

Método SOR Paralelo na Simulação das Equações de Navier-Stokes 2D

Darlon Vasata

Departamento de Engenharia Elétrica - Escola Politécnica da USP
Avenida Prof. Luciano Gualberto, travessa 3 nº 158, 05508-900, São Paulo, SP, Brasil
darlonv@gmail.com

Guilherme Galante, Rogério L. Rizzi

Centro de Ciências Exatas e Tecnológicas, UNIOESTE, Campus de Cascavel
Rua Universitária, 2069, 85801-110, Cascavel, PR, Brasil
gui.galante@gmail.com, rogerio@unioeste.br

Resumo

Neste trabalho, são construídos e implementados modelos computacionais baseados na equações de Navier-Stokes bidimensionais. A complexidade matemática dessas equações impossibilita encontrar soluções analíticas a partir da análise teórica, havendo a demanda de métodos numéricos para resolvê-las. Considerando que tais métodos numéricos requerem grande quantidade de processamento e de memória, o uso de ambientes computacionais de alto desempenho e de estratégias de paralelismo se fazem necessários. Os resultados mostraram que as estratégias e técnicas empregadas obtiveram boa qualidade numérica e bom desempenho computacional.

1. Introdução

A dinâmica dos fluidos computacional (DFC) é uma abordagem contemporânea de pesquisa em dinâmica dos fluidos. Ela surgiu com o advento dos computadores e vem sendo utilizada na solução de vários problemas complexos, como o escoamento de fluidos.

Um motivo que leva os pesquisadores a utilizar a DFC em seus trabalhos é o fato de que a realização de experimentos possui um alto custo e muitas vezes são inviáveis. Assim, a partir das informações fornecidas pela DFC, pesquisas teóricas podem ser validadas ou novas abordagens podem ser sugeridas [12].

Neste trabalho são construídos e implementados modelos matemáticos aplicados à DFC baseados nas equações de Navier-Stokes bidimensional. As equações de Navier-Stokes são equações diferenciais que descrevem o escoamento de fluidos e que possibilitam determinar os campos de velocidade e de pressão entre outras variáveis de inte-

resse. A complexidade matemática dessas equações impossibilita encontrar soluções analíticas a partir da análise teórica, havendo a demanda de métodos numéricos para resolvê-las.

Considerando que tais métodos numéricos requerem grande quantidade de processamento e de memória, o uso de ambientes computacionais de alto desempenho e estratégias de paralelismo se fazem necessários.

2. Método Pressão-Velocidade

Para se formular uma solução das equações de Navier-Stokes alguns fatores devem ser levados em conta. Considera-se inicialmente que um fluido é um contínuo de massa. Com isso, as propriedades do escoamento podem ser descritas em termos de grandezas macroscópicas como velocidade, pressão, temperatura e densidade [10].

Para conservar as propriedades físicas para o modelo é necessário resolver a equação da continuidade, que é uma representação matemática para a conservação da massa. Essa equação pode ser representada matematicamente como:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (1)$$

onde u e v são as componentes da velocidade do fluido nas direções coordenadas. Uma estratégia de solução para as equações de Navier-Stokes é reescrevê-las, após certa manipulação algébrica, na forma de uma equação de Poisson para a pressão a cada passo de tempo. A representação da equação de Poisson em sua forma analítica é mostrada na Equação 2.

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f \quad (2)$$

onde f agrega as contribuições dos operadores explícitos para a velocidade e demais termos (veja a Equação 5), sendo que p designa pressão [7].

2.1 Discretização das Equações

Em diversas abordagens encontradas na literatura para soluções de escoamentos, a velocidade (u), a velocidade (v) e a pressão (p) são representadas no mesmo ponto, ao centro de cada célula da malha [1]. Porém, essa representação pode levar a instabilidades na solução numérica do problema, existindo assim a necessidade de uma nova representação física das velocidades e pressão na malha. Assim, de modo a evitar este tipo de inconveniente, neste trabalho utiliza-se a abordagem de malha deslocada.

A malha deslocada consiste em representar a velocidade v nas paredes inferior e superior da célula e a velocidade u nas paredes esquerda e direita das células. Já para a pressão, não existe necessidade de alteração, permanecendo no centro da célula. Sendo assim, as velocidades e a pressão p são representadas na malha como mostrado na Figura 1.

