

Universidade Estadual da Paraíba  
Centro de Ciências Humanas e Exatas  
Curso de Licenciatura Plena em Matemática

Francisco das Chagas dos Santos Marques

Geração de Malhas Estruturadas  
Generalizadas por Meio da Solução  
Numérica de Equações Diferenciais Parciais  
Elípticas

Monteiro - PB, Brasil

Julho de 2014

**Francisco das Chagas dos Santos Marques**

**Geração de Malhas Estruturadas Generalizadas por  
Meio da Solução Numérica de Equações Diferenciais  
Parciais Elípticas**

Trabalho de Conclusão de Curso submetido  
ao Centro de Ciências Humanas e Exatas -  
CCHE da Universidade Estadual da Paraíba -  
UEPB, em cumprimento às exigências legais  
para a obtenção do Título de Graduado no  
Curso de Licenciatura Plena em Matemática.

Orientador: Prof. Dr. Brauner Gonçalves Coutinho

Monteiro - PB, Brasil

Julho de 2014



É expressamente proibida a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano da dissertação.

M357g Marques, Francisco das Chagas dos Santos.

Geração de malhas estruturadas generalizadas por meio da solução numérica de equações diferenciais parciais elípticas [manuscrito] : / Francisco das Chagas dos Santos Marques. - 2014.

48 p. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Matemática) - Universidade Estadual da Paraíba, Centro de Ciências Humanas e Exatas, 2014.

"Orientação: Prof. Dr. Brauner Gonçalves Coutinho, Departamento de Matemática".

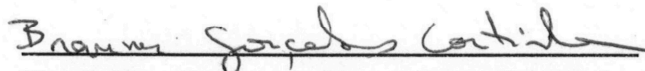
1. Geração de malhas. 2. Equações diferenciais parciais elípticas. 3. Solução numérica. 4. Malhas estruturadas bidimensionais. 5. Linguagem python. I. Título. 21. ed. CDD 510

Francisco das Chagas dos Santos Marques

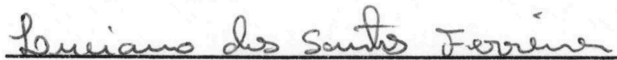
**Geração de Malhas Estruturadas Generalizadas por  
Meio da Solução Numérica de Equações Diferenciais  
Parciais Elípticas**

Trabalho de Conclusão de Curso submetido  
ao Centro de Ciências Humanas e Exatas -  
CCHE da Universidade Estadual da Paraíba -  
UEPB, em cumprimento às exigências legais  
para a obtenção do Título de Graduado no  
Curso de Licenciatura Plena em Matemática.

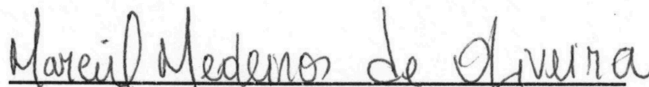
Trabalho aprovado. Monteiro - PB, Brasil, 29 de julho de 2014:



**Prof. Dr. Brauner Gonçalves Coutinho**  
Orientador - UEPB



**Prof. Ms. Luciano dos Santos Ferreira**  
UEPB



**Prof. Ms. Marciel Medeiros de Oliveira**  
UEPB / UFCG

Monteiro - PB, Brasil

Julho de 2014

*Este trabalho é dedicado aos meus pais:  
José Edvaldo e Helena, a minha  
noiva Francielle Cristina,  
a minha sobrinha Luna Loise  
e aos meus avós José Rodrigues  
e Maria Tereza (in memoriam).*

# Agradecimentos

Agradeço à minha família por todo apoio, especialmente a minha mãe, que desde sempre vem me incentivando à estudar.

Agradeço à minha amiga Cleidilaine, pelo apoio e cobrança para que este trabalho fosse realizado.

Agradeço aos meus colegas de turma, que diretamente ou indiretamente contribuíram para realização deste trabalho.

Agradeço a minha noiva Francielle Cristina pela colaboração para realização deste trabalho.

Agradeço aos meus professores, pela transmissão de conhecimento e experiências para que nos tornássemos bons profissionais, em especial ao meu orientador Brauner, que desde o início mostrou-se muito atencioso e colaborativo para a realização deste trabalho.

*"O segredo de progredir é começar. O segredo de começar é dividir as tarefas árduas e complicadas em tarefas pequenas e fáceis de executar, e depois começar pela primeira"*  
*(Mark Twain)*

# Resumo

Alguns problemas físicos da ciência e engenharia são modelados matematicamente por conjuntos de equações cuja solução analítica é difícil ou mesmo impossível. Nestes casos, uma alternativa atraente é o uso de técnicas numéricas, especialmente hoje, com a evolução no processamento e o baixo custo dos computadores. Normalmente, quando o modelo matemático envolve equações diferenciais parciais, a solução exige que o domínio de estudo seja discretizado em um número finito de sub-regiões ou pontos, tornando-se necessário o uso de malhas computacionais. O presente trabalho demonstra uma técnica numérica para gerar malhas estruturadas bidimensionais a partir da solução de equações diferenciais parciais elípticas. O método numérico de Gauss-Seidel foi utilizado para resolver o sistema linear resultante. O algoritmo desenvolvido durante esta pesquisa foi inteiramente implementado em linguagem Python. Adicionalmente, foi desenvolvida uma interface gráfica que exibe a malha gerada para melhor visualização dos resultados numéricos. Geometrias com bordas irregulares foram utilizadas para validar a metodologia.

**Palavras-chaves:** geração de malhas. equações diferenciais parciais elípticas. solução numérica. malhas estruturadas bidimensionais. linguagem Python.

# Abstract

Some physical problems in science and engineering are modeled mathematically by sets of equations whose analytical solution is difficult or impossible. In these cases an attractive alternative is the use of numerical techniques, especially today, with developments in processing and low cost of computers. Usually, when the mathematical model involves partial differential equations, the solution requires the discretization of the domain of study in a finite number of sub-regions or points, making necessary the use of computational meshes. The present work demonstrates a numerical technique to generate two-dimensional structured meshes from the solution of elliptic partial differential equations. Gauss-Seidel method used to solve the resulting linear system. The algorithm developed during this research was fully implemented in Python language. Additionally, a graphical interface that displays the generated mesh for better visualization of the numerical results was created. Geometries with irregular borders were used to validate the methodology.

**Key-words:** grid generation. elliptic partial differential equations. numerical solution. bidimensional structured grids. Python language.

# Lista de ilustrações

Figura 1 – Tipos de malhas computacionais . . . . .	19
Figura 2 – Discretização de uma geometria com fronteira curvilínea . . . . .	20
Figura 3 – Malha de uma traqueia e dos tubos bronquiais . . . . .	21
Figura 4 – Malha do perfil do aerofólio . . . . .	21
Figura 5 – Plano físico e plano transformado . . . . .	23
Figura 6 – Ponto interno P e seus vizinhos (N, S, E, W, NW, NE, SW, SE) em uma malha bidimensional no domínio transformado . . . . .	31
Figura 7 – Malha 2x2 volumes . . . . .	34
Figura 8 – Arquivo de saída contendo as coordenadas dos pontos de uma malha com 5x5 volumes . . . . .	37
Figura 9 – Fluxograma do código computacional . . . . .	38
Figura 10 – Geometrias com fronteiras irregulares . . . . .	39
Figura 11 – Malhas com 5x5 volumes para as três geometrias estudadas . . . . .	39
Figura 12 – Malhas com 10x10 volumes para as três geometrias estudadas . . . . .	40
Figura 13 – Malhas com 15x15 volumes para as três geometrias estudadas . . . . .	40
Figura 14 – Tempo de execução em função da quantidade de volumes . . . . .	41
Figura 15 – Visualização das malhas geradas no software <i>Transcal 1.1</i> . . . . .	41
Figura 16 – Malha circular com 15x15 volumes com diversos valores de parada . . . . .	42
Figura 17 – Tempo de execução em função do critério de parada . . . . .	42
Figura 18 – Malha com 15x15 volumes para critérios de parada diferentes . . . . .	43
Figura 19 – Evolução da solução no decorrer do processo iterativo . . . . .	44
Figura 20 – Janela para entrada de dados . . . . .	45
Figura 21 – Janela para visualização da malha . . . . .	45



## Lista de tabelas

Tabela 1 – Comparação entre as três técnicas de solução . . . . .	18
Tabela 2 – Vantagens e desvantagens de malhas estruturadas e não-estruturadas .	22
Tabela 3 – Aproximações por Gauss-Seidel para a solução do sistema . . . . .	36

# Lista de abreviaturas e siglas

CFD	<i>Computational Fluid Dynamics</i>
EDP	<i>Equações Diferenciais Parciais</i>
MDF	<i>Método das Diferenças Finitas</i>

# Lista de símbolos

## *Letras Latinas*

$A$	Coeficiente de equações algébricas, [-]
$J$	Jacobiano da transformação de coordenadas, [-]
$a$	Fator usado na transformação das equações de geração, [-]
$b$	Fator usado na transformação das equações de geração, [-]
$c$	Fator usado na transformação das equações de geração, [-]
$d$	Fator usado na transformação das equações de geração, [-]
$d^T$	Domínio transformado, [-]
$d^F$	Domínio físico, [ $m^2$ ]
$e$	Fator usado na transformação das equações de geração, [-]
$x$	Coordenada cartesiana, [ $m$ ]
$x_\xi$	Derivada de $x$ com relação a $\xi$ , [-]
$x_\eta$	Derivada de $x$ com relação a $\eta$ , [-]
$y$	Coordenada cartesiana, [ $m$ ]
$y_\xi$	Derivada de $y$ com relação a $\xi$ , [-]
$y_\eta$	Derivada de $y$ com relação a $\eta$ , [-]

## *Letras Gregas*

$\Phi$	Variável dependente da equação discretizada, [-]
$\Delta x$	Dimensão na direção $x$ , [ $m$ ]
$\Delta y$	Dimensão na direção $y$ , [ $m$ ]
$\Delta \xi$	Dimensão de uma célula do plano transformado, na direção $\xi$ , [-]
$\Delta \eta$	Dimensão de uma célula do plano transformado, na direção $\eta$ , [-]
$\alpha$	Fator usado na transformação das equações de geração, [-]
$\beta$	Fator usado na transformação das equações de geração, [-]

$\xi$	Coordenada no plano transformado, [-]
$\xi_x$	Derivada de $\xi$ com relação a $x$ , [-]
$\xi_y$	Derivada de $\xi$ com relação a $y$ , [-]
$\eta$	Coordenada no plano transformado, [-]
$\eta_x$	Derivada de $\eta$ com relação a $x$ , [-]
$\eta_y$	Derivada de $\eta$ com relação a $y$ , [-]

***Subscritos***

$E, N, S, W$  Pontos centrais dos volumes vizinhos diretos do volume  $P$

$NE, NW, SE, SW$  Pontos centrais dos volumes vizinhos cruzados do volume  $P$

# Sumário

	Introdução . . . . .	15
1	REVISÃO BIBLIOGRÁFICA . . . . .	16
1.1	Dinâmica dos Fluidos Computacional (CFD) . . . . .	17
1.2	Malha . . . . .	18
1.2.1	Malha estruturada . . . . .	19
1.2.1.1	Malha generalizada . . . . .	20
1.2.2	Malha não estruturada . . . . .	21
1.2.3	Malha híbrida . . . . .	22
2	MODELAGEM MATEMÁTICA . . . . .	23
2.1	Métricas e Jacobiano da transformação . . . . .	24
2.2	Geração de malhas bidimensionais . . . . .	25
2.2.1	Transformação das equações de geração da malha . . . . .	26
3	MODELAGEM NUMÉRICA E COMPUTACIONAL . . . . .	30
3.1	Discretização das Equações . . . . .	30
3.2	Solução do Sistema Linear . . . . .	34
3.2.1	Método iterativo de Gauss-Seidel . . . . .	35
3.2.2	Critério de parada . . . . .	36
3.3	Implementação do código computacional . . . . .	36
3.3.1	A linguagem Python . . . . .	36
3.3.2	A interface gráfica . . . . .	37
3.3.3	Fluxograma . . . . .	38
4	RESULTADOS E DISCUSÕES . . . . .	39
4.1	Gerador de malha . . . . .	39
4.1.1	A influência do critério de parada na qualidade da solução . . . . .	42
4.1.2	A evolução da solução no processo iterativo . . . . .	43
4.2	O programa desenvolvido . . . . .	44
	Considerações finais . . . . .	46
	Referências . . . . .	47

# Introdução

O constante avanço na tecnologia dos computadores permite aos pesquisadores uma alternativa para solucionar diversos problemas físicos que podem ser modelados matematicamente, minimizando riscos e custos. Um exemplo disso é o estudo de fenômenos relacionados ao movimento dos fluidos.

A solução de tais modelos é muito complexa, sendo na maioria das vezes utilizadas técnicas numéricas que fornecem uma solução aproximada, mas muito útil do ponto de vista da ciência e engenharia. Uma característica desta técnica é a divisão do domínio de solução em células ou pontos, determinados por uma malha.

Existem diversas formas para se obter uma malha. O presente trabalho mostra um método de geração de malhas generalizadas coincidentes com as fronteiras através da solução numérica de equações diferenciais parciais.

