

        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];
    end
end

[m_po, n_po] = size(Position_final);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);

for var = 1:n_po
    for j = str2double(Position_j(1))
        for i = M
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var), constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
        end
    end
end
end
end

```

---

---

 Algoritmo 9 – Auxiliar do Algoritmo 1: Escrita\_matriz\_M\_j.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_j = replace(Position_j, 'j', 'r');
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
i = M;
j = 4;
Valor_i = [];
Valor_p_i = [];
for ui = 1:T_Linha
    old_i = Valor_i;
    i_row = [str2func(['@(M)' vectorize(Position_i(ui))])];
    Valor_i(ui) = i_row(i);
    Valor_i = [old_i; Valor_i(ui)];
    Valor_i = string(Valor_i);
end
Valor_p_i = [Valor_p_i, Valor_i];
Valor_j = [];
Valor_p_j = [];
while j <= N-2
    Valor_j = [];
    for ui = 1:T_Linha
        old_j = Valor_j;
        j_row = [str2func(['@(r)' vectorize(Position_j(ui))])];
        Valor_j(ui) = j_row(j);
        Valor_j = [old_j; Valor_j(ui)];
        Valor_j = string(Valor_j);
    end
    Valor_p_j = [Valor_p_j, Valor_j];
    j = j + 1;
end

```



```

Position_final= [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);
for j = 1:n_j
    for i = 1:n_i
        old = Position_final;
        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];
    end
end
[m_po, n_po] = size(Position_final);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
j = 4;
for var = 1:n_po
    for j = 4:N-2
        for i = str2double(Valor_p_i(1))
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
        end
        for q = 1:T_Linha_new
            if strcmp(linha_A(q,var),constante_c) == 1
                k = q;
            end
        end
        for q = 1:T_Linha_new
            W(k,q) = {linha_A(q,var)};
        end
    end
    j = j + 1;
end
end

```

---

 Algoritmo 10 – Auxiliar do Algoritmo 1: Escrita\_matriz\_M\_N.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
i = M;
j = N;
Valor_i = [];
Valor_p_i = [];
for ui = 1:T_Linha
    old_i = Valor_i;
    i_row = [str2func(['@(M)' vectorize(Position_i(ui))])];
    Valor_i(ui) = i_row(i);
    Valor_i = [old_i; Valor_i(ui)];
    Valor_i = string(Valor_i);
end
Valor_p_i = [Valor_p_i, Valor_i];
Valor_j = [];
Valor_p_j = [];
for uj = 1:T_Linha
    old_j = Valor_j;
    j_row = [str2func(['@(N)' vectorize(Position_j(uj))])];
    Valor_j(uj) = j_row(j);
    Valor_j = [old_j; Valor_j(uj)];
    Valor_j = string(Valor_j);
end
Valor_p_j = [Valor_p_j, Valor_j];
Position_final = [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);

```

```

for j = 1:n_j
    for i = 1:n_i
        old = Position_final;
        Valor_i = Valor_p_i(:,i);
        Valor_j = Valor_p_j(:,j);
        Position_final = horzcat(Valor_i,Valor_j);
        Position_final = join(Position_final,',' );
        Position_final = [old,Position_final];
    end
end
[m_po, n_po] = size(Position_final);
linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);
for var = 1:n_po
    for j = N
        for i = M
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var), constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
        end
    end
end
end
end

```

---

---

 Algoritmo 11 – Auxiliar do Algoritmo 1: Escrita\_matriz\_N.
 

---

```

f = split(g, ' - ');
f(1,:) = [];
h = split(f, '*W');
Constantes = h(:,1);
Position = h(:,2);
Position = split(Position, ',');
Position_i = Position(:,1);
Position_j = Position(:,2);
Position_i = erase(Position_i, '(');
Position_j = erase(Position_j, ')');
Constantes = convertCharsToStrings(Constantes);
Position_i = convertCharsToStrings(Position_i);
Position_j = convertCharsToStrings(Position_j);
[T_Linha, n] = size(Constantes);
[T_Linha_new, nA] = size(new);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);

i = str2double(Position_i(1));
j = N;
Valor_p_i = Position_i;

Valor_j = [];
Valor_p_j = [];
for uj = 1:T_Linha
    old_j = Valor_j;
    j_row = [str2func(['@(N)' vectorize(Position_j(uj))])];
    Valor_j(uj) = j_row(j);
    Valor_j = [old_j; Valor_j(uj)];
    Valor_j = string(Valor_j);
end

Valor_p_j = [Valor_p_j, Valor_j];

Position_final = [];
[m_i, n_i] = size(Valor_p_i);
[m_j, n_j] = size(Valor_p_j);

for j = 1:n_j
    for i = 1:n_i
        old = Position_final;

```

```

    Valor_i = Valor_p_i(:,i);
    Valor_j = Valor_p_j(:,j);
    Position_final = horzcat(Valor_i,Valor_j);
    Position_final = join(Position_final,',' );
    Position_final = [old,Position_final];
end
end
[m_po, n_po] = size(Position_final);

linha_A = zeros(tamanho, n_po);
linha_A = string(linha_A);

for var = 1:n_po
    for j = N
        for i = str2double(Position_i(1))
            for p = 1:T_Linha_new
                for q = 1:T_Linha
                    if strcmp(Position_final(q,var), new(p,:)) == 1
                        linha_A(p,var) = {Constantes(q,:)};
                    end
                end
            end
            for q = 1:T_Linha_new
                if strcmp(linha_A(q,var), constante_c) == 1
                    k = q;
                end
            end
            for q = 1:T_Linha_new
                W(k,q) = {linha_A(q,var)};
            end
        end
    end
end
end
end

```

---

## Apêndice B – Trabalhos realizados

Houve a apresentação dos trabalhos intitulados, “Aplicação de Métodos Numéricos na Deformação de Placas Engastadas” e “Rotina de Cálculo de Deformação de Placas Delgadas” em formato de pôster para o XXXVII Congresso Nacional de matemática aplicada e computacional 2017. Ambos os trabalhos também foram publicados na revista “Proceeding Series of the Brazilian Society of Computational and Applied Mathematics” em que o próprio congresso fornece para os autores.

Houve também a apresentação do trabalho “Aplicação do Método das Diferenças Finitas para cálculo de deflexão de placa bi-apoiada” em formato de pôster para o VII Ciclo de palestras e perspectivas matemáticas (CIPEM).

Pode-se citar também o trabalho apresentação no 7º Seminário de Extensão e Inovação da UTFPR, pelo trabalho “Atividades e estudo sobre deflexão de placas com o uso do método de diferenças finitas” apresentado em formato de pôster.