

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

JÚLIA BARCELOS DE MENEZES

**RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS PARCIAIS EM MALHAS NÃO-
ESTRUTURADAS VIA MÉTODO DOS VOLUMES FINITOS UTILIZANDO UM
MÉTODO DE ALTA ORDEM**

UBERLÂNDIA

2019

JÚLIA BARCELOS DE MENEZES

**RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS PARCIAIS EM MALHAS NÃO-
ESTRUTURADAS VIA MÉTODO DOS VOLUMES FINITOS UTILIZANDO UM
MÉTODO DE ALTA ORDEM**

Trabalho de Conclusão de Curso apresentado ao corpo docente da Faculdade de Engenharia Mecânica, como exigência parcial para a obtenção do título de Bacharel em Engenharia Mecânica.

Área de concentração: Mecânica de fluidos

Orientador: Prof. Dr. Alessandro Alves Santana

UBERLÂNDIA

2019

**RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS PARCIAIS EM MALHAS NÃO-
ESTRUTURADAS VIA MÉTODO DOS VOLUMES FINITOS UTILIZANDO UM
MÉTODO DE ALTA ORDEM**

Trabalho de Conclusão de Curso aprovado
para a obtenção do título de Bacharel em
Engenharia Mecânica pela banca examinadora
formada por:

Uberlândia, 18 de fevereiro de 2019.

Prof. Dr. Alessandro Alves Santana, UFU/MG

Prof. Dr. Elie Luis Martínez Padilla, UFU/MG

Prof.^a Dr.^a Elaine Gomes Assis, UFU/MG

AGRADECIMENTOS

Agradeço a meu orientador sobretudo pela paciência, incentivo ao crescimento acadêmico e pessoal e apoio emocional em situações nas quais o desenvolvimento do trabalho encontrou obstáculos, seja pelo acúmulo de atividades ou pelas mudanças que o mercado de trabalho me exigiu.

Agradeço a meus familiares por sempre colocarem a educação como valor prioritário em minha vida, e durante a faculdade, pelo apoio e incentivo durante toda minha trajetória acadêmica.

RESUMO

O presente trabalho ilustra um método de resolução numérica para equações diferenciais parciais baseado no método dos volumes finitos. Esse esquema foi desenvolvido por Carl Olliver-Gooch, e tem por principal característica a reconstrução da solução em cada volume de controle, com precisão de alta ordem, em um domínio discretizado por malhas não-estruturadas de triângulos. A solução reconstruída em cada volume de controle é um polinômio embasado em série de Taylor, e os gradientes desse polinômio são utilizados para calcular os fluxos, exigidos pelo método dos volumes finitos, nas faces dos volumes de controle. Para dar materialidade na ilustração do esquema supracitado, foi implementado, em linguagem C, sob plataforma Linux, um código para realizar testes envolvendo problemas de transporte, problemas de difusão e problemas de transporte e difusão. As malhas computacionais referentes aos domínios discretizados foram construídas utilizando o software *Gmsh*. Os resultados obtidos corroboraram com as referências estudadas comprovando a eficácia da metodologia. Tal proposta tem por aplicação o estudo de problemas de engenharia que envolvam equações diferenciais parciais que não podem ser resolvidas analiticamente, a exemplo de alguns problemas de escoamento de fluidos.

Palavras-chave: equações diferenciais parciais, alta ordem, malhas não-estruturadas.

ABSTRACT

The present work illustrates a method for numerical resolution of partial differential equations based on the finite volume method. This scheme was developed by Carl Olliver-Gooch and it has as main characteristic the reconstruction of the solution at each control volume, with high order precision, in a domain discretized by unstructured meshes of triangles. The reconstructed solution in each control volume is a Taylor's Series based on polynomial, and the gradients of this polynomial are used to calculate the flows required by the finite volume method on the faces of the control volumes. To give materiality in the illustration of the aforementioned scheme, a code to perform tests involving transport problems, diffusion problems and transport and diffusion problems was implemented in C language using the Linux platform. The computational meshes for the discretized domains were constructed using the software Gmsh. The obtained results corroborated with the references studied proving the effectiveness of the methodology. This proposal has applications on the study of engineering problems that involve partial differential equations which cannot be solved analytically, as some fluids flow problems.

Keywords: *partial differential equations, high order, unstructured meshes.*

LISTA DE FIGURAS

Figura 2.1 - Terminologia das partes de uma malha	13
Figura 2.2 - Tipos de elementos de malha	14
Figura 2.3 - Malhas estruturadas em domínios a) tridimensional e b) bidimensional	15
Figura 2.4 - Malhas não-estruturadas em domínios a) tridimensional e b) bidimensional	16
Figura 2.5 - Malha híbrida aplicada a peça de motor de combustão interna	17
Figura 2.6 - Esquemas de armazenamento para o MVF, vertex-centered à esquerda e cell-centered à direita	20
Figura 2.7 - Integração numérica através de polinômios de Gauss-Legendre	24
Figura 2.8 - Coordenadas naturais aplicadas ao triângulo	25
Figura 3.1 – Tela inicial do Gmsh.....	29
Figura 3.2 – Tela do Gmsh evidenciando menu expansível e desenho de pontos e retas	30
Figura 3.3 – Janela para inserção de pontos por coordenadas	30
Figura 3.4 – Tela do Gmsh com imagem de fundo para cópia do contorno do domínio	31
Figura 3.5 – Malha bidimensional gerada em um domínio quadrado.....	32
Figura 3.6 – Botão de configurações de visualização	32
Figura 3.7 – Trecho do arquivo .geo que gera o domínio da Figura 3.5.....	33
Figura 3.8 – Trecho do arquivo .mesh gerado pela malha da Figura 3.5	36
Figura 3.9 – Janela de opções de para salvar parâmetros físicos	37
Figura 4.1 – Composição de estênceis da malha da Figura 3.5 em casos com e sem compartilhamento de vizinhos em uma camada	42
Figura 5.1 – Esquema de volumes de controle vizinhos I e J que dividem a fronteira AB	51
Figura 6.1 – Reconstrução da função 6.7	57
Figura 7.1 – Evolução da resolução da equação de Poisson.....	60
Figura 7.2 – Reconstrução da função 7.4	
Figura 7.3 – Evolução da resolução do problema do cone	62
Figura 7.4 – Evolução da resolução da Equação de Advecção-Difusão	63
Figura 7.5 – Reconstrução da função 6.7 em um domínio de geometria complexa ..	64

SUMÁRIO

1	INTRODUÇÃO	9
2	REVISÃO BIBLIOGRÁFICA	11
2.1	Introdução	11
2.2	Modelo matemático	11
2.3	Malhas computacionais	12
2.3.1	Malhas estruturadas	14
2.3.2	Malhas não-estruturadas	15
2.3.3	Malhas híbridas	16
2.4	Métodos de discretização de equações	17
2.4.1	Método das Diferenças Finitas – MDF	18
2.4.2	Método dos Elementos Finitos – MEF	18
2.4.3	Método dos Elementos de Contorno – MEC	19
2.4.4	Método Espectral	19
2.4.5	Método dos Volumes Finitos – MVF	20
2.5	Métodos de alta ordem	21
2.6	Métodos de integração	23
2.6.1	Quadratura gaussiana convencional	23
2.6.2	Quadratura gaussiana aplicada ao triângulo	24
3	MALHA E ESTRUTURA DE DADOS	27
3.1	Introdução	27
3.2	Geração de malhas no Gmsh	28
3.2.1	Construção manual do domínio	29
3.2.2	Construção via leitura de arquivo do domínio	32
3.2.3	Extraindo dados do Gmsh	35
3.3	Estrutura de dados	37

4	RECONSTRUÇÃO DE ALTA ORDEM	38
4.1	Introdução	38
4.2	Método de reconstrução de alta ordem via mínimos quadrados	38
4.3	Reconstrução na fronteira	43
4.3.1	Condição de contorno de Dirichlet	43
4.3.2	Condição de contorno de Neumann	44
4.3.3	Condição de contorno de Robin	45
4.3.4	Resolução do sistema	46
5	RESOLUÇÃO NUMÉRICA DA EQUAÇÃO DE ADVECÇÃO-DIFUSÃO	49
5.1	Introdução	49
5.2	Desenvolvimento da equação	49
5.3	Cálculo das integrais	51
5.4	Termo temporal	52
6	VERIFICAÇÃO COMPUTACIONAL	54
6.1	Introdução	54
6.2	Testes de verificação	55
7	TESTES DE VERIFICAÇÃO	59
7.1	Introdução	59
7.2	Equação de Poisson	59
7.3	Problema do cone	61
7.4	Advecção-difusão	61
7.5	Geometrias complexas	64
8	CONCLUSÃO	65
	REFERÊNCIAS	67

1 INTRODUÇÃO

Muitos fenômenos físicos em problemas de engenharia são modelados, isto é, representados matematicamente, por meio de Equações Diferenciais. Devido à limitação dos métodos analíticos quanto à presença de não-linearidade, a determinados tipos de condições iniciais e/ou de fronteira, e à geometria do domínio, muitas dessas equações só podem ser resolvidas numericamente. Por isso é tão importante o desenvolvimento de métodos numéricos que auxiliem na criação e no aprimoramento de tecnologias na engenharia.

Dentre as várias equações diferenciais conhecidas na literatura, a Equação de Advecção-Difusão é um modelo matemático representativo para fenômenos de transporte e difusão de propriedades físicas, como os que ocorrem em problemas de mecânica dos fluidos. Essa equação, no caso bidimensional, é dada por

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{\omega} \phi) - \nabla \cdot (\kappa \nabla \phi) = f \quad 1.1$$

em que $\vec{\omega} = (u, v)$ é o vetor velocidade de advecção, κ é o coeficiente de difusão, ambos variáveis, ou não, no domínio, e f , o termo fonte.

Uma questão importante no processo de resolução numérica de Equações Diferenciais Parciais (EDP), como a equação 1.1, é a geometria do domínio. Quando esta pode ser considerada complexa, o uso de malhas não-estruturadas é mais conveniente por sua melhor adaptação aos contornos da geometria. Isso exige o uso de esquemas de resolução numérica específicos, a exemplo do Método dos Volumes Finitos (MVF), que se adapta bem a esse tipo de malha. Uma questão importante no uso desse tipo de método é que ele exige esquemas de reconstrução para solução aproximada, que têm por finalidade calcular os fluxos nas faces dos volumes de controle.

Um dos esquemas existentes são os métodos de reconstrução de alta ordem embasados em mínimos quadrados, desenvolvidos por Carl Ollivier-Gooch (1997, 2011, 2001).

Este trabalho se propõe a implementar um código computacional em linguagem C para a resolução de Equações Diferenciais Parciais, sobre domínios de geometria complexa, com malhas não-estruturadas via Método dos Volumes Finitos,

utilizando o método de alta ordem embasado em mínimos quadrados desenvolvido por Gooch.

A seção 2 apresenta os principais temas estudados, com um breve resumo dos conceitos mais importantes para o trabalho durante sua elaboração, na ordem em que foram empregados

2 REVISÃO BIBLIOGRÁFICA

2.1 Introdução

A primeira etapa para se desenvolver um projeto acadêmico, depois de definidos seu tema e seu objetivo, é elaborar os passos necessários a esse desenvolvimento e realizar estudos mais aprofundados sobre cada um desses passos. Na revisão bibliográfica deste trabalho, fez-se uso de muitos livros e artigos, que são citados ao longo do capítulo. Para uma melhor exposição dos temas estudados, buscou-se apresentá-los na ordem em que são utilizados no trabalho. Nesse caso, primeiro é necessário conceituar um modelo matemático. Em seguida discutir sobre a discretização dos domínios e das equações do problema. Então parte-se para os métodos matemáticos disponíveis para a resolução e as ferramentas que o método escolhido vai exigir em sua implementação.

2.2 Modelo matemático

O que permite a resolução de problemas físicos é a modelagem matemática destes. Assim, a representação matemática de problemas por meio de equações diferenciais é algo comum, e a solução destas é o que irá proporcionar a análise dos referidos problemas. Boa parte dessas equações são Equações Diferenciais Parciais (EDP). São exemplos de equações diferenciais parciais que aparecem em problemas de engenharia:

- Equação de Difusão, aplicada, por exemplo, a problemas de transferência de calor;
- Equação de Poisson, muito utilizada em problemas de escoamento de fluidos.

Um único problema pode ser descrito por uma ou por um conjunto de equações, limitadas por restrições, ao qual se dá o nome de modelo matemático.

Quando se estuda um problema de engenharia, é importante definir um modelo matemático adequado, que represente, de forma coerente, o comportamento do fenômeno, tanto no interior de seu domínio como em seus contornos. Por isso, geralmente, o que se tem é uma equação que descreve o fenômeno no interior do

domínio e outras equações que fazem essa descrição para as fronteiras. No estudo de uma aeronave, por exemplo, existem diversos problemas que podem ser abordados, o escoamento do ar sobre as asas, a vibração gerada na estrutura durante o voo, a modelagem estrutural da carenagem, a distribuição de calor. E para cada um desses problemas, uma equação adequada ao fenômeno estudado, bem como as condições de contorno adequadas, devem ser aplicadas de forma que o comportamento real do fenômeno seja bem representado, dentro das condições que as hipóteses simplificadoras do problema exigem.

Especificamente sobre uma EDP, podem ser definidos três tipos de condições de contorno, de Dirichlet, que especifica o valor da solução sobre a fronteira; de Neumann, que especifica a derivada na direção normal à fronteira; e de Robin, ou híbrida, que é uma combinação linear das condições de Dirichlet e Neumann.

Neste trabalho, o modelo matemático tratado envolve a Equação de Advecção-Difusão, limitada por condições de contorno tanto de Dirichlet e Neumann como de Robin.

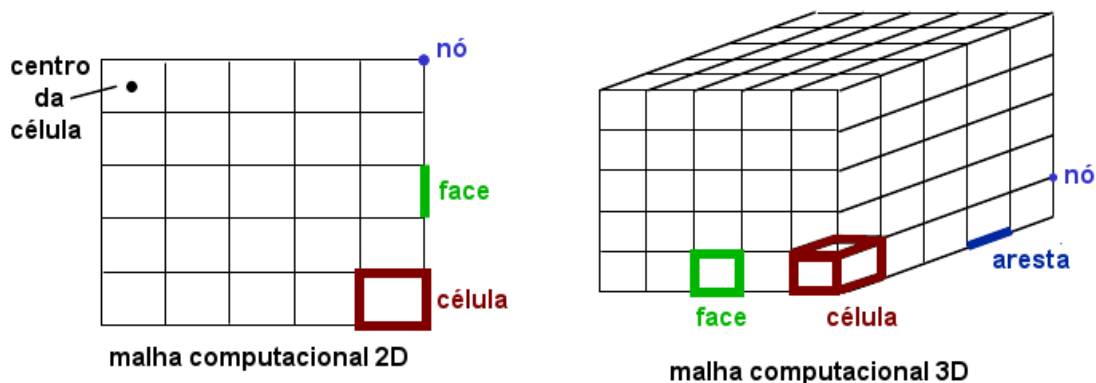
2.3 Malhas computacionais

A resolução numérica de um problema exige a discretização do domínio que se estuda. A discretização nada mais é do que a divisão do domínio em subdomínios, chamados de elementos, ou células; delimitados por faces e pontos, estes chamados de nós. Sobre os pontos centrais dos elementos, ou sobre os nós, as equações do problema são resolvidas. Para um mesmo domínio, de forma genérica, quanto mais subdomínios forem criados, menores eles serão, e melhor será a resposta aproximada obtida para o problema. Ressalta-se que esse é um conceito genérico porque o nível de refinamento da malha necessário a um problema depende da sua proposta. Quando existem, por exemplo, geometrias complexas, com muitas curvas e limites sinuosos do domínio, é de costume aplicar um refinamento da malha naquela região de contornos mais complexos, porque o método exige esse refinamento para se obter uma solução tão precisa nas regiões complexas quanto nas mais simples. Todavia, nas regiões mais simples, esse refinamento pode não ser necessário, de forma que o ganho em precisão nessa região seja ínfimo. Esses aspectos sempre dependem do nível de precisão que o problema exige, e nesses momentos, a consulta a fontes confiáveis, de

pesquisadores mais experientes na área, é de grande importância. Um conceito bastante utilizado para determinar o grau de refinamento de uma malha é a *independência de malha*, onde os resultados em malhas com refinamento subsequentes são comparados, e uma diferença mínima entre os resultados é pré-estabelecida. Quando se atinge essa diferença mínima, já não são mais necessários refinamentos subsequentes, pois o incremento de precisão do método já não é mais significativo.

Ainda assim, os problemas em geral são resolvidos por malhas com um número de elementos que extrapolam os limites humanos para a realização dos cálculos. Por isso há a necessidade do uso de computadores na resolução numérica de problemas. O conjunto de células que aproximam a forma de um domínio é chamado de malha. A Figura 2.1 ilustra bem essa terminologia aplicada à construção de malhas em domínios bidimensionais e tridimensionais.

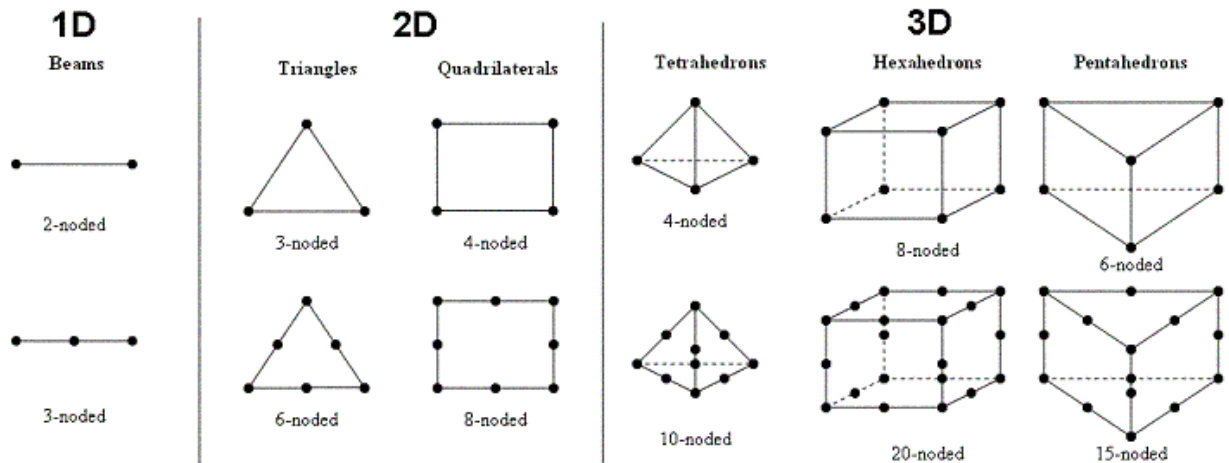
Figura 2.1 - Terminologia das partes de uma malha



Fonte: Adaptada de Bakker (2002-2006).

