



DESEMPENHO DE UM ALGORITMO *MULTIGRID* PARALELO APLICADO ÀS EQUAÇÕES DE NAVIER-STOKES

Réverton Luis Antunes Neundorf

reverton@simepar.br

Sistema Meteorológico do Paraná - SIMEPAR, Curitiba-PR, Brasil

Marcio Augusto Villela Pinto

Luciano Kiyoshi Araki

marcio_villela@ufpr.br

lucaraki@ufpr.br

Universidade Federal do Paraná, Departamento de Engenharia Mecânica, Curitiba-PR, Brasil

Leonardo Calvetti

lcalvetti@ufpel.edu.br

Universidade Federal de Pelotas, Faculdade de Meteorologia (FMet), Pelotas-RS, Brasil

Abstract. *O escoamento laminar bidimensional de um fluido incompressível, governado pelas equações de Navier-Stokes, é o foco deste trabalho. Para tanto, utilizou-se o método de volumes finitos (MVF), com esquema de aproximação de segunda ordem (CDS), além de um método de projeção com correção incremental na pressão e convergência temporal de segunda ordem. O solver utilizado foi o método de Gauss-Seidel red-black. Para a obtenção da solução, foi empregado o método multigrid geométrico, com esquema de correção CS, restrição por ponderação completa, prolongação utilizando interpolação bilinear e número máximo de níveis para os casos estudados. A paralelização do multigrid foi realizada aplicando-se uma metodologia de particionamento do domínio a cada uma de suas componentes algorítmicas: solver, restrição, prolongação e cálculo do resíduo. Foi possível testar a superioridade do multigrid em relação à utilização do método de malha única (singlegrid). Finalmente, com a paralelização do método multigrid foi possível reduzir em até 8 vezes, utilizando 14 processadores, o tempo de CPU necessário para se obter as soluções das equações de Navier-Stokes.*

Keywords: *Navier-Stokes, métodos de projeção, multigrid, métodos iterativos, volumes finitos, paralelização, particionamento do domínio*

1 INTRODUÇÃO

Neste trabalho foi resolvido numericamente o problema do escoamento laminar bidimensional de um fluido incompressível governado pelas equações de Navier-Stokes com condições de contorno de Dirichlet para as velocidades.

Utilizou-se um método de projeção com correção incremental na pressão e convergência temporal de segunda ordem. O primeiro, e mais simples, método considerado de projeção foi proposto por Chorin (1968) e Temam (1969). Utilizando o método de Euler na discretização temporal o algoritmo consiste em criar uma velocidade auxiliar com a qual se estima as velocidades no passo de tempo atual a partir do valor das velocidades no passo anterior ignorando a pressão. A partir da velocidade auxiliar é calculada a pressão, são aplicadas as condições de incompressibilidade e realizada a correção da velocidade no tempo atual. O aspecto mais atrativo do método é a solução de equações elípticas em cada uma das etapas do algoritmo, no entanto, a acurácia do método é prejudicada devido à imposição de condições de contorno de Neumann artificiais.

Goda (1979) observou que adicionando o valor do gradiente de pressão de uma iteração anterior à primeira etapa do algoritmo de Chorin (1968), a qual ignora a pressão completamente na primeira etapa, a acurácia do algoritmo aumenta. Esta observação foi utilizada por Van Kan (1986) na formulação de um método de correção na pressão de segunda ordem. O método em questão é conhecido como esquema de correção incremental na pressão na forma padrão.

Em Bell et al. (1989) encontra-se um método de projeção bastante conhecido, similar ao de Van Kan (1986), ou seja, consiste em um esquema de correção incremental na pressão na forma padrão.

Na tentativa de superar o problema de se ter a acurácia afetada devido a condições de contorno impostas à pressão, Timmermans et al. (1996) propuseram adicionar o divergente das velocidades auxiliares na segunda etapa do algoritmo de Van Kan (1986). Esta pequena modificação conduz a melhorias na taxa de convergência da pressão.

Em Kim e Moin (1985) encontra-se um método de projeção que, como o método de Chorin (1968), ignora a pressão na primeira etapa do algoritmo e não realiza a correção da pressão na segunda etapa, no entanto apresenta taxas de convergência similares às obtidas pelo esquema de correção incremental da pressão na forma rotacional de Timmermans et al. (1996). Pelo fato de ignorar a pressão na primeira etapa e não realizar a correção da mesma na segunda etapa Brown et al. (2001) referem-se ao esquema de Kim e Moin (1985) como um método de projeção independente da pressão.

Em Orszag et al. (1986) e Karniadakis et al. (1991) é apresentado uma nova classe de métodos de projeção conhecidos como esquemas de correção na velocidade. De forma similar ao caso da correção na pressão, os métodos de correção na velocidade também possuem uma versão com correção incremental na forma padrão e, introduzida por Guermond e Shen (2003), na forma rotacional.