O trabalho inicia com uma revisão da literatura no Capítulo 1, descrevendo a importância dos escoamentos fluidos para a sociedade num contexto histórico e como podem ser estudados. O capítulo ainda define o que são malhas computacionais, explicando sua necessidade e quais os principais tipos utilizados.

O Capítulo 2 mostra toda a modelagem matemática, incluindo as equações de geração da malha e como é feita a transformação destas equações do plano físico para o computacional.

O Capítulo 3 descreve conceitos importantes da metodologia numérica: discretização das equações, método iterativo para solução do sistema linear resultante, critério de parada para o procedimento iterativo, etc. O capítulo traz também detalhes importantes sobre a abordagem usada para desenvolvimento do programa, com informações sobre a linguagem de programação Python e a interface gráfica Tkinter.

Os resultados obtidos são apresentados e discutidos no Capítulo 4.

# 1 Revisão bibliográfica

Segundo Fortuna (2000), fluidos em movimento estão presentes em toda natureza: no corpo humano, como na corrente sanguínea; no ar, com o vento; na água, com as correntes e marés. As características do movimento dos fluidos tem sido objeto de estudo há muito tempo, pois, nos afetam diretamente. Os egípcios, por exemplo, mediam o tempo com o auxílio de relógios de água. Arquimedes definiu em seu princípio, as condições para que um corpo, quando mergulhado em um fluido, flutuasse ou não. Historicamente, o estudo de problemas de mecânica dos fluidos ocorreu inicialmente de forma experimental, muito antes do uso de modelos matemáticos. As primeiras equações dos movimentos dos fluidos foram deduzidas por Leonard Euler. Mas os modelos matemáticos que descrevem o comportamento dos fluidos só ganharam força no século XIX, com as equações de Navier-Stokes, que constituem um conjunto de equações diferenciais parciais que incluem termos como densidade, viscosidade, força gravitacional e pressão (MIRANDA, 2007).

$$\begin{aligned} \rho \frac{Du}{Dt} = \rho g_x - \frac{\partial \rho}{\partial x} + \frac{\partial}{\partial x} \left[ \mu \left( 2 \frac{\partial u}{\partial x} - \frac{2}{3} \nabla V \right) \right] + \\ \frac{\partial}{\partial y} \left[ \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial}{\partial z} \left[ \mu \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right] \end{aligned} \quad (1.1)$$

$$\begin{aligned} \rho \frac{Dv}{Dt} = \rho g_y - \frac{\partial \rho}{\partial y} + \frac{\partial}{\partial y} \left[ \mu \left( 2 \frac{\partial v}{\partial y} - \frac{2}{3} \nabla V \right) \right] + \\ \frac{\partial}{\partial z} \left[ \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right] + \frac{\partial}{\partial x} \left[ \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \end{aligned} \quad (1.2)$$

$$\begin{aligned} \rho \frac{Dw}{Dt} = \rho g_z - \frac{\partial \rho}{\partial z} + \frac{\partial}{\partial z} \left[ \mu \left( 2 \frac{\partial w}{\partial z} - \frac{2}{3} \nabla V \right) \right] + \\ \frac{\partial}{\partial x} \left[ \mu \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right] + \frac{\partial}{\partial y} \left[ \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right] \end{aligned} \quad (1.3)$$

Fenômenos relacionados ao movimento dos fluidos podem ser bastante complexos. Análises teóricas apenas não conseguem explicar esses fenômenos de forma precisa. Para um estudo mais preciso, os pesquisadores fazem uso de métodos experimentais<sup>1</sup> (FORTUNA, 2000). Entretanto, métodos experimentais envolvem altos custos e tempo na preparação de modelos, risco ambiental em alguns casos, sem falar na dificuldade de representar em laboratório as condições do problema físico desejado.

<sup>1</sup> Um método experimental é um conjunto de procedimentos científicos de controle experimental e verificação, baseados apenas na experiência.

Devido aos avanços tecnológicos cada vez mais é comum o uso de simulação numérica e computacional para estudos de diversos problemas físicos de alta complexidade, principalmente aqueles que envolvem fluidos em movimento, tais como: análises de fluxo sanguíneo (CIRILO; BORTOLI, 2006), simulações de aerofólios (RIBEIRO, 2002), geração de efeitos visuais para indústria de entretenimento (GIRALDI et al., 2005), estudo de escoamentos multifásicos (COUTINHO, 2013) e (SANTOS, 2010), entre outros.

## 1.1 Dinâmica dos Fluidos Computacional (CFD)

Como visto nas Equações (1.1), (1.2) e (1.3), as equações de Navier-Stokes são equações diferenciais parciais (EDP), não lineares, de difícil ou impossível solução. Para alguns casos é possível utilizar versões simplificadas destas equações, como por exemplo, nas animações de fluidos usadas na indústria do cinema (GIRALDI et al., 2005). Entretanto, dependendo do problema físico ou da complexidade do modelo, a introdução de simplificações pode conduzir a resultados imprecisos ou ainda desviar o modelo do problema físico original (MALISKA, 2004). Isso explica o motivo pelo qual, para problemas mais complexos, muitos pesquisadores não abrem mão do uso de métodos experimentais.

Com a popularização e barateamento dos recursos computacionais, uma alternativa bastante atraente nos dias de hoje para os pesquisadores é a obtenção de soluções para equações complexas (como as de Navier-Stokes) utilizando técnicas numéricas, onde expressões algébricas substituem as derivadas parciais (MALISKA, 2004).

O uso de métodos numéricos para solução de problemas físicos, aliado ao poder de processamento do computador é conhecido como simulação computacional. Quando o problema envolve movimento de fluidos, a área de conhecimento é denominada Dinâmica dos Fluidos Computacional, ou em inglês *Computational Fluid Dynamics* (CFD).

Por meio de CFD, pode-se reduzir o número de experimentos, fazer análises de fenômenos que não poderiam ser representados em laboratórios de forma eficiente. Ainda há a possibilidade de realizar rapidamente alterações nos parâmetros do processo, reduzindo custos e tempo execução. A Tabela 1 mostra as vantagens e desvantagens de cada metodologia (FORTUNA, 2000).

O uso de CFD é tão importante e complexo que foi incluído no programa Grandes Desafios, do governo dos Estados Unidos da América. Esse programa definiu problemas em diversas áreas da ciência e tecnologia cujas soluções têm grande impacto econômico e científico (FORTUNA, 2000).

Para Santos (2010), algumas das principais vantagens da aplicação de CFD são:

- Flexibilidade de alterar as configurações do problema sem grandes custos;



Tabela 1 – Comparação entre as três técnicas de solução, Fonte: (FORTUNA, 2000).

Técnica	Vantagem	Desvantagem
Experimental	Mais Realista	Equipamento exigido Problemas de escala Dificuldades de medição Custo operacional
Teórica	Mais geral Fórmula fechada	Restrita a geometria e processos físicos simples Geralmente restrita a problemas lineares
Numérica	Não há restrição a linearidade Geometria e processos complicados Evolução temporal do processo	Erros de truncamento Prescrição das condições de contorno apropriadas Custos computacionais

- O tempo de resposta é mais rápido do que as experiências em laboratório, proporcionando uma maior quantidade de análises;
- Fornece informações relevantes sobre as variáveis locais do escoamento;
- Fornece informações em regiões de difícil obtenção de dados a partir de experimentos em laboratório.

## 1.2 Malha

Normalmente, a solução numérica de um modelo baseado em equações diferenciais exige que tanto as equações como a região de estudo (domínio) sejam expressas de forma adaptada, uma vez que não é possível solucionar numericamente em uma região contínua, devido à infinidade de pontos da mesma. Inicialmente divide-se o domínio em pequenas partes (células), delimitadas por linhas ou até mesmo pontos. Esse processo de divisão é chamado de discretização. A ideia da técnica numérica é obter as soluções nessas sub-regiões do domínio. O conjunto de linhas que divide o domínio recebe o nome de malha.

A qualidade da solução numérica depende do número de divisões do domínio e de como a malha está distribuída. A solução melhora com o aumento do número de elementos da malha. Entretanto, precisamos balancear a precisão da solução através do refinamento da malha com o custo computacional (SANTOS, 2010). Maliska (2004) afirma que ao aumentarmos a quantidade de células da malha estamos também aumentando o número de incógnitas, juntamente com o sistema linear, tornando a solução mais lenta.

As malhas podem ser definidas como estruturada, não estruturada e híbrida, mostradas na Figura 1.

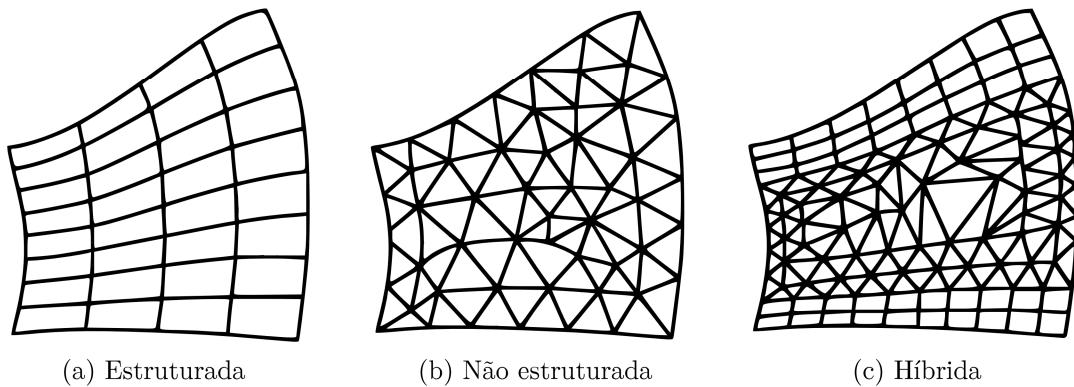


Figura 1 – Tipos de malhas computacionais, Fonte: (SANTOS, 2010).

### 1.2.1 Malha estruturada

Coutinho (2013) traz uma definição de malha estruturada como sendo constituída por linhas agrupadas em famílias que nunca interceptam linhas da mesma família, mas interceptam todas as linhas das outras famílias em uma determinada sequência. Uma malha deste tipo consiste em elementos bidimensionais e tridimensionais que são quadriláteros e hexaédricos, respectivamente. Nesse tipo de malha, cada elemento tem sempre o mesmo número de elementos vizinhos, a não ser quando o elemento pertence ao contorno.

A malha estruturada pode ser moldada para sofrer alterações como torção e esticamento. Geradores de malha estruturada podem utilizar equações diferenciais elípticas sofisticadas para aperfeiçoar a forma da malha automaticamente buscando a uniformidade e ortogonalidade (SANTOS, 2010).

Ainda de acordo com Santos (2010), a base para construção deste tipo de malha é a associação de blocos com diferentes subdomínios geométricos. Esta técnica permite conectar vários blocos juntos e construir o domínio inteiro, com uma distribuição uniforme dos elementos em cada bloco. Este tipo de malha oferece algumas vantagens sobre outros tipos por apresentar uma execução simples e por exigir menor capacidade de armazenamento.

A grande desvantagem de uma malha estruturada é a falta de flexibilidade em se ajustar a geometrias extremamente complexas.

As malhas estruturadas que são mais facilmente obtidas são as ortogonais (cartesiana, cilíndrica ou esférica). As malhas cartesianas podem ser obtidas a partir de linhas paralelas aos eixos de um sistema cartesiano ortogonal (COUTINHO, 2013). No entanto, conforme pode ser visto na Figura 2, quando se tem geometrias irregulares, é muito difícil aplicar condições de contorno, pois é necessária uma trabalhosa determinação dos comprimentos entre a fronteira e os pontos da malha (MALISKA, 2004).

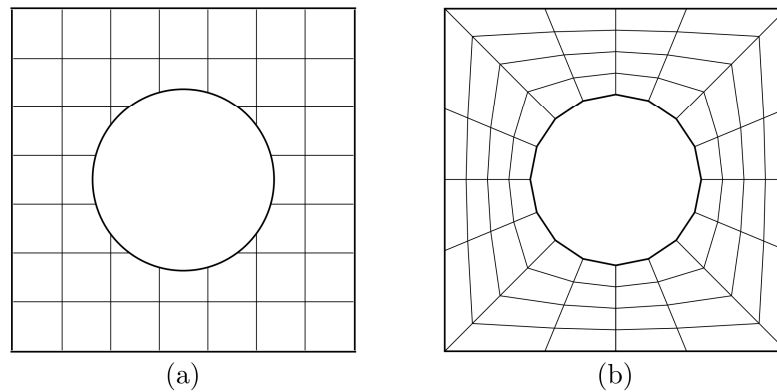


Figura 2 – Discretização de uma geometria com fronteira curvilínea: (a) cartesiana e (b) coincidente com a fronteira, Fonte: (COUTINHO, 2013).

### 1.2.1.1 Malha generalizada

Pode-se observar que a malha da Figura 2 (b) discretiza melhor a geometria mostrada uma vez que as linhas demarcam as células seguindo as fronteiras do domínio. Este tipo de malha coincidente com as fronteiras é também chamado de generalizada ou curvilínea generalizada.