Quando se determina o número de elementos que constitui uma malha, busca-se a melhor adaptação possível destes à geometria do domínio estudado. Além da utilização de um número elevado de elementos para essa adaptação, os elementos podem também assumir diferentes formas, que se adequem melhor à geometria. A Figura 2.2 mostra alguns dos principais tipos de elementos que podem compor uma malha.

Figura 2.2 - Tipos de elementos de malha



Fonte: Adaptada de Moreno (2011).

As malhas, comumente, são divididas em dois grupos principais, as estruturadas e as não-estruturadas. Cada um desses tipos tem ainda subdivisões que classificam os tipos de malha. Esse assunto é discutido em detalhes por Gonçalves (2007). Para este trabalho, bastam as principais diferenças e aplicações das malhas estruturadas e não-estruturadas.

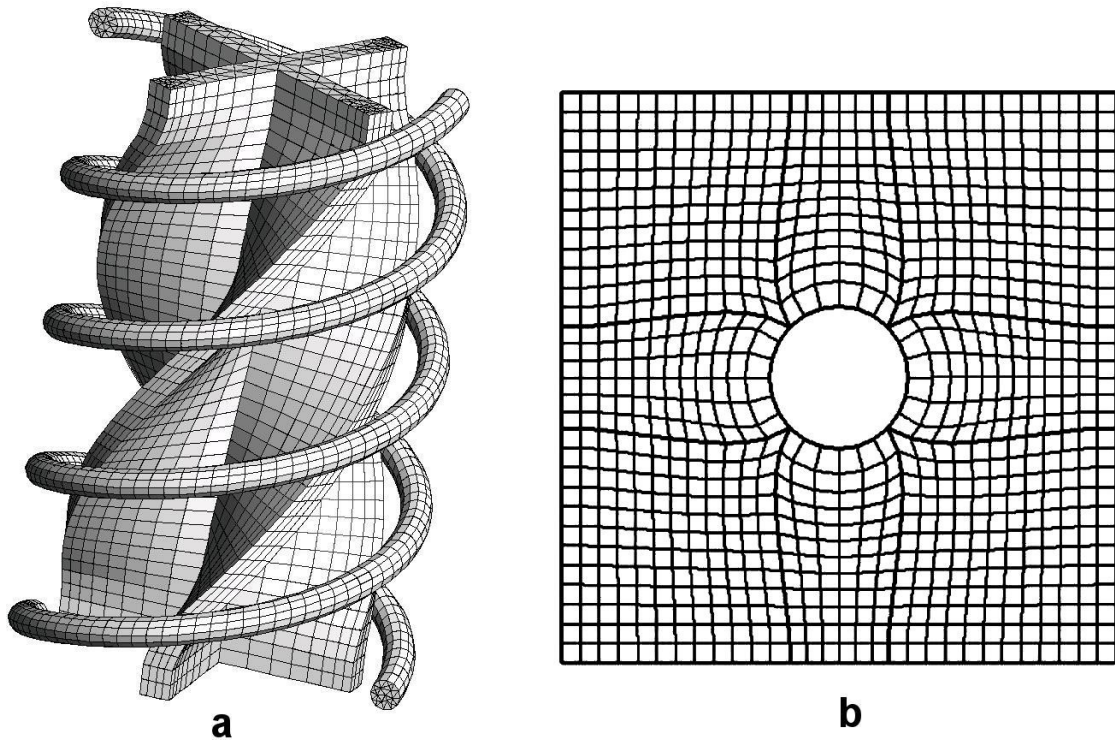
2.3.1 Malhas estruturadas

A principal característica das malhas estruturadas, ou regulares, notada logo de início, é seu padrão de organização. Como definição, a localização dos nós vizinhos a um dado nó é determinada por variações de índices em cada direção. Além disso, todos os seus elementos são posicionados segundo um sistema de coordenadas, e todos eles (exceto os da fronteira) possuem a mesma quantidade de vizinhos, o que lhes permite serem numerados facilmente, seguindo uma ordem simples. A Figura 2.3 mostra exemplos de malhas estruturadas.

Essa organização facilita a montagem das matrizes para resolver as equações envolvidas, implicando em menor demanda computacional, além de haver muitos algoritmos de discretização disponíveis para esse tipo de malha. Entretanto, há algumas dificuldade em aplicá-la a geometrias mais complexas. Isto porque, em quinas e curvas acentuadas, o número de divisões da malha deve ser extremamente grande para se adequar razoavelmente aos contornos do domínio. Existem técnicas para utilização de malhas estruturadas em domínios mais complexos, porém são

limitadas, tanto pela geometria do domínio como pela baixa adaptabilidade dos métodos de discretização (Gonçalves, 2017).

Figura 2.3 - Malhas estruturadas em domínios a) tridimensional e b) bidimensional



Fonte: a) Adaptada de Sussman (2016); b) Adaptada de Pointwise (2012).

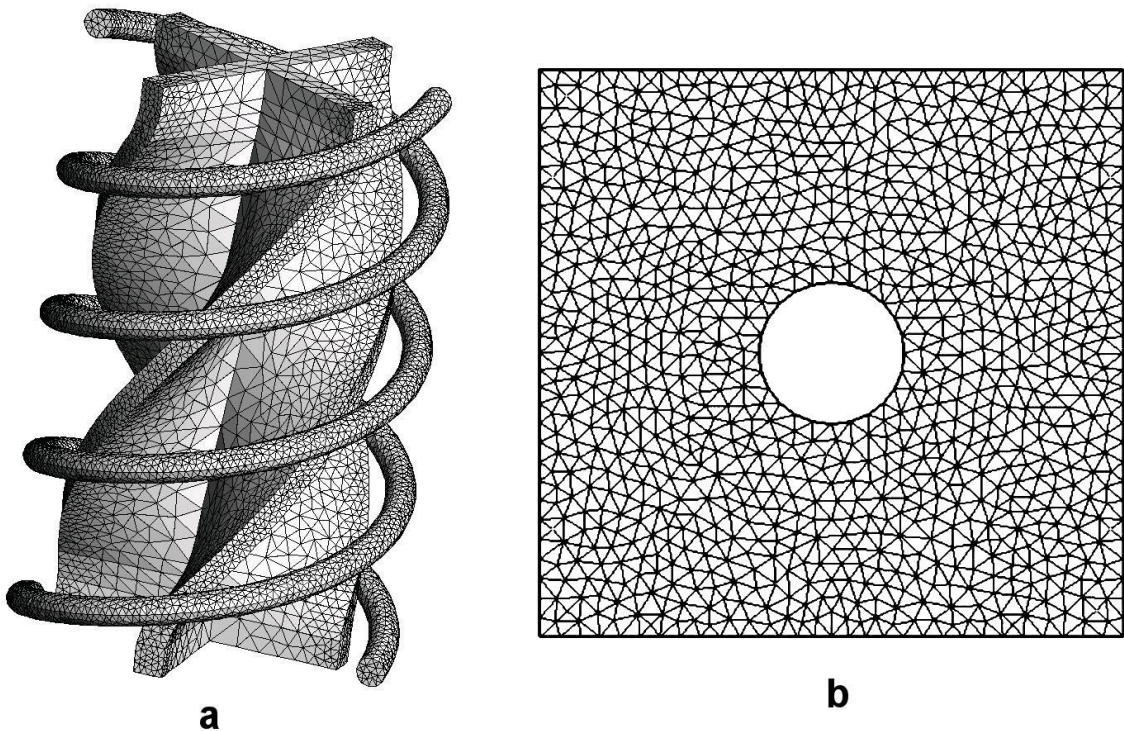
2.3.2 Malhas não-estruturadas

A ideia das malhas não-estruturadas é muito bem traduzida pela composição de triângulos e tetraedros sobre um domínio. Apesar disso, seus elementos podem assumir qualquer forma, com variados números de vértices, além de diferentes números de vizinhos. Isso confere o aspecto de irregularidade marcante desse tipo de malha, ilustrado na Figura 2.4.

Ao contrário das malhas estruturadas, a posição das células não está associada a um sistema de coordenadas, o que dificulta a organização de sua estrutura de dados, além de tornar mais trabalhosa a montagem de matrizes. Os algoritmos de discretização são, em geral, mais complexos, e os de resolução dos sistemas de equações são claramente mais lentos. Apesar disso, seus códigos computacionais são altamente adaptáveis, não sendo necessárias alterações para

refinamento de regiões da malha ou mesmo para mudança da forma de seus elementos. Isso garante a elaboração de programas genéricos, aplicáveis a diferentes problemas, com diferentes domínios.

Figura 2.4 - Malhas não-estruturadas em domínios a) tridimensional e b) bidimensional



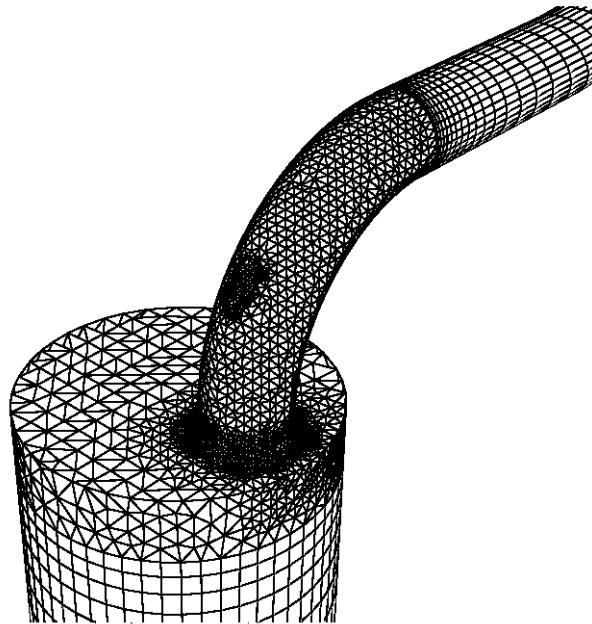
Fonte: a) Adaptada de Sussman (2016); b) Adaptada de Pointwise (2012).

Neste trabalho, serão tratadas somente malhas não-estruturadas sobre domínios de geometrias diversas. Estas são construídas com o auxílio de um software livre chamado *Gmsh* (Geuzaine e Remacle).

2.3.3 Malhas híbridas

Dadas as vantagens e desvantagens das malhas estruturadas e não estruturadas, nada impede que ambas sejam aplicadas a um mesmo problema, se isso for vantajoso. A Figura 2.5 mostra uma malha híbrida sobre uma peça de um motor de combustão interna.

Figura 2.5 - Malha híbrida aplicada a peça de motor de combustão interna



Fonte: Adaptada de Kumar.

Nota-se que, em determinadas regiões da peça, como na conexão entre os cilindros, a complexidade geométrica da união exige uma malha não-estruturada para melhor aproximar a forma do domínio. Contudo, na extensão dos cilindros conectados, não há irregularidades ou curvas acentuadas, logo o uso de malha estruturada nessas regiões facilita a resolução do problema e economiza recursos computacionais, mais necessários à parte não-estruturada da malha. É claro que trabalhar os algoritmos para que as fronteiras entre ambas as malhas apresentem dados coerentes entre si é muito importante. Quando a capacidade computacional da máquina utilizada é limitada, essa técnica apresenta vantagens.

2.4 Métodos de discretização de equações

Depois da discretização do domínio, deve-se estabelecer um método de discretização das equações tratadas no problema. A função desse processo é transformar a EDP que rege o problema em um conjunto de equações algébricas, que serão resolvidas em cada ponto da malha. São exemplos de métodos de discretização utilizados são o Método das Diferenças Finitas (MDF), o Método dos Elementos Finitos (MEF), o Método dos Elementos de Contorno (MEC), o Método dos Volumes Finitos (MVF) e os Métodos Espectrais.

2.4.1 Método das Diferenças Finitas – MDF

No Método de Diferenças Finitas, a discretização pode ser feita através da interpolação polinomial da função incógnita, ou através da expansão em série de Taylor desta. Nesse último caso, para cada ponto da malha, a expansão gera equações que, com algumas manipulações, fornecem aproximações para as derivadas da EDP em função dos pontos vizinhos. Quantos e quais pontos vizinhos são utilizados, e qual a ordem do erro que a aproximação gera são dados pertinentes à forma como a aproximação das derivadas é obtida, dada a manipulação da série de Taylor. Quando se faz o tratamento polinomial para aproximar as derivadas, expressões similares são encontradas. A literatura conta com fórmulas, já desenvolvidas, para aproximações de derivadas de primeira até quarta ordem, através de diferenças progressiva, regressiva e central. Os livros de Hoffmann (2000) e Burden (2010) fazem algumas demonstrações das principais fórmulas de diferenças finitas.

Este é um dos métodos mais antigos desenvolvidos para solução numérica de equações (DEMEC, 2015), utilizado amplamente em malhas estruturadas, o que limita sua aplicação. É um método de aplicação relativamente fácil, com boa adaptação a termos advectivos, contudo limitado a problemas de geometria mais simples, muito utilizado em problemas de mecânica dos fluidos.

2.4.2 Método dos Elementos Finitos – MEF

No Método dos Elementos Finitos, o domínio do problema é dividido em um número finito de subdomínios, sobre os quais as equações serão resolvidas. Este é um método de formulação integral, em que uma forma integral do problema, no caso, da EDP, chamada de forma fraca, deve ser obtida para que o problema seja resolvido. De maneira simplificada, a forma fraca restringe as possíveis soluções do problema, facilitando sua resolução.

Existem três métodos através dos quais se pode obter a forma fraca. No método da integral ponderada, a EDP é multiplicada por uma função de ponderação e só então integrada no domínio do problema. No método variacional, a forma fraca é obtida pela minimização de um funcional dentro de um espaço de funções válidas para o problema. Por fim, no método dos trabalhos virtuais, a partir de um deslocamento virtual de uma estrutura, são determinados os trabalhos virtuais

internos e externos, que quando igualados, resultam na equação da forma fraca. Cada método de obtenção da forma fraca possui suas peculiaridades, e a escolha entre eles depende do problema tratado. Mais detalhes sobre essas formulações são encontrados na apostila Introdução ao MEF (DEMEC).

Esse método surgiu nos anos 40, mas só a partir dos anos 60 começou a ser amplamente utilizado, devido à evolução dos computadores. É um dos métodos utilizados na resolução de EDP, encontrado em diversos softwares computacionais, como CONSOL, Nastran e ANSYS, geralmente associado a malhas não-estruturadas, na resolução de problemas de engenharia diversos (estruturais, térmicos, eletromagnéticos). Suas grandes vantagens são a facilidade de implementação computacional, devido a sua flexibilidade e estabilidade numérica, e a aplicação a problemas envolvendo geometrias complexas, como os da indústria aeroespacial, por exemplo.

2.4.3 Método dos Elementos de Contorno – MEC

O Método dos Elementos de Contorno, em uma explicação mais simplificada, é similar ao MEF, com a diferença que somente o contorno do domínio é discretizado. Isso simplifica a geração da malha e reduz consideravelmente o tamanho do problema tratado. Dentre as principais aplicações desse método, estão os domínios infinitos ou semi-infinitos, como problemas de acústica e de mecânica dos solos, e problemas como propagação de trincas, em que o contorno muda constantemente (GFFM).

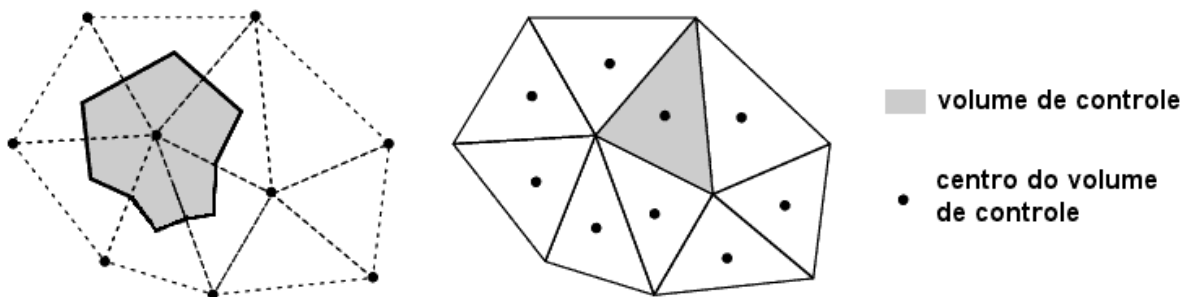
2.4.4 Método Espectral

O Método Espectral envolve técnicas de resolução de equações diferenciais através da aplicação de Transformadas Rápidas de Fourier (TRF). A ideia é escrever a solução para as equações na forma de somatórios de funções básicas, determinando os coeficientes dos termos da soma que melhor satisfazem a equação diferencial. Mais sobre os Métodos Espectrais pode ser estudado no livro de Canuto (2007).

2.4.5 Método dos Volumes Finitos – MVF

O Método dos Volumes Finitos funciona da seguinte forma. Similar ao MEF, o domínio contínuo é discretizado, dividido em elementos menores, sendo cada elemento chamado volume de controle (VC), sobre os quais as equações serão discretizadas. Para isso, duas técnicas de armazenamento podem ser utilizadas, a cell-centered e a vertex-centered. Na primeira, as informações relativas ao volume de controle são armazenadas no centroide dos elementos da malha, e na segunda, nos vértices dos elementos da malha, de forma que os limites do volume de controle são estabelecidos pela ligação entre os centroides dos elementos da malha e o ponto médio de suas faces. Os esquemas são melhor ilustrados da Figura 2.6.

Figura 2.6 - Esquemas de armazenamento para o MVF, vertex-centered à esquerda e cell-centered à direita



Fonte: Adaptada de Hienzl.

Cada esquema de armazenamento tem vantagens e desvantagens. A técnica cell-centered envolve menos equações na montagem dos sistemas matriciais de resolução e uma estrutura de dados mais simples para a construção dos estêncils¹, porém resulta em um número grande de VC. Em contrapartida, a técnica vertex-centered envolve maior complexidade no armazenamento dos estêncils e resulta em matrizes maiores, porém reduz significativamente o número de VC avaliados.

Definidos os volumes de controle, o MVF parte da integração das EDP sobre cada VC, requerendo métodos numéricos de integração. A forma das integrais a serem resolvidas (de linha, de superfície, de volume) vai depender do tipo de domínio (unidimensional, bidimensional, tridimensional). Esse método é, sobretudo,

¹ Estêncil: conjunto de volumes de controle dispostos em camadas segundo sua proximidade com o volume de controle central.

conservativo, isto é, dados os problemas físicos que aborda, o MVF trabalha com balanços de massa, quantidade de movimento e quantidade de energia nos VC . Logo os fluxos das variáveis que atravessam as faces dos VC também exigem um método de cálculo para a obtenção de suas soluções aproximadas.

O MVF pode ser aplicado a qualquer tipo de malha, com qualquer tipo de VC , o que lhe assegura utilização em domínios de geometria complexa. Suas características o tornam atrativo para problemas embasados na lei da conservação, o que explica sua vasta utilização em problemas de mecânica dos fluidos em geral. Segundo Gonçalves (2007), o fato de todos os termos aproximados pelo método terem significado físico, facilita sua compreensão e o torna mais popular entre engenheiros.