O método numérico utilizado para a discretização do modelo matemático, foi o método dos volumes finitos (Maliska, 2004). Foram utilizadas malhas desencontradas e condições de

contorno impostas com o auxílio de volumes fictícios. Para o sistema de equações algébricas resultantes do processo de discretização do domínio e da equação diferencial do modelo matemático utilizou-se o método *multigrid* (Briggs et al., 2000; Trottenberg et al., 2001), que pertence à família dos métodos iterativos utilizados para resolver com eficiência sistemas de equações oriundos da discretização de equações diferenciais parciais e têm como principal objetivo acelerar a convergência do esquema iterativo em que são aplicados (Tannehill et al., 1997). O interesse em aplicar o método *multigrid* em problemas de larga escala, mantendo o custo computacional (tempo de processamento) aceitável, conduz à necessidade de paralelização do mesmo. Uma abordagem que satisfaz às condições exigidas é obtida através da decomposição do domínio em subdomínios menores (com menos incógnitas) e solução destes em paralelo. Vários autores (Chan e Saad, 1986; Chan e Schreiber, 1985; Hempel e Schüller, 1988; Herbin et al., 1988; Thole, 1985; McBryan, 1990) estudaram a paralelização dos métodos *multigrid*. Contudo, segundo Trottenberg et al. (2001), o *multigrid* padrão não é completamente paralelizável, pois as malhas são processadas sequencialmente e o grau de paralelismo é diferente para cada malha considerada (menor para malhas mais grossas).

Neste trabalho a paralelização do método *multigrid* foi realizada adotando-se a estratégia da decomposição de dados através do particionamento do domínio em subdomínios, ou seja, considera-se que existe apenas um único sistema de equações para todos os subdomínios, e portanto, cada subdomínio pode ser resolvido com o mesmo algoritmo.

2 MODELO MATEMÁTICO

O modelo matemático considerado refere-se ao escoamento laminar bidimensional de um fluido incompressível governado pelas equações de Navier-Stokes (Kim e Moin, 1985):

$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial u}{\partial t} + \mathbf{u} \cdot \nabla u = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}} \nabla^2 u. \quad (1)$$

$$\frac{\partial v}{\partial t} + \mathbf{u} \cdot \nabla v = -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}} \nabla^2 v$$

Na Eq. (1) x e y são as coordenadas espaciais (variáveis independentes), t é a coordenada temporal (variável independente), u e v são as componentes do vetor velocidade \mathbf{u} nas direções x e y , respectivamente, p é a pressão do fluido e Re é o número de Reynolds. Cada termo da Eq. (1) foi discretizado utilizando o método dos volumes finitos empregando malhas “desencontradas” (*staggered grid*). Detalhes serão vistos na seção 3.3.

Para testar a metodologia numérica foram resolvidos dois problemas:

Problema 1: Vórtices de Taylor-Green (Pearson, 1964)

Domínio dado por $\{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq \pi \text{ e } 0 \leq y \leq \pi\}$ e solução analítica

$$\begin{aligned} u &= -(\cos x \cdot \text{sen } y) e^{-2(t+Ti)} \\ v &= (\text{sen } x \cdot \cos y) e^{-2(t+Ti)} \\ p &= -\frac{1}{4}(\cos 2x + \text{sen } 2y) e^{-4(t+Ti)} \end{aligned} \quad (2)$$

Na Eq. (2) T_i é o tempo inicial; neste trabalho foi utilizado $T_i = 0$, e $t = kh_t$, onde k é o passo de tempo que se está resolvendo e h_t é o refinamento temporal. As condições iniciais e de contorno são obtidas diretamente da solução analítica.

Problema 2: Cavidade com tampa deslizante (Ghia et al., 1982)

Domínio dado por $\{(x, y) \in R^2 : 0 \leq x \leq 1 \text{ e } 0 \leq y \leq 1\}$ cujas condições de contorno são:

$$\begin{aligned} u(0,1) &= 1 \\ u(0,y) = u(1,y) = u(x,0) &= 0 \\ v(0,y) = v(1,y) = v(x,0) = v(x,1) &= 0 \end{aligned} \quad (3)$$

O Problema 2 não possui solução analítica, no entanto as soluções numéricas apresentadas por Ghia et al. (1982) são amplamente utilizadas na literatura como referência na comparação dos resultados numéricos.

3 METODOLOGIA

3.1 Métodos de projeção

Guermond et al. (2006) fazem uma ampla revisão dos métodos de projeção disponíveis na literatura. Discutem os principais métodos de projeção, as taxas de convergência e a relação destes com outros métodos de projeção. Apresentam uma generalização dos métodos de projeção com correção na pressão que para certa combinação de parâmetros ($q = 2$, $r = 2$ e $\chi = 1$), resulta no esquema descrito a seguir:

1. Primeiro passo:

$$\begin{aligned} \frac{(3\mathbf{u}^t - 4\mathbf{u}^n + \mathbf{u}^{n-1})}{2h_t} &= \beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \nu \nabla^2 \mathbf{u}^t + 2\nabla P^n - \nabla P^{n-1} \\ \mathbf{u}^t|_{\partial\Omega} &= \mathbf{b}^n \end{aligned} \quad (4)$$

onde \mathbf{u}^t é o campo de velocidades auxiliar, \mathbf{u}^n o campo de velocidades no passo de tempo n , h_t é o refinamento temporal, P^n é pressão no passo de tempo n e \mathbf{b}^n as condições de contorno no passo de tempo n , $\beta \mathbf{g}$ com β constante representa os termos convectivos $\mathbf{u} \cdot \nabla \mathbf{u}$. A escolha das constantes β_1 e β_0 vai depender de como se quer tratar (explicitamente ou implicitamente) os termos convectivos. Para este trabalho, foi utilizado $\beta_1 = -3/2$ e $\beta_0 = 1/2$.

