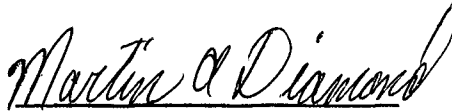
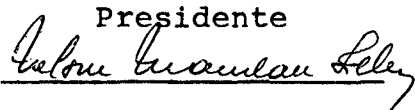
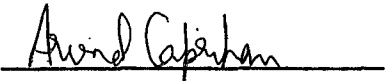


"ESTUDO DE UM MÉTODO DE REDUÇÃO CÍCLICA PARA A RESOLUÇÃO  
DE EQUAÇÕES DIFERENCIAIS PARCIAIS ELÍTICAS"

Angela Maria Fonseca Motta

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

  
Presidente  
  


RIO DE JANEIRO  
ESTADO DA GUANABARA - BRASIL  
AGOSTO DE 1974

A meus pais  
Brenno e Maria Helena

AGRADECIMENTOS

Ao Chefe do Programa de Pós-Graduação de Engenharia de Sistemas e Computação da Universidade Federal do Rio de Janeiro - Prof. Nelson Maculan e ao Prof. Martin Diamond pela orientação e apoio durante a execução deste trabalho.

Aos meus amigos Helio Korenblum , Anna Maria Faria e Silva, Gerhard Schwarz e Dulcinea L. V. Ferreira pelo incentivo e dedicação nas horas mais difíceis.

A todos meus colegas do SERPRO que permitiram com sua compreensão, que me dedicasse ao desenvolvimento final deste trabalho.

RESUMO

Um estudo de um método de redução cíclica para resolver sistemas de equações lineares, mostra as vantagens e desvantagens de sua aplicação com relação a estabilidade. A vantagem deste método se encontra na resolução de equações diferenciais parciais elípticas numa região não retangular com condição de fronteira de Dirichlet.

ABSTRACT

A study of a cyclic reduction method for solving systems of linear equations shows the advantage and disadvantage of its application with relation to stability. The advantage of this method is founded, on the resolution of partial elliptic differential equations on a nonrectangular region with Dirichlet boundary condition.

ÍNDICE

DEDICATÓRIA .....	i
AGRADECIMENTOS .....	ii
RESUMO .....	iii
ABSTRACT .....	iv
CAPÍTULO I - Introdução .....	1
CAPÍTULO II - Modificação do Método de CORF .....	3
CAPÍTULO III - Modificação do algoritmo de BUNEMAN .....	17
CAPÍTULO IV - Instabilidade .....	29
BIBLIOGRAFIA .....	32

## C A P Í T U L O I

### I N T R O D U Ç Ã O

Neste trabalho estudamos um método, desenvolvido em [2], para resolver sistemas de equações lineares. O método é uma modificação de CORF [1]. Ambos são métodos de redução cíclica relativos a sistemas cuja matriz dos coeficientes é bloco-tridiagonal. A diferença entre eles é que o de CORF faz reduções que resultam num sistema envolvendo as incógnitas do meio do sistema inicial; e a modificação do CORF faz reduções que resultam num sistema envolvendo as incógnitas do princípio do sistema inicial. Esta diferença nos dá duas vantagens com relação a quantidade de cálculos. A primeira vantagem está na "back substitution", e a segunda está em problemas onde é preciso achar somente as primeiras incógnitas de um sistema.

Ocorrem problemas de instabilidade com o método CORF e sua modificação. A instabilidade no de CORF ocorre no cálculo do lado direito dos sistemas resultantes das reduções. Na modificação de CORF existem instabilidades no cálculo do lado direito dos sistemas resultantes das reduções e na "back substitution".

A instabilidade do CORF deixa de existir quando aplicamos o algoritmo de BUNEMAN [1]. Neste trabalho desenvolvemos um algoritmo, semelhante ao de BUNEMAN, que retira a instabilidade que

ocorre quando calculamos o lado direito dos sistemas resultantes das reduções. Provamos a estabilidade deste algoritmo e a instabilidade da "back substitution".

Com o nosso algoritmo, a modificação será útil quando apenas as incógnitas do início do sistema são desejadas. Um exemplo é discutido no § 2.2 do capítulo 2.

Em resumo, no § 2.1 do capítulo 2, introduziremos o algoritmo de CORF [1] modificado em [2]. No § 2.2 aplicamos este algoritmo a solução de uma equação diferencial parcial elítica numa região não retangular com condição de fronteira de Dirichlet. No § 3.1 do capítulo 3 fazemos uma modificação do algoritmo de BUNEMAN, que é desenvolvido em [1]. E, no § 3.2 estudamos a estabilidade deste algoritmo no cálculo da primeira componente  $x_0$  do vetor  $x$  de um sistema matricial. No capítulo 4, examinamos a instabilidade do algoritmo apresentado no § 2.1.