Neste trabalho, o MVF foi escolhido para realizar a discretização das equações do modelo matemático. A técnica de armazenamento de dados escolhida foi a cell-centered.

2.5 Métodos de alta ordem

Para resolver um problema numérico, é interessante que os erros envolvidos sejam os menores possíveis. Nesse caso, uma maior ordem de aproximação, desde que não haja descontinuidades abruptas na solução, pode trazer melhores resultados ao problema (Gonçalves, 2007). Isto porque os erros de aproximação de alta ordem decrescem, com o refinamento da malha, mais rápido que com o uso de baixa ordem. Além disso, em mecânica dos fluidos, por exemplo, na análise de ondas, a alta ordem proporciona menores dissipação e dispersão, e em escoamentos com vórtices, preserva esses vórtices por maiores distâncias (ZhiJian, 2016).

Contudo, apesar de uma melhora na exatidão das soluções numéricas, o uso de ordens elevadas gera, muitas vezes, problemas de convergência e estabilidade, produzindo oscilações numéricas, o que motiva o estudo de técnicas de aplicação que minimizem tais problemas. Os esquemas desenvolvidos para eliminar essas oscilações numéricas são conhecidos como High Resolution Schemes (HRS), ou métodos de alta ordem (van Leer, 1979). Tratam-se de métodos conservativos, aplicados a descontinuidades de propriedades e ondas de choques, que eliminam oscilações numéricas para uma solução suave.

O conceito de alguns HRS pode ser resumido da seguinte forma. O Flux-Corrected Transport (FCT) foi desenvolvido por Boris e Book (1973), muito utilizado em funções hiperbólicas. É composto de dois estágios, primeiro o estágio convectivo, ou de transporte, e o segundo estágio anti-difusivo, ou corretivo, que corrige os erros numéricos do primeiro estágio. O Total Variation Diminishing (TVD), desenvolvido por Harten (1983), também se aplica a equações hiperbólicas, e atua diminuindo a variação total da solução discreta, evitando oscilações incorretas. O Monotonic Upstream Scheme for Conservation Law (MUSCL), desenvolvido por van Leer (1979), aplica funções limitadoras de fluxo ao problema.

Em uma classe mais distinta dos métodos de alta ordem, o Essentially Non-Oscillatory (ENO), proposto por Harten (1983), se baseia na medição e escolha dos estênceis mais suaves para aproximar a derivada da função incógnita. Isso permite que diferentes conjuntos de pontos sejam selecionados de forma a evitar pontos distintos de gradientes muito elevados, ou mesmo de descontinuidades, nas fórmulas de interpolação (Barbosa, 2013). Esse esquema conserva as médias e garante que as descontinuidades não sejam maiores que a ordem de truncamento do erro da reconstrução (Ollivier-Gooch, 1997). Entretanto, pode fazer o problema convergir para seu estado estacionário, mesmo que essa não seja sua solução. Esse problema é amenizado por uma derivação do ENO, o Weighted Essentially Non-Oscillatory (WENO), de Liu, Osher e Chan (1994), que utiliza uma combinação ponderada dos estênceis possíveis. Nesse caso, os estênceis que contêm pontos distintos de uma descontinuidade são associados a pesos da ordem do erro de truncamento da reconstrução.

O esquema chamado reconstrução k -exata, estudado por Barth e Frederickson (1990) trabalha com a reconstrução de funções com exatidão de ordem k . Ele garante a conservação das médias e boa exatidão para funções suaves. Utiliza-se um estêncil fixo, o que melhora a eficiência computacional do método. Apesar disso, a precisão do método próximo a descontinuidades é perdida.

Em seu trabalho, Gooch (1997) mescla propriedades dos métodos WENO e reconstrução k -exata e desenvolve um novo método chamado DD-L2-ENO, que utiliza reconstrução ponderada com estênceis fixos. O que o diferencia do método de reconstrução k -exata é o uso de ponderação em cada ponto do estêncil para suavizar os dados de reconstrução. No caso do WENO, a característica pontual da ponderação é o que diferencia o método de Gooch. Uma vantagem significativa do

DD-L2-ENO é sua capacidade de degradar a ordem de reconstrução em regiões com dados insuficientes para a reconstrução.

Neste trabalho, foi utilizado um método de alta ordem. Dentre os estudados, foi escolhido o método de Gooch.

Falando um pouco mais sobre os métodos de alta ordem, é fato que ainda existem alguns julgamentos injustos sobre eles, especialmente no que diz respeito à demanda computacional. Para uma mesma malha, evidentemente, o método de alta ordem será muito mais lento. A questão é que, buscando um mesmo erro de aproximação para determinado problema, o método de alta ordem é capaz de obter esse erro com uma malha computacional muito mais grosseira que um método de baixa ordem. O trabalho de Wang ZhiJian (2016) trata com mais detalhes as vantagens dos métodos de alta ordem, especialmente para os problemas de Dinâmica de Fluidos Computacional (CFD), exemplificando graficamente resultados equivalentes de métodos de baixa ordem aplicados a malhas refinadas e de alta ordem aplicados a malhas grosseiras. Vale lembrar que, para soluções com descontinuidades, os métodos de alta ordem usualmente perdem sua exatidão, fornecendo erros da ordem de métodos de baixa ordem, por isso o caso em que se pretende aplicá-los deve ser bem avaliado.

2.6 Métodos de integração

Como se pode notar nas seções anteriores, métodos de integração são necessários para resolver o problema proposto. Existem inúmeros métodos numéricos de integração, logo seria oneroso tratar todos aqui. No presente trabalho, o método abordado foi o método de quadratura gaussiana, em que são realizadas aproximações por polinômios. Adicionalmente, foi ainda estudado um método de quadratura gaussiana aplicado a um triângulo, desenvolvido e explicado nos trabalhos de Dunavant (1985).

2.6.1 Quadratura gaussiana convencional

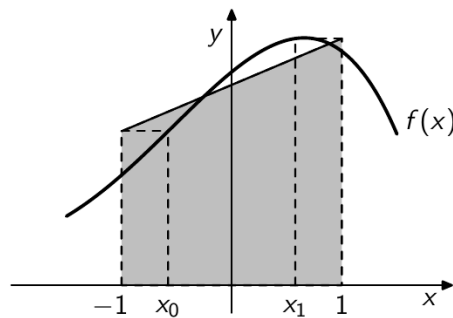
No método de integração via quadratura gaussiana, o valor da integral da função é aproximado por um polinômio interpolador do tipo

$$I = \int_a^b f(x) dx \approx A_0 f(x_0) + A_1 f(x_1) + \dots + A_n f(x_n) \quad .1$$

em que os pesos A_k e os pontos de integração x_k são escolhidos de forma a se obter a melhor aproximação possível da integral (Biloti, 2017). Nesse caso, o procedimento adotado é aplicar uma mudança de domínio, de $[a, b]$ para $[-1, 1]$, para que os pontos e pesos de Gauss-Legendre possam ser escolhidos segundo uma tabela de pontos e pesos já desenvolvida. A

Figura 2.7 mostra um exemplo da aplicação dos polinômios de Gauss-Legendre para a aproximação de uma função qualquer $f(x)$.

Figura 2.7 - Integração numérica através de polinômios de Gauss-Legendre



$$I = \int_{-1}^1 f(x) dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right) \equiv Q_{GL}[f]$$

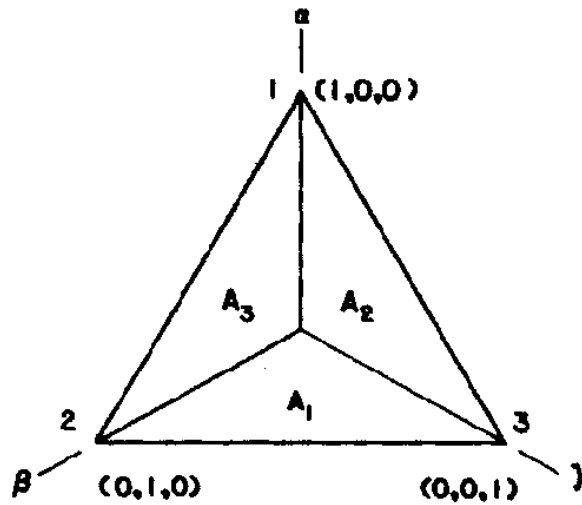
Fonte: Biloti (2017).

O número de termos do polinômio aproximador, ou seja, de pontos de Gauss utilizado, para este trabalho, foi determinado de acordo com a ordem de reconstrução usada no método.

2.6.2 Quadratura gaussiana aplicada ao triângulo

Em seu trabalho, Dunavant (1985) utiliza o que chama de formulação isoparamétrica para transformar um sistema de coordenadas qualquer no chamado sistema de coordenadas natural, representado por um triângulo, conforme a Figura 2.8.

Figura 2.8 - Coordenadas naturais aplicadas ao triângulo



Fonte: Dunavant (1985).

Nesse contexto, as regras de quadratura gaussiana são aplicadas a um conjunto de coeficientes α , β e γ , definidos por

$$\alpha = \frac{A_1}{A}; \beta = \frac{A_2}{A}; \gamma = \frac{A_3}{A} \quad \alpha + \beta + \gamma - 1 = 0 \quad 2.2$$

em que A é a área do triângulo natural. A integral da função é então dada por

$$I = \int_A f(\alpha, \beta, \gamma) dA = A \sum_{i=1}^{ng} w_i f(\alpha_i, \beta_i, \gamma_i) \quad 2.3$$

em que w_i são os pesos de Gauss para o i -ésimo ponto de Gauss, avaliado em $(\alpha_i, \beta_i, \gamma_i)$. Nesse caso, o erro da quadratura tende a zero conforme aumenta o número de pontos de Gauss ng .

De forma mais simplificada, a aplicação da técnica de integração de Dunavant a uma equação bidimensional, descrita apenas em função das variáveis x e y , se resume a revolver a integral

$$I = \iint_{vc} f(x, y) dx dy = A \sum_{i=1}^{ng} w_i f(x_i, y_i) \quad 2.4$$

sendo que as coordenadas dos pontos (x_i, y_i) são combinações lineares das coordenadas dos VC com os coeficientes α , β e γ como pesos dessas combinações, isto é

$$x_i = \alpha_i x_A + \beta_i x_B + \gamma_i x_C \quad e \quad y_i = \alpha_i y_A + \beta_i y_B + \gamma_i y_C, \quad i = 1, 2, 3 \dots ng \quad 2.5$$

sendo (x_A, y_A) , (x_B, y_B) e (x_C, y_C) as coordenadas dos vértices do VC triangular, e ng o número de pontos de Gauss. Nesse caso, α_i , β_i e γ_i representam as coordenadas relativas dos pontos de Gauss, localizados sobre as faces do triângulo natural. Essas coordenadas $(\alpha_i, \beta_i, \gamma_i)$, bem como os pesos de Gauss w_i , têm uma forma de obtenção, que Dunavant aborda em seu trabalho. Posteriormente, ele ainda monta uma tabela com os valores de todas as coordenadas $(\alpha_i, \beta_i, \gamma_i)$ e pesos w_i para cada ordem de precisão da integração, de 1 a 20. Vale lembrar ainda que, para algumas ordens, é necessário permutar as coordenadas do VC com as coordenadas dos pontos de Gauss. Para mais detalhes da forma de resolução, o trabalho de Magalhães (2016) explica o processo de permutação de coordenadas e ainda exemplifica a aplicação do método com um código implementado usando a linguagem do software *Octave*.

Após apresentado todo o material de estudos para o desenvolvimento do trabalho, parte-se para a etapa de desenvolvimento. As seções 3, 4 e 5 apresentam todas as etapas até a resolução do problema proposto no trabalho.

3 MALHA E ESTRUTURA DE DADOS

3.1 Introdução

Foi visto que, após definido um modelo matemático, é preciso construir uma malha sobre o domínio estudado. Como dito no capítulo 2, neste trabalho serão tratadas somente as malhas não-estruturadas, estas, por sua vez, geradas por um software livre, chamado Gmsh (Geuzaine e Remacle). Após definida a geometria bidimensional do domínio e gerada a malha sobre ela, os dados da malha pertinentes à resolução de EDP são armazenados por um código em linguagem C, e a sua estrutura de dados é então montada. Neste capítulo, detalhes sobre o software de geração de malhas, bem como uma prévia sobre o código de montagem da estrutura de dados serão abordados.

As malhas não-estruturadas são preferidas especialmente na discretização de domínios de geometrias complexas, onde a adaptação de malhas estruturadas se torna trabalhosa e com uma elevada demanda computacional. Existem vários métodos para se construir uma malha não-estruturada sobre um domínio, mas a maior parte deles, segundo Owen (1998), se enquadra em uma das três categorias, Método Octree, Método Delaunay e Método do Avanço da Face, que, em resumo, funcionam da seguinte forma:

i. Método Octree: quadrados/cubos são sequencialmente subdivididos em quadrados/cubos menores para adequação aos contornos da forma até que se atinja a resolução desejada.

ii. Método Delaunay: são traçados triângulos/tetraedros a partir de um conjunto de nós. A forma como são definidas as ligações entre os nós é embasada no critério do círculo/esfera vazia, em que o círculo/esfera que passa pelos três pontos da célula triangular/tetraédrica da malha não deve conter nenhum outro nó. Dessa forma, o tamanho das células definidas na malha, bem como os ângulos formados por elas, fica melhor distribuído no domínio.

iii. Método Avanço da Face: o primeiro nível de triângulos/tetraedros da malha é definido ao longo de toda a fronteira. Segmentos de retas, para o caso

bidimensional, ou triângulos, para o caso tridimensional, são estabelecidos ao longo de toda a fronteira do domínio, e a partir de algoritmos específicos, os terceiros nós que formam os triângulos (ou quartos, que formam os tetraedros) da malha, para toda extensão da fronteira, são estabelecidos de forma otimizada, e então, a malha cresce da fronteira para seu interior, até preencher todo o domínio com células.

Não é mérito desse trabalho descrever em detalhes cada um desses métodos, pois a geração manual de malhas não-estruturadas é dispendiosa e exige um estudo aprofundado sobre o assunto. Para agilizar o processo, como o foco deste trabalho está nos cálculos sobre as malhas não-estruturadas, a geração destas foi feita de forma automática, pelo software *Gmsh*. O software permite escolher entre três algoritmos de reconstrução de malhas bidimensionais, e dois para malhas tridimensionais. Por padrão, as malhas são geradas pelo método Delaunay, e este foi mantido.

Abordaremos agora o passo a passo para a geração de malhas neste programa. O foco do trabalho são os domínios bidimensionais, por isso a ferramenta será explorada apenas nessa extensão.

3.2 Geração de malhas no Gmsh

O *Gmsh* é um programa livre com uma interface gráfica de criação de desenhos amigável e de fácil utilização. Para construção de geometrias, é possível fazê-lo inserindo manualmente as coordenadas dos pontos de referência do desenho ou puxando informações para o programa através de um arquivo de extensão compatível, nesse caso, extensão *.geo*. A segunda forma mostra-se mais interessante neste trabalho, especialmente para a construção de geometrias complexas, em que muitos pontos são necessários para se determinar os limites do domínio.

Nas duas opções, o primeiro passo é criar os contornos do domínio sobre o qual a malha será gerada. De início, é preciso determinar os nós da fronteira com base em um sistema de coordenadas cartesianas (x, y, z) , e então unir esses nós com linhas. Essas linhas, no caso do *Gmsh*, podem ser segmentos de reta, segmentos de círculos, de elipses, splines cúbicas, com coeficientes determinados automaticamente pelo software. Com os contornos do domínio finito bem definidos,

cria-se uma superfície, que delimitará a região do interior do domínio. Só então é gerada a malha não-estruturada sobre o esse domínio informado.

3.2.1 Construção manual do domínio

Ao abrir o programa, a tela mostrada na Figura 3.1 é exibida. São observadas as barras de opções nas partes inferior e superior, e à esquerda, um esquema de árvore com os comandos para desenho, construção de malha e aplicação de solver. No canto inferior direito, uma ilustração do sistema de coordenadas cartesianas auxilia na orientação do desenho no espaço tridimensional.

Figura 3.1 – Tela inicial do Gmsh



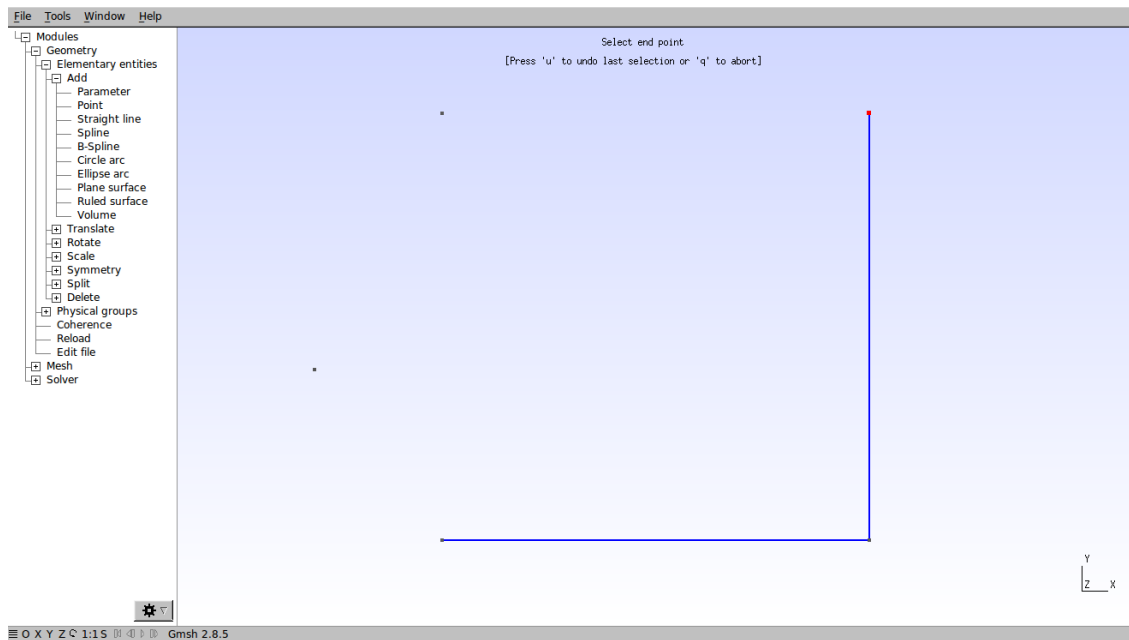
Fonte: Produzida pela autora.

A árvore no canto esquerdo mostra um menu expansível, em que as ferramentas de desenho são apresentadas. Para a construção de uma forma básica, como um quadrilátero, são desenhados quatro pontos no espaço, depois estes são interligados por retas, como mostra a Figura 3.2.