2. Segundo passo:

$$\left\{ \begin{aligned} \nabla^2 \phi^{n+1} &= \frac{3}{2h_t} \nabla \cdot \mathbf{u}^t \\ \mathbf{u}^{n+1} &= \mathbf{u}^t - \frac{2h_t}{3} \nabla \phi^{n+1} \\ P^{n+1} &= P^{n-1} - P^n + \phi^{n+1} - \nu \nabla \cdot \mathbf{u}^t \end{aligned} \right. \quad (5)$$

Na seção de resultados serão apresentadas evidências numéricas que o esquema resultante da combinação $q = 2$, $r = 2$ e $\chi = 1$ é de segunda ordem em relação às normas L_2 , L_1 e L_∞ .

3.2 Tipo de malha utilizada

Neste trabalho foram utilizadas malhas “desencontradas” (*staggered grid*) do tipo Harlow e Welch (1965), também utilizadas por Maliska (2004), onde as velocidades são posicionadas nas faces e a pressão no centro dos volumes. A razão disso é para evitar instabilidades numéricas na solução da pressão (Shih et al, 1989).

A Fig. 1 ilustra uma malha de 6 por 6 volumes, nela é apresentada a ordenação que será adotada ao longo do trabalho. Na Fig. 1 os volumes com linha tracejada são chamados de volumes fictícios e não pertencem ao domínio físico do problema. Além disso, pode-se observar que utilizando a malha desencontrada do tipo Harlow e Welch (1965), as condições de contorno para a variável u são automaticamente prescritas nos contornos oeste e leste. Nos contornos norte e sul é necessário algum tipo de procedimento para que os volumes fictícios levem em consideração a informação dos contornos; neste trabalho utilizou-se extrapolação linear. De forma similar ao caso do variável u , nota-se, na Fig. 1, que as condições de contorno na variável v são automaticamente prescritas nos contornos sul e norte mas nos contornos oeste e leste foi utilizada extrapolação linear.

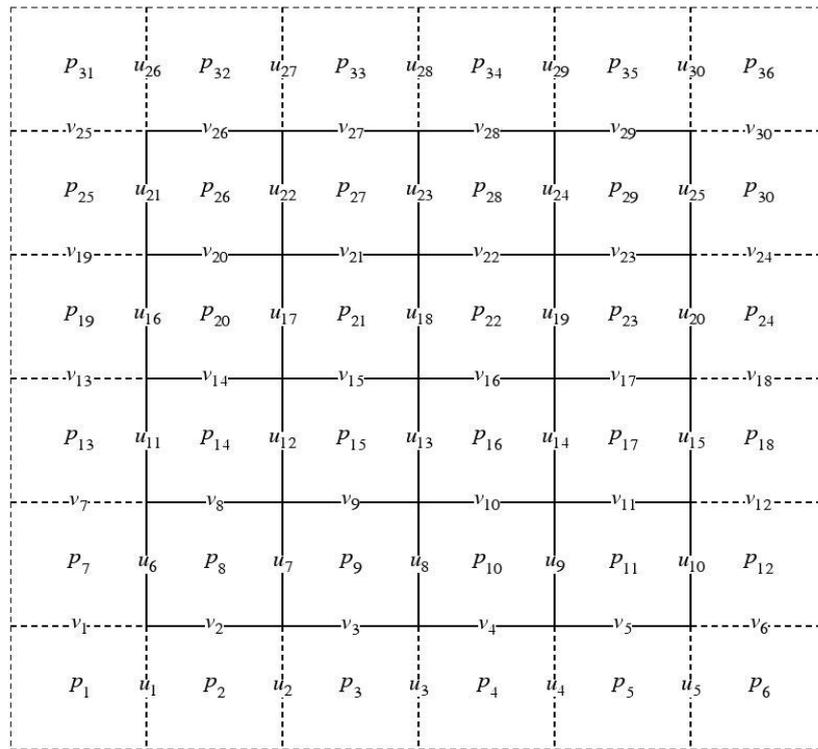


Figura 1. Malha e ordenação lexicográfica das variáveis pressão e velocidade

3.3 Discretização espacial por volumes finitos

A Eq. (1) pode ser reescrita como:

Equação da conservação da massa (massa)

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS = 0. \tag{6}$$

Equação da conservação da quantidade de movimento linear na direção x (QML- x)

$$\frac{\partial}{\partial t} \int_V u dV = -\oint_S u \mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_x dS + \frac{1}{\text{Re}} \oint_S \nabla u \cdot \mathbf{n} dS. \quad (7)$$

Equação da conservação da quantidade de movimento linear na direção y (QML- y)

$$\frac{\partial}{\partial t} \int_V v dV = -\oint_S v \mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_y dS + \frac{1}{\text{Re}} \oint_S \nabla v \cdot \mathbf{n} dS. \quad (8)$$

Cada termo das Eqs. (6), (7) e (8) foram discretizados utilizando o método dos volumes finitos empregando malhas “desencontradas” calculando-se as integrais sobre os volumes de controle definidos na Fig 2. Por questão de simplicidade foi utilizado o mesmo refinamento espacial em todas as direções, ou seja, $h_x = h_y = h$ (malhas quadrangulares). A discretização temporal não será apresentada, pois é dependente do método de projeção utilizado como pode ser observado na Eq. (4); mais detalhes podem ser encontrados em Guermond et al. (2006).

