4
Solução Numérica

4.1
Resolvendo o Sistema Linear

Resolveremos o sistema $(2-7)$ pelo método de Usawa acelerado por Gradiente Conjugado (ver [15, pp. 202]). Primeiro eliminamos a variável u e depois resolvemos o problema em p :

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix},$$

$$Au + B^T p = f \qquad \longrightarrow \qquad u = A^{-1}(f - B^T p).$$

que substituindo na segunda equação $Bu = g$ temos:

$$B(A^{-1}(f - B^T p)) = g \qquad \longrightarrow \qquad BA^{-1}B^T p = -g + BA^{-1}f$$

ou simplesmente:

$$Cp = d \tag{4-1}$$

para $C = BA^{-1}B^T \qquad$ e $\qquad d = -g + BA^{-1}f.$

Observemos que C é simétrica, pois $C^T = (B^T)^T(A^{-1})^T B^T = BA^{-1}B^T$.
Sabendo que A é positiva, temos que:
$(BA^{-1}B^T x, x) = (A^{-1}B^T x, B^T x) = (A^{-1}y, y) = (Ay, y) \qquad \forall y \neq 0$
logo $\qquad (Cx, x) \geq 0 \qquad \forall x \neq 0$, pois $(Ay, y) > 0$, se $y \neq 0$ e $(Ay, y) = 0$ se $y = 0$.

Esta propriedade e mais $g \in$ Imagem(B) garantem que a matriz C possa ser resolvida pelo método Gradiente Conjugado.

Assim, para encontrar a solução do sistema basta resolver a equação $Cp = d$ usando o algoritmo Gradiente Conjugado. É importante observarmos que o sistema com a matriz A deverá ser resolvido várias vezes por estarmos usando um método iterativo. Usaremos o próprio Gradiente Conjugado para realizar esta tarefa.

4.2 Gradiente Conjugado e Precondicionadores

O Gradiente Conjugado(GC) é um método iterativo efetivo para resolver sistemas lineares esparsos formados por matrizes simétricas positivas semi-definidas. O método consiste em gerar seguidas aproximações para a solução com direções ótimas de busca para atualização da iteração e do resíduo. Embora o tamanho desta sequência possa se tornar grande, apenas um pequeno número de vetores são necessariamente mantidos na memória. Em cada iteração do método, dois produtos internos são realizados para que se calculem dois escalares definidos de maneira que a sequência possa obedecer certas condições de ortogonalidade. Em sistemas positivos definidos estas condições implicam que a distância para a verdadeira solução está sendo minimizada na norma $\langle Cx, x\rangle^{\frac{1}{2}}$. Para detalhes, ver [11].

O maior empecilho em relação ao uso de métodos iterativos pela indústria é a sua baixa robustez em relação aos métodos diretos, o que tem sido resolvido com o uso de precondicionadores. O GC é um método que possui boa convergência quando os problemas relacionados ao escoamento de água e óleo em meios porosos são pequenos. Entretanto para problemas grandes torna-se imprescindível o uso de precondicionadores para que se possa acelerar e aumentar a eficiência do algoritmo.

O número de iterações para a convergência do método GC é proporcional à raiz do número espectral $\kappa_2(C) = \lambda_{\max}/\lambda_{\min}$, respectivos máximo e mínimo autovalores de C. Um precondicionador procura diminuir o número espectral da matriz C. Observemos que o precondicionador perfeito tornaria $C = I$ matriz identidade e $\kappa_2(C) = 1$. Detalhes em [9]. Abaixo é dada uma breve explicação do precondicionamento:

Ao querermos resolver um problema do tipo $Cp = d$, podemos buscar uma matriz E tal que $EC \simeq I$ e assim teremos $ECp = Ed$, $p \simeq Ed$. É importante que E continue mantendo as propriedades de simetria e positividade.

No Gradiente Conjugado isto é feito da seguinte forma:

Algorithm 2 PreCondicionador

Require: tol: precisão que definirá
a aproximação desejada

Ensure: $p \Leftarrow p + z$

1: $p \Leftarrow 0$
2: **while** $r < tol$ **do**
3: $r \Leftarrow d - Cp$
4: Encontrar uma boa aproximação para $C^{-1}r = z$, usando no lugar de C a matriz
precondicionada ECtal que $(EC)^{-1}r = E^{-1}z$.
5: $p \Leftarrow p + z$
6: **end while**

Isto é, procuramos aproximar a solução para que haja uma convergência mais rápida a partir de um problema mais fácil de ser resolvido.

