

Marcio Augusto Villela Pinto  
Sebastião Romero Franco

# **MÉTODO *MULTIGRID* PARA PROBLEMAS TRANSIENTES**

Universidade Federal do Paraná – UFPR

Programa de Pós-Graduação em Métodos Numéricos em Engenharia – PPGMNE

Curitiba

2018

# Sumário

	<b>Sumário</b> . . . . .	<b>2</b>
<b>1</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>3</b>
<b>1.1</b>	<b>Método das Diferenças Finitas (MDF)</b> . . . . .	<b>3</b>
1.1.1	Expansão em fórmula de Taylor . . . . .	5
<b>1.2</b>	<b>Notação estêncil</b> . . . . .	<b>8</b>
<b>1.3</b>	<b>Métodos iterativos</b> . . . . .	<b>9</b>
1.3.1	Método de Jacobi . . . . .	10
1.3.2	Método de Gauss-Seidel . . . . .	10
1.3.3	Ordenação das atualizações das incógnitas . . . . .	10
<b>1.4</b>	<b>Métodos para a aproximação temporal</b> . . . . .	<b>12</b>
1.4.1	Formulações temporais de EDPs . . . . .	13
1.4.2	Método de <i>Euler</i> . . . . .	15
1.4.3	Método de Crank-Nicolson . . . . .	15
<b>2</b>	<b>MÉTODO <i>MULTIGRID</i></b> . . . . .	<b>17</b>
<b>2.1</b>	<b>Caracterização do método <i>multigrid</i></b> . . . . .	<b>17</b>
2.1.1	Análise de convergência . . . . .	18
2.1.2	Análise de erros . . . . .	20
<b>2.2</b>	<b>Operadores de transferência entre malhas</b> . . . . .	<b>21</b>
2.2.1	Operadores de restrição . . . . .	21
2.2.2	Operadores de prolongação . . . . .	25
<b>2.3</b>	<b>Esquema de correção e ciclos</b> . . . . .	<b>27</b>
<b>3</b>	<b>MODELOS MATEMÁTICOS E NUMÉRICOS</b> . . . . .	<b>30</b>
<b>3.1</b>	<b>Modelo matemático</b> . . . . .	<b>30</b>
<b>3.2</b>	<b>Modelo numérico</b> . . . . .	<b>31</b>
<b>4</b>	<b>MÉTODOS DE SOLUÇÃO</b> . . . . .	<b>34</b>
<b>4.1</b>	<b>Método <i>Time-Stepping</i></b> . . . . .	<b>34</b>
<b>4.2</b>	<b>Método <i>Waveform Relaxation</i></b> . . . . .	<b>36</b>
<b>4.3</b>	<b>Método <i>Space-Time</i></b> . . . . .	<b>40</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>43</b>

# 1 Fundamentação teórica

Neste capítulo faz-se uma breve descrição dos principais temas abordados nesta tese. Primeiramente, faz-se uma abordagem sobre o Método das Diferenças Finitas (MDF), que é um método clássico e que apesar da base matemática não ser nova, pode apresentar diferentes formulações de aproximação numérica (TANNEHILL et al., 1997; STRIKWERDA, 1989). Depois disso, mostra-se uma das notações usadas neste trabalho, que é a notação estêncil. Também faz-se uma abordagem dos métodos iterativos e os métodos usados para aproximar a variável temporal, como o método de Euler, o método de Crank-Nicolson (CN), entre outros, veja por exemplo, Strikwerda (1989), Hirsch (2007) e Burden e Faires (2016). O método *multigrid* está exposto no capítulo seguinte.

## 1.1 Método das Diferenças Finitas (MDF)

De acordo com Ferziger e Peric (2002), o primeiro passo para a obtenção de uma solução numérica é discretizar o domínio geométrico, isto é, definir ao longo do domínio em quais pontos se deseja conhecer o valor da variável dependente através do auxílio de uma malha geométrica discreta. O princípio fundamental do MDF é aproximar, através de expressões algébricas, cada termo do modelo matemático em cada ponto (nó) dessa malha discretizada. Para isso, no MDF a malha geométrica é geralmente estruturada localmente, ou seja, cada ponto (nó) pode ser considerado como a origem de um sistema de coordenadas local, cujos eixos coincidem com as linhas da malha. A Fig. 1 e a Fig. 2 mostram exemplos de malhas cartesianas unidimensional (1D) e bidimensional (2D) usadas no MDF, respectivamente. Nessas figuras e nas demais figuras desta seção, o símbolo (•) denota os nós sobre os contornos e o símbolo (○) denota os nós internos da malha. No caso 1D, cada nó é representado pela posição  $i$ . No caso 2D, cada nó é unicamente definido pela interseção das linhas de malhas na posição  $(i, j)$ . Os nós vizinhos são definidos aumentando-se ou diminuindo-se uma unidade de cada índice  $i$  ou  $j$ . Assim o nó  $(i - 1, j)$  representa a posição Oeste,  $(i + 1, j)$  representa o nó na posição Leste,  $(i, j - 1)$  representa o nó na posição Sul e  $(i, j + 1)$  representa o nó na posição Norte.

A Fig. 3 mostra o MDF aplicado a um modelo 1D usando a discretização uniforme, isto é, uma malha de nós igualmente espaçados, com espaço entre os nós dado por  $h$ .

A filosofia do MDF é fazer aproximações diretamente da definição de derivada através da série de Taylor. Uma interpretação geométrica das aproximações central, adiantada e atrasada é mostrada na Fig. 4 com o auxílio de uma função contínua  $\Phi$ . A primeira derivada  $\frac{d\Phi}{dx}$  em um ponto  $x_i$  indica a inclinação da reta tangente à curva no ponto  $(x_i, \Phi(x_i))$  e isso é mostrado com a reta marcada "Exata" na Fig. 4. Essa inclinação pode

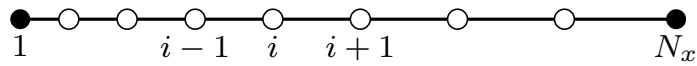


Figura 1 – Exemplo de uma malha unidimensional não uniforme.

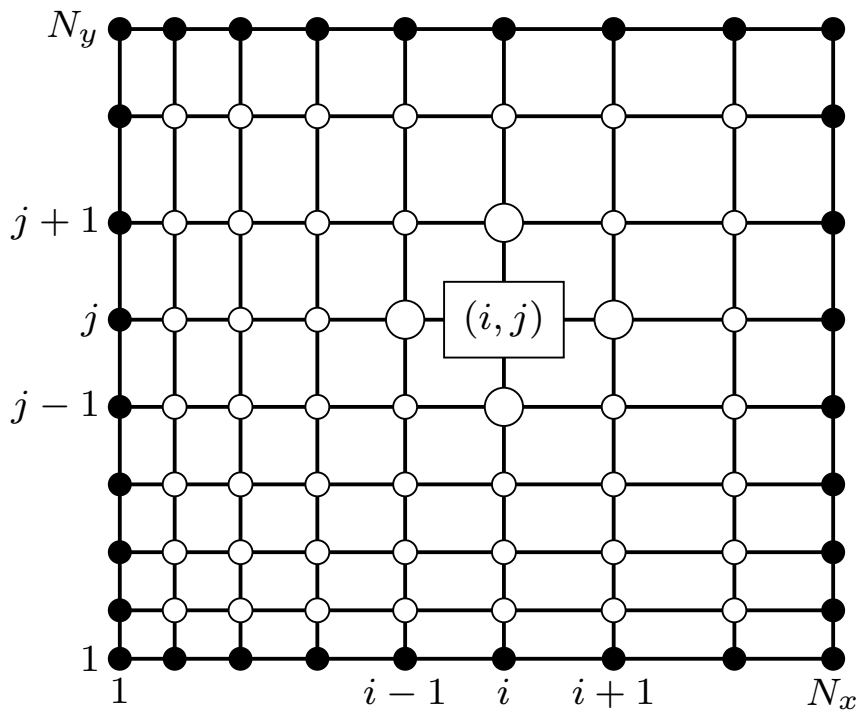


Figura 2 – Exemplo de uma malha bidimensional não uniforme.

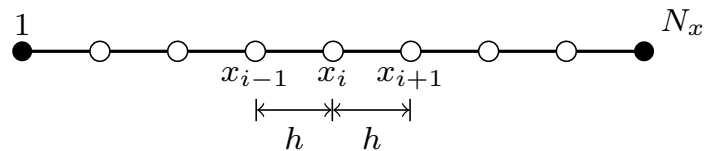


Figura 3 – Malha unidimensional uniforme com tamanho dos elementos de malha  $h$ .

ser aproximada pela inclinação da reta que passa por dois pontos vizinhos na curva. A reta marcada como "Adiantada" mostra que a derivada no ponto  $(x_i, \Phi(x_i))$  é aproximada pela inclinação de uma reta que passa pelos pontos  $(x_i, \Phi(x_i))$  e  $(x_{i+1}, \Phi(x_{i+1}))$ . A reta denotada "Atrasada" ilustra que a derivada no ponto  $(x_i, \Phi(x_i))$  é aproximada pela inclinação de

uma reta que passa pelos pontos  $(x_i, \Phi(x_i))$  e  $(x_{i-1}, \Phi(x_{i-1}))$ . A reta denominada "Central" representa aproximação por uma diferença central, ou seja, a inclinação da reta tangente no ponto  $(x_i, \Phi(x_i))$  é aproximada pela inclinação de uma reta que passa pelos pontos  $(x_{i-1}, \Phi(x_{i-1}))$  e  $(x_{i+1}, \Phi(x_{i+1}))$ , que são pontos situados em lados opostos do ponto em que a derivada é aproximada.

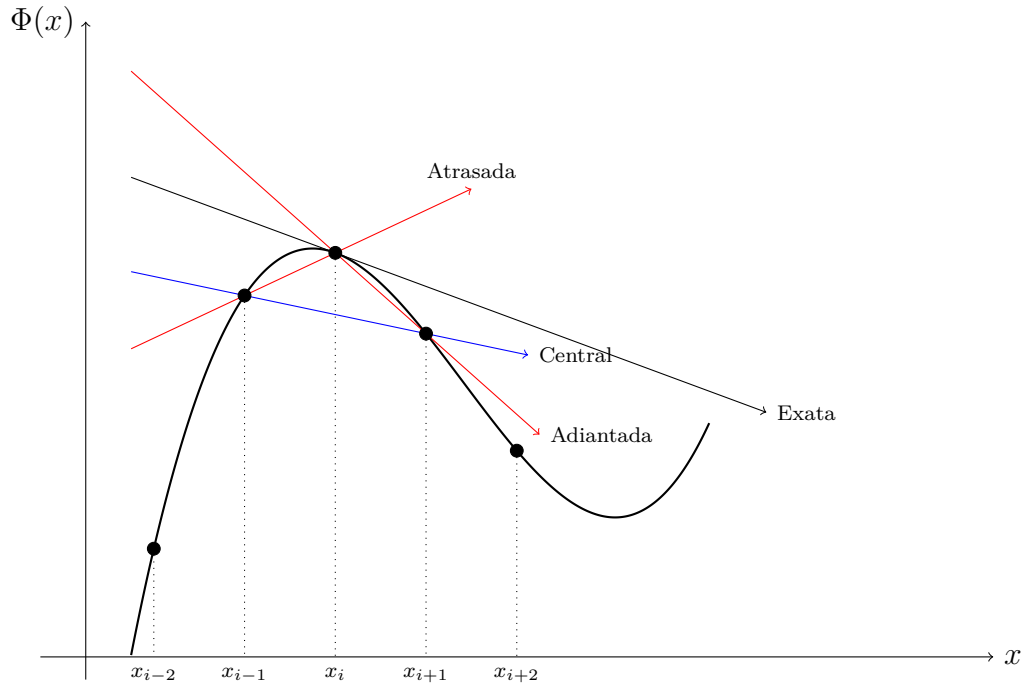


Figura 4 – Reta tangente (Exata) à curva  $\Phi$  no ponto  $P$  e as aproximações Atrasada, Adiantada e Central através da discretização pelo MDF.

Na Fig. 4 observa-se que algumas aproximações para a derivada são melhores que outras (em relação a inclinação com o eixo  $x$ ). Para a reta "Central", verifica-se que ocorre uma melhor aproximação em relação a reta "Exata". Verifica-se também que se  $h$  tende a zero ( $h \rightarrow 0$ ), todas as aproximações (Atrasada, Adiantada e Central) se aproximam da solução "Exata".

**Definição 1.1.** Uma função  $\Phi : I \rightarrow \mathbb{R}$ , definida num intervalo aberto  $I$ , chama-se *analítica* quando, para cada  $x \in I$  existe um  $\varepsilon > 0$  tal que a série de Taylor  $\sum_{n=0}^{+\infty} \frac{\Phi^{(n)}(a)}{n!} h^n$  converge para  $\Phi(a + h)$  desde que  $|h| < \varepsilon$ .

### 1.1.1 Expansão em fórmula de Taylor

Para expressar cada tipo de aproximação e a respectiva ordem do erro, usa-se a série de Taylor (KREYSZIG, 1999; FERZIGER; PERIC, 2002; MARCHI, 2001). Qualquer função  $\Phi(x)$  analítica na vizinhança de  $x_i$ , pode ser expressa como uma série de Taylor:

$$\Phi_x = \Phi_i + (x - x_i) \left( \frac{d\Phi}{dx} \right)_i + \frac{(x - x_i)^2}{2!} \left( \frac{d^2\Phi}{dx^2} \right)_i + \frac{(x - x_i)^3}{3!} \left( \frac{d^3\Phi}{dx^3} \right)_i + \dots \quad (1.1)$$

em que  $\Phi_x$  representa  $\Phi(x)$ ,  $\Phi_i$  representa  $\Phi(x_i)$  e  $\left(\frac{d\Phi}{dx}\right)_i$  representa  $\frac{d\Phi}{dx}(x_i)$ .

O valor de  $\Phi_x$  é exato se forem considerados todos os termos da série de Taylor. Para aplicar a Eq. (1.1) aos nós  $i - 1$  e  $i + 1$ , a partir do ponto  $i$ , como é mostrado na Fig. 3 e com o valor de  $h$  sendo constante, pode-se fazer

$$\begin{aligned} x = x_{i+1} &\implies (x - x_i) = (x_{i+1} - x_i) = h, \\ \Phi_{i+1} &= \Phi_i + h \left(\frac{d\Phi}{dx}\right)_i + \frac{h^2}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i + \frac{h^3}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \dots, \end{aligned} \quad (1.2)$$

$$\begin{aligned} x = x_{i-1} &\implies (x - x_i) = (x_{i-1} - x_i) = -h, \\ \Phi_{i-1} &= \Phi_i - h \left(\frac{d\Phi}{dx}\right)_i + \frac{h^2}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i - \frac{h^3}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \dots \end{aligned} \quad (1.3)$$

De acordo com [Tannehill et al. \(1997\)](#), [Marchi \(2001\)](#) e [Burden e Faires \(2016\)](#) as expressões para a derivada de primeira ordem podem ser obtidas por combinação das Eqs. (1.2) e (1.3). A escolha de quais equações usar define o tipo de esquema e a ordem do erro de truncamento cometido. A seguir serão apresentados alguns dos esquemas mais comuns e que serão usados no decorrer deste texto. Para outras aproximações, veja por exemplo [Burden e Faires \(2016\)](#) e [Ferziger e Peric \(2002\)](#).