Segundo Maliska (2004), as principais razões para uso de discretização coincidente com as fronteiras são:

- a) Necessidade de solução de problemas cada vez mais complexos, em geometrias arbitrárias;
- b) Dificuldade de solução de tais problemas com sistemas convencionais, especialmente na aplicação de condições de contorno;
- c) Possibilidade de uso de malhas adaptativas;
- d) Possibilidade de desenvolvimento de metodologias generalizadas.

Os trabalhos de Cirilo e Bortoli (2006) e Ribeiro (2002) são exemplos de aplicações do uso de malhas estruturadas coincidentes com as fronteiras. O primeiro apresenta um método para a geração de malhas em blocos estruturados bidimensionais para geometrias de uma traqueia e dos tubos bronquiais, visando descrever o fluxo nesses meios para melhor compreender certos males, como por exemplo, a fibrose cística.

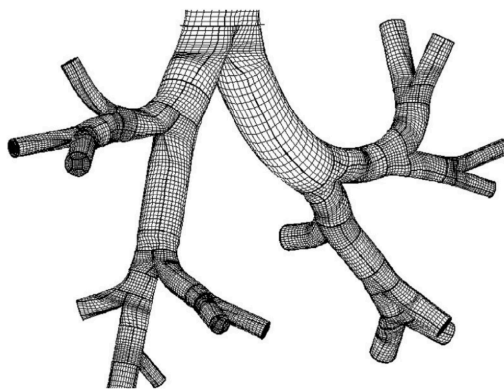


Figura 3 – Malha de uma traqueia e dos tubos bronquiais, Fonte: Google.

Já Ribeiro (2002) utilizou um software comercial (ICEM-CFD) para a construção de malhas estruturadas para simulação de aerofólios (seções de asa de aviões).

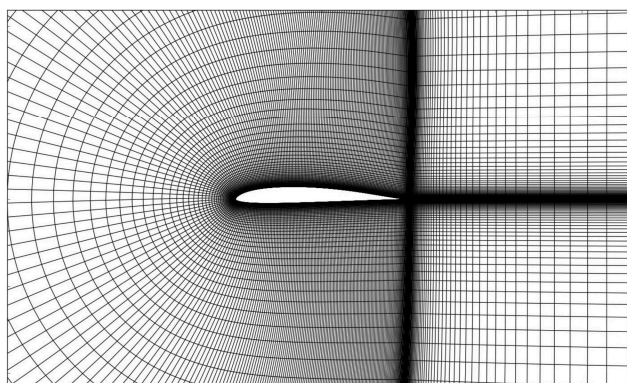


Figura 4 – Malha do perfil do aerofólio, Fonte: Google.

### 1.2.2 Malha não estruturada

Malhas não estruturadas, dividem o domínio em regiões com formas variadas, juntas, que compartilham as fronteiras com as regiões vizinhas (COUTINHO, 2013). Uma malha deste grupo consiste de elementos bidimensionais, triângulos ou quadriláteros, e de elementos tridimensionais, tetraedros ou hexaédricos. Como ocorrem com as malhas estruturadas, os elementos podem sofrer deformações para se ajustar ao domínio (SANTOS, 2010).

Malhas não estruturadas se ajustam com grande facilidade a geometrias complexas (MALISKA, 2004), porém exigem uma capacidade de armazenamento maior que malhas estruturadas. Essas malhas conseguem representar mais facilmente as geometrias complexas, devido à maior flexibilidade de forma que seus elementos podem assumir. Este tipo malha é gerado automaticamente, porém podem conter erros mensuráveis através de valores de Skewness e Aspect Ratio (SANTOS, 2010).

### 1.2.3 Malha híbrida

Uma malha híbrida consiste na utilização simultânea das malhas estruturadas e não estruturadas. A vantagem desta malha é que podemos escolher qual tipo devemos usar para estudar determinadas regiões, por exemplo, nas regiões que necessitam maior detalhamento pode-se usar a malha estruturada, já nas regiões de menor interesse usamos a malha não estruturada. Assim sendo, a habilidade para controlar a forma e distribuição da malha localmente é uma ferramenta poderosa que pode render malhas de boa qualidade e garantir resultados satisfatórios (SANTOS, 2010).

A desvantagem de uma malha híbrida é que exige muita prática e experiência na geração de malhas para geometrias complexas.

Tabela 2 – Vantagens e desvantagens de malhas estruturadas e não-estruturadas, Fonte: (COUTINHO, 2013).

Propriedade	Estruturadas	Não estruturadas
Modelagem de geometrias complexas	-	+
Refinamento de malha local (adaptativo)	-	+
Automatização de geração de malha	-	+
Esforço para geração de malha	+	-
Esforço de programação	+	-
Armazenamento e gerenciamento de dados	+	-
Solução do sistema de equações algébricas	+	-
Paralelização e vetorização de solvers	+	-
+ vantagem, - desvantagem		

## 2 Modelagem matemática

Uma malha generalizada pode ser obtida manualmente, com as posições dos pontos de intersecção das linhas coordenadas sendo especificadas uma a uma. Esse processo seria lento, principalmente caso se deseje que a malha possua mais células. Busca-se, portanto, a utilização de métodos automáticos, capazes de obter uma malha de maneira mais rápida possível. Entre os métodos disponíveis, destaca-se o método de solução de sistemas de equações diferenciais (MALISKA, 2004). De acordo com o autor, a obtenção das linhas coordenadas significa determinar uma transformação definida pelas seguintes relações:

$$\xi = \xi(x, y) \quad (2.1)$$

$$\eta = \eta(x, y). \quad (2.2)$$

Para ilustrar como é feita a transformação, considere a Figura 5, onde um ponto  $P$  pode ser mapeado em dois sistemas coordenados de eixos distintos: um físico em coordenadas cartesianas  $(x, y)$  e outro transformado, baseado em coordenadas generalizadas  $(\xi, \eta)$ .

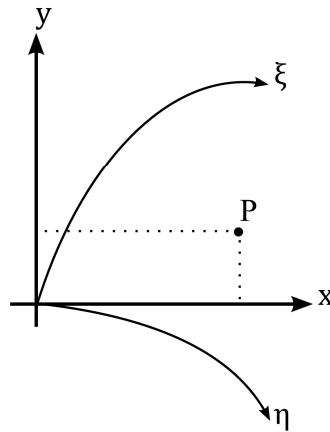


Figura 5 – Plano físico e plano transformado.

A transformação é obtida relacionando as coordenadas dos pontos nos dois sistemas de coordenadas. Usando a regra da cadeia para derivadas parciais, obtêm-se a seguinte expressão:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (2.3)$$

e analogamente,

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial}{\partial \eta} \frac{\partial \eta}{\partial y}. \quad (2.4)$$

Escrevendo as derivadas com a notação de subscritos, onde, por exemplo:  $\frac{\partial \xi}{\partial x} = \xi_x$ , obtemos

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi} \xi_x + \frac{\partial}{\partial \eta} \eta_x \quad (2.5)$$

similarmente tem-se,

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial \xi} \xi_y + \frac{\partial}{\partial \eta} \eta_y. \quad (2.6)$$

As derivadas da transformação  $\xi_x$ ,  $\xi_y$ ,  $\eta_x$  e  $\eta_y$  devem ser determinadas a partir das relações funcionais entre as Equações (2.1) e (2.2).

## 2.1 Métricas e Jacobiano da transformação

Os termos  $\xi_x$ ,  $\xi_y$ ,  $\eta_x$  e  $\eta_y$  das Equações (2.5) e (2.6) são definidos como sendo as *Métricas da Transformação*, ou simplesmente métricas. A interpretação de uma métrica fica mais clara observando a seguinte aproximação:

$$\xi_x = \frac{\partial \xi}{\partial x} \approx \frac{\Delta \xi}{\Delta x}. \quad (2.7)$$

Esta expressão mostra que, no caso bidimensional, as métricas representam razões entre comprimentos do plano físico e do plano transformado. A partir das Equações (2.1) e (2.2), podemos escrever as derivadas em cada um dos eixos transformados, ou seja,

$$d\xi = \xi_x dx + \xi_y dy \quad (2.8)$$

$$d\eta = \eta_x dx + \eta_y dy. \quad (2.9)$$

Escrevendo na forma matricial:

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \quad (2.10)$$

ou ainda,

$$[d^T] = [A] [d^F] \quad (2.11)$$

onde  $d^T$  e  $d^F$  referem-se ao domínio transformado e físico, respectivamente.

Substituindo as variáveis independentes pelas dependentes nas Equações (2.1) e (2.2), tem-se:

$$x = x(\xi, \eta) \quad (2.12)$$

$$y = y(\xi, \eta). \quad (2.13)$$

Analogamente às Equações (2.8) e (2.9), podemos escrever as derivadas das Equações (2.12) e (2.13) como sendo:

$$dx = x_\xi d\xi + x_\eta d\eta \quad (2.14)$$

$$dy = y_\xi d\xi + y_\eta d\eta. \quad (2.15)$$

Na forma matricial:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \quad (2.16)$$

ou,

$$[d^F] = [B] [d^T]. \quad (2.17)$$

Multiplicando a Equação (2.17) por  $[B]^{-1}$ , obtemos:

$$[d^T] = [B]^{-1} [d^F]. \quad (2.18)$$

Analisando as Equações (2.11) e (2.18), podemos observar que:

$$[A] = [B]^{-1}. \quad (2.19)$$

Sabendo que  $[B]^{-1} = \frac{1}{\det[B]} \bar{B}$ , onde  $\bar{B}$  representa a matriz transposta de cofatores da matriz  $[B]$  e  $\frac{1}{\det[B]}$  é o Jacobiano da transformação, através de certa manipulação matemática, podemos chegar a:

$$\xi_x = J y_\eta \quad (2.20)$$

$$\xi_y = -J x_\eta \quad (2.21)$$

$$\eta_x = -J y_\xi \quad (2.22)$$

$$\eta_y = J x_\xi \quad (2.23)$$

onde,

$$J = (x_\xi y_\eta - x_\eta y_\xi)^{-1}. \quad (2.24)$$

O Jacobiano “ $J$ ” é muito importante para a mudança do domínio físico para o domínio transformado. Para o caso bidimensional, representa a razão entre uma área no plano físico e no plano transformado (computacional).

## 2.2 Geração de malhas bidimensionais

É possível gerar malhas de diversas formas, até mesmo manualmente, como já foi observado. Diversos aplicativos pagos de CFD permitem a geração automática de malhas, por exemplo: ANSYS Fluent, ANSYS CFX e PHOENICS. Também existem os gratuitos, tais como: OpenFOAM, NETGEN e MeshLab. Este trabalho mostra um modo de gerar



automaticamente malhas bidimensionais curvilíneas generalizadas por meio da solução numérica de equações diferenciais parciais elípticas.

Segundo Maliska (1986), as equações que permitem gerar as linhas coordenadas de uma malha bidimensional podem ser dadas por:

$$\xi_{xx} + \xi_{yy} = 0 \quad (2.25)$$

$$\eta_{xx} + \eta_{yy} = 0 \quad (2.26)$$

onde  $x$  e  $y$  são as coordenadas cartesianas (domínio físico) enquanto que  $\xi$  e  $\eta$  são as coordenadas generalizadas (domínio computacional). O termo  $\xi_{xx}$  denota a derivada dupla de  $\xi$  com relação à  $x$ .

### 2.2.1 Transformação das equações de geração da malha

Dada a transformação apresentada pelas Equações (2.1) e (2.2) podemos fazer a transformação de coordenadas do domínio físico para o transformado, de acordo com o procedimento descrito a seguir (HOFFMANN; CHIANG, 2000): considere uma função  $f(\xi, \eta)$ . Através da regra da cadeia para derivadas parciais, obtemos as seguintes expressões:

$$f_x = f_\xi \xi_x + f_\eta \eta_x \quad (2.27)$$

e

$$f_y = f_\xi \xi_y + f_\eta \eta_y. \quad (2.28)$$

Podemos rearranjar estas equações utilizando as Equações (2.20) à (2.23), daí,

$$f_x = J y_\eta f_\xi - J y_\xi f_\eta \implies f_x = J(y_\eta f_\xi - y_\xi f_\eta) \quad (2.29)$$

de modo análogo temos,

$$f_y = J(x_\xi f_\eta - x_\eta f_\xi). \quad (2.30)$$

Novamente, por meio da regra da cadeia, podemos calcular a derivada de segunda ordem da função  $f(\xi, \eta)$ . Em seguida, substituindo os valores das métricas representados pelas Equações (2.20) à (2.23), obtemos as seguintes expressões:

$$\begin{aligned} f_{xx} = J^2 (y_\eta^2 f_{\xi\xi} - 2y_\xi y_\eta f_{\xi\eta} + y_\xi^2 f_{\eta\eta}) + J y_\eta \left[ f_\xi \frac{\partial}{\partial \xi} (\xi_x) + f_\eta \frac{\partial}{\partial \xi} (\eta_x) \right] + \\ - J y_\xi \left[ f_\xi \frac{\partial}{\partial \eta} (\xi_x) + f_\eta \frac{\partial}{\partial \eta} (\eta_x) \right] \end{aligned} \quad (2.31)$$

e

$$\begin{aligned} f_{yy} = J^2 (x_\eta^2 f_{\xi\xi} - 2x_\xi x_\eta f_{\xi\eta} + x_\xi^2 f_{\eta\eta}) + J x_\eta \left[ f_\xi \frac{\partial}{\partial \xi} (\xi_y) + f_\eta \frac{\partial}{\partial \xi} (\eta_y) \right] + \\ - J x_\xi \left[ f_\xi \frac{\partial}{\partial \eta} (\xi_y) + f_\eta \frac{\partial}{\partial \eta} (\eta_y) \right]. \end{aligned} \quad (2.32)$$

Mais detalhes sobre a obtenção destas equações podem ser vistos em (HOFFMANN; CHIANG, 2000).