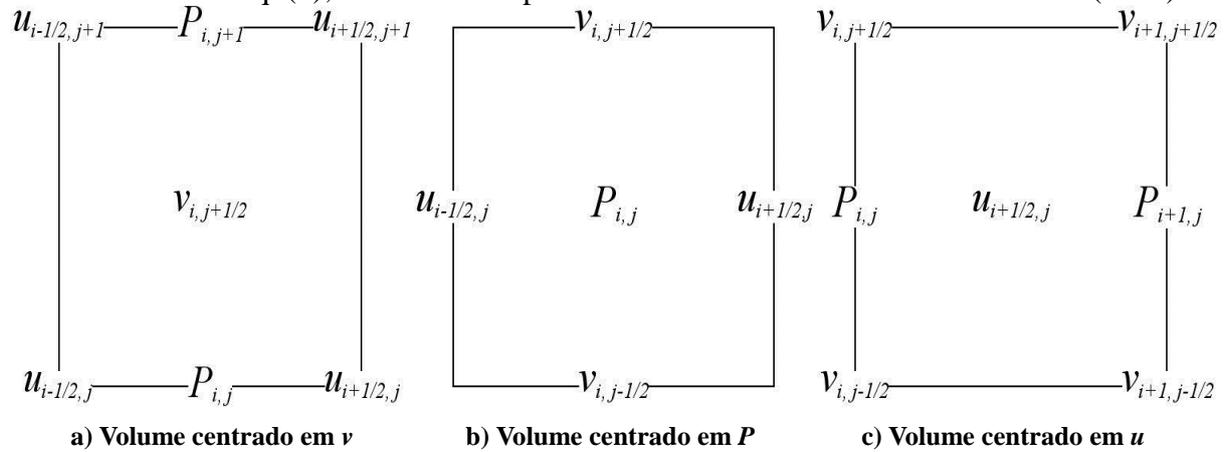


Figura 2. Volumes de controle

A equação da conservação da massa, Eq. (6), é discretizada utilizando a Fig. 2 (b), ou seja, um volume centrado em P

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS \approx u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1} + v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1} = 0. \quad (9)$$

Os termos das pressões são discretizados utilizando os volumes definidos nas Figs. 2 (a) e (c) de acordo com

$$\begin{aligned} \oint_S p n_x dS &\approx (p_{i+1,j} - p_{i,j}) h \\ \oint_S p n_y dS &\approx (p_{i,j+1} - p_{i,j}) h \end{aligned} \quad (10)$$

Utilizando os volumes de controle definidos pelas Figs. 2 (a) e (c) pode-se discretizar os termos difusivos com

$$\begin{aligned} \oint_S \nabla u \cdot \mathbf{n} dS &\approx u_{i+3/2,j}^n + u_{i-1/2,j}^n + u_{i+1/2,j+1}^n + u_{i+1/2,j-1}^n - 4u_{i+1/2,j}^n \\ \oint_S \nabla v \cdot \mathbf{n} dS &\approx v_{i,j+3/2}^n + v_{i,j-1/2}^n + v_{i+1,j+1/2}^n + v_{i-1,j+1/2}^n - 4v_{i,j+1/2}^n \end{aligned} \quad (11)$$

Os termos advectivos são discretizados utilizando os volumes de controle definidos nas Figs. 2 (a) e (c), ou seja, volumes centrados nas componentes das velocidades

$$\begin{aligned} \oint_S \mathbf{u} \cdot \mathbf{n} dS &\approx \left((u^2)_{i+1,j}^n - (u^2)_{i,j}^n + (uv)_{i+1/2,j+1/2}^n - (uv)_{i+1/2,j-1/2}^n \right) h \\ \oint_S \mathbf{v} \cdot \mathbf{n} dS &\approx \left((v^2)_{i,j+1}^n - (v^2)_{i,j}^n + (uv)_{i+1/2,j+1/2}^n - (uv)_{i-1/2,j+1/2}^n \right) h \end{aligned} \quad (12)$$

Na Eq. (13) são apresentados alguns exemplos das aproximações para as componentes das velocidades utilizadas na Eq. (12):

$$\begin{aligned} (u^2)_{i,j}^n &= \left(\frac{1}{2} (u_{i+1/2,j}^n + u_{i-1/2,j}^n) \right)^2 \\ (uv)_{i+1/2,j+1/2}^n &= \left(\frac{1}{2} (u_{i+1/2,j}^n + u_{i+1/2,j+1}^n) \right) \left(\frac{1}{2} (v_{i,j+1/2}^n + v_{i+1,j+1/2}^n) \right) \\ (v^2)_{i,j}^n &= \left(\frac{1}{2} (v_{i,j+1/2}^n + v_{i,j-1/2}^n) \right)^2 \end{aligned} \quad (13)$$