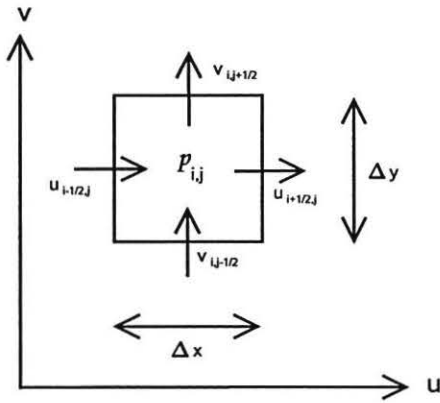


Figura 1. Representação das velocidades u e v e da pressão p na malha deslocada

Assim, pode-se mostrar que as expressões que descrevem o movimento do fluido nas direções coordenadas, são respectivamente dados por:

$$u_{i+1/2,j}^{n+1} = F_{i+1/2,j}^n - \frac{\Delta t}{\rho} \frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\Delta x} \quad (3)$$

e

$$v_{i,j+1/2}^{n+1} = G_{i,j+1/2}^n - \frac{\Delta t}{\rho} \frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{\Delta y} \quad (4)$$

Já a equação utilizada para o cálculo da pressão p é dada

por:

$$\frac{p_{i+1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{p_{i,j+1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j-1}^{n+1}}{(\Delta y)^2} = \frac{\rho}{\Delta t} \left[\frac{F_{i+1/2,j}^n - F_{i-1/2,j}^n}{\Delta x} + \frac{G_{i,j+1/2}^n - G_{i,j-1/2}^n}{\Delta y} \right] \quad (5)$$

sendo Δx e Δy os espaçamentos da malha nas direções coordenadas e Δt o espaço de tempo.

Nas equações 3, 4 e 5 os termos F e G estão relacionados à viscosidade cinemática e aos termos advectivos no instante n , relacionados a u e v , respectivamente. Mais detalhes podem ser obtidos na literatura técnica [1][10][8].

2.2 Problema da Cavidade Quadrada

O problema da cavidade é um problema de escoamento amplamente estudado e sobre ele é possível encontrar muitos trabalhos na literatura, principalmente para o caso bidimensional [7]. É bastante utilizado para validações de implementações quando se pretende analisar questões como acurácia numérica, condições de contorno e eficiência computacional.

Neste problema, um determinado fluido com viscosidade cinemática ν inicialmente encontra-se em repouso, contido em uma cavidade quadrada com lado de tamanho L , onde tal fluido preenche a cavidade por completo. Instantaneamente, a tampa superior da cavidade desloca-se com uma velocidade u_0 . As paredes são rígidas e impermeáveis, onde as paredes laterais e inferior são não-eskorregadias e a superior é móvel e skorregadia.

3. Solução das Equações de Navier-Stokes

Considerando que em abordagens práticas para problemas relacionados à DFC require-se, como já mencionado, grande quantidade de processamento e de memória, faz-se necessário uso de ambientes computacionais de alto desempenho e estratégias de paralelismo. Nesta seção apresentam-se as estratégias para solução e para a exploração do paralelismo.

3.1 Malha Computacional

Neste trabalho a simulação do problema da cavidade quadrada utiliza malhas estruturadas [3] particionadas através de um algoritmo de bissecção recursiva. Esta técnica estabelece as divisões com base nas coordenadas físicas dos vértices da malha. Em cada passo a divisão é feita ao longo da maior dimensão, e então repete-se o processo recursivamente até que se obtenha a quantidade de partições desejadas [6]. Um exemplo de particionamento pode ser observado na Figura 2, onde a malha é particionada em 16 subdomínios.

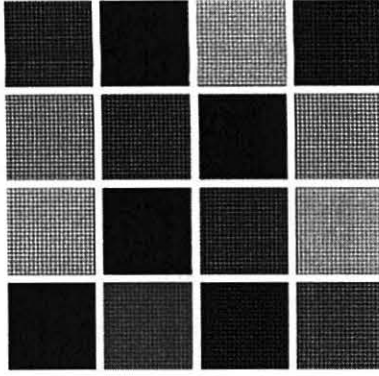


Figura 2. Domínio particionado em 16 sub-domínios

3.2 Algoritmo de Solução

O procedimento para o cálculo das variáveis e da pressão no nível de tempo $n + 1$ consiste em resolver os passos apresentados na Figura 3 e descritos a seguir.

No primeiro passo é feito todo pré-processamento, que inclui a leitura dos arquivos de entrada e a inicialização das variáveis e parâmetros da simulação.