Ainda resta calcular as derivadas das métricas, neste caso usa-se a regra do quociente para derivadas de primeira ordem, isto é:

$$\begin{aligned}
\frac{\partial}{\partial \xi}(\xi_x) &= \frac{\partial}{\partial \xi}(Jy_\eta) \\
&= \frac{\partial}{\partial \xi} \left( \frac{y_\eta}{x_\xi y_\eta - x_\eta y_\xi} \right) \\
&= J^2 [y_{\xi\eta} (x_\xi y_\eta - x_\eta y_\xi) - y_\eta (y_\eta x_{\xi\xi} + x_\xi y_{\xi\eta} - x_\eta y_{\xi\xi}) - y_\xi x_{\xi\eta}] \\
&= J^2 (x_\xi y_\eta y_{\xi\eta} - x_\eta y_\xi y_{\xi\eta} - y_\eta^2 x_{\xi\xi} - x_\xi y_\eta y_{\xi\eta} + x_\eta y_\eta y_{\xi\xi} + y_\xi y_\eta x_{\xi\eta}) \quad (2.33)
\end{aligned}$$

do mesmo modo:

$$\frac{\partial}{\partial \xi}(\eta_x) = -J^2 (x_\xi y_\eta y_{\xi\xi} - x_\eta y_\xi y_{\xi\xi} - y_\xi y_\eta x_{\xi\xi} - x_\xi y_\xi y_{\xi\eta} + x_\eta y_\xi y_{\xi\xi} + y_\eta^2 x_{\xi\eta}) \quad (2.34)$$

$$\frac{\partial}{\partial \eta}(\xi_x) = J^2 (x_\xi y_\eta y_{\eta\eta} - x_\eta y_\xi y_{\eta\eta} - x_\xi y_\eta y_{\eta\eta} - y_\eta^2 x_{\xi\eta} + y_\xi y_\eta x_{\eta\eta} + x_\eta y_\eta y_{\xi\eta}) \quad (2.35)$$

$$\frac{\partial}{\partial \eta}(\eta_x) = -J^2 (x_\xi y_\eta y_{\xi\eta} - x_\eta y_\xi y_{\xi\eta} - y_\xi x_\xi y_{\eta\eta} - y_\xi y_\eta x_{\xi\eta} + y_\xi^2 x_{\eta\eta} + x_\eta y_\xi y_{\xi\eta}). \quad (2.36)$$

Substituindo as Equações (2.33) à (2.36) na Equação (2.31) e rearranjando os termos, obtêm-se:

$$\begin{aligned}
f_{xx} &= J^2 (y_\eta^2 f_{\xi\xi} - 2y_\xi y_\eta f_{\xi\eta} + y_\xi^2 f_{\eta\eta}) + \\
&+ J^3 (y_\eta^2 y_{\xi\xi} - 2y_\eta y_\xi y_{\xi\eta} + y_\xi^2 y_{\eta\eta}) (x_\eta f_\xi - x_\xi f_\eta) + \\
&+ J^3 (y_\eta^2 x_{\xi\xi} - 2y_\eta y_\xi x_{\xi\eta} + y_\xi^2 x_{\eta\eta}) (y_\xi f_\eta - y_\eta f_\xi)
\end{aligned} \quad (2.37)$$

de maneira análoga, substituindo na Equação 2.32, obtêm-se:

$$\begin{aligned}
f_{yy} &= J^2 (x_\eta^2 f_{\xi\xi} - 2x_\xi x_\eta f_{\xi\eta} + x_\xi^2 f_{\eta\eta}) + \\
&+ J^3 (x_\eta^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi^2 y_{\eta\eta}) (x_\eta f_\xi - x_\xi f_\eta) + \\
&+ J^3 (x_\eta^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi^2 x_{\eta\eta}) (y_\xi f_\eta - y_\eta f_\xi).
\end{aligned} \quad (2.38)$$

Agora considere o Laplaciano,

$$\nabla^2 f = f_{xx} + f_{yy}. \quad (2.39)$$

Substituindo as derivadas duplas representadas pelas Equações (2.37) e (2.38), em seguida colocando os termos semelhantes em evidência, obtêm-se a seguinte expressão:

$$\begin{aligned}
\nabla^2 f &= J^2 [(x_\eta^2 + y_\eta^2) f_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) f_{\xi\eta} + (x_\xi^2 + y_\xi^2) f_{\eta\eta}] + \\
&+ J^3 [(x_\eta^2 + y_\eta^2) y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) y_{\xi\eta} + (x_\xi^2 + y_\xi^2) y_{\eta\eta}] (x_\eta f_\xi - x_\xi f_\eta) + \\
&+ J^3 [(x_\eta^2 + y_\eta^2) x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) x_{\xi\eta} + (x_\xi^2 + y_\xi^2) x_{\eta\eta}] (y_\xi f_\eta - y_\eta f_\xi).
\end{aligned} \quad (2.40)$$

Pode-se definir os seguintes termos:

$$x_\eta^2 + y_\eta^2 = a \quad (2.41)$$

$$x_\xi x_\eta + y_\xi y_\eta = b \quad (2.42)$$

$$x_\xi^2 + y_\xi^2 = c \quad (2.43)$$

Dessa forma, tem-se:

$$\begin{aligned} \nabla^2 f &= J^2 (af_{\xi\xi} - bf_{\xi\eta} + cf_{\eta\eta}) + \\ &+ J^3 [(ay_{\xi\xi} - by_{\xi\eta} + cy_{\eta\eta})] (x_\eta f_\xi - x_\xi f_\eta) + \\ &+ J^3 [(ax_{\xi\xi} - bx_{\xi\eta} + cx_{\eta\eta})] (x_\eta f_\xi - x_\xi f_\eta). \end{aligned} \quad (2.44)$$

Considerando os termos mostrados nas Equações (2.41) à (2.43), pode-se reescrever a Equação (2.44) como:

$$\nabla^2 f = J^2 (af_{\xi\xi} - bf_{\xi\eta} + cf_{\eta\eta} + df_\eta + ef_\xi) \quad (2.45)$$

onde,

$$d = J (y_\xi \alpha - x_\xi \beta) \quad (2.46)$$

$$e = J (x_\eta \beta - y_\eta \alpha) \quad (2.47)$$

e

$$\alpha = ax_{\xi\xi} - 2bx_{\xi\eta} + cx_{\eta\eta} \quad (2.48)$$

$$\beta = ay_{\xi\xi} - 2by_{\xi\eta} + cy_{\eta\eta}. \quad (2.49)$$

A transformação das equações do domínio físico para um transformado é realizada por meio das Equações (2.1) e (2.2). Fazendo  $f = \xi$  na Equação (2.45), tem-se:

$$\nabla^2 \xi = J^2 (a\xi_{\xi\xi} - b\xi_{\xi\eta} + c\xi_{\eta\eta} + d\xi_\eta + e\xi_\xi). \quad (2.50)$$

Conforme foi mencionado anteriormente, utilizaremos a equação de Laplace. Sendo assim, considere o sistema linear elíptico representado pelas Equações (2.25) e (2.26). Levando em conta que,  $\xi_\xi = 1$ ,  $\xi_\eta = 0$ ,  $\xi_{\xi\xi} = 0$ ,  $\xi_{\eta\eta} = 0$  e  $\xi_{\xi\eta} = 0$ , a Equação (2.50) resulta em:

$$J^2 e = 0 \quad (2.51)$$

isto é o mesmo que,

$$J^3 (x_\eta \beta - y_\eta \alpha) = 0. \quad (2.52)$$

Similarmente, para  $f = \eta$ , temos:

$$J^2 d = 0 \quad (2.53)$$

ou

$$J^3(y_\xi\alpha - x_\xi\beta) = 0. \quad (2.54)$$

Uma vez que o Jacobiano não pode ser nulo, temos:

$$x_\eta\beta - y_\eta\alpha = 0 \quad (2.55)$$

$$y_\xi\alpha - x_\xi\beta = 0. \quad (2.56)$$

Eliminando  $\alpha$  nas Equações (2.55) e (2.56), obtemos:

$$\beta(x_\xi y_\eta - x_\eta y_\xi) = 0. \quad (2.57)$$

Como,

$$x_\xi y_\eta - x_\eta y_\xi = J^{-1} \quad (2.58)$$

logo,

$$\beta = 0. \quad (2.59)$$

Substituindo este valor na Equação (2.49), temos:

$$ay_{\xi\xi} - 2by_{\xi\eta} + cy_{\eta\eta} = 0. \quad (2.60)$$

Do mesmo modo que obtivemos  $\beta = 0$  também podemos encontrar  $\alpha = 0$ . Assim, da Equação (2.48), temos:

$$ax_{\xi\xi} - 2bx_{\xi\eta} + cx_{\eta\eta} = 0. \quad (2.61)$$

A solução das Equações (2.60) e (2.61) possibilita determinar as coordenadas  $(x, y)$  dos pontos  $(\xi, \eta)$ , quando as condições de contorno são conhecidas. Assim, parte-se das coordenadas  $(x, y)$  de todos os pontos  $(\xi, \eta)$  definidos para a fronteira (pontos externos) para calcular as coordenadas  $x$  e  $y$  dos pontos internos da malha (SILVA, 2007).

## 3 Modelagem numérica e computacional

Para obter a solução numérica de modelos matemáticos, devemos manipular o modelo original de tal maneira que seja possível resolvê-lo por meio de cálculos computacionais (MANGANI, 2008). Para modelos que envolvem equações diferenciais parciais, a tarefa fundamental dos métodos numéricos, é a aproximação das derivadas parciais por expressões algébricas (HOFFMANN; CHIANG, 2000). O próximo passo é calcular a solução do sistema resultante, que geralmente envolve equações com milhares ou até milhões de incógnitas (quanto maior o número de incógnitas, mais precisa será a solução numérica). Tal processo pode ser realizado utilizando um computador, explorando sua precisão e velocidade de processamento.

### 3.1 Discretização das Equações

No Capítulo 2 já foi apresentado o modelo matemático, Equações (2.60) e (2.61). O que deve ser feito a partir de agora é a discretização destas equações (MANGANI, 2008).

No presente trabalho foi utilizado o Método das Diferenças Finitas (MDF). Neste método, as derivadas das equações diferenciais parciais são representadas em pontos do domínio por expressões algébricas de diferenças, obtidas realizando expansões em séries de Taylor das variáveis de solução em vários pontos vizinhos do ponto de avaliação (THOMPSON; SONI; WEATHERILL, 1999).

Para fins ilustrativos considere um ponto interno da malha no domínio transformado, representado por  $P$ , bem como os seus vizinhos (N, S, E, W, NW, NE, SW, e SE), que serão utilizados para definir as derivadas envolvidas. Tais pontos podem ser observados na Figura 6 (SILVA, 2007).

Chamando de  $\Phi$  a variável dependente que representa ao mesmo tempo as coordenadas  $x$  e  $y$  dos pontos da malha a ser gerada, as Equações (2.60) e (2.61) podem ser escritas de uma forma geral, como sendo:

$$a\Phi_{\xi\xi} - 2b\Phi_{\xi\eta} + c\Phi_{\eta\eta} = 0. \quad (3.1)$$

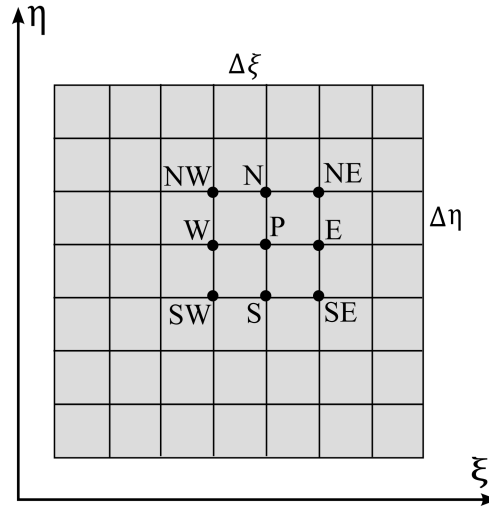


Figura 6 – Ponto interno P e seus vizinhos (N, S, E, W, NW, NE, SW, SE) em uma malha bidimensional no domínio transformado, Fonte: (SILVA, 2007).