3.4 Restrição no passo de tempo

Se o passo de tempo, h_t , for muito grande em relação ao refinamento espacial (h_x e h_y) as simulações podem se tornar instáveis. O critério utilizado neste trabalho para evitar instabilidades é dado por (Maliska, 2004):

$$h_t < \frac{1}{2 \text{Re}} \frac{h_x^2 h_y^2}{h_x^2 + h_y^2}. \quad (14)$$

3.5 Condição de integrabilidade

Uma característica comum dos métodos de projeção é que estes impõem condições de contorno de Neumann homogêneas sobre a pressão nos contornos. Devido a estas condições de contorno, a seguinte condição de integrabilidade precisa ser satisfeita para garantir a existência e unicidade da solução (Briggs et al., 2000):

$$\int_{\Omega} \nabla^2 \phi^{n+1} = \int_{\Omega} \frac{3}{2h_t} \nabla \cdot \mathbf{u}^n = 0. \quad (15)$$

Para detalhes de como a Eq. (15) é satisfeita a cada passo de tempo, ver o procedimento proposto por Villar (2007).

3.6 Método multigrid

O método *multigrid* consiste, basicamente, em aplicar algum método iterativo básico, por exemplo os métodos de Jacobi e Gauss-Seidel (Briggs et al., 2000; Trottenberg et al., 2001), em uma dada malha até que as componentes do erro tornem-se suaves. Na sequência, deve-se transferir o “problema” para uma malha mais grossa (onde os erros suavizados se tornam mais oscilatórios) e transferir o “problema” da malha mais grossa para a malha mais fina melhorando a solução da malha fina com o erro suavizado da malha grossa. Este procedimento para duas malhas pode ser facilmente estendido para diversas malhas e ele pode ser repetido até que a malha mais grossa (ou conveniente) seja atingida.

Considerando malhas uniformes, a razão de engrossamento é definida como $r = H/h$, onde h representa o tamanho dos elementos da malha fina Ω^h e H o tamanho dos elementos da malha grossa Ω^H . Neste trabalho será utilizado $r = 2$. O número de malhas que podem ser visitadas durante o processo de engrossamento depende do problema considerado e da razão de engrossamento utilizada.

A Fig. 3 mostra o processo de engrossamento de uma malha de 32×32 volumes de tamanho h (em cada direção coordenada) e razão de engrossamento $r = 2$, ou seja, $H = 2h$. Neste caso, $L = L_{\max} = 5$ níveis de malha.

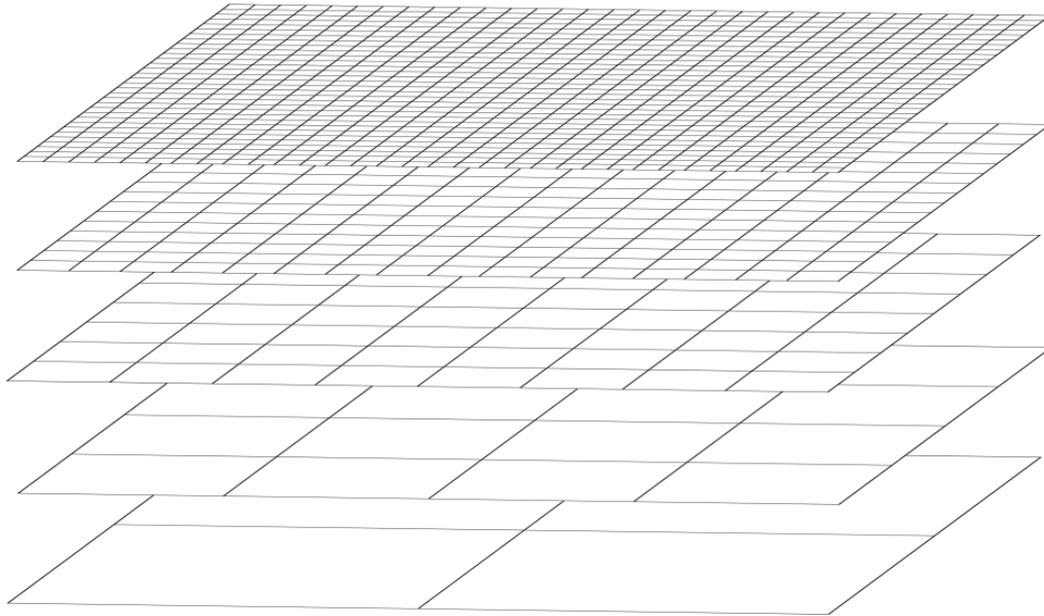


Figura 3. Processo de engrossamento de uma malha uniforme

A ordem na qual as malhas são visitadas é chamada de ciclo *multigrid*. Existem vários tipos de ciclos *multigrid* tais como o ciclo V, o ciclo W, o ciclo dente-de-serra, entre outros. Neste trabalho foi utilizado o ciclo V.

3.7 Processo de restrição

Os operadores que transferem informações da malha fina Ω^h para a malha grossa Ω^{2h} são chamados de operadores de restrição. Estes operadores “convertem” vetores da malha fina \mathbf{v}^h em vetores da malha grossa \mathbf{v}^{2h} .

