5
Implementação

Dos aspectos relacionados à implementação, o mais relevante é a construção do problema grosso. Neste capítulo mostraremos como transformar as variáveis relativas à velocidade e à pressão de um problema fino naquelas de um problema grosso e vice-versa.

Analisemos como transformar, na figura (5.1), os elementos κ_G nos elementos κ_F. Nas variáveis relativas à velocidade, queremos transformar U_1, U_2, U_3 em u_1, u_2, u_3, u_4, u_5.

Observemos que a variável U_2 é a que está na interseção entre cada elemento grosso.

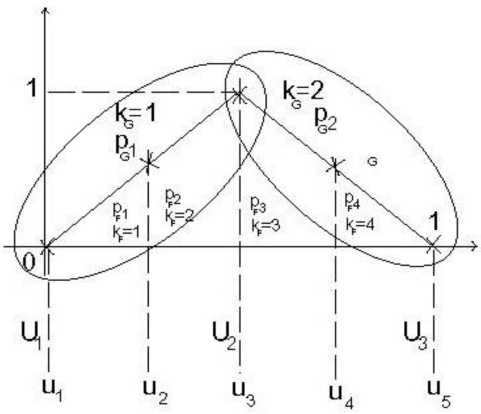

Figura 5.1: Domínio Simplificado.

Chamemos de $\Gamma \in \Re^{4x3}$ a matriz relativa à transformação linear $\Gamma : \Re^3 \to \Re^4$ que duplica o vetor nos pontos onde há interseção. Γ possui a forma:

$$\Gamma = \begin{bmatrix} 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{array}{l} \\ \Rightarrow \text{Posição de fronteira} \\ \Rightarrow \text{Posição de fronteira} \\ \\ \end{array}$$

Chamemos de $\Sigma \in \Re^{5x4}$ a matriz relativa à transformação linear $\Sigma : \Re^4 \rightarrow \Re^5$ que multiplica os pontos de interseção por $\frac{1}{c}$, onde c é o número de domínios que dividem a variável. Neste exemplo $c = 2$, logo:

$$\Sigma = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & 1/2 & 1/2 & \\ & & 1 & \\ & & & 1 \end{bmatrix} \Rightarrow \text{Posições de fronteira.}$$

Assim, para U_2 vetor na fronteira, temos:

$$\Gamma \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_2 \\ U_3 \end{bmatrix}, \quad \Sigma \begin{bmatrix} u_1 \\ 1/2(u_1+u_2) \\ u_2 \\ u_2 \\ 1/2(u_2+u_3) \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ 1/2(u_1+u_2) \\ 1/2u_2 + 1/2u_2 = u_2 \\ 1/2(u_2+u_3) \\ u_3 \end{bmatrix}.$$

Sejam $(MI_H^h)_1, (MI_H^h)_2 \in \Re^{3\times 2}$ as matrizes relativas às transformações lineares $(MI_H^h)_1, (MI_H^h)_2 : \Re^2 \rightarrow \Re^3$ que extendem $(\kappa_G)_{1,2}$ em $(\kappa_F)_{1,2,3}$ e $(\kappa_G)_{2,3}$ em $(\kappa_F)_{3,4,5}$, respectivamente.

A matriz geral relativa à transformação linear $MgI_H^h : \Re^3 \rightarrow \Re^5$ que extendem κ_G em κ_F fica então sendo:

$$MIg_H^h = \Sigma \begin{bmatrix} (MI_H^h)_1 & \\ & (MI_H^h)_2 \end{bmatrix} \Gamma. \tag{5-1}$$

A extensão da variável relativa à pressão é feita da seguinte forma:

Para $(\kappa_G)_1$ faço $(p_F)_1 = (p_F)_2 = (p_G)_1$ e para $(\kappa_G)_2$ faço $(p_F)_3 = (p_F)_4 = (p_G)_2$. A matriz que representa esta transformação será chamada MgJ_H^h.

Nas próximas seções faremos uma generalização destas transformações.

5.1
Extensão das Variáveis Relativas à Velocidade

Vejamos como realizar a transformação linear das quatro variáveis que compões o retângulo grosso nas respectivas variáveis do problemas fino.

Sejam:
$\bar{m}, \bar{n}$ o número de divisões de um elemento grosso na horizontal e na vertical, respectivamente.
q o número de variáveis relativas à velocidade de um elemento grosso, com $q = 2\bar{m}\bar{n} + \bar{n} + \bar{m}$.

Observemos que, para estes valores, e considerando m, n como mostrados na figura (2.1), o número de variáveis relativas à velocidade do problema grosso será de $q_G = 2n'm' + n' + m'$, com $m' = m/\bar{m}, n' = n/\bar{n}$.

Veremos em seguida uma representação local dos nós onde as quatro variáveis relativas à velocidade na malha grossa serão transformadas nas q variáveis relativas à velocidade nas malhas finas.