Isolando-se a derivada de primeira ordem,  $\left(\frac{d\Phi}{dx}\right)_i$ , na Eq. (1.2), tem-se

$$\left(\frac{d\Phi}{dx}\right)_i = \underbrace{\frac{\Phi_{i+1} - \Phi_i}{h}}_{\text{considerado}} - \underbrace{\frac{h}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i - \frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \dots}_{\text{desprezado (erro de truncamento)}} \quad (1.4)$$

Assim, a derivada de primeira ordem de  $\Phi$  é aproximada de forma atrasada (em inglês, *Downstream Difference Scheme*, DDS) a partir do ponto  $i$  através de

$$\left(\frac{d\Phi}{dx}\right)_i^{DDS} = \frac{\Phi_{i+1} - \Phi_i}{h} + O(h), \quad (1.5)$$

e o erro de truncamento  $\left(\varepsilon \left(\frac{d\Phi}{dx}\right)_i^{DDS}\right)$  é dado por

$$\varepsilon \left(\frac{d\Phi}{dx}\right)_i^{DDS} = -\frac{h}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i - \frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i - \frac{h^3}{24} \left(\frac{d^4\Phi}{dx^4}\right)_i - \dots = O(h). \quad (1.6)$$

Isolando-se a primeira derivada na Eq. (1.3), tem-se

$$\left(\frac{d\Phi}{dx}\right)_i = \underbrace{\frac{\Phi_i - \Phi_{i-1}}{h}}_{\text{considerado}} + \underbrace{\frac{h}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i - \frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \dots}_{\text{desprezado (erro de truncamento)}} \quad (1.7)$$

Assim, a derivada de primeira ordem de  $\Phi$  é aproximada de forma adiantada (em inglês, *Upstream Difference Scheme*, UDS) a partir do ponto  $i$ , através de

$$\left(\frac{d\Phi}{dx}\right)_i^{UDS} = \frac{\Phi_{i+1} - \Phi_i}{h} + O(h), \quad (1.8)$$

e o erro de truncamento é dado por

$$\varepsilon \left(\frac{d\Phi}{dx}\right)_i^{UDS} = \frac{h}{2} \left(\frac{d^2\Phi}{dx^2}\right)_i - \frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \frac{h^3}{24} \left(\frac{d^4\Phi}{dx^4}\right)_i - \dots = O(h). \quad (1.9)$$

Subtraindo-se a Eq. (1.3) da Eq. (1.2), tem-se que

$$\Phi_{i+1} - \Phi_{i-1} = 2h \left(\frac{d\Phi}{dx}\right)_i + \frac{2h^3}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i + \frac{2h^5}{120} \left(\frac{d^5\Phi}{dx^5}\right)_i + \dots, \quad (1.10)$$

ou

$$\left(\frac{d\Phi}{dx}\right)_i = \underbrace{\frac{\Phi_{i+1} - \Phi_{i-1}}{2h}}_{\text{considerado}} - \underbrace{\frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i - \frac{h^4}{120} \left(\frac{d^5\Phi}{dx^5}\right)_i - \dots}_{\text{desprezado (erro de truncamento)}}. \quad (1.11)$$

Assim, a derivada de primeira ordem de  $\Phi$  é aproximada de forma centrada (em inglês, *Central Difference Scheme*, CDS) para o ponto  $i$ , através de

$$\left(\frac{d\Phi}{dx}\right)_i^{CDS} = \frac{\Phi_{i+1} - \Phi_{i-1}}{2h} + O(h^2), \quad (1.12)$$

e o erro de truncamento é dado por

$$\varepsilon \left(\frac{d\Phi}{dx}\right)_i^{CDS} = -\frac{h^2}{6} \left(\frac{d^3\Phi}{dx^3}\right)_i - \frac{h^4}{120} \left(\frac{d^5\Phi}{dx^5}\right)_i - \dots = O(h^2). \quad (1.13)$$

De forma semelhante ao que foi feito na aproximação da derivada de primeira ordem, seguem-se de [Tannehill et al. \(1997\)](#), [Marchi \(2001\)](#) e [Burden e Faires \(2016\)](#), as relações para expressar as aproximações da derivada de segunda ordem.

Somando-se a Eq. (1.2) e a Eq. (1.3), tem-se

$$\Phi_{i-1} + \Phi_{i+1} = 2\Phi_i + h^2 \left(\frac{d^2\Phi}{dx^2}\right)_i + \frac{h^4}{12} \left(\frac{d^4\Phi}{dx^4}\right)_i + \frac{h^6}{360} \left(\frac{d^6\Phi}{dx^6}\right)_i + \dots, \quad (1.14)$$

ou

$$\left(\frac{d^2\Phi}{dx^2}\right)_i = \underbrace{\frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{h^2}}_{\text{considerado}} - \underbrace{\frac{h^2}{12} \left(\frac{d^4\Phi}{dx^4}\right)_i - \frac{h^4}{360} \left(\frac{d^6\Phi}{dx^6}\right)_i - \dots}_{\text{desprezado (erro de truncamento)}} \quad (1.15)$$

Assim, a derivada de segunda ordem de  $\Phi$  é aproximada de forma centrada com 3 pontos (CDS) para o ponto  $P$ , através de

$$\left(\frac{d^2\Phi}{dx^2}\right)_i^{CDS} = \frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{h^2} + O(h^2), \quad (1.16)$$

e o erro de truncamento é dado por

$$\varepsilon \left(\frac{d^2\Phi}{dx^2}\right)_i^{CDS-2} = -\frac{h^2}{12} \left(\frac{d^4\Phi}{dx^4}\right)_i - \frac{h^4}{360} \left(\frac{d^6\Phi}{dx^6}\right)_i - \frac{h^6}{20160} \left(\frac{d^8\Phi}{dx^8}\right)_i - \dots = O(h^2). \quad (1.17)$$

## 1.2 Notação estêncil

A notação estêncil é apropriada para definir um operador diferencial discreto  $L_h$  em uma malha cartesiana retangular  $\Omega^h$ . [Trottenberg et al. \(2001\)](#) e [Wienands e Joppich \(2005\)](#) afirmam que é conveniente usar a terminologia estêncil para descrever funções de malhas do tipo  $\omega_h : \Omega^h \rightarrow \mathbb{R}$  e também para facilitar o uso da análise de Fourier local. Assim, considera-se a função de malha

$$\begin{aligned} \omega_h : \Omega^h &\longrightarrow \mathbb{R} \\ (x, y) &\longmapsto \omega_h(x, y). \end{aligned}$$

Um estêncil geral  $[S_{\kappa_1\kappa_2}]_h$  dado por

$$[S_{\kappa_1\kappa_2}]_h = \begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots & S_{-1,1} & S_{0,1} & S_{1,1} & \cdots \\ \cdots & S_{-1,0} & S_{0,0} & S_{1,0} & \cdots \\ \cdots & S_{-1,-1} & S_{0,-1} & S_{1,-1} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}, \quad S_{\kappa_1\kappa_2} \in \mathbb{R},$$

em que  $\kappa_1, \kappa_2 \in \mathbb{Z}$  indicam as posições no estêncil, define um conjunto de funções de malha dadas por

$$[S_{\kappa_1\kappa_2}]_h \omega_h(x, y) = \sum_{\kappa_1, \kappa_2} \omega_h(x + \kappa_1 h_x, y + \kappa_2 h_y), \quad (1.18)$$

em que  $h_x$  e  $h_y$  indicam as distâncias entre os nós espaciais nas direções  $x$  e  $y$ , respectivamente.

Aqui, assume-se que somente um número finito de coeficientes  $S_{\kappa_1\kappa_2}$  são diferentes de zero.

Os estêncis mais comuns são os compactos de cinco pontos e de nove pontos, dados na malha  $h$ , respectivamente por

$$\begin{bmatrix} & S_{0,1} & \\ S_{-1,0} & S_{0,0} & S_{1,0} \\ & S_{0,-1} & \end{bmatrix}_h \quad \text{e} \quad \begin{bmatrix} S_{-1,1} & S_{0,1} & S_{1,1} \\ S_{-1,0} & S_{0,0} & S_{1,0} \\ S_{-1,-1} & S_{0,-1} & S_{1,-1} \end{bmatrix}_h. \quad (1.19)$$





A matriz dos coeficientes  $A$  do sistema (1.22) pode ser dividida na forma:

$$A = -L + D - U, \quad (1.23)$$

em que  $-L$  é uma matriz triangular inferior formada pela parte inferior da matriz  $A$ ,  $D$  é a matriz formada pela diagonal de  $A$  e  $-U$  é uma matriz triangular superior formada pela parte superior da matriz  $A$ .

### 1.3.1 Método de Jacobi

Seguindo Ruggiero e Lopes (1996), Saad (2003) e Ferziger e Peric (2002) e considerando a matriz  $D$  não singular, ou seja  $d_{ii} \neq 0 \forall i$ , pode-se reescrever o sistema linear dado na Eq. (1.21) na forma

$$\begin{aligned} (-L + D - U)u &= f \\ Du &= (L + U)v + f. \end{aligned} \quad (1.24)$$

Isolando-se  $u$  na iteração atual e considerando-se  $v$  como sendo sua aproximação, o procedimento que caracteriza o método de Jacobi na forma matricial é dado por

$$v^{\nu+1} = D^{-1}(L + U)v^{\nu} + D^{-1}f. \quad (1.25)$$

### 1.3.2 Método de Gauss-Seidel

Da mesma forma que o método de Jacobi, considerando-se a matriz  $D$  não singular, pode-se escrever o sistema linear dado na Eq. (1.21) na forma

$$\begin{aligned} (-L + D - U)u &= f, \\ (-L + D)u &= Uu + f. \end{aligned}$$

Isolando-se  $u$  na iteração atual e considerando-se  $v$  como sendo sua aproximação, o procedimento que caracteriza o método de Gauss-Seidel na forma matricial é dado por

$$v^{\nu+1} = (-L + D)^{-1}Uv^{\nu} + (-L + D)^{-1}f. \quad (1.26)$$

Nos procedimentos descritos anteriormente, para Jacobi e Gauss-Seidel, a matriz de iteração é constante ao longo do processo iterativo. Os métodos iterativos que possuem esta propriedade são chamados métodos iterativos básicos ou estacionários (TROTTEMBERG et al., 2001).

### 1.3.3 Ordenação das atualizações das incógnitas

A eficiência destes métodos iterativos está fortemente ligada à ordenação em que as incógnitas são determinadas ou atualizadas. As ordens mais comuns são a ordenação

lexicográfica (Lex) (Fig. 5), a ordenação *red-black* (RB) (Fig. 6) e algumas variantes, como a ordenação de varredura para os métodos de suavização por blocos gerando os métodos por linhas (Fig. 7), zebra (Fig. 8), entre outros (WESSELING, 1992; TROTTENBERG et al., 2001; WIENANDS; JOPPICH, 2005). Um dos objetivos das ordens de atualização coloridas (ou zebras) é possibilitar o desenvolvimento de algoritmos paralelizáveis.

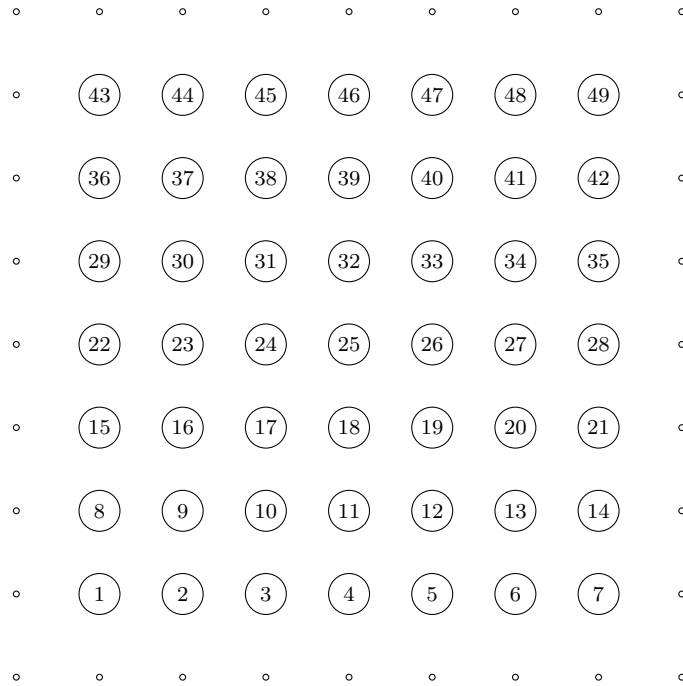


Figura 5 – Ordenação lexicográfica para atualização das incógnitas.

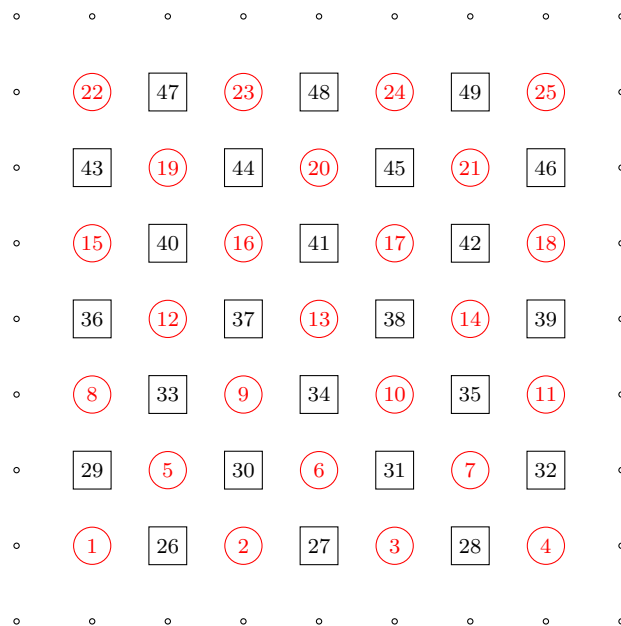
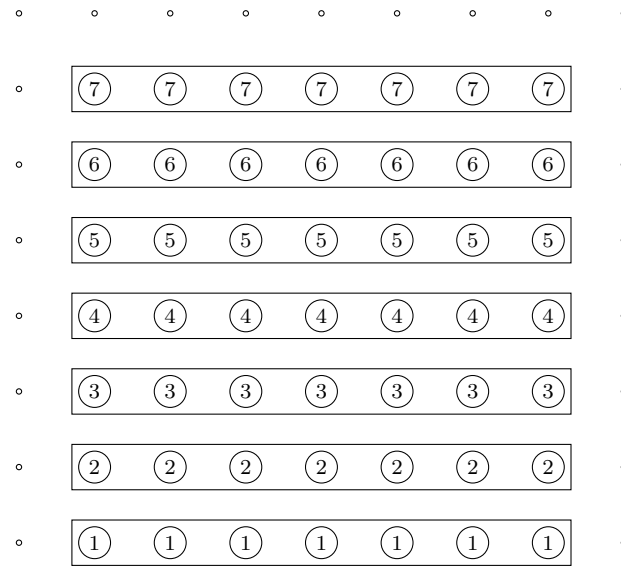
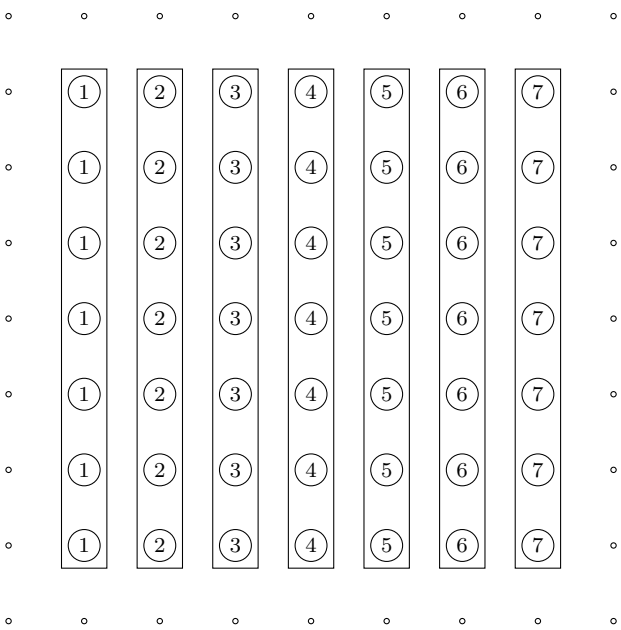


Figura 6 – Ordenação *red-black* para atualização das incógnitas: (○) *red* e (□) *black*.