No segundo passo define-se as *condições de fronteira*. As condições de fronteira são abordagens utilizadas para representar a interação do sistema com o meio externo. Consiste em adicionar valores nas células de borda do domínio computacional, sendo estas assim denominadas *células de fronteira*. Neste trabalho, as células de fronteira aplicam a técnica da reflexão, podendo comportar-se das formas *escorregadia*, ou *fronteira de simetria* e não *escorregadia* ou *fronteira sólida* [7].

Nos dois passos subsequentes, efetua-se as operações relacionadas ao cálculo das variáveis F e G no interior das células e nas bordas do domínio.

Na seqüência resolve-se a Equação 5 para todos os pontos internos da malha. Nessa etapa utiliza-se o método numérico SOR (*Successive Over Relaxation*), que é um método iterativo, o qual é utilizado para resolver as equações geradas pela aplicação do lado direito da Equação 5 a todos os pontos internos da malha [5]. A Equação 6 deve a ser aplicada para cada ponto interno da malha.

$$p_{i,j}^{n+1,(k+1)} = (1 - \omega)p_{i,j}^{n+1,(k)} + \frac{\omega}{2(1+\beta^2)} \left[p_{i+1,j}^{n+1,(k)} + p_{i-1,j}^{n+1,(k)} + \beta^2 p_{i,j+1}^{n+1,(k)} + \beta^2 p_{i,j-1}^{n+1,(k)} - (\Delta x)^2 f_{i,j} \right] \quad (6)$$

onde (k) e $(k + 1)$ representam as iterações do SOR, $\beta = \frac{\Delta x}{\Delta y}$, ω é um fator de relaxação, que deve estar entre $0 < \omega < 2$, e

$$f_{i,j} = \frac{\rho}{\Delta t} \left[\frac{F_{i+1/2,j}^n - F_{i-1/2,j}^n}{\Delta x} + \frac{G_{i,j+1/2}^n - G_{i,j-1/2}^n}{\Delta y} \right] \quad (7)$$

onde ρ denota a densidade.

A abordagem de solução utilizada, que é obter as componentes da velocidade a partir dos valores atualizados da pressão, obriga a que em cada iteração do SOR seja efetuada uma correção da pressão com o intuito de estimar o campo de velocidades, de forma a satisfazer a equação da continuidade e obter uma representação física consistente [7].

Considerando que não há escoamento dentro da cavidade, a pressão deve ser normalizada após cada iteração do método SOR de modo a se evitar que os valores da pressão cresçam ou diminuam arbitrariamente. A normalização consiste em manter constante o valor da pressão em uma determinada célula da malha. Como manter um valor constante pode ocasionar instabilidades na solução, uma alternativa é tomar uma determinada célula como referência e aplicar a correção todos os pontos da malha, utilizando a Equação 8.

$$p_{i,j}^k \leftarrow p_{i,j}^k - p_{ref} \quad (8)$$

onde este passo deve ser feito a cada iteração do método SOR, indicado pelo índice k . Repare que utilizando a técnica de correção, o valor da pressão na célula de referência é mantido constante em 0 [7].

A escolha do critério de parada utilizada no método SOR é a norma euclidiana residual, mostrada na Equação 9.

$$\|R\|_2 = \left(\sum_{i=2}^{NI-1} \sum_{j=2}^{NJ-1} R_{i,j}^2 \right)^{1/2} \quad (9)$$

sendo $NJ - 1$ e $NI - 1$ o número de célula nas direções coordenadas, e

$$R_{i,j} = \frac{\rho}{\Delta t} \left[\frac{F_{i+1,j}^n - F_{i,j}^n}{\Delta x} + \frac{G_{i,j+1}^n - G_{i,j}^n}{\Delta y} \right] - \left(\frac{p_{i+1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{p_{i,j+1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j-1}^{n+1}}{(\Delta y)^2} \right)$$

Uma vez finalizada a resolução dos sistemas de equações, é verificado se a simulação atingiu um estado estacionário. Em caso negativo, inicia-se uma nova iteração no tempo t . Em caso positivo encerra-se a aplicação.