Um operador muito conhecido e utilizado neste trabalho é o operador de restrição por ponderação completa (Villar, 2007), que para o caso da pressão é dado por

$$\phi_{i,j}^{2h} = \frac{(\phi_{i,j}^h + \phi_{i+1,j}^h + \phi_{i,j+1}^h + \phi_{i+1,j+1}^h)}{4}, \quad (16)$$

onde $\phi_{i,j}^{2h}$ é o valor da pressão na posição i, j da malha grossa Ω^{2h} . De forma similar $\phi_{i,j}^h$ é o valor da pressão na posição i, j da malha fina Ω^h .

Para o caso das velocidades u e v foi utilizado o procedimento apresentado por Vanka (1986)

$$\begin{aligned} u_{i,j}^{2h} &= \frac{(u_{i,j}^h + u_{i,j+1}^h)}{2} \\ v_{i,j}^{2h} &= \frac{(v_{i,j}^h + v_{i+1,j}^h)}{2}. \end{aligned} \quad (17)$$

3.8 Processo de prolongação

Os operadores que transferem informações da malha grossa Ω^{2h} para a malha fina Ω^h são chamados de operadores de prolongação, também conhecidos como interpolação. Estes operadores “convertem” vetores da malha grossa \mathbf{v}^{2h} em vetores da malha fina \mathbf{v}^h .

Um dos operadores de prolongação mais utilizados é o operador de interpolação bilinear. Para o caso da pressão foi utilizado o procedimento apresentado em Villar (2007)

$$\left\{ \begin{aligned} \phi_{i,j}^h &= \frac{9\phi_{i,j}^{2h} + 3\phi_{i-1,j}^{2h} + 3\phi_{i,j-1}^{2h} + \phi_{i-1,j-1}^{2h}}{16} \\ \phi_{i+1,j}^h &= \frac{9\phi_{i,j}^{2h} + 3\phi_{i+1,j}^{2h} + 3\phi_{i,j-1}^{2h} + \phi_{i+1,j-1}^{2h}}{16} \\ \phi_{i,j+1}^h &= \frac{9\phi_{i,j}^{2h} + 3\phi_{i-1,j}^{2h} + 3\phi_{i,j+1}^{2h} + \phi_{i-1,j+1}^{2h}}{16} \\ \phi_{i+1,j+1}^h &= \frac{9\phi_{i,j}^{2h} + 3\phi_{i+1,j}^{2h} + 3\phi_{i,j+1}^{2h} + \phi_{i+1,j+1}^{2h}}{16} \end{aligned} \right. \quad (18)$$

Para o caso das variáveis u e v as equações de interpolação utilizadas seguem os procedimentos utilizados por Villar (2007)

$$\left\{ \begin{aligned} u_{i,j}^h &= \frac{3u_{i,j}^{2h} + u_{i,j-1}^{2h}}{4} \\ u_{i-1,j}^h &= \frac{3u_{i,j}^{2h} + 3u_{i-1,j}^{2h} + u_{i,j-1}^{2h} + u_{i-1,j-1}^{2h}}{8} \\ u_{i+1,j}^h &= \frac{3u_{i,j}^{2h} + 3u_{i+1,j}^{2h} + u_{i,j-1}^{2h} + u_{i+1,j-1}^{2h}}{8} \\ u_{i,j+1}^h &= \frac{3u_{i,j}^{2h} + u_{i,j+1}^{2h}}{4} \\ u_{i-1,j+1}^h &= \frac{3u_{i,j}^{2h} + 3u_{i-1,j}^{2h} + u_{i,j+1}^{2h} + u_{i-1,j+1}^{2h}}{8} \\ u_{i+1,j+1}^h &= \frac{3u_{i,j}^{2h} + 3u_{i+1,j}^{2h} + u_{i,j+1}^{2h} + u_{i+1,j+1}^{2h}}{8} \end{aligned} \right\}, \quad \left\{ \begin{aligned} v_{i,j}^h &= \frac{3v_{i,j}^{2h} + v_{i-1,j}^{2h}}{4} \\ v_{i,j+1}^h &= \frac{3v_{i,j}^{2h} + 3v_{i,j+1}^{2h} + v_{i-1,j}^{2h} + v_{i-1,j+1}^{2h}}{8} \\ v_{i,j-1}^h &= \frac{3v_{i,j}^{2h} + 3v_{i,j-1}^{2h} + v_{i-1,j}^{2h} + v_{i-1,j-1}^{2h}}{8} \\ v_{i+1,j}^h &= \frac{3v_{i,j}^{2h} + v_{i+1,j}^{2h}}{4} \\ v_{i+1,j-1}^h &= \frac{3v_{i,j}^{2h} + 3v_{i,j-1}^{2h} + v_{i+1,j}^{2h} + v_{i+1,j-1}^{2h}}{8} \\ v_{i+1,j+1}^h &= \frac{3v_{i,j}^{2h} + 3v_{i,j+1}^{2h} + v_{i+1,j}^{2h} + v_{i+1,j+1}^{2h}}{8} \end{aligned} \right. \quad (19)$$

3.9 Paralelização do método multigrid

A paralelização do método *multigrid* foi realizada adotando-se a estratégia da decomposição de dados através do particionamento do domínio em subdomínios, ou seja, considera-se que existe apenas um único sistema de equações para todos os subdomínios, e

portanto, cada subdomínio pode ser resolvido com o mesmo algoritmo. O modelo de paralelização que aplica o mesmo programa em cada um dos subdomínios é conhecido como SPMD (*Single Program Multiple Data*) (Chapman et al., 2008).