O Algoritmo Gradiente Conjugado usado no programa para resolver $Cp = d$ é o seguinte:

Algorithm 3 Gradiente Conjugado

Dados de entrada:

p_0 : pressão inicial.

tol : prescisão.

maxiter: número máximo de iterações.

Dado de saída:

p : valor de $p = C^{-1}d$.

Variáveis:

Vetores d, p, p_0, r, z, g.

Ponto flutuantes $bnrm2, tol, erro, beta, rho, rho\_1, alpha$.

Inteiros $i, k,$ maxiter.

Matriz Simétrica Positiva Definida Esparsa C.

Funções e operações usadas:

x,y vetores de tamanho n, então $x * y = \sum_{i=1}^{n} x_i * y_i$.

x escalar, y vetor de tamanho n, então $(x * y)_i = x * y_i$.

$norma(x) = \sqrt{(x * x)}$.

```
function GC {
bnrm2 ← norm(d);
if (bnrm2 = 0.0) then bnrm2 ← 1.0;
r ← d − Cp_0;
erro ← norma(r)/bnrm2;
if (erro < tol) then {
    retorna p_0;
    break;
}
for i ← 1 to itmax or erro < tol do{
    z ← pc(r); ⇒ Aqui será feito o precondicionamento!
    rho ← r ∗ z;
    if (k > 1)then{
        beta ← rho/rho_1;
        g ← z + beta ∗ g;
     }
    else g ← z;
    q ← C ∗ g;
    alpha ← rho/(g ∗ q);
    p ← p + alpha ∗ g
    r ← r − alpha ∗ q;
    erro ← norm(r)/bnrm2;
    rho_1 ← rho;
return p.
}
```


Para o nosso problema, decidimos usar como precondicionador a solução aproximada do mesmo problema feito em uma malha mais grossa e a solução aproximada em um subdomínio do problema fino, usando o método conhecido como Método Multinível Aditivo de Schwarz (ver [12]).

4.3 O Método Multinível Aditivo de Schwarz

Em dois passos mostraremos o aspecto multinível e o aspecto aditivo do método de Schwarz.

4.3.1 Multinível

Descreveremos nesta seção um precondicionamento usado para o Gradiente Conjugado e que consiste em tentar aproximar a solução final através da solução de um problema semelhante resolvido em uma malha mais grossa. Será usado X_F para indicar matriz na malha fina e X_G para grossa.

Sabemos que o Gradiente Conjugado inicia com um chute inicial na solução. Seja p_0 este valor. Queremos encontrar p_1 tal que $p_0 + p_1 = p$, com $Cp = d$ como em (4-1).

Temos que $C(p_0+p_1) = d$, $p_1 = C^{-1}(d - Cp_0)$. Como precondicionador usaremos a solução de $C_G^{-1} r_G$ como uma aproximação para $C_F^{-1} r_F$, onde r_F é o resíduo do Gradiente Conjugado em uma certa iteração e r_G é descrito abaixo.

Seja a malha com $m_F = 9, n_F = 8, m_G = 3$ e $n_G = 2$ como na figura 4.1, queremos achar $pc(r_F)$, que será o precondicionador indicado na linha 11 do algoritmo 3.

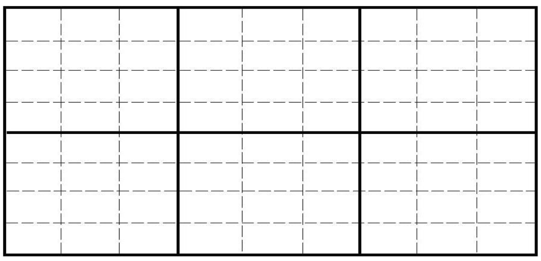

Figura 4.1: Malha Grossa/Fina.

