Programas em MATLAB para Implementação
de Exemplos em Discretização de
Equações Diferenciais Parciais

ALESSANDRO ALVES SANTANA

Orientador: PROF. DR. JOSÉ ALBERTO CUMINATO

*Monografia apresentada ao Instituto de Ciências Matemáticas e de
Computação - USP, como parte dos requisitos para a obtenção do título
de Mestre em Ciências - Área de Ciências da Computação e Matemática
Computacional.*

USP - São Carlos
Agosto de 1998

*Ao meu pai, minha mãe, meus irmãos e a Deus,
que nos ilumina sempre o caminho.*

Agradecimentos

Agradeço ao Prof. Dr. José Alberto Cuminato, meu orientador, por todo apoio e esclarecimento para o desenvolvimento desse trabalho, o qual me enriqueceu muito. À minha família pelo apoio, aos meus amigos do mestrado pelo companheirismo, à minha namorada Teodora pela força, em suma, a todos aqueles que passaram comigo por mais essa etapa da minha vida, etapa esta que será o alicerce para o meu futuro.

Resumo

O presente trabalho tem como finalidade o ensino, e consistiu na implementação de uma classe de métodos de resolução numérica para equações diferenciais parciais que constam em "Discretização de Equações Diferenciais Parciais: Técnica de Diferenças Finitas", Cuminato [7]. O software utilizado foi o MATLAB, e com ele foi desenvolvido um conjunto de programas que são acessados mediante uma interface padrão para entrada de dados e visualização de resultados. No que tange a visualização de resultados, o que foi desenvolvido permite ao usuário analisar os resultados, tanto através das aproximações obtidas como através dos gráficos das mesmas. Esses programas irão acompanhar o texto final de [7] para formar uma biblioteca que o acompanhará, para servir de apoio ao professor que vier a utilizá-lo.

Abstract

This work's main goal is teaching, it consists of the implementation of a class of numerical resolution methods for partial differential equations found on "Discretização de Equações Diferenciais Parciais: Técnica de Diferenças Finitas", Cuminato [7]. The implementation was done using MATLAB, by developing a set of programs with a standard interface for data entry and visualization of results. The user can visualize the results either graphically or textually. These programs will be incorporated into the final text of [7], forming a companion library, to assist the teacher who will use it.

Índice

1	Introdução	2
2	Conceitos Fundamentais	4
2.1	Introdução às EDPs	4
2.2	Classificação das EDPs	5
2.2.1	EDPs de Primeira Ordem	5
2.2.2	EDPs de Segunda Ordem	7
2.3	Problema Bem-Posto	9
2.4	Métodos de Diferenças Finitas	11
2.5	Aproximações das derivadas por diferenças finitas	12
2.6	Operadores Diferenças Finitas	14
2.7	Estabilidade	17
3	Equações Elípticas	18
3.1	Introdução	18
3.2	Métodos de Diferenças Finitas	19
3.2.1	Convergência	20
3.2.2	Condições de Fronteira de Dirichlet	23
3.2.3	Condição de Fronteira de Neumann	33
3.3	Condições de Fronteira em Geral	42
4	Equações Parabólicas	46
4.1	Introdução	46
4.2	Métodos de Diferenças Finitas	47
4.3	Métodos para a Equação do Calor em 1D	47
4.3.1	Método Explícito	47
4.3.2	Método Implícito	56

4.3.3	Método de Crank-Nicolson	61
4.3.4	Método Saul'yev	66
4.3.5	Método Hopscotch	71
4.4	Métodos para Equações Parabólicas com Coeficientes Variáveis	74
4.4.1	Método de Lees	75
4.4.2	Método das Linhas	81
4.4.3	Método Box	84
4.5	Métodos para a Equação do Calor em 2D	91
4.5.1	Método Explícito - 5 pontos	95
4.5.2	Método Explícito - 9 pontos	99
4.5.3	Método das Direções Alternadas - ADI	102
4.5.4	Método Localmente Unidimensional	107
5	Equações Hiperbólicas	113
5.1	Introdução	113
5.2	Características	114
5.3	Método das Características	118
5.4	Métodos de Diferenças Finitas	124
5.4.1	Estabilidade	124
5.5	Métodos de Diferenças Explícitos	126
5.5.1	Esquema Explícito de Primeira Ordem	126
5.5.2	Esquema de Lax-Wendroff	132
5.5.3	Esquema Leap-Frog	136
5.6	Métodos de Diferenças Implícitos	140
5.6.1	Esquema Implícito de Primeira Ordem	140
5.6.2	Esquema Implícito de Wendroff	145
5.7	Método de Diferenças Finitas para a Equação da Onda	148
6	MATLAB	155
7	Conclusão	157
A	Manual para Utilização do Software	158
A.1	Instalação	158
A.2	Como utilizar os programas	158

A.3 Exemplos	161
A.4 Observações	170
Bibliografia	171

Lista de figuras

2.1	Exemplo de uma malha	11
3.1	Ilustração da malha no domínio	19
3.2	Malha no domínio do problema de Dirichlet	24
3.3	Malha no domínio do problema de Neumann	33
3.4	Discretização na fronteira - Problema de Von Neumann	34
3.5	Malha em um domínio qualquer 1	42
3.6	Ampliação da região considerada 1	43
3.7	Malha em um domínio qualquer 2	44
3.8	Ampliação da região considerada 2	45
4.1	Malha definida no domínio do problema envolvendo a Equação do Calor . .	47
4.2	Molécula computacional do método explícito	48
4.3	Molécula computacional do método implícito	57
4.4	Molécula computacional do método de Crank-Nicolson	62
4.5	Molécula computacional do método Saul'yev - Esquerda → Direita	67
4.6	Molécula computacional do método Saul'yev - Direita → Esquerda	68
4.7	Ilustração gráfica do método Hopscotch	72
4.8	Molécula computacional do método Hopscotch	72
4.9	Molécula computacional do método de Lees	77
4.10	Ilustração gráfica do método das linhas	83
4.11	Molécula computacional do método box	85
4.12	Ilustração das malhas nos estágios	92
4.13	Molécula computacional do método explícito - 5 pontos	95
4.14	Molécula computacional do método explícito - 9 pontos	99
4.15	Ilustração gráfica do método localmente unidimensional	109
5.1	Notação usada no desenvolvimento das características	116

5.2	Ilustração das curvas características	118
5.3	Equação Hiperbólica - malha	124
5.4	Representação das características	125
5.5	Equações hiperbólicas - ilustração de uma malha	126
5.6	Molécula computacional do esquema explícito de primeira ordem	126
5.7	Aplicação do esquema explícito de primeira ordem	129
5.8	Molécula computacional do Esquema de Lax-Wendroff	132
5.9	Aplicação do esquema de Lax-Wendroff	133
5.10	Molécula computacional do esquema leap-frog	136
5.11	Aplicação do Esquema leap-frog	137
5.12	Equações hiperbólicas - malha para os métodos de DF implícitos	140
5.13	Molécula computacional do esquema implícito de primeira ordem	141
5.14	Molécula computacional do esquema implícito de Wendroff	145
5.15	Molécula computacional para o método de DF - Equação da Onda	149
5.16	Malha definida no domínio da Equação da Onda	149
A.1	Janela de comandos do MATLAB	161
A.2	Interface principal	162
A.3	Caminho para o método implícito	163
A.4	Janela de entrada de dados do método implícito	163
A.5	Lacunas preenchidas com os dados	164
A.6	Janela de entrada da solução analítica	164
A.7	Aproximações impressas no Wordpad	165
A.8	Método implícito - Gráfico das aproximações	165
A.9	Opções de gráficos	166
A.10	Método implícito - Gráfico da solução analítica	166
A.11	Método implícito - Gráfico dos erros	167
A.12	Método das direções alternadas - Plotagem das superfícies de nível	169
A.13	Método das direções alternadas - opções de gráficos	169
A.14	Método das direções alternadas - solução analítica - nível de tempo 2	170

Capítulo 1

Introdução

A física é um vasto campo de pesquisa para os cientistas. Muitos são os fenômenos estudados, cada qual com diferentes variáveis. Por exemplo, suponhamos que a distribuição de pressão em um dado meio seja alvo de estudo. Esse fenômeno físico é uma função com várias variáveis, pois a pressão vai depender das coordenadas espaciais x , y e z , variáveis de posição, e também da variável tempo t , podendo além destas, estarem envolvidas mais outras. Supondo, nesse exemplo, que a pressão, indicando-a por u , irá depender apenas da posição e do tempo, teremos uma função $u(x, y, z, t)$. A modelagem de um fenômeno físico, ou seja, a sua descrição matemática, não é uma tarefa fácil mas é o que muitas vezes torna possível o seu entendimento. Tais fenômenos em geral são modelados por equações diferenciais, equações que envolvem uma função e suas derivadas, e são elas que modelam eventos da natureza nas mais diversas áreas.

A modelagem de um certo evento com mais de uma variável geralmente envolve derivadas parciais, sendo portanto denominado uma *equação diferencial parcial*, a qual abreviaremos por EDP, ou um *sistema de equações diferenciais parciais*, sistema de EDPs.

Quando se modela um fenômeno da natureza por uma EDP, o que interessa é encontrar a função incógnita que satisfaça a equação, tornando o fenômeno assim possível de ser estudado. O problema é que na maior parte das vezes não é possível encontrar essa função, pois a grande maioria das EDPs não possui solução analítica. Acontece que nas várias áreas das ciências muitos problemas modelados por EDPs surgem e precisam ser resolvidos. Para contornar essa dificuldade, a solução numérica surge como alternativa para o problema.

Não existe um método geral de resolução numérica para as EDPs. Cada EDP requer o desenvolvimento de um ou mais métodos numéricos para a sua resolução, o qual precisa ser validado. Isso implica em, estudar a qualidade das aproximações, e a capacidade do

método de produzir resultados que não se distanciem da solução à medida que os cálculos forem sendo realizados. Em outras palavras, o método tem estar "amarrado" à EDP que estiver sendo resolvida numericamente.

Quando se trata de resolução numérica, o computador é um elemento indispensável. O desenvolvimento tecnológico possibilitou a criação de softwares e hardwares de grande eficiência para a implementação de métodos numéricos. O computador permite a um pesquisador testar um determinado método numérico que o mesmo tenha desenvolvido e analisá-lo. Dependendo do software que estiver usando, como o MATHEMATICA e o MATLAB por exemplo, pode lançar mão dos recursos gráficos, de grande utilidade e fácil uso, que os mesmos possuem para analisar os resultados.

Além de servir para pesquisas, os softwares, como os exemplificados acima, tem servido de auxílio ao professor no exercício de sua profissão, dando possibilidade aos alunos de visualizarem o que é ensinado em teoria. Para citar um exemplo, hoje é possível visualizar o gráfico de uma função juntamente com a sua expansão em série de Taylor, mostrando ao aluno claramente a relação que existe entre as mesmas.

Assim sendo, o presente trabalho tem como finalidade o ensino, e consistiu na implementação de uma classe de métodos de resolução numérica para EDPs apresentados em "Discretização de Equações Diferenciais Parciais: Técnica de Diferenças Finitas", Cuminato [7]. O software utilizado foi o MATLAB, e com ele foi desenvolvido um conjunto de programas que são acessados mediante uma interface padrão para entrada de dados e visualização de resultados. No que tange à visualização de resultados, o que foi desenvolvido permite ao usuário analisar os resultados, tanto através das aproximações obtidas como através dos gráficos das mesmas. Esses programas irão acompanhar o texto final de [7] para formar uma biblioteca que servirá de apoio ao professor que vier a utilizá-lo.

Das três técnicas existentes de resolução numérica de EDPs, diferenças finitas, volumes finitos e elementos finitos, o texto no qual esse trabalho se baseia aborda a primeira, isto é, diferenças finitas. Nos capítulos a seguir apresenta-se a parte teórica necessária para o desenvolvimento dos métodos de diferenças finitas, bem como a descrição dos métodos numéricos e de suas propriedades. Finalizando, segue uma apresentação do MATLAB e a conclusão. O manual de uso do software desenvolvido se encontra no apêndice.

Capítulo 2

Conceitos Fundamentais

2.1 Introdução às EDPs

Uma EDP, *equação diferencial parcial*, é uma equação envolvendo uma ou mais derivadas parciais de uma função incógnita de várias variáveis.

Exemplos:

$$u_x + u_y + x + y = 0$$

$$u_{yy} + (x - y^2)u_x + u^2 = 0$$

$$(1 + y^2)u_{xx}^2 + (1 + y^2)u_{yy} + u_x = 0$$

$$u_{xx}^2 + u^2 u_{yy} + \cos(x + y) = 0$$

$$u_{xxx} + (1 - x^2 - y^2)u_{xyy} + \exp(u)u_y = 0$$

$$yu_{xxy}^3 + xu_{xy} + \sin(u)u_{yy} = 0$$

A **ordem** de uma EDP é definida pela derivada de mais alta ordem que aparece na equação. Assim sendo, observando as EDPs do exemplo anterior, a primeira equação é uma EDP de 1ª ordem, a segunda, terceira e quarta são de 2ª ordem e a quinta e a sexta são de 3ª ordem.

Nesse trabalho nos restringiremos a EDPs de 1ª e 2ª ordens em uma função incógnita com duas ou três variáveis.

Dizemos que uma EDP é:

- **Linear** quando os coeficientes de u e de suas derivadas são funções somente das variáveis independentes.
- **Não-linear** quando a equação não é linear.
- **Quase-linear** quando a equação é linear nas derivadas de mais alta ordem e não-linear nas demais.

Exemplos:

$$yu_{xx} + (x + 2y)u_{xy} - \exp(x)u_{yy} - (x^2 + y^2)u = 0 \quad \text{linear}$$

$$(x + \sin(u))u_{xx} + (y + u)u_{xy} - (x^2)u_y = 0 \quad \text{não-linear}$$

$$xu_{xx}^2 + \exp(x + u)u_{yy} - (x^2)u_y + \sin(x^2 + y^2) = 0 \quad \text{não-linear}$$

$$(x - y)u_{xx}^2 + (x^3 + \exp(y))u_{yy} + xu_x + \cos(u) = 0 \quad \text{não-linear}$$

$$\cos(x + y)u_{xx} + \exp(x + y)u_{xy} - u^2u_y + u\sin(u) = 0 \quad \text{quase-linear}$$

$$u_{xx} - u_{xy} - \sin(u^2)u_x + (x + y)^2 = 0 \quad \text{quase-linear}$$

2.2 Classificação das EDPs

Os critérios de classificação para EDPs dependem da ordem das mesmas. A seguir, apresentaremos dois critérios de classificação para EDPs de 1ª e 2ª ordens.

2.2.1 EDPs de Primeira Ordem

A forma geral de um sistema quase-linear de n EDPs de 1ª ordem em n funções de duas variáveis independentes é dada por

$$\sum_{j=1}^n a_{ij}(u_j)_x + \sum_{j=1}^n b_{ij}(u_j)_y = c_i \quad i = 1, 2, 3, \dots, n \quad (2.1)$$

onde a_{ij} , b_{ij} e c_i são funções de $x, y, u_1, u_2, \dots, u_n$.

Podemos escrever o sistema 2.1 em termos de matrizes da seguinte forma

$$Au_x + Bu_y = c \quad (2.2)$$

onde A e B são ambas matrizes de ordem $n \times n$ e, $u = (u_1, u_2, u_3, \dots, u_n)^T$ e $c = (c_1, c_2, c_3, \dots, c_n)^T$, vetores coluna do sistema.

Suponhamos que A e B sejam não-singulares, $\det(A) \neq 0$ e $\det(B) \neq 0$, e escrevendo

$$P(\lambda)_n = \det(A - \lambda B)$$

classificamos o sistema 2.2 como:

- **Elíptico** se $P(\lambda)_n$ tem todas as raízes complexas.
- **Parabólico** se $P(\lambda)_n$ tem somente raízes reais com pelo menos uma repetida mas sem autovetores linearmente independentes.
- **Hiperbólico** se $P(\lambda)_n$ tem raízes reais e distintas, ou quando repetidas tem autovetores linearmente independentes.

A classificação acima foi feita considerando os coeficientes a_{ij} e b_{ij} constantes. Caso os coeficientes não sejam constantes, então a classificação do sistema poderá variar dependendo da posição, ou seja, o sistema poderá se comportar como elíptico, parabólico ou hiperbólico dependendo do ponto considerado no domínio.

Exemplos:

Considere o seguinte sistema:

$$\begin{cases} u_x - v_y = 0 \\ v_x + u_y = 0 \end{cases} \quad \Downarrow$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Com isso obtemos o polinômio $P(\lambda)_2 = \lambda^2 + 1$, o qual não possui raízes reais. Portanto o sistema é considerado elíptico.

Considerando agora uma EDP de 1ª ordem com uma única função incógnita em duas variáveis, temos de 2.1 a seguinte equação

$$au_x + bu_y = c \quad (2.3)$$

a qual será sempre hiperbólica, pois $P(\lambda) = \det(a - \lambda b) = a - \lambda b$ não terá mais do que um único zero real para quaisquer coeficientes a e b , desde que $b \neq 0$.

2.2.2 EDPs de Segunda Ordem

As EDPs de 2ª ordem em n variáveis tem a seguinte forma geral

$$\sum_{i,j=1}^n a_{ij}u_{x_i x_j} + \sum_{i=1}^n b_i u_{x_i} + cu = d \quad (2.4)$$

onde a_{ij} e b_i podem ou não serem constantes.

Denominamos por **parte principal** de 2.4 aos termos que envolvem as derivadas de mais alta ordem, e por ela classificamos o tipo de uma EDP de 2ª ordem.

Considerando $u_{x_i x_j} = u_{x_j x_i}$, então a parte principal de 2.4 pode ser arranjada de tal maneira que $a_{ij} = a_{ji}$. Com isso podemos assumir que a matriz A , de ordem $n \times n$, formada com os coeficientes a_{ij} é simétrica. A álgebra linear mostra que toda matriz real de ordem $n \times n$ tem n autovalores reais. Estes autovalores são os zeros do polinômio de grau n em λ , $P(\lambda) = \det(A - \lambda I)$, onde I é a matriz identidade de ordem $n \times n$. Assim sendo, classificamos as EDPs de 2ª ordem em:

- **Elíptica** se A não tem autovalores nulos e todos os seus autovalores são positivos ou negativos.
- **Parabólica** se A tem algum autovalor nulo.

- **Hiperbólica** se A não tem autovalores nulos, mas tem autovalores positivos e negativos.

A classificação acima só é válida para EDPs de 2ª ordem, e foi feita considerando os coeficientes a_{ij} constantes. Novamente, caso os coeficientes não sejam constantes, então a classificação poderá variar dependendo da posição, ou seja, a equação poderá se comportar como elíptica, parabólica ou hiperbólica dependendo do ponto considerado no domínio.

Exemplos:

Considerando a EDP $3u_{x_1x_1} + u_{x_2x_2} + 4u_{x_2x_3} + 4u_{x_3x_3} = 0$, com coeficientes constantes, temos que:

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 4 \\ 0 & 0 & 4 \end{bmatrix}$$

Observe que a matriz A não é simétrica. Como estamos considerando $u_{x_i x_j} = u_{x_j x_i}$ então podemos reorganizar a EDP considerada de tal forma que $a_{23} = a_{32}$, fazendo assim, com que a matriz A seja simétrica como segue abaixo:

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix}$$

Calculando agora o polinômio $P(\lambda) = \det(A - \lambda I)$, segue:

$$P(\lambda) = \det(A - \lambda I) = \det \begin{bmatrix} 3 - \lambda & 0 & 0 \\ 0 & 1 - \lambda & 2 \\ 0 & 2 & 4 - \lambda \end{bmatrix}$$

$\Downarrow$

$$P(\lambda) = (3 - \lambda)(\lambda)(\lambda - 5) \tag{2.5}$$

Temos que $\lambda = 0$, uma das raízes de 2.5, é um autovalor nulo, portanto a EDP considerada é parabólica em todo espaço, isto é, $\forall (x_1, x_2, x_3) \in \mathbb{R}^3$.

Considerando agora a EDP $yu_{x_1x_1} + xu_{x_1x_2} + yu_{x_2x_2} = 0$, com coeficientes variáveis, temos que:

$$A = \begin{bmatrix} y & x \\ 0 & y \end{bmatrix}$$

Tornando a matriz A simétrica pelo mesmo processo do exemplo anterior, temos:

$$A = \begin{bmatrix} y & \frac{x}{2} \\ \frac{x}{2} & y \end{bmatrix}$$

Calculando agora o polinômio $P(\lambda) = \det(A - \lambda I)$, segue:

$$\begin{aligned} P(\lambda) &= \det(A - \lambda I) = \det \begin{bmatrix} y - \lambda & \frac{x}{2} \\ \frac{x}{2} & y - \lambda \end{bmatrix} \\ &\Downarrow \\ P(\lambda) &= \lambda^2 - 2y\lambda + y^2 - \frac{x^2}{4} \end{aligned} \tag{2.6}$$

As raízes de 2.6 são dadas por $\lambda = \frac{2y \pm x}{2}$. Com isso, temos que a referida EDP será:

- Elíptica se $x < 2y$, pois a matriz A terá autovalores somente positivos.
- Parabólica se $x = 2y$, pois a matriz A terá um autovalor nulo.
- Hiperbólica se $x > 2y$, pois a matriz A terá autovalores positivos e negativos.

Além desse critério apresentado, para classificação de EDPs de 2ª ordem, existem outros que podem serem encontrados em Duchateau [8] e Lapidus [14].

2.3 Problema Bem-Posto

Uma EDP em geral tem muitas soluções, só que, quando se modela um fenômeno físico por uma EDP, estamos querendo resolver um problema bem determinado, o que implica em uma solução que forneça resultados condizentes com o problema modelado. Para que isso ocorra, é necessário fornecer ao problema o que chamamos de **condições auxiliares**, as quais se dividem em duas categorias:

- **Condições iniciais:** São funções que fornecem os valores que o problema modelado assume ao longo do domínio no início do fenômeno.
- **Condições de fronteira:** São funções que fornecem os valores que o problema modelado assume no contorno S do domínio. As condições de fronteira se dividem em 3 tipos:

1. **Condição de Dirichlet:**

$$u(x, y) = g(x, y) \quad \forall (x, y) \in S$$

2. **Condição de Neumann:**

$$\frac{\partial u(x, y)}{\partial n} = g(x, y) \quad \forall (x, y) \in S$$

onde $\frac{\partial u(x, y)}{\partial n}$ representa a derivada de $u(x, y)$ em relação a normal à fronteira S .

3. **Condição de Robins ou misto:**

$$\alpha(x, y)u(x, y) + \beta(x, y)\frac{\partial u(x, y)}{\partial n} = g(x, y) \quad \forall (x, y) \in S$$

Assim sendo, dado um problema, com as condições iniciais e/ou condições de fronteira, dizemos que o mesmo é um **problema bem-posto** se as 3 condições abaixo forem garantidas:

1. **Existência:** Existe pelo menos uma solução.
2. **Unicidade:** Existe no máximo uma solução.
3. **Estabilidade:** A solução depende continuamente dos dados. Isto significa que a pequenas variações dos dados correspondam pequenas variações na solução.

Se as condições auxiliares, na modelagem matemática de um problema, são dadas em excesso, pode haver alguma incompatibilidade e o problema não terá solução; se não forem suficientes, o problema terá muitas soluções. Somente na medida certa, teremos um problema bem-posto.

2.4 Métodos de Diferenças Finitas

Seja D um domínio onde uma dada EDP esteja definida, e $D_{hk} \subset D$ o seguinte conjunto:

$$D_{hk} = \{(x_i, y_j) = (x_1 + ih, y_1 + jk), \quad i = 1, 2, \dots, M + 1, \quad j = 1, 2, \dots, N + 1\}$$

A esse conjunto definido acima chamamos **malha**, onde:

- (x_1, y_1) é um ponto arbitrário de D ;
- h e k são, respectivamente, o tamanho do espaçamento entre dois pontos consecutivos do eixo x e y na malha.

Ilustração de uma malha:

Figura 2.1: Exemplo de uma malha

A idéia geral do método de diferenças finitas é a discretização das derivadas de $u(x, y)$ que aparecem na EDP, ou seja, na aproximação das derivadas em cada ponto da malha.

O conjunto D_{hk} foi definido para um domínio em $\mathbb{R}^2$, mas pode ser generalizado de modo análogo para $\mathbb{R}^n$.

2.5 Aproximações das derivadas por diferenças finitas

Nos métodos de diferenças finitas, as derivadas que aparecem na equação diferencial são substituídas por aproximações obtidas, mediante manipulações adequadas, da expansão em série de Taylor da função incógnita. As aproximações que serão usadas nesse trabalho são dadas a seguir:

- **Diferença Progressiva**

$$u(x+h, y) = u(x, y) + hu_x(x, y) + \frac{h^2}{2!}u_{xx}(\varepsilon, y) \quad \text{onde } \varepsilon \in (x, x+h) \quad (2.7)$$

$\Downarrow$

$$u_x(x, y) = \frac{u(x+h, y) - u(x, y)}{h} + \frac{h}{2}u_{xx}(\varepsilon, y)$$

$\Downarrow$

$$u_x(x, y) \cong \frac{u(x+h, y) - u(x, y)}{h}$$

onde o erro é dado por $\frac{h}{2!}u_{xx}(\varepsilon, y)$ com $\varepsilon \in (x, x+h)$.

- **Diferença Regressiva**

$$u(x-h, y) = u(x, y) - hu_x(x, y) + \frac{h^2}{2!}u_{xx}(\varepsilon, y) \quad \text{onde } \varepsilon \in (x-h, x) \quad (2.8)$$

$\Downarrow$

$$u_x(x, y) = \frac{u(x, y) - u(x-h, y)}{h} + \frac{h}{2}u_{xx}(\varepsilon, y)$$

$\Downarrow$

$$u_x(x, y) \cong \frac{u(x, y) - u(x-h, y)}{h}$$

onde o erro é dado por $-\frac{h}{2!}u_{xx}(\varepsilon, y)$ com $\varepsilon \in (x-h, x)$.

- **Diferença Centrada**

$$u(x+h, y) = u(x, y) + hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) + \frac{h^3}{3!}u_{xxx}(\varepsilon_1, y) \quad (2.9)$$

onde $\varepsilon_1 \in (x, x+h)$

$$u(x-h, y) = u(x, y) - hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) - \frac{h^3}{3!}u_{xxx}(\varepsilon_2, y) \quad (2.10)$$

onde $\varepsilon_2 \in (x, x+h)$

Fazendo a diferença entre 2.9 e 2.10, obtemos

$$u_x(x, y) = \frac{u(x+h, y) - u(x-h, y)}{2h}$$

onde o erro é dado por $-\frac{h^2}{6}u_{xxx}(\varepsilon, y)$ com $\varepsilon \in (x-h, x+h)$.

• **Diferença Centrada de 2ª Ordem**

$$u(x+h, y) = u(x, y) + hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) + \frac{h^3}{3!}u_{xxx}(x, y) + \frac{h^4}{4!}u_{xxxx}(\varepsilon_1, y) \quad (2.11)$$

onde $\varepsilon_1 \in (x, x+h)$

$$u(x-h, y) = u(x, y) - hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) - \frac{h^3}{3!}u_{xxx}(x, y) + \frac{h^4}{4!}u_{xxxx}(\varepsilon_2, y) \quad (2.12)$$

onde $\varepsilon_2 \in (x-h, x)$

Fazendo a soma de 2.11 com 2.12 obtemos

$$u_{xx}(x, y) \cong \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2}$$

onde o erro é dado por $-\frac{h^2}{12}u_{xxxx}(\varepsilon, y)$ com $\varepsilon \in (x-h, x+h)$.

Indicando por $u_{i,j}$ a solução exata no ponto (x_i, y_j) e $U_{i,j}$ o valor aproximado de $u_{i,j}$, temos:

$$u_x(x_i, y_j) \cong \frac{U_{i+1,j} - U_{i,j}}{h} \quad \text{Diferença Progressiva}$$

$$u_x(x_i, y_j) \cong \frac{U_{i,j} - U_{i-1,j}}{h} \quad \text{Diferença Regressiva}$$

$$u_x(x_i, y_j) \cong \frac{U_{i+1,j} - U_{i-1,j}}{2h} \quad \text{Diferença Centrada}$$

$$u_{xx}(x_i, y_j) \cong \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{2h} \quad \text{Diferença Centrada de 2ª Ordem}$$

As aproximações para $u_y(x, y)$ e $u_{yy}(x, y)$ são obtidas de modo análogo. Outras aproximações para as derivadas apresentadas podem ser vistas em Lapidus [14].

2.6 Operadores Diferenças Finitas

O uso de operadores de diferenças finitas é muito útil para se obter aproximações de ordem mais elevada, para as derivadas. Segue abaixo a apresentação de alguns operadores e, em seguida, a de algumas aproximações para derivadas, utilizando operadores, que serão utilizados em capítulos posteriores.

Derivada	$Dy(x)$	$=$	$y'(x)$
Deslocamento	$Ey(x)$	$=$	$y(x + h)$
Operador Média	$\mu_{\frac{h}{2}}y(x)$	$=$	$\frac{1}{2} \left[y(x + \frac{h}{2}) + y(x - \frac{h}{2}) \right]$
Diferença Centrada	$\delta_h y(x)$	$=$	$y(x + h) - y(x - h)$
Diferença Regressiva	$\nabla_h y(x)$	$=$	$y(x) - y(x - h)$
Diferença Progressiva	$\Delta_h y(x)$	$=$	$y(x + h) - y(x)$
Diferença Centrada de 2ª Ordem	$\delta_{\frac{h}{2}}^2 y(x)$	$=$	$y(x + h) - 2y(x) + y(x - h)$

Acompanhe o desenvolvimento abaixo.

Temos que

$$Ey(x) = y(x + h)$$

Pela fórmula de Taylor, tem-se

$$y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{3!} + \dots$$

$$\begin{aligned}
 &\Downarrow \\
 Ey(x) &= y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{3!} + \dots \\
 &\Downarrow \\
 Ey(x) &= (1 + hD + \frac{h^2}{2}D^2 + \frac{h^3}{3!}D^3 + \dots)y(x) \\
 &\Downarrow \\
 Ey(x) &= e^{hD}y(x) \\
 &\Downarrow \\
 E &= e^{hD} \\
 &\Downarrow \\
 hD &= \log(E) \tag{2.13}
 \end{aligned}$$

Observe que

$$\Delta_h y(x) = y(x+h) - y(x) = (E-1)y(x) \Rightarrow \Delta_h = E-1 \Rightarrow E = \Delta_h + 1$$

Assim sendo, substituindo $E = \Delta_h + 1$ em 2.13, temos

$$hD = \log(\Delta_h + 1) \tag{2.14}$$

Expandindo $\log(x)$ em série de Taylor em torno de 1, segue

$$\log(x) = (x-1) + \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3!} + \dots$$

Formalmente, fazendo $x = \Delta_h + 1$, fica

$$\log(\Delta_h + 1) = \Delta_h - \frac{\Delta_h^2}{2} + \frac{\Delta_h^3}{3!} - \dots$$

Substituindo esse resultado em 2.14, temos

$$\begin{aligned}
 hD &= \Delta_h - \frac{\Delta_h^2}{2} + \frac{\Delta_h^3}{3!} - \dots \\
 &\Downarrow \\
 hDy(x) &= \left(\Delta_h - \frac{\Delta_h^2}{2} + \frac{\Delta_h^3}{3!} - \dots \right) y(x) \\
 &\Downarrow \\
 hy'(x) &= \left(\Delta_h - \frac{\Delta_h^2}{2} + \frac{\Delta_h^3}{3!} - \dots \right) y(x) \tag{2.15}
 \end{aligned}$$

Com 2.15, poderemos obter, pelo truncamento da série, aproximações para a derivada primeira de uma função $y(x)$.

Outras fórmulas podem serem obtidas por processo análogo, tais como:

$$\begin{aligned}
 hy'(x+h) &= \left(\Delta_h + \frac{1}{2}\Delta_h^2 - \frac{1}{6}\Delta_h^3 + \dots \right) y(x) \\
 h^2y''(x) &= \left(\Delta_h^2 - \Delta_h^3 + \frac{11}{12}\Delta_h^4 - \frac{5}{6}\Delta_h^5 + \dots \right) y(x) \\
 h^2y''(x+h) &= \left(\Delta_h^2 - \frac{1}{12}\Delta_h^3 + \frac{1}{12}\Delta_h^6 + \dots \right) y(x) \\
 hy'(x) &= \mu_{\frac{h}{2}} \left(\delta_{\frac{h}{2}} - \frac{1}{6}\delta_{\frac{h}{2}}^3 + \frac{1}{30}\delta_{\frac{h}{2}}^5 + \dots \right) y(x) \\
 h^2y''(x) &= \left(\delta_h^2 - \frac{1}{12}\delta_h^4 + \frac{1}{90}\delta_h^6 - \frac{1}{560}\delta_h^8 \dots \right) y(x)
 \end{aligned}$$

As fórmulas aqui apresentadas são para funções de uma variável, mas podem ser generalizadas para funções com duas variáveis. Assim sendo, para indicar em relação a que variável um operador está sendo aplicado, colocaremos a variável como subscrito, e o passo, h quando em relação a x e k quando em relação a y , nesse caso ficará implícito.

Exemplos:

Derivada	$D_x u(x, y)$	$=$	$u_x(x, y)$
Deslocamento	$E_x u(x, y)$	$=$	$u(x + h, y)$
Diferença Centrada	$\delta_y u(x, y)$	$=$	$u(x, y + k) - u(x, y - k)$
Diferença Regressiva	$\nabla_x u(x, y)$	$=$	$u(x, y) - u(x + h, y)$
Diferença Progressiva	$\Delta_y u(x, y)$	$=$	$u(x, y + k) - u(x, y)$
Diferença Centrada de 2ª Ordem	$\delta_x^2 u(x, y)$	$=$	$u(x + h, y) - 2u(x, y) + u(x - h, y)$

A seguinte fórmula, de aproximação para derivada segunda em relação a x ou a y , terá atenção especial em um capítulo posterior.

$$h^2 u_{xx}(x, y) = \left(\delta_x^2 - \frac{1}{12} \delta_x^4 + \frac{1}{90} \delta_x^6 - \frac{1}{560} \delta_x^8 \dots \right) u(x, y) \quad (2.16)$$

e

$$h^2 u_{yy}(x, y) = \left(\delta_y^2 - \frac{1}{12} \delta_y^4 + \frac{1}{90} \delta_y^6 - \frac{1}{560} \delta_y^8 \dots \right) u(x, y) \quad (2.17)$$

As aproximações 2.16 e 2.17 serão usadas quando for apresentado os métodos para equações parabólicas em 2D. Hildebrand [11] apresenta maiores detalhes sobre operadores de diferenças finitas.

2.7 Estabilidade

Quando se implementa no computador um método numérico para resolver um problema físico modelado por uma EDP, o que se espera é que as aproximações, produzidas no decorrer do processo de cálculo, guardem alguma relação com o fenômeno modelado. Nem sempre isso acontece, e pode se verificar experimentalmente que alguns métodos numéricos produzem soluções cujos erros se amplificam no decorrer dos cálculos. Isso ocorre porque alguns métodos numéricos são muito sensíveis, sob algumas condições, aos erros de arredondamento do computador.

A estabilidade é uma propriedade muito importante em um método numérico, e ela está relacionada ao crescimento ou decrescimento dos erros na medida em que os cálculos são realizados. Dizemos que um método é estável se o mesmo produz aproximações com erros que não são amplificados no decorrer dos cálculos. Caso contrário, dizemos que o método é instável. Portanto, quando se desenvolve um método numérico para resolver uma EDP, é necessário que se faça um estudo sobre a sua estabilidade para validar o seu uso. Os métodos contidos nesse trabalho possuem seus critérios de estabilidade, e serão apresentados na medida em que cada um dos mesmos forem sendo desenvolvidos.