No momento de inserir os pontos, é possível determiná-los clicando no local desejado ou inserindo as coordenadas na janela que se abre automaticamente. Nessa aba, um parâmetro importante a ser determinado, é o comprimento característico l_c dos elementos da malha, que define a referência de tamanho das

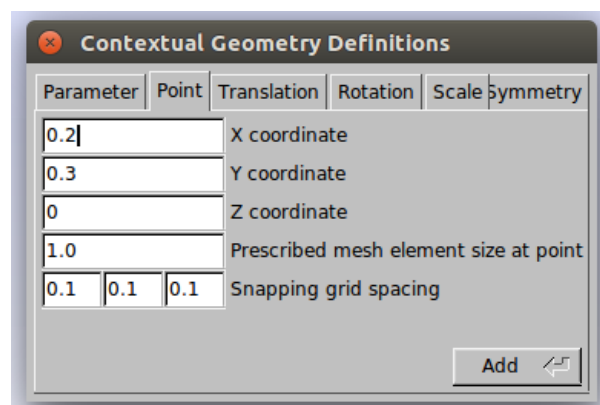
células da malha. Esse parâmetro será abordado com mais detalhe no próximo tópico. A Figura 3.3 mostra a janela exibida para inserção das coordenadas do pontos desenhados e do comprimento característico associado àquele ponto.

Figura 3.2 – Tela do Gmsh evidenciando menu expansível e desenho de pontos e retas



Fonte: Produzida pela autora.

Figura 3.3 – Janela para inserção de pontos por coordenadas

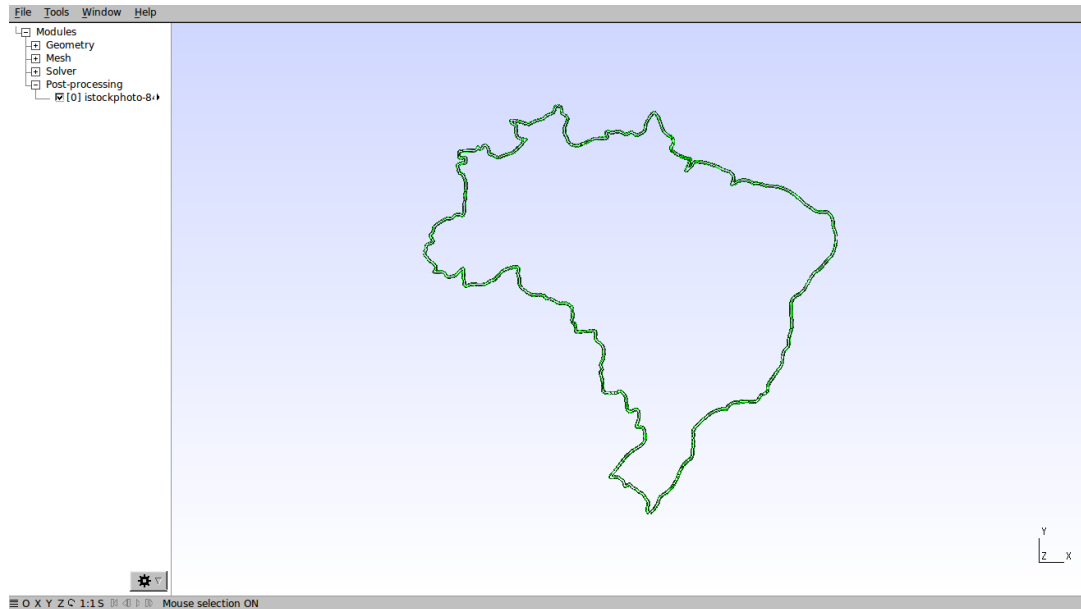


Fonte: Produzida pela autora.

É possível, ainda, utilizar uma imagem base como fundo de tela para se copiar os contornos do domínio. Por exemplo, em um domínio com a forma do mapa

do Brasil, é possível inserir tal imagem no programa para inserir os nós da fronteira com base no contorno dado pela figura, como mostra a Figura 3.4.

Figura 3.4 – Tela do *Gmsh* com imagem de fundo para cópia do contorno do domínio

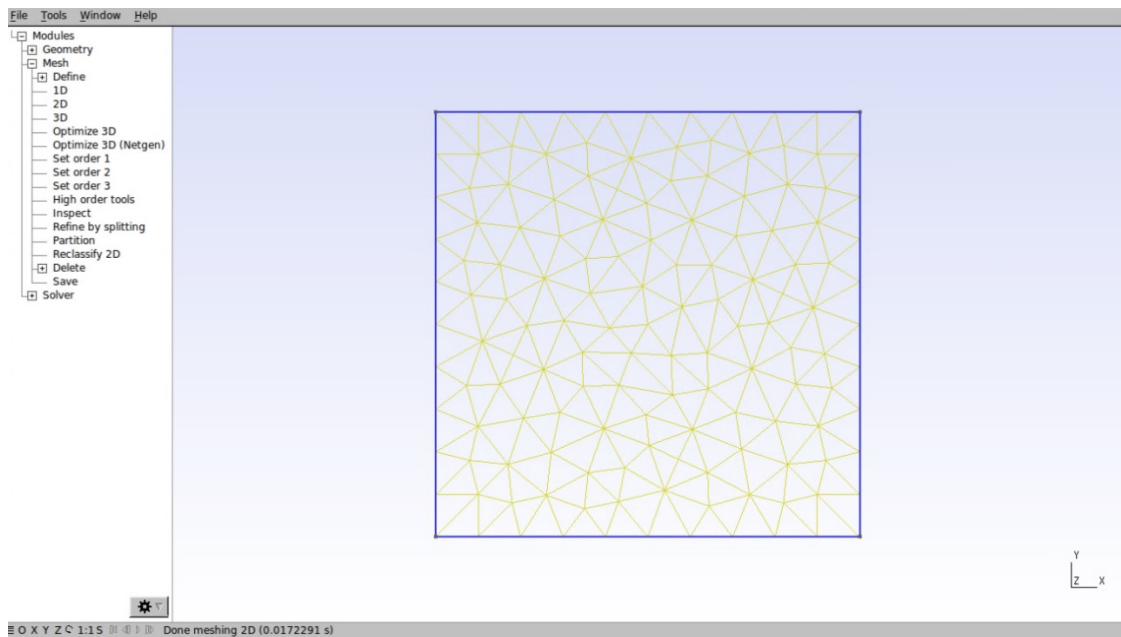


Fonte: Produzida pela autora.

Após um domínio finito estar determinado, para a construção da malha, é preciso estabelecer um plano sobre o domínio criado. Após utilizar a ferramenta de construção do plano, basta expandir a aba de *mesh* e clicar na opção “2D”. O sistema vai processar os dados do domínio, criar a malha de triângulos sobre o plano e exibi-la nas configurações padrão do programa, como mostra a Figura 3.5.

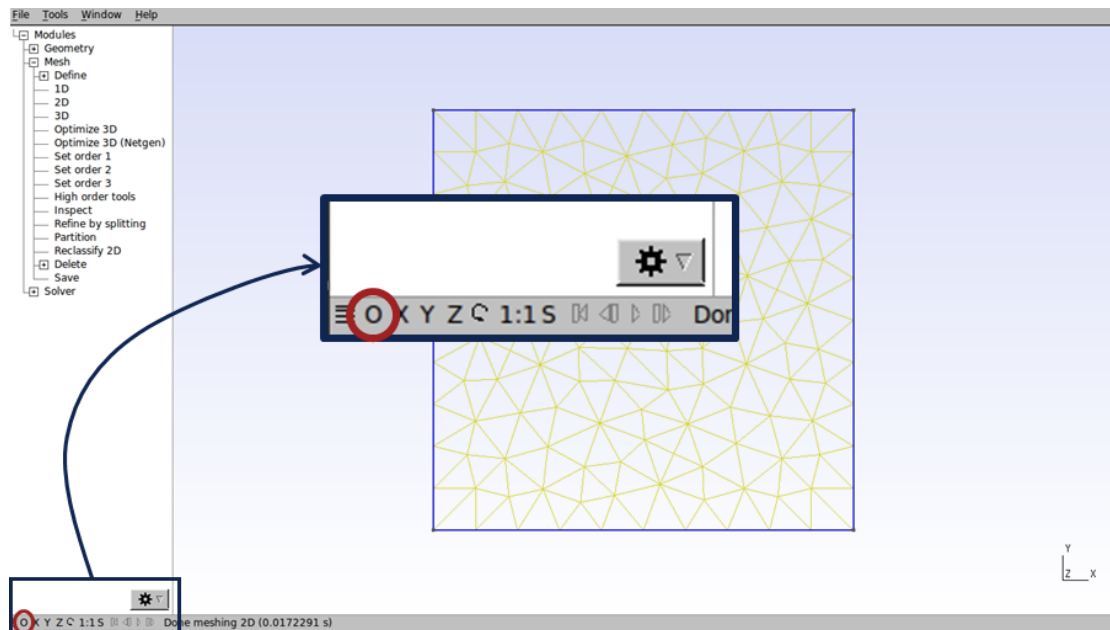
O tamanho médio das células é determinado pelo parâmetro lc , setado no momento da construção dos nós, e não é facilmente modificado após construída a malha sobre o domínio. Opções de visualização, como índice de nós e faces, identificação de fronteiras, cores de cada elementos, estão disponíveis para edição na aba inferior esquerda, clicando no botão “O”, destacado na Figura 3.6.

Figura 3.5 – Malha bidimensional gerada em um domínio quadrado



Fonte: Produzida pela autora.

Figura 3.6 – Botão de configurações de visualização



Fonte: Produzida pela autora.

3.2.2 Construção via leitura de arquivo do domínio

Além das ferramentas gráficas, é possível fazer com que o Gmsh carregue os dados de um arquivo de extensão `.geo` para gerar as fronteiras de um domínio finito, bem como o plano que o define, bastando apenas gerar a malha sobre o domínio.

Esse arquivo deve informar os dados de construção segundo comandos pré-definidos pelo software. Figura 3.7 ilustra um trecho de um arquivo *.geo*.

É possível identificar alguns dos comandos utilizados. Seguindo a mesma ordem da construção gráfica manual, é preciso definir os pontos sobre a fronteira do domínio que serão usados como referência. O comando para gerar um ponto é

$$Point(n_p) = \{x, y, z, lc\}$$

Figura 3.7 – Trecho do arquivo *.geo* que gera o domínio da

Figura 3.5

```

cl = 0.1;
Point(1) = {1, 1, 0, cl};
Point(2) = {0, 1, 0, cl};
Point(3) = {0, 0, 0, cl};
Point(4) = {1, 0, 0, cl};
Line(1) = {3, 4};
Line(2) = {4, 1};
Line(3) = {1, 2};
Line(4) = {2, 3};
Line Loop(6) = {3, 4, 1, 2};
Plane Surface(6) = {6};

```

Fonte: Produzida pela autora.

em que n_p é o índice do ponto, x , y e z são as coordenadas cartesianas do ponto no espaço, e lc é o parâmetro que define o tamanho das faces da malha próximas ao ponto. Esse parâmetro pode variar de ponto a ponto, dependendo do grau de refinamento da malha que o problema exige para cada região do domínio. Quanto menor o valor de lc , maior o refinamento da malha.

Depois de estabelecidos todos os pontos, estes precisam ser conectados. Existem várias possibilidades de se unir os pontos do domínio no Gmsh, usando segmentos de retas, de arcos, splines, etc. Para cada um dos casos, há um número específico de pontos necessários para a construção das formas. No caso das retas, usa-se o comando

$$Line(n_l) = \{ponto_{inicial}, ponto_{final}\}$$

em que n_l é o índice da reta, $ponto_{inicial}$ é o ponto onde começa a construção da reta, e $ponto_{final}$ é o ponto onde termina o segmento de reta. Para a construção de um segmento de arco, usa-se o comando

$$Circle(n_c) = \{ponto_{inicial}, ponto_{centro}, ponto_{final}\}$$

em que, seguindo a mesma lógica, n_c é o índice do arco e $ponto_{centro}$ é o centro da circunferência que gera esse arco. Nesse caso, mesmo que o ponto do centro não componha o domínio, ele deve ser definido na lista de pontos para a construção do arco. Neste trabalho, os domínios foram definidos apenas por retas.

Após a construção de todo o contorno, é preciso concatenar todas as linhas e uni-las em uma superfície. Para isso, pode-se usar o comando

$$Line Loop(n_{ll}) = \{linha_1, linha_2, \dots, linha_n\}$$

em seguida, o comando

$$Plane Surface(n_{ps}) = \{n_{ll}\}$$

que cria a superfície n_{ps} com base no laço fechado de pontos n_{ll} , ou ainda pode-se usar diretamente o comando

$$Plane Surface(n_{ps}) = \{linha_1, linha_2, \dots, linha_n\}$$

Em ambos os casos, o sinal negativo antes de um índice de linha inverte a ordem de ligação entre os pontos desse elemento.

Um recurso que pode facilitar as construções dos arquivos é o comando

$$Define Constant(nome_{constante}) = \{valor_{constante}\}$$

por exemplo para estabelecer um padrão para a variável lc quanto ao tamanho da malha que será gerada.

Por fim, outro recurso a ser explorado é a associação de identificadores aos elementos da malha, as *physical entities*, que permitem uma segregação dos dados segundo uma propriedade física que se queira associar a uma região da malha. Por exemplo, para definir condições de contorno distintas de um conjunto de pontos da fronteira, aplica-se o comando

$$Physical\ Point(n_{pp}) = \{ponto_1, ponto_2, \dots\}$$

ou

$$Physical\ Line(n_{pl}) = \{linha_1, linha_2, \dots\}$$

Nos dados da malha, pontos/linhas identificados apresentarão o índice correspondente à condição de contorno que foram associados.

O manual do software (Geuzaine e Remacle) possui um material muito completo, com explicação detalhada sobre todos os comandos disponíveis.

3.2.3 Extraindo dados do Gmsh

Seja manualmente ou via arquivo *.geo*, após a construção do domínio e da malha não-estruturada, os dados contidos naquela representação gráfica serão utilizados na resolução de problemas dentro de um código em linguagem C. Logo, a melhor forma de se extrair as informações é através de um arquivo padrão, onde os dados possam ser armazenados e extraídos por comandos de leitura.

O software Gmsh permite o armazenamento de dados de diversas formas, em diferentes extensões de arquivo. No presente trabalho, foi escolhido o formato *INRIA Medit*, de extensão *.mesh*, por ser um formato adequado ao que era necessário na implementação do trabalho. A Figura 3.8 é um print reduzido do arquivo *.mesh* gerado pelo mesmo arquivo *.geo* apresentado anteriormente.

Figura 3.8 – Trecho do arquivo .mesh gerado pela malha da Figura 3.5.

```

MeshVersionFormatted 2
Dimension
3
Vertices
142
      1          1          0          1
      0          1          0          2
      0          0          0          3
      1          0          0          4
0.0999999999999815      0          0          1
0.199999999999956      0          0          1
0.299999999999927      0          0          1
0.399999999999897      0          0          1
0.499999999999869      0          0          1
0.599999999999895      0          0          1
0.699999999999921      0          0          1
0.799999999999947      0          0          1
0.899999999999974      0          0          1
|
Edges
40
3 5 0
5 6 0
6 7 0
7 8 0
8 9 0
9 10 0
(...)
Triangles
242
109 142 69 0
4 14 92 0
3 5 93 0
1 23 90 0
2 32 91 0
4 92 13 0
3 93 40 0

```

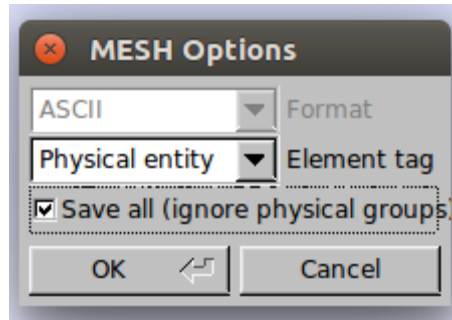
Fonte: Produzida pela autora.

Nesse formato, são apresentados os seguintes conjuntos de dados, nesta ordem:

- Número de vértices;
- Coordenadas cartesianas dos vértices + índice indicando se é vértice de fronteira ou não;
- Número de faces sobre a fronteira (*edges*);
- Índice dos vértices dos extremos das faces + índice informando a *physical entity* da face;
- Número de triângulos;
- Índice dos vértices do triângulo + índice informando a *physical entity* do triângulo;

Ao salvar as informações nesse formato, uma janela é aberta, questionando se deseja salvar as *physical entities* no arquivo. Para isso, basta selecionar a caixa Physical Entity e a opção Save all. A Figura 3.9 mostra essa janela.

Figura 3.9 – Janela de opções de para salvar parâmetros físicos



Fonte: Produzida pela autora.

3.3 Estrutura de dados

Por mais simplificadas que pareçam as informações apresentadas pelo Gmsh no arquivo *.mesh*, elas bastam para a construção de toda a estrutura de dados necessária à resolução de equações sobre a malha. O primeiro passo, é extrair e arquivar corretamente os dados da malha. Nesse caso, foi utilizado um código em linguagem C que, através de funções de leitura de caracteres, armazena as coordenadas e índices dos pontos da malha, bem como parâmetros (índices identificadores) que informam se o ponto pertence ao interior ou à fronteira da malha e a qual triângulo desta. Assim é possível reconhecer, em cada célula da malha, quais são as células vizinhas, as vizinhas das vizinhas, saber quando há faces da célula na fronteira e sobre qual condição de fronteira dentro do problema.

Nessa parte do trabalho, vale ressaltar a importância da construção otimizada da rotina de armazenamento de dados e criação da estrutura de dados. Ao longo do desenvolvimento do projeto, um dos maiores desafios foi adequar os códigos, relativos à estrutura de dados, à capacidade computacional dos computadores disponíveis. Isto porque, quando se trabalha com métodos de alta ordem e/ou malhas com refinamento elevado, sobrecarregar a memória da máquina torna-se tarefa fácil. O trabalho de Santana (2010) foi usado como base para a criação do código da estrutura de dados de forma otimizada.

Após a construção da estrutura de dados, parte-se para a reconstrução da função do problema. A seção 4 aborda essa etapa do trabalho.

4 RECONSTRUÇÃO DE ALTA ORDEM

4.1 Introdução

Após obter uma malha não estruturada e extrair dela seus dados, esses dados serão usados, juntamente com as condições iniciais e de fronteira do problema para resolver uma equação sobre o domínio. Para isso, é necessário um método de reconstrução, capaz de descrever a função incógnita em cada ponto do domínio. Neste trabalho abordaremos o método de reconstrução de alta ordem proposto por Gooch, em seu trabalho (Ollivier-Gooch e Van Altena, 2002).