(a) Linha horizontal.



(b) Linha Vertical.

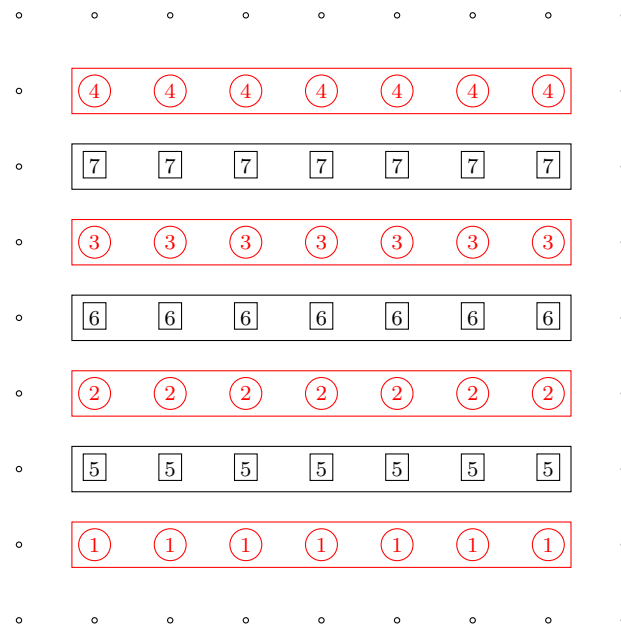
Figura 7 – Ordenação para atualização das incógnitas com um método linha.

## 1.4 Métodos para a aproximação temporal

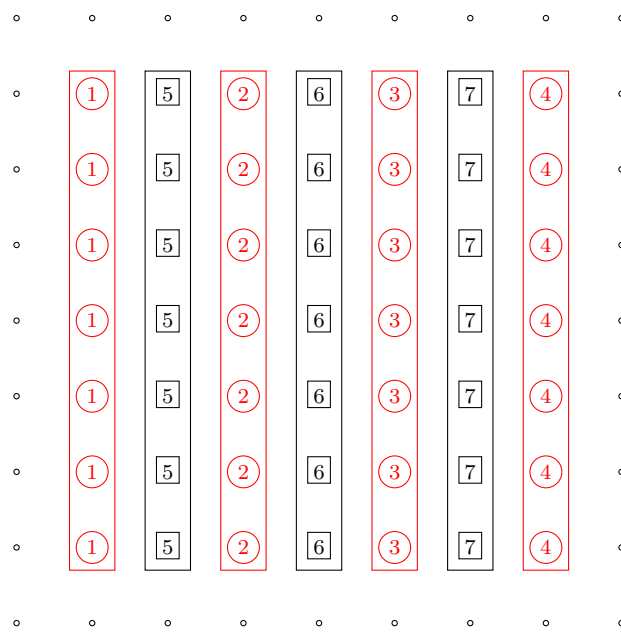
Dado um Problema de Valor Inicial (PVI)

$$\begin{cases} \frac{du}{dt} = f \\ u(x, y, t_0) = u^0 \end{cases} \quad (1.27)$$

com  $u$  e  $f$  funções das variáveis  $(x, y, t)$ , em que  $[x, y]$  está contido em um domínio espacial,  $t \in [t_0, t_f]$ , em que  $t_0$  representa o tempo inicial e  $t_f$  o tempo final,  $u^0$  é a condição inicial (função das variáveis espaciais  $x$  e  $y$ ). Nesta tese, as condições para existência e unicidade



(a) Zebra horizontal.



(b) Zebra Vertical.

Figura 8 – Ordenação para atualização das incógnitas com um método zebra.

da solução são garantidas, ver (BURDEN; FAIRES, 2016). Observa-se que a derivada temporal é ordinária, pois apesar de depender dos valores espaciais  $(x, y)$ , consideram-nos conhecidos durante o cálculo da variável temporal.

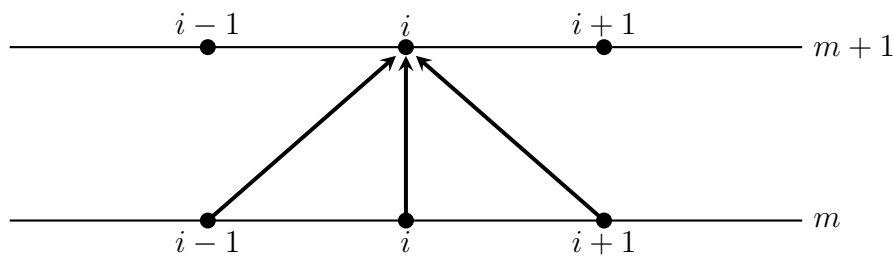
### 1.4.1 Formulações temporais de EDPs

Maliska (2004) relata que uma formulação é dita explícita a um passo de tempo quando todas as incógnitas vizinhas ao ponto  $i$  são avaliadas nos passos de tempo anteriores

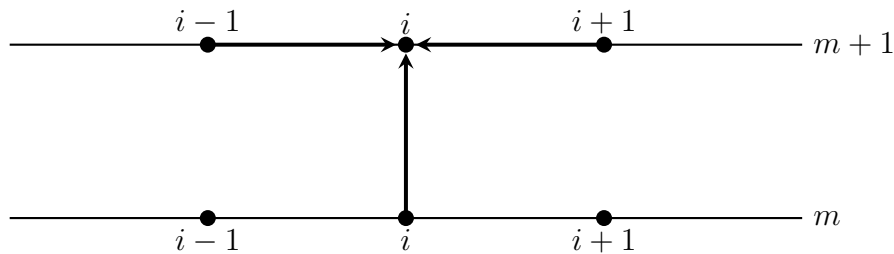
e, portanto, já são conhecidas, como mostra a Fig. 9(a).

Uma formulação é dita totalmente Implícita a um passo de tempo quando todas as incógnitas vizinhas ao ponto  $i$  são avaliadas no passo de tempo atual e, portanto, não são conhecidas em sua totalidade (depende da ordenação de atualização das incógnitas), como mostra a Fig. 9(b).

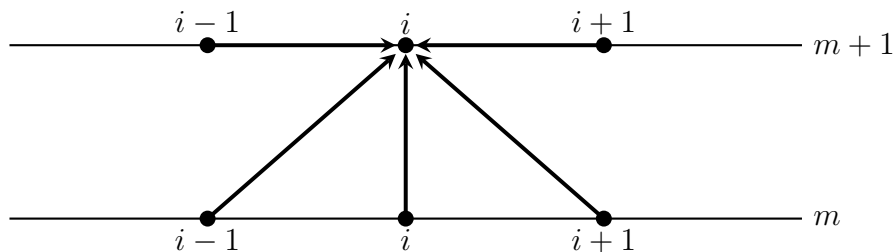
A formulação implícita a um passo de tempo ocorre quando as incógnitas vizinhas ao ponto  $i$  são avaliadas nos passos de tempo atual e anterior, como mostra a Fig. 9(c).



(a) Formulação Explícita.



(b) Formulação Totalmente Implícita.



(c) Formulação Implícita.

Figura 9 – Conexão espacial e temporal em um ponto  $i$  com formulações Explícita, Totalmente Implícita e Implícita (MALISKA, 2004, p. 39).

### 1.4.2 Método de *Euler*

O método de *Euler* consiste em obter uma aproximação para a solução  $u(t)$  de (1.27) de forma discreta em diversos valores no intervalo  $[t_0, t_f]$ , chamados pontos de malha. Para isso, estipula-se que os pontos de malha têm distribuição uniforme no intervalo  $[t_0, t_f]$ , seleciona-se um número positivo inteiro  $N_t$  e os pontos de malha (em  $t$ ) são dados por

$$t_{m+1} = t_0 + (m + 1)\tau \quad \text{para cada } m = 0, 1, \dots, N_t - 1. \quad (1.28)$$

A distância entre os pontos  $\tau = \frac{t_f - t_0}{N_t}$  é chamada de tamanho do passo de tempo.

Seja  $u(t)$  a única solução da Eq. (1.27), com duas derivadas contínuas em  $[t_0, t_f]$ , de modo que sua expansão em série de Taylor para cada  $m = 0, 1, \dots, N_t - 1$  é dada por

$$u(t_{m+1}) = u(t_m) + \tau \frac{d}{dt}u(t_m) + \frac{\tau^2}{2} \frac{d^2}{dt^2}u(\xi_k), \quad (1.29)$$

para algum  $\xi_m \in (t_0, t_f)$ .

Considerando-se  $v$  uma aproximação para  $u$ , o método de Euler constrói  $v^m \approx u(t_m)$  para cada  $m = 0, 1, \dots, N_t - 1$  ao desconsiderar os termos  $\frac{\tau^2}{2} \frac{d^2}{dt^2}u(\xi_k)$ , de segunda ordem de aproximação em  $\tau$ . Dessa forma, o método de *Euler Implícito* (primeira ordem de aproximação no tempo) é dado por

$$\begin{cases} v^0 = y^0 \\ v^{m+1} = v^m + \tau f(v^{m+1}, t_{m+1}) \end{cases} \quad \text{para cada } m = 0, 1, \dots, N_t - 1. \quad (1.30)$$

Esse método é chamado de método de *Euler Implícito*, pois para obter o valor de  $v^{m+1}$  são necessários valores relacionados aos passos de tempo  $m$  e  $m + 1$ . Se para obter o valor de  $v^{m+1}$  utilizar apenas valores relacionados ao passo de tempo  $m + 1$  o método de Euler é dito método de Euler Totalmente Implícito e se são necessários apenas valores relacionados ao passo de tempo  $m$  o método de Euler é dito método de Euler Explícito (BURDEN; FAIRES, 2016; HIRSCH, 2007).

O método Explícito é condicionalmente estável, ou seja, converge quando satisfaz um critério de convergência (BURDEN; FAIRES, 2016) que relaciona o tamanho da discretização espacial ao tamanho do passo temporal. Os métodos Implícitos e Totalmente Implícitos são incondicionalmente estáveis, ou seja, independem do tamanho da discretização espacial e do tamanho do passo temporal (BURDEN; FAIRES, 2016; STRIKWERDA, 1989).

### 1.4.3 Método de Crank-Nicolson

De acordo com Tannehill et al. (1997), Burden e Faires (2016), Strikwerda (1989) e Hirsch (2007), o método de Crank-Nicolson (CN) é um método incondicionalmente estável,

ou seja, a sua convergência não depende da relação entre o tamanho do passo de tempo e o tamanho da discretização espacial e possui segunda ordem de aproximação temporal ( $O(\tau^2)$ ).

Considerando por exemplo a Eq. (1.27), o método de CN consiste em fazer

$$\begin{cases} v^0 = y^0 \\ v^{m+1} = v^m + \frac{\tau}{2} [f(v^{m+1}, t_{m+1}) + f(v^m, t_m)] \end{cases} \quad m = 0, 1, \dots, N_t - 1, \quad (1.31)$$

em que os subíndices  $m+1$  e  $m$  indicam os passos de tempo atual e anterior, respectivamente.



## 2 Método *multigrid*

Quando o problema proposto é discretizado e resolvido em apenas uma malha, diz-se que este problema está sendo resolvido com o método *singlegrid*. Se usar duas malhas para resolver o problema, diz-se que está sendo usado o método de duas malhas (ou *two-grid*). Porém, com o uso de três ou mais malhas para a solução do problema, diz-se que está sendo usado o método *multigrid*.

O método *multigrid* é uma técnica eficiente para acelerar a suavização das fortes oscilações do resíduo usando um método de relaxação (método iterativo básico). Proposto originalmente por Fedorenko (1964), esse método é atualmente muito utilizado para acelerar a convergência de sistemas de equações do tipo  $Au = f$ . O princípio de funcionamento desse método está fortemente relacionado ao comportamento de convergência dos métodos iterativos básicos (Gauss-Seidel e Jacobi, por exemplo). Esses métodos apresentam propriedades de suavização dos erros de alta frequência (componentes oscilatórias), enquanto as baixas frequências são mantidas praticamente inalteradas. Assim, nas primeiras iterações reduzem-se rapidamente as componentes oscilatórias do erro e à medida que esse número vai aumentando, o erro começa a cair mais lentamente, indicando a predominância das componentes suaves (BRANDT, 1977; STÜBEN; TROTTEBERG, 1981; WESSELING, 1992; BRIGGS et al., 2000; TROTTEBERG et al., 2001). Assim, o método *multigrid* consiste no uso de discretização do problema em diversas malhas, ou seja, após a suavização das componentes oscilatórias em uma malha fina, as componentes dos erros (suaves) são transferidas para uma malha mais grossa, tornando-se oscilatórias. Esse processo pode ser repetido até a malha mais grossa possível e faz com que o método iterativo usado não perca a eficiência, pois agirá sempre nas componentes oscilatória dos erros.

Como dito anteriormente, o método *multigrid* geométrico utiliza-se de informações das malhas consideradas na discretização do problema, o que significa que ele é recomendado para problemas em malhas estruturadas. Como neste trabalho serão usadas apenas malhas estruturadas, o método *multigrid* geométrico será a ferramenta de trabalho adotada.

### 2.1 Caracterização do método *multigrid*

A ideia principal do método *multigrid* consiste em usar uma discretização de malha adequada para que, usando um suavizador (Gauss-Seidel, Jacobi, ...), as componentes oscilatórias do erro sejam suavizadas rapidamente nas primeiras iterações. Após a suavização, o método *multigrid* procura trabalhar com uma sequência de malhas cada vez mais grossas, de tal forma que os comprimentos de ondas do erro, que são longos em

malhas finas, sejam transformados em curtos em malhas mais grossas, tornando-se mais oscilatórios, situação na qual as componentes do erro são suavizadas mais rapidamente.