Capítulo 3

Equações Elípticas

3.1 Introdução

As equações elípticas modelam fenômenos físicos conhecidos como problemas de equilíbrio. Tais fenômenos recebem essa denominação por permanecerem em um estado estacionário, não variando a sua situação no decorrer do tempo. Para melhor entender isso, a distribuição de tensões em uma placa carregada, que é um fenômeno modelado por uma equação elíptica, é uma situação que sugere bem um estado de equilíbrio, pois uma vez que a carga esteja distribuída pela placa, a tensão em cada ponto da mesma não irá variar no decorrer do tempo. Velocidade potencial de fluxo constante de um fluido não-viscoso incompressível, potencial elétrico associado a uma distribuição bidimensional de cargas elétricas e distribuição de temperatura no caso estacionário são outros exemplos de fenômenos físicos modelados por equações elípticas. As mais conhecidas equações representantes desse tipo são:

$$u_{xx} + u_{yy} = 0 \quad \text{Equação de Laplace}$$

$$-(u_{xx} + u_{yy}) = f(x, y) \quad \text{Equação de Poisson}$$

Dessas duas equações, trabalharemos com a segunda, **Equação de Poisson**, nas condições de fronteira de **Dirichlet**

$$u(x, y) = g(x, y) \quad (x, y) \in \text{front}(D)$$

e **Neumann**

$$u_\eta(x, y) = g(x, y) \quad (x, y) \in \text{front}(D)$$

onde $front(D)$ representa a fronteira do domínio D de definição da equação em questão, e $u_n(x, y)$ a derivada da função incógnita em relação à normal a fronteira.

Iremos trabalhar, nas referidas condições de fronteira, com uma malha definida em um domínio retangular. O motivo disso será justificado em uma seção a ser apresentada posteriormente.

3.2 Métodos de Diferenças Finitas

Como já foi dito no capítulo anterior, a idéia geral do método de diferenças finitas consiste em substituir as derivadas, presentes nas EDPs, por aproximações. Estas aproximações também foram apresentadas no capítulo precedente. Assim sendo, vamos aplicá-las agora nas derivadas presentes na Equação de Poisson, e a partir daí desenvolver o processo de cálculo das aproximações, da função incógnita $u(x, y)$, nas duas condições de fronteira anteriormente mencionadas.

Primeiramente, vamos definir uma malha em um domínio retangular R . Considere os intervalos $[le, ld]$ e $[li, ls]$ os domínios, respectivamente, nos eixos x e y . M e N partições, respectivamente, em $[le, ld]$ e $[li, ls]$. Com isso, temos a seguinte malha:

$$R_{hk} = \{(x_i, y_j) = (le + (i - 1)h, li + (j - 1)k), i = 1, 2, \dots, M + 1, j = 1, 2, \dots, N + 1\}$$

onde h e k são os espaçamentos definidos pelas partições M e N , respectivamente, nos intervalos $[le, ld]$ e $[li, ls]$.

Figura 3.1: Ilustração da malha no domínio

Temos que em cada ponto (x_i, y_j) da malha vale a seguinte igualdade:

$$-(u_{xx}(x_i, y_j) + u_{yy}(x_i, y_j)) = f(x_i, y_j)$$

Usando agora diferenças centradas de 2ª ordem para aproximar u_{xx} e u_{yy} , obtemos:

$$-\left(\frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} + \frac{u(x_i, y_j + k) - 2u(x_i, y_j) + u(x_i, y_j - k)}{k^2} \right) \cong f(x_i, y_j) \quad (3.1)$$

Observe que em 3.1 não temos uma igualdade. Isso ocorreu porque substituímos u_{xx} e u_{yy} por suas aproximações. Podemos transformar 3.1 em uma igualdade se trocarmos os valores exatos $u_{i,j}$ por $U_{i,j}$, o qual esperamos que represente uma aproximação para $u_{i,j}$. Assim procedendo, temos:

$$-\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2} \right) = f_{i,j} \quad (3.2)$$

Os valores de $U_{i,j}$ realmente representarão aproximações para $u_{i,j}$ e, além disso, $U_{i,j}$ "converge" para $u_{i,j}$ quando a malha é refinada, isto é, quando h e k tendem a zero. Essas afirmações são baseadas em teoremas que serão enunciados mais adiante.

Definindo o operador $-\Delta U_{i,j}$ como sendo o primeiro membro de 3.2, temos:

$$-\Delta U_{i,j} = -\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2} \right) \Rightarrow -\Delta U_{i,j} = f_{i,j} \quad (3.3)$$

Com isso temos então a equação discretizada.

3.2.1 Convergência

Comentamos, anteriormente, sobre a "convergência" de $U_{i,j}$ para a solução exata $u_{i,j}$ à medida que a malha fosse refinada. Dizemos convergência entre aspas porque $U_{i,j}$ não é

uma sequência. Como veremos mais adiante, o que estamos querendo dizer é que, fixado um ponto $(\bar{x}, \bar{y})$, à medida que a malha é refinada, a solução aproximada U obtida se aproximará da solução exata $u(\bar{x}, \bar{y})$. Enunciaremos agora os teoremas, com os respectivos comentários, no qual essa afirmação é baseada.

Note que a expressão 3.1 não é uma igualdade, de forma que podemos definir o **Erro de Truncamento Local**, o qual abreviaremos por ETL, como sendo a seguinte quantidade:

$$T_{i,j} = -\Delta_\delta u_{i,j} - f(x_i, y_j) \quad (3.4)$$

TEOREMA 3.1 : *Se a solução $u(x, y)$ é diferenciável até ordem 4 com derivada limitada, então o ETL definido por 3.4 satisfaz*

$$|T_{i,j}| \leq \frac{P}{12}h^2 + \frac{Q}{12}k^2$$

onde $P = \max |u_{xxxx}(x, y)|$ e $Q = \max |u_{yyyy}(x, y)|$

Pelo teorema 3.1 vemos que o ETL decresce ao refinarmos a malha, isto é, ao tornarmos h e k menores. Além disso, o teorema nos fornece também a velocidade de convergência do método que é quadrática.

TEOREMA 3.2 : *Seja $U(x, y)$ qualquer função discreta definida sobre os conjuntos R_{hk} e $frontR_{hk}$, onde R_{hk} é formado pelos pontos interiores ao domínio, e $frontR_{hk}$ pelos pontos sobre a fronteira do domínio. Então,*

$$\max_{(x,y) \in R_{hk}} |U(x, y)| \leq \max_{(x,y) \in frontR_{hk}} |U(x, y)| + \frac{a^2}{2} \max_{(x,y) \in R_{hk}} |\Delta_\delta U(x, y)| \quad (3.5)$$

O teorema 3.2 nos mostra que a função discreta $U(x, y)$ definida sobre a malha é limitada, isto é, possui um limitante superior, desde que $\Delta_\delta U(x, y)$ também o seja.

Definindo o erro global por $e_{i,j} = u_{i,j} - U_{i,j}$, vamos encontrar uma estimativa para esse erro.

Tomando $U(x, y) = e(x, y)$ no teorema anterior e substituindo em 3.5, obtemos:

$$\pm e_{i,j} \leq \max_{(x,y) \in \text{front} R_{hk}} |e_{i,j}| + \frac{a^2}{2} N_0$$

onde $N_0 = \max_{(x,y) \in R_{hk}} |\Delta_\delta U(x, y)|$.

Como $U_{i,j} = u_{i,j} = g_{i,j}$ na fronteira de R_{hk} , temos que $e_{i,j} = 0$ na mesma. Assim,

$$\pm e_{i,j} \leq \frac{a^2}{2} N_0 \Rightarrow |e_{i,j}| \leq \frac{a^2}{2} N_0 = \frac{a^2}{2} \max_{(x,y) \in R_{hk}} |\Delta_\delta e_{i,j}|$$

Agora

$$\Delta_\delta e_{i,j} = \Delta_\delta u_{i,j} - \Delta_\delta U_{i,j} = \Delta_\delta u_{i,j} + f_{i,j} = -T_{i,j}$$

Assim

$$|e_{i,j}| \leq \frac{a^2}{2} \max_{R_{hk}} |T_{i,j}| \leq \frac{a^2}{12} (Ph^2 + Qk^2)$$

Portanto, o método numérico definido em 3.1 produz uma solução $U_{i,j}$ que converge pontualmente para $u(x_i, y_j)$ quando $h \rightarrow 0$ e $k \rightarrow 0$, com (x_i, y_j) fixos.

TEOREMA 3.3 : (a) Se $U(x, y)$ é uma função discreta (de malha) definida sobre $R_{hk} \cup \text{front} R_{hk}$ e satisfaz

$$\Delta_\delta U(x, y) \geq 0 \quad \forall (x, y) \in R_{hk} \quad \text{então} \quad \max_{(x,y) \in R_{hk}} U(x, y) \leq \max_{(x,y) \in \text{front} R_{hk}} U(x, y)$$

(b) Alternativamente, se $U(x, y)$ satisfaz

$$\Delta_\delta U(x, y) \leq 0 \quad \forall (x, y) \in R_{hk} \quad \text{então} \quad \min_{(x,y) \in R_{hk}} U(x, y) \geq \min_{(x,y) \in \text{front} R_{hk}} U(x, y)$$

Com o teorema 3.3 podemos mostrar que o sistema linear, definido pela Equação de Poisson e as condições de fronteira de Dirichlet, tem uma única solução.

Considere o sistema linear homogêneo a seguir:

$$\begin{cases} -\Delta_\delta U_{i,j} = 0 & (x_i, y_j) \in R_{hk} \\ U_{i,j} = 0 & (x_i, y_j) \in \text{front} R_{hk} \end{cases} \quad (3.6)$$

Pelo teorema 3.3, como $-\Delta_\delta U_{i,j} = 0$, pela parte (a) temos:

$$\max_{(x,y) \in R_{hk}} U(x,y) \leq \max_{(x,y) \in \text{front} R_{hk}} U(x,y) = 0 \quad (*)$$

E pela parte (b) que:

$$\min_{(x,y) \in R_{hk}} U(x,y) \geq \min_{(x,y) \in \text{front} R_{hk}} U(x,y) = 0 \quad (**)$$

Para que (*) e (**) ocorram, devemos ter

$$\max_{(x,y) \in R_{hk}} U(x,y) = 0 \quad \text{e} \quad \min_{(x,y) \in R_{hk}} U(x,y) = 0$$

Com isso temos que o máximo e o mínimo são nulos. Portanto a única solução é a trivial. Assim, o sistema linear construído à partir de 3.6 tem uma única solução para $f(x, y)$ e $g(x, y)$ arbitrários.

Assim, com esses resultados, vemos que o método de diferenças finitas aplicado na resolução numérica de uma equação elíptica produz uma única solução e, além disso, que quanto mais refinada for a malha, melhores serão as aproximações obtidas.

Vamos desenvolver agora o processo de resolução numérica para a equação elíptica em questão nas condições de fronteira de Dirichlet e Neumann.

3.2.2 Condições de Fronteira de Dirichlet

Considere o seguinte problema

$$\begin{cases} -(u_{xx} + u_{yy}) = f(x, y) & (x, y) \in [le, ld] \times [li, ls] \\ u(le, y) = g_e(y) & x = le \quad e \quad li \leq y \leq ls \\ u(ld, y) = g_d(y) & x = ld \quad e \quad li \leq y \leq ls \\ u(x, li) = g_i(x) & le \leq x \leq ld \quad e \quad y = li \\ u(x, ls) = g_s(x) & le \leq x \leq ld \quad e \quad y = ls \end{cases}$$

e a malha, na figura abaixo, com as condições de fronteira definidas acima.

Figura 3.2: Malha no domínio do problema de Dirichlet

Como a condição de fronteira de Dirichlet nos informa os valores que a função assume na borda do domínio, então precisaremos aplicar o operador 3.3 apenas nos pontos interiores, isto é, nos pontos que não estão na fronteira. Assim sendo, aplicando o operador 3.3 em cada um dos $(M - 1) \times (N - 1)$ pontos interiores da malha acima, obteremos um sistema de equações lineares com $(M - 1) \times (N - 1)$ equações e o mesmo número de incógnitas. Para um melhor entendimento dessa afirmação, vamos desenvolver um sistema exemplo e colocá-lo na forma matricial $AU = b$.

Temos por 3.2 que:

$$-\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2}\right) = f_{i,j}$$

Fazendo $a = -\frac{1}{h^2}$, $b = \frac{2}{h^2}$, $c = -\frac{1}{k^2}$ e $d = \frac{2}{k^2}$, segue:

$$aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j-1} + cU_{i,j+1} = f_{i,j} \quad (3.7)$$

Observando a malha, enumerando os pontos interiores da esquerda para a direita e de baixo para cima, teremos, em 3 casos, as seguintes equações:

Se $M = 1$ e $N = 1$ então não há o que fazer, pois a malha se resumirá em um

retângulo envolvendo apenas as condições de fronteira, as quais são conhecidas.

Se $M = 2$ e $N = 2$ então $(b + d)U_{i,j} = f_{i,j} - aU_{i-1,j} - aU_{i+1,j} - cU_{i,j-1} - cU_{i,j+1}$

Se $M = 2$ e $N > 2$ então

- Para $(i = 2 \text{ e } j = 2) \Rightarrow (b + d)U_{i,j} + cU_{i,j+1} = f_{i,j} - aU_{i-1,j} - aU_{i+1,j} - cU_{i,j-1}$
- Para $(i = 2 \text{ e } 3 \leq j \leq N-1) \Rightarrow cU_{i,j-1} + (b + d)U_{i,j} + cU_{i,j+1} = f_{i,j} - aU_{i-1,j} - aU_{i+1,j}$
- Para $(i = 2 \text{ e } j = N) \Rightarrow cU_{i,j-1} + (b + d)U_{i,j} = f_{i,j} - aU_{i-1,j} - aU_{i+1,j} - cU_{i,j+1}$

Se $M > 2$ e $N = 2$ então

- Para $(i = 2 \text{ e } j = 2) \Rightarrow (b + d)U_{i,j} + aU_{i+1,j} = f_{i,j} - aU_{i-1,j} - cU_{i,j-1} - cU_{i,j+1}$
- Para $(3 \leq i \leq M-1 \text{ e } j = 2) \Rightarrow aU_{i-1,j} + (b + d)U_{i,j} - aU_{i+1,j} = f_{i,j} - cU_{i,j-1} - cU_{i,j+1}$
- Para $(i = 2 \text{ e } j = N) \Rightarrow (b + d)U_{i,j} - aU_{i-1,j} = f_{i,j} - aU_{i+1,j} - cU_{i,j-1} - cU_{i,j+1}$

Se $M > 2$ e $N > 2$ então

- Para $(i = 2 \text{ e } j = 2) \Rightarrow (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j+1} = f_{i,j} - aU_{i-1,j} - cU_{i,j-1}$
- Para $(3 \leq i \leq M-1 \text{ e } j = 2) \Rightarrow aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j+1} = f_{i,j} - cU_{i,j-1}$
- Para $(i = M \text{ e } j = 2) \Rightarrow aU_{i-1,j} + (b + d)U_{i,j} + cU_{i,j+1} = f_{i,j} - aU_{i+1,j} - cU_{i,j-1}$
- Para $(i = 2 \text{ e } 3 \leq j \leq N-1) \Rightarrow cU_{i,j-1} + (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j+1} = f_{i,j} - aU_{i-1,j}$
- Para $(i = 2 \text{ e } j = N) \Rightarrow cU_{i,j-1} + (b + d)U_{i,j} + aU_{i+1,j} = f_{i,j} - aU_{i-1,j} - cU_{i,j+1}$
- Para $(3 \leq i \leq M-1 \text{ e } j = N) \Rightarrow cU_{i,j-1} + aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} = f_{i,j} - cU_{i,j+1}$
- Para $(i = M \text{ e } 3 \leq j \leq N-1) \Rightarrow cU_{i,j-1} + aU_{i-1,j} + (b + d)U_{i,j} + cU_{i,j+1} = f_{i,j} - aU_{i+1,j}$
- Para $(i = M \text{ e } j = N) \Rightarrow cU_{i,j-1} + aU_{i-1,j} + (b + d)U_{i,j} = f_{i,j} - aU_{i+1,j} - cU_{i,j+1}$
- Para $(3 \leq i \leq M-1 \text{ e } 3 \leq j \leq N-1) \Rightarrow cU_{i,j-1} + aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j+1} = f_{i,j}$

Observe que em todas as equações acima, os elementos do 1º membro que passaram para o 2º foram os elementos cujos valores são conhecidos por estarem na fronteira.

Exemplo:

Considere um domínio retangular $R = [le, ld] \times [li, ls]$ qualquer, onde $M = 5$ e $N = 4$. Com isso, segue que:

- Para $(i = 2 \text{ e } j = 2) \Rightarrow (b + d)U_{2,2} + aU_{3,2} + cU_{2,3} = f_{2,2} - aU_{1,2} - cU_{2,1}$
- Para $(3 \leq i \leq 4 \text{ e } j = 2) \Rightarrow \begin{cases} aU_{2,2} + (b + d)U_{3,2} + aU_{4,2} + cU_{3,3} = f_{3,2} - cU_{3,1} \\ aU_{3,2} + (b + d)U_{4,2} + aU_{5,2} + cU_{4,3} = f_{4,2} - cU_{4,1} \end{cases}$
- Para $(i = 5 \text{ e } j = 2) \Rightarrow aU_{4,2} + (b + d)U_{5,2} + cU_{5,3} = f_{5,2} - aU_{6,2} - cU_{5,1}$
- Para $(i = 2 \text{ e } j = 3) \Rightarrow cU_{2,2} + (b + d)U_{2,3} + aU_{3,3} + cU_{2,4} = f_{2,3} - aU_{1,3}$
- Para $(3 \leq i \leq 4 \text{ e } j = 3) \Rightarrow \begin{cases} cU_{3,2} + aU_{2,3} + (b + d)U_{3,3} + aU_{4,3} + cU_{3,4} = f_{3,3} \\ cU_{4,2} + aU_{3,3} + (b + d)U_{4,3} + aU_{5,3} + cU_{4,4} = f_{4,3} \end{cases}$
- Para $(i = 5 \text{ e } j = 3) \Rightarrow cU_{5,2} + aU_{4,3} + (b + d)U_{5,3} + cU_{5,4} = f_{5,3} - aU_{6,3}$
- Para $(i = 2 \text{ e } j = 4) \Rightarrow cU_{2,3} + (b + d)U_{2,4} + aU_{3,4} = f_{2,4} - aU_{1,4} - cU_{2,5}$
- Para $(3 \leq i \leq 4 \text{ e } j = 4) \Rightarrow \begin{cases} cU_{3,3} + aU_{2,4} + (b + d)U_{3,4} + aU_{4,4} = f_{3,4} - cU_{3,5} \\ cU_{4,3} + aU_{3,4} + (b + d)U_{4,4} + aU_{5,4} = f_{4,4} - cU_{4,5} \end{cases}$
- Para $(i = 5 \text{ e } j = 4) \Rightarrow cU_{5,3} + aU_{4,4} + (b + d)U_{5,4} = f_{5,4} - aU_{6,4} - cU_{5,5}$

Com isso, teremos o seguinte sistema:

$$\begin{cases} (b + d)U_{2,2} + aU_{3,2} + cU_{2,3} = f_{2,2} - aU_{1,2} - cU_{2,1} \\ aU_{2,2} + (b + d)U_{3,2} + aU_{4,2} + cU_{3,3} = f_{3,2} - cU_{3,1} \\ aU_{3,2} + (b + d)U_{4,2} + aU_{5,2} + cU_{4,3} = f_{4,2} - cU_{4,1} \\ aU_{4,2} + (b + d)U_{5,2} + cU_{5,3} = f_{5,2} - aU_{6,2} - cU_{5,1} \\ cU_{2,2} + (b + d)U_{2,3} + aU_{3,3} + cU_{2,4} = f_{2,3} - aU_{1,3} \\ cU_{3,2} + aU_{2,3} + (b + d)U_{3,3} + aU_{4,3} + cU_{3,4} = f_{3,3} \\ cU_{4,2} + aU_{3,3} + (b + d)U_{4,3} + aU_{5,3} + cU_{4,4} = f_{4,3} \\ cU_{5,2} + aU_{4,3} + (b + d)U_{5,3} + cU_{5,4} = f_{5,3} - aU_{6,3} \\ cU_{2,3} + (b + d)U_{2,4} + aU_{3,4} = f_{2,4} - aU_{1,4} - cU_{2,5} \\ cU_{3,3} + aU_{2,4} + (b + d)U_{3,4} + aU_{4,4} = f_{3,4} - cU_{3,5} \\ cU_{4,3} + aU_{3,4} + (b + d)U_{4,4} + aU_{5,4} = f_{4,4} - cU_{4,5} \\ cU_{5,3} + aU_{4,4} + (b + d)U_{5,4} = f_{5,4} - aU_{6,4} - cU_{5,5} \end{cases} \quad (3.8)$$

$$\underbrace{\begin{bmatrix}
 b+d & a & 0 & 0 & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 a & b+d & a & 0 & 0 & c & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & a & b+d & a & 0 & 0 & c & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & a & b+d & 0 & 0 & 0 & c & 0 & 0 & 0 & 0 \\
 c & 0 & 0 & 0 & b+d & a & 0 & 0 & c & 0 & 0 & 0 \\
 0 & c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 & 0 \\
 0 & 0 & c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 \\
 0 & 0 & 0 & c & 0 & 0 & a & b+d & 0 & 0 & 0 & c \\
 0 & 0 & 0 & 0 & c & 0 & 0 & 0 & b+d & a & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & c & 0 & 0 & a & b+d & a & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & c & 0 & 0 & a & b+d & a \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & c & 0 & 0 & a & b+d
 \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \\ U_i \end{bmatrix}}_U = \underbrace{\begin{bmatrix} f_{2,2} - aU_{1,2} - cU_{2,1} \\ f_{3,2} - cU_{3,1} \\ f_{4,2} - cU_{4,1} \\ f_{5,2} - aU_{6,2} - cU_{5,1} \\ f_{2,3} - aU_{1,3} \\ f_{3,3} \\ f_{4,3} \\ f_{5,3} - aU_{6,3} \\ f_{2,4} - aU_{1,4} - cU_{2,5} \\ f_{3,4} - cU_{3,5} \\ f_{4,4} - cU_{4,5} \\ f_{5,4} - aU_{6,4} - cU_{5,5} \end{bmatrix}}_b \Rightarrow AU = b \quad (3.9)$$

Assim sendo, dados $a, b, c, d, f(x, y)$ e as condições de fronteira, a resolução do sistema $AU = b$ acima nos fornecerá aproximações da função incógnita $u(x, y)$, da Equação de Poisson, nos pontos da malha do domínio $D = R = [l_e, l_d] \times [l_i, l_s]$ considerado. Observe que a matriz dos coeficientes é diagonal dominante, o que implica que A é inversível, logo o sistema linear $AU = b$ possui uma única solução. Na forma como desenvolvemos o sistema, considerando a enumeração dos nós, pontos da malha, da esquerda para a direita e de baixo para cima, a matriz dos coeficientes terá sempre, quando $M > 2$ e $N > 2$, a "cara" do exemplo apresentado, ou seja, uma matriz de banda onde:

- A diagonal principal será formada pelo vetor

$$\text{vetor1} = \overbrace{(b + d, b + d, b + d, b + d, \dots, b + d)}^{(M-1) \times (N-1) \text{ elementos}}$$

- A diagonal acima e abaixo da diagonal principal será formada pelo vetor

$$\text{vetor2} = \overbrace{(a, a, a, \dots, a, 0, a, a, a, \dots, a, 0, \dots, a, a, a, \dots, a)}^{(M-1) \times (N-1) - 1 \text{ elementos}}$$

$\underbrace{\hspace{1.5cm}}_{M-1 \text{ posições}} \quad \underbrace{\hspace{1.5cm}}_{M-1 \text{ posições}} \quad \underbrace{\hspace{1.5cm}}_{M-1 \text{ posições}}$

- A M -ésima diagonal acima e baixo diagonal principal será formada pelo vetor

$$\text{vetor3} = \overbrace{(c, c, c, c, c, c, c, c, c, \dots, c)}^{(M-1) \times (N-1) - M \text{ elementos}}$$

A partir do desenvolvimento do processo de cálculo no exemplo apresentado, temos o seguinte algoritmo

Algoritmo para Equação de Poisson com Condição de Fronteira de Dirichlet

1) Início

Entrada de dados. Entre com

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$[li, ls] \rightarrow$ Intervalo do domínio no eixo y , com $li < ls$.

$M \rightarrow$ Número de partições no intervalo $[le, ld]$.

$N \rightarrow$ Número de partições no intervalo $[li, ls]$.

$f(x, y) \rightarrow$ Segundo membro da Equação de Poisson.

$g_e(y) \rightarrow$ Condição de fronteira à esquerda.

$g_d(y) \rightarrow$ Condição de fronteira à direita.

$g_i(x) \rightarrow$ Condição de fronteira inferior.

$g_s(x) \rightarrow$ Condição de fronteira superior.

2) Cálculo dos espaçamentos h e $k \Rightarrow \left\{ \begin{array}{l} h = \frac{ld-le}{M} \\ k = \frac{ls-li}{N} \end{array} \right.$

3) Faça $\Rightarrow \left\{ \begin{array}{l} dim = (M-1) \times (N-1) \rightarrow \text{Dimensão da matriz dos coeficientes} \\ a = -\frac{1}{h^2}, b = \frac{2}{h^2}, c = -\frac{1}{k^2} \text{ e } d = \frac{2}{k^2} \end{array} \right.$

4) Cálculo das condições de fronteira

Para $i = 1$ a $N + 1$ faça

$$y = li + (i - 1)k$$

$$U(1, i) = g_e(y)$$

$$U(M + 1, i) = g_d(y)$$

Fim para

Para $i = 1$ a $M + 1$ faça

$$x = le + (i - 1)h$$

$$U(i, 1) = g_i(x)$$

$$U(i, M + 1) = g_s(x)$$

Fim para

5) Geração da matriz A dos coeficientes

Para $i = 1$ a dim faça

$$A(i, i) = b + d$$

Fim para

Faça $cont = M$

Se ($M > 2$ e $N = 2$) faça

Para $i = 1$ a $dim - 1$ faça

$$A(i, i + 1) = a$$

$$A(i + 1, i) = A(i, i + 1)$$

Fim para

Fim se

Se ($M = 2$ e $N > 2$) faça

Para $i = 1$ a $dim - 1$ faça

$$A(i, i + 1) = c$$

$$A(i + 1, i) = A(i, i + 1)$$

Fim para

Fim se

Se ($M > 2$ e $N > 2$) faça

Para $i = 1$ a $dim - 1$ faça

Se ($i \neq cont$) então faça

$$A(i, i + 1) = a$$

$$A(i + 1, i) = A(i, i + 1)$$

senão faça

$$A(i, i + 1) = 0$$

$$A(i + 1, i) = A(i, i + 1)$$

$$cont = cont + M$$

Fim se

Fim para

Para $i = 1$ a $dim - M$ faça

$$A(i, i + M) = c$$

$$A(i + M, i) = A(i, i + M)$$

Fim para

Fim se

6) Geração do vetor b

Faça $cont = 0$

Se ($M = 2$ e $N = 2$) faça

$$cont = cont + 1$$

$$x = le + h$$

$$y = li + k$$

$$F = f(x, y)$$

$$b(cont) = F - aU(i - 1, j) - aU(i + 1, j) - cU(i, j - 1) - cU(i, j + 1)$$

Fim se

Se ($M > 2$ e $N = 2$) faça

Para $i = 2$ a M faça

$$x = le + (i - 1)h$$

Para $j = 2$ a N faça

$$y = li + (j - 1)k$$

$$F = f(x, y)$$

$$cont = cont + 1$$

Se ($i = 2$ e $j = 2$) então faça

$$b(cont) = F - aU(i - 1, j) - cU(i, j + 1) - cU(i, j - 1)$$

senão se ($3 \leq i \leq M - 1$ e $j = 2$) faça

$$b(cont) = F - cU(i, j - 1) - cU(i, j + 1)$$

senão se ($i = M$ e $j = 2$) faça

$$b(cont) = F - aU(i + 1, j) - cU(i, j - 1) - cU(i, j + 1)$$

Fim se

Fim para

Fim para

Fim se

Se ($M = 2$ e $N > 2$) faça

Para $i = 2$ a M faça

```

 $x = le + (i - 1)h$ 
Para  $j = 2$  a  $N$  faça
     $y = li + (j - 1)k$ 
     $F = f(x, y)$ 
     $cont = cont + 1$ 
    Se  $(i = 2 \text{ e } j = 2)$  então faça
         $b(cont) = F - aU(i-1, j) - aU(i+1, j) - cU(i, j-1)$ 
        senão se  $(i = 2 \text{ e } 3 \leq j \leq N - 1)$  faça
             $b(cont) = F - aU(i - 1, j) - aU(i + 1, j)$ 
        senão se  $(i = 2 \text{ e } j = N)$  faça
             $b(cont) = F - aU(i-1, j) - aU(i+1, j) - cU(i, j+1)$ 
    Fim se
Fim para
Fim para
Fim se
Se  $(M > 2 \text{ e } N > 2)$  faça
    Para  $i = 2$  a  $M$  faça
         $x = le + (i - 1)h$ 
        Para  $j = 2$  a  $N$  faça
             $y = li + (j - 1)k$ 
             $F = f(x, y)$ 
             $cont = cont + 1$ 
            Se  $(i = 2 \text{ e } j = 2)$  então faça
                 $b(cont) = F - aU(i - 1, j) - cU(i, j - 1)$ 
                senão se  $(3 \leq i \leq M - 1 \text{ e } j = 2)$  faça
                     $b(cont) = F - cU(i, j - 1)$ 
                senão se  $(i = M \text{ e } j = 2)$  faça
                     $b(cont) = F - aU(i + 1, j) - cU(i, j - 1)$ 
                senão se  $(i = 2 \text{ e } 3 \leq j \leq N - 1)$  faça
                     $b(cont) = F - aU(i - 1, j)$ 
                senão se  $(i = 2 \text{ e } j = N)$  faça
                     $b(cont) = F - aU(i - 1, j) - cU(i, j + 1)$ 
                senão se  $(3 \leq i \leq M - 1 \text{ e } j = N - 1)$  faça
                     $b(cont) = F - cU(i, j + 1)$ 
            Fim se
        Fim para
    Fim para
Fim se

```


senão se ($i = M$ e $3 \leq j \leq N - 1$) faça
 $b(cont) = F - aU(i + 1, j)$
 senão se ($i = M$ e $j = N$) faça
 $b(cont) = F - aU(i + 1, j) - cU(i, j + 1)$
 senão ($3 \leq i \leq M - 1$ e $3 \leq j \leq N - 1$) faça
 $b(cont) = F$

Fim se

Fim para

Fim para

Fim se

7) Resolva o sistema linear $Ax = b$ por algum método numérico, e a seguir

Faça $cont = 1$

Para $j = 2$ a N faça

Para $i = 2$ a M faça

$U(i, j) = x(cont)$

$cont = cont + 1$

Fim para

Fim para

8) Fim do algoritmo

Observações importantes sobre o método:

- Este algoritmo é válido somente para uma enumeração dos nós, pontos da malha, feita da esquerda para direita e de baixo para cima. Uma outra enumeração dos nós acarretará em uma mudança na matriz dos coeficientes.
- Observe, no exemplo dado anteriormente, que a matriz A dos coeficientes é esparsa, matriz onde o número de elementos não-nulos é bem menor que os elementos nulos, e devido a isso, no algoritmo, armazenamos apenas as diagonais para não ocupar memória desnecessária e, além disso, reduzir o esforço computacional.

3.2.3 Condição de Fronteira de Neumann

Considere o seguinte problema

$$\begin{cases} -(u_{xx} + u_{yy}) = f(x, y) & (x, y) \in [le, ld] \times [li, ls] \\ u_x(le, y) = g_e(y) & x = le \text{ e } li \leq y \leq ls \\ u_x(ld, y) = g_d(y) & x = ld \text{ e } li \leq y \leq ls \\ u_y(x, li) = g_i(x) & le \leq x \leq ld \text{ e } y = li \\ u_y(x, ls) = g_s(x) & le \leq x \leq ld \text{ e } y = ls \end{cases}$$

onde as condições de fronteira são derivadas direcionais normais à fronteira. Consideraremos aqui a malha, na figura a seguir, nas condições de fronteira mencionadas acima.

Figura 3.3: Malha no domínio do problema de Neumann

Observe que não são fornecidos os valores que a função incógnita assume na fronteira, e sim as derivadas direcionais, como ilustra a figura acima. Portanto, na malha, devemos também incluir os pontos que estão na fronteira como valores a serem calculados. O problema que surge, com a condição de fronteira de Neumann, é que quando se aplica o operador 3.3 nos pontos que estão sobre a fronteira, entram na fórmula pontos que não fazem parte da malha. Considere o ponto O na figura a seguir.

Ilustração:

Figura 3.4: Discretização na fronteira - Problema de Von Neumann

Note que o ponto A não pertence a malha. Aplicando o operador 3.3 no ponto O , temos:

$$-\left(\frac{U_A - 2U_O + U_B}{h^2} + \frac{U_E - 2U_O + U_D}{k^2}\right) = f_O \quad (3.10)$$

Não temos U_A , mas essa situação pode ser contornada utilizando a condição de fronteira, que é uma derivada, aproximando-a por diferença centrada, como segue abaixo.

$$\frac{U_B - U_A}{2h} = g_O$$

Isolando U_A , temos:

$$U_A = U_B - 2hg_O \quad (3.11)$$

Substituindo U_A em 3.10, após algumas operações, obteremos:

$$-\left(\frac{2U_B - 2U_O}{h^2} + \frac{U_E - 2U_O + U_D}{k^2}\right) = f_O - \frac{2g_O}{h^2}$$

Assim podemos aplicar a discretização no ponto O , ponto da fronteira, para calcular uma aproximação para $u(x, y)$ nesse ponto. Aplicando essa técnica para os pontos que estiverem na fronteira, será possível desenvolver um sistema de equações envolvendo apenas pontos da malha. Com isso, teremos, aplicando o operador 3.3 em cada um dos pontos da malha, um sistema com $(M + 1) \times (N + 1)$ equações e o mesmo número de incógnitas.

Vamos desenvolver um sistema exemplo para ilustrar o método. Por 3.7 temos:

$$aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} + cU_{i,j-1} + cU_{i,j+1} = f_{i,j}$$

Primeiramente, para os pontos que estão na fronteira, temos que

$$(u_x)_j = \frac{U_{i+1,j} - U_{i-1,j}}{2h} \Rightarrow \begin{cases} U_{i-1,j} = U_{i+1,j} - 2h(g_e)_j & \text{fronteira à esquerda} \\ U_{i+1,j} = U_{i-1,j} + 2h(g_d)_j & \text{fronteira à direita} \end{cases}$$

onde $1 \leq j \leq N + 1$ e

$$(u_y)_i = \frac{U_{i,j+1} - U_{i,j-1}}{2h} \Rightarrow \begin{cases} U_{i,j-1} = U_{i,j+1} - 2h(g_i)_i & \text{fronteira inferior} \\ U_{i,j+1} = U_{i,j-1} + 2h(g_s)_i & \text{fronteira superior} \end{cases}$$

onde $1 \leq i \leq M + 1$.

Da mesma forma que o método anterior, **enumerando os pontos interiores da esquerda para a direita e de baixo para cima**, teremos as seguintes equações:

- Para $(i = 1 \text{ e } j = 1) \Rightarrow (b + d)U_{i,j} + 2aU_{i+1,j} + 2cU_{i,j+1} = f_{i,j} + 2ah(g_e)_j + 2ck(g_i)_i$
- Para $(2 \leq i \leq M \text{ e } j = 1) \Rightarrow aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} + 2cU_{i,j+1} = f_{i,j} + 2ck(g_i)_i$
- Para $(i = M + 1 \text{ e } j = 1) \Rightarrow 2aU_{i-1,j} + (b + d)U_{i,j} + 2cU_{i,j+1} = f_{i,j} - 2ah(g_d)_j + 2ck(g_i)_i$
- Para $(i = 1 \text{ e } 2 \leq j \leq N) \Rightarrow cU_{i,j-1} + (b + d)U_{i,j} + 2aU_{i+1,j} + cU_{i,j+1} = f_{i,j} + 2ah(g_e)_j$
- Para $(i = 1 \text{ e } j = N + 1) \Rightarrow 2cU_{i,j-1} + (b + d)U_{i,j} + 2aU_{i+1,j} = f_{i,j} + 2ah(g_e)_j - 2ck(g_s)_i$
- Para $(2 \leq i \leq M \text{ e } j = N + 1) \Rightarrow$

$$2cU_{i,j-1} + aU_{i-1,j} + (b + d)U_{i,j} + aU_{i+1,j} = f_{i,j} - 2ck(g_s)_i$$

- Para $(i = M + 1 \text{ e } 2 \leq j \leq N) \Rightarrow$

$$cU_{i,j-1} + 2aU_{i-1,j} + (b+d)U_{i,j} - cU_{i,j+1} = f_{i,j} - 2ah(g_d)_j$$

- Para $(2 \leq i \leq M \text{ e } 2 \leq j \leq N) \Rightarrow aU_{i-1,j} + (b+d)U_{i,j} + aU_{i+1,j} + cU_{i,j-1} + cU_{i,j+1} = f_{i,j}$

- Para $(i = M + 1 \text{ e } j = N + 1) \Rightarrow$

$$2cU_{i,j-1} + 2aU_{i-1,j} + (b+d)U_{i,j} = f_{i,j} - 2ah(g_d)_j - 2ck(g_s)_i$$

Considere um domínio retangular $R = [le, ld] \times [li, ls]$ qualquer, onde $M = 3$ e $N = 2$.