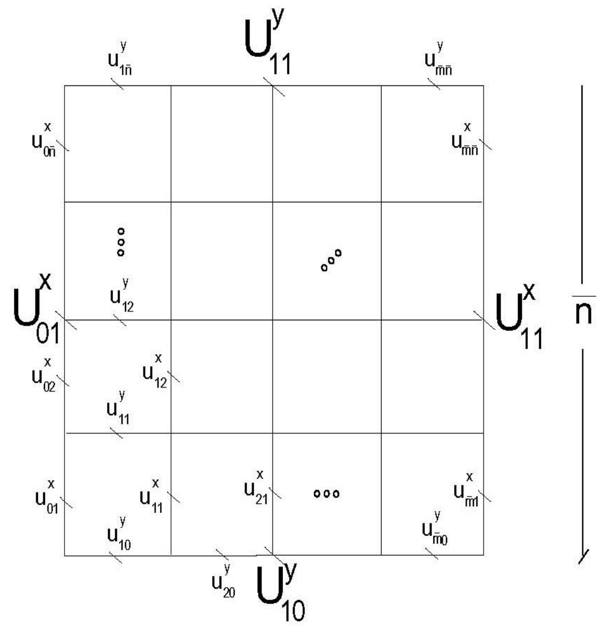

Figura 5.2: Transformação $U \Leftrightarrow u$.

Seja I_H^h a seguinte transformação:

$$I_H^h : \Re^4 \longrightarrow \Re^q$$
$$(U_{01}^x, U_{11}^x, U_{10}^y, U_{11}^y) \mapsto (u_{01}^x, u_{11}^x, ..., u_{\bar{m}1}^x, u_{02}^x, u_{12}^x, ...u_{\bar{m}2}^x, ..., u_{\bar{m}\bar{n}}^x,$$
$$u_{10}^y, u_{11}^y, ..., u_{1\bar{n}}^y, u_{20}^y, u_{21}^y, ...u_{2m}^y, ..., u_{\bar{m}\bar{n}}^y).$$

Sejam $X \in \Re^{(\bar{m}+1)\times 4}, Y \in \Re^{(\bar{n}+1)\times 4}$ matrizes da forma:

$$X = \begin{bmatrix} (\bar{m}-0)/\bar{m} & 0/\bar{m} & 0 & 0 \\ (\bar{m}-1)/\bar{m} & 1/\bar{m} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ (\bar{m}-\bar{m})/\bar{m} & \bar{m}/\bar{m} & 0 & 0 \end{bmatrix},$$

$$Y = \begin{bmatrix} 0 & 0 & (\bar{n}-0)/\bar{n} & 0/\bar{n} \\ 0 & 0 & (\bar{n}-1)/\bar{n} & 1/\bar{n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & (\bar{n}-\bar{n})/\bar{n} & \bar{n}/\bar{n} \end{bmatrix}.$$

A forma matricial MI_H^h que representa I_H^h fica então sendo:

$$\begin{bmatrix} u_{01}^x \\ u_{11}^x \\ \vdots \\ u_{\bar{m}\bar{n}}^x \\ \vdots \\ u_{10}^y \\ \vdots \\ u_{\bar{m}\bar{n}}^y \end{bmatrix} = MI_H^h \begin{bmatrix} U_{01}^x \\ U_{11}^x \\ U_{10}^y \\ U_{11}^y \end{bmatrix} = \begin{bmatrix} X \\ X \\ X \\ \vdots \\ X \\ \vdots \\ Y \\ Y \\ \vdots \\ Y \end{bmatrix} \begin{bmatrix} U_{01}^x \\ U_{11}^x \\ U_{10}^y \\ U_{11}^y \end{bmatrix}.$$

Observemos que X repete-se $\bar{n}$ vezes e Y $\bar{m}$ vezes.

A matriz relativa à transformação linear $MgI_H^h : \Re^{2n'm'+n'+m'} \longrightarrow \Re^{2mn+m+n}$ que extendem κ_G em κ_F, pode então ser escrito como:

$$MIg_H^h = \Sigma \begin{bmatrix} (MI_H^h)_1 & & & \\ & (MI_H^h)_2 & & \\ & & \ddots & \\ & & & (MI_H^h)_{mn} \end{bmatrix} \Gamma. \tag{5-2}$$

Para construir a matriz A do problema grosso, será usada uma rotina de acumulação a ser explicada em (5.3.1).

5.2
Transformação das Variáveis Relativas à Pressão

Chamemos $\bar{m}, \bar{n}$ o número de divisões de um elemento grosso na horizontal e na vertical, respectivamente. Para transformarmos um problema grosso em um problema fino nas variáveis relativas à pressão basta observarmos em qual elemento grosso cada elemento fino está contido, e então fazemos $p_F = p_G$, como mostrado na figura (5.3).

