Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um método iterativo eficiente para resolver sistemas de equações pentadiagonais

Diego Fernando Moro[1]

Programa de Pós-Graduação em Engenharia Mecânica (PGMec) – UFPR

Centro Politécnico – UFPR, Curitiba, Paraná, Brasil

Carlos Henrique Marchi[2]

Márcio Augusto Villela Pinto[3]

Departamento de Engenharia Mecânica – UFPR

Centro Politécnico – UFPR, Curitiba, Paraná, Brasil

Resumo. É proposto e testado um algoritmo para resolver sistemas de equações pentadiagonais esparsos. Ele é baseado no TDMA (*Tridiagonal Matrix Algorithm*), sendo denominado neste trabalho de PDMA (Pentadiagonal Matrix Algorithm). Este sistema de equações possui duas diagonais lado a lado da diagonal principal da matriz e mais duas diagonais não nulas. A dedução mostrada neste trabalho pode ser feita para qualquer número de diagonais na matriz de coeficientes. O método PDMA foi testado na solução das equações de Laplace 2D e Burgers 2D. Os resultados foram comparados com outros métodos da literatura: Gauss-Seidel, ADI (*Alternating Direction Implicit*), TDMAX (a direção axial é explicita e a direção transversal é implícita) e MSI (*Modified Strongly Implicit*). Para as equações de Burgers, em uma malha de 256x256 volumes, o PDMA resolveu o problema em menos de 11% do tempo do método MSI, que por sua vez resolveu o problema em cerca de 12% do ADI.

Palavras-chave. Métodos iterativos, transferência de calor computacional, dinâmica dos fluidos computacional, sistema de equações lineares, *solver*

1 Introdução

A motivação para este trabalho foi a constante busca para diminuir o tempo computacional necessário para obter soluções numéricas.

Ao observar as equações do método iterativo (*solver*) *ADI* (*Alternating Direction Implicit*) [3], que utilizam uma dedução do TDMA (*Tridiagonal Matrix Algorithm*) [7] implícito em uma direção do plano e explícito na outra direção e vice-versa, surgiu a ideia de acoplar essas duas deduções em uma única fórmula e deduzir coeficientes para resolver o sistema linear pentadiagonal.

[1] difmoro@ufpr.br, difmoro@gmail.com
[2] marchi@ufpr.br, chmcfd@gmail.com
[3] marcio_villela@ufpr.br

2 Metodologia

O objetivo do *solver* PDMA (*Pentadiagonal Matrix Algorithm*), proposto neste trabalho, é a resolução de sistemas de equações cuja matriz de coeficientes seja do tipo pentadiagonal, ou seja, possui apenas 5 diagonais não nulas na matriz dos coeficientes. A matriz mencionada pode ser originada, por exemplo, da discretização via volumes finitos de uma malha bidimensional onde 2 diagonais são distantes da diagonal principal com nx coeficientes de distância, sendo que nx é o número de volumes na direção axial.

A equação geral deste tipo de sistema de equações é dado pela expressão:

$$a_p X_p + a_e X_E + a_w X_W + a_n X_N + a_s X_S = b_p . \quad (1)$$

Sendo que:

$X_{p,E,W,N,S}$: Incógnita, com a indicação da posição em relação a diagonal principal.

$a_{p,e,w,n,s}$: Coeficientes do sistema de equações, com a indicação da posição do coeficiente em relação à diagonal principal.

b_p : Termo fonte do sistema linear.

2.1 Equações genéricas para leste e norte

Neste trabalho será introduzido o método PDMA, utilizando-se duas equações genéricas com as seguintes expressões:

$$X_p = V_{1,p} X_E + V_{4,p} . \quad (2)$$

$$X_p = V_{2,p} X_N + V_{5,p} . \quad (3)$$

Sendo que:

$V_{i,p}$, i=1, 2, 4 e 5: Vetores incógnita do método PDMA, p é o índice em relação aos vetores incógnita do método PDMA.