Com isso, segue que:

- Para $(i = 1 \text{ e } j = 1) \Rightarrow (b+d)U_{1,1} + 2aU_{2,1} + 2cU_{1,2} = f_{1,1} + 2ah(g_e)_1 + 2ck(g_i)_1$
- Para $(2 \leq i \leq 3 \text{ e } j = 1) \Rightarrow \begin{cases} aU_{1,1} + (b+d)U_{2,1} + aU_{3,1} + 2cU_{2,2} = f_{2,1} + 2ck(g_i)_2 \\ aU_{2,1} + (b+d)U_{3,1} + aU_{4,1} + 2cU_{3,2} = f_{3,1} + 2ck(g_i)_3 \end{cases}$
- Para $(i = 4 \text{ e } j = 1) \Rightarrow 2aU_{3,1} + (b+d)U_{4,1} + 2cU_{4,2} = f_{4,1} - 2ah(g_d)_1 + 2ck(g_i)_4$
- Para $(i = 1 \text{ e } j = 2) \Rightarrow cU_{1,1} + (b+d)U_{1,2} + 2aU_{2,2} + cU_{1,3} = f_{1,2} + 2ah(g_e)_2$
- Para $(2 \leq i \leq 3 \text{ e } j = 2) \Rightarrow \begin{cases} aU_{1,2} + (b+d)U_{2,2} + aU_{3,2} + cU_{2,1} + cU_{2,3} = f_{2,2} \\ aU_{2,2} + (b+d)U_{3,2} + aU_{4,2} + cU_{3,1} + cU_{3,3} = f_{3,2} \end{cases}$
- Para $(i = 4 \text{ e } j = 2) \Rightarrow cU_{4,1} + 2aU_{3,2} + (b+d)U_{4,2} - cU_{4,3} = f_{4,2} - 2ah(g_d)_2$
- Para $(i = 1 \text{ e } j = 3) \Rightarrow 2cU_{1,2} + (b+d)U_{1,3} + 2aU_{2,3} = f_{1,3} + 2ah(g_e)_3 - 2ck(g_s)_1$
- Para $(2 \leq i \leq 3 \text{ e } j = 3) \Rightarrow \begin{cases} 2cU_{2,2} + aU_{1,3} + (b+d)U_{2,3} + aU_{3,3} = f_{2,3} - 2ck(g_s)_2 \\ 2cU_{3,2} + aU_{2,3} + (b+d)U_{3,3} + aU_{4,3} = f_{3,3} - 2ck(g_s)_3 \end{cases}$
- Para $(i = 4 \text{ e } j = 3) \Rightarrow 2aU_{3,3} + (b+d)U_{4,3} + 2cU_{4,2} = f_{4,3} - 2ah(g_d)_3 - 2ck(g_s)_4$

Assim sendo, teremos o seguinte sistema:

$$\begin{cases} (b+d)U_{1,1} + 2aU_{2,1} + 2cU_{1,2} = f_{1,1} + 2ah(g_e)_1 + 2ck(g_i)_1 \\ aU_{1,1} + (b+d)U_{2,1} + aU_{3,1} + 2cU_{2,2} = f_{2,1} + 2ck(g_i)_2 \\ aU_{2,1} + (b+d)U_{3,1} + aU_{4,1} + 2cU_{3,2} = f_{3,1} + 2ck(g_i)_3 \\ 2aU_{3,1} + (b+d)U_{4,1} + 2cU_{4,2} = f_{4,1} - 2ah(g_d)_1 + 2ck(g_i)_4 \\ cU_{1,1} + (b+d)U_{1,2} + 2aU_{2,2} + cU_{1,3} = f_{1,2} + 2ah(g_e)_2 \\ aU_{1,2} + (b+d)U_{2,2} + aU_{3,2} + cU_{2,1} + cU_{2,3} = f_{2,2} \\ aU_{2,2} + (b+d)U_{3,2} + aU_{4,2} + cU_{3,1} + cU_{3,3} = f_{3,2} \\ cU_{4,1} + 2aU_{3,2} + (b+d)U_{4,2} + cU_{4,3} = f_{4,2} - 2ah(g_d)_2 \\ 2cU_{1,2} + (b+d)U_{1,3} + 2aU_{2,3} = f_{1,3} + 2ah(g_e)_3 - 2ck(g_s)_1 \\ 2cU_{2,2} + aU_{1,3} + (b+d)U_{2,3} + aU_{3,3} = f_{2,3} - 2ck(g_s)_2 \\ 2cU_{3,2} + aU_{2,3} + (b+d)U_{3,3} + aU_{4,3} = f_{3,3} - 2ck(g_s)_3 \\ 2aU_{3,3} + (b+d)U_{4,3} + 2cU_{4,2} = f_{4,3} - 2ah(g_d)_3 - 2ck(g_s)_4 \end{cases} \quad (3.12)$$

Colocando na forma matricial $AU = b$, temos

$$\underbrace{\begin{bmatrix} b+d & 2a & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a & b+d & a & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a & b+d & a & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2a & b+d & 0 & 0 & 0 & 2c & 0 & 0 & 0 & 0 \\ c & 0 & 0 & 0 & b+d & 2a & 0 & 0 & c & 0 & 0 & 0 \\ 0 & c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 & 0 \\ 0 & 0 & c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 \\ 0 & 0 & 0 & c & 0 & 0 & 2a & b+d & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 0 & 2c & 0 & 0 & 0 & b+d & 2a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & a & b+d & a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & a & b+d & a \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & 2a & b+d \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}}_U = \underbrace{\begin{bmatrix} f_{1,1} + 2ah(g_e)_1 + 2ck(g_i)_1 \\ f_{2,1} + 2ck(g_i)_2 \\ f_{3,1} + 2ck(g_i)_3 \\ f_{4,1} - 2ah(g_d)_1 + 2ck(g_i)_4 \\ f_{1,2} + 2ah(g_e)_2 \\ f_{2,2} \\ f_{3,2} \\ f_{4,2} - 2ah(g_d)_2 \\ f_{2,4} - aU_{1,4} - cU_{2,5} \\ f_{1,3} + 2ah(g_e)_3 - 2ck(g_s)_1 \\ f_{3,3} - 2ck(g_s)_3 \\ f_{4,3} - 2ah(g_d)_3 - 2ck(g_s)_4 \end{bmatrix}}_b \Rightarrow AU = b \quad (3.13)$$

Surge aqui um problema, a matriz dos coeficientes é singular, portanto não é inversível. As colunas são linearmente dependentes, pois a última coluna é igual a soma das demais. Assim sendo, não podemos resolver o sistema $AU = b$. Isso ocorre porque o problema é mal-posto. As condições de fronteira de Neumann sozinhas não são suficientes para resolver o problema, e é necessário que seja fornecido mais algum dado para que o problema seja bem-posto. Fornecendo, além das condições de fronteira de Neumann, um valor que a função assume na fronteira, o problema se torna bem-posto, como veremos no desenvolvimento a seguir.

Considere o problema abaixo:

$$\begin{cases} -(u_{xx} + u_{yy}) = f(x, y) & (x, y) \in [le, ld] \times [li, ls] \\ u(le, li) = u_0 & x = le \quad e \quad y = li \\ u_x(le, y) = g_e(y) & x = le \quad e \quad li \leq y \leq ls \\ u_x(ld, y) = g_d(y) & x = ld \quad e \quad li \leq y \leq ls \\ u_y(x, li) = g_i(x) & le \leq x \leq ld \quad e \quad y = li \\ u_y(x, ls) = g_s(x) & le \leq x \leq ld \quad e \quad y = ls \end{cases}$$

Temos aí acrescentada mais uma condição de fronteira que é $u(le, ld) = u_0$. Com isso, **mantendo a enumeração da esquerda para a direita e de baixo para cima**, chegaremos ao seguinte sistema de equações na forma matricial $AU = b$:

$$\underbrace{\begin{bmatrix} b+d & a & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 0 & 0 \\ a & b+d & a & 0 & 0 & 2c & 0 & 0 & 0 & 0 & 0 \\ 0 & 2a & b+d & 0 & 0 & 0 & 2c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b+d & 2a & 0 & 0 & c & 0 & 0 & 0 \\ c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 & 0 \\ 0 & c & 0 & 0 & a & b+d & a & 0 & 0 & c & 0 \\ 0 & 0 & c & 0 & 0 & 2a & b+d & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 2c & 0 & 0 & 0 & b+d & 2a & 0 & 0 \\ 0 & 0 & 0 & 0 & 2c & 0 & 0 & a & b+d & a & 0 \\ 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & a & b+d & a \\ 0 & 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & 2a & b+d \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{2,1} \\ U_{3,1} \\ U_{4,1} \\ U_{1,2} \\ U_{2,2} \\ U_{3,2} \\ U_{4,2} \\ U_{1,3} \\ U_{2,3} \\ U_{3,3} \\ U_{4,3} \end{bmatrix}}_U = b$$

$$= \underbrace{\begin{bmatrix} f_{2,1} + 2ck(g_i)_2 - au_0 \\ f_{3,1} + 2ck(g_i)_3 \\ f_{4,1} - 2ah(g_d)_1 + 2ck(g_i)_4 \\ f_{1,2} + 2ah(g_e)_2 - cu_0 \\ f_{2,2} \\ f_{3,2} \\ f_{4,2} - 2ah(g_d)_2 \\ f_{2,4} - aU_{1,4} - cU_{2,5} \\ f_{1,3} + 2ah(g_e)_3 - 2ck(g_s)_1 \\ f_{3,3} - 2ck(g_s)_3 \\ f_{4,3} - 2ah(g_d)_3 - 2ck(g_s)_4 \end{bmatrix}}_b \Rightarrow AU = b \quad (3.14)$$

Observe que o acréscimo da condição de fronteira, $u(le, li) = u_0$, fez com que a matriz dos coeficientes deixasse de ser singular. Essa afirmação pode ser verificada experimen-

talmente. Logo, A é inversível e a resolução numérica do sistema $AU = b$ nos fornecerá, após o fornecimento de todos os dados, uma única solução que serão as aproximações da função incógnita $u(x, y)$ na malha considerada.

Algoritmo para Equação de Poisson com Condição de fronteira de Neumann

1) Início

Entrada de dados. Entre com

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$[li, ls] \rightarrow$ Intervalo do domínio no eixo y , com $li < ls$.

$M \rightarrow$ Número de partições no intervalo $[le, ld]$.

$N \rightarrow$ Número de partições no intervalo $[li, ls]$.

$f(x, y) \rightarrow$ Segundo membro da Equação de Poisson.

$u_0 \rightarrow$ Condição de fronteira em (le, li)

$g_e(y) \rightarrow$ Condição de fronteira à esquerda.

$g_d(y) \rightarrow$ Condição de fronteira à direita.

$g_i(x) \rightarrow$ Condição de fronteira inferior.

$g_s(x) \rightarrow$ Condição de fronteira superior.

2) Cálculo dos espaçamentos h e $k \Rightarrow \left\{ \begin{array}{l} h = \frac{ld-le}{M} \\ k = \frac{ls-li}{N} \end{array} \right.$

3) Faça $\Rightarrow \left\{ \begin{array}{l} dim = (M-1) \times (N-1) - 1 \rightarrow \text{Dimensão da matriz dos coeficientes} \\ a = -\frac{1}{h^2}, b = \frac{2}{h^2}, c = -\frac{1}{k^2} \text{ e } d = \frac{2}{k^2} \end{array} \right.$

4) Cálculo das condições de fronteira

Para $j = 1$ a $N + 1$ faça

$$y = li + (j - 1)k$$

$$ge(j) = g_e(y)$$

$$gd(j) = g_d(y)$$

Fim para

Para $i = 1$ a $M + 1$ faça

$$x = le + (i - 1)h$$

$$gi(i) = g_i(x)$$

$$gs(i) = g_s(x)$$

Fim para

5) Geração da matriz A dos coeficientes

Para $i = 1$ a dim faça

$$A(i, i) = b + d$$

Fim para

Faça $cont1 = M$

Faça $cont2 = M + 1$

Faça $cont3 = M - 2$

Para $i = 1$ a dim faça

Se $(i \neq cont1 \text{ e } i \neq cont3)$ então faça

$$A(i, i + 1) = a$$

$$A(i + 1, i) = a$$

senão se $(i = cont1)$ faça

$$A(i, i + 1) = 0$$

$$A(i + 1, i) = 0$$

$$cont1 = cont1 + M$$

senão se $(i \neq cont1 \text{ e } i = cont3)$ faça

$$A(i, i + 1) = a$$

$$A(i + 1, i) = 2a$$

$$cont3 = cont3 + M$$

senão se $(i = cont2)$ faça

$$A(i, i + 1) = 2a$$

$$A(i + 1, i) = a$$

$$cont2 = cont2 + M$$

Fim se

Fim para

Para $i = 1$ a $dim - M$ faça

Se $(i \leq M - 1)$ então faça

$$A(i, i + M) = 2c$$

$$A(i + M, i) = c$$

senão faça

$$A(i, i + 1) = c$$

$$A(i + 1, i) = 2c$$

Fim se

Fim para

6) Geração do vetor b

Faça $cont = 0$

Para $i = 2$ a $M + 1$ faça

$$x = le + (i - 1)h$$

Para $j = 2$ a $N + 1$ faça

$$y = li + (j - 1)k$$

$$F = f(x, y)$$

$$cont = cont + 1$$

Se ($i = 2$ e $j = 1$) então faça

$$b(cont) = F + 2ck(gi(i)) - au_0$$

senão se ($3 \leq i \leq M$ e $j = 1$) faça

$$b(cont) = F + 2ck(gi(i))$$

senão se ($i = M + 1$ e $j = 1$) faça

$$b(cont) = F - 2ah(gd(j)) + 2ck(gi(i))$$

senão se ($i = 1$ e $j = 2$) faça

$$b(cont) = F + 2ah(ge(j)) - cu_0$$

senão se ($i = 1$ e $3 \leq j \leq N$) faça

$$b(cont) = F + 2ah(ge(j))$$

senão se ($i = 1$ e $j = N + 1$) faça

$$b(cont) = F + 2ah(ge(j)) - 2ck(gs(i))$$

senão se ($2 \leq i \leq M$ e $j = N + 1$) faça

$$b(cont) = F - 2ck(gs(i))$$

senão se ($i = M + 1$ e $2 \leq j \leq N$) faça

$$b(cont) = F - 2ah(gd(j))$$

senão se ($i = M + 1$ e $j = N + 1$) faça

$$b(cont) = F - 2ah(gd(j)) - 2ck(gs(i))$$

senão ($2 \leq i \leq M$ e $2 \leq j \leq N$) faça

$$b(cont) = F$$

Fim se

Fim para

Fim para

Fim se

7) Resolva o sistema linear $Ax = b$ por algum método numérico, e a seguir

Faça $cont = 1$

Para $j = 1$ a $N + 1$ faça

Para $i = 1$ a $M + 1$ faça

Se $(i = 1 \text{ e } j = 1)$ então faça $U(1, 1) = u_0$

senão faça $U(i, j) = x(cont)$ e $cont = cont + 1$

Fim se

Fim para

Fim para

8) Fim do algoritmo

3.3 Condições de Fronteira em Geral

Na prática, a maioria dos problemas envolvendo equações parabólicas e hiperbólicas são definidas em domínios retangulares. Isso já não acontece com as equações elípticas. Nos métodos desenvolvidos para a Equação de Poisson, trabalhamos com uma malha definida em um domínio retangular pela facilidade de definição da malha, bem como de realização dos cálculos. Acontece que, na prática, a grande maioria das equações elípticas não são definidas em domínios retangulares. Devido a isso fica difícil adaptar uma malha perfeitamente ao domínio. Como consequência, dependendo do domínio ou da malha, alguns ou nenhum elemento da malha pode estar na fronteira. Veja a figura abaixo.

Figura 3.5: Malha em um domínio qualquer 1

Observe na figura que temos pontos da malha, pontos A e D , que não se situam sobre a fronteira. Com isso, na fórmula

$$-\left(\frac{U_A - 2U_B + U_C}{h^2} + \frac{U_E - 2U_B + U_D}{k^2}\right) = f_B \quad (3.15)$$

não temos como calcular U_A e U_D , pelo fato dos pontos A e D não se situarem sobre a fronteira.

Uma das técnicas conhecidas para contornar essa situação se baseia na utilização de um polinômio de primeiro grau. Assim, de acordo com a figura acima, utilizamos a extrapolação polinomial da seguinte forma:

Figura 3.6: Ampliação da região considerada 1

O polinômio linear que extrapola U_P e U_B é

$$[(x - x_B)U_P - (x - x_P)U_B] \frac{1}{x_P - x_B}$$

Uma aproximação para U_B pode ser facilmente deduzida substituindo x e x_A na expressão acima para obter

$$U_A \cong [(x_A - x_B)U_P - (x_A - x_P)U_B] \frac{1}{x_P - x_B} = -\frac{1}{h\theta_1}[-hU_P + (1 - \theta_1)hU_B] = -\frac{1}{\theta_1}[U_P - (1 - \theta_1)U_B]$$

⇓

$$U_A \cong -\frac{1}{\theta_1}[U_P - (1 - \theta_1)U_B] \quad (3.16)$$

De modo análogo

$$U_D \cong -\frac{1}{\theta_2}[U_Q - (1 - \theta_2)U_B] \quad (3.17)$$

Substituindo 3.16 e 3.17 em 3.15, após algumas operações, obtemos:

$$-\left(\frac{1}{h^2}\left[U_C - \frac{(1 + \theta_1)U_B}{\theta_1}\right] + \frac{1}{k^2}\left[U_E - \frac{(1 + \theta_2)U_B}{\theta_2}\right]\right) = f_B + \frac{U_P}{h^2\theta_1} + \frac{U_Q}{k^2\theta_2}$$

Assim, com a equação acima, podemos aplicar a discretização no ponto B.

Cunha [6] apresenta uma outra técnica, a qual consiste na construção de uma malha irregular. Veja figura abaixo:

Figura 3.7: Malha em um domínio qualquer 2

Nesse caso, a discretização da Equação de Poisson, para os pontos próximos da fronteira é feita de maneira diferente, pelo fato dos espaçamentos para os outros pontos não serem todos iguais. Observe isso na ampliação na figura a seguir referente ao ponto B:

Ilustração:

Figura 3.8: Ampliação da região considerada 2

Segundo Cunha, expandindo $u(x_B + h_1, y)$ e $u(x_B - h_2, y)$ em série de Taylor em torno de (x_B, y_B) obteremos, após algumas operações convenientes, uma aproximação para u_{xx} . De modo análogo, a mesma é feita para obter uma aproximação para u_{yy} . Substituindo-as na Equação de Poisson, chegaremos na seguinte equação discretizada.

$$-2 \left\{ \frac{U_A}{h_2(h_1 + h_2)} + \frac{U_C}{h_1(h_1 + h_2)} + \frac{U_D}{h_4(h_3 + h_4)} + \frac{U_E}{h_3(h_3 + h_4)} - \left(\frac{1}{h_1 h_2} + \frac{1}{h_3 h_4} \right) U_B \right\} = f_B \quad (3.18)$$

Podemos aplicar essa discretização em todos os pontos de uma malha irregular. No caso dos pontos que não estiverem próximos da fronteira, se $h_1 = h_2 = h_3 = h_4$, 3.18 se reduzirá a 3.2.

O problema dos dois métodos apresentados, para malhas irregulares, está na complexidade de sua aplicação. No primeiro caso, teremos que determinar θ_1 e θ_2 para cada ponto próximo da fronteira, e no segundo, os espaçamentos h_1 , h_2 , h_3 e h_4 .

Capítulo 4

Equações Parabólicas

4.1 Introdução

As equações parabólicas estão relacionadas aos chamados problemas de evolução ou propagação. A equação mais conhecida representante desse tipo é a **Equação do Calor** ou da difusão

$$u_t = a(x, t)u_{xx} + f(x, t) \quad a(x, t) > 0 \quad (x, t) \in \mathbb{R}_+^2 \quad (4.1)$$

onde x é a variável espacial, ou de posição, e t a variável tempo. A equação 4.1 modela um problema unidimensional com relação à variável espacial. Para problemas bidimensionais e tridimensionais nas variáveis espaciais, teremos, respectivamente

$$u_t = a(x, y, t)(u_{xx} + u_{yy}) + f(x, y, t) \quad a(x, y, t) > 0 \quad (x, y, t) \in \mathbb{R}_+^3 \quad (4.2)$$

$$u_t = a(x, y, z, t)(u_{xx} + u_{yy} + u_{zz}) + f(x, y, z, t) \quad a(x, y, z, t) > 0 \quad (x, y, z, t) \in \mathbb{R}_+^4 \quad (4.3)$$

Desenvolveremos, nesse capítulo, métodos numéricos para as equações 4.1 e 4.2, para a constante e $f \equiv 0$. Além disso, serão também desenvolvidos métodos para outros tipos de equações parabólicas.

4.2 Métodos de Diferenças Finitas

Desenvolveremos a seguir métodos numéricos para alguns tipos de equações parabólicas.

4.3 Métodos para a Equação do Calor em 1D

Consideremos o seguinte problema:

$$\begin{cases} u_t = au_{xx} & a > 0, \text{ constante}, (x, t) \in [0, L] \times [0, T] \\ u(x, 0) = f(x) & x \in [0, L] \text{ condição inicial} \\ u(0, t) = g(t) & t \in [0, T] \text{ condição de fronteira em } x = 0 \\ u(L, t) = h(t) & t \in [0, T] \text{ condição de fronteira em } x = L \end{cases} \quad (4.4)$$

Apresentaremos 5 métodos para a Equação do Calor em uma dimensão, considerando uma malha definida em $[0, L] \times [0, T]$ com M partições em $[0, L]$ e N partições em $[0, T]$, conforme ilustra a figura abaixo.

Figura 4.1: Malha definida no domínio do problema envolvendo a Equação do Calor

4.3.1 Método Explícito

O método explícito é obtido aproximando u_t por diferenças progressivas e u_{xx} por diferenças centradas de 2ª ordem. Acompanhe o desenvolvimento abaixo:

$$u_t = au_{xx} \quad (4.5)$$

Discretizando as derivadas em 4.5, temos:

$$\frac{U_{i,j+1} - U_{i,j}}{k} = a \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2}$$

$$\Downarrow$$

$$U_{i,j+1} - U_{i,j} = \sigma(U_{i-1,j} - 2U_{i,j} + U_{i+1,j})$$

$$\Downarrow$$

$$U_{i,j+1} = \sigma U_{i-1,j} + (1 - 2\sigma)U_{i,j} + \sigma U_{i+1,j} \quad (4.6)$$

onde $\sigma = a \frac{k}{h^2}$.

Assim sendo, em 4.6 temos a equação à diferenças do método explícito, e por ela vemos que uma aproximação, em um determinado nível, é obtido a partir de três aproximações do nível anterior.

Esse método é representado pela seguinte molécula computacional:

Figura 4.2: Molécula computacional do método explícito

A idéia de aplicação do método explícito é bem simples. Definida uma malha em um domínio $[0, L] \times [0, T]$, com M e N partições, respectivamente, em $[0, L]$ e $[0, T]$, fixamos $j = 2$ e fazemos i variar de 2 a M , e depois a mesma coisa para $j = 3, 4, 5, \dots, N$.

Erro de Truncamento Local

Quando realizamos a discretização de uma EDP, isto é, quando substituímos as derivadas presentes na equação por aproximações, ocorrerá inevitavelmente um erro na mesma. Acompanhe o desenvolvimento abaixo com a equação que estamos trabalhando.

$$u_t = au_{xx}$$

Discretizando a equação, temos:

$$\frac{u_{i,j+1} - u_{i,j}}{k} \cong a \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} \quad (4.7)$$

Observe que podemos corrigir 4.7 da seguinte forma

$$\frac{u_{i,j+1} - u_{i,j}}{k} = a \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \tau_{i,j} \quad (4.8)$$

A esse valor $\tau_{i,j}$ chamamos **Erro de Truncamento Local**, abreviadamente ETL, e ele representa uma maneira de medir a grandeza do erro associado da discretização da equação no ponto (x_i, t_j) . O ETL é obtido expandindo $u_{i,j+1}$, $u_{i+1,j}$ e $u_{i-1,j}$ em série de Taylor em torno do ponto (x_i, t_j) , como é mostrado abaixo:

$$\begin{aligned} u_{i,j+1} &= u(x + ih, t + (j + 1)k) \\ &= u(x + ih, t + jk) + ku_t(x + ih, t + jk) + \frac{k^2}{2}u_{tt}(x + ih, t + jk) + O(k^3) \end{aligned}$$

$$\begin{aligned} u_{i+1,j} &= u(x + (i + 1)h, t + jk) \\ &= u(x + ih, t + jk) + hu_x(x + ih, t + jk) + \frac{h^2}{2}u_{xx}(x + ih, t + jk) \\ &= +\frac{h^3}{3!}u_{xxx}(x + ih, t + jk) + \frac{h^4}{4!}u_{xxxx}(x + ih, t + jk) + O(h^5) \end{aligned}$$

$$\begin{aligned} u_{i-1,j} &= u(x + (i - 1)h, t + jk) \\ &= u(x + ih, t + jk) - hu_x(x + ih, t + jk) + \frac{h^2}{2}u_{xx}(x + ih, t + jk) \\ &= -\frac{h^3}{3!}u_{xxx}(x + ih, t + jk) + \frac{h^4}{4!}u_{xxxx}(x + ih, t + jk) + O(h^5) \end{aligned}$$

Substituindo as expansões acima em 4.8, temos:

$$\tau_{i,j} = \frac{k}{2}u_{tt} - a\frac{h^2}{12}u_{xxxx} + O(k^2) + O(h^3) = O(k + h^2) \quad (4.9)$$

Observe, em 4.9, que quando h e k tendem a 0, o ETL $\tau_{i,j}$ também tende a 0 e a equação discretizada tende a Equação do Calor, ou seja, a equação discretizada converge para a Equação do Calor. Quando isso acontece, dizemos que o método é consistente.

A consistência de um método numérico é muito importante, pois ela nos diz se realmente estamos resolvendo a equação de interesse.

Note também, em 4.9, que para h e k próximos de 0, o maior erro que ocorrer na discretização vai ser determinado pelo termo de menor potência entre h e k . Ao expoente do termo de menor potência chamamos **ordem**. No caso da Equação do Calor, para a discretização usada, o termo de menor potência é $\frac{k}{2}u_{tt}$. Assim sendo, dizemos que a ordem do método é 1. Observe que, quanto maior for a ordem de um método numérico, melhores aproximações serão fornecidas pelo mesmo.

Estabilidade

Um método simples e usual, para analisar a estabilidade, é o critério de Von Neumann. Esse critério consiste em examinar o efeito da amplificação ou amortecimento de um modo de Fourier, fixado em uma linha ou estágio j . Esse critério, entretanto, só pode ser usado quando a equação é linear com coeficiente a constante. Acompanhe o desenvolvimento abaixo:

Vamos supor que exista uma solução da equação $u_t = au_{xx}$ do tipo

$$u(x, t) = e^{t\lambda} e^{x\beta I}$$

onde $I = \sqrt{-1}$.

Definida uma malha, temos

$$u(x_i, t_j) = e^{t_j\lambda} e^{x_i\beta I} \quad (4.10)$$

onde $x_i = ih$ e $t_j = jk$. Substituindo esses valores em 4.10 segue

$$U_{i,j} = e^{jk\lambda} e^{ih\beta I} \quad (4.11)$$

Vamos supor que 4.11 seja uma solução da equação à diferenças 4.6, e a partir daí tentaremos encontrar λ e β que satisfaçam 4.6.

Assim sendo, da equação à diferenças do método explícito, temos:

$$U_{i,j+1} = \sigma U_{i-1,j} + (1 - 2\sigma)U_{i,j} + \sigma U_{i+1,j} \quad (4.12)$$

Substituindo 4.11 em 4.12, segue

$$\begin{aligned}
 e^{(j+1)k\lambda} e^{ih\beta I} &= \sigma e^{jk\lambda} e^{(i-1)h\beta I} + (1 - 2\sigma) e^{jk\lambda} e^{ih\beta I} + \sigma e^{jk\lambda} e^{(i+1)h\beta I} \\
 &\Downarrow \\
 e^{k\lambda} e^{jk\lambda} e^{ih\beta I} &= (1 - 2\sigma) e^{jk\lambda} e^{ih\beta I} + \sigma e^{-h\beta I} e^{jk\lambda} e^{ih\beta I} + \sigma e^{h\beta I} e^{jk\lambda} e^{ih\beta I} \\
 &\Downarrow \\
 e^{k\lambda} U_{i,j} &= (1 - 2\sigma) U_{i,j} + \sigma e^{-h\beta I} U_{i,j} + \sigma e^{h\beta I} U_{i,j} \\
 &\Downarrow \\
 e^{k\lambda} U_{i,j} &= [1 - 2\sigma + \sigma(e^{h\beta I} + e^{-h\beta I})] U_{i,j} \\
 &\Downarrow \\
 e^{k\lambda} &= 1 - 2\sigma + \sigma(e^{h\beta I} + e^{-h\beta I}) \\
 &\Downarrow \\
 e^{k\lambda} &= 1 - 2\sigma + 2\sigma \cos(h\beta) \\
 &\Downarrow \\
 e^{k\lambda} &= 1 + 2\sigma [\cos(h\beta) - 1] \\
 &\Downarrow \\
 e^{k\lambda} &= 1 - 4\sigma \sin^2\left(\frac{h\beta}{2}\right) \quad (4.13)
 \end{aligned}$$

Sendo σ não negativo, por 4.13 temos que $e^{k\lambda}$ é menor que 1 sempre. Para que haja o amortecimento de 4.6, deveremos ter

$$|e^{k\lambda}| < 1$$

$\Downarrow$

$$-1 < e^{k\lambda} < 1 \quad (4.14)$$

Como a segunda desigualdade, $e^{k\lambda} < 1$, é satisfeita, resta analisar a primeira. Temos que:

$$\begin{aligned} e^{k\lambda} &> -1 \\ \Downarrow \\ 1 - 4\sigma \operatorname{sen}^2\left(\frac{h\beta}{2}\right) &> -1 \\ \Downarrow \\ -4\sigma \operatorname{sen}^2\left(\frac{h\beta}{2}\right) &> -2 \\ \Downarrow \\ \sigma \operatorname{sen}^2\left(\frac{h\beta}{2}\right) &< \frac{1}{2} \end{aligned} \quad (4.15)$$

Note que o maior valor que $\operatorname{sen}^2\left(\frac{\theta}{2}\right)$ pode assumir é 1. Assim sendo, para que a desigualdade 4.15 seja satisfeita, devemos ter

$$\sigma < \frac{1}{2} \quad (4.16)$$

Portanto, para que haja estabilidade no método explícito, deveremos ter a condição 4.16 satisfeita. Devido a essa restrição, dizemos que o método explícito é **condicionalmente estável**.

Uma outra maneira de analisar a estabilidade pode ser feita usando notação matricial. Acompanhe o desenvolvimento a seguir:

Definimos o erro em um ponto (x_i, t_j) , $e_{i,j}$, por

$$e_{i,j} = U_{i,j} - u_{i,j}$$

isto é, a diferença entre a solução aproximada e a solução exata no ponto (x_i, t_j) da malha. Assim sendo, de 4.6 temos

$$\begin{aligned}
e_{i,j+1} &= U_{i,j+1} - u_{i,j+1} \\
&\Downarrow \\
e_{i,j+1} &= \sigma U_{i-1,j} + (1-2\sigma)U_{i,j} + \sigma U_{i+1,j} - [\sigma u_{i-1,j} + (1-2\sigma)u_{i,j} + \sigma u_{i+1,j} + k\tau_{i,j}] \\
&\Downarrow \\
e_{i,j+1} &= \sigma(U_{i-1,j} - u_{i-1,j}) + (1-2\sigma)(U_{i,j} - u_{i,j}) + \sigma(U_{i+1,j} - u_{i+1,j}) - k\tau_{i,j} \\
&\Downarrow \\
e_{i,j+1} &= \sigma e_{i-1,j} + (1-2\sigma)e_{i,j} + \sigma e_{i+1,j} - k\tau_{i,j} \tag{4.17}
\end{aligned}$$

Fixando agora um estágio $j+1$, de uma dada malha, com M partições no domínio considerado no eixo x , e fazendo i variar de 2 a M , teremos aplicando 4.18, no estágio fixado, o seguinte sistema matricial:

$$\underbrace{\begin{bmatrix} e_{2,j+1} \\ e_{3,j+1} \\ \vdots \\ e_{M-1,j+1} \\ e_{M,j+1} \end{bmatrix}}_{e_{j+1}} = \underbrace{\begin{bmatrix} 1-2\sigma & \sigma & 0 & \dots & 0 \\ \sigma & 1-2\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & 1-2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & 1-2\sigma & \sigma \\ 0 & \dots & 0 & \sigma & 1-2\sigma \end{bmatrix}}_A \underbrace{\begin{bmatrix} e_{2,j} \\ e_{3,j} \\ \vdots \\ e_{M-1,j} \\ e_{M,j} \end{bmatrix}}_{e_j} - k \underbrace{\begin{bmatrix} \tau_{2,j} - e_{1,j} \\ \tau_{3,j} \\ \vdots \\ \tau_{M-1,j} \\ \tau_{M,j} - e_{M+1,j} \end{bmatrix}}_{\tau_j}$$

Observe que $e_{1,j} = e_{M+1,j} = 0$, para qualquer estágio, pois são pontos que estão na fronteira onde a solução é conhecida. Assim sendo, temos

$$\underbrace{\begin{bmatrix} e_{2,j+1} \\ e_{3,j+1} \\ \vdots \\ e_{M-1,j+1} \\ e_{M,j+1} \end{bmatrix}}_{e_{j+1}} = \underbrace{\begin{bmatrix} 1-2\sigma & \sigma & 0 & \dots & 0 \\ \sigma & 1-2\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1-2\sigma & \sigma \\ 0 & \dots & 0 & \sigma & 1-2\sigma \end{bmatrix}}_A \underbrace{\begin{bmatrix} e_{2,j} \\ e_{3,j} \\ \vdots \\ e_{M-1,j} \\ e_{M,j} \end{bmatrix}}_{e_j} - k \underbrace{\begin{bmatrix} \tau_{2,j} \\ \tau_{3,j} \\ \vdots \\ \tau_{M-1,j} \\ \tau_{M,j} \end{bmatrix}}_{\tau_j}$$

$\Downarrow$

$$e_{j+1} = Ae_j - k\tau_j \quad (4.18)$$

Aplicando 4.18 para cada nível de tempo, temos para

$$j = 1 \rightarrow e_2 = Ae_1 - k\tau_1 = -k\tau_1, \text{ pois } e_1 = 0$$

$$j = 2 \rightarrow e_3 = Ae_2 - k\tau_2 = -kA\tau_1 - k\tau_2 = -k(A\tau_1 + \tau_2)$$

$$j = 3 \rightarrow e_4 = Ae_3 - k\tau_3 = -k(A^2\tau_1 + A\tau_2) - k\tau_3 = -k(A^2\tau_1 + A\tau_2 + \tau_3)$$

$$j = 4 \rightarrow e_5 = Ae_4 - k\tau_4 = -k(A^3\tau_1 + A^2\tau_2 + A\tau_3) - k\tau_4 = -k(A^3\tau_1 + A^2\tau_2 + A\tau_3 + \tau_4)$$

$$j = 5 \rightarrow e_6 = Ae_5 - k\tau_5 = -k(A^4\tau_1 + A^3\tau_2 + A^2\tau_3 + A\tau_4 + \tau_5)$$

Generalizando, teremos:

$$e_{j+1} = -k \sum_{p=0}^{j-1} A^p \tau_{j-p} \quad (4.19)$$

Em 4.19 vemos que o erro depende essencialmente da matriz A e de suas potências. A matriz A é chamada **matriz de amplificação**. O que interessa é que não haja amplificação dos erros à medida que os cálculos forem sendo realizados. Em Blum [3] mostra-se que $\lim_{p \rightarrow \infty} A^p = 0$ se, e somente se, todos os autovalores de A são, em módulo, menores que 1. Pode-se também verificar que os autovalores de A são dados por

$$\mu_i = 1 - 4\sigma \operatorname{sen}^2 \frac{i\pi}{2M}, \quad i = 1, 2, 3, \dots, M-1$$

Assim sendo, para haver estabilidade, deveremos ter

$$\begin{aligned} |\mu_i| &< 1 \\ \Downarrow \\ -1 &< 1 - 4\sigma \operatorname{sen}^2 \frac{i\pi}{2M} < 1 \\ \Downarrow \end{aligned}$$

$$0 < \sigma \operatorname{sen}^2 \frac{i\pi}{2M} < \frac{1}{2} \quad (4.20)$$

Essa desigualdade será sempre verdadeira se

$$\sigma < \frac{1}{2}$$

e isso ocorrendo, haverá estabilidade para o método explícito. Com isso, chegamos ao mesmo resultado obtido pelo critério de Von Neumann.