4.2 Método de reconstrução de alta ordem via mínimos quadrados

O método de Gooch consiste em representar a função incógnita ϕ sobre o domínio através de sua expansão em Série de Taylor. Para um volume de controle de índice i da malha, sobre um domínio bidimensional, tem-se que a reconstrução, a partir da expansão em Série de Taylor da função ϕ_i , em torno de (x_i, y_i) , é dada por:

$$\begin{aligned} \phi_i^R(x, y) = & \phi|_i + \frac{\partial \phi}{\partial x}\bigg|_i (x - x_i) + \frac{\partial \phi}{\partial y}\bigg|_i (y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2}\bigg|_i (x - x_i)^2 \\ & + \frac{\partial^2 \phi}{\partial x \partial y}\bigg|_i (x - x_i)(y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2}\bigg|_i (y - y_i)^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3}\bigg|_i (x - x_i)^3 + \dots \end{aligned} \quad 4.1$$

Nesse caso, a reconstrução da função ϕ é feita ponto a ponto através de um polinômio aproximador, cujos coeficientes são definidos pelas derivadas parciais da série de Taylor. O número de termos da expansão usados na construção do polinômio definirá a ordem de reconstrução. Por exemplo, utilizando os termos com derivadas de ordem 1 (ϕ , ϕ_x e ϕ_y), a aproximação obtida é de segunda ordem. Se utilizamos a expansão até todos os termos de derivadas de ordem 2 (ϕ , ϕ_x , ϕ_y , ϕ_{xx} , ϕ_{xy} e ϕ_{yy}), a aproximação será dita de terceira ordem, e assim sucessivamente. De forma genérica, para uma aproximação de ordem k , são necessários k^2 termos ou $\frac{k(k+1)}{2}$ termos.

² Lê-se “k termial”, termo que designa a seguinte correspondência, $k^2 = k + (k - 1) + (k - 2) + \dots + 1$

Dada a forma do polinômio aproximador, que é a série de Taylor, os seus coeficientes, as derivadas parciais, devem ser determinados de forma que o valor médio da função $\bar{\phi}_i$, para um volume de controle i , seja conservado. O valor médio da função ϕ em um VC_i é dado por:

$$\bar{\phi}_i = \frac{1}{A_i} \iint_{A_i} \phi_i^R dA \quad 4.2$$

Substituindo a equação 4.1 na equação 4.2, temos

$$\begin{aligned} \bar{\phi}_i = \frac{1}{A_i} \iint_{A_i} & \left[\phi|_i + \frac{\partial \phi}{\partial x} \Big|_i (x - x_i) + \frac{\partial \phi}{\partial y} \Big|_i (y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_i (x - x_i)^2 \right. \\ & \left. + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i (x - x_i)(y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_i (y - y_i)^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_i (x - x_i)^3 + \dots \right] dA \end{aligned} \quad 4.3$$

Fazendo $\bar{k}_i = (k - k_i)$, podemos escrever a equação 4.3 como

$$\bar{\phi}_i = \phi|_i + \frac{\partial \phi}{\partial x} \Big|_i \bar{x}_i + \frac{\partial \phi}{\partial y} \Big|_i \bar{y}_i + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_i \bar{x}_i^2 + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i \bar{x}_i \bar{y}_i + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_i \bar{y}_i^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_i \bar{x}_i^3 + \dots \quad 4.4$$

em que

$$\bar{x}^m \bar{y}^n = \frac{1}{A_i} \iint_{A_i} (x - x_i)^m (y - y_i)^n dA \quad 4.5$$

Esse termo é chamado momento, e pode ser calculado por uma regra de quadratura gaussiana. O cálculo da média da função no VC_i porém ainda não é suficiente para fornecer os valores das derivadas parciais, é preciso determinar mais restrições ao problema. Uma delas é que, assim como a média no volume de controle i deve ser conservada, a média dos volumes vizinhos também o deve. Assim,

$$\bar{\phi}_j = \frac{1}{A_j} \iint_{A_j} \phi_i^R dA \quad 4.6$$

com $j = 1, 2, 3, \dots, N$, em que N é o número de VC vizinhos ao volume i , segundo a ordem de reconstrução desejada. Substituindo a equação 4.1 na equação 4.6, tem-se

$$\begin{aligned} \bar{\phi}_j = \frac{1}{A_j} \iint_{A_j} & \left[\phi|_i + \frac{\partial \phi}{\partial x} \Big|_i (x - x_i) + \frac{\partial \phi}{\partial y} \Big|_i (y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_i (x - x_i)^2 \right. \\ & \left. + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i (x - x_i)(y - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_i (y - y_i)^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} \Big|_i (x - x_i)^3 + \dots \right] dA \end{aligned} \quad 4.7$$

Manipulando a equação através da seguinte equivalência de termos:

$$(x - x_i) = (x - x_j) + (x_j - x_i) \quad e \quad (y - y_i) = (y - y_j) + (y_j - y_i) \quad 4.8$$

temos, da integração da equação 4.7

$$\begin{aligned} \bar{\phi}_j = \phi|_i + \frac{\partial \phi}{\partial x} \Big|_i [\bar{x}_j + (x_j - x_i)] + \frac{\partial \phi}{\partial y} \Big|_i [\bar{y}_j + (y_j - y_i)] \\ + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_i [\bar{x}_j^2 + 2\bar{x}_j(x_j - x_i) + (x_j - x_i)^2] \\ + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i [\bar{x}_j \bar{y}_j + \bar{x}_j(y_j - y_i) + \bar{y}_j(x_j - x_i) + (x_j - x_i)(y_j - y_i)] \\ + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_i [\bar{y}_j^2 + 2\bar{y}_j(y_j - y_i) + (y_j - y_i)^2] + \dots \end{aligned} \quad 4.9$$

Essa mesma equação pode ser escrita como

$$\bar{\phi}_j = \phi|_i + \frac{\partial \phi}{\partial x} \Big|_i \widehat{x}_{ij} + \frac{\partial \phi}{\partial y} \Big|_i \widehat{y}_{ij} + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_i \widehat{x}_{ij}^2 + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i \widehat{x}_{ij} \widehat{y}_{ij} + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_i \widehat{y}_{ij}^2 + \dots \quad 4.10$$

sendo

$$\widehat{x^m y^n}_{ij} = \frac{1}{A_j} \iint_{A_j} [(x - x_j) + (x_j - x_i)]^m [(y - y_j) + (y_j - y_i)]^n dA \quad 4.11$$

que é o mesmo que

$$\widehat{x^m y^n}_{ij} = \sum_{l=0}^n \sum_{k=0}^m \binom{n}{l} \binom{m}{k} (x_j - x_i)^k (y_j - y_i)^l \widehat{x^{m-k} y^{n-l}}_{ij} \quad 4.12$$

Os termos $\widehat{x^m y^n}_{ij}$ são chamados termos geométricos.

Com as restrições então estabelecidas, pelas equações 4.4 e 4.10, monta-se um sistema linear sobredeterminado que representa o problema no interior do domínio:

$$\begin{bmatrix} 1 & \bar{x}_l & \bar{y}_l & \bar{x}_l^2 & \bar{x}_l \bar{y}_l & \bar{y}_l^2 & \dots \\ w_{i1} & w_{i1} \widehat{x}_{i1} & w_{i1} \widehat{y}_{i1} & w_{i1} \widehat{x}_{i1}^2 & w_{i1} \widehat{x}_{i1} \widehat{y}_{i1} & w_{i1} \widehat{y}_{i1}^2 & \dots \\ w_{i2} & w_{i2} \widehat{x}_{i2} & w_{i2} \widehat{y}_{i2} & w_{i2} \widehat{x}_{i2}^2 & w_{i2} \widehat{x}_{i2} \widehat{y}_{i2} & w_{i2} \widehat{y}_{i2}^2 & \dots \\ w_{i3} & w_{i3} \widehat{x}_{i3} & w_{i3} \widehat{y}_{i3} & w_{i3} \widehat{x}_{i3}^2 & w_{i3} \widehat{x}_{i3} \widehat{y}_{i3} & w_{i3} \widehat{y}_{i3}^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN} \widehat{x}_{iN} & w_{iN} \widehat{y}_{iN} & w_{iN} \widehat{x}_{iN}^2 & w_{iN} \widehat{x}_{iN} \widehat{y}_{iN} & w_{iN} \widehat{y}_{iN}^2 & \dots \end{bmatrix} \begin{bmatrix} \phi \\ \phi_x \\ \phi_y \\ \frac{1}{2} \phi_{xx} \\ \phi_{xy} \\ \frac{1}{2} \phi_{yy} \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{\phi}_l \\ w_{i1} \bar{\phi}_1 \\ w_{i2} \bar{\phi}_2 \\ w_{i3} \bar{\phi}_3 \\ \vdots \\ w_{iN} \bar{\phi}_N \end{bmatrix} \quad 4.13$$

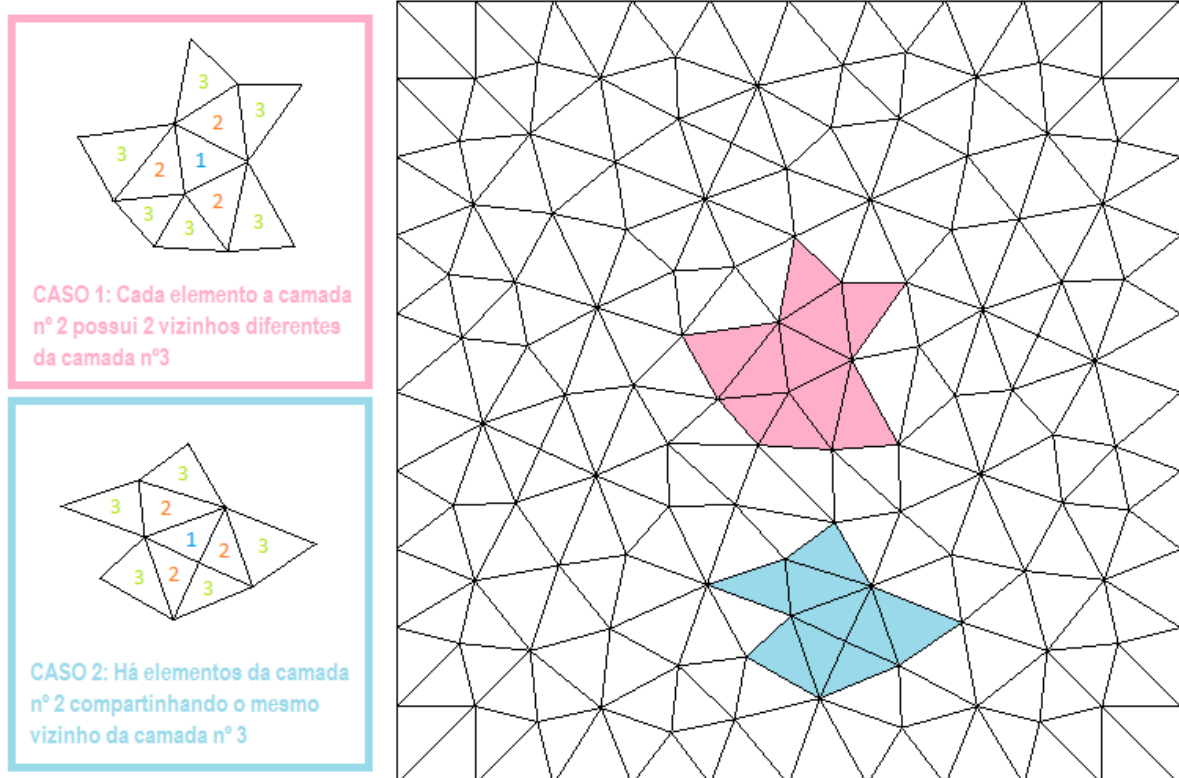
sendo os termos w_{ij} pesos inseridos no contexto para que a distância de um volume controle vizinho do volume principal seja considerada nos cálculos. Assim volumes de controle vizinhos mais próximos do volume de reconstrução terão maior influência sobre o resultado do que volumes mais distantes. Barth (1990) introduz esses pesos na resolução de problemas de dinâmica de fluidos para corrigir a contribuição de cada volume de controle vizinho. Esses pesos são dados por

$$w_{ij} = \frac{1}{|\vec{x}_j - \vec{x}_i|^2} \quad 4.14$$

A resolução desse sistema consiste em determinar as incógnitas do vetor do lado direito, de forma que o produto de matrizes ofereça uma solução mais próxima possível dos valores do lado esquerdo.

Conhecidos os pesos para os volumes vizinhos, a quantidade destes que será utilizada na resolução dependerá da ordem de reconstrução e da configuração da malha não-estruturada do problema. Por exemplo, em uma malha de triângulos, considerando o interior do domínio, à primeira ordem de reconstrução, serão sempre necessários três vizinhos. Já a segunda de camada de vizinhos possui um número máximo, dado pela soma de dois triângulos adjacentes a cada um desses vizinhos da primeira camada. Contudo, há a possibilidade de dois triângulos da primeira camada compartilharem o mesmo vizinho, o que reduz o número de volumes na reconstrução de segunda ordem. A Figura 4.1 ilustra essa distinção.

Figura 4.1 – Composição de estênceis da malha da Figura 3.5 em casos com e sem compartilhamento de vizinhos em uma camada



Fonte: Produzida pela autora.

Por isso é importante uma estrutura de dados bem articulada, que seja capaz de identificar os volumes de controle em cada nível da camada vizinha ao volume de reconstrução.

Novamente, no conceito do método de reconstrução via mínimos quadrados, busca-se reconstruir a solução em cada volume de controle de forma a aproximar o

lado esquerdo ao lado direito na equação 4.13. Para resolver o sistema nos pontos do interior do domínio, na linha de pesquisa de Gooch, afim de melhorar os resultados das aproximações, utiliza-se o método da eliminação de Gauss com pivoteamento a partir da segunda coluna. Então aplicam-se transformações de Householder, transformações ortogonais que resultam em uma matriz triangular superior no lado esquerdo da equação 4.13. Essas transformações tornam estáveis as equações a serem resolvidas, e nesse caso, resolvidas a partir do método da substituição regressiva. A aplicação sequencial desses métodos fornece a aproximação do problema para os pontos do interior do domínio. No código em linguagem C construído, foi utilizada a biblioteca GSL (Free Software Foundation, Inc., 2009), que conta com algumas sub-rotinas prontas para facilitar o uso da transformação de Householder.

Para os pontos da fronteira, a próxima seção trata suas especificidades.

4.3 Reconstrução na fronteira

Quando se parte para os volumes de controle que integram a fronteira do domínio, algumas abordagens devem ser feitas para que seja possível reconstruir a solução. O primeiro ponto é que o número de vizinhos se torna restrito, já que ao menos uma das faces do VC estará integrando a fronteira. Dessa forma, os únicos pontos a serem avaliados na reconstrução, pela face que integra a fronteira, são aqueles contidos na própria fronteira, sendo a seleção destes não aleatória, mas sim seguindo parâmetros específicos. O trabalho de Saito (2008) trata bem essa questão. Ela estuda a obra de Van Altena e resume suas conclusões, que levam à preferência pela estratégia da restrição de contorno de Gauss ao invés da de Taylor. Neste trabalho, abordamos a restrição e Gauss, que pode ser aplicada a fronteiras retas ou curvas, determinadas por condições de Dirichlet, Neumann ou Robin. Para isso, basta conhecer a posição dos pontos de Gauss no contorno e suas respectivas normais. O número de pontos de Gauss utilizados na fronteira é o que determinará a ordem de precisão do método.

4.3.1 Condição de contorno de Dirichlet

Na condição de fronteira de Dirichlet, é conhecido o próprio valor da função, isto é, $\phi(x, y) = f_1(x, y)$, e conhece-se o valor de $f_1(x, y)$, no caso, para todos os

pontos de Gauss (x_g, y_g) da fronteira em que tal condição se aplica. Inserindo essa condição à equação 4.1,

$$f_1(x_g, y_g) = \phi|_i + \frac{\partial \phi}{\partial x}\bigg|_i (x_g - x_i) + \frac{\partial \phi}{\partial y}\bigg|_i (y_g - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2}\bigg|_i (x_g - x_i)^2 + \frac{\partial^2 \phi}{\partial x \partial y}\bigg|_i (x_g - x_i)(y_g - y_i) + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2}\bigg|_i (y_g - y_i)^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3}\bigg|_i (x_g - x_i)^3 + \dots \quad 4.15$$

Assim, uma nova restrição surge no sistema linear que é montado para a reconstrução da solução em um VC que possui ao menos uma face sobre a fronteira do domínio.

$$\begin{bmatrix} 1 & \bar{x}_i & \bar{y}_i & \bar{x}_i^2 & \bar{x}_i \bar{y}_i & \bar{y}_i^2 & \dots \\ 1 & \Delta x_1 & \Delta y_1 & \Delta x_1^2 & \Delta x_1 \Delta y_1 & \Delta y_1^2 & \dots \\ 1 & \Delta x_2 & \Delta y_2 & \Delta x_2^2 & \Delta x_2 \Delta y_2 & \Delta y_2^2 & \dots \\ \hline w_{i1} & w_{i1} \widehat{x}_{i1} & w_{i1} \widehat{y}_{i1} & w_{i1} \widehat{x}_{i1}^2 & w_{i1} \widehat{x}_{i1} \widehat{y}_{i1} & w_{i1} \widehat{y}_{i1}^2 & \dots \\ w_{i2} & w_{i2} \widehat{x}_{i2} & w_{i2} \widehat{y}_{i2} & w_{i2} \widehat{x}_{i2}^2 & w_{i2} \widehat{x}_{i2} \widehat{y}_{i2} & w_{i2} \widehat{y}_{i2}^2 & \dots \\ w_{i3} & w_{i3} \widehat{x}_{i3} & w_{i3} \widehat{y}_{i3} & w_{i3} \widehat{x}_{i3}^2 & w_{i3} \widehat{x}_{i3} \widehat{y}_{i3} & w_{i3} \widehat{y}_{i3}^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN} \widehat{x}_{iN} & w_{iN} \widehat{y}_{iN} & w_{iN} \widehat{x}_{iN}^2 & w_{iN} \widehat{x}_{iN} \widehat{y}_{iN} & w_{iN} \widehat{y}_{iN}^2 & \dots \end{bmatrix} \begin{bmatrix} \phi \\ \phi_x \\ \phi_y \\ \frac{1}{2} \phi_{xx} \\ \phi_{xy} \\ \frac{1}{2} \phi_{yy} \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{\phi}_i \\ f_1(x_1, y_1) \\ f_1(x_2, y_2) \\ \overline{w_{i1} \bar{\phi}_1} \\ \overline{w_{i2} \bar{\phi}_2} \\ \overline{w_{i3} \bar{\phi}_3} \\ \vdots \\ \overline{w_{iN} \bar{\phi}_N} \end{bmatrix} \quad 4.16$$

em que $\Delta x_g = (x_g - x_i)$, $\Delta y_g = (y_g - y_i)$ e g varia de 1 ao número de pontos de Gauss calculados sobre a fronteira.