3.3 Paralelização do Modelo

O modelo paralelo foi implementado de modo que possibilite que o software seja executado de forma distribuída sob o ambiente de cluster de computadores. São utilizadas todas as funcionalidades do modelo bidimensional, apresentadas na Figura 3, acrescentando ainda os procedimentos para controle das trocas de mensagens. Utilizou-se a biblioteca

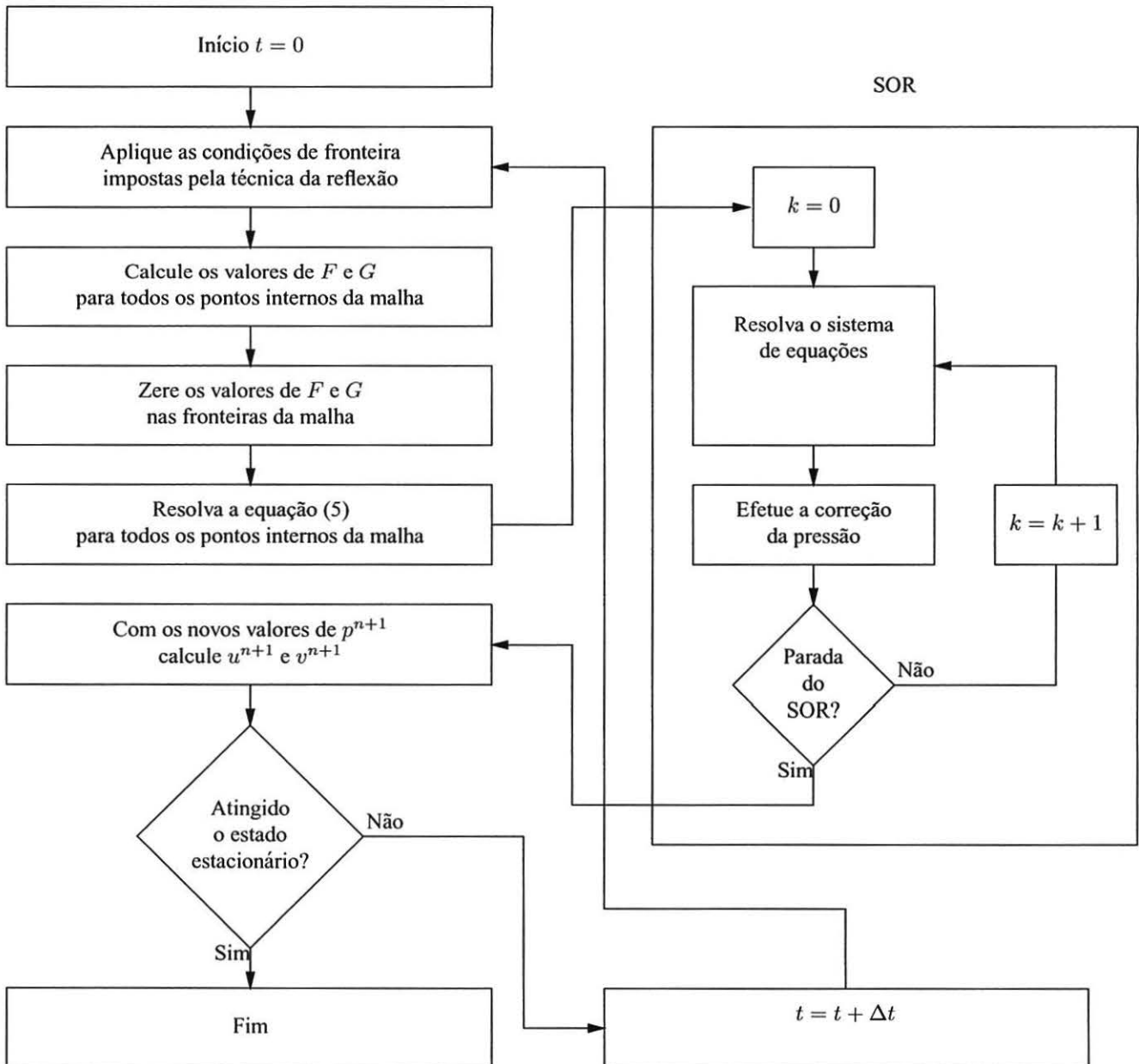


Figura 3. Esquema do método de resolução da pressão e velocidades

MPICH (*MPI Chameleon*) para possibilitar a comunicação entre processos utilizando troca de mensagens [11].

O processo de paralelização inicia-se no particionamento da malha. A geração de cada subdomínio é feito no pré-processamento, antes que a aplicação inicie. As informações referentes a cada processo são salvas em arquivo pelo particionador e abertas posteriormente. Nestes arquivos armazena-se as informações locais mais uma área de sobreposição. A sobreposição consiste em que cada sub-processo possua células também de domínios vizinhos, formando uma sobreposição, de tal forma a manter a continuidade da solução nos valores da fronteira. Um exemplo de particionamento do domínio com sobreposição de duas células pode ser visto na Figura 4.