Segue um algoritmo para o método explícito.

Algoritmo para o Método Explícito

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Cálculo das aproximações

Para $j = 1$ a N faça

Para $i = 2$ a M faça

$$U(i, j+1) = \sigma U(i-1, j) + (1-2\sigma)U(i, j) + \sigma U(i+1, j)$$

Fim para

Fim Para

Solução $\rightarrow U(1 \text{ a } M+1, 1 \text{ a } N+1)$

5) Fim do algoritmo

4.3.2 Método Implícito

O método implícito é obtido usando diferenças regressivas para u_t e diferenças centradas de 2ª ordem para u_{xx} . Siga o desenvolvimento abaixo.

$$u_t = au_{xx} \quad (4.21)$$

Discretizando as derivadas em 4.21, temos:

$$\begin{aligned} \frac{U_{i,j} - U_{i,j-1}}{k} &= a \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} \\ &\Downarrow \\ U_{i,j} - U_{i,j-1} &= \sigma(U_{i-1,j} - 2U_{i,j} + U_{i+1,j}) \text{ onde } \sigma = a \frac{k}{h^2} \\ &\Downarrow \\ U_{i,j-1} &= U_{i,j} - \sigma(U_{i-1,j} - 2U_{i,j} + U_{i+1,j}) \\ &\Downarrow \\ U_{i,j-1} &= -\sigma U_{i-1,j} + (1+2\sigma)U_{i,j} - \sigma U_{i+1,j} \end{aligned} \quad (4.22)$$

Ou melhor escrevendo

$$-\sigma U_{i-1,j} + (1 + 2\sigma)U_{i,j} - \sigma U_{i+1,j} = U_{i,j-1} \quad (4.23)$$

Assim sendo, temos em 4.23 a equação às diferenças do método implícito, e tal método é representado pela seguinte molécula computacional:

Figura 4.3: Molécula computacional do método implícito

A aplicação desse método requer a resolução de um sistema linear em cada estágio. Isso será melhor ilustrado com a explanação a seguir.

Fixando um estágio j e variando i de 2 a M , temos:

$$\begin{cases}
 -\sigma U_{1,j} + (1 + 2\sigma)U_{2,j} - \sigma U_{3,j} & = & U_{2,j-1} \\
 -\sigma U_{2,j} + (1 + 2\sigma)U_{3,j} - \sigma U_{4,j} & = & U_{3,j-1} \\
 \vdots & & \vdots \\
 -\sigma U_{M-2,j} + (1 + 2\sigma)U_{M-1,j} - \sigma U_{M,j} & = & U_{M-1,j-1} \\
 -\sigma U_{M-1,j} + (1 + 2\sigma)U_{M,j} - \sigma U_{M+1,j} & = & U_{M,j-1}
 \end{cases}$$

$$\Downarrow$$

$$\underbrace{\begin{bmatrix} 1 + 2\sigma & -\sigma & 0 & \dots & 0 \\ -\sigma & 1 + 2\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 + 2\sigma & -\sigma \\ 0 & \dots & 0 & -\sigma & 1 + 2\sigma \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M-1,j} \\ U_{M,j} \end{bmatrix}}_{U_{j-1}} = \underbrace{\begin{bmatrix} U_{2,j-1} + \sigma U_{1,j} \\ U_{3,j-1} \\ \vdots \\ U_{M-1,j-1} \\ U_{M,j-1} + \sigma U_{M+1,j} \end{bmatrix}}_{U_{j-1}}$$

$$\Downarrow$$

$$AU_j = U_{j-1} \quad (4.24)$$

Portanto, teremos que resolver o sistema linear 4.24 para cada estágio j . Assim sendo, para aplicar o método, basta definir uma malha, gerar a matriz A em 4.24, e aí então ir resolvendo o sistema linear 4.24 a partir de $j = 2$ até o último estágio. Observe que a matriz A é diagonal dominante. Logo, isso implica que A é inversível e o sistema tem solução única.

Erro de Truncamento Local

O ETL do método implícito é calculado da mesma forma que o método explícito. Assim sendo, seja $\tau_{i,j}$ o ETL ocorrido no cálculo de $u_{i,j}$. Então, podemos escrever

$$\frac{u_{i,j} - u_{i,j-1}}{k} = a \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \tau_{i,j} \quad (4.25)$$

Expandindo $u_{i-1,j}$, $u_{i+1,j}$ e $u_{i,j-1}$ em série de Taylor em torno do ponto (x_i, t_j) , e a seguir, substituindo-os em 4.25, chegaremos a

$$\tau_{i,j} = -\frac{k}{2}u_{tt} - a\frac{h^2}{12}u_{xxxx} + O(k^2) + O(h^3) = O(k + h^2) \quad (4.26)$$

onde concluímos que o método implícito é consistente e de ordem 1.

Estabilidade

Analogamente ao que foi feito para o método explícito, pelo critério de Von Neumann, suponhamos que

$$U_{i,j} = e^{jk\lambda} e^{ih\beta I} \quad (4.27)$$

seja uma solução da equação à diferenças 4.23, e tentamos encontrar λ e β que satisfaçam essa equação.

Substituindo 4.27 em 4.22, temos:

$$e^{(j-1)k\lambda} e^{ih\beta I} = -\sigma e^{jk\lambda} e^{(i-1)h\beta I} + (1 + 2\sigma) e^{jk\lambda} e^{ih\beta I} - \sigma e^{jk\lambda} e^{(i+1)h\beta I}$$

$$\begin{aligned}
& \Downarrow \\
e^{-k\lambda} e^{jk\lambda} e^{ih\beta I} &= (1 + 2\sigma) e^{jk\lambda} e^{ih\beta I} - \sigma e^{-h\beta I} e^{jk\lambda} e^{ih\beta I} - \sigma e^{h\beta I} e^{jk\lambda} e^{ih\beta I} \\
& \Downarrow \\
e^{-k\lambda} U_{i,j} &= (1 + 2\sigma) U_{i,j} - \sigma e^{-h\beta I} U_{i,j} - \sigma e^{h\beta I} U_{i,j} \\
& \Downarrow \\
e^{-k\lambda} U_{i,j} &= [1 + 2\sigma - \sigma(e^{h\beta I} + e^{-h\beta I})] U_{i,j} \\
& \Downarrow \\
e^{-k\lambda} &= 1 + 2\sigma - \sigma(e^{h\beta I} + e^{-h\beta I}) \\
& \Downarrow \\
e^{-k\lambda} &= 1 + 2\sigma - 2\sigma \cos(h\beta) \\
& \Downarrow \\
e^{-k\lambda} &= 1 + 2\sigma[1 - \cos(h\beta)] \\
& \Downarrow \\
e^{k\lambda} &= \frac{1}{1 + 4\sigma \sin^2\left(\frac{\beta}{2}\right)} \tag{4.28}
\end{aligned}$$

Observe que $|e^{k\lambda}| < 1$ para qualquer σ , que é justamente a condição para que haja estabilidade. Portanto o método implícito é **incondicionalmente estável**.

Fazendo uma análise matricial do erro, analogamente ao que foi feito para o método explícito, tomando

$$e_{i,j-1} = U_{i,j-1} - u_{i,j-1}$$

chegaremos a

$$e_j = k \sum_{p=0}^{j-2} (A^{-1})^{p+1} \tau_{j-p} \tag{4.29}$$

onde A^{-1} é a inversa da matriz A em 4.24.

Novamente, para que haja estabilidade, os autovalores de A^{-1} terão que ser, em módulo, menores que 1. Podemos verificar que os autovalores de A são dados por

$$\mu_i = 1 + 4\sigma \operatorname{sen}^2 \frac{i\pi}{2M}, \quad i = 1, 2, 3, \dots, M-1$$

e são todos maiores que 1. Como os autovalores de A são não nulos, existe a inversa A^{-1} e seus autovalores são, respectivamente, o inverso dos autovalores de A . Logo, todos os autovalores de A^{-1} são menores que 1. Portanto, independente de σ , o método implícito é incondicionalmente estável.

Geralmente os métodos que envolvem a resolução de sistemas lineares, como o método em questão, são incondicionalmente estáveis, isto é, serão sempre estáveis, independente da malha. Com isso, o esforço computacional pode ser reduzido usando-se uma malha menos fina.

Algoritmo para o Método Implícito

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Geração da matriz dos coeficientes

Para $i = 1$ a $M - 1$ faça

$$A(i, i) = 1 + 2\sigma$$

Fim para

Para $i = 1$ a $M - 2$ faça

$$A(i + 1, i) = -\sigma$$

$$A(i, i + 1) = A(i + 1, i)$$

Fim para

5) Cálculo das aproximações

Para $j = 2$ a $N + 1$ faça

Resolva o sistema linear $AU_j = U_{j-1}$ em 4.24.

Fim para

Fim Para

Solução $\rightarrow U(1 \text{ a } M + 1, 1 \text{ a } N + 1)$

6) Fim do algoritmo

4.3.3 Método de Crank-Nicolson

Esse método é a média aritmética do método explícito com método implícito. Veja o desenvolvimento a seguir

Aplicando a discretização, na Equação do Calor, no ponto (x_i, t_j) , pelo método explícito temos

$$U_{i,j+1} = \sigma U_{i-1,j} + (1 - 2\sigma)U_{i,j} + \sigma U_{i+1,j} \quad (4.30)$$

e, aplicando agora a discretização no ponto (x_i, t_{j+1}) , temos pelo método implícito

$$-\sigma U_{i-1,j+1} + (1 + 2\sigma)U_{i,j+1} - \sigma U_{i+1,j+1} = U_{i,j} \quad (4.31)$$

Fazendo a média aritmética de 4.30 com 4.31, segue

$$-\frac{\sigma}{2}U_{i-1,j+1} + (1 + \sigma)U_{i,j+1} - \frac{\sigma}{2}U_{i+1,j+1} = \frac{\sigma}{2}U_{i-1,j} + (1 - \sigma)U_{i,j} + \frac{\sigma}{2}U_{i+1,j} \quad (4.32)$$

donde obtemos a equação à diferenças do método em questão.

Uma outra maneira, de obter o método considerado, é discretizar a Equação do Calor em $(x_i, t_{j+\frac{1}{2}})$, o ponto médio entre (x_i, t_j) e (x_i, t_{j+1}) . Usando diferenças centradas para u_t e diferenças centradas de 2ª ordem para u_{xx} , após algumas operações, teremos:

$$\frac{U_{i,j+1} - U_{i,j}}{k} = \frac{a}{2h^2}(U_{i-1,j} - 2U_{i,j} + U_{i+1,j} + U_{i-1,j+1} - 2U_{i,j+1} + U_{i+1,j+1})$$

O método de Crank-Nicolson tem a seguinte molécula computacional:

Figura 4.4: Molécula computacional do método de Crank-Nicolson

O método de Crank-Nicolson exige a resolução de um sistema linear para cada estágio. Para entender essa afirmação, fixando um estágio j e fazendo i variar de 2 a $M-1$, teremos, aplicando 4.32, o seguinte sistema de equações:

$$\begin{cases}
 -\frac{\sigma}{2}U_{1,j+1} + (1+\sigma)U_{2,j+1} - \frac{\sigma}{2}U_{3,j+1} & = & \frac{\sigma}{2}U_{1,j} + (1-\sigma)U_{2,j} + \frac{\sigma}{2}U_{3,j} \\
 -\frac{\sigma}{2}U_{2,j+1} + (1+\sigma)U_{3,j+1} - \frac{\sigma}{2}U_{4,j+1} & = & \frac{\sigma}{2}U_{2,j} + (1-\sigma)U_{3,j} + \frac{\sigma}{2}U_{4,j} \\
 \vdots & \vdots & \vdots \\
 -\frac{\sigma}{2}U_{M-2,j+1} + (1+\sigma)U_{M-1,j+1} - \frac{\sigma}{2}U_{M,j+1} & = & \frac{\sigma}{2}U_{M-2,j} + (1-\sigma)U_{M-1,j} + \frac{\sigma}{2}U_{M,j} \\
 -\frac{\sigma}{2}U_{M-1,j+1} + (1+\sigma)U_{M,j+1} - \frac{\sigma}{2}U_{M+1,j+1} & = & \frac{\sigma}{2}U_{M-1,j} + (1-\sigma)U_{M,j} + \frac{\sigma}{2}U_{M+1,j}
 \end{cases}$$

$$\Downarrow$$

$$\underbrace{\begin{bmatrix} 1+\sigma & -\frac{\sigma}{2} & 0 & \dots & 0 \\ -\frac{\sigma}{2} & 1+\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1+\sigma & -\frac{\sigma}{2} \\ 0 & \dots & 0 & -\frac{\sigma}{2} & 1+\sigma \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{2,j+1} \\ U_{3,j+1} \\ \vdots \\ U_{M-1,j+1} \\ U_{M,j+1} \end{bmatrix}}_{U_{j+1}} =$$

$$\underbrace{\begin{bmatrix} 1-\sigma & \frac{\sigma}{2} & 0 & \dots & 0 \\ \frac{\sigma}{2} & 1-\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1-\sigma & \frac{\sigma}{2} \\ 0 & \dots & 0 & \frac{\sigma}{2} & 1-\sigma \end{bmatrix}}_B \underbrace{\begin{bmatrix} U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M-1,j} \\ U_{M,j} \end{bmatrix}}_{U_j} + \underbrace{\begin{bmatrix} \frac{\sigma}{2}(U_{1,j} + U_{1,j+1}) \\ 0 \\ \vdots \\ 0 \\ \frac{\sigma}{2}(U_{M+1,j} + U_{M+1,j+1}) \end{bmatrix}}_b$$

Finalizando, segue

$$AU_{j+1} = BU_j + b \quad (4.33)$$

Portanto, analogamente ao método implícito, teremos que resolver o sistema linear 4.33 para cada estágio j . Observe que a matriz A é diagonalmente dominante, logo A é inversível e o sistema considerado tem solução única.

Erro de Truncamento Local

Temos

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{a}{2h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j} + u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}) + \tau_{i,j} \quad (4.34)$$

Expandindo $u_{i,j}$, $u_{i-1,j}$, $u_{i+1,j}$, $u_{i-1,j+1}$, $u_{i,j+1}$ e $u_{i+1,j+1}$, em série de Taylor, em torno do ponto $(x_i, t_{j+\frac{1}{2}})$, obteremos

$$\tau_{i,j} = \frac{k^2}{24}u_{ttt} - \frac{ah^2}{12}u_{xxxx} - \frac{ak^2}{8}u_{xxt} - \frac{ah^2k^2}{96}u_{xxxxt} + O(h^4) + O(k^4) = O(h^2 + k^2)$$

Portanto, o método de Crank-Nicolson é consistente e de ordem 2.

Em comparação aos métodos explícito e implícito, o método de Crank-Nicolson gera aproximações melhores, devido ao mesmo ser de ordem 2.

Estabilidade

Analogamente ao que foi feito para os métodos explícito e implícito, pelo critério de Von Neumann chegaremos a

$$e^{k\lambda} = \frac{1 - 4\sigma \operatorname{sen}^2\left(\frac{\theta}{2}\right)}{1 + 4\sigma \operatorname{sen}^2\left(\frac{\theta}{2}\right)} \quad (4.35)$$

Note que $|e^{k\lambda}|$ é sempre menor que 1, logo o método de Crank-Nicolson é incondicionalmente estável.

Algoritmo para o Método de Crank-Nicolson

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

$$2) \text{ Cálculo dos espaçamentos } h, k \text{ e de sigma} \Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = \alpha \frac{k}{h^2} \end{array} \right.$$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Geração da matriz dos coeficientes

Para $i = 1$ a $M - 1$ faça

$$A(i, i) = 1 + \sigma$$

$$B(i, i) = 1 - \sigma$$

Fim para

Para $i = 1$ a $M - 2$ faça

$$A(i + 1, i) = -\frac{\sigma}{2}$$

$$A(i, i + 1) = A(i + 1, 1)$$

$$B(i + 1, i) = -A(i + 1, i)$$

$$B(i, i + 1) = -A(i + 1, 1)$$

Fim para

5) Cálculo das aproximações

Para $j = 1$ a N faça

$$\text{Resolva o sistema linear } AU_{j+1} = BU_j + b \text{ em 4.33.}$$

Fim para

Fim Para

$$\text{Solução} \rightarrow U(1 \text{ a } M + 1, 1 \text{ a } N + 1)$$

6) Fim do algoritmo

4.3.4 Método Saul'yev

O método Saul'yev, desenvolvido por Saul'yev [17], é um método incondicionalmente estável. Acompanhe o processo de obtenção do método.

$$\begin{aligned}
 u_t &= au_{xx} \\
 \Downarrow \\
 \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{u_x(x_{i+\frac{1}{2}}, t_j) - u_x(x_{i-\frac{1}{2}}, t_j)}{h}
 \end{aligned} \tag{4.36}$$

Substituímos $u_x(x_{i-\frac{1}{2}}, t_j)$ em 4.36 pelo seu correspondente do estágio $j + 1$, ou seja, do estágio acima, e a seguir usamos diferenças centradas para u_x . Assim sendo, segue

$$\begin{aligned}
 \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{u_x(x_{i+\frac{1}{2}}, t_j) - u_x(x_{i-\frac{1}{2}}, t_j)}{h} \\
 \Downarrow \\
 \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{u_x(x_{i+\frac{1}{2}}, t_j) - u_x(x_{i-\frac{1}{2}}, t_{j+1})}{h}
 \end{aligned}$$

Usando agora diferenças centradas para u_x , temos

$$\begin{aligned}
 \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{\frac{U_{i+1,j} - U_{i,j}}{h} - \frac{U_{i,j+1} - U_{i-1,j+1}}{h}}{h} \\
 \Downarrow \\
 \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{U_{i+1,j} - U_{i,j} - U_{i,j+1} + U_{i-1,j+1}}{h^2}
 \end{aligned} \tag{4.37}$$

$$\begin{aligned}
 \Downarrow \\
 U_{i,j+1} - U_{i,j} &= \sigma(U_{i+1,j} - U_{i,j} - U_{i,j+1} + U_{i-1,j+1})
 \end{aligned}$$

$$\begin{aligned}
 \Downarrow \\
 U_{i,j+1} &= \frac{(1 - \sigma)}{(1 + \sigma)} U_{i,j} + \frac{\sigma}{(1 + \sigma)} (U_{i+1,j} + U_{i-1,j+1})
 \end{aligned} \tag{4.38}$$

onde $\sigma = a \frac{k}{h^2}$.

Assim sendo, em 4.38, temos a equação do método Saul'yev.

Esse método é representado pela seguinte molécula computacional:

Figura 4.5: Molécula computacional do método Saul'yev - Esquerda $\rightarrow$ Direita

Observe que a molécula acima representa 4.38, onde pode se notar que, quando aplicado, realiza-se, para cada estágio, o processo de cálculo das aproximações da esquerda para a direita. Podemos também realizar esses cálculos da direita para a esquerda. Para isso, basta substituir $u_x(x_{i+\frac{1}{2}}, t_j)$ em 4.36 pelo seu correspondente no estágio $j+1$, e então seguir o mesmo procedimento anterior até obter a equação do método, isto é,

$$\frac{U_{i,j+1} - U_{i,j}}{k} = a \frac{u_x(x_{i+\frac{1}{2}}, t_{j+1}) - u_x(x_{i-\frac{1}{2}}, t_j)}{h}$$

daí segue

$$\begin{aligned} \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{\frac{U_{i+1,j+1} - U_{i,j+1}}{h} - \frac{U_{i,j} - U_{i-1,j}}{h}}{h} \\ &\Downarrow \\ \frac{U_{i,j+1} - U_{i,j}}{k} &= a \frac{U_{i+1,j+1} - U_{i,j+1} - U_{i,j} + U_{i-1,j}}{h^2} \\ &\Downarrow \\ U_{i,j+1} - U_{i,j} &= \sigma (U_{i+1,j+1} - U_{i,j+1} - U_{i,j} + U_{i-1,j}) \\ &\Downarrow \\ U_{i,j+1} &= \frac{(1 - \sigma)}{(1 + \sigma)} U_{i,j} + \frac{\sigma}{(1 + \sigma)} (U_{i-1,j} + U_{i+1,j+1}) \end{aligned} \quad (4.39)$$

Para aplicarmos 4.39 teremos, para cada estágio, de caminhar da direita para a esquerda.

A molécula computacional que representa 4.39 é dada por:

Figura 4.6: Molécula computacional do método Saul'yev - Direita $\rightarrow$ Esquerda

Definida uma malha, podemos aplicar as fórmulas 4.38 e 4.39 de várias maneiras na resolução numérica da Equação do Calor. Podemos, por exemplo,

- Aplicar 4.38 e 4.39 alternando-os nos estágios.
- Aplicar 4.38 e 4.39 em um mesmo estágio, e tomar como aproximação final a média dos dois.

Erro de Truncamento Local

Temos que

$$\frac{u_{i,j+1} - u_{i,j}}{k} = a \frac{u_{i+1,j} - u_{i,j} - u_{i,j+1} + u_{i-1,j+1}}{h^2} + \tau_{i,j} \quad (4.40)$$

Calculando o ETL, mediante a expansão série de Taylor de $u_{i,j+1}$, $u_{i+1,j}$, $u_{i,j+1}$ e $u_{i-1,j+1}$ em torno do ponto (x_i, t_j) , verificaremos que ele é ordem $O(\frac{k}{h} + h^2)$, portanto **condicionalmente consistente**. Isso pode ser observado se, por exemplo, k for múltiplo de h , isto é, $k = mh$, onde m é um inteiro positivo. Nesse caso, a equação 4.40, quando h e k tenderem a zero, não convergirá para a Equação do Calor original, pois haverá sempre um resíduo de valor igual a m .

Estabilidade

De 4.39, temos que:

$$(1 + \sigma)U_{i,j+1} = (1 - \sigma)U_{i,j} + \sigma(U_{i-1,j} + U_{i+1,j+1})$$

Pelo critério de Von Neumann, segue

$$(1 + \sigma)U_{i,j+1} = (1 - \sigma)U_{i,j} + \sigma(U_{i-1,j} + U_{i+1,j+1})$$

$$\Downarrow$$

$$(1 + \sigma)e^{k\lambda} = (1 - \sigma) + \sigma(e^{k\lambda}e^{-h\beta I} + e^{h\beta I})$$

$$\Downarrow$$

$$(1 + \sigma)e^{k\lambda} = (1 - \sigma) + \sigma e^{k\lambda}e^{-h\beta I} + \sigma e^{h\beta I}$$

$$\Downarrow$$

$$(1 + \sigma - \sigma e^{-h\beta I})e^{k\lambda} = 1 - \sigma + \sigma e^{h\beta I}$$

$$\Downarrow$$

$$e^{k\lambda} = \frac{1 - \sigma + \sigma e^{h\beta I}}{1 + \sigma - \sigma e^{-h\beta I}}$$

$$\Downarrow$$

$$e^{k\lambda} = \frac{1 - \sigma + \sigma \cos(h\beta) + I\sigma \sin(h\beta)}{1 + \sigma - \sigma \cos(h\beta) + I\sigma \sin(h\beta)}$$

$$\Downarrow$$

$$|e^{k\lambda}| = \frac{\sqrt{[1 - \sigma + \sigma \cos(h\beta)]^2 + \sigma^2 \sin^2(h\beta)}}{\sqrt{[1 + \sigma - \sigma \cos(h\beta)]^2 + \sigma^2 \sin^2(h\beta)}}$$

$$\Downarrow$$

$$|e^{k\lambda}| = \frac{\sqrt{[1 - \sigma(1 + \cos(h\beta))]^2 + \sigma^2 \sin^2(h\beta)}}{\sqrt{[1 + \sigma(1 + \cos(h\beta))]^2 + \sigma^2 \sin^2(h\beta)}} \quad (4.41)$$

Analisando 4.41, vemos que $|e^{k\lambda}|$ é sempre menor que 1. Portanto, o método Saul'yev é incondicionalmente estável.

Segue o algoritmo do método Saul'yev em 3 modos (Esquerda $\rightarrow$ Direita, Direita $\rightarrow$ Esquerda e a média ED-DE).

Algoritmo para o Método Saul'yev

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

$$2) \text{ Cálculo dos espaçamentos } h, k \text{ e de sigma } \Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Cálculo das aproximações

Para calcular as aproximações da esquerda para a direita (ED), use

Para $j = 1$ a N faça

Para $i = 2$ a M faça

$$U(i, j + 1) = \frac{(1 - \sigma)}{(1 + \sigma)} U(i, j) + \frac{\sigma}{(1 + \sigma)} (U(i + 1, j) + U(i - 1, j + 1))$$

Fim para

Fim para

Para calcular as aproximações da direita para a esquerda (DE), use

Para $j = 1$ a N faça

Para $i = M$ a 2 faça

$$U(i, j+1) = \frac{(1-\sigma)}{(1+\sigma)}U(i, j) + \frac{\sigma}{(1+\sigma)}(U(i-1, j) + U(i+1, j+1))$$

Fim para

Fim para

Para calcular as aproximações, em cada estágio, tomando como resultado a média entre ED e DE, use

Para $j = 1$ a N faça

Para $i = 2$ a M faça

$$AUX1(i) = \frac{(1-\sigma)}{(1+\sigma)}U(i, j) + \frac{\sigma}{(1+\sigma)}(U(i+1, j) + U(i-1, j+1))$$

Fim para

Para $i = M$ a 2 faça

$$AUX2(i) = \frac{(1-\sigma)}{(1+\sigma)}U(i, j) + \frac{\sigma}{(1+\sigma)}(U(i-1, j) + U(i+1, j+1))$$

Fim para

Para $i = 2$ a M faça

$$U(i, j+1) = \frac{AUX1(i) + AUX2(i)}{2}$$

Fim para

Solução $\rightarrow U(1 \text{ a } M+1, 1 \text{ a } N+1)$

5) Fim do algoritmo

4.3.5 Método Hopscotch

O método hopscotch é um método muito usado para dimensões altas, pois reduz o esforço computacional necessário para realizar os cálculos. Essa rapidez vem do fato, como veremos mais adiante, das aproximações, em um determinado nível de tempo j , serem calculados explicitamente em função das aproximações do nível anterior. A idéia básica do método consiste em aplicar, em um dado estágio j , os métodos explícito e implícito em pontos alternados. Observe a próxima figura . No nível 1 usamos primeiro o método explícito para calcular todos as aproximações nos pontos marcados por um círculo, em seguida voltamos e calculamos as aproximações nos pontos marcados por um quadrado usando o método implícito. No nível 2 a mesma coisa, usamos primeiro o método explícito nos pontos marcados por um círculo e em seguida, nos pontos marcados por um quadrado, o método implícito, e assim sucessivamente. A maneira como aplicamos a sequência dos cálculos faz com que o método implícito passe a ser explícito, não necessitando a resolução

de um sistema linear. Acompanhe o desenvolvimento a seguir.

Figura 4.7: Ilustração gráfica do método Hopscotch

Para j fixado, aplicamos primeiro o método explícito em pontos alternados

$$U_{i,j+1} = \sigma U_{i-1,j} + (1 - 2\sigma)U_{i,j} + \sigma U_{i+1,j}$$

e em seguida o método implícito,

$$-\sigma U_{i-1,j+1} + (1 + 2\sigma)U_{i,j+1} - \sigma U_{i+1,j+1} = U_{i,j}$$

mas observe que os pontos $U_{i-1,j+1}$ e $U_{i+1,j+1}$ já foram calculados quando o método explícito foi aplicado. Portanto podemos calcular $U_{i,j+1}$ explicitamente. Assim sendo, basta isolar $U_{i,j+1}$, como é mostrado abaixo

$$U_{i,j+1} = \frac{U_{i,j} + \sigma U_{i+1,j+1} + \sigma U_{i-1,j+1}}{1 + 2\sigma}$$

Esse método é representado pela seguinte molécula computacional:

Figura 4.8: Molécula computacional do método Hopscotch

Gordon [10] demonstra a consistência e a estabilidade desse método.

Algoritmo para Método Hopscotch

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

$$2) \text{ Cálculo dos espaçamentos } h, k \text{ e de sigma } \Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Cálculo das aproximações

Para $j = 1$ a N faça

Se resto de $(j/2) \neq 0$ então faça

Para $i = 2$ a M com incremento 2 faça

$$U(i, j + 1) = \sigma U(i - 1, j) + (1 - 2\sigma)U(i, j) + \sigma U(i + 1, j)$$

Fim para

Para $i = 3$ a M com incremento 2 faça

$$U_{i,j+1} = \frac{U_{i,j} + \sigma U_{i+1,j+1} + \sigma U_{i-1,j+1}}{1 + 2\sigma}$$

Fim para

Senão

Para $i = 3$ a M com incremento 2 faça

$$U(i, j + 1) = \sigma U(i - 1, j) + (1 - 2\sigma)U(i, j) + \sigma U(i + 1, j)$$

Fim para

Para $i = 2$ a M com incremento 2 faça

$$U_{i,j+1} = \frac{U_{i,j} + \sigma U_{i+1,j+1} + \sigma U_{i-1,j+1}}{1 + 2\sigma}$$

Fim para

Fim se

Fim Para

Solução $\rightarrow U(1 \text{ a } M + 1, 1 \text{ a } N + 1)$

5) Fim do algoritmo

4.4 Métodos para Equações Parabólicas com Coeficientes Variáveis

A maioria dos problemas envolvendo equações parabólicas envolvem coeficientes variáveis. Grande parte dos métodos discutidos anteriormente podem ser generalizados para equações lineares com coeficientes variáveis, ou seja, $a = a(x, t)$ na Equação do Calor. Trabalhar com coeficientes variáveis acarretam algumas dificuldades como:

- As matrizes continuarão a serem diagonais dominantes, logo inversíveis, mas σ deixará de ser constante.
- Aumento do esforço computacional devido a inconstância da matriz dos coeficientes. Por cada estágio, por exemplo, no método implícito, terá que ser calculada uma nova matriz.
- Não poderemos usar o Critério de Von Neumann para fazer a análise da estabilidade de maneira global, apenas localmente.

Quando $a = a(x, t, u)$ ou $a = a(u)$ temos um problema não linear, nesse caso o método explícito é usado normalmente. Já nos métodos implícito e de Cranck-Nicolson obtemos

um sistema de equações não lineares, e podemos resolvê-lo por aproximações sucessivas ou pelo método de Newton, com a aproximação inicial calculada pelo método explícito. Apresentaremos a seguir alguns métodos numéricos para resolução de equações parabólicas com coeficientes variáveis. Cada método que irá ser representado trabalhará com uma equação parabólica específica.

4.4.1 Método de Lees

Método desenvolvido por Lees [15], consiste na resolução numérica da equação não linear

$$b(u)u_t = (a(u)u_x)_x, \quad a(u) > 0 \quad b(u) > 0 \quad (4.42)$$

onde Lees considerou a construção de métodos com 3 características:

- Que exigissem apenas a resolução de sistemas lineares.
- Preservassem a estabilidade.
- Preservassem a precisão, ou seja, a ordem de convergência.

Acompanhe o desenvolvimento do método.

Temos de 4.42

$$b(u)u_t = \underbrace{(a(u) \overbrace{u_x}^I)}_{II}_x$$

Aplicando diferenças centradas em I, e de depois em II, de modo a ter-se espaçamento h , e diferenças progressivas em u_t , segue

$$\begin{aligned} b(U_{i,j}) \frac{U_{i,j+1} - U_{i,j}}{2k} &= \left(a(U_{i,j}) \frac{U_{i+\frac{1}{2},j} - U_{i-\frac{1}{2},j}}{h} \right)_x \\ &\Downarrow \\ b(U_{i,j}) \frac{U_{i,j+1} - U_{i,j}}{2k} &= \frac{1}{h^2} \left[(a(U_{i-\frac{1}{2},j})(U_{i+1,j} - U_{i,j}) - a(U_{i+\frac{1}{2},j})(U_{i,j} - U_{i-1,j})) \right] \\ &\Downarrow \end{aligned}$$

$$b_{i,j}(U_{i,j+1} - U_{i,j}) = \sigma \left[a_{i+\frac{1}{2},j}(U_{i+1,j} - U_{i,j}) - a_{i-\frac{1}{2},j}(U_{i,j} - U_{i-1,j}) \right]$$

onde $\sigma = \frac{2k}{h^2}$, $a_{i+\frac{1}{2},j} = a(U_{i+\frac{1}{2},j})$, $a_{i-\frac{1}{2},j} = a(U_{i-\frac{1}{2},j})$ e $b_{i,j} = b(U_{i,j})$

Continuando

$$U_{i,j+1} = \frac{\sigma a_{i-\frac{1}{2},j}}{b_{i,j}} U_{i-1,j} + \frac{\sigma(b_{i,j} - a_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j})}{b_{i,j}} U_{i,j} + \frac{\sigma a_{i+\frac{1}{2},j}}{b_{i,j}} U_{i+1,j}$$

Fazendo $\alpha_{i,j} = \frac{\sigma a_{i-\frac{1}{2},j}}{b_{i,j}}$, $\beta_{i,j} = \frac{\sigma(b_{i,j} - a_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j})}{b_{i,j}}$ e $\gamma_{i,j} = \frac{\sigma a_{i+\frac{1}{2},j}}{b_{i,j}}$, segue

$$U_{i,j+1} = \alpha_{i,j} U_{i-1,j} + \beta_{i,j} U_{i,j} + \gamma_{i,j} U_{i+1,j} \quad (4.43)$$

Segundo Lees, a relação 4.43 será instável quando $a \equiv b \equiv 1$, mas se substituirmos $U_{i-1,j}$, $U_{i,j}$ e $U_{i+1,j}$ pelas médias

$$\begin{aligned} U_{i-1,j} &= \frac{1}{3}(U_{i-1,j-1} + U_{i-1,j} + U_{i-1,j+1}) \\ U_{i,j} &= \frac{1}{3}(U_{i,j-1} + U_{i,j} + U_{i,j+1}) \\ U_{i+1,j} &= \frac{1}{3}(U_{i+1,j-1} + U_{i+1,j} + U_{i+1,j+1}) \end{aligned} \quad (4.44)$$

e também os coeficientes, fazendo

$$\begin{aligned} a_{i-\frac{1}{2},j} &= a(U_{i-\frac{1}{2},j}) = a\left(\frac{U_{i-1,j} + U_{i,j}}{2}\right) \\ a_{i+\frac{1}{2},j} &= a(U_{i+\frac{1}{2},j}) = a\left(\frac{U_{i,j} + U_{i+1,j}}{2}\right) \end{aligned} \quad (4.45)$$

o método será, segundo Lees, consistente de ordem 2, estável e convergente, para h e k suficientemente pequenos, com erro global de ordem 2. Assim sendo, continuando o desenvolvimento, fazendo as substituições 4.44 em 4.43, teremos

$$\begin{aligned}
& -\alpha_{i,j}U_{i-1,j+1} + (3 - \beta_{i,j})U_{i,j+1} - \gamma_{i,j}U_{i+1,j+1} = \\
& \alpha_{i,j}U_{i-1,j-1} + \beta_{i,j}U_{i,j-1} + \gamma_{i,j}U_{i+1,j-1} \\
& + \alpha_{i,j}U_{i-1,j} + \beta_{i,j}U_{i,j} + \gamma_{i,j}U_{i+1,j}
\end{aligned} \tag{4.46}$$

A aplicação desse exige a resolução de um sistema linear para cada estágio.

Um dos problemas desse método está no fato do mesmo envolver 3 níveis de tempo, tornando assim necessário o desenvolvimento, como mostraremos mais adiante, de um método para calcular as aproximações iniciais para o segundo nível.