C A P Í T U L O    I I

2.1 - Modificação do método de CORF

Com uma modificação no método de CORF [1] que apresentaremos a seguir, encontramos uma forma de que a única incógnita de um sistema a não ser eliminada seja  $x_0$ . Com essa modificação obtém-se uma grande vantagem na "back substitution" para o cálculo das incógnitas eliminadas. Infelizmente a "back substitution" é instável. A prova desta instabilidade é um dos resultados deste trabalho. Começaremos com a apresentação do método e no Capítulo 4 provaremos a instabilidade.

Seja

$$(1) \quad Mx = y$$

um sistema, onde

$$M = \begin{bmatrix} A & I & & & \\ & I & A & I & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & I \\ & & & & I & A \end{bmatrix}$$

é uma matriz real, simétrica e bloco-tridiagonal, com dimensão de bloco  $q = 2^{k+1}$ , onde  $k$  é um número inteiro positivo.

Considerando a estrutura de bloco da matriz  $M$  podemos escrever os vetores  $x$  e  $y$  da seguinte forma:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_{q-1} \end{bmatrix} \quad e \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \cdot \\ \cdot \\ y_{q-1} \end{bmatrix}$$

sendo,

$$x_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \cdot \\ \cdot \\ x_{pi} \end{bmatrix} \quad e \quad y_i = \begin{bmatrix} y_{1i} \\ y_{2i} \\ \cdot \\ \cdot \\ y_{pi} \end{bmatrix}$$

onde  $q$  é a dimensão de bloco de  $M$  e  $p$  a dimensão de  $A$ .

O sistema (1) poderá então ser escrito da seguinte forma:

$$\begin{aligned} A x_0 + x_1 &= y_0 \\ (2) \quad x_{j-1} + A x_j + x_{j+1} &= y_j, \quad j = 1, 2, \dots, q-2 \\ x_{q-2} + A x_{q-1} &= y_{q-1} \end{aligned}$$

Aplicando em (1) o procedimento descrito nos ítems abaixo, podemos eliminar algumas incógnitas do sistema:

i - Multiplica-se a primeira equação por  $(-A)$  e soma-se com a segunda. Obtemos assim  $(I^2 - A^2)x_0 + x_2 = y_1 - A y_0$



$$= \begin{bmatrix} y_1 - A y_0 - x_0 \\ y_3 + y_1 - A y_2 \\ \vdots \\ y_{q-3} + y_{q-1} - A y_{q-2} \end{bmatrix}$$

e

$$(4) \quad \begin{bmatrix} A & 0 & 0 & & \\ 0 & A & 0 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & 0 \\ & & & 0 & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{q-1} \end{bmatrix} =$$

$$= \begin{bmatrix} y_1 - (x_0 + x_2) \\ y_3 - (x_2 + x_4) \\ \vdots \\ y_{q-1} - x_{q-2} \end{bmatrix}$$

O sistema (4) é chamado sistema de incógnitas eliminadas. O número de incógnitas de (3) é  $2^k$  logo, reduzimos a metade o número de incógnitas de (1). Após repetidas reduções, obtemos um único bloco com  $x_0$  sendo a única incógnita.



Após  $k$  reduções obtemos o sistema

$$\begin{bmatrix} A^{(k)} & I \\ I & A^{(k)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_{2^k} \end{bmatrix} = \begin{bmatrix} y_0^{(k)} + q^{(k)} x_0 \\ y_{2^k}^{(k)} \end{bmatrix}$$

Fazendo mais uma redução, obtemos um sistema com a matriz correspondente de dimensão de bloco igual a um:

$$(I^2 - (A^{(k)})^2)x_0 = -A^{(k)}y_0^{(k)} - A^{(k)}q^{(k)}x_0 + y_{2^k}^{(k)}$$

Adicionando  $I^2x_0$  a ambos os lados da equação acima e usando as fórmulas de recursão (5), (6) e (8), temos:

$$(9) \quad A^{(k+1)}x_0 = y_0^{(k+1)} + q^{(k+1)}x_0$$

ou

$$(10) \quad (A^{(k+1)} - q^{(k+1)})x_0 = y_0^{(k+1)}$$

Como no algoritmo de CORF, o polinômio  $(A^{(k+1)} - q^{(k+1)})$  é fatorado de forma a possibilitar a resolução de (10). Temos de [1] que:

$$(11) \quad A^{(r)} = P_{2^r}(A, I) = P_{2^r}(A) = - \prod_{j=1}^{2^r} (A + 2 \cos \phi_j^{(r)} I)$$

onde  $\phi_j^{(r)} = \frac{(2j-1)\pi}{2^{r+1}}$ .