Para determinar uma expressão algébrica para  $\Phi_{\xi\xi}$ , utilizando expansão em série de Taylor para uma função e desprezando os termos de ordem superior a 2, segundo sugere Silva (2007), tem-se:

$$\Phi(\xi + \Delta\xi) = \Phi(\xi) + \Phi_{\xi}\Delta\xi + \frac{1}{2}\Phi_{\xi\xi}(\Delta\xi)^2 \quad (3.2)$$

e

$$\Phi(\xi - \Delta\xi) = \Phi(\xi) - \Phi_{\xi}\Delta\xi + \frac{1}{2}\Phi_{\xi\xi}(\Delta\xi)^2. \quad (3.3)$$

Somando as Equações (3.2) e (3.3), obte-se:

$$\Phi_{\xi\xi} = \frac{\Phi(\xi + \Delta\xi) + \Phi(\xi - \Delta\xi) - 2\Phi(\xi)}{(\Delta\xi)^2}. \quad (3.4)$$

Observando a Figura 6, pode-se escrever  $\Phi(\xi + \Delta\xi) = \Phi_E$ ,  $\Phi(\xi - \Delta\xi) = \Phi_W$  e  $\Phi(\xi) = \Phi_P$ , assim sendo, a Equação (3.4) resulta em:

$$\Phi_{\xi\xi} = \frac{\Phi_E + \Phi_W - 2\Phi_P}{(\Delta\xi)^2}. \quad (3.5)$$

Analogamente, pode-se obter:

$$\Phi_{\eta\eta} = \frac{\Phi_N + \Phi_S - 2\Phi_P}{(\Delta\xi)^2}. \quad (3.6)$$

Subtraindo a Equação (3.3) da Equação (3.2), tem-se uma expressão para o termo  $\Phi_{\xi}$ :

$$\Phi_{\xi} = \frac{\Phi(\xi + \Delta\xi) - \Phi(\xi - \Delta\xi)}{2\Delta\xi}. \quad (3.7)$$

Considerando a disposição dos pontos notados na Figura 6, pode-se reescrever a equação anterior como sendo:

$$\Phi_\xi = \frac{\Phi_E - \Phi_W}{2\Delta\xi}. \quad (3.8)$$

Analogamente para  $\Phi_\eta$ , temos:

$$\Phi_\eta = \frac{\Phi_N - \Phi_S}{2\Delta\xi}. \quad (3.9)$$

Como as funções são contínuas,  $\Phi_{\xi\eta}$  pode ser escrito de acordo com a Equação (3.9), ou seja,

$$(\Phi_\xi)_\eta = \frac{(\Phi_\xi)_N - (\Phi_\xi)_S}{2\Delta\xi}. \quad (3.10)$$

Substituindo o valor de  $\Phi_\xi$ , expresso pela Equação (3.8), a Equação (3.10) resulta em:

$$\Phi_{\xi\eta} = \frac{\Phi_{NE} + \Phi_{SW} - \Phi_{SE} - \Phi_{NW}}{4\Delta\xi\Delta\eta}. \quad (3.11)$$

Agora, substituindo as Equações (3.5), (3.6) e (3.11), na Equação (3.1), tem-se:

$$\begin{aligned} a \left( \frac{\Phi_E + \Phi_W - 2\Phi_P}{(\Delta\xi)^2} \right) - 2b \left( \frac{\Phi_{NE} + \Phi_{SW} - \Phi_{SE} - \Phi_{NW}}{4\Delta\xi\Delta\eta} \right) + \\ + c \left( \frac{\Phi_N + \Phi_S - 2\Phi_P}{(\Delta\eta)^2} \right) = 0. \end{aligned} \quad (3.12)$$

Fazendo as devidas operações matemáticas, em seguida colocando as variáveis dependentes em evidência, obte-se a seguinte expressão:

$$\begin{aligned} \Phi_P \left[ -\frac{2a}{(\Delta\xi)^2} - \frac{2c}{(\Delta\eta)^2} \right] + \Phi_E \left[ \frac{a}{(\Delta\xi)^2} \right] + \Phi_W \left[ \frac{a}{(\Delta\xi)^2} \right] + \\ + \Phi_N \left[ \frac{c}{(\Delta\eta)^2} \right] + \Phi_S \left[ \frac{c}{(\Delta\eta)^2} \right] + \Phi_{NE} \left( \frac{-2b}{4\Delta\xi\Delta\eta} \right) + \\ + \Phi_{NW} \left( \frac{2b}{4\Delta\xi\Delta\eta} \right) + \Phi_{SE} \left( \frac{2b}{4\Delta\xi\Delta\eta} \right) + \Phi_{SW} \left( \frac{-2b}{4\Delta\xi\Delta\eta} \right) = 0. \end{aligned} \quad (3.13)$$

Pode-se definir os coeficientes dos termos dependentes da seguinte maneira:

$$A_P = 2 \left[ \frac{a}{(\Delta\xi)^2} + \frac{c}{(\Delta\eta)^2} \right] \quad (3.14)$$

$$A_E = \left[ \frac{a}{(\Delta\xi)^2} \right] \quad (3.15)$$

$$A_W = \left[ \frac{a}{(\Delta\xi)^2} \right] \quad (3.16)$$

$$A_N = \left[ \frac{c}{(\Delta\eta)^2} \right] \quad (3.17)$$

$$A_S = \left[ \frac{c}{(\Delta\eta)^2} \right] \quad (3.18)$$

$$A_{NE} = \left( \frac{-2b}{4\Delta\xi\Delta\eta} \right) \quad (3.19)$$

$$A_{NW} = \left( \frac{2b}{4\Delta\xi\Delta\eta} \right) \quad (3.20)$$

$$A_{SE} = \left( \frac{2b}{4\Delta\xi\Delta\eta} \right) \quad (3.21)$$

$$A_{SW} = \left( \frac{-2b}{4\Delta\xi\Delta\eta} \right). \quad (3.22)$$

Feito isto, a Equação (3.13), resulta em:

$$\begin{aligned} A_P\Phi_P &= A_E\Phi_E + A_W\Phi_W + A_N\Phi_N + A_S\Phi_S + \\ &+ A_{NE}\Phi_{NE} + A_{NW}\Phi_{NW} + A_{SE}\Phi_{SE} + A_{SW}\Phi_{SW}. \end{aligned} \quad (3.23)$$

Nesta expressão os coeficientes dependem dos termos  $a$ ,  $b$  e  $c$ , que ainda envolvem derivadas, como pode ser visto nas Equações (2.41), (2.42) e (2.43). Assim, é necessário determinar expressões aproximadas para as derivadas  $x_\xi$ ,  $y_\xi$ ,  $x_\eta$  e  $y_\eta$ .

Tomando como referência a Figura 6 para determinar as expressões aproximadas para estas derivadas, pode-se obter as seguintes expressões:

$$x_\xi = \frac{x_E - x_W}{2\Delta\xi} \quad (3.24)$$

$$y_\xi = \frac{y_E - y_W}{2\Delta\xi} \quad (3.25)$$

$$x_\eta = \frac{x_N - x_S}{2\Delta\eta} \quad (3.26)$$

$$y_\eta = \frac{y_N - y_S}{2\Delta\eta}. \quad (3.27)$$



Cada ponto da malha é representado pela Equação algébrica (3.23). O conjunto dessas equações pode ser representado por uma equação matricial do tipo:

$$[A][\Phi] = [B]. \quad (3.28)$$

A solução do sistema linear representado pela Equação (3.28), nos fornecerá as posições dos pontos internos da malha baseados nas posições dos pontos das fronteiras (bordas) da geometria (COUTINHO, 2013).

## 3.2 Solução do Sistema Linear

Os métodos para solução de sistemas lineares são comumente classificados em dois principais grupos: métodos diretos e métodos iterativos. Os métodos diretos lidam com matrizes utilizando as regras da álgebra linear, reduzindo-as de tal forma que o cálculo da solução final pode ser feito com facilidade. Métodos iterativos partem de uma aproximação inicial qualquer, após esta aproximação (chute inicial), a solução final do sistema é obtida utilizando-se uma técnica de refinamento sucessivo até que se atinja o resultado desejado, tendo um custo computacional muito menor (MANGANI, 2008).

A matriz resultante da metodologia utilizada no presente trabalho é esparsa<sup>1</sup>, pois a solução para um ponto da malha depende apenas dos seus vizinhos. Para melhor compreensão, considere uma malha com 2x2 volumes, vista na Figura 7.

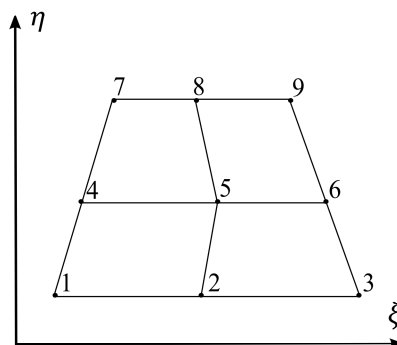


Figura 7 – Exemplo de malha com 2x2 volumes.

Considerando que  $\Phi$  representa as coordenadas  $x$  e  $y$  de cada ponto, o sistema linear a ser resolvido para a geração da malha representada na Figura 7 tem a seguinte

<sup>1</sup> Uma matriz é dita esparsa quando a maioria de seus elementos são nulos (CARMO, 2005).

forma:

$$\begin{bmatrix} A_{P_0} & A_{E_0} & 0 & A_{N_0} & A_{NE_0} & 0 & 0 & 0 & 0 \\ A_{W_1} & A_{P_1} & A_{E_1} & A_{NW_1} & A_{N_1} & A_{NE_1} & 0 & 0 & 0 \\ 0 & A_{W_2} & A_{P_2} & 0 & A_{NW_2} & A_{N_2} & 0 & 0 & 0 \\ A_{S_3} & A_{SE_3} & 0 & A_{P_3} & A_{E_3} & 0 & A_{N_3} & A_{NE_3} & 0 \\ A_{SW_4} & A_{S_4} & A_{SE_4} & A_{W_4} & A_{P_4} & A_{E_4} & A_{NW_4} & A_{N_4} & A_{E_4} \\ 0 & A_{SW_5} & A_{S_5} & 0 & A_{W_5} & A_{P_5} & 0 & A_{NW_5} & A_{N_5} \\ 0 & 0 & 0 & A_{S_6} & A_{SE_6} & 0 & A_{P_6} & A_{E_6} & 0 \\ 0 & 0 & 0 & A_{SW_7} & A_{S_7} & A_{SE_7} & A_{W_7} & A_{P_7} & A_{E_7} \\ 0 & 0 & 0 & 0 & A_{SW_8} & A_{S_8} & 0 & A_{W_8} & A_{P_8} \end{bmatrix} \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.29)$$

Percebe-se que a matriz de coeficientes resultante possui muitos elementos nulos.

### 3.2.1 Método iterativo de Gauss-Seidel

Sistemas lineares que envolvem matrizes esparsas devem, preferencialmente, ser resolvidos via métodos iterativos (ARAÚJO, 2013). No presente trabalho o método iterativo de Gauss-Seidel foi utilizado. Conforme descreve Quadros e Bortoli (2009), neste método, um sistema linear qualquer  $Ax = b$ , pode ser escrito de uma forma equivalente separando os elementos da diagonal, isto é,

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} = \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)}) \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}} (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)}) \end{cases} \quad (3.30)$$

Dessa forma, quando estamos calculando o valor para  $x_j^{(k+1)}$  fazemos uso de todos os valores  $x_1^{(k+1)}, \dots, x_{j-1}^{(k+1)}$  já calculados e o valores de  $x_{j+1}^{(k)}, \dots, x_n^{(k)}$  restantes até que a condição de convergência seja satisfeita. Para entender melhor como funciona este método, considere o seguinte exemplo extraído de (QUADROS; BORTOLI, 2009):

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ x_1 + 10x_2 + x_3 = 12 \\ x_1 + x_2 + 10x_3 = 12 \end{cases} \quad (3.31)$$

É fácil perceber que a solução é  $X = [1 \ 1 \ 1]$ . Para resolver pelo método de Gauss-Seidel, inicialmente reescrevemos o sistema da seguinte forma:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{10} (12 - x_2^{(k)} - x_3^{(k)}) \\ x_2^{(k+1)} = \frac{1}{10} (12 - x_1^{(k+1)} - x_3^{(k)}) \\ x_3^{(k+1)} = \frac{1}{10} (12 - x_1^{(k+1)} - x_2^{(k+1)}) \end{cases} \quad (3.32)$$

Em seguida, iniciando o processo iterativo de Gauss-Seidel com o chute inicial  $X^0 = [0 \ 0 \ 0]$ , teremos como resultados, em iterações sucessivas, as aproximações apresentadas na Tabela 3.

Tabela 3 – Aproximações por Gauss-Seidel para a solução do sistema, Fonte: (QUADROS; BORTOLI, 2009).

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	1,2	1,8	0,972
2	0,9948	1,0033	1,0002
3	0,9996	1,0000	1,0000

Como podemos observar na Tabela 3, o método iterativo de Gauss-Seidel produz uma sequência de aproximações que convergem para solução real à medida que as iterações avançam.

### 3.2.2 Critério de parada

Para qualquer método iterativo, é fundamental estabelecer um critério de parada (convergência da solução). Bittencourt e Feijóo (1997), afirmam que um bom critério de parada deve possuir as seguintes características: identificar quando o erro  $\varepsilon$  é suficientemente pequeno; parar se o erro não está decrescendo ou se taxa de decaimento é muito baixa; limitar a quantidade máxima de iterações.