Podemos representar o método pela seguinte molécula computacional:

Figura 4.9: Molécula computacional do método de Lees

Acompanhe o desenvolvimento do sistema gerado pelo método.

Considere a equação 4.42 em questão com as condições auxiliares da Equação do Calor, definidas em uma malha $(M + 1) \times (N + 1)$ pontos. Fixando um estágio j e variando i de 2 a M , teremos, aplicando 4.46, o seguinte sistema de equações:

$$\left\{ \begin{array}{l}
 \alpha_{2,j}U_{1,j+1} + (3 - \beta_{2,j})U_{2,j+1} - \gamma_{2,j}U_{3,j+1} = \\
 \alpha_{2,j}U_{1,j-1} + \beta_{2,j}U_{2,j-1} + \gamma_{2,j}U_{3,j-1} \\
 + \alpha_{2,j}U_{1,j} + \beta_{2,j}U_{2,j} + \gamma_{2,j}U_{3,j} \\
 \\
 -\alpha_{3,j}U_{2,j+1} + (3 - \beta_{3,j})U_{3,j+1} - \gamma_{3,j}U_{4,j+1} = \\
 \alpha_{3,j}U_{2,j-1} + \beta_{3,j}U_{3,j-1} + \gamma_{3,j}U_{4,j-1} \\
 + \alpha_{3,j}U_{2,j} + \beta_{3,j}U_{3,j} + \gamma_{3,j}U_{4,j} \\
 \vdots \\
 \vdots \\
 -\alpha_{M-1,j}U_{M-2,j+1} + (3 - \beta_{M-1,j})U_{M-1,j+1} - \gamma_{M-1,j}U_{M,j+1} = \\
 \alpha_{M-1,j}U_{M-2,j-1} + \beta_{M-1,j}U_{M-1,j-1} + \gamma_{M-1,j}U_{M,j-1} \\
 + \alpha_{M-1,j}U_{M-2,j} + \beta_{M-1,j}U_{M-1,j} + \gamma_{M-1,j}U_{M,j} \\
 \\
 -\alpha_{M,j}U_{M-1,j+1} + (3 - \beta_{M,j})U_{M,j+1} - \gamma_{M,j}U_{M+1,j+1} = \\
 \alpha_{M,j}U_{M-1,j-1} + \beta_{M,j}U_{M,j-1} + \gamma_{M,j}U_{M+1,j-1} \\
 + \alpha_{M,j}U_{M-1,j} + \beta_{M,j}U_{M,j} + \gamma_{M,j}U_{M+1,j}
 \end{array} \right.$$

Colocando na forma matricial, segue

$$\underbrace{\begin{bmatrix} 3 - \beta_{2,j} & -\gamma_{2,j} & 0 & \dots & 0 \\ -\alpha_{3,j} & 3 - \beta_{3,j} & -\gamma_{3,j} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha_{M-1,j} & 3 - \beta_{M-1,j} & -\gamma_{M-1,j} \\ 0 & \dots & 0 & -\alpha_{M,j} & 3 - \beta_{M,j} \end{bmatrix}}_{A_j} \underbrace{\begin{bmatrix} U_{2,j+1} \\ U_{3,j+1} \\ \vdots \\ U_{M-1,j+1} \\ U_{M,j+1} \end{bmatrix}}_{U_{j+1}} = \\
 \underbrace{\begin{bmatrix} \beta_{2,j} & \gamma_{2,j} & 0 & \dots & 0 \\ \alpha_{3,j} & \beta_{3,j} & \gamma_{3,j} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \alpha_{M-1,j} & \beta_{M-1,j} & \gamma_{M-1,j} \\ 0 & \dots & 0 & \alpha_{M,j} & \beta_{M,j} \end{bmatrix}}_{B_j} \underbrace{\begin{bmatrix} U_{2,j-1} \\ U_{3,j-1} \\ \vdots \\ U_{M-1,j-1} \\ U_{M,j-1} \end{bmatrix}}_{U_{j-1}} + \\
 \underbrace{\begin{bmatrix} \beta_{2,j} & \gamma_{2,j} & 0 & \dots & 0 \\ \alpha_{3,j} & \beta_{3,j} & \gamma_{3,j} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \alpha_{M-1,j} & \beta_{M-1,j} & \gamma_{M-1,j} \\ 0 & \dots & 0 & \alpha_{M,j} & \beta_{M,j} \end{bmatrix}}_{B_j} \underbrace{\begin{bmatrix} U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M-1,j} \\ U_{M,j} \end{bmatrix}}_{U_{j-1}} +$$

$$\underbrace{\begin{bmatrix} \alpha_{2,j}(U_{1,j-1} + U_{1,j} + U_{1,j+1}) \\ 0 \\ \vdots \\ 0 \\ \gamma_{M,j}(U_{M+1,j-1} + U_{M+1,j} + U_{M+1,j+1}) \end{bmatrix}}_b \Rightarrow A_j U_{j+1} = B_j U_{j-1} + B_j U_j + b \quad (4.47)$$

Portanto, para cada estágio, teremos que resolver o sistema linear 4.47, onde

$$\begin{aligned} \alpha_{i,j} &= \frac{\sigma a_{i-\frac{1}{2},j}}{b_{i,j}} = \frac{\sigma a(\frac{U_{i,j} + U_{i-1,j}}{2})}{b(U_{i,j})} \\ \beta_{i,j} &= \frac{\sigma(b_{i,j} - a_{i+\frac{1}{2}} - a_{i-\frac{1}{2},j})}{b_{i,j}} = \frac{\sigma \left[b(U_{i,j}) - a(\frac{U_{i+1,j} + U_{i,j}}{2}) - a(\frac{U_{i,j} + U_{i-1,j}}{2}) \right]}{b(U_{i,j})} \\ \gamma_{i,j} &= \frac{\sigma a_{i+\frac{1}{2},j}}{b_{i,j}} = \frac{\sigma a(\frac{U_{i+1,j} + U_{i,j}}{2})}{b(U_{i,j})} \end{aligned}$$

Observe que, para cada estágio, as matrizes A_j e B_j terão que serem calculadas. Segue abaixo o algoritmo para o método.

Para que possamos aplicar esse método, é necessário antes calcularmos aproximações para o segundo nível. Isso pode ser feito usando 4.43 para calcular as aproximações nesse estágio. Tendo calculado esses valores, poderemos começar a aplicar o Método de Lees.

Algoritmo para o Método de Lees

1) Início

Entrada de dados. Entre com

$a(u) \rightarrow$ Coeficiente de u_{xx} , com $a(u) > 0$.

$b(u) \rightarrow$ Coeficiente de u_t , com $b(u) > 0$.

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$g(t) \rightarrow$ Condição de fronteira em $x = 0$.

$h(t) \rightarrow$ Condição de fronteira em $x = L$.

$$2) \text{ Cálculo dos espaçamentos } h, k \text{ e de sigma} \Rightarrow \left\{ \begin{array}{l} h = \frac{|le - ld|}{M} \quad k = \frac{T}{N} \quad \sigma = \frac{2k}{h^2} \end{array} \right.$$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

$$U(M + 1, j) = h(t)$$

Fim para

4) Cálculo das aproximações

Para $i = 2$ a M faça

$$\alpha = \frac{\sigma a \left(\frac{U(i,1) + U(i-1,1)}{2} \right)}{b(U(i,1))}$$

$$\beta = \frac{\sigma \left[b(U(i,j)) - a \left(\frac{U(i+1,j) + U(i,j)}{2} \right) - a \left(\frac{U(i,j) + U(i-1,j)}{2} \right) \right]}{b(U(i,j))}$$

$$\gamma = \frac{\sigma a \left(\frac{U(i+1,j) + U(i,j)}{2} \right)}{b(U(i,j))}$$

$$U(i, 2) = \alpha U(i - 1, 1) + \beta U(i, 1) + \gamma U(i + 1, 1)$$

Fim para

Para $j = 2$ a N faça

Para $i = 2$ a M faça

$$A(i, i) = 3 - \frac{\sigma \left[b(U(i,j)) - a \left(\frac{U(i+1,j) + U(i,j)}{2} \right) - a \left(\frac{U(i,j) + U(i-1,j)}{2} \right) \right]}{b(U(i,j))}$$

$$B(i, i) = \frac{\sigma \left[b(U(i, j)) - a\left(\frac{U(i+1, j) + U(i, j)}{2}\right) - a\left(\frac{U(i, j) + U(i-1, j)}{2}\right) \right]}{b(U(i, j))}$$

Fim para

Para $i = 1$ a $M - 2$ faça

$$A(i + 1, i) = -\frac{\sigma a\left(\frac{U(i, j) + U(i-1, j)}{2}\right)}{b(U(i, j))}$$

$$A(i, i + 1) = -\frac{\sigma a\left(\frac{U(i+1, j) + U(i, j)}{2}\right)}{b(U(i, j))}$$

$$B(i + 1, i) = -A(i + 1, i)$$

$$B(i, i + 1) = -A(i, i + 1)$$

Fim para

$$\alpha = \frac{\sigma a\left(\frac{U(2, 1) + U(1, 1)}{2}\right)}{b(U(2, j))}$$

$$\gamma = \frac{\sigma a\left(\frac{U(M+1, j) + U(M, j)}{2}\right)}{b(U(M + 1, j))}$$

$$b(1) = \alpha(U(1, j - 1) + U(1, j) + U(1, j + 1))$$

$$b(M - 1) = \gamma(U(M + 1, j - 1) + U(M + 1, j) + U(M + 1, j + 1))$$

Resolva o sistema linear $A_j U_{j+1} = B_j U_{j-1} + B_j U_j + b$ em 4.47

Solução $\rightarrow U(1 \text{ a } M + 1, 1 \text{ a } N + 1)$

Fim para

5) Fim do algoritmo

4.4.2 Método das Linhas

A essência do método das linhas consiste em transformar a EDP a ser resolvida em um sistema de EDOs, com valor inicial, o qual pode ser resolvido por métodos numéricos para equações ordinárias. O esquema de aplicação desse método consiste nos seguintes passos:

1. Particione o intervalo do domínio no eixo x , e a seguir calcule os valores, usando as condições iniciais, que a função incógnita u assume nos pontos definidos pelo particionamento.

2. Discretize, na EDP de interesse, as derivadas em relação a x .
3. Aplique a EDP discretizada em cada um dos pontos definidos pelo item anterior.
4. Monte um sistema com as equações definidas no item anterior. Com isso teremos gerado um sistema de EDOs.
5. Defina um passo k no tempo e resolva o sistema de EDOs gerado, por algum método numérico apropriado para o mesmo.

Poderemos, dessa maneira, calcular aproximações ao longo do tempo. Acompanhe o desenvolvimento do exemplo a seguir.

Considere o seguinte problema

$$\begin{cases} u_t = u_{xx} + (u_x)^2, & x \in [0, 1], \quad t \in [0, T] \\ u(x, 0) = x(1 - x), & x \in [0, 1] \\ u(0, t) = 0, & t \in [0, T] \\ u(1, t) = \text{sen}(t), & t \in [0, T] \end{cases} \quad (4.48)$$

Seguindo os passos do esquema apresentado, temos:

1. Consideremos M partições em $[0, 1]$, e com $u(x, 0)$, a condição inicial, calculamos os valores nos pontos definidos pelo particionamento.
2. Discretizando as derivadas em relação a x na EDP do problema, usando diferenças centradas para u_x e diferenças centradas de 2ª ordem para u_{xx} , segue

$$U'_i(t) = \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} - \frac{(U_{i+1,j} - U_{i-1,j})^2}{4h^2}, \quad i = 2, 3, 4, \dots, M-1$$

$\Downarrow$

$$U'_i(t) = \alpha(U_{i-1,j} - 2U_{i,j} + U_{i+1,j}) - \beta(U_{i-1,j} - U_{i+1,j})^2$$

$\Downarrow$

$$\begin{cases} U_1(t) &= 0 \\ U'_2(t) &= \alpha(U_{1,j} - 2U_{2,j} + U_{3,j}) - \beta(U_{3,j} - U_{1,j})^2 \\ U'_3(t) &= \alpha(U_{2,j} - 2U_{3,j} + U_{4,j}) - \beta(U_{4,j} - U_{2,j})^2 \\ \vdots &\vdots \\ U'_{M-1}(t) &= \alpha(U_{M-2,j} - 2U_{M-1,j} + U_{M,j}) - \beta(U_{M,j} - U_{M-2,j})^2 \\ U'_M(t) &= \alpha(U_{M-1,j} - 2U_{M,j} + U_{M+1,j}) - \beta(U_{M+1,j} - U_{M-1,j})^2 \\ U_{M+1}(t) &= \text{sen}(t) \end{cases}$$

onde $\alpha = \frac{1}{h^2}$ e $\beta = \frac{1}{4h^2}$.

Desconsiderando, por simplicidade, o índice j , obtemos o seguinte sistema de EDOs

$$\left\{ \begin{array}{lcl} U_1(t) & = & 0 \\ U_2'(t) & = & \alpha(U_1 - 2U_2 + U_3) - \beta(U_3 - U_1)^2 \\ U_3'(t) & = & \alpha(U_2 - 2U_3 + U_4) - \beta(U_4 - U_2)^2 \\ \vdots & \vdots & \vdots \\ U_{M-1}'(t) & = & \alpha(U_{M-2} - 2U_{M-1} + U_M) - \beta(U_M - U_{M-2})^2 \\ U_M'(t) & = & \alpha(U_{M-1} - 2U_M + U_{M+1}) - \beta(U_{M+1} - U_{M-1})^2 \\ U_{M+1}(t) & = & \text{sen}(t) \end{array} \right.$$

com as seguintes condições iniciais em $t = 0$:

$$\left\{ \begin{array}{lll} U_1(0) & = & u(x_1, 0) = 0 \\ U_2(0) & = & u(x_2, 0) = x_2(1 - x_2) \\ U_3(0) & = & u(x_3, 0) = x_3(1 - x_3) \\ \vdots & \vdots & \vdots \\ U_{M-1}(0) & = & u(x_{M-1}, 0) = x_{M-1}(1 - x_{M-1}) \\ U_M(0) & = & u(x_M, 0) = x_M(1 - x_M) \\ U_{M+1}(0) & = & u(x_{M+1}, 0) = \text{sen}(0) = 0 \end{array} \right.$$

Com isso temos um sistema de EDOs com valor inicial. Resta agora resolver o sistema considerado por algum método numérico para obter aproximações $u(x, t)$ ao longo das linhas (x_i, t) . A figura abaixo ilustra a idéia do método.

Figura 4.10: Ilustração gráfica do método das linhas

A precisão aqui vai depender do método numérico que estiver sendo usado para resolver o sistema de EDOs. Para maiores detalhes sobre o método das linhas, ver Schiesser [18].

4.4.3 Método Box

O método box foi desenvolvido por Keller [13], e é um método que pode ser usado tanto para problemas lineares como não-lineares. Possui precisão de ordem 2 e é incondicionalmente estável. Acompanhe o desenvolvimento do exemplo a seguir.

Considere o seguinte problema:

$$\begin{cases} u_t = [a(x)u_x]_x + c(x)u \\ u(x, 0) = \psi(x) & x \in [0, 1] \\ \alpha_0 u(0, t) + \alpha_1 u_x(0, t) = \alpha_2(t) & t > 0 \\ \beta_0 u(1, t) + \beta_1 u_x(1, t) = \beta_2(t) & t > 0 \end{cases} \quad (4.49)$$

O primeiro passo é transformar a equação do problema em um sistema de equações de primeira ordem. Essa transformação pode ser realizada de várias maneiras. Assim sendo, consideraremos a transformação a seguir

$$v = a(x)u_x \quad (4.50)$$

$$v_x = u_t - c(x)u \quad (4.51)$$

Com essa transformação as condições auxiliares sofrem as seguintes alterações.

Por 4.50 temos

$$v(0, t) = a(0)u_x(0, t) \Rightarrow u_x(0, t) = \alpha v(0, t) \quad \text{onde} \quad \alpha = \frac{1}{a(0)} \quad (4.52)$$

$$v(1, t) = a(1)u_x(1, t) \Rightarrow u_x(1, t) = \beta v(1, t) \quad \text{onde} \quad \beta = \frac{1}{a(1)} \quad (4.53)$$

Substituindo 4.52 e 4.53 nas condições auxiliares, nesse caso, apenas as condições de fronteira, em 4.49 segue

$$\begin{aligned} u(x, 0) &= \psi(x) & x &\in [0, 1] \\ \alpha_0 u(0, t) + \alpha_3 v(0, t) &= \alpha_2(t) & t &> 0 \\ \beta_0 u(1, t) + \beta_3 v(1, t) &= \beta_2(t) & t &> 0 \end{aligned} \quad (4.54)$$

onde $\alpha_3 = \alpha\alpha_1$ e $\beta_3 = \beta\beta_1$.

Esse método pode ser usado para malhas de tamanho variável. Assim sendo, podemos definir uma malha da seguinte maneira

- M partições diferenciadas em $[0, 1]$. Com isso, os pontos definidos por esta partição não serão igualmente espaçados, ou seja, teremos

$$x_{i+1} = x_i + h_i, \quad i = 1, 2, 3, \dots, M$$

- N partições diferenciadas em $[0, T]$, onde T é o nível máximo de tempo. Da mesma forma que o ítem anterior, os pontos definidos por esta partição não serão igualmente espaçados, ou seja, teremos

$$t_{j+1} = t_j + k_j, \quad j = 1, 2, 3, \dots, N$$

Por simplicidade, trabalharemos com h_i e k_j constantes.

O método box é representado pela seguinte molécula computacional:

Figura 4.11: Molécula computacional do método box

O método box é obtido discretizando a EDP no ponto $(x_{i-\frac{1}{2}}, t_j)$, de modo que as aproximações sejam médias dos pontos das células da malha. Assim sendo, temos

$$x_{i-\frac{1}{2}} = \frac{x_{i-1} + x_i}{2}$$

$$t_{j-\frac{1}{2}} = \frac{t_{j-1} + t_j}{2}$$

$$\phi_{i\pm\frac{1}{2},j} = \frac{\phi_{i,j} + \phi_{i\pm 1,j}}{2}$$

$$\phi_{i,j\pm\frac{1}{2}} = \frac{\phi_{i,j} + \phi_{i,j\pm 1}}{2}$$

onde ϕ representa u ou v .

Segue o desenvolvimento do método.

Discretizando 4.50 em $(x_{i-\frac{1}{2}}, t_j)$ temos

$$\begin{aligned} v &= a(x)u_x \\ v_{i-\frac{1}{2},j} &= a_{i-\frac{1}{2}} \frac{U_{i,j} - U_{i-1,j}}{h} \\ &\Downarrow \\ \frac{V_{i-1,j} + V_{i,j}}{2} &= a_{i-\frac{1}{2}} \frac{U_{i,j} - U_{i-1,j}}{h} \\ &\Downarrow \\ \rho_i U_{i-1,j} - \rho_i U_{i,j} + V_{i-1,j} + V_{i,j} &= 0 \end{aligned} \tag{4.55}$$

onde $\rho_i = 2 \frac{a_{i-\frac{1}{2}}}{h}$.

Para 4.51 segue

$$\begin{aligned} v_x &= u_t - c(x)u \\ &\Downarrow \\ \frac{V_{i,j} - V_{i-1,j}}{h} &= \frac{U_{i-\frac{1}{2},j} - U_{i-\frac{1}{2},j-1}}{k} - c_{i-\frac{1}{2}} U_{i-\frac{1}{2},j} \\ &\Downarrow \end{aligned}$$

$$(\omega_i - \sigma)U_{i-1,j} + (\omega_i - \sigma)U_{i,j} - V_{i-1,j} + V_{i,j} = -\sigma(U_{i-1,j-1} + U_{i,j-1}) \tag{4.56}$$

onde $\sigma = \frac{h}{2k}$, $\omega_i = h \frac{c_{i-\frac{1}{2}}}{2}$.

Discretizando as condições auxiliares, temos

$$u_{i,1} = \psi(x) \quad (4.57)$$

$$\alpha_0 U_{1,j} + \alpha_3 V_{1,j} = \alpha_2(t_j) \quad (4.58)$$

$$\beta_0 U_{M+1,j} + \beta_3 V_{M+1,j} = \beta_2(t_j) \quad (4.59)$$

Consideremos agora uma malha em $[0, 1] \times [0, T]$ com M partições em $[0, 1]$ e N partições em $[0, T]$. A aplicação desse irá requerer, para cada estágio, a resolução de um sistema linear com $2M$ equações e o mesmo número de incógnitas. Podemos proceder a construção desse sistema da seguinte forma:

Fixando um estágio j , e variando i de 1 a $M + 1$, segue:

1. Para $i = 1$, por 4.58 temos $\alpha_0 u_{1,j} + \alpha_3 v_{1,j} = \alpha_2(t_j)$.

2. Para $2 \leq i \leq M$, por 4.56 temos

$$(\omega_i - \sigma)U_{i-1,j} + (\omega_i - \sigma)U_{i,j} - V_{i-1,j} + V_{i,j} = -\sigma(U_{i-1,j-1} + U_{i,j-1}).$$

3. Para $2 \leq i \leq M$, por 4.55 temos $\rho_i U_{i-1,j} - \rho_i U_{i,j} + V_{i-1,j} + V_{i,j} = 0$.

4. Para $i = M+1$, por 4.59 temos $\beta_0 u_{M+1,j} + \beta_3 v_{M+1,j} = \beta_2(t_j)$.

Assim sendo, teremos o seguinte sistema linear:

$$\left\{ \begin{array}{lcl} \alpha_0 U_{1,j} + \alpha_3 V_{1,j} & = & \alpha_2(t_j) \\ (\omega_2 - \sigma)U_{1,j} + (\omega_2 - \sigma)U_{2,j} - V_{1,j} + V_{2,j} & = & -\sigma(U_{1,j-1} + U_{2,j-1}) \\ (\omega_3 - \sigma)U_{2,j} + (\omega_3 - \sigma)U_{3,j} - V_{2,j} + V_{3,j} & = & -\sigma(U_{2,j-1} + U_{3,j-1}) \\ \vdots & \vdots & \vdots \\ (\omega_{M-1} - \sigma)U_{M-2,j} + (\omega_{M-1} - \sigma)U_{M-1,j} - V_{M-1,j} + V_{M,j} & = & -\sigma(U_{M-2,j-1} + U_{M-1,j-1}) \\ (\omega_M - \sigma)U_{M-1,j} + (\omega_M - \sigma)U_{M,j} - V_{M-1,j} + V_{M,j} & = & -\sigma(U_{M-1,j-1} + U_{M,j-1}) \\ \rho_2 U_{1,j} - \rho_2 U_{2,j} + V_{1,j} + V_{2,j} & = & 0 \\ \rho_3 U_{2,j} - \rho_3 U_{3,j} + V_{2,j} + V_{3,j} & = & 0 \\ \vdots & \vdots & \vdots \\ \rho_{M-1} U_{M-2,j} - \rho_{M-1} U_{M-1,j} + V_{M-1,j} + V_{M,j} & = & 0 \\ \rho_M U_{M-1,j} - \rho_M U_{M,j} + V_{M-1,j} + V_{M,j} & = & 0 \\ \beta_0 U_{M+1,j} + \beta_3 V_{M+1,j} & = & \beta_2(t_j) \end{array} \right.$$

Colocando-o em notação matricial, obteremos um sistema da forma

$$A_j U_j = b_j \quad (4.60)$$

onde A_j é dada por

$$\begin{bmatrix} \alpha_0 & 0 & \cdots & \cdots & \cdots & 0 & \alpha_3 & 0 & \cdots & \cdots & \cdots & 0 \\ \omega_2 - \sigma & \omega_2 - \sigma & \ddots & & & \vdots & -1 & 1 & \ddots & & & \vdots \\ 0 & \omega_3 - \sigma & \omega_3 - \sigma & \ddots & & \vdots & 0 & -1 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \omega_{M-1} - \sigma & \omega_{M-1} - \sigma & 0 & \vdots & & \ddots & -1 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & \omega_M - \sigma & \omega_M - \sigma & 0 & \cdots & \cdots & 0 & -1 & 1 \\ \rho_2 & -\rho_2 & 0 & \cdots & \cdots & 0 & 1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \rho_3 & -\rho_3 & \ddots & & \vdots & 0 & 1 & 1 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \rho_{M-1} & -\rho_{M-1} & 0 & \vdots & & \ddots & 1 & 1 & 0 \\ \vdots & & & \ddots & \rho_M & -\rho_M & \vdots & & & \ddots & 1 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & \beta_0 & 0 & \cdots & \cdots & \cdots & 0 & \beta_3 \end{bmatrix}$$

e U_j e b_j por

$$U_j = \begin{bmatrix} U_{1,j} \\ U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M,j} \\ U_{M+1,j} \\ V_{1,j} \\ V_{2,j} \\ V_{3,j} \\ \vdots \\ V_{M,j} \\ V_{M+1,j} \end{bmatrix} \quad \text{e} \quad b_j = \begin{bmatrix} \alpha_2(t_j) \\ -\sigma(U_{1,j-1} + U_{2,j-1}) \\ -\sigma(U_{2,j-1} + U_{3,j-1}) \\ \vdots \\ -\sigma(U_{M-2,j-1} + U_{M-1,j-1}) \\ -\sigma(U_{M-1,j-1} + U_{M,j-1}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \beta_2(t_j) \end{bmatrix}$$

Portanto, definida uma malha, teremos que resolver o sistema $A_j U_j = b_j$ para cada estágio j .

Algoritmo para o Método Box

1) Início

Entrada de dados. Entre com

$a(x)$ e $c(x)$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$\alpha_0, \alpha_1, \alpha_2(t), \beta_0, \beta_1$ e $\beta_2(t)$.

$f(x) \rightarrow$ Condição inicial.

2) Cálculo dos espaçamentos h, k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = \frac{h}{2k} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $i = 1$ a $M + 1$ faça

$x = (i - 1)h$

$U(i, 1) = f(x)$

Fim para

4) Cálculo das aproximações

$$A(1, 1) = \alpha_0$$

$$A(1, M + 2) = \alpha_3$$

$$A(2M, M + 1) = \beta_0$$

$$A(2M, 2M) = \beta_3$$

Para $j = 2$ a N faça

Para $i = 2$ a $M + 1$ faça

$$x1 = (i - 1)h$$

$$x = \frac{2x1 - h}{2}$$

$$\omega = h \frac{c(x)}{2}$$

$$\rho = 2 \frac{a(x)}{h}$$

$$A(i, i - 1) = \omega - \sigma$$

$$A(i, i) = \omega - \sigma$$

$$A(i, i + M) = -1$$

$$A(i, i + M + 1) = 1$$

$$A(i + M, i - 1) = \rho(i)$$

$$A(i + M + 1, i) = -\rho(i)$$

$$A(i + M, i + M) = 1$$

$$A(i + M, i + M + 1) = 1$$

Fim para

Para $i = 1$ a $2M$ faça

Se $(i = 1)$ então faça $b(i) = \alpha_2(t_j)$

senão se $(2 \leq i \leq M + 1)$ então faça $b(i) = -\sigma(U(i - 1, j - 1) + U(i, j - 1))$

senão se $(M + 2 \leq i \leq 2M - 1)$ então faça $b(i) = 0$

senão se $(i = 2M)$ então faça $b(i) = \beta_2(t_j)$

Fim para

Resolva o sistema linear $A_j U_j = b_j$.

Solução em $j = i \rightarrow U(1 \text{ a } M + 1, j) = U_j(1 \text{ a } M + 1)$

Fim para

5) Fim do algoritmo

Os algoritmos dos métodos, para equações parabólicas com coeficientes variáveis, que foram desenvolvidos, são específicos para as EDPs dos problemas que foram apresentados. Para outras equações, os algoritmos terão que ser modificados.

4.5 Métodos para a Equação do Calor em 2D

Considere o seguinte problema:

$$\left\{ \begin{array}{ll} u_t = a_1 u_{xx} + a_2 u_{yy} & a_1 > 0 \quad a_2 > 0 \\ u(x, y, 0) = f_1(x, y) & (x, y) \in [0, L] \times [0, L] \quad L > 0 \\ u(x, 0, t) = f_2(x, t) & (x, t) \in [0, L] \times \mathbb{R}_+ \\ u(x, L, t) = f_3(x, t) & (x, t) \in [0, L] \times \mathbb{R}_+ \\ u(0, y, t) = f_4(y, t) & (y, t) \in [0, L] \times \mathbb{R}_+ \\ u(L, y, t) = f_5(y, t) & (y, t) \in [0, L] \times \mathbb{R}_+ \end{array} \right. \quad (4.61)$$

A EDP desse problema é a Equação do Calor em duas dimensões com relação ao espaço, ou seja, possui duas variáveis de posição x e y .

Desenvolveremos 4 métodos de resolução para a equação parabólica em questão, definida em uma malha com os seguintes dados:

- M partições no intervalo $[0, L]$, ou seja, o mesmo número de partições, tanto para o domínio no eixo x como para no eixo y .
- N partições no intervalo $[0, T]$ no eixo t .

Nos métodos que serão apresentados, a partir de cada nível de tempo t teremos uma malha de onde obteremos aproximações da função incógnita no nível seguinte. A figura a seguir ilustra melhor essa idéia.

Ilustração:

Figura 4.12: Ilustração das malhas nos estágios

Observe a figura. Como veremos mais adiante, a partir de valores do primeiro nível, o qual é conhecido pela condição inicial $f_1(x, y)$, calcularemos as aproximações no segundo nível, e deste, para o terceiro, e assim por diante.

Acompanhe o desenvolvimento da teoria dos métodos:

Considere o seguinte problema:

$$\begin{cases} u_t = Au \\ u(x, y, t_0) = u_0 \end{cases} \quad (4.62)$$

onde A é um operador qualquer, por exemplo, $Au = u_{xx}$.

Temos que

$$u(x, y, t) = e^{(t-t_0)A}u_0 \quad (4.63)$$

é solução do problema 4.62, onde $e^{(t-t_0)A}$, chamado de **semigrupo**, é definido por

$$e^{(t-t_0)A} = [I + (t-t_0)A + (t-t_0)^2 A^2 + (t-t_0)^3 A^3 + \dots] \quad (4.64)$$

sendo válido para qualquer operador A . Para maiores detalhes sobre o que foi afirmado e sobre semigrupos, ver [12].

Indo agora para a EDP de interesse, isto é, $u_t = a_1 u_{xx} + a_2 u_{yy}$, e considerando o operador $L \equiv a_1 u_{xx} + a_2 u_{yy}$, temos que

$$u_t = Lu$$

Assim sendo, considere o seguinte problema:

$$\begin{cases} u_t = Lu \\ u(x, y, t_k) = u_k \end{cases} \quad (4.65)$$

Temos, analogamente ao problema 4.62, que

$$u(x, y, t) = e^{(t-t_k)L} u_k$$

é solução do problema 4.65.

Dessa forma, a solução em um determinado nível, ou estágio, t_{k+1} , será dada por:

$$u(x, y, t_{k+1}) = e^{(t_{k+1}-t_k)L} u_k$$

$$\Downarrow$$

$$u(x, y, t_{k+1}) = e^{kL} u_k \quad (4.66)$$

Temos que 4.66 fornece a solução exata da EDP considerada, em um determinado nível k , a partir do nível anterior. Os métodos que serão apresentados são obtidos por truncamento da expansão de $e^{kL}u_k$, conforme ao que foi mostrado em 4.64.

Discretizando as variáveis de posição x e y em 4.66, temos

$$U_{i,j}^{k+1} = e^{kL}U_{i,j}^k \quad (4.67)$$

Segue o desenvolvimento. Considere a Equação do Calor com $a_1 = a_2 = 1$

$$u_t = u_{xx} + u_{yy} \quad (4.68)$$

Assim, o operador L fica sendo

$$L \equiv u_{xx} + u_{yy} = D_1^2 + D_2^2 \quad (4.69)$$

onde $D_1 = u_x$ e $D_2 = u_y$.

Com esses dados, a equação 4.68 torna-se:

$$U^{k+1} = e^{(kD_1^2)}e^{(kD_2^2)}U^k \quad (4.70)$$

onde $U^k = U_{i,j}^k$.

Como

$$D_1^2 = \frac{1}{h^2} \left(\delta_x^2 - \frac{1}{12}\delta_x^4 + \frac{1}{90}\delta_x^6 \dots \right) \quad (4.71)$$

$$D_1^2 = \frac{1}{h^2} \left(\delta_y^2 - \frac{1}{12} \delta_y^4 + \frac{1}{90} \delta_y^6 \dots \right) \quad (4.72)$$

Substituindo 4.71 e 4.72 em 4.70, segue

$$U^{k+1} = \left[1 + \sigma \delta_x^2 + \frac{\sigma}{2} \left(\sigma - \frac{1}{6} \right) \delta_x^4 \dots \right] \left[1 + \sigma \delta_y^2 + \frac{\sigma}{2} \left(\sigma - \frac{1}{6} \right) \delta_y^4 \dots \right] U^k \quad (4.73)$$

onde $\sigma = \frac{k}{h^2}$.

Vários são os métodos que podem serem obtidos pelo truncamento das séries na equação 4.73. Os métodos explícitos de 5 pontos e 9 pontos, a serem apresentados nas duas seções a seguir, são obtidos por truncamentos adequados das referidas séries.

4.5.1 Método Explícito - 5 pontos

Como já foi mencionado anteriormente, esse método é obtido pelo truncamento das duas séries em 4.73, e é dada por

$$U^{k+1} = [1 + \sigma(\delta_x^2 + \delta_y^2)]U^k + O(k^2 + kh^2) \quad (4.74)$$

e pode ser representada pela seguinte molécula computacional:

Figura 4.13: Molécula computacional do método explícito - 5 pontos

Acompanhe o desenvolvimento do método.

$$U_{i,j}^{k+1} = [1 + \sigma(\delta_x^2 + \delta_y^2)]U_{i,j}^k$$

$$\begin{aligned}
& \Downarrow \\
U_{i,j}^{k+1} &= U_{i,j}^k + \sigma \delta_x^2 U_{i,j}^k + \sigma \delta_y^2 U_{i,j}^k \\
& \Downarrow \\
U_{i,j}^{k+1} &= U_{i,j}^k + \sigma U_{i-1,j}^k - 2\sigma U_{i,j}^k + \sigma U_{i+1,j}^k + \sigma U_{i,j-1}^k - 2\sigma U_{i,j}^k + \sigma U_{i,j+1}^k \\
& \Downarrow \\
U_{i,j}^{k+1} &= \sigma(U_{i-1,j}^k + U_{i+1,j}^k) + (1 - 4\sigma)U_{i,j}^k + \sigma(U_{i,j-1}^k + U_{i,j+1}^k) \quad (4.75)
\end{aligned}$$

Por 4.74 observamos que para calcular as aproximações em um dado nível, precisaremos, para cada ponto da malha nesse nível, de 5 aproximações do nível anterior.

Pela ordem do ETL em 4.74, vemos que o método é de ordem 2. Agora, a análise da estabilidade pelo critério de Von Neumann é feita, analogamente ao realizado para a Equação do Calor Unidimensional, fixando uma posição (x, y) , e a partir daí analisar o efeito de amplificação ou amortecimento de um modo de Fourier, que para 3 variáveis é dada por

$$U_{i,j}^k = e^{jk\alpha} e^{ih\beta I} e^{ih\gamma I} \quad (4.76)$$

onde $I = \sqrt{-1}$, e β e γ são números arbitrários e $\alpha \equiv \alpha(\beta, \gamma)$ é em geral complexo.

O erro não crescerá com o tempo se

$$|e^{k\alpha}| < 1$$

Como foi feito para a Equação do Calor Unidimensional, se substituirmos 4.76 em 4.75, chegaremos a

$$e^{k\alpha} = 1 - 4\sigma \left(\sin^2 \frac{\beta h}{2} + \sin^2 \frac{\gamma h}{2} \right) \quad (4.77)$$

Para que haja estabilidade devemos ter $|e^{k\alpha}| < 1$

$$-1 \leq 1 - 4\sigma \left(\operatorname{sen}^2 \frac{\beta h}{2} + \operatorname{sen}^2 \frac{\gamma h}{2} \right) \leq 1$$

$$\Downarrow$$

$$\sigma \leq \frac{1}{2 \left(\operatorname{sen}^2 \frac{\beta h}{2} + \operatorname{sen}^2 \frac{\gamma h}{2} \right)} \quad (4.78)$$

Temos aqui uma restrição. Observe que o maior valor que o segundo membro de 4.79 pode assumir é $\frac{1}{4}$, portanto, para que haja estabilidade, deveremos ter a condição

$$\sigma \leq \frac{1}{4} \quad (4.79)$$

satisfeita.