A razão de engrossamento entre as malhas  $q$  é dada por

$$q = \frac{h_2}{h_1}, \quad (2.1)$$

em que  $h_1$  e  $h_2$  são o tamanhos dos elementos da malha consecutivamente mais grossa e fina, respectivamente. A Fig. 10 mostra uma sequência de três malhas bidimensionais ( $\Omega^h$ ,  $\Omega^{2h}$  e  $\Omega^{4h}$ ) com razão de engrossamento padrão  $q = 2$ .

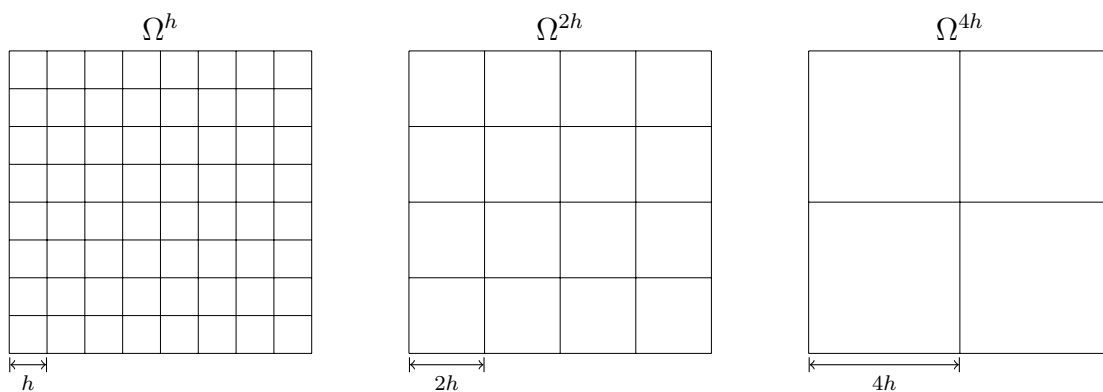


Figura 10 – Sequência de 3 malhas bidimensionais e uniformes com engrossamento padrão.

### 2.1.1 Análise de convergência

Seja o sistema linear na forma

$$Au = f. \quad (2.2)$$

Supondo que este sistema tenha uma única solução e que  $v$  é uma aproximação para  $u$ , pode-se definir duas medidas importantes de  $v$  como aproximação de  $u$ .

**Definição 2.1.** O erro (ou erro algébrico) é dado por:

$$e = u - v. \quad (2.3)$$

Porém, o erro de uma solução numérica é inacessível quando não se conhece a solução analítica. Assim, uma medida calculável para verificar o quanto  $v$  se aproxima de  $u$  é o resíduo.

**Definição 2.2.** O resíduo da Eq. (2.2) para uma aproximação  $v$  é dado por

$$r = f - Av. \quad (2.4)$$

Como a solução aproximada  $v$  é obtida através de um processo iterativo, pode-se escrever o erro e o resíduo em função da iterada atual  $\nu$ , respectivamente, como

$$e^\nu = u - v^\nu, \quad (2.5)$$

$$r^\nu = f - Av^\nu, \quad (2.6)$$

sendo que a magnitude de ambos pode ser medida pelas normas vetoriais conhecidas (BURDEN; FAIRES, 2016).

Uma importante relação entre erro e resíduo é mostrado em Briggs et al. (2000), através da expressão

$$\frac{1}{\text{cond}(A)} \frac{\|r\|}{\|f\|} \leq \frac{\|e\|}{\|u\|} \leq \text{cond}(A) \frac{\|r\|}{\|f\|}, \quad (2.7)$$

com  $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$  sendo o número de condicionamento da matriz  $A$ .

Observe que, pelas inequações (2.7),  $\|r\| \approx 0$  implica em  $\|e\| \approx 0$  somente se  $\text{cond}(A)$  é pequeno, ou seja, próximo da unidade ( $\text{cond}(A) \approx 1$ ). Neste caso, diz-se que a matriz  $A$  é bem condicionada.

A partir das Eqs. (1.25) e (1.26) pode-se construir o método iterativo dado por

$$v^{\nu+1} = Sv^\nu + g, \quad (2.8)$$

em que  $v$  é uma aproximação para a solução desejada. Nota-se que  $S$  dependerá do método iterativo utilizado.

**Definição 2.3.** O método iterativo  $v^{\nu+1} = Sv^\nu + g$  é chamado de convergente se e somente se  $\lim_{\nu \rightarrow \infty} \|S\|^\nu = 0$ .

**Definição 2.4.** O fator de convergência do método iterativo dado na Eq. (2.8) é

$$\rho(S) = \max |\lambda(S)|.$$

Na definição 2.4,  $\lambda(S)$  representa os autovalores da matriz  $S$  e o fator de convergência  $\rho$  (máximo autovalor, em módulo, de  $S$ ) é também chamado de raio espectral da matriz  $S$ . Este  $\rho$  indica a pior redução do erro com o passo iterativo (BRIGGS et al., 2000).

**Teorema 2.1.** O método iterativo  $v^{\nu+1} = Sv^\nu + g$  é convergente se e somente se  $\rho(S) \leq 1$ .

Prova: ver Burden e Faires (2016).

O raio espectral ou fator de convergência pode ser aproximado numericamente como descrito em Janssen (1997, p. 43) por

$$\rho^{\nu+1} = \frac{\|e^{\nu+1}\|}{\|e^\nu\|}. \quad (2.9)$$

O mesmo autor comenta que se usar o resíduo ao invés do erro na Eq. (2.9), os resultados serão semelhantes.

O fator de convergência médio pode ser definido como a média geométrica dos fatores de convergência de (2.9). Assim,

$$\rho_m = \sqrt[n]{\rho^{(1)} \cdot \rho^{(2)} \cdot \dots \cdot \rho^{(n)}} , \quad (2.10)$$

em que  $n$  representa o número de iteradas ou ciclos *multigrid* realizados.

### 2.1.2 Análise de erros

Para uma análise do comportamento dos erros, Briggs et al. (2000) afirmam que é suficiente utilizar o sistema homogêneo  $Au = 0$ . Neste caso, tem-se a vantagem da solução exata ( $u = 0$ ) ser conhecida e o erro da aproximação dada por  $v$  é simplesmente  $-v$ .

Para facilitar os cálculos, considera-se aqui o caso unidimensional da equação de Poisson com condições de contorno de Dirichlet com estimativa inicial, ou modos de Fourier, dada por  $v_j = \text{sen}\left(\frac{jk\pi}{N}\right)$ ,  $1 \leq j \leq N - 1$  e  $k \leq N - 1$ , em que  $v$  é uma aproximação da solução  $u$ ,  $N$  é o número de pontos da discretização,  $k$  é o número de em que ou modos de Fourier e  $j$  é a componente do vetor  $v$ .

**Definição 2.5.** Os modos de Fourier localizados na metade inferior do espectro, com  $1 \leq k < \frac{N}{2}$ , são chamados de modos de Fourier de baixa frequência ou modos suaves. Os modos de Fourier localizados na metade superior do espectro, com  $\frac{N}{2} \leq k \leq N - 1$ , são chamados de modos de Fourier de alta frequência ou modos oscilatórios.

A Fig. 11(a) e a Fig. 11(b) mostram os efeitos do engrossamento de malhas para suavizar os modos de Fourier. Observa-se que o número de ondas  $k$  permanece inalterado, enquanto que o número de pontos é reduzido, fazendo com que a onda torne-se mais oscilatória ( $\frac{N}{2} \leq k \leq N - 1$ ).

Ao se projetar um modo suave com número de ondas  $1 \leq k < \frac{N}{2}$  da malha fina  $\Omega^h$  para a malha grossa  $\Omega^{2h}$ , transforma-se o modo em oscilatório (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992). Por isso é recomendável transferir o problema de relaxação para a malha grossa após a suavização dos erros oscilatórios, pois lá os modos de erros suaves se apresentarão mais oscilatórios, e o processo de relaxação será mais eficiente.

Com essa filosofia para suavizar os erros de baixa frequência com o uso de malhas mais grossas, o método *multigrid* possui uma taxa (ou fator) de convergência ideal (teórica) que independe do tamanho da malha, isto é, independe do número de pontos da discretização da malha (FERZIGER; PERIC, 2002; ROACHE, 1998). Para obter um bom desempenho do método *multigrid*, devem-se usar diversos níveis de malhas (TANNEHILL et al., 1997; PINTO; MARCHI, 2007).

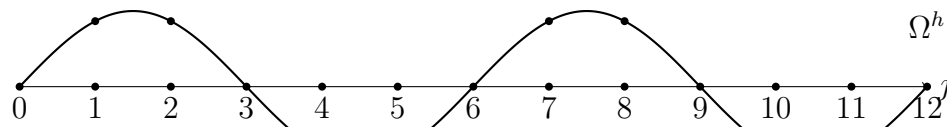
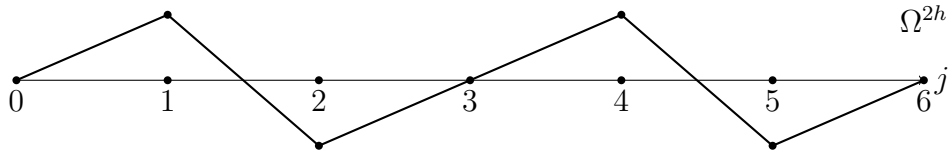
(a)  $k = 4$  ondas sobre a malha com  $N = 12$  pontos.(b)  $k = 4$  ondas sobre a malha com  $N = 6$  pontos.

Figura 11 – Número de ondas  $k = 4$  sobre uma malha fina com  $N = 12$  pontos e sobre uma malha grossa com  $N = 6$  pontos (BRIGGS et al., 2000, p. 32).

## 2.2 Operadores de transferência entre malhas

Os elementos essenciais do método *multigrid* são os operadores de restrição  $(I_h^{2h})$  e de prolongação  $(I_{2h}^h)$ . Com o problema representado em uma malha fina ( $\Omega^h$ ) é necessário transferir as informações para as malhas mais grossas ( $\Omega^{2h}, \Omega^{4h}, \dots$ ). Isto é feito por um operador chamado *operador de restrição*. Para fazer o caminho contrário e transferir as informações das malhas grossas para as malhas mais finas, faz-se necessário um operador chamado *operador de prolongação*.

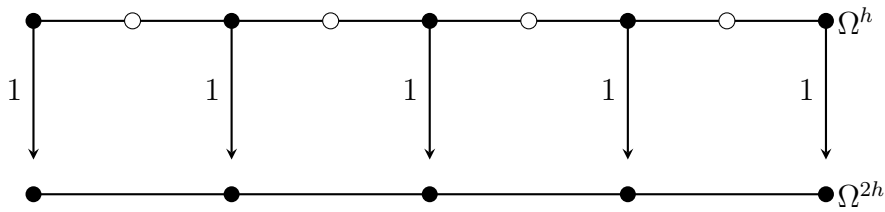
Nas figuras deste capítulo, o símbolo (•) representa os nós que pertencem a ambas as malhas, o símbolo (◦) representa os nós que pertencem apenas à malha fina e os números (ao lado das setas) representam os pesos atribuídos a cada tipo de restrição ou prolongação.

### 2.2.1 Operadores de restrição

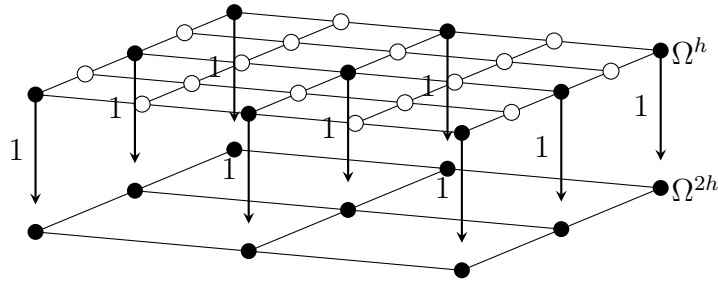
O operador de restrição  $(I_h^{2h})$  é responsável por transferir informações (resíduo e/ou solução) de uma determinada malha  $\Omega^h$ , com discretização de malhas de tamanho  $h$  (em todas as direções espaciais), para a malha mais grossa  $\Omega^{2h}$  com  $2h$ . A restrição pode ser por Injeção (Inj), por meia ponderação (HW), por ponderação completa (FW), entre outros (BRIGGS et al., 2000; TROTTEBERG et al., 2001; HORTON; VANDEWALLE, 1995).

O operador de restrição por injeção  $(I_h^{2h})_{Inj}$  apenas transfere a informação de um ponto na malha fina em seu correspondente ponto na malha mais grossa. A Fig. 12(a) e a Fig. 12(b) ilustram este procedimento para os casos unidimensional (1D) e bidimensional (2D), respectivamente.

Por exemplo, no caso 2D, o resíduo calculado pelo operador  $(I_h^{2h})_{Inj}$  na malha



(a) Problema 1D.



(b) Problema 2D.

Figura 12 – Operadores de restrição por injeção 1D e 2D.

grossa, é dado por

$$\begin{aligned} r_{2h}(x, y) &= \left( I_h^{2h} \right)_{Inj} r_h(x, y) \\ &= r_h(x, y). \end{aligned} \tag{2.11}$$

Em notação estêncil, tem-se

$$\left( I_h^{2h} \right)_{Inj} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h}. \tag{2.12}$$

O operador de restrição por meia ponderação  $\left( I_h^{2h} \right)_{HW}$ , no caso 2D, utiliza as informações dos 4 pontos mais próximos ao ponto calculado na malha fina para o correspondente ponto na malha mais grossa. A Fig. 13 ilustra esse procedimento.

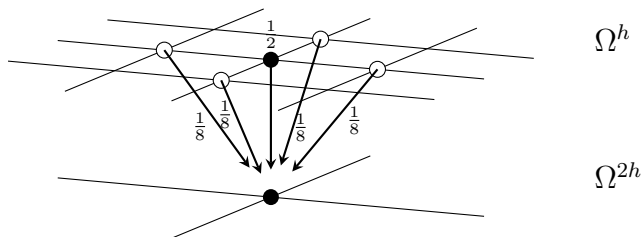


Figura 13 – Operador de restrição por meia ponderação.