E de [2], temos:

$$(12) \quad \bar{A}^{(r)} = A^{(r)} - q^{(r)} = \bar{P}_{2^r}(A) = P_{2^r}(A) - q^{(r)}$$

$$= - \prod_{j=1}^{2^r} (A + 2 \cos \theta_j^{(r)} I)$$

onde

$$\theta_j^{(r)} = \frac{j\pi}{2^r + 1}$$

Podemos assim, determinar o vetor  $x_0$  a partir de (10) pela fatorização de  $(A^{(k+1)} - q^{(k+1)})$  de acordo com (12), resolvendo um sistema para cada fator. Como no parágrafo 3 de [1], defina

$$G_j^{(k+1)} = A + 2 \cos \theta_j^{(k+1)} I, \quad j = 1, \dots, 2^{k+1}$$

Para resolver (10), façamos

$$z_0 = -y_0^{(k+1)}$$

e recursivamente calculamos  $z_j$  em

$$G_j^{(k+1)} z_j = z_{j-1}, \quad j = 1, \dots, 2^{k+1}$$

Portanto,

$$z_{2^{k+1}} = x_0$$

Logo, conhecendo  $x_0$ , podemos calcular as incógnitas restantes  $x_1, \dots, x_{2^{k+1}-1}$  usando o sistema  $Mx = y$  da seguinte forma:

$$x_1 = y_0 - Ax_0$$

e

$$x_j = y_{j-1} - Ax_{j-1} - x_{j-2} \quad , \quad j = 2, \dots, 2^{k+1} - 1.$$

O algoritmo apresentado neste capítulo requer menos cálculos que o algoritmo de CORF. Para calcular  $x_0$  é preciso resolver  $2^{k+1}$  sistemas e, para calcular cada uma das  $2^{k+1} - 1$  incógnitas restantes é necessário resolver apenas um sistema. O número total de sistemas a serem resolvidos é igual a  $(4 \times 2^k - 1)$  que, para  $k > 2$  é um número menor que  $(k+1)2^k$  obtido no algoritmo de CORF. Mais uma vantagem deste algoritmo modificado é que a metade dos sistemas a serem resolvidos são sistemas cujas matrizes dos coeficientes são iguais a identidade (I).

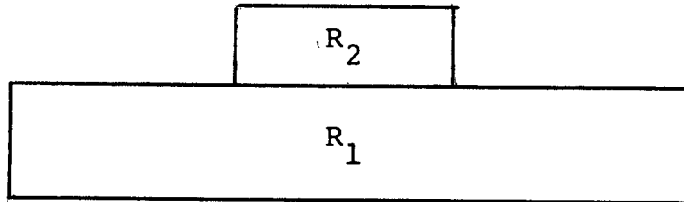
A grande desvantagem deste algoritmo está no fato dele ser instável e o algoritmo de CORF ser estável. O problema vem da determinação das incógnitas  $x_1, \dots, x_{2^{k+1}-1}$ . A determinação de  $x_0$  usando os cálculos deste capítulo é também instável. Esse problema é idêntico a instabilidade do algoritmo de CORF [§10 de [1]]. No capítulo 3, modificamos o algoritmo de BUNEMAN para que o cálculo de  $x_0$  seja estável. Com essa estabilidade no cálculo de  $x_0$ , o método deste capítulo pode ser usado no algoritmo do parágrafo 9 de [1], onde somente precisamos do vetor  $x_0$ . O parágrafo a seguir define o algoritmo do §9 em [1].

## 2.2 - Resolução de sistemas que surgem da discretização de uma equação diferencial parcial elíptica numa região não retangular.

Muitos problemas na prática necessitam resolver uma equa



ção diferencial parcial elíptica em uma região não retangular, como a representada abaixo



cuja condição inicial é a de Dirichlet.

No parágrafo 5 de [2] verifica-se que o algoritmo do § 2.1 nos dá uma grande vantagem quando usamos o algoritmo desenvolvido no § 9 de [1]. Pois, a quantidade de computação será bem reduzida. Descreveremos a seguir este método.

Obtemos a seguinte equação matricial quando for usado o mesmo  $\Delta x$  em toda região:

$$(13) \quad \left[ \begin{array}{c|c} G & 0 \\ \hline 0 & H \end{array} \right] \begin{array}{c} p \\ p^t \\ x^{(1)} \\ x^{(2)} \end{array} = \begin{array}{c} y^{(1)} \\ y^{(2)} \end{array}$$

onde

$$(13.1) \quad G = \begin{bmatrix} A & T & & & \\ & T & A & & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & T \\ & & & & T & A \end{bmatrix}, \quad H = \begin{bmatrix} B & S & & & \\ & S & B & S & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & S \\ & & & & S & B \end{bmatrix}$$

e

$$(13.2) \quad x^{(1)} = \begin{bmatrix} x_0^{(1)} \\ x_1^{(1)} \\ \vdots \\ x_r^{(1)} \end{bmatrix}, \quad x^{(2)} = \begin{bmatrix} x_0^{(2)} \\ x_1^{(2)} \\ \vdots \\ x_s^{(2)} \end{bmatrix}$$

Sendo  $r \times r$  e  $s \times s$ , respectivamente, as dimensões de bloco de  $G$  e  $H$ , com  $r = 2^{k_1+1} - 1$  e  $s = 2^{k_2+1} - 1$ .

Supondo que  $A$  tenha dimensão  $n_1 \times n_1$  e  $B$  tenha dimensão  $n_2 \times n_2$  concluímos que  $P$  terá dimensão  $n_1 \times n_2$ .