Esse não é muito usado para resolver problemas de valor de fronteira e inicial em duas ou mais variáveis porque possuem estabilidade muito restritiva.

Algoritmo para a Equação do Calor em 2D - Método Explícito - 5 pontos

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f_1(x, y)$, $f_2(x, t)$, $f_3(x, t)$, $f_4(y, t)$ e $f_5(y, t)$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, j, 1) = f_1(x, y)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1, p) = f_2(x, t)$$

$$U(i, M + 1, p) = f_3(x, t)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)h$$

$$U(1, j, p) = f_4(x, y)$$

$$U(M + 1, j, p) = f_5(x, y)$$

Fim para

Fim para

4) Cálculo das aproximações

Para $p = 1$ a N faça

Para $i = 2$ a M faça

Para $j = 2$ a M faça

$$aux1 = U(i - 1, j, p) + U(i + 1, j, p)$$

$$aux2 = U(i, j, p)$$

$$aux3 = U(i, j - 1, p) + U(i, j + 1, p)$$

$$U(i, j, p + 1) = \sigma aux1 + (1 - 4\sigma)aux2 + \sigma aux3$$

Fim para

Fim Para

Fim Para

Faça $p = 1$ a $N + 1$ e imprima, para cada p , $U(1 \leq i \leq M + 1, 1 \leq j \leq M + 1, p)$

5) Fim do algoritmo

4.5.2 Método Explícito - 9 pontos

Esse é um outro método obtido pelo truncamento das séries em 4.73, o qual é dado por

$$U^{k+1} = (1 + \sigma\delta_x^2)(1 + \sigma\delta_y^2)U^k + O(k^2 + kh^2) \quad (4.80)$$

e pode ser representada pela seguinte molécula computacional:

Figura 4.14: Molécula computacional do método explícito - 9 pontos

Acompanhe o desenvolvimento do método.

$$\begin{aligned}
 U_{i,j}^{k+1} &= (1 + \sigma\delta_x^2)(1 + \sigma\delta_y^2)U_{i,j}^k \\
 &\Downarrow \\
 U_{i,j}^{k+1} &= (1 + \sigma\delta_y^2 + \sigma\delta_x^2 + \sigma^2\delta_x^2\delta_y^2)U_{i,j}^k \\
 &\Downarrow \\
 U_{i,j}^{k+1} &= U_{i,j}^k + \sigma\delta_y^2U_{i,j}^k + \sigma\delta_x^2U_{i,j}^k + \sigma^2\delta_x^2\delta_y^2U_{i,j}^k \\
 &\Downarrow \\
 U_{i,j}^{k+1} &= U_{i,j}^k + \sigma(U_{i,j-1}^k - 2U_{i,j}^k + U_{i,j+1}^k) + \sigma(U_{i-1,j}^k - 2U_{i,j}^k + U_{i+1,j}^k) + \sigma^2\delta_x^2(U_{i,j-1}^k - 2U_{i,j}^k + U_{i,j+1}^k) \\
 &\Downarrow
 \end{aligned}$$

$$\begin{aligned}
U_{i,j}^{k+1} = & \sigma^2(U_{i-1,j-1}^k + U_{i+1,j-1}^k + U_{i-1,j+1}^k + U_{i+1,j+1}^k) \\
& + \sigma(1 + \sigma)(U_{i-1,j}^k + U_{i+1,j}^k) + \sigma(1 - 2\sigma)(U_{i,j-1}^k + U_{i,j+1}^k) \\
& + (1 - 4\sigma - 2\sigma^2)U_{i,j}^k
\end{aligned} \tag{4.81}$$

Por 4.81 observamos que para calcular as aproximações em um dado nível, precisaremos, para cada ponto da malha nesse nível, de 9 aproximações do nível anterior.

A ordem desse método é a mesma do anterior, da mesma forma que a condição de estabilidade.

Algoritmo para a Equação do Calor em 2D - Método Explícito - 9 pontos

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f_1(x, y)$, $f_2(x, t)$, $f_3(x, t)$, $f_4(y, t)$ e $f_5(y, t)$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, j, 1) = f_1(x, y)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1, p) = f_2(x, t)$$

$$U(i, M + 1, p) = f_3(x, t)$$

Fim para

Fim para

Para $p = 1$ a N faça

$$t = (p - 1)k$$

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)h$$

$$U(1, j, p) = f_4(x, y)$$

$$U(M + 1, j, p) = f_5(x, y)$$

Fim para

Fim para

4) Cálculo das aproximações

Para $p = 2$ a N faça

Para $i = 2$ a M faça

Para $j = 2$ a M faça

$$aux1 = U(i - 1, j - 1, p) + U(i + 1, j - 1, p)$$

$$aux2 = U(i - 1, j + 1, p) + U(i + 1, j + 1, p)$$

$$aux3 = U(i - 1, j, p) + U(i + 1, j, p)$$

$$aux4 = U(i, j - 1, p) + U(i, j + 1, p)$$

$$aux5 = U(i, j, p)$$

$$aux6 = \sigma^2(aux1 + aux2)$$

$$aux7 = \sigma(1 + \sigma)aux3$$

$$aux8 = \sigma(1 - 2\sigma)aux4$$

$$aux9 = (1 - 4\sigma - 2\sigma^2)aux5$$

$$U(i, j, p + 1) = aux6 + aux7 + aux8 + aux9$$

Fim para

Fim Para

Fim Para

Faça $p = 1$ a $N + 1$ e imprima, para cada p , $U(1 \leq i \leq M + 1, 1 \leq j \leq M + 1, p)$

5) Fim do algoritmo

4.5.3 Método das Direções Alternadas - ADI

O método das direções alternadas são métodos de dois passos, envolvendo a solução de sistemas tridiagonais ao longo de linhas paralelas aos eixos coordenados x e y . A idéia básica é a seguinte:

- Resolve-se primeiramente o sistema envolvido no método na direção x , obtendo com isso uma aproximação intermediária.
- Usando essa aproximação intermediária, resolvemos agora o sistema na direção y . A resolução nessa direção funciona como um corretor da primeira aproximação.

Acompanhe o desenvolvimento do método a seguir.

De 4.70, temos

$$\begin{aligned}
 U^{k+1} &= e^{k(D_1^2 + D_2^2)} U^k \\
 &\Downarrow \\
 U^{k+1} &= e^{\frac{k}{2}(D_1^2 + D_2^2) + \frac{k}{2}(D_1^2 + D_2^2)} U^k \\
 &\Downarrow \\
 e^{-\frac{k}{2}(D_1^2 + D_2^2)} U^{k+1} &= e^{\frac{k}{2}(D_1^2 + D_2^2)} U^k \\
 &\Downarrow \\
 e^{-\frac{k}{2}D_1^2} e^{-\frac{k}{2}D_2^2} U^{k+1} &= e^{\frac{k}{2}D_1^2} e^{\frac{k}{2}D_2^2} U^k
 \end{aligned} \tag{4.82}$$

Expandindo e truncando série em 4.82, obteremos a equação:

$$\left(1 - \frac{\sigma}{2}\delta_x^2\right) \left(1 - \frac{\sigma}{2}\delta_y^2\right) U^{k+1} = \left(1 + \frac{\sigma}{2}\delta_x^2\right) \left(1 + \frac{\sigma}{2}\delta_y^2\right) U^k + O(k^2 + kh^2) \tag{4.83}$$

onde $\sigma = \frac{k}{h^2}$.

Podemos decompor 4.83, introduzindo com isso um passo intermediário, e resolver primeiramente na direção x , obtendo assim as aproximações nesse passo intermediário, e a seguir na direção y . Para entender melhor isso, acompanhe o desenvolvimento a seguir:

Fazendo

$$U^{k+1*} = \left(1 - \frac{\sigma}{2}\delta_y^2\right) U^{k+1} \quad (4.84)$$

e substituindo 4.84 em 4.83, segue

$$\left(1 - \frac{\sigma}{2}\delta_x^2\right) U^{k+1*} = \left(1 + \frac{\sigma}{2}\delta_x^2\right) \left(1 + \frac{\sigma}{2}\delta_y^2\right) U^k \quad (4.85)$$

$$\left(1 - \frac{\sigma}{2}\delta_y^2\right) U^{k+1} = U^{k+1*} \quad (4.86)$$

O esquema 4.85 e 4.86 ilustra a idéia do método. Primeiro resolvemos 4.85 na direção x , obtendo U^{k+1*} , e a seguir usamos esses valores para calcular, mediante a resolução de 4.86, as aproximações em U^{k+1} .

Segue o desenvolvimento do sistema gerado pelo método:

Por 4.85, temos

$$\begin{aligned} \left(1 - \frac{\sigma}{2}\delta_x^2\right) U^{k+1*} &= \left(1 + \frac{\sigma}{2}\delta_x^2\right) \left(1 + \frac{\sigma}{2}\delta_y^2\right) U^k \\ &\Downarrow \\ -\frac{\sigma}{2}U_{i-1,j}^{k+1*} + (1 + \sigma)U_{i,j}^{k+1*} + \frac{\sigma}{2}U_{i+1,j}^{k+1*} &= \\ \frac{\sigma^2}{4}U_{i-1,j-1}^k + \frac{\sigma(1-\sigma)}{2}U_{i,j-1}^k + \frac{\sigma^2}{4}U_{i+1,j-1}^k & \\ + \frac{\sigma(1-\sigma)}{2}U_{i-1,j}^k + (1-\sigma)^2U_{i,j}^k + \frac{\sigma(1-\sigma)}{2}U_{i+1,j}^k & \\ + \frac{\sigma^2}{4}U_{i-1,j+1}^k + \frac{\sigma(1-\sigma)}{2}U_{i,j+1}^k + \frac{\sigma^2}{4}U_{i+1,j+1}^k & \end{aligned} \quad (4.87)$$

Colocando em notação matricial, teremos um sistema da forma

$$AU_j^{k+1*} = BU_{j-1}^k + CU_j^k + BU_{j+1}^k + x^k \quad j = 2, 3, \dots, M \quad (4.88)$$

onde

$$A = \begin{bmatrix} 1+\sigma & -\frac{\sigma}{2} & 0 & \dots & 0 \\ -\frac{\sigma}{2} & 1+\sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1+\sigma & -\frac{\sigma}{2} \\ 0 & \dots & 0 & -\frac{\sigma}{2} & 1+\sigma \end{bmatrix} \quad U_j^{k+1*} = \begin{bmatrix} U_{2,j}^{k+1*} \\ U_{3,j}^{k+1*} \\ \vdots \\ U_{M-1,j}^{k+1*} \\ U_{M,j}^{k+1*} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\sigma-\sigma^2}{2} & \frac{\sigma^4}{4} & 0 & \dots & 0 \\ \frac{\sigma^4}{4} & \frac{\sigma-\sigma^2}{2} & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \frac{\sigma-\sigma^2}{2} & \frac{\sigma^4}{4} \\ 0 & \dots & 0 & \frac{\sigma^4}{4} & \frac{\sigma-\sigma^2}{2} \end{bmatrix} \quad U_{j-1}^{k+1*} = \begin{bmatrix} U_{2,j-1}^{k+1*} \\ U_{3,j-1}^{k+1*} \\ \vdots \\ U_{M-1,j-1}^{k+1*} \\ U_{M,j-1}^{k+1*} \end{bmatrix}$$

$$C = \begin{bmatrix} (1-\sigma)^2 & \frac{\sigma-\sigma^2}{2} & 0 & \dots & 0 \\ \frac{\sigma-\sigma^2}{2} & (1-\sigma)^2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & (1-\sigma)^2 & \frac{\sigma-\sigma^2}{2} \\ 0 & \dots & 0 & \frac{\sigma-\sigma^2}{2} & (1-\sigma)^2 \end{bmatrix} \quad U_{j+1}^{k+1*} = \begin{bmatrix} U_{2,j+1}^{k+1*} \\ U_{3,j+1}^{k+1*} \\ \vdots \\ U_{M-1,j+1}^{k+1*} \\ U_{M,j+1}^{k+1*} \end{bmatrix}$$

$$x^k = \begin{bmatrix} \frac{\sigma^4}{4} U_{1,j-1}^{k+1*} + \frac{\sigma-\sigma^2}{2} U_{1,j}^{k+1*} + \frac{\sigma^4}{4} U_{1,j+1}^{k+1*} + \frac{\sigma}{2} U_{1,j}^{k+1} \\ 0 \\ \vdots \\ 0 \\ \frac{\sigma^4}{4} U_{M+1,j-1}^{k+1*} + \frac{\sigma-\sigma^2}{2} U_{M+1,j}^{k+1*} + \frac{\sigma^4}{4} U_{M+1,j+1}^{k+1*} + \frac{\sigma}{2} U_{M+1,j}^{k+1} \end{bmatrix}$$

Resolvido o sistema gerado por 4.88, de posse de U^{k+1*} , passamos a resolução do sistema 4.86, correndo agora na direção y , o qual terá a seguinte notação matricial

$$AU_i^{k+1} = U_i^{k+1*} \quad i = 2, 3, \dots, M \quad (4.89)$$

onde

$$A = \begin{bmatrix} 1 + \sigma & -\frac{\sigma}{2} & 0 & \dots & 0 \\ -\frac{\sigma}{2} & 1 + \sigma & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 + \sigma & -\frac{\sigma}{2} \\ 0 & \dots & 0 & -\frac{\sigma}{2} & 1 + \sigma \end{bmatrix} \quad U_i^{k+1*} = \begin{bmatrix} U_{i,2}^{k+1*} \\ U_{i,3}^{k+1*} \\ \vdots \\ U_{i,M-1}^{k+1*} \\ U_{i,M}^{k+1*} \end{bmatrix}$$

$$U_j^{k+1*} = \begin{bmatrix} U_{i,2}^{k+1*} + \frac{\sigma}{2} U_{i,1}^{k+1} \\ U_{i,3}^{k+1*} \\ \vdots \\ U_{i,M-1}^{k+1*} \\ U_{i,M}^{k+1*} + \frac{\sigma}{2} U_{i,M+1}^{k+1} \end{bmatrix}$$

Pelo que foi desenvolvido, nota-se que o esquema de aplicação do método é simples. Repetindo, primeiro varremos a malha na direção x no estágio k , calculando as aproximações em um nível intermediário $k + 1^*$, e a seguir calculamos as aproximações no nível $k + 1$, a partir de $k + 1^*$, varrendo na direção y .

Algoritmo para a Equação do Calor em 2D - Método das Direções Alternadas

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f_1(x, y)$, $f_2(x, t)$, $f_3(x, t)$, $f_4(y, t)$ e $f_5(y, t)$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \begin{cases} h = \frac{L}{M} & k = \frac{T}{N} & \sigma = a \frac{k}{h^2} \end{cases}$

3) Cálculo da condição inicial e de fronteira

Para $j = 1$ a $M + 1$ faça

$y = (j - 1)k$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, j, 1) = f_1(x, y)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1, p) = f_2(x, t)$$

$$U(i, M + 1, p) = f_3(x, t)$$

Fim para

Fim para

Para $p = 1$ a N faça

$$t = (p - 1)k$$

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)h$$

$$U(1, j, p) = f_4(x, y)$$

$$U(M + 1, j, p) = f_5(x, y)$$

Fim para

Fim para

4) Geração da matriz dos coeficientes

Para $i = 1$ a $M - 1$ faça

$$A(i, i) = 1 + \sigma$$

$$B(i, i) = \frac{\sigma - \sigma^2}{2}$$

$$C(i, i) = (1 - \sigma)^2$$

Fim para

Para $i = 1$ a $M - 2$ faça

$$A(i + 1, i) = -\frac{\sigma}{2}$$

$$A(i, i + 1) = A(i + 1, i)$$

$$B(i + 1, i) = \frac{\sigma^2}{4}$$

$$B(i, i + 1) = B(i + 1, i)$$

$$C(i+1, i) = \frac{\sigma - \sigma^2}{2}$$

$$C(i, i+1) = C(i+1, i)$$

Fim para

5) Cálculo das aproximações

Para $p = 1$ a N faça

Para $j = 2$ a M faça

Resolva o sistema $AU_j^{k+1*} = BU_{j-1}^k + CU_j^k + BU_{j+1}^k + x^k$ em 4.88

Fim para

Para $i = 2$ a M faça

Resolva o sistema $AU_i^{k+1} = U_i^{k+1*}$ em 4.89

Fim Para

Fim Para

Faça $p = 1$ a $N + 1$ e imprima, para cada p , $U(1 \text{ a } M + 1, 1 \text{ a } M + 1, p)$

6) Fim do algoritmo

4.5.4 Método Localmente Unidimensional

Considere a Equação do Calor em 2D

$$u_t = u_{xx} + u_{yy} \quad (4.90)$$

Podemos reescrever 4.90 da seguinte maneira

$$\frac{1}{2}u_t = u_{xx} \quad \frac{1}{2}u_t = u_{yy} \quad (4.91)$$

Com isso dividimos o problema original em duas equações, que juntas representam equação original. A idéia aqui consiste em resolver um problema bidimensional, que é a equação 4.90, mediante dois problemas localmente unidimensionais que localmente representam o problema principal, ambas em 4.91.

As discretizações explícitas mais simples das duas equações em 4.91 são dadas por:

$$U_{i,j}^{k+\frac{1}{2}} = (1 + \sigma\delta_y^2)U_{i,j}^k \quad (4.92)$$

$$U_{i,j}^{k+1} = (1 + \sigma\delta_x^2)U_{i,j}^{k+\frac{1}{2}} \quad (4.93)$$

De 4.92 temos

$$\begin{aligned} U_{i,j}^{k+\frac{1}{2}} &= (1 + \sigma\delta_y^2)U_{i,j}^k \\ &\Downarrow \\ U_{i,j}^{k+\frac{1}{2}} &= U_{i,j}^k + \sigma\delta_y^2 U_{i,j}^k \\ &\Downarrow \\ U_{i,j}^{k+\frac{1}{2}} &= U_{i,j}^k + \sigma(U_{i,j-1}^k - 2U_{i,j}^k + U_{i,j+1}^k) \\ &\Downarrow \\ U_{i,j}^{k+\frac{1}{2}} &= \sigma U_{i,j-1}^k + (1 - 2\sigma)U_{i,j}^k + \sigma U_{i,j+1}^k \end{aligned} \quad (4.94)$$

De 4.93 temos

$$\begin{aligned} U_{i,j}^{k+1} &= (1 + \sigma\delta_x^2)U_{i,j}^{k+\frac{1}{2}} \\ &\Downarrow \\ U_{i,j}^{k+1} &= U_{i,j}^{k+\frac{1}{2}} + \sigma\delta_x^2 U_{i,j}^{k+\frac{1}{2}} \\ &\Downarrow \\ U_{i,j}^{k+1} &= U_{i,j}^{k+\frac{1}{2}} + \sigma(U_{i-1,j}^{k+\frac{1}{2}} - 2U_{i,j}^{k+\frac{1}{2}} + U_{i+1,j}^{k+\frac{1}{2}}) \\ &\Downarrow \\ U_{i,j}^{k+1} &= \sigma U_{i-1,j}^{k+\frac{1}{2}} + (1 - 2\sigma)U_{i,j}^{k+\frac{1}{2}} + \sigma U_{i+1,j}^{k+\frac{1}{2}} \end{aligned} \quad (4.95)$$

A idéia desse método é a seguinte:

Dado um estágio, onde a partir do qual calcularemos aproximações no estágio posterior, calculamos

1. as aproximações na direção y usando 4.94, e a seguir
2. calculamos, usando as aproximações obtidas no item anterior, as aproximações no estágio de interesse.

Observe a ilustração desse esquema abaixo:

Figura 4.15: Ilustração gráfica do método localmente unidimensional

Primeiro calculamos, correndo na direção y , aproximações em um nível intermediário $k+1^*$, e depois, disso finalizamos o processo correndo na direção x , para obter o resultado em $k+1$ utilizando as aproximações desse nível intermediário.

O ETL e o critério de estabilidade desse método é o mesmo dos métodos explícito de 5 e 9 pontos. A vantagem do método em questão em relação a esses está na economia dos cálculos.

Algoritmo para a Equação do Calor em 2D - Método Localmente Unidimensional

- 1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} , com $a > 0$.

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f_1(x, y)$, $f_2(x, t)$, $f_3(x, t)$, $f_4(y, t)$ e $f_5(y, t)$.

2) Cálculo dos espaçamentos h , k e de sigma $\Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \quad k = \frac{T}{N} \quad \sigma = a \frac{k}{h^2} \end{array} \right.$

3) Cálculo da condição inicial e de fronteira

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, j, 1) = f_1(x, y)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$U(i, 1, p) = f_2(x, t)$$

$$U(i, M + 1, p) = f_3(x, t)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - \frac{1}{2})k$$

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$UI(i, 1, p) = f_2(x, t)$$

$$UI(i, M + 1, p) = f_3(x, t)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - 1)k$$

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)h$$

$$U(1, j, p) = f_4(x, y)$$

$$U(M + 1, j, p) = f_5(x, y)$$

Fim para

Fim para

Para $p = 1$ a $N + 1$ faça

$$t = (p - \frac{1}{2})k$$

Para $j = 1$ a $M + 1$ faça

$$y = (j - 1)h$$

$$UI(1, j, p) = f_4(x, y)$$

$$UI(M + 1, j, p) = f_5(x, y)$$

Fim para

Fim para

4) Cálculo das aproximações

Para $p = 1$ a N faça

$$A(1 \text{ a } M + 1, 1 \text{ a } M + 1, p) = U(1 \text{ a } M + 1, 1 \text{ a } M + 1, p)$$

$$B(1 \text{ a } M + 1, 1 \text{ a } M + 1, p) = UI(1 \text{ a } M + 1, 1 \text{ a } M + 1, p)$$

Para $i = 2$ a M faça

Para $j = 2$ a M faça

$$B(i, j, p) = \sigma A(i, j - 1, p) + (1 - 2\sigma)A(i, j, p) + \sigma A(i, j + 1, p)$$

Fim para

Para $j = 2$ a M faça

$$U(i, j, p + 1) = \sigma B(i - 1, j, p) + (1 - 2\sigma)B(i, j, p) + \sigma B(i + 1, j, p)$$

Fim para

Delete A e B

Fim Para

Fim Para

Faça $p = 1$ a $N + 1$ e imprima, para cada p , $U(1 \leq i \leq M + 1, 1 \leq j \leq M + 1, p)$

5) Fim do algoritmo

Capítulo 5

Equações Hiperbólicas

5.1 Introdução

Assim como as equações parabólicas, as equações hiperbólicas modelam os chamados problemas de propagação ou evolução. Recebem essa denominação por modelarem fenômenos físicos cuja entidade física em estudo, velocidade ou amplitude por exemplo, variam no decorrer do tempo. Temos abaixo alguns exemplos de fenômenos físicos modelados por equações hiperbólicas.

Exemplos:

(a) O sistema de primeira ordem

$$\begin{cases} (\rho u)_x + \rho_t = 0 \\ uu_x + u_t = -\frac{1}{\rho}p_x \\ up_x + p_t = -\gamma pu_x \end{cases}$$

governa o fluxo adiabático unidimensional de um gás ideal com velocidade u , densidade ρ e pressão p .

(b) A voltagem v e corrente elétrica i em uma linha de transmissão satisfaz o sistema de primeira ordem

$$\begin{cases} i_x + Cv_t = -Gv \\ v_x + Li_t = -Ri \end{cases}$$

onde R , L , C e G denotam, respectivamente, a resistência, a indutância, a capacitância e a vazão de corrente elétrica, todas por unidade de comprimento.

5.2 Características

Quando se trabalha com equações hiperbólicas, as **características** desempenham um papel importante, tanto para resolução analítica quanto para resolução numérica. Para compreender o significado, conceito de características, acompanhe o desenvolvimento a seguir:

Considere o seguinte sistema de equações hiperbólicas quase-lineares:

$$\sum_{i=1}^n (a_{ji}u_x^i + b_{ji}u_y^i) + d_j = 0, \quad j = 1, 2, \dots, n \quad (5.1)$$

com n equações em n incógnitas u^i , $i = 1, \dots, n$, onde os coeficientes a_{ji} , b_{ji} e d_j são funções de x , y e u^i .

Colocando o sistema 5.1 na forma matricial, temos:

$$Au_x + Bu_y + d = 0 \quad (5.2)$$

onde A e B são matrizes $n \times n$, e d e u são vetores coluna.

Seja T , T inversível, uma matriz com elementos t_{ji} que podem depender de x , y e u^i , mas não das derivadas de u^i . Aplicando essa transformação em 5.2, segue

$$TAu_x + TBU_y + Td = 0 \quad (5.3)$$

Esse novo sistema obtido, isto é, 5.3, é equivalente ao sistema 5.2, no sentido de que toda solução de 5.3 também é solução 5.2.

A transformação linear feita em 5.2 é usada para o desenvolvimento de uma *forma canônica* para o sistema. Uma das maneiras convenientes de obter uma forma canônica é fazendo

$$TA = ETB \quad (5.4)$$

onde E é a seguinte matriz diagonal:

$$E = \begin{bmatrix} e_1 & 0 & \dots & \dots & 0 \\ 0 & e_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & e_{n-1} & 0 \\ 0 & \dots & \dots & 0 & e_n \end{bmatrix}. \quad (5.5)$$

Assumindo 5.4, podemos reescrever o sistema de equações 5.3 da seguinte forma

$$ETBu_x + TBu_y + Td = 0 \quad (5.6)$$

Vamos analisar a forma das equações do sistema 5.6 colocando $TB = A^* = [a_{ji}]$, $Td = d^* = [d_j^*]$. Fazendo as respectivas substituições no sistema de equações 5.6, temos

$$EA^*u_x + A^*u_y + d^* = 0. \quad (5.7)$$

Assim sendo, uma equação j qualquer do sistema 5.7 tem a seguinte forma :

$$\sum_{i=1}^n a_{ji}^*(e_j u_x^i + u_y^i) + d_j^* = 0 \quad (5.8)$$

Ilustração:

Figura 5.1: Notação usada no desenvolvimento das características

Agora, se $\alpha i + \beta j$ é o vetor unitário para o qual

$$e_j = \frac{\alpha}{\beta} = \cotg\theta,$$

podemos escrever, conforme a figura acima,

$$\begin{aligned} e_j u_x^i + u_y^i &= \frac{\alpha}{\beta} u_x^i + u_y^i \\ &\Downarrow \\ e_j u_x^i + u_y^i &= \frac{1}{\beta} (u_x^i \alpha + u_y^i \beta) \\ &\Downarrow \\ e_j u_x^i + u_y^i &= \frac{1}{\beta} \underbrace{(u_x^i \cos\theta + u_y^i \sin\theta)}_I \end{aligned}$$

onde temos em I, exceto pelo fator $\frac{1}{\beta}$, a derivada direcional na direção definida pelo vetor $\alpha i + \beta j$. Observe que essa derivada direcional depende de e_j . Assim sendo, toda equação do sistema transformado *possui diferenciação em uma direção somente*.

O cálculo da matriz diagonal E é realizado a partir da definição $TA = TEB$. Esta relação é equivalente ao sistema de equações

$$\sum_{k=1}^n t_{jk} a_{ki} = \sum_{k=1}^n e_j t_{jk} b_{ki} \quad i = 1, 2, \dots, n$$

ou

$$\sum_{k=1}^n (a_{ki} - e_j b_{ki}) t_{jk} = 0$$

o qual representa, por sua vez, um sistema de equações homogêneas onde a incógnita é dada por t_{jk} , $k = 1, 2, \dots, n$. Para uma solução não-trivial existir, uma condição necessária e suficiente é que

$$\det(A - e_j B) = 0. \quad (5.10)$$

Ou seja, e_j deve ser um autovalor de $A - \lambda B$. Note que em 5.10 as matrizes A e B podem depender de x , y e u^i e portanto e_j pode depender dessas variáveis.

Vamos considerar o caso no qual

$$\det(A - \lambda B) = 0$$

tem n raízes reais e distintas, nesse caso escolhemos n raízes para e_j , $j = 1, 2, \dots, n$, e determinamos a matriz T a partir das equações em 5.9. A equação 5.7 é chamada *forma normal*. A direção $\alpha_{ki} + \beta_{kj}$, para o qual $e_k = \frac{\alpha_k}{\beta_k}$, é chamada k -ésima direção característica. As n equações diferenciais

$$\cot g \theta = \frac{dx}{dy} = e_k \quad (5.11)$$

são chamadas de características do sistema.

Com isso temos então definido o conceito de característica. O método das características, a ser apresentado na próxima seção, baseia-se no cálculo das aproximações de uma equação hiperbólica ao longo das características.

5.3 Método das Características

O método das características é uma técnica muito importante na análise e solução de equações hiperbólicas de primeira e segunda ordem. Desenvolveremos esse método para uma equação hiperbólica de primeira ordem.

Considere o seguinte problema:

$$\begin{cases} a(x, t)u_t + b(x, t)u_x = c(x, t) & (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = f(x) & x \in \mathbb{R} \end{cases} \quad (5.12)$$

Seja $u(x, t)$ a solução da equação 5.12 e $(x(s), t(s))$ curvas definidas parametricamente por s em $\mathbb{R} \times \mathbb{R}_+$. Sejam essas curvas as **características** da equação hiperbólica considerada.

Ilustração das características:

Figura 5.2: Ilustração das características para $a(x, t)$ e $b(x, t)$ constantes

Ao longo das características podemos calcular $u(x, t)$ por $u(x(s), t(s))$, isto é, em relação ao parâmetro s . Assim sendo, podemos também calcular as derivadas da referida função, ao longo das características, em relação ao parâmetro s . Então, pela regra da cadeia temos:

$$\frac{du}{ds} = \frac{du}{dx} \frac{dx}{ds} + \frac{du}{dt} \frac{dt}{ds} \Rightarrow \frac{du}{ds} = \frac{dx}{ds} u_x + \frac{dt}{ds} u_t \quad (5.13)$$

Comparando 5.13 com 5.12 temos:

$$\frac{dx}{ds} = b(x, y) \quad \frac{dt}{ds} = a(x, y) \quad \frac{du}{ds} = c(x, y) \quad (5.14)$$

Com 5.14, é possível obter aproximações da solução ao longo das características, fazendo:

$$\frac{x_Q - x_P}{\delta_s} = b_P \quad \text{onde } b_P = b(x_P, t_P)$$

$$\frac{t_Q - t_P}{\delta_s} = a_P \quad \text{onde } a_P = a(x_P, t_P)$$

$$\frac{u_Q - u_P}{\delta_s} = c_P \quad \text{onde } c_P = c(x_P, t_P)$$

Dessa forma, fixando t_Q , um nível de tempo onde se deseja calcular uma aproximação, obtemos:

$$\begin{cases} x_Q = x_P + \frac{b_P}{a_P}(t_Q - t_P) \\ U_Q = U_P + \frac{c_P}{a_P}(t_Q - t_P) \end{cases} \quad (5.15)$$

Dessa forma, com 5.15 podemos, ao longo de uma dada característica, calcular aproximações da solução. Note que 5.12 é um problema de valor inicial, logo dado um ponto x_P no eixo x onde começa uma característica, já temos um valor inicial de $u(x_P, t_P) = u(x_P, 0) = U_P$. De posse desse valor, podemos calcular uma aproximação ao longo dessa característica, bastando fornecer um nível de tempo t_Q , isto é, o nível onde se deseja calcular uma aproximação.

Exemplos:

(a) Considere

$$f(x) = \text{sen}(x)$$

$$a(x, t) = 1$$

$$b(x, t) = 1$$

$$c(x, t) = 2\cos(x + t)$$

$$u(x, t) = \sin(x + t) \rightarrow \text{solução analítica}$$

Tomemos a característica da equação, com os dados acima, iniciando em $x = 2$ e suponhamos que se deseje calcular uma aproximação da função incógnita $u(x, t)$ em $t = 0.1$. Aplicando 5.15, temos:

$$x_P = 2$$

$$t_P = 0$$

$$t_Q = 0.1$$

$$a_P = b_P = 1 \rightarrow \frac{b_P}{a_P} = 1$$

$$\frac{c_P}{a_P} = 2\cos(x_P + t_P) = -0.8323$$

$$U_P = f(2) = \sin(2) = 0.9093$$

$$t_Q - t_P = 0.1 - 0 = 0.1$$

Com esses dados, segue

$$\begin{cases} x_Q = 2 + 1(0.1) = 2.100 \\ U_Q = 0.9093 - 0.8323(0.1) = 0.8261 \end{cases}$$

onde obtemos a aproximação $U_Q = 0.8261$.

Essa aproximação pode ser melhorada se dividirmos o intervalo $[0, 0.1]$ em 4 partes iguais, e daí calcularmos recursivamente as aproximações U_Q até $t = 0.1$, que é o nível de interesse. Segue abaixo um exemplo dessa idéia.

(b) Considerando o mesmo problema do exemplo anterior, temos:

1ª iteração

$$x_P = 2$$

$$t_P = 0$$

$$t_Q = 0.025$$

$$a_P = b_P = 1 \Rightarrow \frac{b_P}{a_P} = 1$$

$$\frac{c_P}{a_P} = 2\cos(x_P + t_P) = -0.8323$$

$$U_P = f(2) = \sin(2) = 0.9093$$

$$t_Q - t_P = 0.025 - 0 = 0.025$$

Com isso, segue

$$\begin{cases} x_Q = 2 + 1(0.025) = 2.025 \\ U_Q = 0.9093 - 0.8323(0.025) = 0.8885 \end{cases}$$

Temos uma aproximação U_Q para $t = 0.025$. Usaremos agora x_Q e U_Q , obtidos nessa iteração, para calcular U_Q em $t = 0.05$ na próxima iteração.

2ª iteração

$$x_P = 2.025$$

$$t_P = 0.025$$

$$t_Q = 0.05$$

$$a_P = b_P = 1 \Rightarrow \frac{b_P}{a_P} = 1$$

$$\frac{c_P}{a_P} = 2\cos(x_P + t_P) = -0.9221$$

$$U_P = 0.8885$$

$$t_Q - t_P = 0.05 - 0.025 = 0.025$$

$$\begin{cases} x_Q = 2.025 + 1(0.025) = 2.05 \\ U_Q = 0.8885 - 0.9221(0.025) = 0.8654 \end{cases}$$

3ª iteração

$$x_P = 2.05$$

$$t_P = 0.05$$

$$t_Q = 0.075$$

$$a_P = b_P = 1 \rightarrow \frac{b_P}{a_P} = 1$$

$$\frac{c_P}{a_P} = 2\cos(x_P + t_P) = -1.0097$$

$$U_P = 0.8654$$

$$t_Q - t_P = 0.075 - 0.05 = 0.025$$

$$\begin{cases} x_Q = 2.05 + 1(0.025) = 2.075 \\ U_Q = 0.8654 - 1.0097(0.025) = 0.8402 \end{cases}$$

4ª iteração

$$x_P = 2.075$$

$$t_P = 0.075$$

$$t_Q = 0.1$$

$$a_P = b_P = 1 \rightarrow \frac{b_P}{a_P} = 1$$

$$\frac{c_P}{a_P} = 2\cos(x_P + t_P) = -1.0947$$

$$U_P = 0.8402$$

$$t_Q - t_P = 0.1 - 0.075 = 0.025$$

$$\begin{cases} x_Q = 2.075 + 1(0.025) = 2.1 \\ U_Q = 0.8402 - 1.0947(0.025) = 0.8128 \end{cases}$$

Com isso, obtemos a aproximação final $U_Q = 0.8128$ em $t = 0.1$

(c) Para 10 partições, obteremos $U_Q = 0.8102$ em $t = 0.1$.

Observando a tabela

nº de partições	Aproximação	Solução	Erro
1	0.8261	0.8085	0.0176
4	0.8128	0.8085	0.0043
10	0.8102	0.8085	0.0017

vemos que à medida que aumentamos o número de partições, melhoramos as aproximações em $t = 0.1$. Dessa maneira, além de obter a aproximação desejada, no nível de tempo especificado, obtemos outras aproximações ao longo da característica iniciando em $x = 2$. Note que podemos aplicar esse método ao longo das características que tem início em um intervalo $[le, ld]$. Essas características terão início em pontos definidos por um número partições feito nesse intervalo. Por exemplo, se particionarmos, dividirmos o intervalo $[1, 4]$ em 3 partes iguais, aplicaremos o método nas características que tem início em $x = 1, 2, 3$ e 4. O algoritmo abaixo, aplicando o método das características, foi desenvolvido com esse fim.

Algoritmo para o Método das Características

1) Início

Entrada de dados. Entre com

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[le, ld]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$a(x, t) \rightarrow$ Coeficiente de u_t .