O resíduo para o caso 2D, por exemplo, calculado pelo operador  $(I_h^{2h})_{HW}$  na malha grossa é dado por

$$\begin{aligned} r_{2h}(x, y) &= (I_h^{2h})_{HW} r_h(x, y) \\ &= \frac{1}{8} [4r_h(x, y) + r_h(x, y - h) + r_h(x - h, y) + r_h(x + h, y) + r_h(x, y + h)]. \end{aligned} \quad (2.13)$$

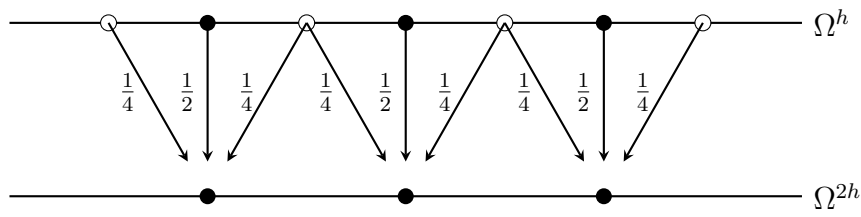
Em notação estêncil, tem-se

$$(I_h^{2h})_{HW} = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}_h^{2h}. \quad (2.14)$$

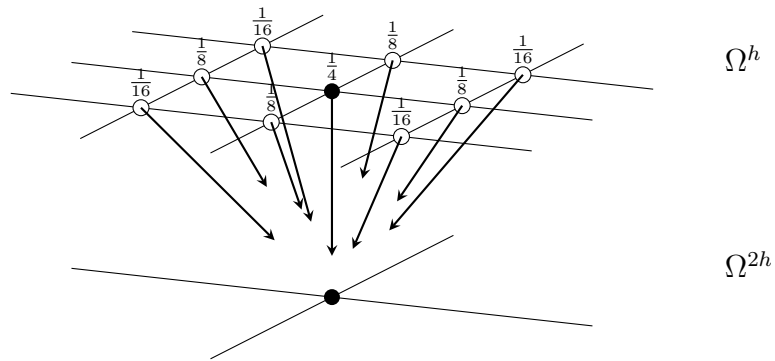
O operador de restrição por ponderação completa  $(I_h^{2h})_{FW}$  utiliza informações de todos os pontos vizinhos ao ponto calculado na malha fina para o correspondente ponto na malha grossa. A Fig. 14(a) e a Fig. 14(b) ilustram este procedimento para os casos unidimensional (1D) e bidimensional (2D), respectivamente.

Por exemplo, o resíduo para o caso 2D calculado pelo operador  $(I_h^{2h})_{FW}$  na malha grossa é dado por

$$\begin{aligned} r_{2h}(x, y) &= (I_h^{2h})_{FW} r_h(x, y) \\ &= \frac{1}{16} \{4r_h(x, y) + 2[r_h(x, y - h) + r_h(x - h, y) + r_h(x + h, y) + r_h(x, y + h)] + \\ &\quad + r_h(x - h, y - h) + r_h(x + h, y - h) + r_h(x - h, y + h) + r_h(x + h, y + h)\}. \end{aligned} \quad (2.15)$$



(a) Problema 1D.



(b) Problema 2D.

Figura 14 – Operadores de restrição por ponderação completa para os casos 1D e 2D.

Em notação estêncil, tem-se

$$\left(I_h^{2h}\right)_{FW} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h}. \quad (2.16)$$



### 2.2.2 Operadores de prolongação

O operador de prolongação (ou interpolação)  $I_{2h}^h$  é responsável por transferir informação (correção) de uma determinada malha grossa  $\Omega^{2h}$ , com discretização de malhas de tamanho  $2h$ , para a malha fina  $\Omega^h$  com malha de tamanho  $h$ . A prolongação pode ser por interpolação linear (para o caso 1D), bilinear (para o caso 2D), entre outras. Para esses operadores serão empregados Hackbusch (1985), Wesseling (1992), Briggs et al. (2000) e Trottenberg et al. (2001).

A interpolação linear da correção pelo operador  $I_{2h}^h$  da malha grossa para a malha fina, no caso 1D (Fig. 15), é dado por

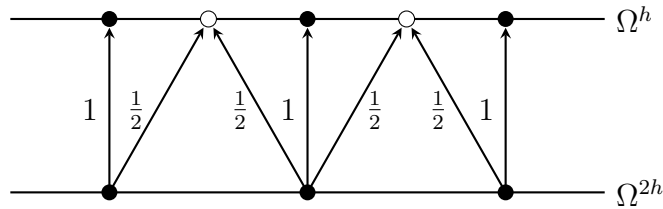


Figura 15 – Operador de interpolação linear para o caso 1D.

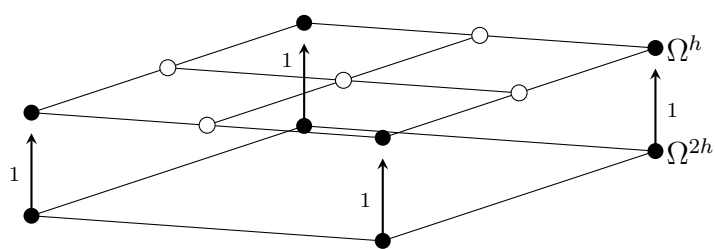
$$\begin{aligned}
 v_h(x) &= I_{2h}^h v_{2h}(x) \\
 &= \begin{cases} \frac{1}{2}[v_{2h}(x-h) + v_{2h}(x+h)] & \text{para } \circ \\ v_{2h}(x) & \text{para } \bullet \end{cases} .
 \end{aligned}
 \tag{2.17}$$

Em notação estêncil, tem-se

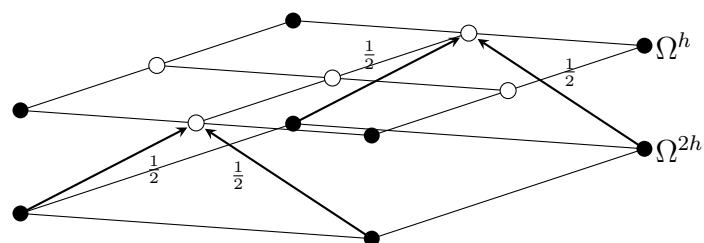
$$I_{2h}^h = \frac{1}{2} \begin{bmatrix} 1 & 2 & 1 \\ \hline & & \end{bmatrix} \begin{matrix} h \\ 2h \end{matrix} .
 \tag{2.18}$$

Para o caso 2D, usa-se um operador de interpolação bilinear  $I_{2h}^h$  (Fig. 16). Nesse caso, a interpolação bilinear da correção pelo operador  $I_{2h}^h$  da malha  $\Omega^{2h}$  para a malha  $\Omega^h$  (Fig. 16(a) a 16(d)) é dado por

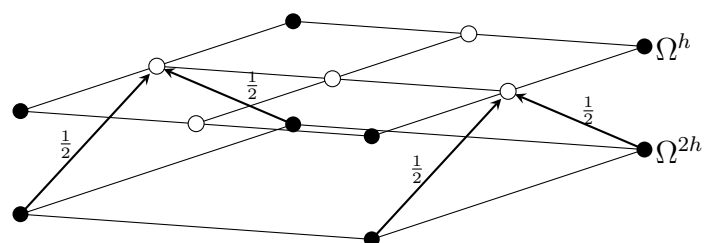
$$\begin{aligned}
 v_h(x, y) &= I_{2h}^h v_{2h}(x, y), \\
 v_h(x, y) &= \begin{cases} v_{2h}(x, y) \\ \frac{1}{2}[v_{2h}(x-h, y) + v_{2h}(x+h, y)] \\ \frac{1}{2}[v_{2h}(x, y-h) + v_{2h}(x, y+h)] \\ \frac{1}{4}[v_{2h}(x-h, y) + v_{2h}(x+h, y) + v_{2h}(x, y-h) + v_{2h}(x, y+h)] \end{cases}
 \end{aligned}
 \tag{2.19}$$



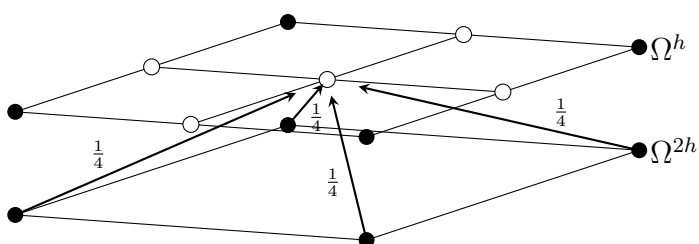
(a) Injeção direta.



(b) Interpolação linear na direção  $x$ .



(c) Interpolação linear na direção  $y$ .



(d) Interpolação bilinear.

Figura 16 – Operador de Interpolação bilinear para o caso 2D.

Em notação estêncil, tem-se

$$I_h^{2h} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{matrix} \left[ \begin{matrix} h \\ \\ \end{matrix} \right] \\ \\ \left[ \begin{matrix} \\ \\ 2h \end{matrix} \right] \end{matrix} . \quad (2.20)$$

## 2.3 Esquema de correção e ciclos

Neste trabalho o método *multigrid* foi implementado utilizando o esquema de correção (em inglês, *Correction Scheme*, CS). De acordo com Briggs et al. (2000), o esquema CS é mais indicado para problemas lineares.

A filosofia do esquema CS pode ser representada pelo seguinte procedimento:

- Suavize  $Au = f$  na malha mais fina  $\Omega^h$  para obter uma aproximação  $v^h$ .
- Calcule o resíduo  $r = f - Av^h$ .

Suavize a equação residual  $Ae = r$  em  $\Omega^{2h}$  com estimativa inicial  $e = 0$  para obter uma aproximação para o erro  $e^{2h}$ .

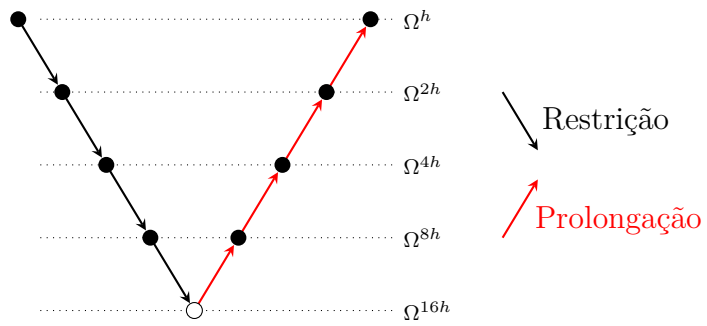
- Corrija a aproximação obtida em  $\Omega^h$  com o erro estimado em  $\Omega^{2h}$ :  $v^h \leftarrow v^h + e^{2h}$ .

Esse procedimento expõe um esquema CS para o caso de duas malhas, porém a mesma ideia pode ser estendida para o número de malhas que se queira.

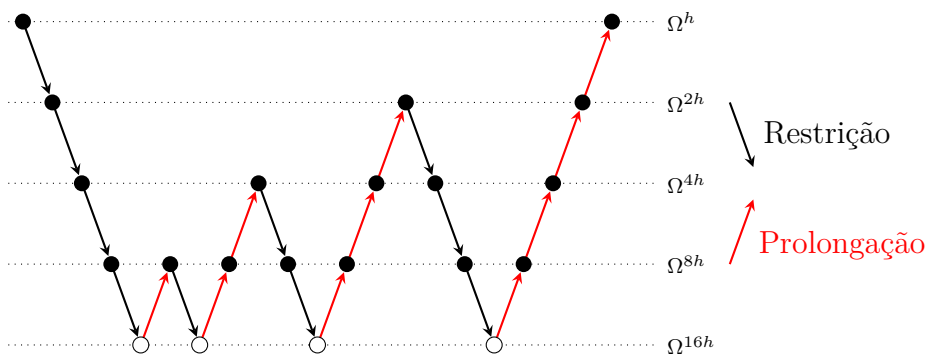
A sequência com que as malhas são percorridas dá origem aos chamados ciclos de suavização. Wesseling (1992), Briggs et al. (2000) e Trottenberg et al. (2001) apresentaram a generalização dos ciclos  $V$ ,  $F$  e  $W$  como parte de uma família ciclos chamada de  $\mu$ -ciclo. As Fig. 17(a), Fig. 17(b) e Fig. 17(c) ilustram cada um desses ciclos  $V$ ,  $F$  e  $W$ , respectivamente.

O número de suavizações, ou iterações do suavizador ( $\nu$ ), geralmente depende do suavizador e não precisa ser necessariamente o mesmo em todos os níveis de malha, tanto no processo de restrição quanto na prolongação. Define-se como  $\nu_1$  o número de suavizações realizadas quando ocorre o processo de restrição (pré-suavização) e  $\nu_2$  o número de suavizações realizadas quando ocorre o processo de prolongação (pós-suavização). Briggs et al. (2000) afirmam que para problemas que não apresentam dificuldades de convergência, são empregadas entre 1 e 3 suavizações.

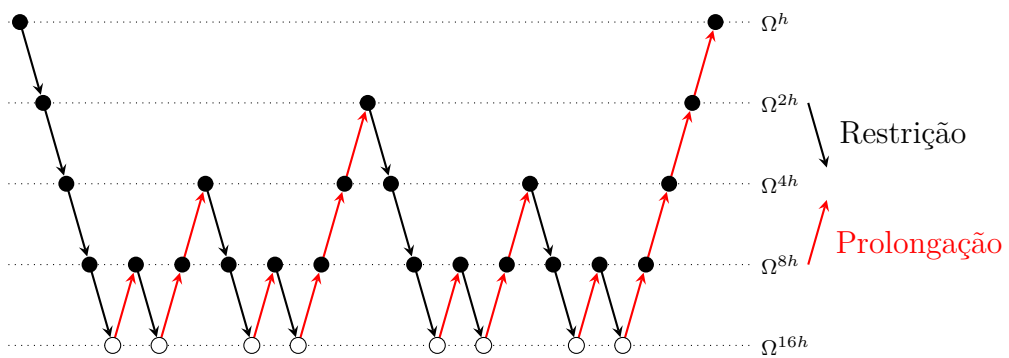
O algoritmo 1 mostra um esquema para o ciclo  $V$  ou o ciclo  $W$ . Se  $\mu = 1$  o algoritmo 1 realiza um ciclo  $V$  e se  $\mu = 2$  realiza um ciclo  $W$ . O algoritmo 2 mostra um esquema para o ciclo  $F$ .



(a) Ciclo  $V$ .



(b) Ciclo  $F$ .



(c) Ciclo  $W$ .

Figura 17 – Estrutura dos ciclos  $V$ ,  $F$  e  $W$ . O símbolo  $(\bullet)$  representa suavização e o símbolo  $(\circ)$  a solução exata.