De (13) verificamos que

$$(14) \quad x^{(1)} = G^{-1}y^{(1)} - G^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ P \end{bmatrix} x_0^{(2)}$$

$$(15) \quad x^{(2)} = H^{-1}y^{(2)} - H^{-1} \begin{bmatrix} p^t \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_r^{(1)}$$

Defina

$$(16) \quad z^{(1)} = G^{-1}y^{(1)}, \quad z^{(2)} = H^{-1}y^{(2)}$$

$$(17) \quad w^{(1)} = G^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ P \end{bmatrix}, \quad w^{(2)} = H^{-1} \begin{bmatrix} p^t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Substituindo (16) e (17) em (14) e (15), obtemos

$$x^{(1)} = z^{(1)} - W^{(1)} x_0^{(2)}$$

e

$$x^{(2)} = z^{(2)} - W^{(2)} x_r^{(1)}$$

Particionando os vetores  $z^{(1)}$  e  $z^{(2)}$  e as matrizes  $W^{(1)}$  e  $W^{(2)}$ , teremos

$$(18) \quad x_j^{(1)} = z_j^{(1)} - W_j^{(1)} x_0^{(2)}, \quad j = 0, \dots, r$$

e

$$(19) \quad x_j^{(2)} = z_j^{(2)} - W_j^{(2)} x_r^{(1)}, \quad j = 0, \dots, s$$

Faça  $j = r$  em (18) e  $j = 0$  em (19) logo, obtemos o seguinte sistema matricial:

$$(20) \quad \begin{bmatrix} I & W_r^{(1)} \\ W_0^{(2)} & I \end{bmatrix} \begin{bmatrix} x_r^{(1)} \\ x_0^{(2)} \end{bmatrix} = \begin{bmatrix} z_r^{(1)} \\ z_0^{(2)} \end{bmatrix}$$

que pode ser reduzido a

$$(21) \quad (I - W_r^{(1)} W_0^{(2)}) x_r^{(1)} = z_r^{(1)} - W_r^{(1)} z_0^{(2)}$$

Resolvendo o sistema (21) utilizando o método de eliminação de Gauss, obtemos  $x_r^{(1)}$ . Substituindo esse valor em (20) calculamos  $x_0^{(2)}$  e as outras incógnitas restantes são então calculadas usando (14) e (15).

É interessante notar que necessitamos calcular apenas as componentes  $z_r^{(1)}$  e  $z_0^{(2)}$  dos vetores  $z^{(1)}$  e  $z^{(2)}$ , respectivamente. Da mesma forma, sô calculamos  $W_r^{(1)}$  e  $W_0^{(2)}$  das matrizes  $W^{(1)}$  e  $W^{(2)}$ , sendo que as componentes restantes nós nunca usaremos. Observemos

que usando o método desenvolvido em [1] é impossível calcular apenas  $z_r^{(1)}$ ,  $z_0^{(2)}$ ,  $W_r^{(1)}$  e  $W_0^{(2)}$ . Porém, se usarmos o algoritmo em 2.1 deste capítulo,  $z_r^{(1)}$ ,  $z_0^{(2)}$ ,  $W_r^{(1)}$  e  $W_0^{(2)}$  podem ser encontradas diretamente.

Para calcular  $W_r^{(1)}$  precisamos resolver o sistema

$$(22) \quad GW^{(1)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ P \end{bmatrix}$$

onde  $W^{(1)}$  tem dimensão de bloco  $r \times 1$  e, cada uma das matrizes  $W_0^{(1)}$ ,  $W_1^{(1)}$ , ...,  $W_r^{(1)}$  possuem dimensão  $n_1 \times n_2$ . Para que possamos resolver (22) particionaremos  $W^{(1)}$  em colunas como segue:

$$(23) \quad W^{(1)} = \left[ W_{c_1}^{(1)}, W_{c_2}^{(1)}, \dots, W_{c_{n_2}}^{(1)} \right]$$

Cada coluna  $W_{c_i}^{(1)}$  é particionada em blocos como em (13.2),

$$W_{c_i}^{(1)} = \begin{bmatrix} W_{c_{i0}}^{(1)} \\ W_{c_{i1}}^{(1)} \\ \vdots \\ W_{c_{ir}}^{(1)} \end{bmatrix}$$

onde  $W_{c_{ij}}^{(1)}$  tem dimensão  $n_1 \times 1$ .

Como queremos calcular  $W_r^{(1)}$  temos que calcular cada  $W_{c_{ij}}^{(1)}$ ,  $i = 1, \dots, n_2$ , onde

$$W_r^{(1)} = \left[ W_{c_{1r}}^{(1)}, W_{c_{2r}}^{(1)}, \dots, W_{c_{n_2r}}^{(1)} \right]$$

Faça,

$$(24) \quad \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ P \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ P_1 & P_2 & \dots & P_{n_2} \end{bmatrix}$$

onde  $P_i$ ,  $i = 1, \dots, n_2$ , é uma matriz de dimensão  $n_1 \times 1$ .