4.3.2 Condição de contorno de Neumann

Na condição de fronteira de Neumann, é conhecida derivada do valor da função, isto é, $\frac{\partial \phi}{\partial n} = f_2(x, y)$, e conhece-se o valor de $f_2(x, y)$, no caso, para todos os pontos de Gauss (x_g, y_g) da fronteira em que tal condição se aplica. Em se tratando de uma derivada direcional, pode-se escrever:

$$f_2(x, y) = \frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} \vec{n}_x + \frac{\partial \phi}{\partial y} \vec{n}_y = \nabla \phi \cdot \vec{\eta} \quad 4.17$$

Novamente, inserindo essa condição na equação 4.1,

$$f_2(x_g, y_g) = n_x \left[\frac{\partial \phi}{\partial x} \Big|_i (x_g - x_i) + \frac{\partial^2 \phi}{\partial x^2} \Big|_i (x_g - x_i) + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i (y_g - y_i) + \dots \right] + n_y \left[\frac{\partial \phi}{\partial y} \Big|_i (y_g - y_i) + \frac{\partial^2 \phi}{\partial y^2} \Big|_i (y_g - y_i) + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_i (x_g - x_i) + \dots \right] \quad 4.18$$

As novas restrições que surgem no sistema linear para o VC que possui ao menos uma face sobre essa fronteira do domínio são:

$$\begin{bmatrix} 1 & \bar{x}_i & \bar{y}_i & \bar{x}_i^2 & \bar{x}_i \bar{y}_i & \bar{y}_i^2 & \dots \\ 0 & n_{x_1} & n_{y_1} & 2n_{x_1} \Delta x_1 & n_{y_1} \Delta x_1 + n_{x_1} \Delta y_1 & 2n_{y_1} \Delta y_1 & \dots \\ 0 & n_{x_2} & n_{y_2} & 2n_{x_2} \Delta x_2 & n_{y_2} \Delta x_2 + n_{x_2} \Delta y_2 & 2n_{y_2} \Delta y_2 & \dots \\ \hline w_{i1} & w_{i1} \widehat{x}_{i1} & w_{i1} \widehat{y}_{i1} & w_{i1} \widehat{x}_{i1}^2 & w_{i1} \widehat{x}_{i1} \widehat{y}_{i1} & w_{i1} \widehat{y}_{i1}^2 & \dots \\ w_{i2} & w_{i2} \widehat{x}_{i2} & w_{i2} \widehat{y}_{i2} & w_{i2} \widehat{x}_{i2}^2 & w_{i2} \widehat{x}_{i2} \widehat{y}_{i2} & w_{i2} \widehat{y}_{i2}^2 & \dots \\ w_{i3} & w_{i3} \widehat{x}_{i3} & w_{i3} \widehat{y}_{i3} & w_{i3} \widehat{x}_{i3}^2 & w_{i3} \widehat{x}_{i3} \widehat{y}_{i3} & w_{i3} \widehat{y}_{i3}^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN} \widehat{x}_{iN} & w_{iN} \widehat{y}_{iN} & w_{iN} \widehat{x}_{iN}^2 & w_{iN} \widehat{x}_{iN} \widehat{y}_{iN} & w_{iN} \widehat{y}_{iN}^2 & \dots \end{bmatrix} \begin{bmatrix} \phi \\ \phi_x \\ \phi_y \\ \frac{1}{2} \phi_{xx} \\ \phi_{xy} \\ \frac{1}{2} \phi_{yy} \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{\phi}_i \\ f_2(x_1, y_1) \\ f_2(x_2, y_2) \\ \hline w_{i1} \bar{\phi}_1 \\ w_{i2} \bar{\phi}_2 \\ w_{i3} \bar{\phi}_3 \\ \vdots \\ w_{iN} \bar{\phi}_N \end{bmatrix} \quad 4.19$$

novamente, em que $\Delta x_g = (x_g - x_i)$, $\Delta y_g = (y_g - y_i)$ e g varia de 1 ao número de pontos de Gauss calculados sobre a fronteira.

4.3.3 Condição de contorno de Robin

A condição de fronteira de Robin, também chamada mista, consiste em uma combinação linear das condições de Dirichlet e de Neumann, ou seja:

$$f_3(x, y) = a\phi(x, y) + b(\nabla\phi \cdot \vec{\eta}) \quad 4.20$$

Aplicando a combinação linear, de acordo com os coeficientes a e b , para os termos dos pontos da fronteira (lado esquerdo do sistema), e os resultados conhecidos de f_3 (lado direito do sistema), constrói-se o sistema linear para os pontos regidos por essa condição de fronteira, similar às condições de Dirichlet e Neumann.

4.3.4 Resolução do sistema

Para resolver o sistema linear da fronteira, agora com algumas linhas adicionais de restrição, cuja forma é definida por uma das três condições de fronteira citadas anteriormente, adotam-se alguns procedimentos antes da aplicação do método da eliminação de Gauss em si, que melhoram os resultados da solução do sistema. Primeiro, para as linhas de restrição, determina-se o maior coeficiente da linha, em módulo, e então, dividem-se todos os elementos da linha por esse coeficiente, em módulo. Em seguida, trocam-se as colunas do maior coeficiente (sempre em módulo) e do pivô. Nesse ponto, é importante lembrar de trocar também as linhas do vetor solução correspondentes às colunas trocadas. Na sequência, aplica-se a eliminação gaussiana para as linhas abaixo do pivô. Repete-se esse procedimento para todas as linhas de restrição, apenas. Segue-se então a aplicação das transformações de Householder até transformar a matriz dos coeficientes em uma matriz triangular verdadeira, a matriz dos coeficientes abaixo da diagonal principal passa a ter a aparência de um trapézio. O sistema então pode ser resolvido por substituição regressiva.

No cálculo das integrais e linha, para aproximar numericamente uma integral de uma função de uma única variável $f(x)$ em um intervalo $[a, b]$, em essência, aproxima-se $f(x)$ pelo polinômio interpolador $p_n(x)$ de grau n no intervalo considerado, e daí integra-se $p_n(x)$, tomando a integral desse polinômio como sendo uma aproximação para integral de $f(x)$ no intervalo $[a, b]$. Em ideias gerais,

$$\int_a^b f(x) dx \cong \int_a^b \underbrace{\sum_{k=0}^n f(x_k) L_k(x)}_{p_n(x)} dx = \sum_{k=0}^n A_k f(x_k) dx \quad 4.21$$

em que

$$A_k = \int_a^b L_k(x) dx \quad L_k = \frac{\prod_{i=0, i \neq k}^n (x - x_i)}{\prod_{i=0, i \neq k}^n (x_k - x_i)} \quad 4.22$$

O grau de precisão depende do grau do polinômio interpolador, e portanto do número de pontos utilizados para obter o polinômio interpolador. Nas regras de quadratura gaussiana, os pontos utilizados na interpolação são as raízes dos

polinômios ortogonais. No caso dos polinômios de Gauss-Legendre, que são polinômios ortogonais, suas raízes se encontram no intervalo $[-1,1]$. Assim, a função a ser integrada deve estar no intervalo $[-1,1]$, e caso não esteja, deve-se fazer uma mudança de variáveis, como segue.

$$\int_a^b f(x)dx \cong \frac{b-a}{2} \int_{-1}^1 f(x(t)) dt \quad x(t) = \frac{(b-a)t}{2} + \frac{a+b}{2} \quad 4.23$$

Nessa transformação do intervalo $[a, b]$ para o intervalo $[-1,1]$, a integral no intervalo $[-1,1]$ fica multiplicada pela razão $\frac{b-a}{2}$, que é uma razão entre o comprimento do intervalo $[a, b]$ e o comprimento do intervalo $[-1, 1]$. Essa mesma ideia deve ser seguida na hora de calcular uma integral de contorno em um triângulo. Considerando o lado \overline{AB} um triângulo ΔABC , coordenadas $A(x_A, y_A)$ e $B(x_B, y_B)$. Qualquer ponto da reta \overleftrightarrow{AB} que esteja entre A e B pode ser definido parametricamente por

$$r(t) = (x_A, y_A) + 0,5(x_B - x_A, y_B - y_A)(t + 1) \quad t \in [-1,1] \quad 4.24$$

fazendo t variar de -1 a 1 . Isso significa que as coordenadas $x(t)$ e $y(t)$ são dadas por

$$x(t) = x_A + 0,5(x_B - x_A)(t + 1) \quad y(t) = y_A + 0,5(y_B - y_A)(t + 1) \quad 4.25$$

Essas últimas expressões podem ser utilizadas para obter os pontos de Gauss no segmento \overline{AB} a partir dos pontos de Gauss já conhecidos nas tabelas apresentadas em livros.

Para calcular a integral de contorno

$$\oint_{\partial\Delta ABC} \nabla u \cdot n \, ds \quad 4.26$$

basta utilizar a fórmula

$$\oint_{\partial\Delta ABC} \nabla u \cdot n \, ds \cong \sum_{i=1}^3 \sum_{k=1}^{NPG} \omega_i^{(k)} [u_x(x_g^{(k)}, y_g^{(k)})n_x^{(k)} + u_y(x_g^{(k)}, y_g^{(k)})n_y^{(k)}] \quad 4.27$$

em que $\omega_i^{(k)} = 0,5d_i A_k$ são os pesos e $(x_g^{(k)}, y_g^{(k)})$ são os pontos de Gauss obtidos na equação 4.25. Nos pesos d_i , $i = 1, 2, 3$ são, respectivamente, os comprimentos dos lados AB , BC e CA do triângulo ΔABC . Nota-se que $0,5d_i$ nada mais é do que a razão do comprimento de um lado do triângulo ΔABC com a medida do intervalo $[-1,1]$, onde se localizam os pontos de Gauss.

Reconstruída a função do problema, parte-se para a resolução numérica do problema. A seção 5 trata dos métodos utilizados e desenvolvimento da equação problema.

5 RESOLUÇÃO NUMÉRICA DA EQUAÇÃO DE ADVECÇÃO-DIFUSÃO

5.1 Introdução

O método de reconstrução para uma função ϕ foi descrito na seção anterior. Definida a metodologia de resolução do problema, o próximo passo é estudar o tipo de equação que se pretende resolver neste trabalho. Trataremos da equação de Advecção-Difusão, presente em muitos problemas de engenharia, especialmente os relacionados a escoamento de fluidos. Vale lembrar que serão tratados casos bidimensionais, variáveis com o tempo. A equação, já apresentada no capítulo 1 (1.1), pode ser escrita como:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{\omega} \phi - \nabla \cdot (\kappa \nabla \phi) = f \quad (x, y) \in \Omega \quad e \quad t > 0 \quad 5.1$$

em que $\vec{\omega} = (u, v)$ é o vetor velocidade de advecção, e κ é o coeficiente de difusão, ambos variáveis, ou não, no domínio Ω . Este problema será tratado como um problema pseudo-transiente, em que os termos advectivo e difusivo e o termo fonte variam apenas no espaço, segundo x e y . Logo $\frac{\partial \phi}{\partial t} = 0$, mas o termo é inserido na equação para auxiliar a obtenção da solução numérica do problema. No desenvolvimento dessa seção, isso será evidenciado.

5.2 Desenvolvimento da equação

Como o MVF é um método embasado em integrais, aplica-se a integração dos termos temporal, advectivo e difusivo sobre a equação 5.1 para cada VC do domínio Ω discretizado,

$$\iint_{\Omega} \frac{\partial \phi}{\partial t} d\Omega + \iint_{\Omega} [\nabla \cdot \vec{\omega} \phi - \nabla \cdot (\kappa \nabla \phi)] d\Omega = \iint_{\Omega} f d\Omega \quad 5.2$$

Aplicando o teorema de Gauss sobre a equação 5.2, tem-se

$$\iint_{\Omega} \frac{\partial \phi}{\partial t} d\Omega + \oint_{\Omega} (\bar{\omega}\phi - \kappa\nabla\phi) \cdot \hat{\eta} ds = \iint_{\Omega} f d\Omega \quad 5.3$$

Consideraremos a aproximação do termo temporal por seu valor médio no domínio Ω , de área dada por A_{Ω} . Em seguida, utilizando diferença progressiva para aproximar esse termo, a equação 5.3 fica

$$\begin{aligned} \frac{\partial \bar{\phi}}{\partial t} A_{\Omega} + \oint_{\Omega} (\bar{\omega}\phi - \kappa\nabla\phi) \cdot \hat{\eta} ds &= \iint_{\Omega} f d\Omega \\ \frac{\bar{\phi}^{n+1} - \bar{\phi}^n}{\Delta t} A_{\Omega} + \oint_{\Omega} (\bar{\omega}\phi - \kappa\nabla\phi) \cdot \hat{\eta} ds &= \iint_{\Omega} f d\Omega \end{aligned} \quad 5.4$$

Nesse trabalho, o método de Runge-Kutta multiestágios (Haelterman, J. e Van Heule, 2009) foi utilizado para calcular a evolução temporal da equação. Nesse caso, acrescentam-se os parâmetros α_k para cada estágio percorrido nas iterações, sendo que $k = 1, 2, 3, \dots, n_p$, onde n_p é o número de estágios utilizado. Isolando $\bar{\phi}$ em $n + 1$ da equação 5.4,

$$\bar{\phi}^{n+1} = \bar{\phi}^n - \frac{\alpha_k}{A_{\Omega}\Delta t} \left[\underbrace{\oint_{\Omega} (\bar{\omega}\phi - \kappa\nabla\phi) \cdot \hat{\eta} ds}_{IF} - \underbrace{\iint_{\Omega} f d\Omega}_{TF} \right] \quad 5.5$$

Aqui é possível distinguir dois termos importantes na resolução da equação, a integral de fluxo IF e o termo fonte TF . Ao aplicar o método de Runge-Kutta multiestágios, uma solução inicial é integrada progressivamente até que se atinja o estado estacionário, já que o problema proposto não varia no tempo. Assim reconstroem-se as soluções com as aproximações nodais de cada volume de controle. Os polinômios obtidos são então utilizados no cálculo das integrais de fluxo iterativamente. Nesse caso, é necessário estabelecer um critério de parada, que defina se o valor da aproximação estará suficientemente próximo da solução exata, considerando que o método converge para essa solução. No caso estacionário, foi estabelecida a parada segundo o valor do erro relativo entre duas iterações

consecutivas. Para o caso transiente, pode-se trabalhar com o número de iterações como limitador, ou mesmo o intervalo de tempo fixado para o problema.

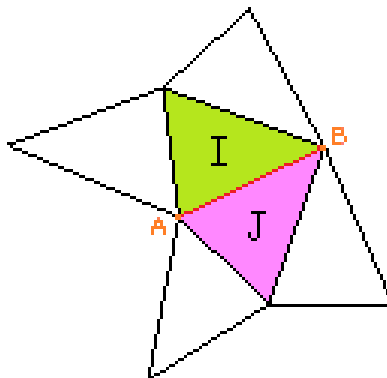
O termo α_k da equação 5.5 define o “peso” de cada estágio no tempo. Literaturas diversas utilizam diferentes valores para esses parâmetros, bem como diferentes números de parâmetros. Neste trabalho, optou-se pelo método de Runge-Kutta multiestágios segundo o trabalho de Gooch (1995), com 5 parâmetros dados por:

$$\alpha_1 = \frac{1}{15}; \alpha_2 = \frac{7}{45}; \alpha_3 = \frac{2}{7}; \alpha_4 = \frac{1}{2} \text{ e } \alpha_5 = 1 \quad 5.6$$

5.3 Cálculo das integrais

Os fluxos advectivo e difusivo da integral de fluxo são dados por $\vec{\omega}\phi$ e $\kappa\nabla\phi$, como evidenciado na equação 5.5. Como o nome diz, trata-se de uma integral de contorno, ou seja, que é calculada sobre as faces dos volumes de controle. Para entender como calculá-los, toma-se a Figura 5.1 para exemplo.

Figura 5.1 – Esquema de volumes de controle vizinhos I e J que dividem a fronteira \overline{AB}



Fonte: Produzida pela autora.

Coloca-se em foco a reconstrução no volume de controle I , VC_I , onde se calcula a integral de fluxo sobre a face AB , que faz fronteira com o VC_J . Nesse caso, para o termo advectivo, a direção do fluxo é dada com base no sinal produto interno $\vec{\omega} \cdot \vec{\eta}_g$, sendo $\vec{\omega}$ o vetor velocidade de advecção, e $\vec{\eta}_g$, o vetor normal à face AB no ponto de Gauss g sobre esta. Se o termo for produto for positivo, toma-se como referência a integral do volume do controle I . Se esse produto for negativo, toma-se

como referência o volume de controle J . Para o termo difusivo, tomam-se as médias, ponderadas pelas áreas de VC_I e VC_J , do gradiente de ϕ .

Expressando matematicamente, tem-se para as integrais de fluxo a seguinte expressão:

$$IF = \sum_{i=0}^{NF} \sum_{k=0}^{NP} [\vec{\omega}_g^j \phi^R(x_g^k, y_g^k) - \kappa(x_g^k, y_g^k) \nabla \phi_m^R(x_g^k, y_g^k)] \cdot \vec{\eta}_g^k \quad 5.7$$

sendo NF o número de faces do VC , NP , o número de pontos de Gauss sobre uma face, e (x_g^k, y_g^k) as coordenadas dos pontos de Gauss sobre as faces dos VC . $\vec{\eta}_g^k = (\eta_x, \eta_y)$ é a componente normal, $\vec{\omega}_g^j = [u(x_g, y_g), v(x_g, y_g)]$ é a velocidade de advecção, e $\nabla \phi_m^R(x_g^k, y_g^k)$ é a média do gradiente, todos sobre o ponto de Gauss. Esse último termo é dado por

$$\nabla \phi_m^R(x_g^k, y_g^k) = \frac{\kappa_I(x_g^k, y_g^k) \nabla_I \phi^R(x_g^k, y_g^k) A_{VC_I} + \kappa_J(x_g^k, y_g^k) \nabla_J \phi^R(x_g^k, y_g^k) A_{VC_J}}{A_{VC_I} + A_{VC_J}} \quad 5.8$$

em que A_{VC_I} e A_{VC_J} são as áreas, respectivamente, dos volumes de controle I e J .

No cálculo do termo fonte, tomou-se o trabalho de Dunavant (1985) para determinar os valores dos pontos e dos pesos de Gauss. O programa desenvolvido permite obter aproximações de ordem 1 a 20 para o termo fonte.

5.4 Termo temporal

Uma observação importante a se fazer é que o passo de tempo para a equação proposta depende de uma constante empírica, CFL , mas também da participação ou não dos termos difusivo e advectivo. Nesse trabalho, foram usados valores entre 1 e 0,5 para o número CFL .

Quando se tratar de uma equação puramente difusiva, ou seja, quando $\vec{\omega} = (0,0)$,

$$(\Delta t_i)_{dif} = CFL \frac{A_{VC_i}}{\kappa_c} \quad 5.9$$

Quando se tratar de uma equação puramente advectiva, ou seja, quando $\kappa = 0$,

$$(\Delta t_i)_{adv} = CFL \frac{A_{VC_i}}{|\vec{\omega}_c|} \quad 5.10$$

Quando se tratar de uma equação advection-difusiva,

$$(\Delta t_i)_{adv/dif} = \min\{(\Delta t_i)_{dif}, (\Delta t_i)_{adv}\} \quad 5.11$$

sendo que A_{VC_i} é a área do i -ésimo VC , e κ_c e $|\vec{\omega}_c|$ são o coeficiente de difusão e a velocidade de advecção, respectivamente, calculados no centroide do VC_i .