Seja R^T a matriz que representa a interpolação linear da malha grossa para a malha fina. O operador R é conhecido como matriz de restrição e R^T como matriz de extensão. Seja C_G a forma discreta de C_F para a malha grossa, isto é $C_G = RC_F R^T$. Uma correção usando a malha grossa pode ser feita da seguinte maneira:

$$pc(r_F) = R^T C_G^{-1} R(r_F). \tag{4-2}$$

Isto é, restringimos o resíduo para a malha grossa, resolvemo-lo e então o extendemos para a malha fina novamente.

Observemos que a matriz $R^T C_G^{-1} R$ possui um núcleo bastante grande se comparado ao do problema fino, e muitos dos componentes de $d_F - C_F p_F$ que estão no núcleo de $R^T C_G^{-1} R$ não serão encontrados, em particular os que possuem uma freqüência alta. Assim 4-2 estará resolvendo as freqüencias mais baixas (para detalhes ver [8]).

4.3.2
Aditivo

Vejamos a matriz C apenas reparando em m blocos de matrizes C_i como mostrado abaixo (os valores omitidos podem ser quaisquer):

$$C = \begin{bmatrix} C_1 & & & & \\ & C_2 & & & \\ & & \ddots & & \\ & & & C_{m-1} & \\ & & & & C_m \end{bmatrix},$$

onde estamos somente interessado nos valores mostrados. O Método Aditivo de Schwarz consiste em aproximar a matriz C^{-1} por C_S mostrado abaixo:

$$C_S = \begin{bmatrix} C_1^{-1} & 0 & 0 & 0 & 0 \\ 0 & C_2^{-1} & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & C_{m-1}^{-1} & 0 \\ 0 & 0 & 0 & 0 & C_m^{-1} \end{bmatrix}.$$

O precondicionador fica então sendo $pc(r_F) = \sum_{i=1}^{m} C_i^{-1}(r_F^i)$ onde $r_F^i = [r_F^1, r_F^2, ..., r_F^m]^T$, cada r_F^i com o mesmo número de linhas de ser respectivo C_i^{-1}, de maneira a fazer sentido cada operação $C_i^{-1}(r_F^i)$.

No atual programa, C_i são blocos de tamanho um, isto é, representam a diagonal da matriz C e, por simplificação, usamos apenas uma aproximação, pois seria muito custoso do ponto de vista computacional calcular seu valor.

Para encontrar uma aproximação da diagonal de A_G simplificamos os cálculos feitos em (3-1) e (3-2) fazendo

$\phi^x_{s,t}\phi^x_{i,j} = 1$ nos termos $\lambda^{-1}\phi^x_{s,t}\phi^x_{i,j}$ e $\lambda^{-1}\phi^y_{s,t}\phi^y_{i,j}$ da integral que gera A_G, isto é,

$\int_\Omega(\lambda^{-1}\phi^x_{i,j}\phi^x_{k,l})dxdy$ e $\int_\Omega(\lambda^{-1}\phi^y_{i,j}\phi^y_{k,l})dxdy$ e,

consideramos a função λ^{-1} constante avaliada no centro de cada retângulo grosso κ. No método CG isto equivale a multiplicar r_F por λ^{-1}, isto é, $pc(r_F) = \sum\limits_{i=1}^{m}\lambda_i^{-1}(r_F^i)$ sendo $\lambda_i = \lambda$ avaliada no centro do elemento i.

Observemos que a matriz do precondicionador reflete o comportamento local do problema, o que leva este precondicionador a capturar as componentes de $r_F = d_F - C_F p_F$ com freqüências altas (para detalhes observar em [8]).

4.3.3
Multinível Aditivo

Para a resolução do sistema em que estamos interessados, usamos as duas idéias mostradas acima em conjunto e, assim, tivemos a oportunidade de capturar tanto as componentes de alta quanto de baixa freqüência do resíduo para, dessa maneira, obtermos melhor convergência.

Sendo $B_G = R^T C_G^{-1} R$ o precondicionador relativo à solução do problema grosso, e $B_F = \sum\limits_{i=1}^{m} C_i^{-1}$ relativo ao fino, então o precondicionador em dois passos é:

$r_F = B_F(r_F) + B_G(r_F)$, isto é,
$r_F = \sum\limits_{i=1}^{m}\lambda_i^{-1} r_F + R^T C_G^{-1} R r_F$.