O método PDMA consiste em deixar a Eq. (1) dependendo apenas das duas direções do plano, tanto Leste (E) quanto Norte (N). Basta portanto deslocar as Eqs. (2) e (3) para Oeste (W) e Sul (S), respectivamente. Teremos:

$$X_W = V_{1,W} X_p + V_{4,W} . \quad (4)$$

$$X_S = V_{2,S} X_p + V_{5,S} \quad (5)$$

Substituindo-se as Eqs. (4) e (5) na Eq. (1) e isolando os coeficientes que multiplicam as incógnitas X_E e X_N podemos encontrar os valores dos vetores incógnita do método PDMA. A equação fica da forma

$$X_p = \frac{V_{1,p}}{2} X_e + \frac{V_{2,p}}{2} X_n + \frac{V_{3,p}}{2} \tag{6}$$

Sendo que $V_{3,p}$ é a soma dos vetores $V_{4,p}$ e $V_{5,p}$, os vetores incógnitas $V_{1,p}$ e $V_{2,p}$ ficam na forma das Eqs. (7) a (9).

2.2 Equações para os vetores incógnitas

$$V_{1,p} = \frac{-2\, a_e}{a_p + a_w\, V_{1,w} + a_s\, V_{2,s}} \tag{7}$$

$$V_{2,p} = \frac{-2\, a_n}{a_p + a_w\, V_{1,w} + a_s\, V_{2,s}} \tag{8}$$

$$V_{3,p} = 2\frac{b_p - a_w V_{4,w} - a_s V_{5,s}}{a_p + a_w\, V_{1,w} + a_s\, V_{2,s}} \tag{9}$$

Na Eq. (9), os vetores $V_{4,W}$ e $V_{5,S}$ não são conhecidos, mas se os isolarmos nas Eqs. (4) e (5) e substituirmos, podemos encontrar uma expressão para $V_{3,p}$ que depende apenas de vetores calculados, desta forma teremos a Eq. (10):

$$V_{3,p} = 2\frac{b_p - a_w\left(X_w - V_{1,w} X_p\right) - a_s\left(X_s - V_{2,s} X_p\right)}{a_p + a_w\, V_{1,w} + a_s\, V_{2,s}} \tag{10}$$

Agora é possível calcular os vetores incógnitas do método PDMA seguindo a ordenação lexicográfica. Como é possível observar, o terceiro vetor incógnita do método PDMA, Eq. (10), depende da solução, nos pontos W, S e no ponto P, ou seja, o *solver* se torna iterativo.

2.3 Algoritmo

O algoritmo parte do princípio que a matriz utilizada foi obtida utilizando-se a ordem lexicográfica.

1) Estima-se um campo de valores para a variável dependente incógnita;

2) Calcula-se os vetores incógnitas do PDMA com as Eqs. (7), (8), e (10) notando os casos especiais, onde $V_{1,w}$, $V_{2,s}$, X_w, X_s não existem, calculando-se estas incógnitas em ciclo progressivo na matriz dos coeficientes;

3) Em ciclo regressivo, resolve-se o sistema linear para a variável dependente com a Eq. (6);

4) Volta-se ao passo 2 até atingir uma tolerância.

3 Modelos matemáticos e numéricos

3.1 Equações de Laplace e Burgers

Para as simplificações: condução de calor bidimensional, regime permanente, ausência de geração de calor, teremos a equação de Laplace, Eq. (11), que é a equação diferencial que modela o problema. O domínio é uma placa quadrada unitária e as condições de contorno estão na Eq. (12)

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{11}$$

$$\begin{cases} T(0,y) = T(1,y) = T(x,0) = 0 \\ T(x,1) = sen(\pi x) \end{cases} \tag{12}$$

Considerando um escoamento bidimensional de fluido incompressível com propriedades constantes, as equações diferenciais que modelam o problema são as equações de Burgers, Eq. (13). O domínio também é uma placa quadrada unitária e as condições de contorno estão na Eq. (14):

$$\begin{cases} \rho\dfrac{\partial(u^2)}{\partial x} + \rho\dfrac{\partial(uv)}{\partial y} = \mu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) - \dfrac{\partial p}{\partial x} \\ \rho\dfrac{\partial(uv)}{\partial x} + \rho\dfrac{\partial(v^2)}{\partial y} = \mu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right) - \dfrac{\partial p}{\partial y} - B(x,y,\mathrm{Re}=1) \end{cases} \tag{13}$$

$$\begin{cases} u(0,y) = u(1,y) = u(x,0) = 0 \\ u(x,1) = 16\left(x^4 - 2x^3 + x^2\right) \\ v(0,y) = v(1,y) = v(x,0) = v(x,1) = 0 \end{cases} \tag{14}$$

O termo B(x, y, Re=1) na Eq. (13) foi utilizado para encontrar uma solução analítica (neste caso, utilizou-se o método das soluções fabricadas), juntamente com o campo de pressão obtidos no artigo de Shih et al [5].