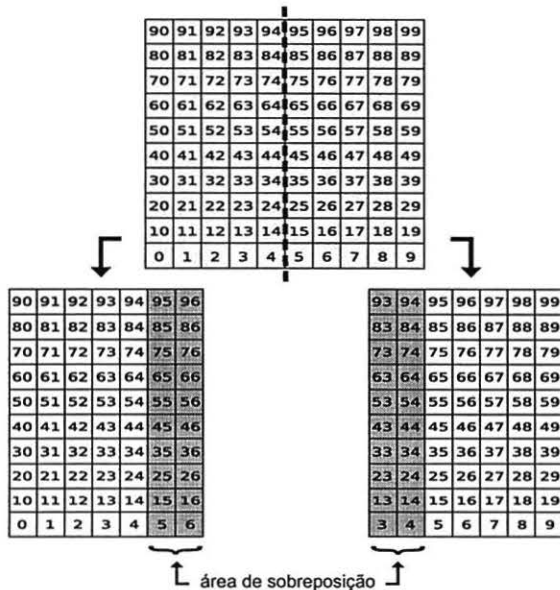


Figura 4. Particionamento da malha em 2 subdomínios com 2 células de sobreposição

A paralelização do método de solução é baseada nas características das Equações (6), (7) e (8). Como visto na seção 3.2, no cálculo das pressões $p_{i,j}^{n+1,(k+1)}$ em cada ponto da malha (Equação 6), deve-se calcular $f_{i,j}$. O cálculo deste termo (Equação 7) utiliza os valores de $F_{i+1/2,j}^n$, $F_{i-1/2,j}^n$, $G_{i,j+1/2}^n$ e $G_{i,j-1/2}^n$, que correspondem aos valores de F e G armazenados nas células vizinhas de uma dada célula (i, j) do domínio computacional. Note que com o particionamento da malha, tais valores podem estar em subdomínios distintos, e por consequência processos diferentes, havendo a necessidade de comunicação entre eles para a obtenção dos valores atualizados. Dessa forma, ao fim de cada ciclo do método de solução, deve haver a troca de dados entre os subdomínios para a atualização dos valores de F s e G s das áreas de sobreposição.

Deve-se trocar também dados relacionados à pressão de referência. Para isso, toma-se uma determinada célula como referência e aplica-se a Equação 8 a todos os pontos da malha. No entanto, a célula de referência situa-se em um subdomínio particular e dessa forma deve-se enviar o valor de tal célula para os demais processos através de uma comunicação coletiva (*broadcast*).

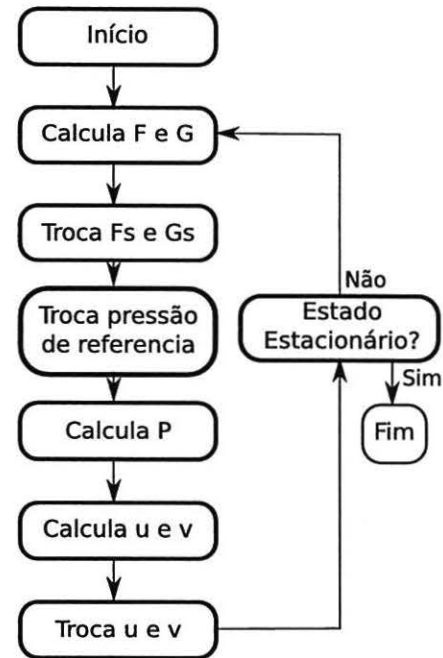


Figura 5. Algoritmo Paralelo

Como pode-se observar na Figura 5, após as trocas de dados para atualização de F s, G s e para a correção da pressão, cada processo calcula a pressão (p) e as velocidades (u e v) de suas células locais. Na seqüência, mais uma troca de dados entre os processos é feita para a atualização dos valores de u e v , utilizados para o cálculo de F e G no próximo ciclo do método.

O procedimento descrito nessa seção é repetido até que o modelo alcance um estado estacionário. O algoritmo paralelo é apresentado na Figura 5.

4 Resultados

O algoritmo para o cálculo bidimensional em paralelo da pressão e das velocidades foi implementado utilizando a linguagem de programação *C* com o compilador *GCC* (*Gnu Compiler Collection*) sob sistema operacional *Linux* versão 2.6.5. Para as simulações foi utilizado o cluster *Krusty* disponível no laboratório *LCAD* da *UNIOESTE*, Campus de *Cascavel*. O cluster é formado por 18 nodos *Pentium 4 3.0*

GHz, memória RAM de 1 GB e disco rígido de 80 GB, interligados por rede Gigabit Ethernet.