Visando a precisão dos resultados, foi usada a condição de parada  $\varepsilon$ , de forma que o erro relativo entre a solução calculada  $X^n$  menos a solução calculada anteriormente  $X^{n-1}$ , em valor absoluto, seja no máximo  $10^{-3}$ , isto é:

$$\left| \Phi^n - \Phi^{n-1} \right| = 10^{-3} \quad (3.33)$$

onde,  $\Phi$  representa tanto a coordenada  $x$  quanto a coordenada  $y$ .

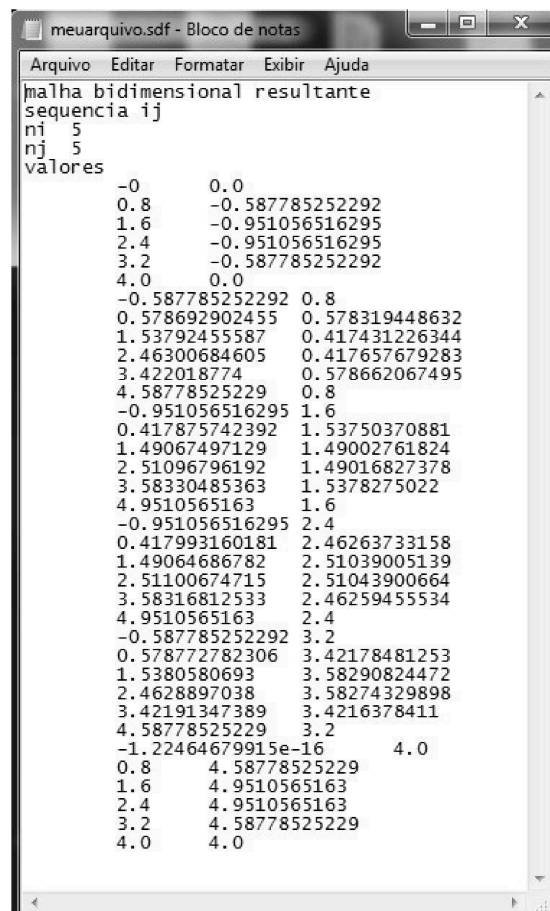
## 3.3 Implementação do código computacional

### 3.3.1 A linguagem Python

A linguagem de programação escolhida para implementação do código computacional foi Python, por apresentar algumas características que a tornam ao mesmo tempo poderosa e fácil de aprender. Possui um sintaxe compacta e pode ser combinada com outras linguagens, como Fortran, C e C++, amplamente utilizadas para cálculos científicos (LANGTANGEN, 2009). Além de ser gratuita, Python está disponível para os mais diversos sistemas operacionais.

### 3.3.2 A interface gráfica

No presente trabalho, solucionamos numericamente o sistema de equações lineares representado pela Equação (3.28), resultando em uma grande quantidade de algarismos. A Figura 8 mostra um exemplo de um arquivo de saída com coordenadas numéricas de pontos referentes a uma malha com 5x5 volumes, calculados pelo código computacional desenvolvido. Sem uma ferramenta de visualização gráfica, seria uma tarefa bastante difícil interpretar os resultados obtidos.



```
meuarquivo.sdf - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
malha bidimensional resultante
sequencia ij
ni 5
nj 5
valores
-0.0 0.0
0.8 -0.587785252292
1.6 -0.951056516295
2.4 -0.951056516295
3.2 -0.587785252292
4.0 0.0
-0.587785252292 0.8
0.578692902455 0.578319448632
1.53792455587 0.417431226344
2.46300684605 0.417657679283
3.422018774 0.578662067495
4.58778525229 0.8
-0.951056516295 1.6
0.417875742392 1.53750370881
1.49067497129 1.49002761824
2.51096796192 1.49016827378
3.58330485363 1.5378275022
4.9510565163 1.6
-0.951056516295 2.4
0.417993160181 2.46263733158
1.49064686782 2.51039005139
2.51100674715 2.51043900664
3.58316812533 2.46259455534
4.9510565163 2.4
-0.587785252292 3.2
0.578772782306 3.42178481253
1.5380580693 3.58290824472
2.4628897038 3.58274329898
3.42191347389 3.4216378411
4.58778525229 3.2
-1.22464679915e-16 4.0
0.8 4.58778525229
1.6 4.9510565163
2.4 4.9510565163
3.2 4.58778525229
4.0 4.0
```

Figura 8 – Arquivo de saída contendo as coordenadas dos pontos de uma malha com 5x5 volumes.

No trabalho de Polo (2008) pode-se encontrar detalhes sobre as principais bibliotecas (PyGtk, PyQt, Tkinter e wxPython) para desenvolvimento de interfaces gráficas com a linguagem Python. O autor ainda afirma que cada uma destas ferramentas tem vantagens e desvantagens, ficando a cargo do usuário escolher qual delas é melhor para suas necessidades particulares.

No presente trabalho, optou-se pelo uso da biblioteca Tkinter por ser relativamente fácil de aprender, por ser uma biblioteca de código aberto e porque é compatível com os principais sistemas operacionais (Windows, Linux e MAC). O desenvolvimento com

Tkinter é rápido e não exige instalações extras uma vez que a biblioteca faz parte da instalação padrão do Python (LUTZ, 2010).

### 3.3.3 Fluxograma

O algoritmo computacional que representa a seqüência de passos executados para a geração de uma malha usando a metodologia apresentada no presente trabalho, pode ser visto na Figura 9.

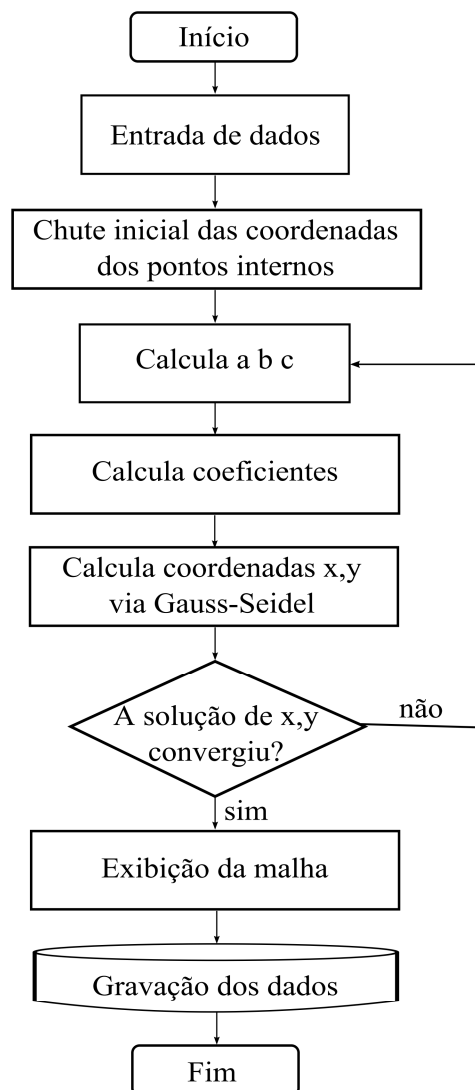


Figura 9 – Fluxograma do código computacional.

## 4 Resultados e discussões

### 4.1 Gerador de malha

Para demonstrar a eficácia do programa desenvolvido e da modelagem apresentada na geração de malhas estruturadas generalizadas, foram escolhidas três geometrias distintas com fronteiras irregulares, todas mostradas na Figura 10. A cor verde representa a fronteira norte (N), azul é a fronteira leste (E), preta representa a borda localizada ao sul (S) e, por fim, a cor vermelha indica a fronteira oeste (W).

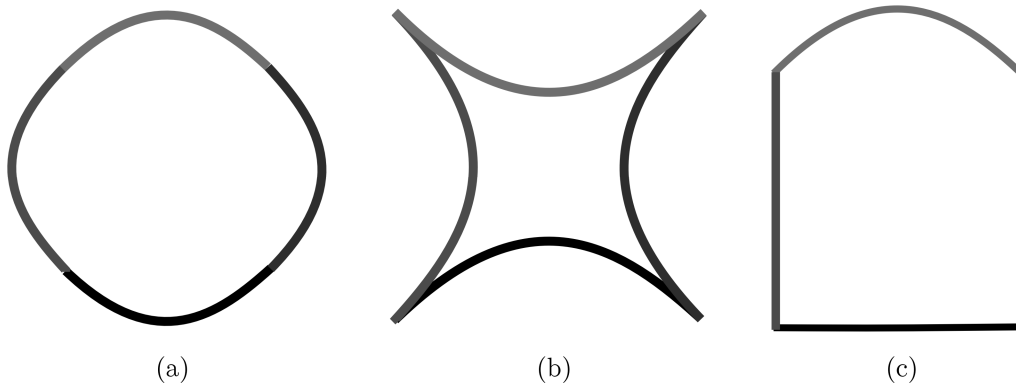


Figura 10 – Geometrias com fronteiras irregulares.

Para cada uma das geometrias foram geradas três malhas: com 5x5 (25 volumes), 10x10 (100 volumes) e 15x15 (225 volumes), mostradas nas Figuras 11, 12 e 13, respectivamente.

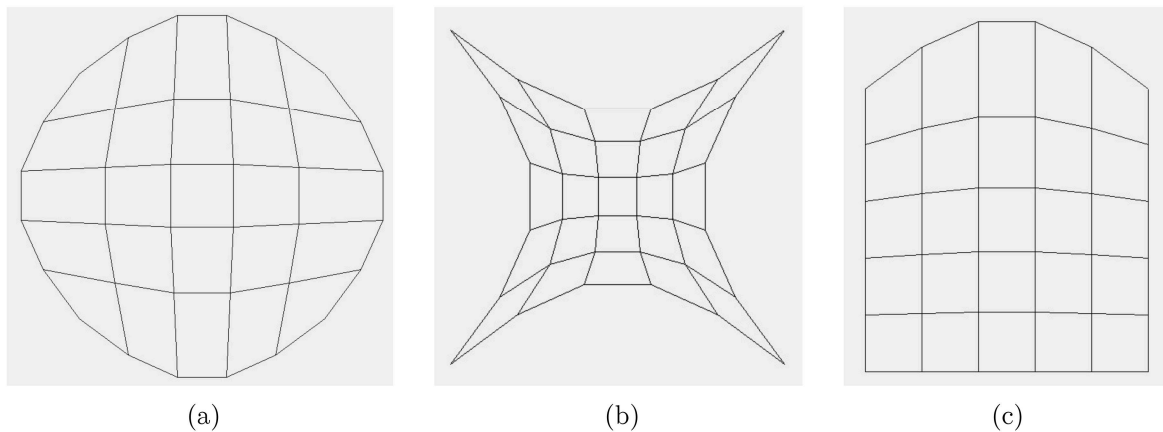


Figura 11 – Malhas com 5x5 volumes para as três geometrias estudadas.

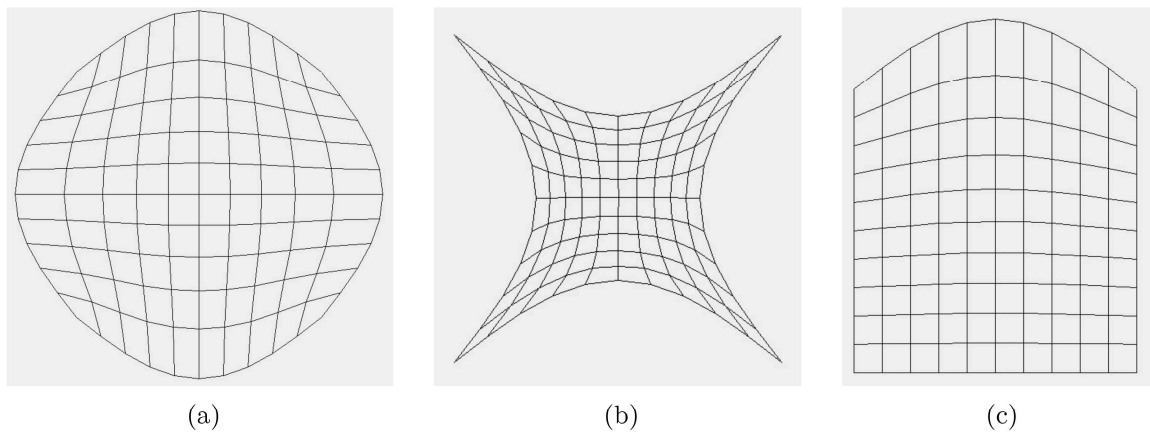


Figura 12 – Malhas com 10x10 volumes para as três geometrias estudadas.