Usando (23) e (24) podemos reescrever o sistema (22) na forma

$$GW_{c_i}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ P_i \end{bmatrix}, \quad i = 1, \dots, n_2.$$

Logo ao invés de considerarmos o sistema (22), o substituímos por  $n_2$  sistemas da forma de (1). Para resolver estes sistemas aplicamos o algoritmo em 2.1 deste capítulo.

De forma análoga calcula-se  $W_0^{(2)}$ .

Em síntese, o algoritmo é o seguinte:

1º) Calcular  $z_r^{(1)}$  e  $z_0^{(2)}$  em (16)

2ºº) Calcular  $W_r^{(1)}$  e  $W_0^{(2)}$  em (17)

3º) Resolver (21) usando o método de eliminação de Gauss.

4º) Calcular  $x_0^{(2)}$  usando (20).

5º) Calcular  $x^{(1)}$  e  $x^{(2)}$  em (14) e (15).

C A P Í T U L O    I I I

3.1 - Modificação do Algoritmo de BUNEMAN

Como observamos no final do § 2.1, o cálculo de  $x_0$  é instável. A prova deste fato é semelhante a prova da instabilidade de CORF no § 10 de [1]. Foi observado também, que um algoritmo desenvolvido por BUNEMAN [1], torna o CORF estável. Neste capítulo apresentamos um algoritmo semelhante ao de BUNEMAN que torna o cálculo de  $x_0$  estável.

O problema da instabilidade aparece no cálculo do lado direito dos sistemas definidos nas reduções, isto é, os vetores  $y_j^{(r)}$ . Para  $j \neq 0$ , estes vetores são exatamente os do algoritmo de CORF. O algoritmo de BUNEMAN [1] fornece um método de "quebrar" os vetores  $x_j^{(r)}$  em duas partes,

$$y_j^{(r)} = A^{(r)} p_j^{(r)} + q_j^{(r)} ,$$

onde o cálculo de  $p_j^{(r)}$  e  $q_j^{(r)}$  é estável. O vetor  $y_j^{(r)}$  não é calculado, apenas  $p_j^{(r)}$  e  $q_j^{(r)}$  são calculados.

Desenvolvemos um algoritmo semelhante para  $y_0^{(r)}$ , usando  $\bar{A}^{(r)}$  no lugar de  $A^{(r)}$ . Assim, definiremos  $p_0^{(r)}$  e  $q_0^{(r)}$  tal que

$$(25) \quad y_0^{(r)} = \bar{A}^{(r)} p_0^{(r)} + q_0^{(r)} .$$

Como usaremos os vetores  $p_j^{(r)}$  e  $q_j^{(r)}$  do algoritmo de BUNEMAN, da

remos a definição destes vetores :

$$P_j^{(1)} = A^{-1} y_j, \quad q_j^{(1)} = y_{j-1} + y_{j+1} - 2p_j^{(1)} .$$

$$P_j^{(r+1)} = P_j^{(r)} - (A^{(r)})^{-1} \left[ P_{j-2r}^{(r)} + P_{j+2r}^{(r)} - q_j^{(r)} \right]$$

(26)

$$q_j^{(r+1)} = q_{j-2r}^{(r)} + q_{j+2r}^{(r)} - 2P_j^{(r)} .$$

Definamos

$$P_0^{(1)} = A^{-1} y_0, \quad q_0^{(1)} = y_1 - P_0^{(1)},$$

$$(27) \quad P_0^{(r+1)} = P_0^{(r)} - (\bar{A}^{(r)})^{-1} (P_{2r}^{(r)} - q_0^{(r)})$$

e

$$q_0^{(r+1)} = q_{2r}^{(r)} - P_0^{(r+1)} .$$

Provaremos agora que estes vetores satisfazem

$$y_0^{(r)} = \bar{A}^{(r)} P_0^{(r)} + q_0^{(r)} .$$

Usando o processo de indução em  $r$ , temos que para  $r = 1$

$$\begin{aligned} y_0^{(1)} &= y_1 - Ay_0 = A^{-1}y_0 - Ay_0 + y_1 - A^{-1}y_0 \\ &= (I^2 - A^2) A^{-1}y_0 + y_1 - A^{-1}y_0 \\ &= \bar{A}^{(1)} A^{-1}y_0 + y_1 - A^{-1}y_0 \\ &= \bar{A}^{(1)} P_0^{(1)} + q_0^{(1)} \end{aligned}$$

onde  $\bar{A} = I - A^{(0)} \bar{A}^{(0)} = I^2 - A^2$  .