O programa em linguagem C para resolver equações de advecção-difusão foi desenvolvido com base no conteúdo dos capítulos apresentados até aqui. Para se ter certeza que os resultados apresentam a ordem de precisão pretendida com o programa, é necessário realizar a verificação computacional do código. A forma como essa verificação é feita será apresentada na próxima seção.

6 VERIFICAÇÃO COMPUTACIONAL

6.1 Introdução

A verificação computacional é etapa obrigatória para trabalhos que envolvam resolução numérica de equações, dado que esta é a única forma de comprovar que os resultados estão sendo obtidos segundo a ordem de precisão, ou seja, com uma taxa de convergência que a teoria método se propõe a entregar.

Para o método de reconstrução de alta ordem, aplicado a malhas não-estruturadas, o processo de verificação computacional é feito da seguinte maneira. Traça-se um gráfico do logaritmo, na base 10, do número de volumes de controle (eixo x) pelo logaritmo, na mesma base, da norma do erro da equação (eixo y). Aplica-se então o método de resolução a uma malha grosseira, com um número x de volumes de controle. Depois para uma malha um pouco mais refinada, com $2x$ volumes de controle. Em seguida, $4x$ volumes de controle, $8x$, $16x$, e assim por diante, até se obter uma malha bastante refinada. Para cada resolução da equação sobre essas malhas, calcula-se a norma, e em um gráfico, plotam-se os logaritmos do número de VC e da norma correspondente, como dito anteriormente. Os pontos obtidos podem ser unidos por uma reta de inclinação negativa, cujo valor dessa inclinação corresponde, aproximadamente, à metade do valor da ordem de precisão do método.

Isso pode ser entendido de forma mais fácil pela seguinte estratégia, considerando uma malha não-estruturada regular, ou seja, cujos volumes de controle possuem aproximadamente a mesma área, tem-se que

$$A_i = \frac{A_T}{N} \quad 6.1$$

em que A_i é a área de um VC, A_T é a área total do domínio e N é o número de volumes do domínio. Portanto A_i é proporcional a $1/N$. Do conceito de escalas de comprimento, ou *length scale*, como é mais abordado, para esse domínio, a ordem de magnitude de seus volumes é dada por

$$\Delta L = \sqrt{A_i} \quad 6.2$$

Então, sendo o erro $O(\Delta L^n)$ para uma ordem de precisão n , a norma do erro pode ser escrita como

$$L_k = \sqrt[k]{\frac{N \frac{1}{N} (N^{-\frac{n}{2}})^k}{N \frac{1}{N}}} \Rightarrow L_k = N^{-\frac{n}{2}} \quad 6.3$$

Aplicando o logaritmo dos dois lados da expressão,

$$\frac{\log(L_k)}{\log(N)} = -\frac{n}{2} \Rightarrow n = -2 \frac{\log(L_k)}{\log(N)} \quad 6.4$$

Para este trabalho, foram usadas as normas L_1 , L_2 e L_∞ , dadas por

$$L_k = \left[\frac{\sum_{i=1}^N A_i E_i}{\sum_{i=1}^N A_i} \right]^{\frac{1}{k}} \quad i = 1; 2 \quad e \quad L_\infty = \max_{1 \leq i \leq N} |E_i| \quad 6.5$$

Nesse caso, N é o número de volumes de controle da malha, A_i é a área do i -ésimo volume de controle, e $E_i = |R_{i_{ext}} - R_{i_{aprox}}|$, onde

$$R = \iint_{\Omega} [\nabla \cdot \vec{\omega} \phi - \nabla \cdot (\kappa \nabla \phi) - f] d\Omega \quad 6.6$$

sendo R o resíduo da função. Para o cálculo do erro, subtrai-se o resíduo, da solução exata da função ϕ , da solução aproximada. Obviamente, este processo precisa ser feito com uma função cuja solução exata seja conhecida.

Alguns dos testes computacionais que foram feitos no trabalho serão apresentados na sequência.

6.2 Testes de verificação

Diversos testes de verificação computacional foram executados afim de se observar a conformidade dos resultados obtidos com a ordem de aproximação proposta. Para praticidade do trabalho, será apresentado um caso específico.

Recapitulando a metodologia do trabalho, para este problema, foi utilizado um domínio quadrado de lado unitário, sobre o qual foi construída a malha computacional não-estruturada. Os dados da malha foram exportados para o programa, devidamente agrupados, e foram montados os sistemas lineares sobredeterminados para a obtenção dos polinômios de reconstrução. Montados os polinômios de reconstrução, foi obtida a solução da função incógnita nos pontos da malha. Nesse teste, buscou-se a reconstrução da função

$$\phi(x, y) = \text{sen}(\pi x)\text{sen}(\pi y) \quad 6.7$$

em um domínio quadrado, como citado, $\Omega = [0,1] \times [0,1]$.

Para a visualização da função nesse problema teste, os gráficos, plotados com o *Gnuplot* (Williams e Kelley, 2018), da solução aproximada, para ordem 5 de reconstrução, em uma malha de 5872 triângulos, são mostrados na Figura 6.1. Observa-se que a reconstrução da função exibe um resultado bastante próximo da solução analítica do problema, quando se constrói o domo com pico centralizado no quadrado, e valor de máximo aproximadamente igual a 1.

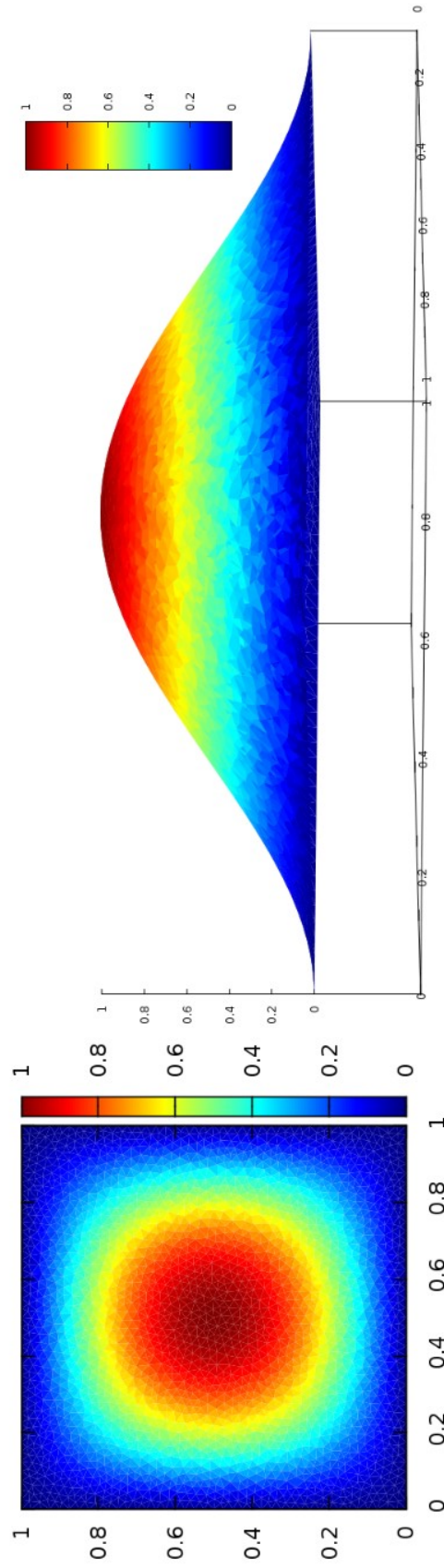
Aplicando as formulações para a obtenção das normas, para as ordens de reconstrução de 1 a 7, têm-se os seguintes resultados:

Tabela 6.1 – Ordens de 1 a 7 obtidas com as três normas do resíduo no problema teste

ORDEM TEÓRICA	Ordem para norma L_1	Ordem para norma L_2	Ordem para norma L_∞
2	2.08554	2.07492	1.77144
3	3.48512	3.19188	2.29150
4	4.61270	4.51450	3.70268
5	5.50730	5.52846	5.04620
6	6.42214	6.08196	5.09804
7	7.26852	7.26056	6.01754

Fonte: Produzida pela autora.

Figura 6.1 – Reconstrução da função 6.7



Fonte: Produzida pela autora.

Nota-se, pela Tabela 6.1, que os resultados da reconstrução produziram valores, para as normas L_1 e L_2 , condizentes com a ordem aplicada ao método. Já a norma L_∞ apresentou resultados mais distantes da ordem real de aproximação. É consenso entre vários autores que esta norma é um parâmetro mais difícil de se adaptar aos problemas. Vários outros testes foram realizados seguindo a mesma linha de raciocínio, para diferentes funções ϕ .

Após verificar e aprovar os resultados de reconstrução apresentados pelo código, foram realizados testes computacionais com a resolução de alguns problemas envolvendo a Equação de Advecção-Difusão, apresentados na seção 7.

7 TESTES DE VERIFICAÇÃO

7.1 Introdução

Após implementado o código em linguagem C capaz de resolver a Equação de Advecção-Difusão em malhas não-estruturadas, em domínios de geometrias complexas, para qualquer ordem de precisão, limitada à capacidade de processamento dos computadores utilizados, é interessante mostrar alguns testes computacionais realizados para variações nos termos difusivos e advectivos da equação, bem como nas condições de contorno, para observar como os resultados se comportam. Alguns dos testes realizados serão aqui demonstrados.

7.2 Equação de Poisson

A Equação de Poisson é uma variante da Equação de Advecção-Difusão, que trata da difusão pura, em que a velocidade de advecção é nula, isto é, $\vec{\omega} = 0$, o que nos dá a equação

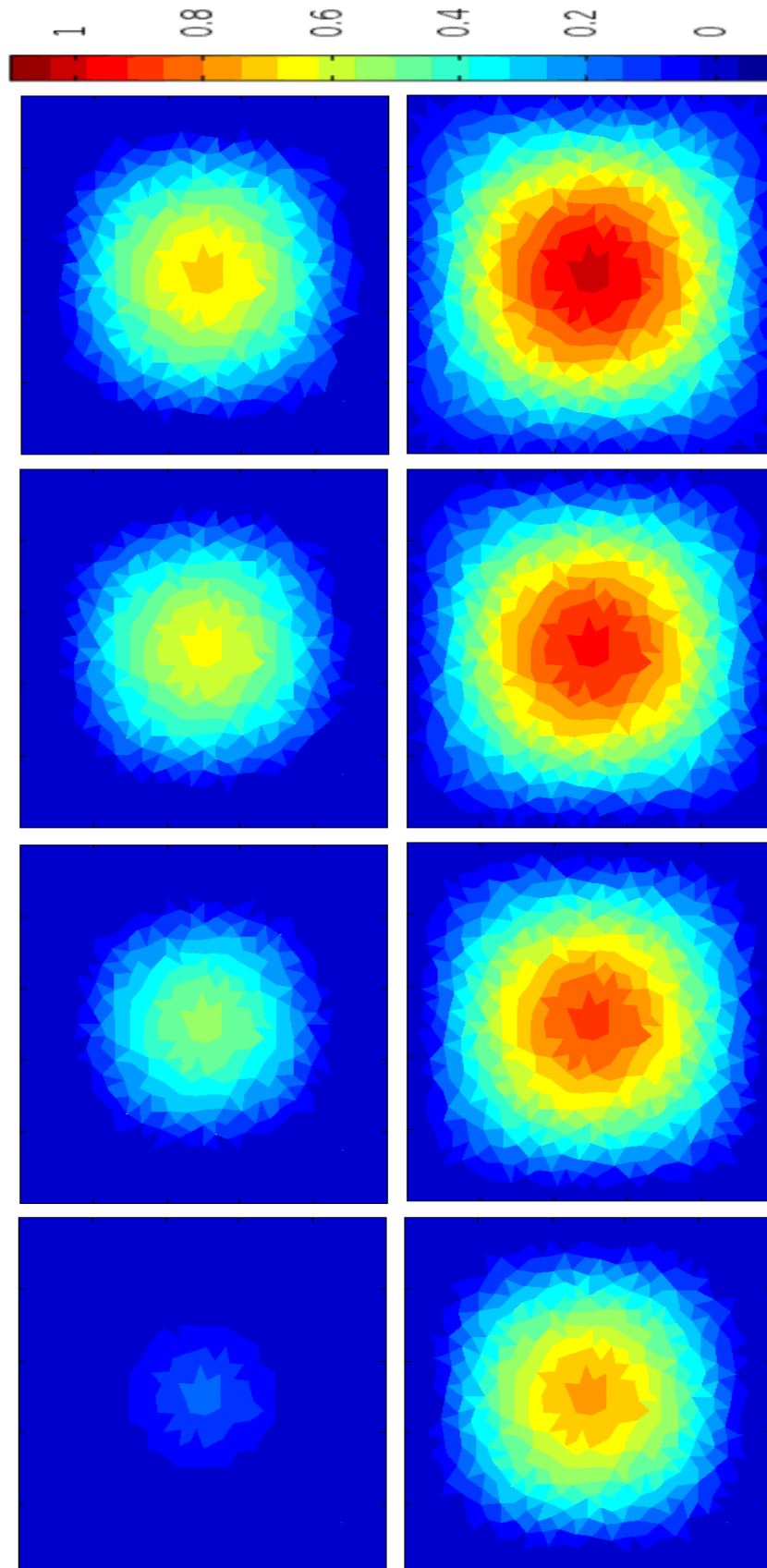
$$-\nabla \cdot (\kappa \nabla \phi) = f \quad 7.1$$

Esse teste foi executado em um domínio quadrado de lado unitário, $\Omega = [0,1] \times [0,1]$, com a função 6.7 seguinte solução analítica do problema, a qual fornece o termo fonte,

$$f = 2\pi^2 \text{sen}(\pi x) \text{sen}(\pi y) \quad 7.2$$

A reconstrução da função para esse problema fornece o gráfico mostrado na Figura 6.1. Nesse caso, parte-se de uma condição inicial nula em todo o domínio, e com a evolução temporal na resolução da equação, o gráfico se aproxima cada vez mais daquela mostrado na Figura 6.1, até que a diferença entre duas iterações consecutivas do problema seja tão pequena quanto se deseja. Nesse caso, diz-se que o problema é estacionário, já que evolui até que se atinja o estado estacionário. Para $\kappa = 1$, com ordem de reconstrução 3 em uma malha de 1088 triângulos, e condição de Dirichlet em todas as fronteiras, a Figura 7.1 mostra a evolução dessa função até o estado estacionário. A evolução ao longo das iterações parte da condição nula em todo o domínio e leva à construção do domo até que este se estabilize com valor de máximo aproximadamente igual a 1.

Figura 7.1 – Evolução da resolução da equação de Poisson



Fonte: Produzida pela autora.

7.3 Problema do cone

O problema do cone é um caso de advecção pura, ou seja, o coeficiente de difusão é nulo, $\kappa = 0$, resultando na equação

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{\omega} \phi = f \quad 7.3$$

Nesse teste, busca-se a reconstrução para a função

$$\phi(x, y) = \begin{cases} \cos^2(2\pi r) & ; \text{ se } r = \sqrt{(x - 0,5)^2 + (y - 1)^2} \leq 0,25 \\ 0 & ; \text{ caso contrário} \end{cases} \quad 7.4$$

em um domínio quadrado $\Omega = [0,2] \times [0,2]$, com $\vec{\omega} = [-2\pi(y - 1); 2\pi(x - 1)]$, e condição de fronteira de Dirichlet em toda a fronteira. A Figura 7.2 mostra os gráficos da reconstrução desse problema teste, para ordem 5, em uma malha de 16794 triângulos.

A evolução dessa equação é mostrada nos gráficos da Figura 7.3. Nesta, é mostrada a resolução da equação, em que o cone percorre uma trajetória circular no domínio e retorna para sua posição inicial.

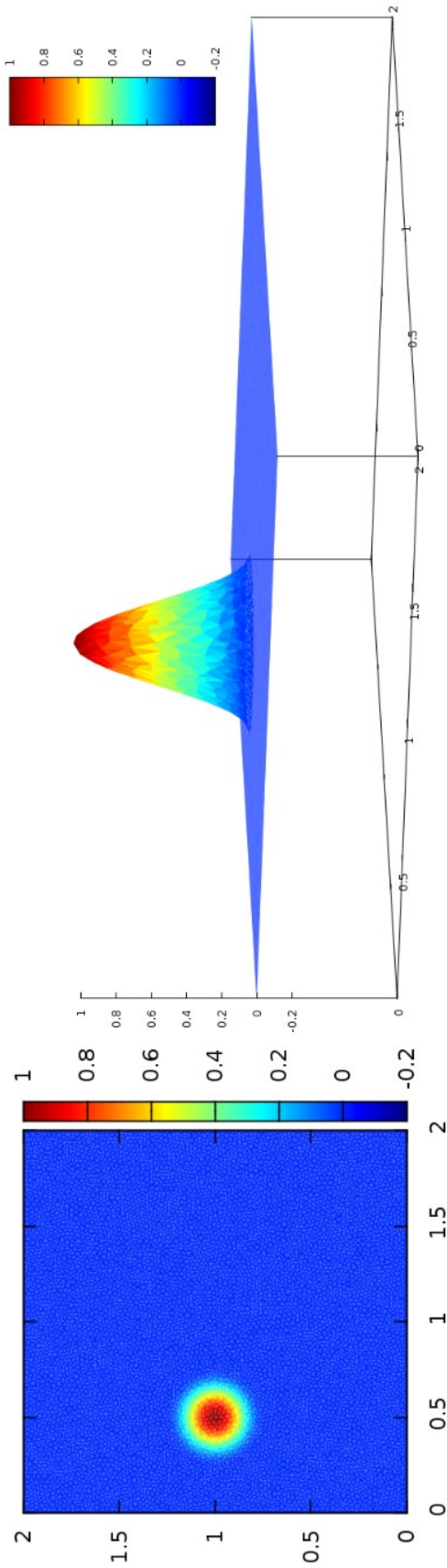
7.4 Advecção-difusão

Foram feitos testes também para casos de advecção-difusão, representados pela equação 5.1,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{\omega} \phi - \nabla \cdot (\kappa \nabla \phi) = f \quad 7.5$$

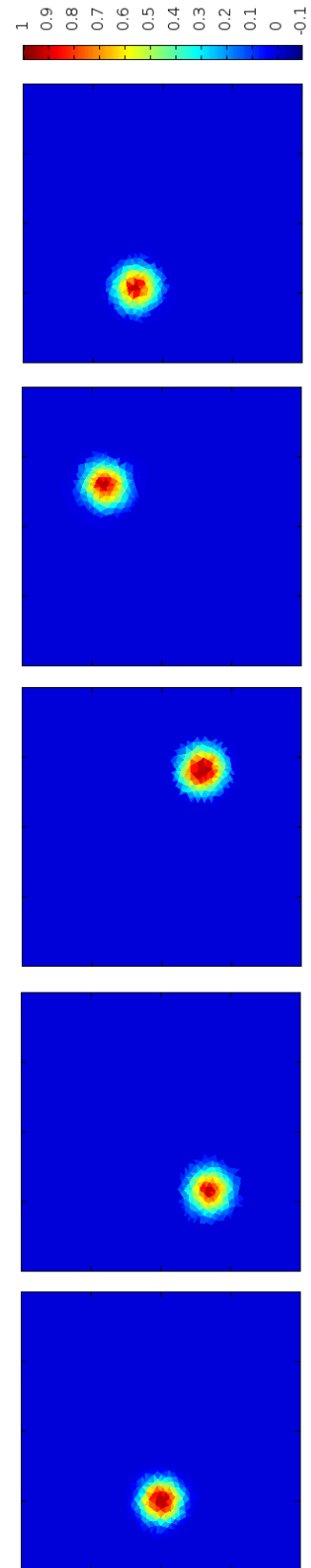
Similar à equação de Poisson, foi estabelecido um domínio quadrado de lado unitário, $\Omega = [0,1] \times [0,1]$, com a solução analítica do problema novamente dada pela função 6.7. A Figura 7.4 mostra a evolução no tempo da equação para um coeficiente de difusão $\kappa = 1$ e velocidade de advecção $\vec{\omega} = [10y(1 - y); 0]$, com condição de Neumann na fronteira direita e de Dirichlet nas demais, ordem de reconstrução 3 em uma malha de 1088 triângulos. Nesse caso, o domo da função é transportado para a direita, e simultaneamente dissipado, reduzindo seu valor de pico até que este seja zero.