**Algoritmo 1:** MG- $\mu$ -ciclo ( $l$ )

---

```

if  $l = L_{max}$  é o nível de malha mais grossa then
  Resolva o sistema  $A_l v^{(l)} = f^{(l)}$  na malha grossa  $\Omega^{2^{l-1}h}$ .
else
  Suavize  $\nu_1$  vezes  $A_l v^{(l)} = f^{(l)}$  na malha  $\Omega^{2^{l-1}h}$ .
  Calcule e restrinja o resíduo:  $f^{(l+1)} = I_{2^{l-1}h}^{2^l h}(f^{(l)} - A_l v^{(l)})$ .
  for  $ciclo = 1 : \mu$  do
    Resolva no próximo nível: MG- $\mu$ -ciclo ( $l + 1$ ).
  end for
  Corrija usando interpolação:  $v^{(l)} \leftarrow v^{(l)} + I_{2^l h}^{2^{l-1} h} v^{(l+1)}$ .
  Suavize  $\nu_2$  vezes  $A_l v^{(l)} = f^{(l)}$  na malha  $\Omega^{2^{l-1}h}$ .
end if

```

---

**Algoritmo 2:** MG-Ciclos-F ( $l$ )

---

```

if  $l = L_{max}$  é o nível de malha mais grossa then
  Resolva o sistema  $A_l v^{(l)} = f^{(l)}$  na malha grossa  $\Omega^{2^{l-1}h}$ .
else
  Suavize  $\nu_1$  vezes  $A_l v^{(l)} = f^{(l)}$  na malha  $\Omega^{2^{l-1}h}$ .
  Calcule e restrinja o resíduo:  $f^{(l+1)} = I_{2^{l-1}h}^{2^l h}(f^{(l)} - A_l v^{(l)})$ .
  Resolva no próximo nível: MG-Ciclos-F ( $l + 1$ ).
  Corrija usando interpolação:  $v^{(l)} \leftarrow v^{(l)} + I_{2^l h}^{2^{l-1} h} v^{(l+1)}$ .
  Suavize  $\nu_2$  vezes  $A_l v^{(l)} = f^{(l)}$  na malha  $\Omega^{2^{l-1}h}$ .
  if  $l \neq 1$  then
    Resolva no próximo nível: MG- $\mu$ -ciclo ( $l$ ) usando o algoritmo 1 e  $\mu = 1$ .
  end if
end if

```

---

## 3 Modelos matemáticos e numéricos

Neste capítulo apresentam-se os modelos matemáticos e numéricos para o problema da difusão de calor (equação de Fourier).

Para os modelos numéricos, as aproximações espaciais são feitas através do Método das Diferenças Finitas (MDF) e a aproximação temporal é feita com os métodos de Euler ou Crank-Nicolson (CN).

### 3.1 Modelo matemático

O modelo matemático para o problema da difusão de calor dependente do tempo (regime transiente) é representado pela equação do calor (EDP parabólica)

$$\frac{\partial u}{\partial t} - \Delta u = f, \quad (3.1)$$

no domínio espacial dado por  $\Omega \subset \mathbb{R}^d$ , com  $d \in \{1, 2\}$ . Nesta notação  $x_1 = x$  e  $x_2 = y$ , o operador Laplaciano  $d$ -dimensional é dado por  $\Delta = \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2}$  e o intervalo de tempo ( $t$ ) considerado é  $(0, t_f]$ .

Supõe-se a condição inicial dada por

$$u(\mathbf{x}, 0) = \prod_{i=1}^d \sin(\pi x_i), \quad \mathbf{x} \in \Omega, \quad (3.2)$$

com  $\mathbf{x} = x_1$  e  $\Omega = [0, 1]$  se  $d = 1$  ou  $\mathbf{x} = (x_1, x_2)$  e  $\Omega = [0, 1] \times [0, 1]$  se  $d = 2$ . As condições de contornos são do tipo *Dirichlet*, ou seja,

$$u(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \partial\Omega, \quad 0 < t \leq t_f. \quad (3.3)$$

O termo fonte é dado por

$$f(\mathbf{x}, t) = (d\pi^2 - 1)e^{-t} \prod_{i=1}^d \sin(\pi x_i), \quad \mathbf{x} \in \Omega, \quad 0 < t \leq t_f, \quad (3.4)$$

e a solução analítica é

$$u(\mathbf{x}, t) = e^{-t} \prod_{i=1}^d \sin(\pi x_i), \quad \mathbf{x} \in \Omega, \quad 0 \leq t \leq t_f. \quad (3.5)$$

## 3.2 Modelo numérico

Para o modelo numérico, considera-se o modelo matemático apresentado na seção 3.1 e  $v$  como sendo uma aproximação da solução analítica  $u$ . Discretiza-se o domínio espacial através do MDF com aproximação espacial do tipo CDS, como descrito na seção 1.1. Faz-se a aproximação temporal e a conexão espacial e temporal usando um método de discretização implícita (seção 1.4), como por exemplo neste caso, os métodos de Euler Implícito e de Crank-Nicolson.

Para o caso do método de Euler, considerando-se  $d = 1$  e discretizando a Eq. (3.1) tem-se

$$\frac{v_i^{m+1} - v_i^m}{\tau} = \frac{v_{i-1}^{m+1} - 2v_i^{m+1} + v_{i+1}^{m+1}}{h^2} + f_i^{m+1}, \quad (3.6)$$

ou

$$v_i^{m+1} = v_i^m + \frac{\tau}{h^2} [v_{i-1}^{m+1} - 2v_i^{m+1} + v_{i+1}^{m+1}] + \tau f_i^{m+1}, \quad (3.7)$$

em que  $i$ ,  $i - 1$  e  $i + 1$  indicam a discretização espacial,  $m + 1$  e  $m$  indicam os passos de tempo atual e anterior, respectivamente. O tamanho do passo de tempo é dado por  $\tau$  e o comprimento da discretização espacial é dado por  $h$ .

Observa-se que os coeficientes da Eq. (3.7) dependem de um parâmetro que leva em consideração o tamanho do passo do tempo ( $\tau$ ) e o quadrado do comprimento da discretização espacial ( $h$ ). Esse parâmetro é chamado de *fator de anisotropia* (HORTON; VANDEWALLE, 1995) e é dado por

$$\lambda = \frac{\tau}{h^2}. \quad (3.8)$$

Substituindo  $\lambda$  e reorganizando os termos semelhantes na Eq. (3.7), tem-se com o método de Euler

$$(1 + 2\lambda) v_i^{m+1} = v_i^m + \lambda (v_{i-1}^{m+1} + v_{i+1}^{m+1}) + \tau f_i^{m+1}. \quad (3.9)$$

Para o Método de Crank-Nicolson (CN), a Eq. (3.1) é discretizada da seguinte forma

$$\frac{v_i^{m+1} - v_i^m}{\tau} = \frac{1}{2} \left( \frac{v_{i-1}^{m+1} - 2v_i^{m+1} + v_{i+1}^{m+1}}{h^2} + f_i^{m+1} \right) + \frac{1}{2} \left( \frac{v_{i-1}^m - 2v_i^m + v_{i+1}^m}{h^2} + f_i^m \right), \quad (3.10)$$

reorganizando os termos, tem-se com o método de CN

$$(1 + \lambda) v_i^{m+1} = \frac{\lambda}{2} (v_{i-1}^{m+1} + v_{i+1}^{m+1} + v_{i-1}^m + v_{i+1}^m) + (1 - \lambda) v_i^m + \frac{\tau}{2} (f_i^{m+1} + f_i^m). \quad (3.11)$$

Para associar os métodos de Euler Implícito e o método de Crank-Nicolson usou-se um esquema chamado de esquema  $\theta$ . Esse esquema associa o parâmetro de mistura ( $\theta$ )

aos esquemas de Euler Implícito e de Crank-Nicolson conforme indicado na Eq. (3.13) e é definido na forma

$$\frac{v_i^{m+1} - v_i^m}{\tau} = \theta \left( \frac{v_{i-1}^{m+1} - 2v_i^{m+1} + v_{i+1}^{m+1}}{h^2} + f_i^{m+1} \right) + (1 - \theta) \left( \frac{v_{i-1}^m - 2v_i^m + v_{i+1}^m}{h^2} + f_i^m \right), \quad (3.12)$$

em que  $0 \leq \theta \leq 1$  e

$$\theta = \begin{cases} \frac{1}{2}, & \text{método de Crank-Nicolson} \\ 1, & \text{método de Euler} \end{cases}. \quad (3.13)$$

Assim sendo, a Eq. (3.12) pode ser reescrita como

$$(1 + 2\lambda\theta) v_i^{m+1} = \lambda\theta (v_{i-1}^{m+1} + v_{i+1}^{m+1}) + \tau\theta f_i^{m+1} + \lambda(1-\theta) (v_{i-1}^m - 2v_i^m + v_{i+1}^m) + v_i^m + \tau(1-\theta) f_i^m. \quad (3.14)$$

O operador discreto para a equação do calor 1D usando o esquema  $\theta$  pode ser escrito em notação estêncil da seguinte forma

$$\begin{bmatrix} 0 & 0 & 0 \\ -\lambda\theta & 1 + 2\lambda\theta & -\lambda\theta \\ -\lambda(1-\theta) & 2\lambda(1-\theta) - 1 & -\lambda(1-\theta) \end{bmatrix}. \quad (3.15)$$

Nota-se que com  $\theta = 1$  esse estêncil representa o método de Euler implícito e se  $\theta = \frac{1}{2}$  representa o método de Crank-Nicolson. Como esse é um estêncil espaço/tempo e o problema é unidimensional, em cada linha representa-se a evolução no tempo (de baixo para cima) e em cada coluna, a discretização no espaço (da esquerda para a direita), por isso e pelo fato de não usar passos de tempos futuros, a primeira linha é nula.

Considerando-se  $d = 2$  e discretizando a Eq. (3.1) considerando o comprimento das discretizações espaciais nas direções  $x$  e  $y$  iguais a  $h_x = h_y = h$ , tem-se

$$\frac{v_{i,j}^{m+1} - v_{i,j}^m}{\tau} = \frac{v_{i-1,j}^{m+1} - 2v_{i,j}^{m+1} + v_{i+1,j}^{m+1}}{h^2} + \frac{v_{i,j-1}^{m+1} - 2v_{i,j}^{m+1} + v_{i,j+1}^{m+1}}{h^2} + f_{i,j}^{m+1}, \quad (3.16)$$

considerando o fator de anisotropia  $\lambda$  dado na Eq. (3.8) e reorganizando os termos, tem-se

$$(1 + 4\lambda)v_{i,j}^{m+1} = v_{i,j}^m + \lambda [v_{i,j-1}^{m+1} + v_{i-1,j}^{m+1} + v_{i+1,j}^{m+1} + v_{i,j+1}^{m+1}] + \tau f_{i,j}^{m+1}. \quad (3.17)$$

Usando o método de Crank-Nicolson, a Eq. (3.1) para o caso bidimensional é discretizada da seguinte forma

$$\begin{aligned} \frac{v_{i,j}^{m+1} - v_{i,j}^m}{\tau} &= \frac{1}{2} \left[ \frac{v_{i-1,j}^{m+1} - 2v_{i,j}^{m+1} + v_{i+1,j}^{m+1}}{h^2} + \frac{v_{i,j-1}^{m+1} - 2v_{i,j}^{m+1} + v_{i,j+1}^{m+1}}{h^2} + f_{i,j}^{m+1} \right] + \\ &\frac{1}{2} \left[ \frac{v_{i-1,j}^m - 2v_{i,j}^m + v_{i+1,j}^m}{h^2} + \frac{v_{i,j-1}^m - 2v_{i,j}^m + v_{i,j+1}^m}{h^2} + f_{i,j}^m \right]. \end{aligned} \quad (3.18)$$



Reorganizando os termos, fica

$$\begin{aligned}
(1 + 2\lambda) v_{i,j}^{m+1} &= \frac{\lambda}{2} [v_{i,j-1}^{m+1} + v_{i-1,j}^{m+1} + v_{i+1,j}^{m+1} + v_{i,j+1}^{m+1}] + \\
&+ \frac{\lambda}{2} [v_{i,j-1}^m + v_{i-1,j}^m + v_{i+1,j}^m + v_{i,j+1}^m] + \\
&+ [1 - 2\lambda] v_{i,j}^m + \frac{\tau}{2} [f_{i,j}^{m+1} + f_{i,j}^m].
\end{aligned} \tag{3.19}$$

Para associar os métodos de Euler Implícito e o método de Crank-Nicolson usando o esquema  $\theta$ , usam-se os valores de  $\theta$  descritos na Eq. (3.13) e faz-se na Eq. (3.18) a seguinte alteração

$$\begin{aligned}
\frac{v_{i,j}^{m+1} - v_{i,j}^m}{\tau} &= \theta \left[ \frac{v_{i-1,j}^{m+1} - 2v_{i,j}^{m+1} + v_{i+1,j}^{m+1}}{h^2} + \frac{v_{i,j-1}^{m+1} - 2v_{i,j}^{m+1} + v_{i,j+1}^{m+1}}{h^2} + f_{i,j}^{m+1} \right] + \\
&+ (1 - \theta) \left[ \frac{v_{i-1,j}^m - 2v_{i,j}^m + v_{i+1,j}^m}{h^2} + \frac{v_{i,j-1}^m - 2v_{i,j}^m + v_{i,j+1}^m}{h^2} + f_{i,j}^m \right].
\end{aligned} \tag{3.20}$$

Reorganizando os termos, tem-se

$$\begin{aligned}
(1 + 4\lambda\theta) v_{i,j}^{m+1} &= \lambda\theta [v_{i,j-1}^{m+1} + v_{i-1,j}^{m+1} + v_{i+1,j}^{m+1} + v_{i,j+1}^{m+1}] + \\
&+ \lambda(1 - \theta) [v_{i,j-1}^m + v_{i-1,j}^m + v_{i+1,j}^m + v_{i,j+1}^m] + \\
&+ [1 - 4\lambda(1 - \theta)] v_{i,j}^m + \tau [\theta f_{i,j}^{m+1} + (1 - \theta) f_{i,j}^m].
\end{aligned} \tag{3.21}$$

O operador discreto para a equação do calor 2D usando o esquema  $\theta$  pode ser escrito em notação estêncil da seguinte forma

$$\left[ \begin{bmatrix} 0 & -\lambda(1 - \theta) & 0 \\ -\lambda(1 - \theta) & 4\lambda(1 - \theta) - 1 & -\lambda(1 - \theta) \\ 0 & -\lambda(1 - \theta) & 0 \end{bmatrix} \begin{bmatrix} 0 & -\lambda\theta & 0 \\ -\lambda\theta & 1 + 4\lambda\theta & -\lambda\theta \\ 0 & -\lambda\theta & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right]. \tag{3.22}$$

Nota-se que com  $\theta = 1$  esse estêncil representa o método de Euler implícito e se  $\theta = \frac{1}{2}$  representa o método de Crank-Nicolson. Neste caso, cada um dos estêncis representa as posições espaciais para um passo de tempo, por isso o estêncil da direita é nulo. Assim, em cada estêncil os termos da esquerda para a direita representam a direção  $x$  e de baixo para cima representam a direção  $y$ .

## 4 Métodos de solução

Neste capítulo serão descritos os métodos (ou técnicas) *Time-Stepping*, *Waveform Relaxation* e *Space-Time*, usados para a solução de EDPs parabólicas.

Por simplicidade, para a descrição dessas técnicas, segue-se como base o problema modelo padrão para classe de EDPs parabólicas, dado pela equação do calor (ou equação de Fourier) bidimensional descrita na Eq. (3.1) como

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f. \quad (4.1)$$

Este problema combina características de EDPs elípticas estacionárias para o caso de se considerar um passo de tempo fixo (gerando o método *Time-Stepping*), equações diferenciais ordinárias (EDOs) transientes para o caso de se considerar um ponto espacial fixo e todos os passos de tempo (gerando o método *Waveform Relaxation*) e uma descrição espaço-tempo com suavizador por pontos (gerando o método *Space-Time*). Será exposta também a descrição do método *Waveform Relaxation* para o problema da poroelasticidade.