4.1 Testes de Desempenho

Foram efetuados testes com malhas com 1600, 2500, 3600, 4900 e 6400 células, onde estas possuem as dimensões de 40x40, 50x50, 60x60, 70x70 e 80x80, respectivamente. As simulações foram calculadas utilizando 1, 2, 4, 8, e 16 computadores.

Considera-se apenas o tempo de computação dos valores, desconsiderando as etapas de pré e pós processamento. Os parâmetros utilizados na computação dos valores são mostrados na Tabela 1, e os gráficos de tempo de execução, speedup e eficiência são mostrados nas Figuras 6, 7 e 8, respectivamente. Nestas figuras, cada conjunto de barras do gráfico é referente a um domínio testado e a cor das barras indicam a quantidade de processos utilizado.

Tabela 1. Parâmetros utilizados para as simulações em paralelo

Parâmetro	Valor
Δt	$1.000000e + 00$
Δx	$1.000000e - 01$
Δy	$1.000000e - 01$
u_0	$3.083000e - 05$
ν	$1.850000e - 03$
μ	$1.200000e + 00$
ε	$1.000000e - 06$

Analisando os gráficos pode-se notar que a implementação paralela obteve desempenho satisfatório, principalmente para os domínios de maior tamanho. Nestes casos existe uma quantidade maior de processamento para cada troca de dados e por conseqüência um melhor aproveitamento da capacidade de processamento dos nodos.

Em geral, a implementação apresentou um comportamento bastante homogêneo, onde o tempo de execução decresceu com o aumento da quantidade de processos. Conforme se dobrava o número de processadores o tempo de execução ficava, em média, 1,3 vez menor. O *speedup* máximo alcançado foi de aproximadamente 6,5 utilizando um domínio 70x70 com 16 processadores e a eficiência máxima de 72% foi alcançada no domínio 60x60 com 2 processadores.

Pode-se destacar que o método SOR não é particularmente eficiente quando implementado de forma sequencial ou paralela visto a sua necessidade de fazer k iterações a cada passo de tempo para a integração temporal e para correção de pressão. Para os experimentos numéricos realizados utilizou-se 100 subpassos de tempo em cada ciclo,

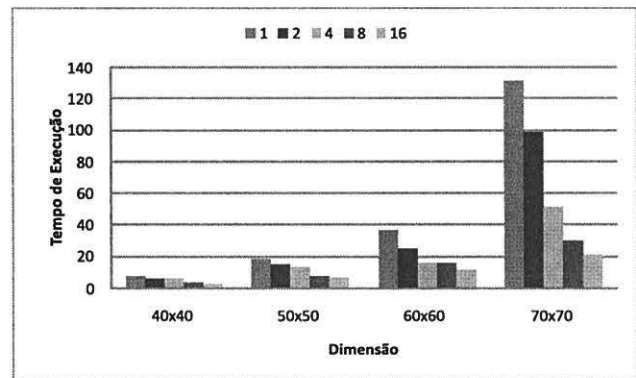


Figura 6. Tempo de execução

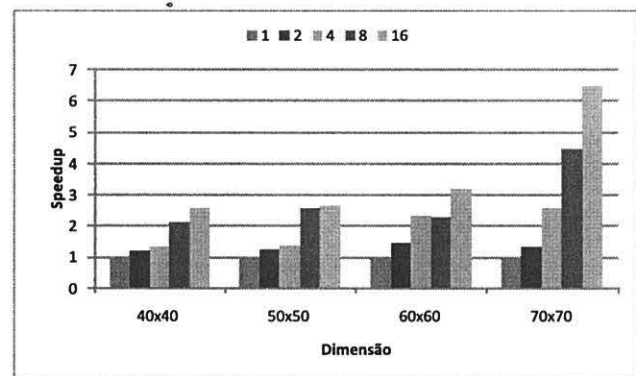


Figura 7. Speedup

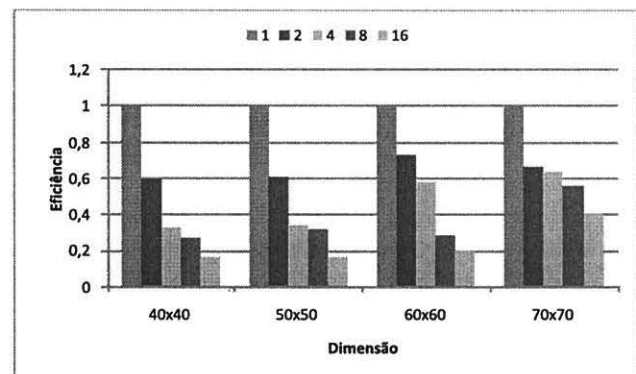


Figura 8. Eficiência

isto é, k variou de 1 a 100, o que acabou influenciando nos ganhos de desempenho.