Dados numéricos para as duas equações:

- O método numérico utilizado para discretizar a equação diferencial é o método dos volumes finitos (MVF).
- As aproximações utilizadas para as derivadas são de segunda ordem de acurácia (*Central Difference Scheme, CDS*).
- As condições de contorno são aplicadas utilizando-se volumes fictícios, com o valor no contorno conhecido [1].
- A estimativa inicial do campo de temperaturas ou velocidades é nulo.
- O critério de parada do método iterativo na resolução do sistema linear, se dá com base numa tolerância sobre o resíduo adimensionalizado (razão entre o resíduo atual

pelo resíduo da primeira iteração) do sistema linear e a tolerância utilizada é de 1.10^{-10}.

3.2 *Solvers* e método de comparação dos resultados

Para a comparação do *solver* deste trabalho com outros da literatura, selecionou-se os seguintes: (1) Gauss-Seidel, (2) *Alternating Direction Implicit* (ADI) [3], (3) TDMA com a Direção Y Implícita e (4) *Modified Strongly Implicit* (MSI) [4].

Para cada tamanho de problema n é possível encontrar a expressão que representa o comportamento do tempo de CPU utilizado no *solver* da seguinte forma:

$$t_{CPU} = a\, n^{p}. \tag{15}$$

Sendo que geralmente as constantes a e p representam o desempenho de um determinado *solver*, aliado às características da programação utilizada e n é o número de incógnitas do sistema de equações. Sabe-se experimentalmente que os *solvers* utilizados para a comparação do PDMA neste trabalho possuem o expoente p da Eq. (15) próximo a 2. Segundo [6], o método Gauss-Seidel com *multigrid* resulta em um expoente p próximo a 1.

3.3 Dedução do PDMA para outros pares de volumes

Ao considerar outros pares de volumes nas Eqs. (2) e (3), por exemplo N e W, W e S e S e E e aplicar o mesmo procedimento mostrado, obteremos outros 3 *solvers* distintos, que podem influenciar no tempo de CPU se utilizados em conjunto.

Neste trabalho as deduções do método PDMA são chamadas: PDMA-EN (dedução demostrada neste trabalho com E e N), PDMA-EN-NW (uso do *solver* PDMA-EN com a dedução N e W utilizada em seguida), PDMA-EN-NS-WS (uso do *solver* PDMA-EN, PDMA-NS e do PDMA-WS um em seguida do outro nesta ordem).

4 Resultados e conclusão

4.1 Equação de Laplace e Burgers

A comparação dos resultados dos *solvers* da literatura e o PDMA estão nas Figs.1 e 2 e na Tab. 1.

Neste trabalho foi proposto um novo *solver* utilizando a média de duas deduções do método TDMA. Foi estudado também o desempenho deste *solver* frente a outros *solvers* da literatura em dois problemas: equação de Laplace e equações de Burgers. Os *solvers* da literatura utilizados foram: Gauss-Seidel, ADI, TDMAX, MSI. Foi realizado também uma análise quanto ao uso conjunto de 2 e 3 deduções do PDMA.

Tabela 1: Constantes da Eq. (15) para as equações de Laplace e Burgers.

Laplace	*p*	*a*
PDMA-EN	1.563	4.190E-08
PDMA-EN-NW	1.496	7.184E-08
PDMA-EN-NW-WS-SE	1.521	5.646E-08
GS	1.988	1.840E-08
ADI	1.991	1.727E-08
TDMAX	1.934	2.322E-08
MSI	1.934	5.573E-09

Burgers	*p*	*a*
PDMA-EN	1.572	8.296E-08
PDMA-EN-NW	1.515	1.163E-07
PDMA-EN-NW-WS-SE	1.557	8.764E-08
GS	2.011	4.393E-08
ADI	2.001	3.181E-08
TDMAX	1.979	4.409E-08
MSI	1.828	2.444E-08