### 4.1 Método *Time-Stepping*

Segundo [Tannehill et al. \(1997\)](#) alguns estudos têm dado ênfase para algoritmos que tratam a equação parabólica como uma sequência de equações elípticas em cada passo de tempo. Este método é chamado de método *Time-Stepping*.

Nesse método, resolve-se o sistema discretizado em cada passo de tempo de forma consecutiva até o passo de tempo final. Assim, em cada passo de tempo, pode-se usar algum método para o problema estacionário. Com isso, a estrutura do sistema a ser resolvido é correspondente à discretização de EDPs elípticas ([TANNEHILL et al., 1997](#); [BURDEN; FAIRES, 2016](#); [LENT, 2006](#); [HIRSCH, 2007](#)).

A principal característica do método *Time-Stepping* é mostrado na Fig. 18, retirada de ([VANDEWALLE, 1993](#), p. 18). Em cada passo de tempo é usada a solução do passo de tempo anterior como estimativa inicial e resolve-se o sistema de equações correspondente no passo de tempo atual, considerando-se um sistema estacionário. Com isso, esse sistema de equações diferenciais parciais parabólicas pode ser comparado às equações diferenciais parciais elípticas, para as quais o método *multigrid* é comprovadamente eficiente ([WESSELING, 1992](#); [BRIGGS et al., 2000](#); [TROTTEBERG et al., 2001](#)). Dessa forma, na programação do método *Time-Stepping* deve-se fazer um laço externo para os passos de tempo ( $m + 1$ ), até o número de passos de tempo  $N_t$ , e um laço interno para as variáveis espaciais ( $h$  ou  $i, j$ ). Esse procedimento é especificado através do algoritmo 3.

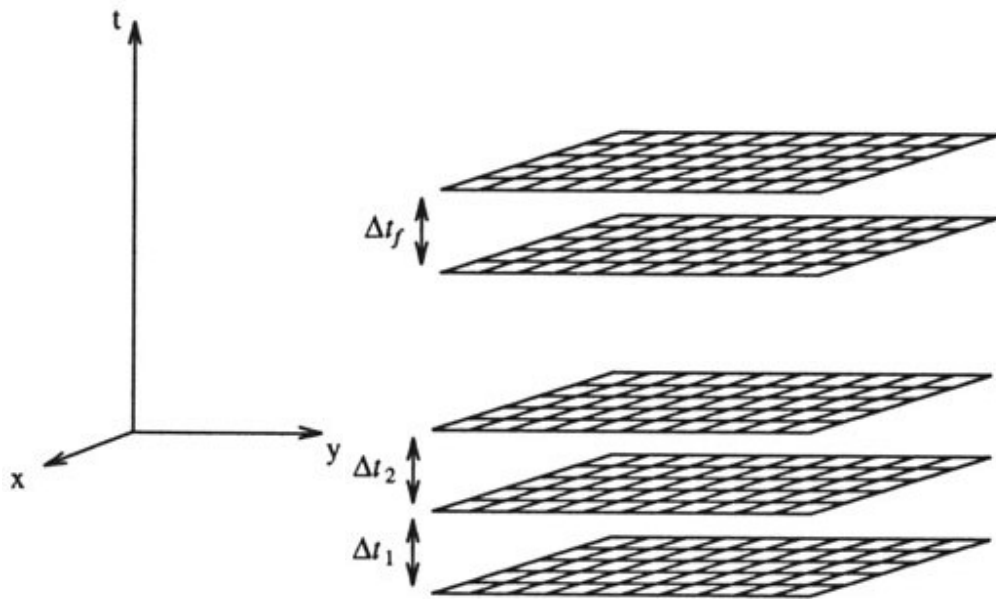


Figura 18 – Procedimento de atualização das incógnitas no método *Time-Stepping* (VANDEWALLE, 1993, p. 18) para o caso 2D.

---

**Algoritmo 3:** Método *Time-Stepping* em uma malha

---

Entre com os dados de entrada, condição inicial e condições de contornos.

**for**  $m = 1 : N_t$  **do**

**while** Não alcançar algum critério de parada **do**

        Suavize o sistema correspondente no passo de tempo  $m + 1$ .

**end while**

**end for**

---

Devido à característica do método *Time-Stepping*, para os casos em que se usa a técnica *multigrid*, além da malha fina  $\Omega^h$ , as malhas mais grossas  $\Omega^{2h}, \Omega^{4h}, \dots, \Omega^{lh}$ , com  $l = 2^{(L_{\max}-1)}$ , em que  $L_{\max}$  é o número máximo de malhas a serem utilizadas, são discretizadas apenas nas variáveis espaciais, ou seja, em cada passo de tempo fixo. Por exemplo, considerando-se a discretização das variáveis espaciais  $x, y$  na malha  $\Omega^h$  com  $33 \times 33$  pontos, as malhas mais grossas  $\Omega^{2h}, \Omega^{4h}, \Omega^{8h}$  e  $\Omega^{16h}$  terão  $17 \times 17, 9 \times 9, 5 \times 5$  e  $3 \times 3$  pontos, respectivamente.

Pode-se especificar o método *multigrid* com os ciclos  $V(\nu_1, \nu_2), F(\nu_1, \nu_2)$  ou  $W(\nu_1, \nu_2)$  para o método *Time-Stepping* através do algoritmo 4. A sigla **MG-ciclo** representa qualquer um dos ciclos ( $V, W$  ou  $F$ ) definidos anteriormente nos Alg. 1 e Alg. 2. A partir de agora, tal símbolo será usado nesses algoritmos.

**Algoritmo 4:** Método *Time-Stepping - multigrid*


---

Entre com os dados de entrada, condição inicial e condições de contornos.  
**for**  $m = 1 : N_t$  **do**  
     **while** Não alcançar algum critério de parada **do**  
         Um ciclo *multigrid* com **MG-ciclo** (1).  
     **end while**  
**end for**

---

## 4.2 Método *Waveform Relaxation*

O método *Waveform Relaxation* é um método iterativo contínuo no tempo e foi desenvolvido para resolver numericamente grandes sistemas de equações diferenciais ordinárias (ODEs) (LELARASMEE et al., 1982; LUBICH; OSTERMANN, 1987; VANDEWALLE, 1993). No entanto, esse método também pode ser aplicado para resolver equações diferenciais parciais (EDPs) dependentes do tempo, em que as EDPs são transformadas em um grande conjunto de equações diferenciais ordinárias (EDOs). Inicialmente é feita uma discretização do problema apenas espacialmente, transformando a Eq. (4.1) em um conjunto de EDOs da forma

$$\frac{dv_h}{dt} = G_h(v_h, f_h), \quad (4.2)$$

em que  $G_h$  está em função dos valores de  $v_h$  e  $f_h$  que são funções ou vetores contendo informações do tempo para cada posição espacial  $h$ . Por isso, assumindo-se conhecidos os valores das posições espaciais, pode-se considerar  $G_h$  um sistema de EDOs em  $t$ . Esse método iterativo, calculado em função do tempo, é conhecido como MOL (em inglês, *Method of Lines*). O método *Waveform Relaxation* é a forma de esse tipo de sistema por relaxação (VANDEWALLE, 1993; LENT, 2006; FALGOUT et al., 2017).

Cada componente do sistema dado na Eq. (4.2) pode ser escrito como uma EDO, da seguinte forma

$$\left\{ \begin{array}{ll} \frac{d}{dt} v_1 = G_1(v_1, v_2, \dots, v_d, f_1) & v_1(0) = v_1^0 \\ \frac{d}{dt} v_2 = G_2(v_1, v_2, \dots, v_d, f_2) & \text{com } v_2(0) = v_2^0 \\ \vdots & \\ \frac{d}{dt} v_d = G_d(v_1, v_2, \dots, v_d, f_d) & v_d(0) = v_d^0 \end{array} \right., \quad (4.3)$$

em que  $d$  é a dimensão do sistema linear. A notação  $v_i(0) = v_i^0$ ,  $1 \leq i \leq d$ , indica as condições iniciais ( $t=0$ ) em cada ponto da discretização espacial. Esse procedimento de solução pode ser observado na Fig. 19, conforme Lent (2006, p. 94) para o caso 1D.

Considerando-se o método de Gauss-Seidel (GS) e a ordem Lexicográfica, o sistema

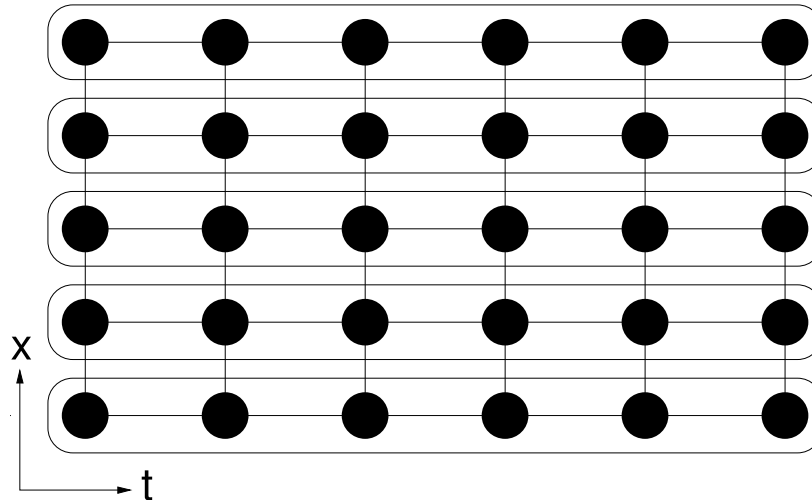


Figura 19 – Procedimento de atualização das incógnitas no método *Waveform Relaxation* (LENT, 2006, p. 94) para o caso 1D.

dado pelas Eqs. (4.3) pode ser escrito como

$$\left\{ \begin{array}{l} \frac{d}{dt} v_1^{\nu+1} = G_1(v_1^{\nu+1}, v_2^\nu, \dots, v_d^\nu, f_1) \\ \frac{d}{dt} v_2^{\nu+1} = G_2(v_1^{\nu+1}, v_2^{\nu+1}, v_3^\nu, \dots, v_d^\nu, f_2) \\ \vdots \\ \frac{d}{dt} v_d^{\nu+1} = G_d(v_1^{\nu+1}, v_2^{\nu+1}, \dots, v_d^{\nu+1}, f_d) \end{array} \right. \text{ com } \begin{array}{l} v_1^\nu(0) = v_1^{0,\nu} \\ v_2^\nu(0) = v_2^{0,\nu} \\ \vdots \\ v_d^\nu(0) = v_d^{0,\nu} \end{array}. \quad (4.4)$$

Observa-se a relação entre as equações do sistema (4.4) e as linhas do tempo na Fig. 19, ou seja, para cada ponto da discretização espacial é resolvida uma EDO temporal. Os métodos de aproximação da derivada temporal podem ser o método de Euler implícito e de Crank-Nicolson, entre outros. Após realizar esse procedimento em todos os pontos espaciais, considera-se realizada uma iteração  $\nu$ .

Considerando-se que a principal característica do método *Waveform Relaxation*, em que para cada ponto da discretização espacial resolve-se uma EDO temporal, como mostra a Fig. 19, na programação computacional deve-se fazer um laço externo para as iterações  $\nu$ , um laço intermediário para percorrer os pontos da discretização espacial  $h$  e um laço interno para os passos de tempo  $m$  (ou para resolver a EDO temporal). Esse procedimento é especificado através do algoritmo 5, baseado em Vandewalle (1993, p. 26) para o método *Waveform Relaxation* com *singlegrid* (método de malha única).

Se considerar o uso do método *Waveform Relaxation* com *multigrid*, deve-se seguir a teoria descrita no capítulo 2. Vandewalle (1993) comenta que o princípio do método *multigrid* para problemas dependentes do tempo é essencialmente da mesma forma que para os métodos clássicos de relaxamento.

**Algoritmo 5:** Método *Waveform Relaxation* em uma malha.

---

Entre com os dados de entrada, condição inicial e condições de contornos.  
**while** Não alcançar algum critério de parada **do**  
  **for**  $i = 1 : d$  **do**  
    Resolva a EDO discretizada correspondente à posição espacial  $i$  no sistema de Eq. (4.4).  
  **end for**  
**end while**

---

Devido à filosofia do método *Waveform Relaxation*, no caso de se usar a técnica *multigrid*, a malha fina  $\Omega^h$  e as malhas mais grossas  $\Omega^{2h}, \Omega^{4h}, \dots, \Omega^{lh}$ , com  $l = 2^{(L_{\max}-1)}$ , são simultaneamente discretizadas no espaço e aproximadas no tempo, porém, o processo de engrossamento das malhas ocorre somente nas direções espaciais. Por exemplo, considerando-se engrossamento padrão  $q = 2$ , a discretização das variáveis espaciais  $x, y$  e o número de passos no tempo  $t$  na malha  $\Omega^h$  com  $x \times y \times t$  com  $33 \times 33 \times 100$  pontos, as malhas mais grossas  $\Omega^{2h}, \Omega^{4h}, \Omega^{8h}$  e  $\Omega^{16h}$  terão  $17 \times 17 \times 100$ ,  $9 \times 9 \times 100$ ,  $5 \times 5 \times 100$  e  $3 \times 3 \times 100$  pontos, respectivamente. Com isso, os valores dos resíduos são calculados e restritos e a correção é prolongada e suavizada em todos os pontos espaciais e passos de tempo, simultaneamente.

No caso da equação do calor, os suavizadores espaciais mais comuns nesse método são Gauss-Seidel (lexicográfico ou *red-black*), o método de Jacobi ponderado e os métodos de relaxação por linhas. Como já foi mencionado, os métodos de aproximação da derivada temporal podem ser os métodos de Euler implícito e de Crank-Nicolson, entre outros. Pode-se especificar o método *multigrid* para o esquema CS com ciclos  $V(\nu_1, \nu_2)$ ,  $F(\nu_1, \nu_2)$  ou  $W(\nu_1, \nu_2)$  através do algoritmo 6 (VANDEWALLE, 1993; FALGOUT et al., 2017).

**Algoritmo 6:** Método *Waveform Relaxation - multigrid*


---

Entre com os dados de entrada, condição inicial e condições de contornos.  
**while** Não alcançar algum critério de parada **do**  
  Um ciclo *multigrid* com **MG-ciclo** (1), e considerando engrossamento apenas no espaço em todos os passos de tempo. Para o processo de suavização, em cada nível de malha use o método descrito no algoritmo 5.  
**end while**

---

Para as equações da poroelasticidade, devido à presença de pontos de sela no sistema de equações, considerou-se o uso do suavizador Vanka de três pontos. Esse suavizador atualiza simultaneamente em cada passo de iteração um bloco de três incógnitas, na forma de Gauss-Seidel. Dessa maneira, todas as incógnitas do deslocamento são calculadas juntamente com a incógnita da pressão, posicionada no ponto central, de forma simultânea em cada passo de tempo. Esse cálculo se dá consecutivamente em todos os passos de tempo através do método *Waveform Relaxation*. Para o caso unidimensional, as incógnitas,  $u_{i-1}$ ,  $p_i$  e  $u_{i+1}$  são resolvidas simultaneamente. Esse procedimento será chamado de suavização

*Waveform Relaxation* Vanka de três pontos, e a ideia geométrica está descrita na Fig. 20. No caso bidimensional, as incógnitas,  $u_{i,j-1}$ ,  $u_{i-1,j}$ ,  $p_{i,j}$ ,  $u_{i+1,j}$  e  $u_{i,j+1}$  são resolvidas simultaneamente. Esse procedimento será chamado de suavização *Waveform Relaxation* Vanka de cinco pontos, e a ideia geométrica está descrita na Fig. 21.