Suponha que

$$y_0^{(r)} = \bar{A}^{(r)} p_0^{(r)} + q_0^{(r)}$$

Para  $y_0^{(r+1)}$  temos

$$\begin{aligned} y_0^{(r+1)} &= y_{2^r}^{(r)} - A^{(r)} y_0^{(r)} \\ &= A^{(r)} p_{2^r}^{(r)} + q_{2^r}^{(r)} - A^{(r)} (\bar{A}^{(r)} p_0^{(r)} + q_0^{(r)}) \\ &= -A^{(r)} \bar{A}^{(r)} \left[ p_0^{(r)} - (\bar{A}^{(r)})^{-1} (p_{2^r}^{(r)} - q_0^{(r)}) \right] + q_{2^r}^{(r)} \\ &= -A^{(r)} \bar{A}^{(r)} p_{2^r}^{(r+1)} + q_{2^r}^{(r)} \\ &= (I - A^{(r)} \bar{A}^{(r)}) p_0^{(r+1)} + q_{2^r}^{(r)} - p_0^{(r+1)} \end{aligned}$$

mas,  $\bar{A}^{(r+1)} = I - A^{(r)} \bar{A}^{(r)}$

Então

$$y_0^{(r+1)} = \bar{A}^{(r+1)} p_0^{(r+1)} + q_0^{(r+1)}$$

Para calcularmos  $p_0^{(r+1)}$  resolvemos o sistema

$$\bar{A}^{(r)} (p_0^{(r)} - p_0^{(r+1)}) = p_{2^r}^{(r)} - q_0^{(r)}$$

usando a fatorização de  $\bar{A}^{(r)}$  (12). Para calcularmos  $q_0^{(r+1)}$  utilizamos a última relação em (27).

Após  $k+1$  reduções, temos a equação

$$\bar{A}^{(k+1)} x_0 = y_0^{(k+1)}$$

Usando (25), temos

$$\bar{A}^{(k+1)} x_0 = \bar{A}^{(k+1)} p_0^{(k+1)} + q_0^{(k+1)}$$

Logo, podemos determinar o vetor  $x_0$  resolvendo a equação

$$x_0 = p_0^{(k+1)} + (\bar{A}^{(k+1)})^{-1} q_0^{(k+1)}$$

Para mostrar a estabilidade do processo descrito acima, escrevemos os  $p_0$ 's e  $q_0$ 's em função dos vetores  $x_j$ 's da seguinte forma:

Temos que

$$p_0^{(1)} = A^{-1} y_0 \quad \text{e} \quad q_0^{(1)} = y_1 - p_0^{(1)}$$

logo

$$p_0^{(1)} = x_0 + A^{-1} x_1$$

e

$$\begin{aligned} q_0^{(1)} &= x_0 + Ax_1 + x_2 - x_0 - A^{-1} x_1 \\ &= -A^{-1} (I^2 - A^2) x_1 + x_2 \\ &= -A^{-1} \bar{A}^{(1)} x_1 + x_2 \end{aligned}$$

Pelo processo de indução, provaremos que:

$$(28) \quad p_0^{(r)} = x_0 + (-1)^{r+1} s^{(r-1)} (\bar{A}^{(r-1)})^{-1} \sum_{k=1}^{2^{r-1}} x_{2k-1}$$

e

$$(29) \quad q_0^{(r)} = (-1)^r s^{(r-1)} (\bar{A}^{(r-1)})^{-1} \bar{A}^{(r)} \sum_{k=1}^{2^{r-1}} x_{2k-1} + x_{2^r}$$

onde  $s^{(r)} = (A^{(r-1)} A^{(r-2)} \dots A^{(0)})^{-1}$ ,  $r = 1, 2, \dots, k$

e  $s^{(0)} = I$

Sabemos que para  $r = 1$

$$p_0^{(1)} = x_0 + A^{-1}x_1 \quad \text{onde } A = \bar{A}^{(0)}$$

e

$$q_0^{(1)} = -A^{-1} \bar{A}^{(1)} x_1 + x_2 \quad \text{onde } I = s^{(0)}$$

Supondo válido para  $r$ , então

$$\begin{aligned} p_0^{(r+1)} &= p_0^{(r)} - (\bar{A}^{(r)})^{-1} (p_{2^r}^{(r)} - q_0^{(r)}) \\ &= x_0 + (-1)^{r+1} s^{(r-1)} (\bar{A}^{(r-1)})^{-1} \sum_{k=1}^{2^{r-1}} x_{2^{k-1}} - \\ &\quad - (\bar{A}^{(r)})^{-1} [x_{2^r} + (-1)^{r+1} s^{(r)} \sum_{k=1}^{2^{r-1}} (x_{2^{r-(2k-1)}} 2^{r+(2k-1)} + \\ &\quad + (-1)^{r+1} s^{(r-1)} (\bar{A}^{(r-1)})^{-1} \bar{A}^{(r)} \sum_{k=1}^{2^{r-1}} x_{2^{k-1}} - x_{2^r}] \\ &= x_0 + (-1)^{r+2} s^{(r)} (\bar{A}^{(r)})^{-1} \sum_{k=1}^{2^r} x_{2^{k-1}} \end{aligned}$$