O particionamento do domínio consiste em dividir o domínio em subdomínios de forma que cada thread (processador físico ou lógico) resolva uma parcela do sistema de equações que define o problema em um menor intervalo de tempo.

O método de particionamento do domínio deste trabalho está diretamente relacionado com a ordenação lexicográfica dos volumes do domínio. Como são considerados apenas modelos com condições de contorno de Dirichlet, ou seja, modelos com solução conhecida nos contornos, o número de volumes de cada subdomínio pode ser calculado como

$$N_{sub} = \left\lfloor \frac{(N_x - 2)(N_y - 2)}{\bar{T}} \right\rfloor, \quad (20)$$

onde \bar{T} é o número total de *threads* disponíveis na máquina. O símbolo $\lfloor \cdot \rfloor$ refere-se à função piso.

A Eq. (20) distribui o número de volumes no interior do domínio entre os *threads* disponíveis de forma que o número máximo de volumes que podem “sobrar” na divisão é $\bar{T} - 1$, e estes, podem ser redistribuídos entre os *threads* de forma que a diferença entre o número de volumes entre os subdomínios não seja maior que um. A escolha de quais *threads* receberão os volumes que “sobram” é arbitrária, desde que cada *thread* receba apenas 1 volume extra (balanceamento da carga). Neste trabalho optou-se por redistribuir os nós entre os *threads* $t \geq t_c$, onde

$$t_c = \bar{T} + 1 - \left[(N_x - 2)(N_y - 2) - \bar{T}N_{sub} \right]. \quad (21)$$

Os índices que particionam o domínio em subdomínios, em relação à ordenação lexicográfica, são dados por

$$\begin{aligned} i_t &= (t-1)N_{sub} + 1 + C(t-t_c) \\ f_t &= tN_{sub} + C(t-t_c + 1) \end{aligned}, \quad (22)$$

onde i_t representa o início e f_t representa o fim de cada subdomínio. O termo C é calculado com

$$C = \begin{cases} 1 & \text{se } t \geq t_c \\ 0 & \text{se } t < t_c \end{cases}. \quad (23)$$

Desta forma cada *thread* a partir de t_c (este *thread* incluso) receberá 1 volume dos que sobraram na divisão do domínio com a Eq. (20).

A Fig. 4 ilustra um domínio com 49 volumes distribuídos, utilizando a Eq. (22), entre: (a) 4 *threads* e (b) 8 *threads*. As formas geométricas representam os subdomínios. Pode-se observar nas duas situações, (a) e (b), que a diferença no número de volumes entre os subdomínios não passa de um.

Na avaliação da paralelização do método *multigrid* foram utilizados os parâmetros tempo de CPU (t_{CPU}), medido com a função `OMP_GET_WTIME()`, e o fator *speed-up* (S_p),

interpretado como uma medida do aumento na velocidade, ou “aceleração”, do programa, definido como (Chapman et al., 2008)

$$S_p = \frac{t_1}{t_p}, \tag{24}$$

onde t_1 é o tempo de CPU utilizando apenas 1 processador e t_p é o tempo de CPU utilizando P processadores (tempo de execução em paralelo do programa).

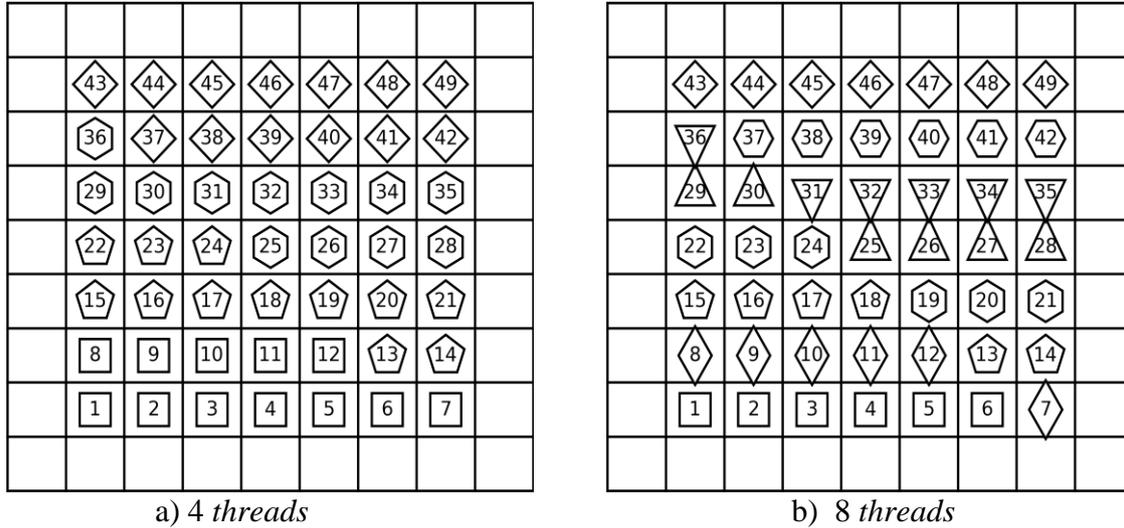


Figura 4. Domínio com 49 volumes particionado em (a) 4 subdomínios e (b) 8 subdomínios

Na literatura (Sun e Chen, 2010) é comum encontrar o *speed-up* teórico máximo, conhecido como lei de Amdahl (Amdahl, 1967), em sua forma mais genérica como

$$S_{\text{Amdahl}} = \frac{1}{(1-f) + \frac{f}{m}}, \tag{25}$$

onde f representa a parcela do processamento que pode ser melhorada por um fator m e o restante $(1-f)$, a parcela que não pode ser melhorada. No contexto do paralelismo, f representa a parcela do trabalho que pode ser paralelizada e m , o número e processadores utilizados.