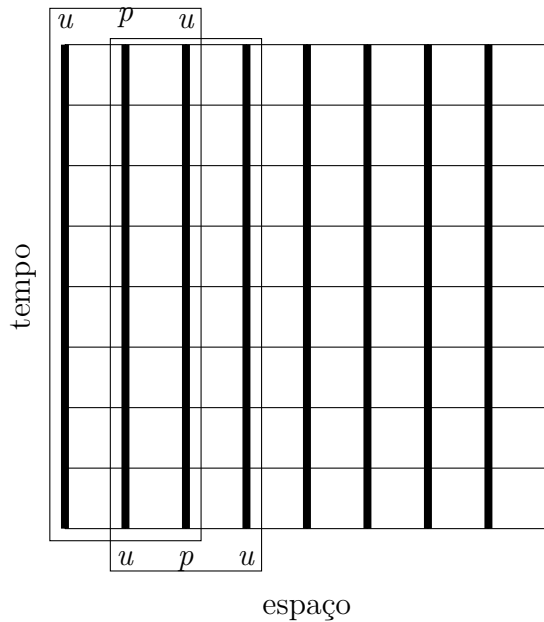


Figura 20 – Método *Waveform Relaxation* Vanka de três pontos, usada para o problema da poroelasticidade 1D.

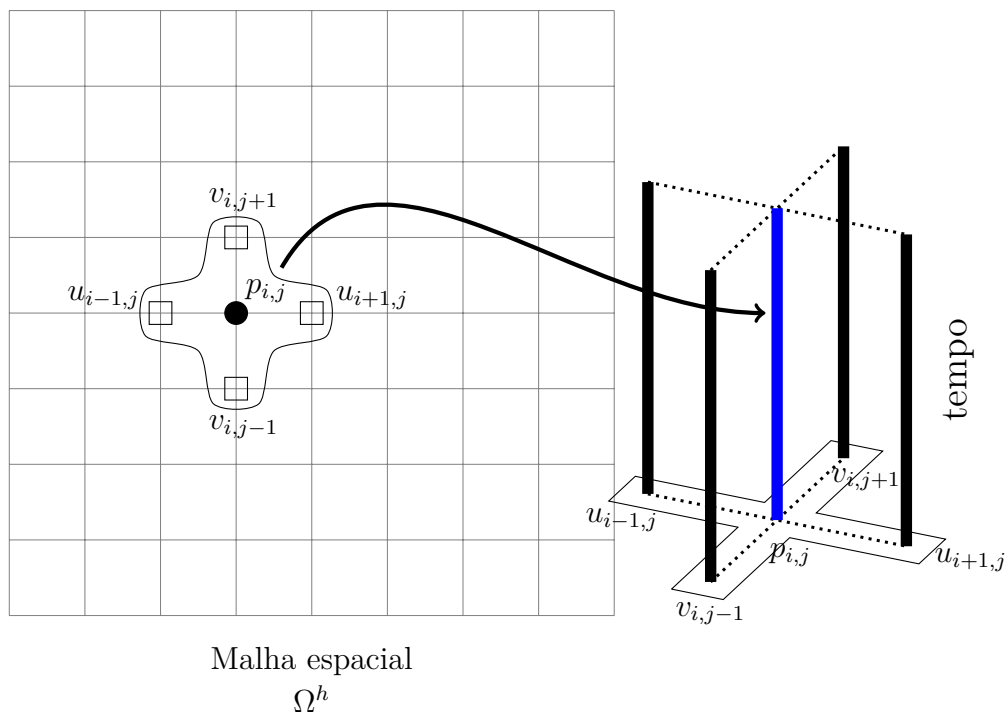


Figura 21 – Método *Waveform Relaxation* Vanka de cinco pontos, usada para o problema da poroelasticidade 2D.

O algoritmo do método *multigrid* com o suavizador Vanka para as equações da poroelasticidade 1D e 2D segue a mesma ideia do algoritmo mostrado para a equação do calor, algoritmo 6.

### 4.3 Método *Space-Time*

O *Space-Time* é um método que se caracteriza por usar um suavizador por pontos, como o Gauss-Seidel *red-black*, sendo que em cada iterada atualizam-se todos os pontos no espaço e no tempo. O algoritmo 7 descreve o uso desse método em uma malha (*singlegrid*).

---

#### Algoritmo 7: Método *Space-Time* em uma malha

---

Entre com os dados de entrada, condição inicial e condições de contornos.

**while** Não alcançar algum critério de parada **do**

Suavize (ponto-a-ponto) o sistema correspondente em todos os pontos no espaço e no tempo.

**end while**

---

Em particular, o suavizador frequentemente usado no algoritmo 7 é o Gauss-Seidel com ordenação *red-black* nas direções espaciais.

Com o método *multigrid*, Horton e Vandewalle (1995) comentam que o *Space-Time* se caracteriza por adotar uma estratégia de semiengrossamento no espaço ou no tempo, e que tal estratégia depende de um parâmetro chamado *fator de anisotropia* ( $\lambda_l$ ) que relaciona o tamanho do passo temporal e espacial em cada nível de malha  $l$ . A razão de engrossamento usada para ambos os semiengrossamentos é  $q = 2$ . O valor de  $\lambda$  que define o momento do semiengrossamento espacial ou temporal é denominado  $\lambda_{crit}$  e em Horton e Vandewalle (1995) e no presente trabalho foi determinado através da LFA. Segundo Horton e Vandewalle (1995) e Falgout et al. (2017) seu valor é  $\lambda_{crit} = 0,60$ .

A estratégia de semiengrossamentos adotada no método *Space-Time* está descrita no algoritmo 8.

---

#### Algoritmo 8: Estratégia de engrossamento no método *Space-Time* - *multigrid*

---

**if**  $\lambda < \lambda_{crit}$  **then**

Faça semiengrossamento na direção temporal.

**else**

Faça semiengrossamento nas direções espaciais.

**end if**

---

Como  $\lambda_l = \frac{\tau_l}{h_l^2}$ , quando é usado semiengrossamento espacial ( $h_{l+1} = 2h_l$ ), o valor de  $\lambda_{l+1}$  na malha imediatamente mais grossa é  $\lambda_{l+1} = \frac{1}{4}\lambda_l$ . Se o engrossamento ocorrer na direção temporal ( $\tau_{l+1} = 2\tau_l$ ), o valor de  $\lambda_{l+1}$  na malha imediatamente mais grossa é



$\lambda_{l+1} = 2\lambda_l$ . Dessa forma, pode-se observar que o valor de  $\lambda_l$  pode aumentar ou diminuir dependendo do tipo de semiengrossamento escolhido. Após restar apenas uma incógnita (ou 3 pontos) em uma direção, o engrossamento ocorrerá na outra direção até restar apenas uma incógnita ou 3 pontos naquela direção. Com isso, o número máximo de malhas possíveis nesse método é dado por  $L_{\max} = nm_x + nm_t - 1$ , em que  $nm_x$  representa o número máximo de malhas na direção espacial e  $nm_t$  é o número máximo de malhas na direção temporal.

Os operadores de prolongação ( $I_{2h}^h$ ) e restrição ( $I_h^{2h}$ ) são escolhidos de acordo com Horton e Vandewalle (1995). Para o caso 1D, se o engrossamento ocorrer na direção espacial, os operadores são dados por

$$I_{2h}^h = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}_{2h}^h, \quad I_h^{2h} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h}. \quad (4.5)$$

e se o engrossamento ocorrer na direção temporal, os operadores são dados por

$$I_{2h}^h = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h}^h, \quad I_h^{2h} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}_h^{2h}. \quad (4.6)$$

Como esses estêncis são do tipo espaço/tempo e o problema é unidimensional, como já mencionado na seção 3.2, em cada linha representa-se a evolução no tempo (de baixo para cima) e em cada coluna, a discretização no espaço (da esquerda para a direita). Por isso, os estêncis descritos na Eq. (4.5) apresentam valores apenas na linha do meio, ou seja, apenas nas posições espaciais e no passo de tempo atual, enquanto que os estêncis descritos na Eq. (4.6) apresentam valores apenas na coluna do meio, ou seja, apenas nas posições temporal e para uma posição espacial. Observa-se que como não se usam valores relacionados a tempos futuros, os estêncis descritos na Eq. (4.6) têm uma linha nula.

Para o caso 2D, se o engrossamento ocorrer na direção espacial, os operadores são dados por

$$I_{2h}^h = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h}^h & \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^h & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h}^h \end{bmatrix}, \quad (4.7)$$

$$I_h^{2h} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h} & \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h} \end{bmatrix},$$

e se o engrossamento ocorrer na direção temporal, os operadores são dados por

$$I_{2h}^h = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h} & \begin{bmatrix} 1 \\ 4 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{2h} \end{bmatrix}, \quad (4.8)$$

$$I_h^{2h} = \frac{1}{2} \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h \end{bmatrix}^{2h},$$

Observa-se que a notação de estêncil nas Eq. (4.7) e Eq. (4.8) corresponde a uma sequência de estêncis aplicados em sucessivos passos de tempo, do mais baixo (estêncil esquerdo) para o mais alto (estêncil direito).

O fato desse método trabalhar com o espaço e o tempo simultaneamente possibilita o desenvolvimento de algoritmos paralelizáveis mais eficientes, diminuindo assim o tempo necessário para encontrar uma solução (PRIETO et al., 2001; FALGOUT et al., 2017).

O algoritmo 9 expressa os ciclos  $V(\nu_1, \nu_2)$ ,  $W(\nu_1, \nu_2)$  ou  $F(\nu_1, \nu_2)$  para o método *Space-Time - multigrid*.

---

**Algoritmo 9:** Método *Space-Time - multigrid*

---

Entre com os dados de entrada, condição inicial e condições de contornos.

**while** Não alcançar algum critério de parada **do**

Um ciclo *multigrid* com **MG-ciclo** (1) e respeitando a estratégia de engrossamento descrita no algoritmo 8. Caso não seja possível engrossar em uma direção, utilize engrossamento em outra direção até se obter a malha mais grossa possível.

**end while**

---

# Referências

- BRANDT, A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, v. 31, n. 138, p. 333–390, 1977.
- BRIGGS, W. L.; HENSON, V. E.; MCCORMICK, S. F. *A Multigrid Tutorial*. 2nd. ed. Philadelphia: SIAM, 2000.
- BURDEN, R. L.; FAIRES, J. D. *Numerical Analysis*. 10nd. ed. Boston: Cengage Learning, 2016.
- FALGOUT, R. D.; FRIEDHOFF, S.; KOLEV, T. V.; MACLACHLAN, S. P.; SCHRODER, J. B.; VANDEWALLE, S. Multigrid methods with space-time concurrency. *Computing and Visualization in Science*, v. 18, p. 123–143, 2017.
- FEDORENKO, R. P. On the speed of convergence of an iteration process. *USSR Comput. Math. and Math. Phys.*, n. 4, p. 227, 1964.
- FERZIGER, J. H.; PERIC, M. *Computational Methods for Fluid Dynamics*. 3nd. ed. New York: Springer, 2002.
- HACKBUSCH, W. *Multi-grid Methods and Applications*. Berlin: Springer-Verlag, 1985.
- HIRSCH, C. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. 2nd. ed. New York, USA: John Wiley & Sons, 2007.
- HORTON, G.; VANDEWALLE, S. A space time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.*, v. 16, n. 4, p. 848–864, 1995.
- JANSSEN, J. *Acceleration of Waveform Relaxation Methods for Linear Ordinary and Partial Differential Equations*. Tese (Doutorado) — Katholieke Universiteit Leuven, Heverlee, Belgium, 1997.
- KREYSZIG, E. *Advanced Engineering Mathematics*. 8nd. ed. Wiley: Springer, 1999.
- LELARASMEE, E.; RUEHLI, A. E.; VINCENNELLI, S. A. L. The waveform relaxation method for time domain analysis of large scale integrated circuits theory and applications. *IEEE Trans. Comput. Aided Design Integr. Circ. Systems*, n. 1, p. 131–145, 1982.
- LENT, J. V. *Multigrid Method for Time-Dependent Partial Differential Equations*. Tese (Doutorado) — Katholieke Universiteit Leuven, 2006.
- LUBICH, C.; OSTERMANN, A. Multigrid dynamic iteration for parabolic equations. *BIT*, v. 27, n. 2, p. 216–234, 1987.
- MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacionais*. 2nd edição. ed. Rio de Janeiro: LTC, 2004.
- MARCHI, C. H. *Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, 2001.

- PINTO, M. A. V.; MARCHI, C. H. Optimum parameters of a geometric multigrid for the two-dimensional laplace's equation. *Proceedings of COBEM*, 2007.
- PRIETO, M.; SANTIAGO, R.; ESPADAS, D.; LLORENTE, I. M.; TIRADO, F. Parallel multigrid for anisotropic elliptic equations. *Journal of Parallel and Distributed Computing*, v. 61, p. 96–114, 2001.
- ROACHE, P. J. *Fundamentals of Computational Fluid Dynamics*. Albuquerque, USA: Hermosa Publishers, 1998.
- RUGGIERO, M. A. G.; LOPES, V. L. d. R. *Cálculo Numérico - Aspectos Teóricos e Computacionais*. 2nd edição. ed. São Paulo: Pearson, 1996.
- SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2nd. ed. Philadelphia: PWS, 2003.
- STÜBEN, K.; TROTTEBERG, U. Multigrid methods: Fundamental algorithms, model problem analysis and applications. In: HACKBUSCH, W.; TROTTEBERG, U. (Ed.). *Proceedings of the Conference Held at Köln-Porz*. New York: Springer-Verlag, 1981. p. 1–176.
- STRIKWERDA, J. C. *Finite Difference Schemes and Partial Differential Equations*. 2nd. ed. Pacific Grove, CA: SIAM, 1989.
- TANNEHILL, J. C.; ANDERSON, D. A.; PLETCHER, R. H. *Computational Fluid Mechanics and Heat Transfer*. Philadelphia: Taylor & Francis, 1997.
- TROTTEBERG, U.; OOSTERLEE, C.; SCHÜLLER, A. *Multigrid*. San Diego: Academic Press, 2001.
- VANDEWALLE, S. *Parallel Multigrid Waveform Relaxation for Parabolic Problems*. Stuttgart: Teubner, 1993.
- WESSELING, P. *An Introduction to Multigrid Methods*. Chichester: John Wiley & Sons, 1992.
- WIENANDS, R.; JOPPICH, W. *Practical Fourier Analysis for Multigrid Methods*. Washington, USA: Chapman & Hall/CRC Press, 2005. v. 4.