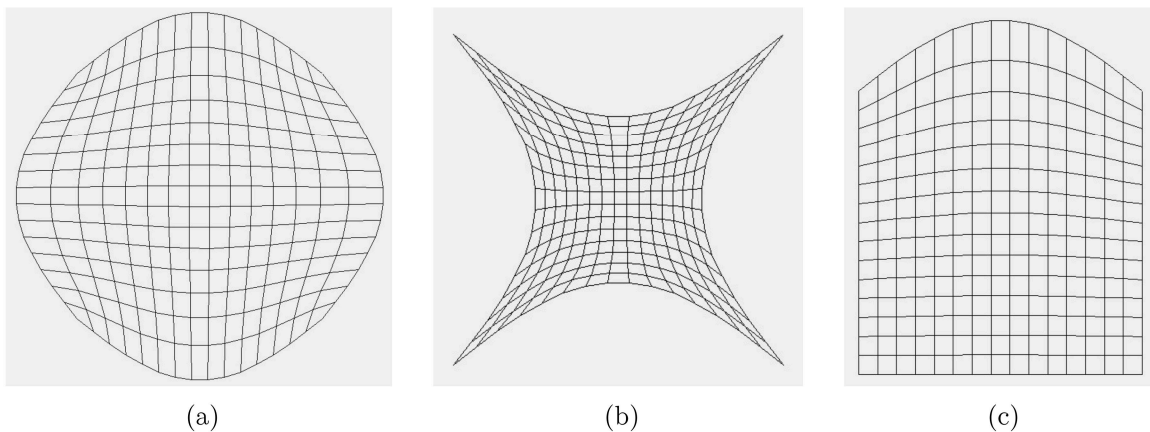


Figura 13 – Malhas com 15x15 volumes para as três geometrias estudadas.

As malhas apresentadas nas Figuras 11, 12 e 13, foram geradas com um critério de parada de  $\varepsilon = 1 \times 10^{-3}$ . Uma análise desse valor na qualidade da malha será realizada na seção 4.1.1.

As malhas mais refinadas mostradas na Figura 13, quando utilizadas para solução de problemas físicos, oferecem mais precisão na obtenção dos resultados. O domínio com mais células tende a se aproximar mais de uma região contínua. Para gerar malhas com mais volumes é necessário mais tempo de execução do gerador.

Com o objetivo de analisar como a quantidade de volumes da malha influencia no tempo de geração, foi plotado o gráfico mostrado na Figura 14, obtido a partir de quatro malhas para a geometria circular com: 10x10, 15x15, 20x20 e 25x25 volumes. Pode-se notar que o tempo de execução cresce exponencialmente em função da quantidade de volumes, uma vez que o sistema linear resultante torna-se maior para malhas mais refinadas, com mais equações e mais incógnitas. Testes realizados indicaram este mesmo comportamento para as demais geometrias.

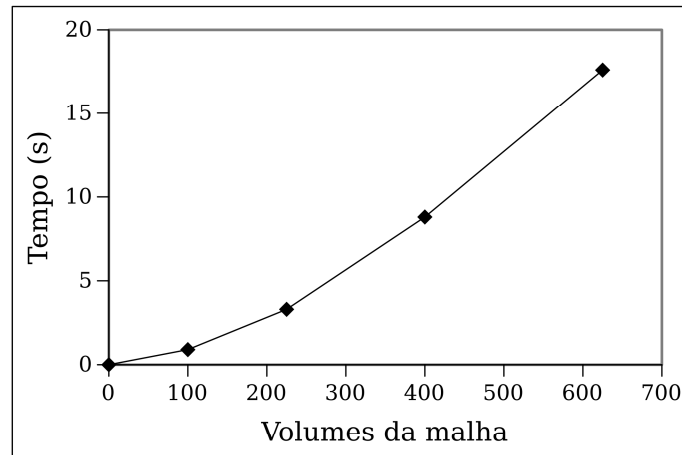


Figura 14 – Tempo de execução em função da quantidade de volumes.

Os tempos de execução mostrados na Figura 14 foram medidos em um computador laptop com processador Intel Celeron 570 de 2,27 GHz.

Todas as malhas obtidas com o gerador desenvolvido no presente trabalho podem ser, além de vistas na janela de visualização, armazenadas em arquivos de texto. O formato escolhido para o arquivo de saída (sdf) permite que as malhas sejam usadas no aplicativo de engenharia *Transcal 1.1*. O formato sdf é o tipo de arquivo padrão deste software. O *Transcal 1.1*, foi desenvolvido pelo Laboratório de Simulação Numérica em Mecânica dos Fluidos e Transferência de Calor da Universidade Federal de Santa Catarina para o ensino de conceitos de condução de calor através de simulação e visualização (UFSC - SINMEC, 2014).

A Figura 15 mostra a visualização no *Transcal 1.1* das malhas com 15x15 volumes geradas neste trabalho. Essa compatibilidade indica que malhas obtidas com o gerador desenvolvido no presente trabalho podem ser usadas perfeitamente para simulações de problemas de condução de calor em corpos bidimensionais no *Transcal 1.1*.

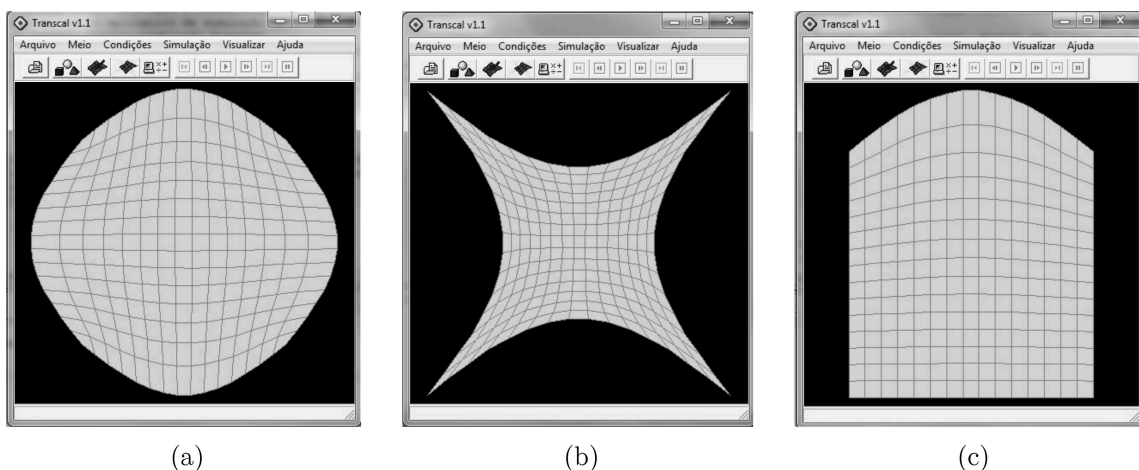


Figura 15 – Visualização das malhas geradas no software *Transcal 1.1*.



### 4.1.1 A influência do critério de parada na qualidade da solução

Para analisar a influência da tolerância na qualidade da malha gerada, foram geradas malhas 15x15 volumes para a geometria circular mostrada Figura 10 (a) com três critérios de parada:  $5 \times 10^{-1}$ ,  $5 \times 10^{-2}$  e  $1 \times 10^{-3}$ . Os resultados obtidos podem ser vistos na Figura 16.

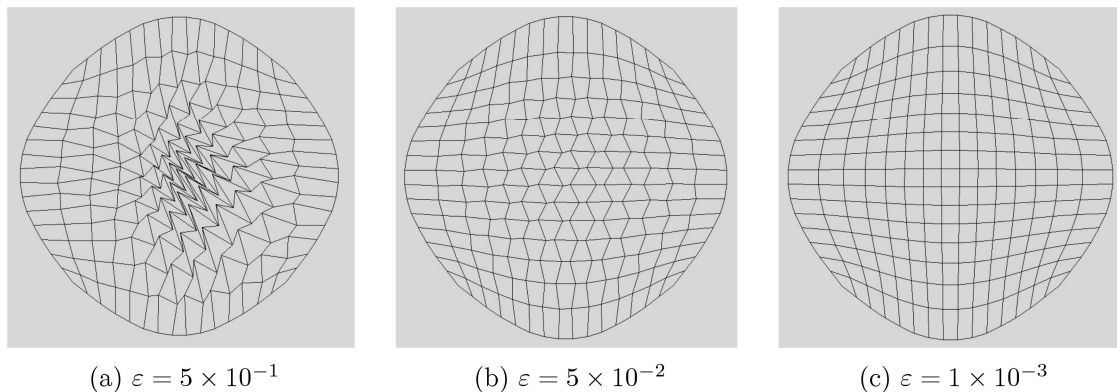


Figura 16 – Malha circular com 15x15 volumes com diversos valores de parada.

Observando a Figura 16, podemos perceber a influência do critério de convergência no cálculo das coordenadas dos pontos internos da malha. A Figura 16 (c) apresenta uma malha com uma distribuição uniforme das células em todo o domínio, diferente do que é visto nas demais malhas, onde os pontos centrais aparecem ainda desordenados. Um maior valor indica que a solução final obtida está mais distante da solução real do problema.

O gráfico da Figura 17 mostra o comportamento do tempo de execução como uma função do critério de parada do método iterativo. Foram testadas para elaboração do gráfico as seguintes tolerâncias:  $\varepsilon = 10^{-3}$ ,  $\varepsilon = 10^{-5}$ ,  $\varepsilon = 10^{-7}$  e  $\varepsilon = 10^{-9}$ , representadas em escala logarítmica.

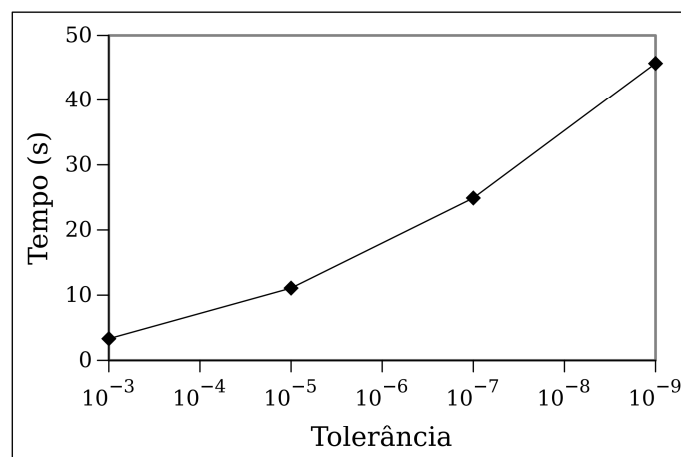


Figura 17 – Tempo de execução em função do critério de parada.

Como pode ser notado, o comportamento do crescimento do tempo é exponencial. O tempo de solução para  $\varepsilon = 10^{-3}$  é cerca de dez vezes menor do que quando se usa uma tolerância de  $\varepsilon = 10^{-9}$ . O uso de um critério de parada muito baixo aumenta bastante o tempo de execução do programa para a geração da malha pois mais iterações são realizadas. Todavia, os resultados nem sempre melhoram com a mesma intensidade.

Malhas obtidas com  $\varepsilon = 10^{-3}$  e  $\varepsilon = 10^{-9}$  são vistas na Figura 18. Esta figura mostra que a diferença na posição dos pontos da malha é mínima, quase imperceptível. Assim, pode-se concluir que com a tolerância  $\varepsilon = 10^{-3}$ , os resultados são satisfatórios, e obtidos com um tempo de processamento bem menor.

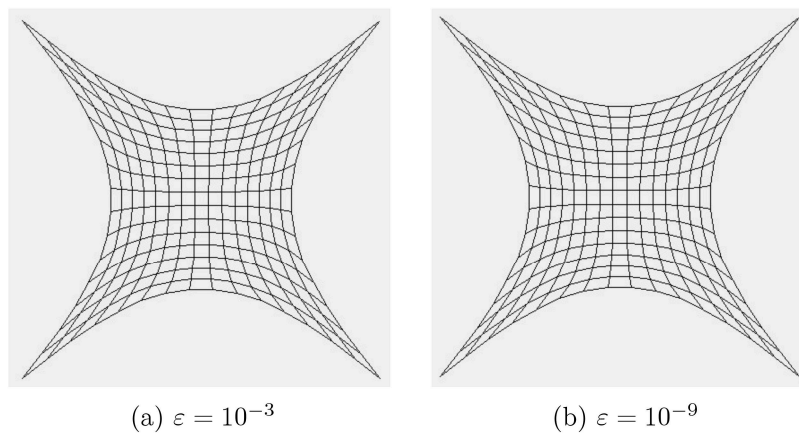


Figura 18 – Malha com 15x15 volumes para critérios de parada diferentes.

### 4.1.2 A evolução da solução no processo iterativo

No presente trabalho são atribuídos valores aleatórios para as coordenadas  $x$  e  $y$  dos pontos internos no início do processo iterativo. Uma interessante funcionalidade para a janela de visualização é permitir observar a posição inicial dos pontos da malha e acompanhar a sua evolução no decorrer das iterações até que a malha seja gerada, conforme mostra a Figura 19 para a geometria circular.

Como todos os pontos são unidos por linhas que definem a malha, a Figura 19(a) que representa o chute inicial (cujos pontos internos possuem coordenadas aleatórias) mostra apenas um emaranhado de linhas. À medida que as iterações avançam, os pontos internos tendem a formar a malha gradativamente, pois se aproximam cada vez mais da solução das Equações (2.60) e (2.61).