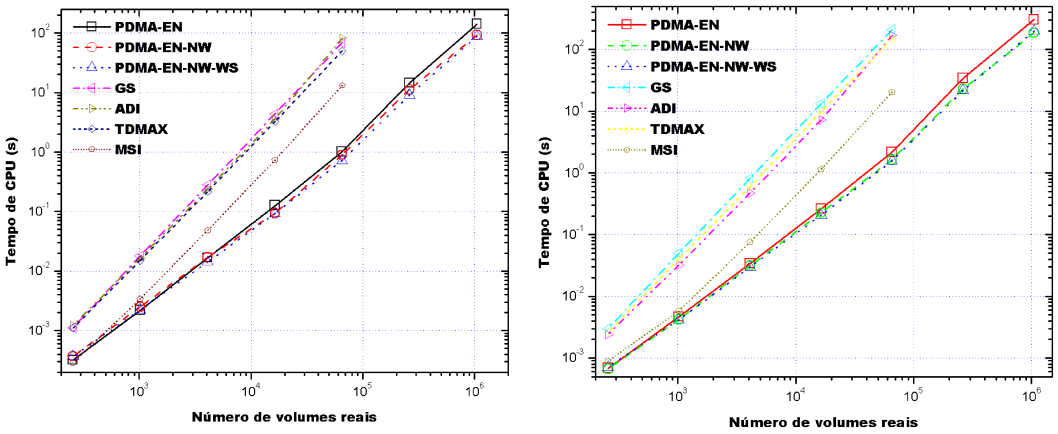

Figura 1: Comparação entre os tempos de CPU para as equações de Laplace (esquerda) e de Burgers (direita).

Com base nos testes realizados, pode-se concluir que:

1 – Todas as deduções do PDMA propostas, convergiram mais rápido do que os *solvers* da literatura utilizados neste trabalho.

Na malha 256x256 das equações de Burgers, o método PDMA-EN convergiu em 10,69% do tempo necessário para o MSI (melhor *solver* da literatura utilizado), que por sua vez convergiu em 11,94% do tempo necessário para o *solver* ADI.

2 – Utilizando-se o PDMA em conjunto, o tempo computacional diminui consideravelmente: na malha mais fina simulada para as equações de Burgers, o *solver* PDMA-EN-NW-WS resolveu o problema 34,32% mais rápido que o *solver* PDMA-EN. O *solver* PDMA-EN-NW foi 37,95% mais rápido que o PDMA-EN.

3 – O uso do *solver* proposto neste trabalho é mais rápido que os *solvers* da literatura utilizados.

4 – Com base na Eq. (15), o expoente *p* para todas as versões do PDMA aproxima-se de 1,5.

Este valor do expoente p torna o *solver* deste trabalho mais próximo do *solver* Gauss-Seidel com *multigrid* (no qual o expoente desta curva se aproxima de 1,0) e

melhor do que todos os outros *solvers* da literatura testados neste trabalho (que se aproximam de 2,0).

Lembrando que para um mesmo expoente, a diferença de tempo computacional se encontra no número de operações necessárias para o determinado *solver*. No entanto a mudança neste expoente se reflete em um comportamento diferente do tempo de CPU, tornando o *solver* mais rápido que os outros com o refino da malha.

Foi constatado que para problemas com condições de contorno diferentes de *Dirichlet* (valor conhecido no contorno) o *solver* não funciona, gerando divisão por zero em alguns pontos da malha. É possível, no entanto, utilizá-lo apenas para o interior do domínio e os contornos resolvidos com *Gauss-Seidel*, desta forma o *solver* funciona mas é mais parecido com os *solvers* tradicionais da literatura.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). O primeiro autor é bolsista da CAPES. O segundo autor é bolsista do CNPq.

Referências

[1] S. C Chapra e R. P. Canale, Métodos Numéricos para Engenharia, 5 Ed., McGraw-Hill, São Paulo (2008).

[2] F. P. Incropera e D. P. DeWitt, Introduction to Heat Transfer, 3rd Ed., Wiley, New York (1996).

[3] D. W. Peaceman and H. H. Rachford, The Numerical Solution of Parabolic and Eliptic Differential Equations, J. Soc. Ind. Appl. Math, vol. 3, pp. 28-41 (1955).

[4] G. Schneider and M. Zedan, A modified strongly implicit procedure for the numerical solution of field problems. Numerical Heat Transfer, vol. 4, pp. 1-19 (1981).

[5] T. M. Shih, C. H. Tan and B. C. Hwang, Effects of Grid Staggering on Numerical Schemes, Int. J. Numer. Meth. Fluids, vol. 9, pp. 193–212 (1989).

[6] R. Suero, M. A. V. Pinto, C. H. Marchi, L. K. Araki e A. C. Alves, Otimização do método multigrid algébrico para as equações bidimensionais de Laplace e Poisson, VI Congresso Nacional de Engenharia Mecânica, Paraíba, Brasil (2010).

[7] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics – The Finite Volume Method, Pearson, Prentice Hall, 2. Ed, (2007).