4.2 Resultados da Aplicação

Nesta seção apresenta-se alguns resultados obtidos com o modelo paralelo implementado. Para a visualização dos resultados são utilizados gráficos de campos de pressão (Figura 9), os campos de velocidade (Figuras 10 e 11) e as linhas de corrente geradas pelo escoamento (Figura 12). Nesta simulação empregou-se um domínio com dimensões de 40×40 células, distribuídos entre 4 processos utilizando o algoritmo de bissecção recursiva.

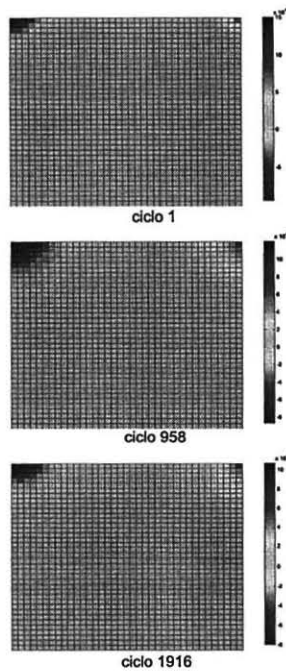


Figura 9. Campo de pressão obtido nos ciclos 1, 958 e 1916 da resolução do problema da cavidade quadrada

Para os parâmetros utilizados os resultados obtidos estão consistentes com aqueles disponíveis na literatura técnica. Analisando visualmente os campos de velocidades relativas apresentados neste trabalho, pode-se observar que os resultados numéricos são muito próximos àqueles disponíveis por outros autores [14] [4] [2] [9]. Algumas diferenças observadas devem-se às distintas condições de contorno utilizadas e aos diferentes métodos de solução implementados. Isso ocorre quando comparamos os resultados com um único processo com os demais trabalhos, visto que não é comum encontrar simulações paralelas das equações de Navier-Stokes utilizando o método SOR.

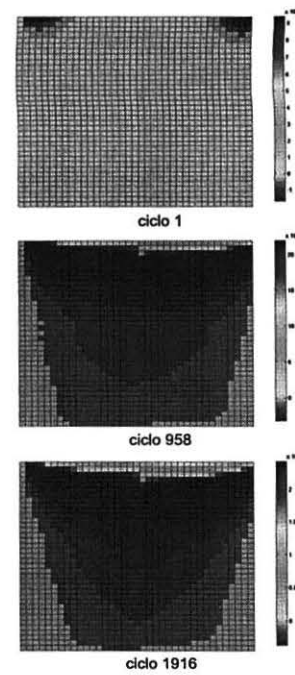


Figura 10. Campo de velocidade u obtido nos ciclos 1, 958 e 1916 da resolução do problema da cavidade quadrada

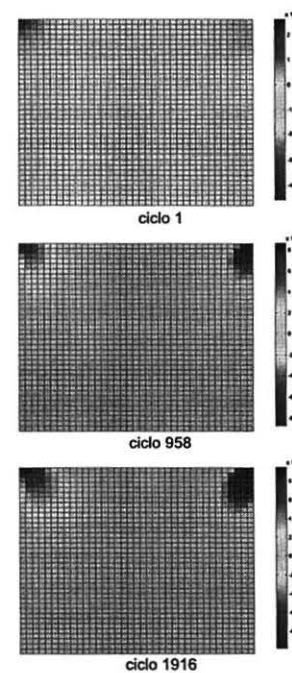


Figura 11. Campo de velocidade v obtido nos ciclos 1, 958 e 1916 da resolução do problema da cavidade quadrada

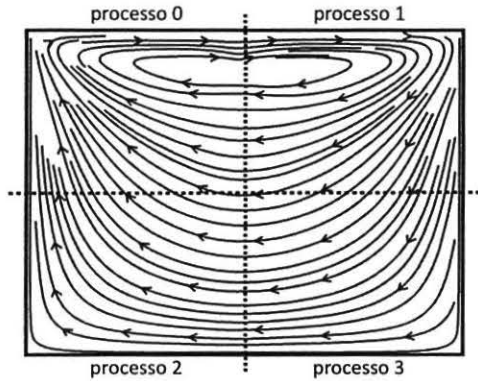


Figura 12. Linhas de Corrente obtidas no ciclo 1916 em simulação utilizando quatro processos