e

$$\begin{aligned} q_0^{(r+1)} &= q_{2^r}^{(r)} - p_{(r+1)}^{(r)} \\ &= x_0 + (-1)^r s^{(r)} A^{(r)} \left[ \sum_{k=1}^{2^{r-1}} (x_{2^{r-(2k-1)}} + x_{2^{r+(2k-1)}}) \right] + x_{2^{r+1}} - \\ &\quad - x_0 + (-1)^{r+1} s^{(r)} (\bar{A}^{(r)})^{-1} \sum_{k=1}^{2^r} x_{2^{k-1}} = \\ &= (-1)^{r+1} s^{(r)} [-A^{(r)} + (\bar{A}^{(r)})^{-1}] \sum_{k=1}^{2^r} x_{2^{k-1}} + x_{2^{r+1}} \end{aligned}$$

$$= (-1)^{r+1} s^{(r)} (\bar{A}^{(r)})^{-1} [I - A^{(r)} \bar{A}^{(r)}] \sum_{k=1}^{2^r} x_{2k-1} + x_{2^{r+1}}$$

$$= (-1)^{r+1} s^{(r)} (\bar{A}^{(r)})^{-1} \bar{A}^{(r+1)} \sum_{k=1}^{2^r} x_{2k-1} + x_{2^{r+1}}$$

Note que para  $j = 2, \dots, q - 2$  foi usada as seguintes fórmulas do

§ 11 de [1] para calcular os  $p_j^i$ s e  $q_j^i$ s:

$$p_j^{(r)} = x_j + (-1)^{r+1} s^{(r)} \left\{ \sum_{k=1}^{2^{r-1}} (x_{j-(2k-1)} + x_{j+(2k-1)}) \right\}$$

e

$$q_j^{(r)} = x_{j-2^r} + (-1)^r s^{(r)} A^{(r)} \left\{ \sum_{k=1}^{2^{r-1}} (x_{j-(2k-1)} + x_{j+(2k-1)}) \right\} + x_{j+2^r}$$

$$\text{onde } s^{(r)} = (A^{(r-1)} \dots A^{(0)})^{-1}$$

### 3.2 - Estabilidade

Estudaremos agora a estabilidade do algoritmo de 3.1 no cálculo do vetor  $x_0$ .

Vamos escrever (43) e (44) da seguinte forma

$$p_0^{(r+1)} = x_0 + g_0^{(r)}$$

e

$$q_0^{(r+1)} = x_{2^{r+1}} - \bar{A}^{(r+1)} g_0^{(r)}$$

onde

$$g_0^{(r)} = (-1)^{r+2} s^{(r)} (\bar{A}^{(r)})^{-1} \sum_{k=1}^{2^r} x_{2k-1}$$

e

$$s^{(r)} = (A^{(r-1)} \dots A^{(0)})^{-1}$$

Então

$$(30) \quad \left\| \left\| p_0^{(r+1)} - x_0 \right\|_2 \right\| = \left\| \left\| (-1)^{r+2} s^{(r)} (\bar{A}^{(r)})^{-1} \sum_{k=1}^{2^r} x_{2k-1} \right\|_2 \right\| \leq \\ \leq \left\| \left\| s^{(r)} (\bar{A}^{(r)})^{-1} \right\|_2 \right\| \|x\|$$

e

$$(31) \quad \left\| \left\| q_0^{(r+1)} - x_{2^{r+1}} \right\|_2 \right\| = \left\| \left\| (-1)^{r+1} \bar{A}^{(r+1)} s^{(r)} (\bar{A}^{(r)})^{-1} \sum_{k=1}^{2^r} x_{2k-1} \right\|_2 \right\| \\ \leq \left\| \left\| \bar{A}^{(r+1)} (\bar{A}^{(r)})^{-1} s^{(r)} \right\|_2 \right\| \|x\| ,$$

onde  $\|x\| = \sum_{j=1}^{q-1} \|x_j\|_2$

$\|v\|_2$  é a norma euclidiana do vetor  $v$

e

$\|c\|_2$  é a norma spectral da matriz  $c$ . Quando a matriz  $c$  for simétrica,  $\|c\|_2$  será o máximo do módulo dos autovalores de  $c$ .

Para obtermos o limite de (30) e (31), precisamos dos autovalores de  $A^{(r)}$ ,  $\bar{A}^{(r)}$  e  $s^{(r)}$ . Esses autovalores podem ser calculados em função dos autovalores de  $A$ . Se o sistema (1), vem de uma discretização de cinco pontos da equação de POISSON sobre uma região retangular com condições de fronteira de DIRICHLET, então os autovalores de  $A$  são

$$(32) \quad \lambda_i = -2(1 + \delta^2 (1 - \cos \frac{i\pi}{p+1})) , \quad i = 1, \dots, p$$

onde  $\delta$  é igual a  $\Delta x / \Delta y$  ou  $\Delta y / \Delta x$  dependendo da ordem das equações (VEJA [1]) e  $p$  é a ordem de  $A$ .