$b(x, t) \rightarrow$ Coeficiente de u_x .

$c(x, t) \rightarrow$ Segundo membro da equação $a(x, t)u_t + b(x, t)u_x = c(x, t)$

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \left\{ \begin{array}{l} h = \frac{|ld - le|}{M} \\ k = \frac{T}{N} \end{array} \right.$$

3) Cálculo da condição inicial

Para $i = 1$ a $M + 1$ faça

$$x = le + (i - 1)k$$

$$U(i, 1) = f(x)$$

Fim para

4) Cálculo das aproximações

Para $i = 1$ a $M + 1$ faça

$$x = le + (i - 1)h$$

Para $j = 2$ a $N + 1$ faça

$$t = (j - 1)k$$

$$a = a(x, t)$$

$$b = b(x, t)$$

$$c = c(x, t)$$

$$x = x + \frac{b}{a}k$$

$$U(i, j) = U(i, j - 1) + \frac{c}{a}k$$

Fim para

Fim para

5) Fim do algoritmo

Observe que, para uma melhor utilização dessa técnica, é aconselhável tomar um intervalo $[0, T]$ pequeno, e um número de partições N de bom tamanho, isto é, grande, para obter aproximações consideráveis. O acúmulo de erros é inevitável e, dependendo da equação, se tomarmos um intervalo $[0, T]$ grande, e um número de partições também grande, as aproximações obtidas poderão divergir muito da solução exata.

Apresentaremos a seguir os métodos de diferenças finitas para equações hiperbólicas.

5.4 Métodos de Diferenças Finitas

Considere a equação hiperbólica abaixo:

$$u_t + au_x = 0 \quad a > 0, \quad a \text{ constante}, \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}_+ \quad (5.16)$$

Trabalharemos com a equação 5.16 em uma malha como o da figura abaixo

Figura 5.3: Equação Hiperbólica - malha

onde

$[l_e, l_d]$ é o intervalo de domínio no eixo x .

$[0, T]$ é o intervalo de domínio no eixo t .

h e k são os espaçamentos definidos pelas partições nos eixos x e t , respectivamente.

Em alguns métodos trabalharemos em domínios da forma $[0, L]$ no eixo x , onde L é o limite à direita do intervalo considerado.

Antes de iniciarmos o desenvolvimento dos métodos, apresentaremos a condição de estabilidade para os métodos de diferenças finitas, quando aplicados na resolução numérica de uma equação hiperbólica.

5.4.1 Estabilidade

O que garante a estabilidade dos métodos de diferenças finitas, quando aplicados na resolução numérica de equações hiperbólicas, é a condição de **Courant-Friedrichs-Lewy**, abreviadamente, **condição CFL**. Para que essa condição seja satisfeita, a característica passando por um determinado ponto P no nível $t = jk$, atravesse a molécula

computacional, do método que estiver sendo usado, por entre dois pontos considerados pela fórmula. Para compreender melhor essa idéia, acompanhe o desenvolvimento abaixo.

Seja R o ponto onde a característica passando por P intercepta a malha passando pelo segmento $\overline{QS}$. Para que a condição CFL seja satisfeita devemos ter, nesse esquema, o ponto R situado entre os pontos do segmento $\overline{QS}$. Veja a ilustração a seguir:

Figura 5.4: Representação das características

Temos da equação 5.16 que

$$\frac{dx}{ds} = a \quad e \quad \frac{dt}{ds} = 1 \quad (5.17)$$

Isolando ds na primeira e substituindo-a na segunda, segue

$$\frac{dt}{dx} = \frac{1}{a}$$

onde obtemos a inclinação das características, as quais, para a equação hiperbólica que estamos trabalhando, são retas. Assim sendo, temos a seguinte relação

$$\operatorname{tg} \theta = \frac{dt}{dx} = \frac{1}{a} = \frac{k}{d}$$

onde d é a distância entre os pontos R e S . Daí temos que $d = ka$. A condição CFL exige que $0 \leq d \leq h$, ou $0 < ka < h$, ou melhor ainda

$$0 \leq pa \leq 1 \quad (h \neq 0) \quad \text{onde } p = \frac{k}{h}$$

Com isso, temos a condição de estabilidade para a equação hiperbólica que iremos trabalhar. Os métodos explícitos, que apresentaremos a seguir, serão sempre estáveis

quando a condição $0 \leq pa \leq 1$ for satisfeita, caso contrário, haverá instabilidade. Já nos métodos implícitos, que serão apresentados após a próxima seção, a condição CFL sempre será satisfeita.

5.5 Métodos de Diferenças Explícitos

Considere o seguinte problema

$$\begin{cases} u_t + au_x = 0 & a > 0 \quad (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = f(x) & x \in \mathbb{R} \end{cases} \quad \text{condição inicial} \quad (5.18)$$

Assim sendo, definindo uma malha para o problema acima, teremos:

Figura 5.5: Equações hiperbólicas - ilustração de uma malha

5.5.1 Esquema Explícito de Primeira Ordem

Nesse método, considere os pontos $P(ih, (j+1)k)$, $Q((i-1)h, jk)$ e $S(ih, jk)$, os quais representam a seguinte molécula computacional:

Figura 5.6: Molécula computacional do esquema explícito de primeira ordem

Suponhamos que seja conhecida uma solução, ou uma aproximação nos pontos Q e S , e determinaremos uma aproximação no ponto P . Seja u a solução do problema. Desenvolvendo u em série de Taylor no ponto P , temos:

$$u_P = u_S + k(u_S)_t + \frac{k^2}{2!}(u_S)_{tt} + \frac{k^3}{3!}(u_S)_{ttt} + \dots$$

Observe que

$$u_P = \exp[k(u_S)_t] \Rightarrow u_P = \exp[-ka(u_S)_x], \text{ pois } u_t = -au_x$$

Considerando agora as aproximações de primeira ordem:

$$u_P \cong u_S - ak(u_S)_x \quad \text{e} \quad (u_S)_x \cong \frac{1}{h}(u_S - u_Q) \quad (5.19)$$

Com isso, temos:

$$U_P = (1 - pa)U_S + paU_Q \quad \text{onde} \quad p = \frac{k}{h} \quad (5.20)$$

$\Downarrow$

$$U_{i,j} = (1 - pa)U_{i,j-1} + paU_{i-1,j-1} \quad \text{onde} \quad p = \frac{k}{h} \quad (5.21)$$

Por 5.21, vemos que as aproximações em um determinado nível de tempo são calculados a partir de aproximações do nível de tempo anterior. Mais adiante apresentaremos um exemplo do processo de aplicação do método.

Erro de Truncamento Local

Para determinar o ETL, utiliza-se a série de Taylor:

$$u_P = u_S + k(u_S)_t + \frac{k^2}{2!}(u_S)_{tt} + \frac{k^3}{3!}(u_S)_{ttt} + \dots \quad (5.22)$$

$$u_Q = u_S - h(u_S)_t + \frac{h^2}{2!}(u_S)_{tt} - \frac{h^3}{3!}(u_S)_{ttt} + \dots \quad (5.23)$$

Substituindo 5.22 e 5.23 em 5.20 obteremos o seguinte ETL:

$$\frac{k^2}{2!} \left[(u_S)_{tt} - a \frac{k}{h} (u_S)_{xx} \right] + \dots = O \left(k^2 + \frac{k^3}{h} \right)$$

Vamos ilustrar a aplicação desse método com um exemplo.

Exemplo:

Considere o seguinte problema

$$\begin{cases} u_t + 0.5u_x = 0 & (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = \text{sen}(x) & x \in \mathbb{R} \end{cases} \quad \text{condição inicial} \quad (5.24)$$

e os dados para a definição da malha

$$[le, ld] = [1, 1.3]$$

$$[0, T] = [0, 0.18]$$

$$M = 3 \rightarrow \text{partições em } [le, ld]$$

$$N = 3 \rightarrow \text{partições em } [0, T]$$

de onde temos que

$$h = \frac{|ld - le|}{M} = \frac{|1.3 - 1|}{3} = 0.1$$

$$k = \frac{T}{N} = \frac{0.18}{3} = 0.06$$

$$pa = \frac{k}{h}a = \frac{0.06}{0.1}0.5 = 0.3$$

Surge aqui um problema. Observe que, para calcular as aproximações nos pontos na fronteira à esquerda da malha, necessitaremos de valores em pontos que estão fora da malha. Mas o cálculo desses valores é possível de ser realizado, pois temos conhecimento da condição inicial em $[-\infty, +\infty]$, e por ela poderemos calcular aproximações de $u(x, y)$ em

pontos auxiliares à esquerda da malha, e com essas aproximações, calcular as aproximações na fronteira à esquerda da malha. Para compreender melhor essa idéia, considere a ilustração a seguir.

Figura 5.7: Aplicação do esquema explícito de primeira ordem

Note, na figura, que poderemos calcular as aproximações em A, B, C, D, E e F , calculando os pontos marcados por um x , usando a condição inicial, da qual temos conhecimento.

Assim sendo, aplicando o método, temos:

- Fixando $j = 1$ e variando i de 1 a $M + N + 1 = 7$, obteremos os valores que a função incógnita assume no eixo x . Calculamos isso usando a condição inicial $f(x) = \text{sen}(x)$. Os valores para o nosso exemplo são:

$$\begin{aligned} U_{1,1} &= u(x_1, 0) = u(0.7, 0) = 0.6442 \\ U_{2,1} &= u(x_2, 0) = u(0.8, 0) = 0.7174 \\ U_{3,1} &= u(x_3, 0) = u(0.9, 0) = 0.7833 \\ U_{4,1} &= u(x_4, 0) = u(1, 0) = 0.8415 \\ U_{5,1} &= u(x_5, 0) = u(1.1, 0) = 0.8912 \\ U_{6,1} &= u(x_6, 0) = u(1.2, 0) = 0.9320 \\ U_{7,1} &= u(x_7, 0) = u(1.3, 0) = 0.9636 \end{aligned}$$

- Fixando agora $j = 2$ e variando i de 2 a $M + N + 1 = 7$, obteremos através de 5.21 as seguintes aproximações:

$$\begin{aligned} U_{2,2} &= 0.7U_{2,1} + 0.3U_{1,1} = 0.6954 \\ U_{3,2} &= 0.7U_{3,1} + 0.3U_{2,1} = 0.7635 \\ U_{4,2} &= 0.7U_{4,1} + 0.3U_{3,1} = 0.8240 \end{aligned}$$

$$U_{5,2} = 0.7U_{5,1} + 0.3U_{4,1} = 0.8763$$

$$U_{6,2} = 0.7U_{6,1} + 0.3U_{5,1} = 0.9198$$

$$U_{7,2} = 0.7U_{7,1} + 0.3U_{6,1} = 0.9541$$

- Fixando agora $j = 3$ e variando i de 3 a $M + N + 1 = 7$, segue

$$U_{3,3} = 0.7U_{3,2} + 0.3U_{2,2} = 0.7431$$

$$U_{4,3} = 0.7U_{4,2} + 0.3U_{3,2} = 0.8059$$

$$U_{5,3} = 0.7U_{5,2} + 0.3U_{4,2} = 0.8606$$

$$U_{6,3} = 0.7U_{6,2} + 0.3U_{5,2} = 0.9067$$

$$U_{7,3} = 0.7U_{7,2} + 0.3U_{6,2} = 0.9438$$

- Fixando agora $j = 4$ e variamos $i = 4$ a $M + N + 1 = 7$, segue

$$U_{4,4} = 0.7U_{4,3} + 0.3U_{3,3} = 0.7870$$

$$U_{5,4} = 0.7U_{5,3} + 0.3U_{4,3} = 0.8442$$

$$U_{6,4} = 0.7U_{6,3} + 0.3U_{5,3} = 0.8929$$

$$U_{7,4} = 0.7U_{7,3} + 0.3U_{6,3} = 0.9327$$

Colocando a matriz com as aproximações em uma tabela, temos:

$U_{i,j}$	1	2	3	4
1	0.6442	-	-	-
2	0.7174	0.6954	-	-
3	0.7833	0.7635	0.7431	-
4	0.8415	0.8240	0.8059	0.7870
5	0.8912	0.8763	0.8606	0.8442
6	0.9320	0.9198	0.9067	0.8929
7	0.9636	0.9541	0.9438	0.9327

Observe a tabela acima. Nas linhas de 1 a 3, temos uma matriz formada pelas aproximações auxiliares, as quais foram necessárias para calcular as aproximações de interesse na fronteira à esquerda da malha, as que estão na linha 4 e colunas de 2 a 4. As aproximações de interesse, na tabela, se encontram delimitada pelas linhas de 4 a 7. Segue o algoritmo para a aplicação desse método.

Algoritmo para o Esquema Explícito de Primeira Ordem

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_x .

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[le, ld]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \left\{ \begin{array}{l} h = \frac{|ld - le|}{M} \\ k = \frac{T}{N} \end{array} \right.$$

3) Cálculo da condição inicial

$$aux = le - hN$$

Para $i = 1$ a $M + N + 1$ faça

$$x = aux + (i - 1)k$$

$$U(i, 1) = f(x)$$

Fim para

4) Cálculo das aproximações

$$\text{Faça } p = \frac{k}{h}$$

$$cont = -1$$

Para $j = 2$ a $N + 1$ faça

$$cont = cont + 1$$

Para $i = 2 + cont$ a $M + N + 1$ faça

$$U(i, j) = (1 - pa)U(i, j - 1) + paU(i - 1, j - 1)$$

Fim para

Fim para

$$\text{Solução} \rightarrow U(N + 1 \text{ a } M + N + 1, 1 \text{ a } N + 1)$$

5) Fim do algoritmo

5.5.2 Esquema de Lax-Wendroff

Os pontos envolvidos nesse método são $P(ih, (j+1)k)$, $Q((i-1)h, jk)$, $S(ih, jk)$ e $T((i+1)h, jk)$, e a molécula computacional correspondente é:

Figura 5.8: Molécula computacional do Esquema de Lax-Wendroff

Expandindo u_P em série de Taylor, segue

$$u_P = u_S + k(u_S)_t + \frac{k^2}{2!}(u_S)_{tt} + \frac{k^3}{3!}(u_S)_{ttt} + \dots \quad (5.25)$$

Como $(u_S)_t = -a(u_S)_x$, então

$$(u_P) = u_S - ak(u_S)_x + a^2 \frac{k^2}{2!}(u_S)_{xx} - \dots$$

Aproximando u_P da expansão acima pelos termos até segunda ordem, e substituindo as derivadas em x pela aproximação de diferença centrada, obteremos:

$$U_Q = (1 - p^2 a^2)U_S - \frac{pa}{2}(1 - pa)U_T + \frac{pa}{2}U_Q \quad \text{onde } p = \frac{k}{h}$$

$\Downarrow$

$$U_{i,j} = (1 - p^2 a^2)U_{i,j-1} - \frac{pa}{2}(1 - pa)U_{i-1,j-1} + \frac{pa}{2}U_{i+1,j-1} \quad (5.26)$$

De maneira análoga ao método anterior, as aproximações em um determinado nível dependerá de aproximações do nível anterior.

Erro de Truncamento Local

O método de Lax-Wendroff tem o ETL da ordem $O(h^2 + k^2)$

Da mesma forma que o método anterior, após definida uma malha, o método de Lax-Wendroff precisa de aproximações auxiliares em pontos fora da malha para calcular as aproximações de interesse, isto é, na malha desejada. A diferença para o método anterior está no fato do mesmo precisar de aproximações auxiliares à esquerda e à direita da malha. A figura abaixo ilustra essa situação:

Figura 5.9: Aplicação do esquema de Lax-Wendroff

Observe que precisaremos calcular aproximações em pontos fora da malha para obter aproximações na fronteira à esquerda, pontos A, B, C, D, E e F , e na fronteira à direita, pontos G, H, I, J, L e M .

Vejamos um exemplo a seguir:

Exemplo:

Considere o problema 5.29 e os seguintes dados para a definição da malha:

$$[l_e, l_d] = [1, 1.2]$$

$$[0, T] = [0, 0.12]$$

$$M = 2 \rightarrow \text{partições em } [l_e, l_d]$$

$N = 2 \rightarrow$ partições em $[0, T]$

Desses dados, temos que:

$$h = \frac{|ld - le|}{M} = \frac{|1.2 - 1|}{2} = 0.1$$

$$k = \frac{T}{N} = \frac{0.12}{2} = 0.06$$

$$pa = \frac{k}{h}a = \frac{0.1}{0.06}0.5 = 0.3$$

Aplicando o método, temos:

• Fixando $j = 1$ e variando $i = 1$ a $M + 2N + 1 = 7$ para obter os valores que a função incógnita assume no eixo x . Calculamos isso usando a condição inicial $f(x) = \text{sen}(x)$. Os valores para o nosso exemplo são:

$$\begin{aligned} U_{1,1} &= u(x_1, 0) = u(0.8, 0) = 0.7174 \\ U_{2,1} &= u(x_2, 0) = u(0.9, 0) = 0.7833 \\ U_{3,1} &= u(x_3, 0) = u(1, 0) = 0.8415 \\ U_{4,1} &= u(x_4, 0) = u(1.1, 0) = 0.8912 \\ U_{5,1} &= u(x_5, 0) = u(1.2, 0) = 0.9320 \\ U_{6,1} &= u(x_6, 0) = u(1.3, 0) = 0.9636 \\ U_{7,1} &= u(x_7, 0) = u(1.4, 0) = 0.9854 \end{aligned}$$

• Fixando agora $j = 2$ e variando $i = 2$ a $M + 2N = 6$, obtemos através de 5.26 as seguintes aproximações:

$$\begin{aligned} U_{2,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.7644 \\ U_{3,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.7635 \\ U_{4,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.8240 \\ U_{5,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.8763 \\ U_{6,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.9198 \end{aligned}$$

• Fixando $j = 3$ e variando $i = 3$ a $M + 2N - 1 = 5$, temos:

$$\begin{aligned} U_{3,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.8076 \\ U_{4,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.8624 \\ U_{5,2} &= 0.91U_{2,1} - 0.105U_{1,1} + 0.15U_{3,1} = 0.9087 \end{aligned}$$

Com essas aproximações montamos a seguinte tabela:

$U_{i,j}$	1	2	3
1	0.7174	-	-
2	0.7833	0.7644	-
3	0.8415	0.8249	0.8076
4	0.8912	0.8772	0.8624
5	0.9320	0.9208	0.9087
6	0.9636	0.9551	-
7	0.9854	-	-

Temos na tabela, nas linhas de 1 e 2, as aproximações auxiliares à esquerda da malha. Nas linhas 6 e 7, as aproximações auxiliares à direita da malha. Nas linhas de 3 a 5 as aproximações desejadas. Segue abaixo o algoritmo do método.

Algoritmo para o Esquema de Lax-Wendroff

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_x .

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[le, ld]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \begin{cases} h = \frac{|ld - le|}{M} & k = \frac{T}{N} \end{cases}$$

3) Cálculo da condição inicial

$$aux = le - hN$$

Para $i = 1$ a $M + 2N + 1$ faça

$$x = aux + (i - 1)k$$

$$U(i, 1) = f(x)$$

Fim para

4) Cálculo das aproximações

$$\text{Faça } p = \frac{k}{h}$$

$$cont = -1$$

Para $j = 1$ a N faça

$$cont = cont + 1$$

Para $i = 2 + \text{cont}$ a $M + 2N - \text{cont}$ faça

$$U(i, j + 1) = (1 - pa^2)U(i, j) - \frac{pa}{2}(1 - pa)U(i + 1, j) + \frac{pa}{2}(1 + pa)U(i - 1, j)$$

Fim para

Fim para

Solução $\rightarrow U(N + 1 \quad a \quad M + N + 1, 1 \quad a \quad N + 1)$

5) Fim do algoritmo

5.5.3 Esquema Leap-Frog

Esse método envolve 3 níveis de tempo e os pontos envolvidos são $P(ih, (j + 1)k)$, $W(ih, (j - 1)k)$, $Q((i - 1)h, jk)$ e $T((i + 1)h, jk)$. A molécula computacional que representa esse método é:

Figura 5.10: Molécula computacional do esquema leap-frog

A fórmula desse método é dado por:

$$U_P = U_W - pa(U_T - U_Q) \quad (5.27)$$

$\Downarrow$

$$U(i, j) = U_{i,j-2} - pa(U_{i+1,j-1} - U_{i-1,j-1}) \quad (5.28)$$

O problema na aplicação desse método está em ele ser de dois passos. Portanto, precisaremos calcular as aproximações no segundo nível por algum outro método numérico.

Esse problema pode ser resolvido usando o Método Lax-Wendroff para calcular as aproximações do segundo nível. Outro problema é que esse método também precisa de aproximações auxiliares fora da malha de interesse, para obter aproximações na fronteira à direita e à esquerda. A figura a seguir ilustra a situação.

Figura 5.11: Aplicação do Esquema leap-frog

Tomemos um exemplo para ilustrar o método.

Exemplo:

Consideremos o seguinte problema

$$\begin{cases} u_t + 0.25u_x = 0 & (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = \text{sen}(x) & x \in \mathbb{R} \end{cases} \quad \text{condição inicial} \quad (5.29)$$

com os seguintes dados de definição da malha:

$$[l_e, l_d] = [2, 2.2]$$

$$[0, T] = [0, 0.24]$$

$$M = 2 \rightarrow \text{partições em } [l_e, l_d]$$

$$N = 2 \rightarrow \text{partições em } [0, T]$$

Desses dados, temos que:

$$h = \frac{|l_d - l_e|}{M} = \frac{|2.2 - 2|}{2} = 0.1$$

$$k = \frac{T}{N} = \frac{0.24}{2} = 0.12$$

$$pa = \frac{k}{h}a = \frac{0.12}{0.1}0.24 = 0.288$$

Aplicando o método, temos:

- Fixando $j = 1$ e variando i de 1 a $M + 2N + 1 = 7$ para obter os valores que a função incógnita assume no eixo x . Novamente, calculamos isso usando a condição inicial $f(x) = \sin(x)$. Os valores para o nosso exemplo são:

$$U_{1,1} = u(x_1, 0) = u(1.8, 0) = 0.9738$$

$$U_{2,1} = u(x_2, 0) = u(1.9, 0) = 0.9463$$

$$U_{3,1} = u(x_3, 0) = u(2, 0) = 0.9093$$

$$U_{4,1} = u(x_4, 0) = u(2.1, 0) = 0.8632$$

$$U_{5,1} = u(x_5, 0) = u(2.2, 0) = 0.8085$$

$$U_{6,1} = u(x_6, 0) = u(2.3, 0) = 0.7457$$

$$U_{7,1} = u(x_7, 0) = u(2.4, 0) = 0.6755$$

- Fixando agora $j = 2$ e variando i de 2 a $M + 2N = 6$, obtemos através de 5.21, Método de Lax-Wendroff, as seguintes aproximações:

$$U_{2,2} = 0.9171U_{2,1} - 0.1321U_{1,1} + 0.144U_{3,1} = 0.9552$$

$$U_{3,2} = 0.9171U_{3,1} - 0.1321U_{2,1} + 0.144U_{4,1} = 0.9209$$

$$U_{4,2} = 0.9171U_{4,1} - 0.1321U_{3,1} + 0.144U_{5,1} = 0.8774$$

$$U_{5,2} = 0.9171U_{5,1} - 0.1321U_{4,1} + 0.144U_{6,1} = 0.8251$$

$$U_{6,2} = 0.9171U_{6,1} - 0.1321U_{5,1} + 0.144U_{7,1} = 0.7646$$

- Fixando $j = 3$ e variando i de 3 a $M + 2N - 1 = 5$, obtemos por 5.28 as seguintes aproximações:

$$U(3, 3) = U_{3,1} - pa(U_{4,2} - U_{2,2}) = 0.9317$$

$$U(4, 3) = U_{4,1} - pa(U_{5,2} - U_{3,2}) = 0.8908$$

$$U(5, 3) = U_{5,1} - pa(U_{6,2} - U_{4,2}) = 0.8410$$

Colocando esses dados em uma tabela, temos:

$U_{i,j}$	1	2	3
1	0.9738	-	-
2	0.9463	0.9552	-
3	0.9093	0.9209	0.9317
4	0.8632	0.8774	0.8908
5	0.8085	0.8251	0.8410
6	0.7457	0.7646	-
7	0.6755	-	-

Temos na tabela, nas linhas 1, 2, 6 e 7 aproximações auxiliares que foram necessárias para o cálculo das aproximações de interesse, as quais se encontram nas linhas 3, 4 e 5. Segue abaixo o algoritmo do Método Leap-Frog.

Algoritmo para o Esquema Leap-Frog

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_x .

$[le, ld] \rightarrow$ Intervalo do domínio no eixo x , com $le < ld$.

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[le, ld]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \begin{cases} h = \frac{|ld - le|}{M} & k = \frac{T}{N} \end{cases}$$

3) Cálculo da condição inicial

$$aux = le - hM$$

Para $i = 1$ a $M + 2N + 1$ faça

$$x = aux + (i - 1)h$$

$$U(i, 1) = f(x)$$

Fim para

4) Cálculo das aproximações

$$\text{Faça } p = \frac{k}{h}$$

$$cont = -1$$

Para $j = 2$ a $N + 1$ faça

$$cont = cont + 1$$

Para $i = 2 + cont$ a $M + 2N - cont$ faça

Se $(j = 2)$ então faça

$$U(i, j) = (1 - pa^2)U(i, j-1) - \frac{pa}{2}(1 - pa)U(i+1, j-1) + \frac{pa}{2}(1 + pa)U(i-1, j-1)$$

senão faça

$$U(i, j) = U_{i,j-2} - pa(U_{i+1,j-1} - U_{i-1,j-1})$$

Fim se

Fim para

Fim para

Solução $\rightarrow U(N+1, a, M+N+1, 1, a, N+1)$

5) Fim do algoritmo

5.6 Métodos de Diferenças Implícitos

A aplicação desses métodos requer o acréscimo de uma condição de fronteira para a sua utilização. Assim sendo, trabalharemos com o seguinte problema

$$\begin{cases} u_t + au_x = 0 & a > 0 & (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = f(x) & x \in \mathbb{R}_+ & \text{condição inicial} \\ u(0, t) = g(t) & t \in \mathbb{R}_+ & \text{condição de fronteira} \end{cases} \quad (5.30)$$

definido na seguinte malha:

Figura 5.12: Equações hiperbólicas - malha para os métodos de DF implícitos

Segue agora a apresentação de dois métodos: Método Implícito de Primeira Ordem e Método Implícito de Wendroff. Esses dois métodos são incondicionalmente estáveis, pois ambos satisfazem a condição CFL.

5.6.1 Esquema Implícito de Primeira Ordem

Esse método envolve os pontos $P(ih, jk)$, $V((i-1)h, (j+1)k)$ e $S(ih, jk)$, e a molécula computacional correspondente desse método é dada por:

Molécula computacional:

Figura 5.13: Molécula computacional do esquema implícito de primeira ordem

Vamos desenvolver agora o processo de obtenção do método.

Desenvolvendo u_S em série de Taylor, temos:

$$u_S = u_P - k(u_P)_t + \frac{k^2}{2!}(u_P)_{tt} + \dots \quad (5.31)$$

Como $u_t = -au_x$, segue de 5.31 que

$$u_S = u_P + ak(u_P)_x + a^2 \frac{k^2}{2!}(u_P)_{xx} + \dots \quad (5.32)$$

Aproximando u_S , até o termo de primeira ordem em 5.32, e utilizando a aproximação

$$(u_P)_x = \frac{u_P - u_V}{h}$$

obteremos

$$U_S = (1 + pa)U_P - paU_V \quad (5.33)$$

Devido ao problema ter uma condição inicial e uma condição de fronteira, podemos isolar U_P em 5.33, obtendo

$$U_P = \frac{U_S}{1+pa} + \frac{paU_V}{1+pa} \Rightarrow U_{i,j} = \frac{U_{i,j-1}}{1+pa} + \frac{paU_{i-1,j}}{1+pa} \quad (5.34)$$

Dessa forma, podemos calcular U_P explicitamente.

Vamos ilustrar agora a aplicação desse método com um exemplo.

Exemplo:

Considere o seguinte problema:

$$\begin{cases} u_t + 2u_x = 0 & (x, t) \in \mathbb{R} \times \mathbb{R}_+ \\ u(x, 0) = \cos(x) & x \in \mathbb{R} \\ u(0, t) = \cos(-2t) & t \in \mathbb{R} \end{cases} \quad \begin{array}{l} \text{condição inicial} \\ \text{condição de fronteira} \end{array} \quad (5.35)$$

e os dados para a definição da malha

$$[0, L] = [0, 0.3]$$

$$[0, T] = [0, 0.03]$$

$$M = 3 \rightarrow \text{partições em } [0, L]$$

$$N = 3 \rightarrow \text{partições em } [0, T]$$

Desses dados, temos que:

$$h = \frac{L}{M} = \frac{0.3}{3} = 0.1$$

$$k = \frac{T}{N} = \frac{0.03}{3} = 0.01$$

$$pa = \frac{k}{h}a = \frac{0.01}{0.1}2 = 0.2$$

Aplicando o método, temos:

- Fixando $j = 1$ e variando $i = 1$ a $M + 1 = 4$, temos pela condição inicial que

$$U_{1,1} = u(x_1, 0) = u(0, 0) = f(0) = 1.0000$$

$$U_{2,1} = u(x_2, 0) = u(0.1, 0) = f(0.1) = 0.9950$$

$$U_{3,1} = u(x_3, 0) = u(0.2, 0) = f(0.2) = 0.9801$$

$$U_{4,1} = u(x_4, 0) = u(0.3, 0) = f(0.3) = 0.9553$$

e pela condição de fronteira, fixando $i = 1$ e variando j de 1 a $N + 1$, temos

$$\begin{aligned}U_{1,1} &= u(0, t_1) = u(0, 0) = g(0) = 1.0000 \\U_{1,2} &= u(0, t_2) = u(0, 0.01) = g(0.01) = 0.9998 \\U_{1,3} &= u(0, t_3) = u(0, 0.02) = g(0.02) = 0.9992 \\U_{1,4} &= u(0, t_4) = u(0, 0.03) = g(0.03) = 0.9982\end{aligned}$$

- Fixando $j = 2$ e variando $i = 2$ a $M + 1 = 4$, temos por 5.34 as seguintes aproximações:

$$\begin{aligned}U_{2,2} &= 0.8333U_{2,1} + 0.1961U_{1,2} = 0.9958 \\U_{3,2} &= 0.8333U_{3,1} + 0.1961U_{2,2} = 0.9827 \\U_{4,2} &= 0.8333U_{4,1} + 0.1961U_{3,2} = 0.9599\end{aligned}$$

- Fixando $j = 3$ e variando $i = 2$ a $M + 1 = 4$, temos:

$$\begin{aligned}U_{2,3} &= 0.8333U_{2,2} + 0.1961U_{1,3} = 0.9964 \\U_{3,3} &= 0.8333U_{3,2} + 0.1961U_{2,3} = 0.9850 \\U_{4,3} &= 0.8333U_{4,2} + 0.1961U_{3,3} = 0.9641\end{aligned}$$

- Fixando $j = N + 1 = 4$ e variando $i = 2$ a $M + 1 = 4$, temos:

$$\begin{aligned}U_{2,4} &= 0.8333U_{2,3} + 0.1961U_{1,4} = 0.9967 \\U_{3,4} &= 0.8333U_{3,3} + 0.1961U_{2,4} = 0.9869 \\U_{4,4} &= 0.8333U_{4,3} + 0.1961U_{3,4} = 0.9679\end{aligned}$$

Colocando em uma tabela essas aproximações, temos:

$U_{i,j}$	1	2	3	4
1	1.0000	0.9998	0.9992	0.9982
2	0.9950	0.9958	0.9964	0.9967
3	0.9801	0.9827	0.9850	0.9869
4	0.9553	0.9599	0.9641	0.9679

A vantagem desse método, em relação aos métodos explícitos, está no fato de mesmo não precisar de aproximações auxiliares. Segue o algoritmo do método.

Algoritmo para o Esquema Implícito de Primeira Ordem

1) Início

Entrada de dados. Entre com

 $a \rightarrow$ Coeficiente de u_x . $[0, L] \rightarrow$ Intervalo do domínio no eixo x . $T \rightarrow$ Nível de tempo máximo, com $T > 0$. $M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$. $N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$. $f(x) \rightarrow$ Condição inicial.2) Cálculo dos espaçamentos h e $k \Rightarrow \begin{cases} h = \frac{L}{M} & k = \frac{T}{N} \end{cases}$

3) Cálculo da condição inicial e da condição de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)k$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

Fim para

4) Cálculo das aproximações

$$\text{Faça } p = \frac{k}{h}$$

Para $j = 1$ a N façaPara $i = 2$ a $M + 1$ faça

$$U_{i,j+1} = \frac{U_{i,j}}{1 + pa} + \frac{paU_{i-1,j+1}}{1 + pa}$$

Fim para

Fim para

5) Fim do algoritmo

5.6.2 Esquema Implícito de Wendroff

Esse método envolve os pontos $P(ih, (j+1)k)$, $W((i+1)h, (j+1)k)$, $S(ih, jk)$ e $T((i+1)h, jk)$, com a molécula computacional da forma:

Figura 5.14: Molécula computacional do esquema implícito de Wendroff

A fórmula desse método é dada por:

$$U_W = U_S + \frac{1 - pa}{1 + pa}(U_T - U_P) \quad (5.36)$$

$\Downarrow$

$$U_{i,j} = U_{i-1,j-1} + \frac{1 - pa}{1 + pa}(U_{i,j-1} - U_{i-1,j}) \quad (5.37)$$

Note que U_W pode ser calculado explicitamente. Segue abaixo um exemplo de aplicação desse método.

Exemplo:

Considerando o mesmo problema exemplo do método anterior, mudando apenas o valor de a para 0.4, e a seguinte malha

$$[le, ld] = [0, 0.3]$$

$$[0, T] = [0, 0.15]$$

$$M = 3 \rightarrow \text{partições em } [0, L]$$

$$N = 3 \rightarrow \text{partições em } [0, T]$$

$$h = \frac{L}{M} = \frac{0.3}{3} = 0.1$$

$$h = \frac{T}{N} = \frac{0.15}{3} = 0.05$$

$$pa = \frac{k}{h}a = \frac{0.05}{0.1}0.4 = 0.2$$

temos:

- Fixando $j = 1$ e variando $i = 1$ a $M + 1 = 4$, temos pela condição inicial que

$$\begin{aligned} U_{1,1} &= u(x_1, 0) = u(0, 0) = f(0) = 1.0000 \\ U_{2,1} &= u(x_2, 0) = u(0.1, 0) = f(0.1) = 0.9950 \\ U_{3,1} &= u(x_3, 0) = u(0.2, 0) = f(0.2) = 0.9801 \\ U_{4,1} &= u(x_4, 0) = u(0.3, 0) = f(0.3) = 0.9553 \end{aligned}$$

e pela condição de fronteira, fixando $i = 1$ e variando j de 1 a $N + 1$, temos

$$\begin{aligned} U_{1,1} &= u(0, t_1) = u(0, 0) = g(0) = 1.0000 \\ U_{1,2} &= u(0, t_2) = u(0, 0.01) = g(0.01) = 0.9998 \\ U_{1,3} &= u(0, t_3) = u(0, 0.02) = g(0.02) = 0.9992 \\ U_{1,4} &= u(0, t_4) = u(0, 0.03) = g(0.03) = 0.9982 \end{aligned}$$

- Fixando $j = 2$ e variando $i = 2$ a $M + 1 = 4$, temos por 5.37 as seguintes aproximações:

$$\begin{aligned} U_{2,2} &= U_{1,1} + 0.6667(U_{2,1} - U_{1,2}) = 0.9968 \\ U_{3,2} &= U_{2,1} + 0.6667(U_{3,1} - U_{2,2}) = 0.9838 \\ U_{4,2} &= U_{3,1} + 0.6667(U_{4,1} - U_{3,2}) = 0.9611 \end{aligned}$$

- Fixando $j = 3$ e variando $i = 1$ a $M + 1 = 4$, temos:

$$\begin{aligned} U_{2,3} &= U_{1,2} + 0.6667(U_{2,2} - U_{1,3}) = 0.9982 \\ U_{3,3} &= U_{1,2} + 0.6667(U_{3,2} - U_{2,3}) = 0.9872 \\ U_{4,3} &= U_{1,2} + 0.6667(U_{4,2} - U_{3,3}) = 0.9664 \end{aligned}$$

- Fixando $j = N + 1 = 4$ e variando $i = 1$ a $M + 1 = 4$, temos:

$$\begin{aligned} U_{2,4} &= U_{1,2} + 0.6667(U_{2,3} - U_{1,4}) = 0.9992 \\ U_{3,4} &= U_{1,2} + 0.6667(U_{3,3} - U_{2,4}) = 0.9902 \\ U_{4,4} &= U_{1,2} + 0.6667(U_{4,3} - U_{3,4}) = 0.9713 \end{aligned}$$

Colocando os valores em uma tabela, temos:

$U_{i,j}$	1	2	3	4
1	1.0000	0.9998	0.9992	0.9982
2	0.9950	0.9968	0.9982	0.9992
3	0.9801	0.9838	0.9872	0.9902
4	0.9553	0.9611	0.9664	0.9713

Segue abaixo o algoritmo do método.