Também destaca-se a estratégia de paralelização implementada. Esta baseou-se no fato que as condições de contorno utilizadas estão agregadas, de acordo com a discretização, na Equação 7 que incorpora os termos explícitos F e G . Tal equação é um dos fatores a ser resolvido pelo método SOR, já que é um dos componentes da Equação 6. Desse modo, bastou uma troca de dados nas sobreposições para garantir a continuidade da solução, já que os valores requeridos para tal já foram calculados e estão disponíveis nos termos F e G . Dessa forma, o cálculo da pressão em uma célula (i, j) é feito de forma física e numericamente correta, e conseqüentemente, os atributos relacionados às velocidades. A continuidade da solução é o resultado que atesta este fato, como comprova o conjunto de linhas de corrente mostrado na Figura 12.

5 Conclusões

Este trabalho apresentou um modelo computacional paralelo para a solução das equações de Navier-Stokes bidimensional aplicada ao problema da cavidade quadrada. Outros experimentos numéricos, bem como a utilização de outros parâmetros geométricos e físicos foram realizados pelos autores [15] visando a verificação dos seus resultados com aqueles disponíveis na literatura. Não obstante essas realizações devem-se fazer outros ainda buscando um melhor equilíbrio entre a acurácia numérica e a eficiência computacional. De fato, testes preliminares com o número de iterações do SOR entre outros fatores, indicam que a eficiência pode ser melhorada significativamente simplesmente obtendo o número mínimo de iterações que ainda corrija satisfatoriamente a pressão.

Outros trabalhos podem indicar a viabilidade, ou não, de se utilizar métodos iterativos não estacionários como o GMRES e o CG [13]. Alguns experimentos numéricos re-

alizados indicaram dificuldades de conciliar a utilização de tais métodos de solução com a abordagem empregada neste trabalho para a solução das equações de Navier-Stokes, onde constatou-se certas inconsistências com a utilização do método acoplamento pressão-velocidade e correção de pressão com o GMRES.

Finalmente, acredita-se que a abordagem paralela implementada tem certo grau de inovação, visto não foram encontrados trabalhos relacionados na literatura disponível.

Referências

- [1] J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, New York, 1995.
- [2] E. O. B. Asenjo, J. C. Claeysen, and O. Rubio. Numerical simulation for the natural convection flow. *Revista de Pesquisa e Pós-Graduação*, pages 237–254, 2000.
- [3] M. Bern and P. Plassmann. Mesh generation. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. Elsevier Science, 2000.
- [4] J. A. da Rosa. Determinação dos campos de velocidade e temperatura em ambientes ventilados. Mestrado em engenharia, UFRGS, Porto Alegre, 2001.
- [5] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, New York, 1997.
- [6] P.-O. Fjällström. Algorithms for graph partitioning: a survey. In *Proceedings Linköping Electronic Articles in Computer and Information Science*, volume 3, Linköping, 1998.
- [7] A. O. Fortuna. *Técnicas Computacionais para Dinâmica dos Fluidos: Conceitos Básicos e Aplicações*. EDUSP, São Paulo, 2000.
- [8] H. P. Langtangen, K.-A. Mardal, and R. Winther. Numerical methods for incompressible viscous flow. Lecture Notes, 2002.
- [9] M. Matyka. Solution to two-dimensional incompressible navier-stokes equations with simple, simpler and vorticity-stream function approaches. driven-lid cavity problem: Solution and visualization. Technical report, University of Linköping, 2003.
- [10] J. M. McDonough. Lecture in computational fluid dynamics of incompressible flow: Mathematics, algorithms and implementations. Lecture Notes, 2003.
- [11] P. S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufman, San Francisco, 1997.
- [12] R. L. Rizzi. *Modelo Computacional Paralelo para a Hidrodinâmica e para o Transporte de Substâncias Bidimensional e Tridimensional*. PhD thesis, UFRGS, Porto Alegre, 2002.
- [13] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
- [14] R. Suero. Verificação de soluções numéricas de escoamentos bidimensionais laminares em malhas uniformes. Mestrado em métodos numéricos em engenharia, UFPR, Curitiba, 2006.
- [15] D. Vasata. Solução computacional paralela da equação de navier-stokes bidimensional e tridimensional em malhas estruturadas. Trabalho de conclusão de curso, UNIOESTE, Cascavel, 2007.