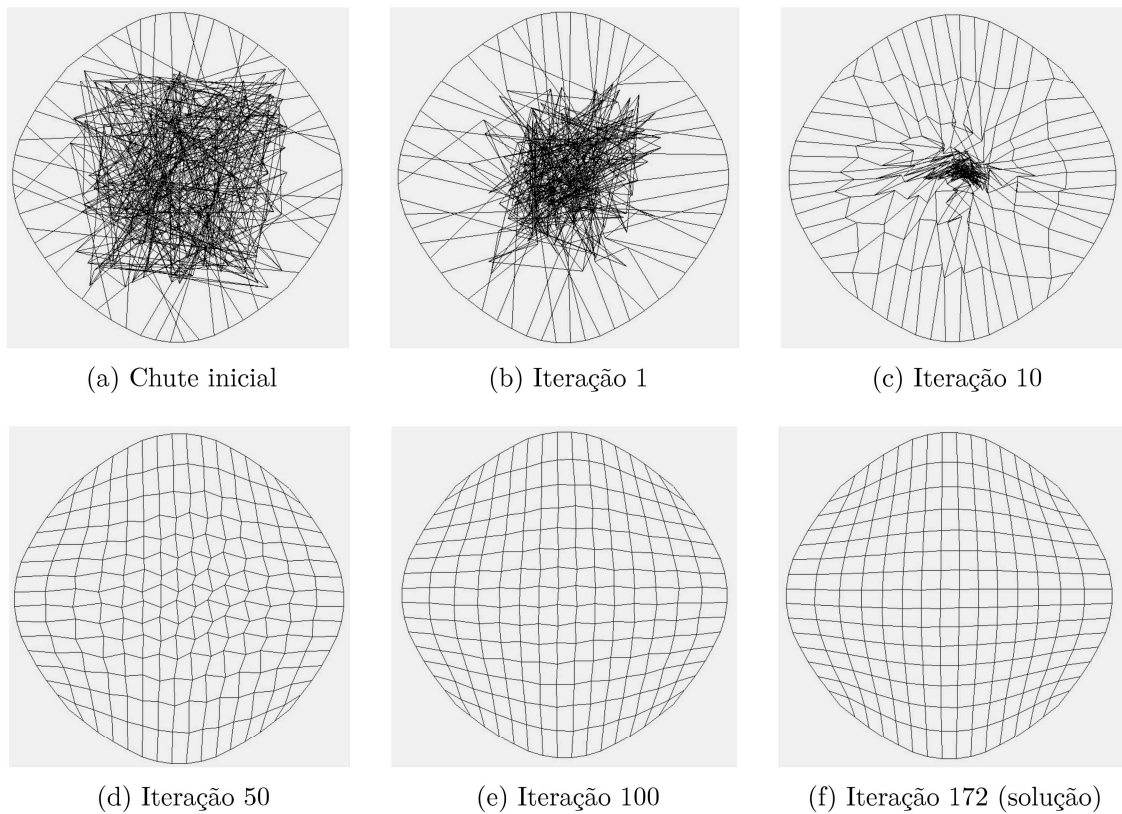


Figura 19 – Evolução da solução no decorrer do processo iterativo.

## 4.2 O programa desenvolvido

Para tornar o uso do programa mais amigável para o usuário, foi desenvolvida uma interface gráfica<sup>1</sup> com duas janelas. A primeira delas, Figura 20, para a definição das dimensões da malha (número de volumes na direção  $x$  e  $y$ ), do critério de parada (tolerância) para o cálculo da solução e para a escolha de uma das geometrias apresentadas neste trabalho. Nessa janela, há dois botões: *Gerar a malha* para iniciar o processo iterativo e *Salvar as coordenadas* para armazenar a malha em um arquivo de saída (sdf). A outra janela, exibida na Figura 21, aparece sempre que uma malha é gerada, mostrando o resultado final.

<sup>1</sup> Uma área onde se realiza interação entre usuário e programa (NASCIMENTO; AMARAL, 2010).

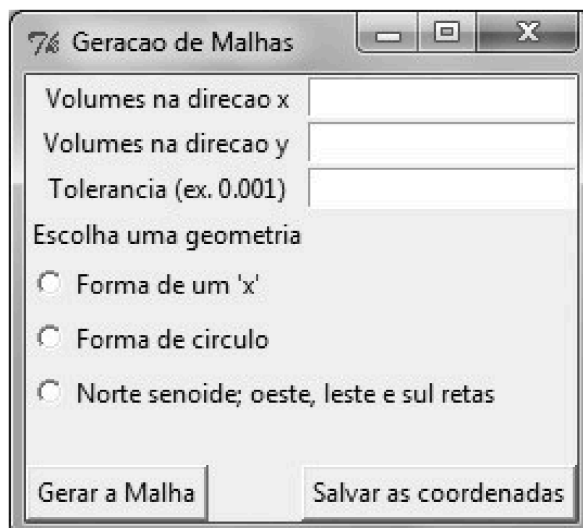


Figura 20 – Janela para entrada de dados.

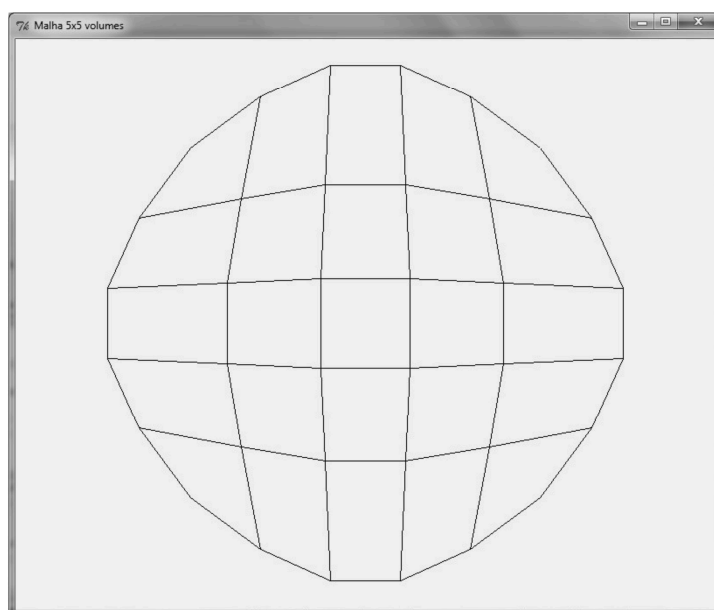


Figura 21 – Janela para visualização da malha.

Comparando as Figuras 8 e 21, pode-se perceber que a interpretação dos resultados (disposição dos pontos da malha) pelo usuário pode ser feita com muito mais comodidade por meio da janela de visualização.

## Considerações finais

O presente trabalho demonstrou que a solução numérica de um sistema de equações diferenciais parciais elípticas pode ser usada para gerar malhas estruturadas bidimensionais coincidentes com as fronteiras. A partir da metodologia numérica, foi desenvolvido um programa computacional em linguagem Python. Para obtenção dos resultados foram escolhidas três geometrias distintas com bordas irregulares. Malhas com diversas quantidades de volumes foram geradas com sucesso para cada uma das geometrias testadas. O gerador desenvolvido permitiu a visualização da malha em uma janela gráfica bem como o armazenamento da malha em arquivo texto de formato sdf, compatível com o software acadêmico de engenharia *Transcal 1.1*.

A realização deste trabalho permitiu observar a aplicação de forma bastante prática de diversos conceitos matemáticos tais como: matrizes, determinantes, matriz inversa, derivadas simples e parciais, regra da cadeia, expansão em série de Taylor, funções, funções inversas, sistemas lineares, etc.

A solução do sistema linear resultante do procedimento de discretização e aproximação das derivadas parciais por diferenças finitas foi obtida com a utilização do método iterativo de Gauss-Seidel. O método numérico mostrou-se bastante rápido e preciso para a obtenção da solução.

Foi observado que o critério de parada escolhido com valor  $10^{-3}$  produz resultados satisfatórios. Para valores maiores, exigem mais tempo de processamento, porém sem ganhos expressivos na qualidade da malha.

Outro fator que influencia bastante no tempo de processamento do programa é a quantidade de volumes da malha, devido a maior quantidade de equações e incógnitas a serem calculadas. Com mais volumes região discretizadas aproxima-se mais de uma região contínua.

## Referências

- ARAÚJO, A. L. M. *Matemática Computacional: Notas de apoio às aulas de matemática computacional do mestrado integrado em engenharia electrotécnica e de computadores*. 169 p. — Faculdade de Ciência e Tecnologia da Universidade de Coimbra, Coimbra, 2013. Citado na página 35.
- BITTENCOURT, M. L.; FEIJÓO, R. A. Análise comparativa de métodos diretos e iterativos para a solução de sistema de equações. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, v. 13, n. 2, p. 123–148, 1997. Citado na página 36.
- CARMO, F. C. D. *Análise da Influência de Algoritmos de Reordenação de Matrizes Esparsas no Desempenho do Método CCCG( $\eta$ )*. 150 f. Dissertação (Mestrado em em Ciência da Computação) — Universidade Federal de Minas Gerais, Belo Horizonte, 2005. Citado na página 34.
- CIRILO, E. R.; BORTOLI, A. L. D. Geração da malha da traquéia e dos tubos bronquiais por splines cúbico. *Semina: Ciências Exatas e Tecnológicas*, v. 27, n. 2, p. 147–155, 2006. Citado 2 vezes nas páginas 17 e 20.
- COUTINHO, B. G. *Modelagem Matemática em Coordenadas Generalizadas e Desenvolvimento de Simulador Computacional para Aplicação em Processos de Moldagem por Transferência de Resina*. 199 f. Tese (Doutorado em Engenharia de Processos) — Universidade Federal de Campina Grande, Campina Grande, 2013. Citado 6 vezes nas páginas 17, 19, 20, 21, 22 e 34.
- FORTUNA, A. *Técnicas Computacionais para Dinâmica dos Fluidos: Conceitos básicos e aplicações*. [S.l.]: Edusp, 2000. Citado 3 vezes nas páginas 16, 17 e 18.
- GIRALDI, G. A. et al. *Animação de Fluidos via Técnicas de Visualização Científica e Mecânica computacional*. 160 p. — Laboratório Nacional de Computação Científica, Rio de Janeiro, 2005. Citado na página 17.
- HOFFMANN, K.; CHIANG, S. *Computational Fluid Dynamics*. [S.l.]: Engineering Education System, 2000. (Computational Fluid Dynamics, v. 1). Citado 3 vezes nas páginas 26, 27 e 30.
- LANGTANGEN, H. P. *A Primer on Scientific Programming with Python: Texts in computational science and engineering*. [S.l.]: Springer, 2009. Citado na página 36.
- LUTZ, M. *Programming Python: Powerful object-oriented programming*. [S.l.]: O'Reilly Media, 2010. Citado na página 38.
- MALISKA, C. Solução numérica de problemas de transferência de calor e mecânica dos fluidos em coordenadas generalizadas. In: I ENCONTRO NACIONAL DE CIÊNCIAS TÉRMICAS. Rio de Janeiro, 1986. p. 27–38. Citado na página 26.
- MALISKA, C. *Transferência de calor e mecânica dos fluidos computacional*. [S.l.]: Livros Tecnicos e Científicos, 2004. Citado 6 vezes nas páginas 17, 18, 19, 20, 21 e 23.

- MANGANI, L. *Development and Validation of an Object Oriented CFD Solver for Heat Transfer and Combustion Modeling in Turbomachinery Applications*. 260 f. Tese (Doutorado em Energia e Tecnologia) — Università degli Studi di Firenze, Florença, 2008. Citado 2 vezes nas páginas 30 e 34.
- MIRANDA, A. M. *Escoamento Laminar de Fluidos Viscosos: Equações de Navier Stokes*. 29 f. Monografia (Especialização em Matemática) — Universidade Federal de Goiás, Catalão, 2007. Citado na página 16.
- NASCIMENTO, J. A. M.; AMARAL, S. A. *Avaliação de usabilidade na internet*. [S.l.]: Thesaurus, 2010. Citado na página 44.
- POLO, G. Pygtk, pyqt, tkinter and wxpython comparison. *The Python Papers*, v. 3, n. 1, p. 26–37, 2008. Citado na página 37.
- QUADROS, R.; BORTOLI, A. L. D. *Introdução ao Cálculo Numérico para Engenheiros*. [S.l.]: Thesaurus, 2009. Citado 2 vezes nas páginas 35 e 36.
- RIBEIRO, D. E. Simulação numérica de aerofólios de alta sustentação. 2002. CREEM 2002 - IX Congresso Nacional de Estudantes de Engenharia Mecânica. Citado 3 vezes nas páginas 17, 20 e 21.
- SANTOS, M. M. *Simulação Numérica do Escoamento Bifásico Óleo-Água em Tubo*. 99 f. Dissertação (Mestrado em Engenharia Mecânica) — Universidade Federal de Itajubá, Itajubá, 2010. Citado 5 vezes nas páginas 17, 18, 19, 21 e 22.
- SILVA, W. P. *Transporte Difusivo em Sólidos com Forma Arbitrária Usando Coordenadas Generalizadas*. 298 f. Tese (Doutorado em Engenharia de Processos) — Universidade Federal de Campina Grande, Campina Grande, 2007. Citado 3 vezes nas páginas 29, 30 e 31.
- THOMPSON, J.; SONI, B.; WEATHERILL, N. *Handbook of Grid Generation*. [S.l.]: CRC Press LLC, 1999. Citado na página 30.
- UFSC - SINMEC: Site. 2014. Disponível em: <<http://www.sinmec.ufsc.br/sinmec/software/transcal.html>>. Acesso em: 11 de jul. de 2014. Citado na página 41.