A transformação representada por MgJ_H^h generalizada é a seguinte: para cada elemento grosso $(\kappa_G)_i$, faço $(p_F)_j = (p_G)_i \forall j$ com $(\kappa_F)_j \subset (\kappa_G)_i$.

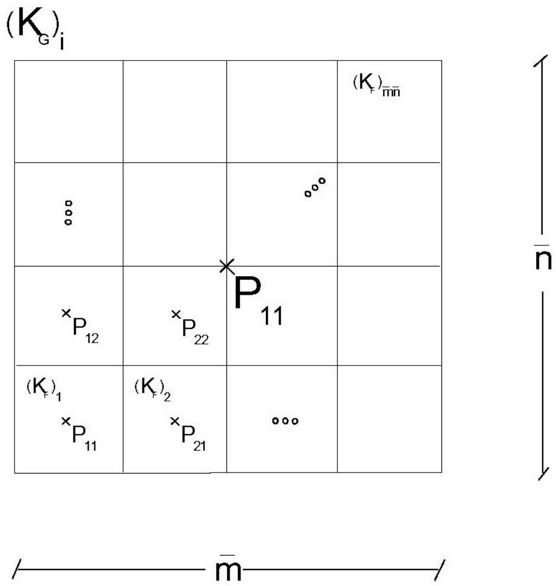

Figura 5.3: Elemento Grosso.

Uma vez que tenho MgI_H^h e MgJ_H^h, automaticamente também tenho MgI_h^H e MgJ_h^H, pois basta fazer: $MgI_h^H = (MgI_H^h)^T$ e $MgJ_h^H = (MgJ_H^h)^T$.

5.3
Construção de C_G

Para construir o operador C_G^{-1} usado em (4-2), primeiro nos preocuparemos em encontrar A_G, B_G e B_G^T, pois $C_G = B_G A_G^{-1} B_G^T$.

5.3.1 Construção da matriz A_G

Construímos a matriz A_G utilizando um tipo de interpolação nos elementos finos para permitir que o programa construa a matriz A referente ao problema grosso ao mesmo tempo em que constrói a respectiva matriz fina. Para isto consideramos o valor da variável λ como uma média entre os valores dos vários elementos finos que compõe o elemento grosso. A função linear Φ do elemento grosso é então uma combinação linear das funções ϕ do elemento fino, permitindo assim que a integral feita em cada elemento grosso para a formação da matriz A_g seja uma combinação linear das integrais usadas em cada elemento fino multiplicado pelo respectivo coeficiente de transformação da matriz $MIg_h^H = (MIg_H^h)^T$. Vejamos um exemplo.

Na figura (5.4), temos um elemento grosso e os seus respectivos elementos finos, para $h_x = h_y$ e $\bar{n} = \bar{m} = 4$.

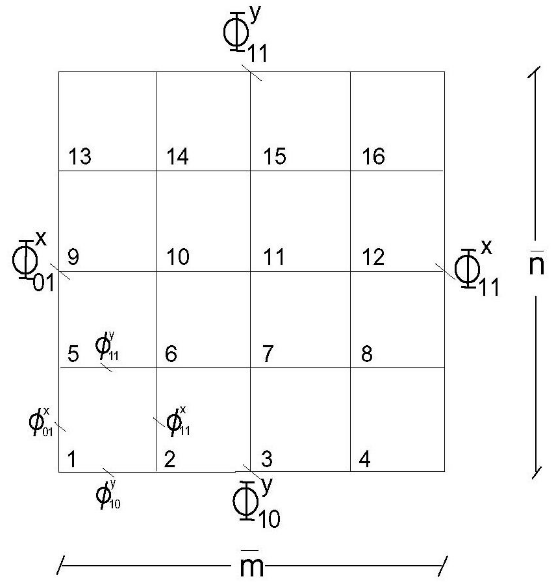

Figura 5.4: Funções Grossas.

A matriz MIg_h^H para este exemplo tem o seguinte formato:

$$MIg_h^H = \begin{array}{cc} & \begin{array}{ccccc} & {[10]} & & {[11]} & \\ & \downarrow & & \downarrow & \end{array} \\ \begin{array}{c} \\ {[10]} \longrightarrow \\ \\ {[11]} \longrightarrow \\ \\ \end{array} & \begin{pmatrix} & \vdots & & \vdots & \\ & 1 & \cdots & 3/4 & \cdots \\ & \vdots & & \vdots & \\ \cdots & 0 & \cdots & 1/4 & \cdots \\ & \vdots & & \vdots & \end{pmatrix} \end{array}.$$