Os autovalores de  $A^{(r)}$ ,  $\bar{A}^{(r)}$  e  $s^{(r)}$  são

$$p_{2^r}(\lambda_i) \quad i = 1, \dots, p$$

$$\bar{p}_{2^r}(\lambda_i) \quad i = 1, \dots, p$$

e

$$\left[ \prod_{j=0}^{r-1} p_{2^j}(\lambda_i) \right]^{-1} \quad i = 1, \dots, p$$

respectivamente. Para avaliar esses autovalores usamos as seguintes definições de  $p_{2^r}$  e  $\bar{p}_{2^r}$  (VEJA [2]) :

$$p_{2^r}(a) = \begin{cases} -2 \cos 2^r \theta & , \text{ para } |a| < 2 \\ \text{onde } a = -2 \cos \theta \\ -2 \cosh 2^r \theta & , \text{ para } |a| \geq 2 \\ \text{onde } a = -2 \cosh \theta \end{cases}$$

e

$$\bar{p}_{2^r}(a) = \begin{cases} -(\sin(2^r+1)\theta)/\sin \theta & , \text{ para } |a| < 2 \\ \text{onde } a = -2 \cos \theta \\ -(\sinh(2^r+1)\theta)/\sinh \theta & , \text{ para } |a| \geq 2 \\ \text{onde } a = -2 \cosh \theta \end{cases}$$

De (32) verificamos que  $|\lambda_i| > 2$ ,  $i = 1, \dots, p$ . Logo, para avaliar  $p_{2^r}(\lambda_i)$  e  $\bar{p}_{2^r}(\lambda_i)$  precisamos de

$$(33) \quad \theta_i = \cosh^{-1}(-\lambda_i/2) = \cosh^{-1}(1 + \delta^2(1 - \cos \frac{4i\pi}{p+1}))$$

Agora a norma da matriz do lado direito da desigualdade (30) pode ser escrita como

$$(34) \quad ||s^{(r)}(\bar{A}^{(r)})^{-1}||_2 \leq \left( \prod_{j=0}^{r-1} ||(A^{(j)})^{-1}||_2 \right) \cdot ||(\bar{A}^{(r)})^{-1}||_2$$

Para o primeiro fator, temos

$$(35) \quad \prod_{j=0}^{r-1} ||(A^{(j)})^{-1}||_2 = \prod_{j=0}^{r-1} \max_{\{\lambda_i\}} |(P_{2^j}(\lambda_i))^{-1}| \\ = 2^{-r} \prod_{j=0}^{r-1} \max_{\{\theta_i\}} [\cosh 2^j \theta_i]^{-1} < 2^{-r}$$

onde  $\lambda_i$  é dado por (32) e  $\theta_i$  por (33).

Para o segundo fator, temos

$$(36) \quad ||(\bar{A}^{(r)})^{-1}||_2 = \max_{\{\lambda_i\}} |(\bar{p}_{2^r}(\lambda_i))^{-1}| \\ = \max_{\{\theta_i\}} \left| \frac{\sinh \theta_i}{\sinh(2^r+1)\theta_i} \right| < 1$$

pois  $|\sinh(2^r+1)\theta_i| > |\sinh \theta_i|$ ,  $\forall i$ .

Logo, de (35) e (36), concluímos que

$$(37) \quad ||s^{(r)}(\bar{A}^{(r)})^{-1}||_2 < 2^{-r} < 1$$

Portanto, de (30), temos

$$||p_0^{(r+1)} - x_0||_2 \leq ||s^{(r)}(\bar{A}^{(r)})^{-1}||_2 ||x||$$

e por (37) vemos que quando  $r$  cresce,  $2^{-r}$  tende a zero, logo o cálculo de  $p_0^{(r)}$  é estável.

A norma da matriz do lado direito da desigualdade (31) pode ser escrita como

$$(38) \quad \|\bar{A}^{(r+1)} (\bar{A}^{(r)})^{-1} s^{(r)}\|_2 \leq \|\bar{A}^{(r+1)} (\bar{A}^{(r)})^{-1}\|_2 \|s^{(r)}\|_2$$

De |1| temos que

$$(39) \quad \|s^{(r)}\|_2 < e^{-c_r \theta} p$$

onde  $c_r = 2^r - 1$  e  $p$  é a dimensão de  $A$ . Para o primeiro fator em (38), temos

$$(40) \quad \|\bar{A}^{(r+1)} (\bar{A}^{(r)})^{-1}\|_2 = \max_{\{\lambda_i\}} |\bar{p}_{2^{r+1}}(\lambda_i) (\bar{p}_{2^r}(\lambda_i))^{-1}|$$

$$= \max_{\{\theta_i\}} \left| \frac{\text{sen}(2^{r+1}+1)\theta_i}{\text{senh}(2^r+1)\theta_i} \right|$$

onde  $\lambda_i$  é dado por (32) e  $\theta_i$  por (33).

Para achar esse máximo, consideramos a função

$$g(\theta) = \frac{\text{senh}(2^{r+1}+1)\theta}{\text{senh}(2^r+1)\theta}$$

Temos

$$g'(\theta) = \frac{(2^{r+1}+1) \cosh(2^{r+1}+1)\theta \text{senh}(2^r+1)\theta - (2^r+1) \text{senh}(2^{r+1}+1)\theta \cosh(2^r+1)\theta}{\text{senh}^2(2^r+1)\theta}$$