Algoritmo para o Esquema Implícito de Wendroff

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_x .

$[0, L] \rightarrow$ Intervalo do domínio no eixo x .

$T \rightarrow$ Nível de tempo máximo, com $T > 0$.

$M \rightarrow$ Número de passos, ou partições, no intervalo $[0, L]$.

$N \rightarrow$ Número de passos, ou partições, no intervalo $[0, T]$.

$f(x) \rightarrow$ Condição inicial.

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \begin{cases} h = \frac{L}{M} & k = \frac{T}{N} \end{cases}$$

3) Cálculo da condição inicial e da condição de fronteira

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)k$$

$$U(i, 1) = f(x)$$

Fim para

Para $j = 1$ a $N + 1$ faça

$$t = (j - 1)k$$

$$U(1, j) = g(t)$$

Fim para

4) Cálculo das aproximações

$$\text{Faça } p = \frac{k}{h}$$

Para $j = 1$ a N faça

Para $i = 2$ a $M + 1$ faça

$$U_{i,j} = U_{i-1,j-1} + \frac{1 - pa}{1 + pa}(U_{i,j-1} - U_{i-1,j})$$

Fim para

Fim para

5) Fim do algoritmo

5.7 Método de Diferenças Finitas para a Equação da Onda

Considere o seguinte problema

$$\begin{cases} u_{tt} - a^2 u_{xx} = 0 \\ u(x, 0) = f(x) & x \in [0, L] \\ u_t(x, 0) = g(x) & x \in [0, L] \\ u(0, t) = h_1(t) & t > 0 \\ u(L, t) = h_2(t) & t > 0 \end{cases} \quad (5.38)$$

Discretizando as derivadas segundas da equação da onda em ??, usando diferenças centradas de 2ª ordem, e fazendo uma média ponderada de valores dos estágios $j - 1$, j e $j + 1$, temos

$$\begin{aligned} \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} &= \frac{a^2}{k^2} \{ \eta (U_{i+1,j+1} - 2U_{i+1,j} + U_{i+1,j-1}) \\ &+ (1 - 2\eta) (U_{i,j+1} - 2U_{i,j} + U_{i,j-1}) \\ &+ \eta (U_{i-1,j+1} - 2U_{i-1,j} + U_{i-1,j-1}) \} \end{aligned} \quad (5.39)$$

A seguir apresentaremos a molécula computacional do presente método, o desenvolvimento de todo o processo para obtenção das aproximações e finalizando o algoritmo.

Esse método pode ser representado pela seguinte molécula computacional:

Figura 5.15: Molécula computacional para o método de DF - Equação da Onda

É um método de dois passos, ou 3 níveis, portanto precisaremos calcular uma aproximação inicial para $t = k$, isto é, no segundo nível.

Vamos desenvolver a partir da equação 5.39 o método de cálculo. Vamos chamar $\omega = \frac{a^2}{k^2}$. Isolando no primeiro membro de 5.39 os termos referentes a $j + 1$, e no segundo os referentes a $j - 1$ e j , teremos

$$\begin{aligned} & -\omega\eta U_{i-1,j+1} + (1 + 2\omega\eta) U_{i,j+1} - \omega\eta U_{i+1,j+1} = \\ & \quad \omega\eta U_{i-1,j-1} - (1 + 2\omega\eta) U_{i,j-1} + \omega\eta U_{i+1,j-1} \\ & + \omega(1 - 2\eta) U_{i-1,j} + (2 - 2\omega + 4\omega\eta) U_{i,j} + \omega(1 - 2\eta) U_{i+1,j} \end{aligned} \quad (5.40)$$

Considere agora uma malha definida em um domínio $[0, L] \times [0, T]$ com M partições em $[0, L]$ e N partições em $[0, T]$. Veja figura abaixo:

Figura 5.16: Malha definida no domínio da Equação da Onda

Fixando um nível de tempo, e aplicando 5.40 em cada ponto desse nível, chegaremos a um sistema linear, envolvendo matrizes tridiagonais, com a seguinte forma

$$\begin{aligned}
& \underbrace{\begin{bmatrix} 1+2\omega\eta & -\omega\eta & 0 & \cdots & 0 \\ -\omega\eta & 1+2\omega\eta & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1+2\omega\eta & -\omega\eta \\ 0 & \cdots & 0 & -\omega\eta & 1+2\omega\eta \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M-1,j} \\ U_{M,j} \end{bmatrix}}_{U_{j+1}} = \\
& \underbrace{\begin{bmatrix} -(1+2\omega\eta) & \omega\eta & 0 & \cdots & 0 \\ \omega\eta & -(1+2\omega\eta) & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -(1+2\omega\eta) & \omega\eta \\ 0 & \cdots & 0 & \omega\eta & -(1+2\omega\eta) \end{bmatrix}}_B \underbrace{\begin{bmatrix} U_{2,j-1} \\ U_{3,j-1} \\ \vdots \\ U_{M-1,j-1} \\ U_{M,j-1} \end{bmatrix}}_{U_{j-1}} + \\
& \underbrace{\begin{bmatrix} 2-2\omega+4\omega\eta & \omega-2\omega\eta & 0 & \cdots & 0 \\ \omega-2\omega\eta & 2-2\omega+4\omega\eta & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2-2\omega+4\omega\eta & \omega-2\omega\eta \\ 0 & \cdots & 0 & \omega-2\omega\eta & 2-2\omega+4\omega\eta \end{bmatrix}}_C \underbrace{\begin{bmatrix} U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{M-1,j} \\ U_{M,j} \end{bmatrix}}_{U_j} + \\
& \underbrace{\begin{bmatrix} \omega\eta U_{1,j-1} + (2\omega\eta - 1)U_{1,j} + \omega\eta U_{1,j+1} \\ 0 \\ \vdots \\ 0 \\ \omega\eta U_{M+1,j-1} + (2\omega\eta - 1)U_{M+1,j} + \omega\eta U_{M+1,j+1} \end{bmatrix}}_b \\
& \quad \downarrow
\end{aligned}$$

$$AU_{j+1} = BU_{j-1} + CU_j + b \quad (5.41)$$

onde A , B , C e D são matrizes tridiagonais de ordem $M - 1$. Portanto, para cada nível de tempo, precisaremos resolver o sistema 5.41 para obter aproximações em uma malha de interesse.

Vamos agora ao problema da obtenção das aproximações iniciais em $t = jk$. Podemos contornar essa situação usando $u_t(x, 0)$, que é um dado do problema que estamos trabalhando. Siga o desenvolvimento abaixo.

Aproximando $u_t(x, 0)$, nos pontos da malha no eixo x , temos por diferenças centradas em $i - 1$, i e $i + 1$ que:

$$(dut)_{i-1} = \frac{U_{i-1,j-1} - U_{i-1,j+1}}{2k} \Rightarrow U_{i-1,j-1} = 2k(dut)_{i-1} + U_{i-1,j+1} \quad (5.42)$$

$$(dut)_i = \frac{U_{i,j-1} - U_{i,j+1}}{2k} \Rightarrow U_{i,j-1} = 2k(dut)_i + U_{i,j+1} \quad (5.43)$$

$$(dut)_{i+1} = \frac{U_{i+1,j-1} - U_{i+1,j+1}}{2k} \Rightarrow U_{i+1,j-1} = 2k(dut)_{i+1} + U_{i+1,j+1} \quad (5.44)$$

Substituindo 5.42, 5.43 e 5.44 em 5.40, obteremos:

$$\begin{aligned} & -2\omega\eta U_{i-1,j+1} - 2\omega(1-2\eta)U_{i,j+1} - 2\omega\eta U_{i+1,j+1} = \\ & \alpha\omega\eta(dut)_{i-1} + \alpha\omega(1-2\eta)(dut)_i + \alpha\omega\eta(dut)_{i+1} \\ & + (2\omega\eta - 1)U_{i-1,j} + [2 - 2\omega(1-2\eta)]U_{i,j} + (2\omega\eta - 1)U_{i+1,j} \end{aligned} \quad (5.45)$$

onde $\alpha = 2k$.

Assim sendo, poderemos calcular as aproximações iniciais em $t = k$ aplicando 5.45 em cada um dos pontos desse nível. Fazendo isso, teremos que resolver o sistema linear 5.41 com algumas alterações, como segue abaixo:

$$2AU_{j+1} = \alpha Bdut + CU_j + b' \quad (5.46)$$

onde dut é o vetor

$$dut = \begin{bmatrix} dut_2 \\ dut_3 \\ \vdots \\ dut_{M-1} \\ dut_M \end{bmatrix}$$

com as derivadas de $u_t(x, 0)$ nos pontos da malha em $[0, L]$, para $i = 2$ a M , e b'

$$b' = \begin{bmatrix} \alpha\omega\eta dut_1 + (2\omega\eta - 1)U_{1,j} + 2\omega\eta U_{1,j+1} \\ 0 \\ \vdots \\ 0 \\ \alpha\omega\eta dut_{M+1} + (2\omega\eta - 1)U_{M+1,j} + 2\omega\eta U_{M+1,j+1} \end{bmatrix}$$

Observe que o sistema 5.46 envolve apenas dois níveis, onde os dados necessários para a sua resolução são conhecidos.

Com o que foi desenvolvido, vemos que, informados os dados do problema, e definida uma malha em um domínio $[0, L] \times [0, T]$, com M partições em $[0, L]$ e N partições em $[0, T]$, teremos que resolver

- O sistema 5.46 para calcular as aproximações do primeiro nível, e a seguir
- O sistema 5.41, com j variando de 3 até N , para calcular as aproximações nos níveis seguintes.

Estabilidade

A análise da estabilidade, por Von Neumann, leva a:

$$e^\lambda - 2 \left(1 - \frac{2\sigma^2 \sin \frac{\beta k}{2}}{1 + 4\eta\sigma^2 \sin \frac{\beta k}{2}} \right) e^\lambda + 1 = 0$$

Com alguns cálculos é possível demonstrar que teremos métodos:

$$\begin{array}{ll} \text{Incondicionalmente estáveis se} & \eta \geq \frac{1}{4} \text{ e } \rho < \infty \\ \text{Condicionalmente estáveis se} & 0 < \eta < \frac{1}{4} \text{ e } 0 < \rho < \frac{1}{1-4\eta} \end{array}$$

Algoritmo para a Equação da Onda

1) Início

Entrada de dados. Entre com

$a \rightarrow$ Coeficiente de u_{xx} .

$\eta \rightarrow$ Peso

$L \rightarrow$ Limite à direita do intervalo $[0, L]$.

$T \rightarrow$ Limite superior do intervalo $[0, T]$.

$M \rightarrow$ Número de partições no intervalo $[0, L]$.

$N \rightarrow$ Número de partições no intervalo $[0, T]$.

$fi(x) \rightarrow$ Condição inicial no intervalo $[0, L]$.

$dft(t) \rightarrow$ Condição inicial, derivada em relação a t , no intervalo $[0, L]$.

$fe(t) \rightarrow$ Condição de fronteira à esquerda.

$fd(t) \rightarrow$ Condição de fronteira à direita.

$$2) \text{ Cálculo dos espaçamentos } h \text{ e } k \Rightarrow \left\{ \begin{array}{l} h = \frac{L}{M} \\ k = \frac{T}{N} \end{array} \right.$$

3) Cálculo das condições iniciais e de fronteira

Para $i = 1$ a $N + 1$ faça

$$t = (i - 1)k$$

$$U(1, i) = fe(t)$$

$$U(M + 1, i) = fd(t)$$

Fim para

Para $i = 1$ a $M + 1$ faça

$$x = (i - 1)h$$

$$t = (i - 1)k$$

$$U(i, 1) = fi(x)$$

$$dut(i) = dft(t)$$

Fim para

4) Geração da matriz A , B e C dos coeficientes

$$\text{Faça } \omega = \frac{a^2 h^2}{k^2}$$

Para $i = 1$ a $M - 1$ faça

$$A(i, i) = -\omega (1 - 2\eta)$$

$$B(i, i) = -A(i, i)$$

$$C(i, i) = 2 - 2\omega + 4\omega\eta$$

Fim para

Para $i = 1$ a $M - 2$ faça

$$A(i + 1, i) = -\omega\eta$$

$$A(i, i + 1) = A(i + 1, i)$$

$$B(i+1, i) = -A(i+1, i)$$

$$B(i, i+1) = -A(i+1, i)$$

$$C(i+1, i) = 2\omega\eta - 1$$

$$C(i, i+1) = C(i+1, i)$$

Fim para

5) Cálculo do primeiro nível de tempo.

$$\text{Faça } b'(1) = 2k\omega\eta dut(1) + (2\omega\eta - 1)U(1, 1) + 2\omega\eta U_{1,2}$$

$$\text{Faça } b'(M+1) = 2k\omega\eta dut(1) + (2\omega\eta - 1)U(M+1, 1) + 2\omega\eta U_{M+1,2}$$

Resolva o sistema linear $2AU_2 = \alpha Bdut + CU_1 + b'$ algum método numérico

7) Cálculo das aproximações.

$$\text{Faça } cont = 0$$

Para $j = 3$ a N faça

$$\text{Resolva o sistema linear } AU_{j+1} = BU_{j-1} + CU_j + b$$

$$U(2 \text{ a } M, j) = U_{j+1}$$

Fim para

8) Fim do algoritmo

Capítulo 6

MATLAB

O MATLAB, abreviação de *Matrix Laboratory*, é um software que possui uma linguagem de programação para computação técnica e científica, cuja primeira versão, escrita na Universidade do Novo México e Stanford no final da década de 70, era destinada a cursos de teoria matricial, álgebra linear e análise numérica. Foi desenvolvido a partir dos pacotes de sub-rotinas LINPACK e EISPACK, de manipulação matricial, implementados em FORTRAN. Os autores tinham, como objetivo, encontrar uma maneira através do qual os estudantes pudessem utilizar esses pacotes sem que houvesse necessidade dos mesmos desenvolverem programas em FORTRAN.

A análise numérica é a área da matemática que mais está relacionada ao uso de computadores. Assim sendo, o MATLAB é uma ferramenta poderosa para essa área, pois fornece recursos de processamento numérico e visualização gráfica de fácil emprego. Possui uma grande quantidade de funções que otimizam bastante a execução e análise dos resultados gerados por métodos numéricos.

Segue abaixo alguns dos principais recursos do MATLAB, com os respectivos comentários:

- **Linguagem de programação:** Um programa feito no MATLAB é uma sequência de comandos que, quando executado, são interpretados linha após linha. Não há compilação e se houver algum erro, em uma das linhas dessa sequência, ele simplesmente para acusando erro.
- **Operações com matrizes e vetores:** Quando se trata de análise numérica, quase tudo o que abrange essa área acaba envolvendo matrizes e vetores. O elemento de dados básico do MATLAB é uma matriz que não requer dimensionamento, e as operações aritméticas são realizadas sem que haja necessidade de programar sub-rotinas para

esse fim. Assim sendo, por exemplo, dada duas matrizes A e B , de mesma dimensão, para somar basta digitar $A+B$.

- Resolução de sistemas lineares: Não há necessidade de implementar sub-rotinas para a resolução de sistemas lineares $Ax = b$. Basta digitar $x = inv(A) * b$ para se ter a solução do sistema. Esse recurso foi muito utilizado, já que muitos dos métodos implementados exigiam a resolução de sistemas.
- Matrizes de banda: Como foi visto nos capítulos anteriores, alguns métodos numéricos envolviam matrizes de banda. No MATLAB essas matrizes são geradas em uma única linha do programa, tendo fornecido previamente os vetores que irão formar as diagonais.
- Geração de gráficos bi e tridimensionais: O MATLAB permite a geração de gráficos configuráveis em duas ou três dimensões.

Essas facilidades otimizam bastante a implementação de métodos numéricos, tendo como consequência, a redução dos códigos dos programas, o que não acontece quando se utiliza, por exemplo, FORTRAN ou C.

A interface gráfica é um fator importante quando se desenvolve um software para uso público. É através dele que o usuário comunica-se com os programas. O MATLAB fornece esse ferramental, o qual foi utilizado no corrente trabalho, permitindo o desenvolvimento de janelas gráficas para entrada de dados e visualização de resultados.

Assim sendo, os recursos apresentados pelo MATLAB foram de grande valia para o fim ao qual se destina esse trabalho. Suas aplicações, no que tange ao ensino, não se esgotam apenas no cálculo numérico, mas também as outras mais diversas áreas da matemática.

Capítulo 7

Conclusão

O conjunto de programas, resultante desse projeto, apresenta uma maneira bem prática e fácil de trabalhar, no que tange ao ensino, com métodos numéricos para EDPs.

A resolução numérica de EDPs não é uma coisa fácil, e em geral a implementação de métodos atingem graus elevados de complexidade. O MATLAB, nesse meio, entra como um grande apoio para a implementação de métodos numéricos, devido a vasta gama de recursos que ele oferece para tal fim. Em comparação com outros softwares, o projeto que foi aqui desenvolvido, se feito em C ou PASCAL ou FORTRAN, teria consumido bem mais tempo do que o que foi gasto.

O MATLAB mostrou ser um software de grande valia para o desenvolvimento de suportes ao ensino. Os recursos apresentados pelo mesmo, funções já prontas direcionadas para cálculo numérico, faz com que a atenção, durante a implementação, fique voltada para a essência dos programas. Devido a isso, um professor pode desenvolver em pouco tempo, com o referido software, programas, para serem aplicados no ensino, além do cálculo numérico, das mais diversas áreas da matemática.

Com relação ao tema que o trabalho aborda, métodos de resolução numérica de EDPs por diferenças finitas, a parte gráfica do MATLAB, no que concerne a análise dos resultados, mostrou ser de grande valia para a visualização dos resultados teóricos, por exemplo, sobre estabilidade.

O desenvolvimento de projetos, semelhantes a esse, usando o MATLAB, para outras áreas da análise numérica, tais como métodos de interpolação, métodos para cálculo de raízes, métodos para cálculo de autovalores, etc, pode ser realizado. Isso poderia ser feito de modo análogo ao que foi feito nesse programa, com uma interface padrão para entrada e visualização de resultados. Num sentido maior, poderia se desenvolver um conjunto de programas, com fins de ensino, abrangendo todo o conteúdo de análise numérica.

Apêndice A

Manual para Utilização do Software

O conjunto de programas, que formam o software desse trabalho, foi desenvolvido com o MATLAB na versão 4.2c1 sobre a plataforma do Windows95, e pode ser rodado em qualquer versão do tipo 4.x. Não funciona na versão 5.0. Segue o processo de instalação, utilização dos programas, exemplos e algumas observações sobre o funcionamento.

A.1 Instalação

Crie um sub-diretório, nomeando-o com no máximo 4 letras, e copie para dentro do mesmo os programas do disquete. Copie também para dentro desse sub-diretório o WORDPAD.EXE. Feito isso, entre na janela de comandos do MATLAB e passe agora para a próxima seção, onde será explicado o procedimento para utilização dos programas.

A.2 Como utilizar os programas

1. Entre na janela de comandos do MATLAB, em seguida mude de diretório digitando no prompt dele digite o caminho do sub-diretório para onde foi copiado os programas e tecle Enter. Em seguida digite *ip* e tecle Enter. Essa sequência de comandos fará aparecer a interface principal dos programas. Por exemplo, supondo que o sub-diretório tenha o nome **prog** e esteja no drive C, então deverá digitar

```
cd c:\prog
```

e teclar Enter. Em seguida digite *ip* e tecle Enter.

2. Clique com o mouse, na barra de menus, em ARQUIVOS e a seguir em EQUAÇÕES. Isso fará aparecer um submenu com os três tipos de EDPs.
3. Clicando em um dos 3 tipos de equações, aparecerá um submenu com os respectivos métodos. Escolha um método e clique em cima. Isso fará aparecer uma janela de entrada de dados, relativo ao método escolhido.
4. Para preencher cada uma das lacunas em branco, posicione o cursor com o mouse dentro das mesmas. Preencha as lacunas com todos os dados corretamente, não deixando faltar nenhum, atentando para os seguintes cuidados:
 - Os passos devem ser números inteiros.
 - a , L , pa , σ e η devem ser números maiores que zero.
 - Os intervalos $[a, b]$ com $a < b$.
 - As funções trigonométricas devem ser dadas em inglês. Segue abaixo uma tabela com as funções elementares:

<i>Funções Matemáticas Elementares</i>			
$\text{sqrt}(x)$	Raíz quadrada	$\sin(x)$	Seno
$\exp(x)$	Exponencial	$\sinh(x)$	Seno hiperbólico
$\log(x)$	Logarítmo natural	$\text{asin}(x)$	Arco seno
$\log10(x)$	Logarítmo na base 10	$\text{asinh}(x)$	Arco seno hiperbólico
$\cos(x)$	Cosseno	$\tan(x)$	Tangente
$\cosh(x)$	Cosseno hiperbólico	$\text{atan}(x)$	Arco tangente
$\text{acos}(x)$	Arco cosseno	$\tanh(x)$	Tangente hiperbólico
$\text{acosh}(x)$	Arco cosseno hiperbólico	$\text{atanh}(x)$	Arco tangente hiperbólico

- Multiplicação é dada por $*$ e divisão por $/$.

Para exemplificar, suponhamos que você queira entrar com a função

$$\frac{\cos(0.5t + \sqrt{x})}{\sqrt{te^x - \log(x^2 \sin(0.3xt))}}$$

Na lacuna da entrada de dados escreva

$$\cos(0.5 * t + \text{sqrt}(x)) / \text{sqrt}(t * \exp(x) - \log(x^2 * \sin(0.3 * x * t)))$$

5. Aperte o botão EXECUTAR e espere a execução dos cálculos, até que apareça uma tabela com as aproximações impressas no WORDPAD. Se desejar cancelar a entrada de dados do método escolhido, aperte o botão CANCELAR. Se o programa não rodar, um dos seguintes erros podem ter acontecido:

- Falta de algum dado.
- Algum dado pode ter sido fornecido incorretamente, como o número de passos com forma decimal por exemplo.
- Erro de sintaxe na escrita de alguma função.

Pode ser que o programa rode mesmo tendo fornecido algum dado errado, como por exemplo, um intervalo $[le, ld]$ com $le > ld$. Isso acarretará no cálculo de aproximações imprecisas. De qualquer forma, se houver acontecido algum erro, feche a janela de entrada de dados, chame-a novamente e repita o processo de entrada.

6. Após analisar as aproximações, feche o WORDPAD. Não deixe o wordpad aberto, pois isso poderá travar o programa. Fechando o WORDPAD, entrará na interface de plotagem dos gráficos relativo ao método rodado, já com o gráfico das aproximações plotado. Se houver fornecido a solução analítica, poderá plotar, além das aproximações, os gráficos da solução analítica e de outros tipos de gráficos apresentados no sub-menu. Para isso, basta clicar em ARQUIVOS na barra de menus, e a seguir em GRÁFICOS e escolher o tipo de gráfico desejado.
7. As equações parabólicas em 2D envolvem três variáveis, portanto o gráfico das aproximações está no $\mathbb{R}^4$, logo não é possível de ser visualizado. Para contornar essa situação, o programa desenvolvido permite a visualização das superfícies de nível. Assim sendo, para visualizar as aproximações, basta fornecer o número do estágio, ou nível de tempo, para plotar o gráfico da superfície de nível referente ao método escolhido. Assim sendo, quando rodar um método para tais equações, ao entrar na interface de visualização dos gráficos, na barra de menus clique ARQUIVOS, e a seguir em GRÁFICOS e depois em SUPERFÍCIE DE NÍVEL. Isso fará abrir uma caixa de diálogo onde deverá ser fornecido o nível de tempo, o qual deverá ser um número inteiro. Feito isso, aperte GRÁFICO para visualizar a superfície de nível. Se houver conhecimento da solução analítica, poderá também escolher o tipo de gráfico, como o da solução analítica ou do erro ou de todas as anteriores simultaneamente.

8. Para voltar à interface principal, clique em ARQUIVOS e depois em FECHAR.
9. Para voltar ao MATLAB, clique em ARQUIVOS e depois em FECHAR. Isso pode também ser feito apertando o botão X no canto superior direito da janela.

Segue, na próxima seção, a apresentação de dois exemplos sobre todo procedimento para utilização dos programas, desde a entrada de dados até a visualização dos gráficos.

A.3 Exemplos

Primeiramente, entre na janela de comandos do MATLAB. Consideremos que os programas estejam no seguinte sub-diretório: `c:\user\sandro\prog`. Assim sendo, teremos que ir para esse sub-diretório para chamar a interface principal do programa. Para tanto, primeiro

- digitamos no prompt `cd c:\user\sandro\prog` e teclamos **Enter**;
- em seguida digitamos `ip` e novamente teclamos **Enter**, fazendo assim com a interface principal seja chamada, aparecendo na tela.

Observe as figuras abaixo ilustrando o procedimento:

Figura A.1: Janela de comandos do MATLAB

Ilustração:

Figura A.2: Interface principal

Exemplo 1: Considere o problema abaixo, envolvendo uma equação parabólica, no caso a Equação do Calor, e vamos resolvê-lo numericamente utilizando o método implícito.

$$\left\{ \begin{array}{ll} u_t = au_{xx} & a > 0, \text{ constante}, (x, t) \in [0, L] \times [0, T] \\ u(x, 0) = 50\text{sen}(\pi x/L) & x \in [0, L] \text{ condição inicial} \\ u(0, t) = 0 & t \in [0, \infty) \text{ condição de fronteira em } x = 0 \\ u(L, t) = 0 & t \in [0, \infty) \text{ condição de fronteira em } x = L \\ u(x, t) = 50\text{sen}(\pi x/L)/e^{\pi^2 at/L^2} & \text{solução analítica} \end{array} \right.$$

Considere 10 passos no eixo x e 20 passos no eixo t . Tomemos $a = 0.023$, $L = 1$ e $\text{sigma} = 0.3$. Com isso, temos os seguintes dados de entrada, seguido pelo processo de entrada de dados no programa:

$$\begin{array}{lll} a = 0.023 & L = 1 & \text{sigma} = 0.3 \\ \text{Número de passos no eixo } x = 10 & & \\ \text{Número de passos no eixo } t = 20 & & \\ u(x, 0) = 50\text{sen}(\pi x/L) & u(0, t) = 0 & u(L, t) = 0 \\ u(x, t) = 50\text{sen}(\pi x/L)/e^{\pi^2 at/L^2} & & \text{solução analítica} \end{array}$$

2. Em cada lacuna em branco entre com o dado correspondente. Como temos a solução analítica, aperte SIM, para aparecer uma janela onde irá entrar com a mesma. Aperte CONTINUE. Aperte o botão EXECUTAR e espere até aparecer o WORD-PAD com os resultados.

Figura A.5: Lacunas preenchidas com os dados

Figura A.6: Janela de entrada da solução analítica

Continuação

Tab. apu WordPad

Correr New

TABELA DE APROXIMAÇÕES

Equações Parabólicas

Método Implícito

a= 0.023000
 L= 1.000000
 h= 0.100000
 k= 0.130435
 sigma= 0.300000
 Número de passos em x= 10
 Número de passos em t= 20
 Tempo total de cálculo gasto para cpu (em segundos)= 0.44

x= 0.000000

t	Aproximação	Solução	Erro
0.000000	0.000000e+000	0.000000e+000	0.000000e+000
0.130435	0.000000e+000	0.000000e+000	0.000000e+000
0.260870	0.000000e+000	0.000000e+000	0.000000e+000
0.391304	0.000000e+000	0.000000e+000	0.000000e+000
0.521739	0.000000e+000	0.000000e+000	0.000000e+000
0.652174	0.000000e+000	0.000000e+000	0.000000e+000
0.782609	0.000000e+000	0.000000e+000	0.000000e+000
0.913043	0.000000e+000	0.000000e+000	0.000000e+000

Figura A.7: Aproximações impressas no Wordpad

3. Feche agora o WORDPAD. Com isso visualizará uma interface com o gráfico das aproximações já plotado, conforme ilustra a figura abaixo:

Figura A.8: Método implícito - Gráfico das aproximações

4. Como temos conhecimento da solução analítica, podemos visualizar o gráfico da solução analítica, dos erros, e de todos os anteriores simultaneamente. Para isso, clique em ARQUIVOS → GRÁFICOS e escolha o tipo desejado. As figuras abaixo ilustram esse procedimento:

Figura A.9: Opções de gráficos

Figura A.10: Método implícito - Gráfico da solução analítica

Continuação

Figura A.11: Método implícito - Gráfico dos erros

5. Para sair da interface de plotagem de gráficos e voltar à interface principal, clique em ARQUIVOS e depois em FECHAR. Isso também pode ser feito apertando o botão X no canto superior direito da janela.

Exemplo 2: Considere agora o problema abaixo, envolvendo uma equação parabólica, no caso a Equação do Calor em 2D, e vamos resolvê-lo numericamente utilizando o método das direções alternadas.

$$\left\{ \begin{array}{ll} u_t = u_{xx} + u_{yy} & (x, y) \in [0, L] \times [0, L] \quad L > 0 \\ u(x, y, 0) = \cos(x) + \cos(y) & (x, y) \in [0, L] \times [0, L] \\ u(x, 0, t) = e^{-t} \cos(x) & (x, t) \in [0, L] \times \mathbb{R}_+ \\ u(x, L, t) = e^{-t} (\cos(x) + \cos(L)) & (x, t) \in [0, L] \times \mathbb{R}_+ \\ u(0, y, t) = e^{-t} \cos(y) & (y, t) \in [0, L] \times \mathbb{R}_+ \\ u(L, y, t) = e^{-t} (\cos(L) + \cos(y)) & (y, t) \in [0, L] \times \mathbb{R}_+ \end{array} \right.$$

A solução analítica desse problema é dada por:

$$u(x, t) = e^{-t} (\cos(x) + \cos(y))$$

Considere 10 passos no eixo x e y e 2 passos no eixo t . Tomemos $L = 1$ e $T = 0.2$. Com isso, temos os seguintes dados de entrada:

$$L = 1$$

$$T = 0.2$$

$$\text{Número de passos no eixo } x \text{ e } y = 10$$

$$\text{Número de passos no eixo } t = 2$$

$$u(x, y, 0) = \cos(x) + \cos(y)$$

$$u(x, 0, t) = e^{-t}\cos(x)$$

$$u(x, L, t) = e^{-t}(\cos(x) + \cos(L))$$

$$u(0, y, t) = e^{-t}\cos(y)$$

$$u(L, y, t) = e^{-t}(\cos(L) + \cos(y))$$

$$u(x, y, t) = e^{-t}(\cos(x) + \cos(y))$$

A entrada de dados desse problema é feito de maneira análoga ao anterior, diferindo apenas quanto ao procedimento para visualização das superfícies de nível. Portanto resumiremos mais os passos:

1. Clique a seguinte sequência: ARQUIVOS $\rightarrow$ EQUAÇÕES $\rightarrow$ PARABÓLICAS $\rightarrow$ EQUAÇÕES EM 2 DIMENSÕES $\rightarrow$ MÉTODO DAS DIREÇÕES ALTERNADAS. Em seguida entre com os dados, aperte o botão EXECUTAR e espere a tabela de aproximações.
2. Após analisar os resultados, feche o WORDPAD.
3. Para visualizar as superfícies de nível, clique em ARQUIVOS $\rightarrow$ GRÁFICOS $\rightarrow$ SUPERFÍCIE DE NÍVEL. Com isso será aberto uma caixa de diálogo para entrar com o nível de tempo. Informe o nível desejado e aperte o botão GRÁFICO. O nível de tempo deve ser um número inteiro. Como, nesse exemplo, temos conhecimento da solução analítica, é possível plotar também as superfícies de nível da solução analítica, dos erros e de todos esses simultaneamente. Para isso, abra a caixa de diálogo, entre o nível de tempo, escolha o tipo de gráfico desejado e aperte o botão GRÁFICO. As figuras abaixo ilustram esses procedimentos:

Continuação

Figura A.12: Método das direções alternadas - Plotagem das superfícies de nível

Figura A.13: Método das direções alternadas - opções de gráficos

Continuação

Figura A.14: Método das direções alternadas - solução analítica - nível de tempo 2

4. Para voltar à interface principal, feche a janela como no exemplo anterior.

A.4 Observações

- Caso esqueça de apertar SIM, o programa rodará como se não houvesse solução analítica.
- A resolução dos métodos envolvendo malhas de dimensões muito altas é diretamente ligado a capacidade da memória do computador que estiver sendo utilizado. Devido a isso, os programas poderão travar se a dimensão da malha ultrapassar a capacidade do computador. Se isso ocorrer, desligue o MATLAB, ligue-o novamente e chame o programa.
- Se ao abrir uma janela de entrada de dados, ela desaparecer por ter clicado fora da janela, aperte a seguinte sequência de teclas: ALT + TAB. Isso fará aparecer uma caixa de diálogo mostrando as janelas abertas no Windows. Assim que abrir essa caixa, continue com a tecla ALT pressionada e vá apertando a tecla TAB até que o marcador fique em cima do ícone referente a janela de entrada de dados. Localizado, solte a tecla ALT. Dessa forma, a janela de entrada de dados voltará a aparecer.

Bibliografia

- [1] Ames, W.F. *Numerical Methods for Partial Differential Equations*, Academic Press, New York, 3ed., 1992
- [2] Bernstein, D.L. *Existence Theorems in Partial Differential Equations*, Princeton University Press, N.J., 1950
- [3] Blum, E.K. *Numerical Analysis and Computation: theory and practice*, Addison Wesley, 1972
- [4] Courant, R.; Friedrichs, K.O. *Supersonic Flow and Shock Waves*, Wiley, (interscience), New York, 1948
- [5] Courant, R.; Hilbert D. *Methods of Mathematical Physics*, vol 2, Wiley, (interscience), New York, 1948
- [6] Cunha, C. *Métodos Numéricos para as Engenharias e Ciências Aplicadas*, Editora da UNICAMP, 1962
- [7] Cuminato, J.A.; Meneguette, M. *Discretização de Equações Diferenciais Parciais: Técnicas de Diferenças Finitas*, XIX CNMAC-Set. 1996
- [8] Duchateau, P.; Zachmann, D.W. *Theory and Problems of Partial Differential Equations*, McGrawhill, 1986
- [9] Garabedian, P.R. *Partial Differential Equations*, Wiley, New York, 1964
- [10] Gordon, P. *SIAM J. Appl. Math*, v.13, p.667, 1965
- [11] Hildebrand, F.B. *Introduction to Numerical Analysis*, Dover Publications, Inc., 1974
- [12] Kato, T. *Perturbation Theory for Linear Operators*, Springer-Verlag, 1984

- [13] Keller, H.B. *A new difference scheme for parabolic problems in Numerical Solution of Partial Differential Equations*, v.2(Ed. B. Hubbard), Academic Press, New York, pp. 36-50, 1971.
- [14] Lapidus, L.; Pinder G.F. *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley and Sons, 1982
- [15] Lees, M.A. Linear Three-level Difference Scheme for Quasi-linear Parabolic Equations *Math. Comp.*, 20, 516-522, 1966
- [16] Mitchell, A.R.; Griffiths, D.F. *The finite Difference Method in Partial Differential Equations*, John Wiley and Sons, New York, 1985
- [17] Saul'yev, V.K.; *Integration of Equations of Parabolic Type by the Method of Nets*, Pergamon Press, New York, 1964
- [18] Schiesser, W.E.; *The Numerical Methods of Lines: Integration of Partial Differential Equations*, Academic Press, 1991
- [19] Smith, G.D. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, OUP, 1978
- [20] The Mathworks Inc., *MATLAB: Versão do Estudante - Guia do Usuário*, Makron Books, 1997
- [21] The Mathworks Inc., *MATLAB: Reference Guide*, The Mathworks Inc., 1992
- [22] The Mathworks Inc., *MATLAB: Building a Graphical User Interface*, The Mathworks Inc., 1992