Reparemos que dentro do elemento fino de número 1, duas das funções Φ são tais que:

$$\begin{cases} \Phi_{10}^y = 1\phi_{10}^y + \frac{3}{4}\phi_{11}^y \\ \Phi_{11}^y = 0\phi_{01}^y + \frac{1}{4}\phi_{11}^y \end{cases}.$$

Assim, ao construir a matriz Ag para os índices relativos à [10] e [11] faremos:

$$\int\int_\Omega (\lambda^{-1}\Phi_{11}^y\Phi_{10}^y)dxdy = 1.\frac{1}{4}\int\int_\Omega(\lambda^{-1}\phi_{10}^y\phi_{11}^y)dxdy + \frac{3}{4}.\frac{1}{4}\int\int_\Omega(\lambda^{-1}\phi_{11}^y\phi_{11}^y)dxdy, \tag{5-3}$$

o que, em termos de matrizes, equivale simplesmente a acumular a combinação dos valores já calculados para a matriz A_F multiplicando pelos respectivos coeficientes da matriz de tranformação da velocidade fina para a grossa.

Durante a execução do programa, as matrizes A_G são construídas, ao mesmo tempo que as finas, a partir de uma função de acumulação que, para cada entrada da matriz fina, acumula este ponto nas quatro entradas respectivas da matriz grossa.

De maneira mais geral temos que, para cada um dos elementos que eu percorrer no problema fino, acumulo a contribuição deste no elemento grosso, como mostrado abaixo:

$$\int_{\kappa_G} \lambda\Phi_{a_G}\Phi_{b_G} = \sum_{\kappa_F\in\kappa_G}\left(\int_{\kappa_F}\lambda(MIg_h^H)\phi_{a_F}(MIg_h^H)\phi_{b_F},\ \forall a_F, b_F \text{ relacionado com } \kappa_F.\right), \\ \forall a_G, b_G \text{ relacionado com } \kappa_G. \tag{5-4}$$

Sendo A_H a matriz grossa, A_h a matriz fina, (x, y) as entradas da matriz A_h, $x, y \in \Re^2$ e $\Leftarrow$ indicando acumulação, o pseudo-código da função de acumulação é:

Algorithm 4 Acumulador

Dados de entrada:

Par (x_1, x_2) relativo ao problema fino, com $x_1, x_2 \in \Re^2$.

Funções e operações usadas:

$Mind(x_1, x_2)$: retorna a entrada (X_1, X_2) que representa a função do elemento grosso que contêm a função do elemento fino representado por (x_1, x_2).

MIg_h^H: explicada em (5-2).

function acumula **{**
1 : $A_H(X_1, X_1) \Leftarrow MIg_h^H(x_1, X_1).MIg_h^H(x_2, X_1).A_h(x_1, x_2);$
2 : $A_H(X_1, X_2) \Leftarrow MIg_h^H(x_1, X_1).MIg_h^H(x_2, X_2).A_h(x_1, x_2);$
3 : $A_H(X_2, X_1) \Leftarrow MIg_h^H(x_1, X_2).MIg_h^H(x_2, X_1).A_h(x_1, x_2);$
4 : $A_H(X_2, X_2) \Leftarrow MIg_h^H(x_1, X_2).MIg_h^H(x_2, X_2).A_h(x_1, x_2);$
}

Na primeira parcela da soma em (5-3) temos $x_1 = [11]$, $x_2 = [10]$, $X_1 = [10], X_2 = [11]$ equivalente à linha 3 do algoritmo 4. Ver figura (5.4).

5.4 Estrutura de Dados

Sendo as matrizes A e B esparsas são então tratadas como operadores e são armazenados apenas os valores que podem ser diferentes de zero. Observemos que a matriz B possui quatro entradas (linhas) potencialmente não nulas de acordo com as equações (2-9). Assim $B^x \in \Re^{(m+1)n \times mn}$ e $B^y \in \Re^{(n+1)m \times mn}$ mas foram implementadas como $B^x \in \Re^{(m+1)n \times 2}$ e $B^y \in \Re^{(n+1)m \times 2}$ reduzindo bastante a memória armazenada. Observando as equações (2-8), vemos que a matriz A possui três entradas (linhas) potencialmente não nulas. Assim $A^x \in \Re^{(m+1)n \times (2nm+m+n)}$ e $A^y \in \Re^{(n+1)m \times (2nm+m+n)}$ foram implementadas como $A^x = \Re^{(m+1)n \times 3}$ e $A^y = \Re^{(n+1)m \times 3}$, reduzindo a quantidade de memória armazenada também para o operador A. Ao construir a rotina que multiplica A e B por vetores levamos em conta a topologia e associamos cada posição do vetor à respectiva posição do operador. Notemos que para λ do tipo matricial a matriz A precisaria de cinco entradas, isto é, cinco linhas, sendo então $A^x = \Re^{(m+1)n \times 5}$ e $A^y = \Re^{(n+1)m \times 5}$.