Figura 7.2 – Reconstrução da função 7.4



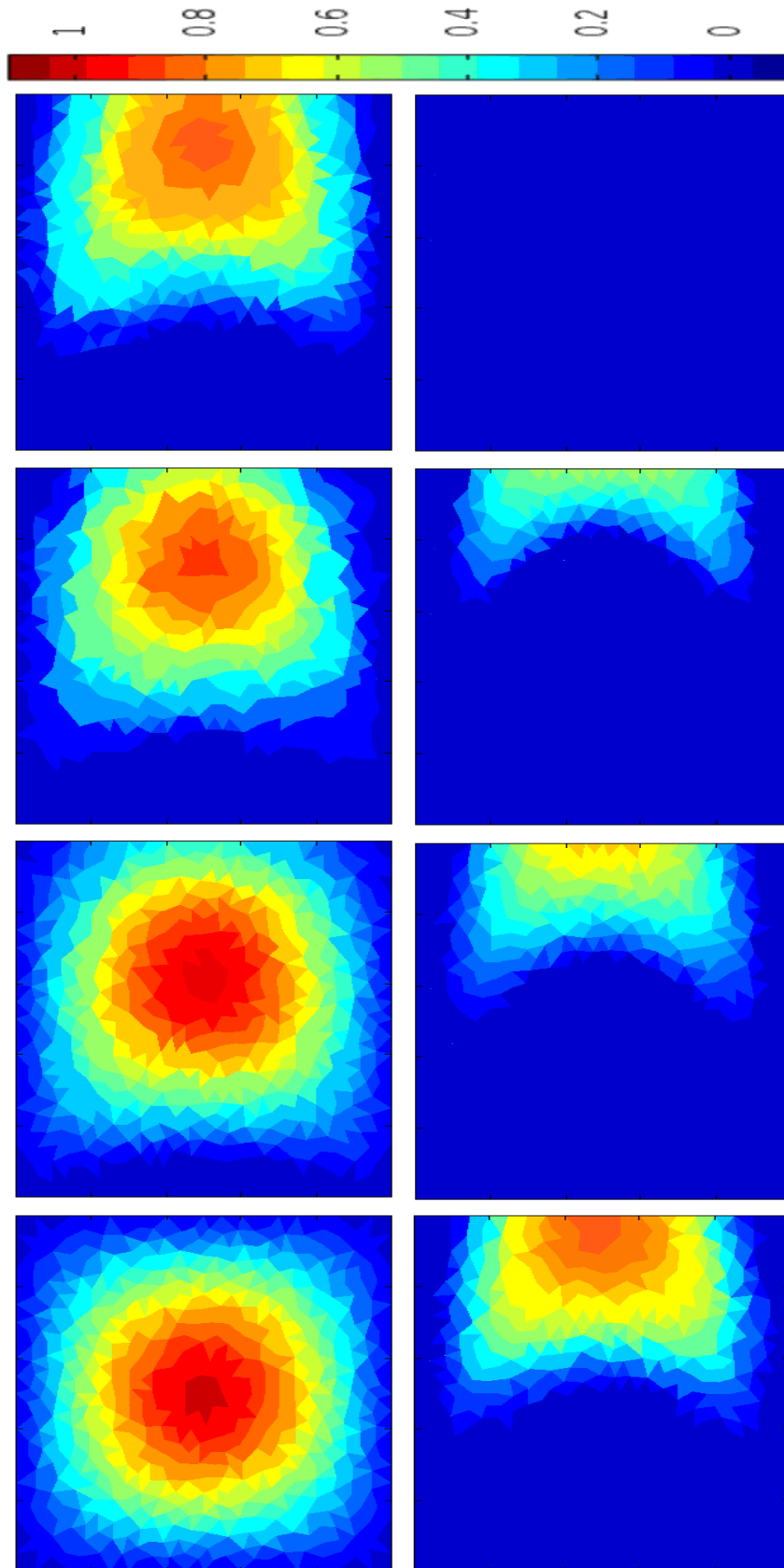
Fonte: Produzida pela autora.

Figura 7.3 – Evolução da resolução do problema do cone



Fonte: Produzida pela autora.

Figura 7.4 – Evolução da resolução da Equação de Advecção-Difusão

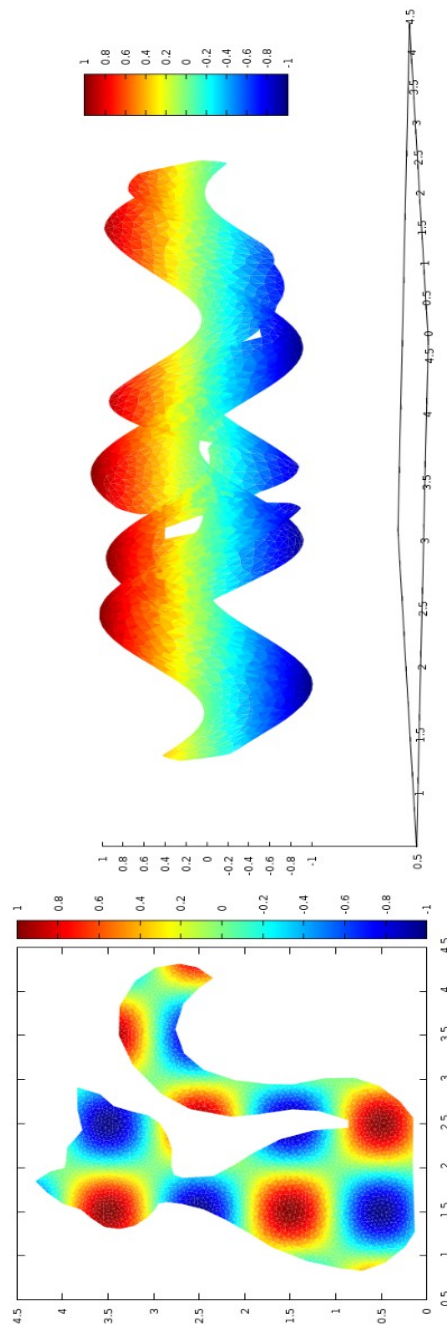


Fonte: Produzida pela autora.

7.5 Geometrias complexas

O método implementado, como citado no início do trabalho, permite a reconstrução de funções em geometrias complexas, a priori, compostas de contornos retilíneos. A Figura 7.5 ilustra a reconstrução da função teste 6.7, para ordem 5, em um domínio de geometria complexa, com uma malha de 5872 triângulos.

Figura 7.5 – Reconstrução da função 6.7 em um domínio de geometria complexa



Fonte: Produzida pela autora.

8 CONCLUSÃO

O desenvolvimento do presente trabalho envolveu muita consulta a referências bibliográficas e um grande aprimoramento de lógica de programação. Muito sobre os métodos numéricos aplicados a problemas reais de engenharia já foi pesquisado, gerando avanços significativos no ramo das simulações computacionais. Mas buscando mais a fundo sobre o tema, muitas oportunidades de desenvolvimento ainda existem, especialmente no que diz respeito à necessidade de simplificações geométricas dos problemas reais. Nesse ponto, o estudo de um método de melhor adaptabilidade a domínios de geometria complexa mostrou-se interessante.

A geração de malhas não-estruturadas, foi dentre as etapas, a que apresentou maior tranquilidade. Já existem muitos softwares capazes de gerar tipos variados de malhas, nos mais diversos domínios, e o programa escolhido, o *Gmsh*, mostrou-se intuitivo e fácil de usar.

Partindo para a montagem da estrutura de dados a partir da malha gerada, um dos grandes desafios foi lidar com a carga computacional e o gerenciamento de memória que essa parte do código em linguagem C demandou, especialmente por envolver um método de reconstrução de alta ordem. Isto porque o trabalho foi desenvolvido em computadores pessoais, de baixa capacidade de processamento. Nesse caso, técnicas de armazenamento de dados foram essenciais no processo. Houve momentos em que a demanda computacional era tão grande, que o computador não era capaz de finalizar a execução do código. Se o trabalho fosse iniciado novamente, este seria um ponto de maior atenção, visto que essa etapa foi uma das mais longas e trabalhosas para a implementação computacional.

A partir daí, as etapas de reconstrução e resolução da equação exigiram uma combinação de conhecimentos de cálculo e lógica de programação para implementação das sub-rotinas finais do programa. Essa fase do trabalho, em especial, mostrou a importância de encontrar e fazer uso adequado de várias fontes de pesquisa. O que acontece, é que muitos artigos expõem boas propostas para a resolução da Equação de Advecção-Difusão, porém as minúcias dos cálculos realizados são, muitas vezes, omitidas, e o cruzamento de informações de vários autores foi o caminho para otimizar os resultados da implementação.

A verificação computacional, como ferramenta indispensável no processo, mostrou a coerência dos resultados obtidos com o que a literatura apresenta sobre o

método empregado. Os resultados obtidos provam, a melhora das respostas do problema conforme aumenta a ordem de reconstrução do problema, o que confirma o mérito dos métodos de alta ordem, como aquele desenvolvido por Gooch e utilizado neste trabalho.

Por fim, quando se observa a capacidade do método de se adaptar a domínios variados sem a necessidade de grandes adaptações do código, percebe-se o potencial desse método para o tratamento de problemas de engenharia ligados ao tema, como os que ocorrem na área de mecânica dos fluidos, por exemplo. O uso das malhas não estruturadas associado ao método de alta ordem é uma combinação poderosa para a resolução de problemas sobre domínios complexos, sem a necessidade de um refinamento exacerbado da malha para uma boa ordem de precisão dos resultados.

Nota-se que, na resolução de problemas que exigem o uso dos métodos numéricos, não só um único método, mas muitos são empregados para resolver partes isoladas de pequenos problemas, que compõem um problema maior. Ficam ainda lançadas como perspectivas de continuação do trabalho o desenvolvimento de um código com adaptação para fronteira curva, a inserção do termo transiente à equação, a aplicação do método a outros tipos de equações e ainda a utilização do esquema estudado em casos tridimensionais, buscando aproximar cada vez mais a capacidade de adaptação a problemas reais de engenharia.

Para os leitores que iniciam os estudos na área de matemática computacional aplicada à engenharia, este trabalho permitiu conhecer de forma simplificada e didática o processo da elaboração de um problema envolvendo EDP e a aplicação do método dos volumes finitos, utilizando alta ordem. Que esta possa ser uma inspiração para estudantes que buscam mais conhecimentos na área que e se interessam pelo mundo das simulações computacionais de problemas de engenharia.

REFERÊNCIAS

- Bakker, A. "Lecture 7 - Meshing, Applied Computational Fluid Dynamics." *bakker.org*. 2002-2006. <http://www.bakker.org/dartmouth06/engs150/07-mesh.pdf>.
- Barbosa, F. P. "Estruturas de dados topológicas aplicadas em simulações de escoamentos compressíveis utilizando volumes finitos e métodos de alta ordem." (Universidade de Sao Paulo, USP) fevereiro 2013.
- Barth, T. J., and P. O. Frederickson. *AIAA Paper 90-0013*, janeiro 1990.
- Biloti, R. "Integração Numérica." (UNICAMP) 2017: 1-16.
- Boris, J. P., and D. L. Book. "Flux corrected transport: I. shasta, a fluid transport algorithm that works." *Journal of Computational Physics*, no. 11 (1973): 38–69.
- Burden, R. L., and J. Douglas Faires. *Numerical Analysis*. 9ª. Vol. 1. BROOKS/COLE CENGAGE Learning, 2010.
- Canuto, C., M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Evolution to complex geometries and applications to fluid dynamics*. Springer Science & Business Media, 2007.
- DEMEC, Departamento de Engenharia Mecânica. "Dinâmica dos Fluidos Computacional." Edited by UFPR - Universidade Federal do Paraná. *Introducao_CFD_2015*, 2015: 1-30.
- . *Introdução ao MEF*. UFPR - Universidade Federal do Paraná, n.d.
- Dunavant, D. A. "High degree efficient symmetrical Gaussian quadrature rules for the triangle." *International journal for numerical methods in engineering* (Wiley Online Library) 21, no. 6 (1985): 1129-1148.
- Free Software Foundation, Inc. *GSL - GNU Scientific Library*. 2009. <https://www.gnu.org/software/gsl/>.
- Geuzaine, C., and J. Remacle. *Gmsh 4.0*. n.d. <http://gmsh.info/doc/texinfo/gmsh.html#Geometry-module>.
- GFFM, Grupo de Fadiga, Fratura e Materiais. *Elementos de Contorno*. Universidade de Brasília - Departamento de Engenharia Mecânica - Faculdade de Tecnologia, n.d.
- Gonçalves, N. D. F. "Método dos Volumes Finitos." Departamento de Matemática Aplicada / Departamento de Matemática Pura/ Faculdade de Ciências da Universidade do Porto. 2007. http://www.fc.up.pt/fcup/contactos/teses/t_050370159.pdf.
- Haelterman, R., Vierendeels J., and D. Van Heule. "A generalization of the Runge–Kutta iteration." *Journal of Computational and Applied Mathematics*, 2009: 52–167.
- Harten, A. "High resolution schemes for hyperbolic conservation laws." *Journal of Computational Physics*, no. 49 (1983): 357–393.
- Heinzi, R. "Finite Volumes." *Concepts for Scientific Computing* (Concepts for Scientific Computing), n.d.
- Hoffmann, K. A., and S. T. Chiang. *Computational Fluid Dynamics*. 4ª. Vol. 1. Wichita, Kansas: Engineering Education System, 2000.
- Kumar, S. S. "Computational Fluid Dynamics (CFD) U7AEA29." *Dept. of Aeronautical Engineering Vel Tech Dr. RR & Dr. SR Technical University, Avadi, Chennai*, n.d.
- Liu, X. D., S. Osher, and T. Chan. "Weighted essentially non-oscillatory schemes." *Journal of Computational Physics*, no. 115 (1994): 200–212.
- Magalhães, Gabriel Marcos. "Resolução Numérica da Equação de Advecção-Difusão Bidimensional via Método dos Volumes Finitos em Malhas Não-Estruturadas usando um Método de Reconstrução de Alta Ordem." (Universidade Federal de Uberlândia - Faculdade de Engenharia Mecânica) 2016: 81.

- Moreno, R. A. "What does Shape Function Mean in Finite Element Formulation? Stochastic Simulation and Lagrangian Dynamics." *Stochastic Simulation and Lagrangian Dynamics*, julho 17, 2011.
- Ollivier-Gooch, C. "Multigrid acceleration of an upwind Euler solver on unstructured meshes." *AIAA journal*, 1995: 1822–1827.
- Ollivier-Gooch, C. "Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction." *Journal of Computational Physics* (Elsevier), 1997: 6–17.
- Ollivier-Gooch, C., A. Nejat, and K. Michalak. "Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations." *AIAA journal*, 2009: 2105–2120.
- Ollivier-Gooch, C., and C. Boivin. "Guaranteed-quality simplicial mesh generation with cell size and grading control." *Engineering with Computers*, 2001: 269–286.
- Ollivier-Gooch, C., and C. Boivin. "Improved cell size and grading in guaranteed quality." *Proceedings* (Sandia National Laboratories), no. 9th International Meshing Roundtable (out 2000): 43-54.
- Ollivier-Gooch, C., and C. Michalak. "High-order finite-volume discretization of the Euler equations on unstructured meshes." *Adaptive high-order methods in computational fluid dynamics*, 2011: 235–268.
- Ollivier-Gooch, C., and M. Van Altena. "A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation." *Journal of Computational Physics*, 2002: 729–752.
- Owen, S. *A survey of Unstructured Mesh Generation Technology*. Dearborii, Michigan: Proceedings of the Seventh International Meshing Roundtable, 1998.
- Pointwise, Inc. The Choice For CFD Meshing. "Easy Meshing with Structured Accuracy: Overset Grids in Pointwise." *The Connector, Pointwise*. May/June 2012.
<http://www.pointwise.com/theconnector/May-2012/Overset-Grids-in-Pointwise.shtml>.
- Saito, O. H. "Esquema numérico com reconstrução mínimos quadrados de alta ordem em malhas não-estruturadas para a formulação euleriana do transporte de partículas." *Tese de doutorado*, janeiro 2008.
- Santana, A. A., and B. F. R. Ribeiro. "Resolução Numérica de EDPs via Méodo dos Volumes Finitos Utilizando um Método de Reconstrução de Alta Ordem Baseado em Mínimos Quadrados." *XXXIII Congresso Nacional de Matemática Aplicada*, 2010.
- Sussman, M. "Two-dimensional Hermite interpolation and mesh generation." *MATH2070: LAB 8: Higher Order Interpolation*, agosto 10, 2016.
- van Leer, B. "Towards the ultimate conservative difference scheme. v. a second order sequel to godunov's method." *Journal of Computational Physics*, no. 32 (1979): 101–136.
- Williams, T., and C. Kelley. *gnuplot homepage*. October 2018. <http://www.gnuplot.info/>.
- ZhiJian, W. "A perspective on high-order methods in computational fluid dynamics." Edited by University of Kansas, Lawrence KS 66045, USA Department of Aerospace Engineering. *SCIENCE CHINA Physics, Mechanics & Astronomy* 59(1), 614701 (Science China Press), 2016.