4 RESULTADOS

4.1 Dados de implementação

Nesta seção são apresentados os parâmetros empregados no método *multigrid* padrão utilizado neste trabalho.

O tipo de ciclo utilizado é o V. O processo de restrição é feito empregando-se as Eqs (16) e (17). O processo de prolongação é realizado empregando-se as Eqs (18) e (19). A razão de engrossamento das malhas (r) é a razão de engrossamento padrão, ou seja, $r = 2$ (Briggs et al., 2000).

O suavizador (*solver*) utilizado foi o método de Gauss-Seidel *red-black*. Neste trabalho, o número de iterações internas empregadas na pré-suavização, ν_1 , é igual ao número de iterações internas utilizadas na pós-suavização, ν_2 . Neste trabalho utilizou-se $\nu_1 = \nu_2 = 3$ (Oliveira, 2010).

A cada passo de tempo são resolvidas numericamente, com *multigrid* ou *singlegrid* (solução obtida pelo suavizador utilizando-se apenas uma malha), as velocidades e a pressão. O critério de parada usado para interromper o processo iterativo de cada uma das variáveis (velocidades e pressão) é a norma euclidiana do resíduo adimensionalizada pela norma euclidiana do resíduo na estimativa inicial (Briggs et al., 2000; Trottenberg et al., 2001), ou seja, $\|\mathbf{r}^{(i)}\|_2 / \|\mathbf{r}^{(0)}\|_2$, onde o termo $\mathbf{r}^{(i)}$ é o resíduo na iteração i e $\mathbf{r}^{(0)}$, o resíduo na estimativa inicial. Entre os trabalhos que usam essa norma pode-se citar: Briggs et al. (2000); Oliveira (2010) e Trottenberg et al. (2001). O processo iterativo é interrompido quando a norma é menor ou igual à tolerância ε , isto é $\|\mathbf{r}^{(i)}\|_2 / \|\mathbf{r}^{(0)}\|_2 \leq \varepsilon$. Neste trabalho foi utilizado $\varepsilon = 10^{-9}$.

Em todas as simulações, o método *multigrid* partiu da malha mais fina, nível 1, e foi até a malha mais grossa possível, nível L_{\max} .

Para avaliar o fator *speed-up* devido à paralelização é necessário medir o tempo de CPU. Este tempo é medido em segundos (s) com o uso da função `OMP_GET_WTIME()`. Entende-se por tempo de CPU o tempo gasto para realizar a geração de malhas, atribuir a estimativa inicial, calcular os coeficientes e resolver o sistema linear até atingir a tolerância estabelecida para o primeiro passo de tempo na comparação *multigrid versus singlegrid*. Como os tempos de CPU entre dois passos de tempo são muito próximos e assumindo o princípio da proporcionalidade, pode-se avaliar o tempo de CPU considerando-se apenas o tempo medido no primeiro passo de tempo. Na avaliação do paralelismo, até atingir a tolerância estabelecida para 10 passos de tempo. Não se considera o tempo de CPU no último passo de tempo porque o número de passos de tempo é variável e depende do refinamento temporal utilizado (h_t).

Os algoritmos foram implementados na linguagem Fortran 2003 utilizando o compilador GNU Fortran (GFortran), versão 4.6.3, com a opção OpenMP e precisão dupla. Os testes foram realizados em um servidor disponibilizados pelo Sistema Meteorológico do Paraná-SIMEPAR. O servidor utilizado está equipado com processadores Intel Xeon E5-2690 de 2,9 GHz e totalizando 32 processadores, dos quais 14 estiveram disponíveis para testes; 32 GB de memória RAM e sistema operacional Linux 64 bits.

4.2 Avaliação das ordens de convergência

Para avaliação do erro numérico foi utilizado o Problema 1 que possui solução analítica conhecida.

Se o erro em uma solução numérica é causado exclusivamente por erros de truncamento então a quantidade $E_h = \Phi - \phi$, onde Φ representa a solução analítica de uma variável e ϕ a solução numérica, é conhecida como erro de discretização (E_h) ou erro numérico (Ferziger e Peric, 2001).

Para Marchi (2001) a ordem efetiva do erro (P_E) pode ser entendida como a inclinação local do erro de discretização (E_h) em relação ao refinamento da malha. A ordem efetiva do

erro é importante pois permite verificar se à medida que a malha é refinada a ordem do erro de discretização tende à ordem assintótica dos erros de truncamento.

Foi verificado se a ordem efetiva do erro (P_E) tende para a ordem assintótica (P_L) à medida que a malha é refinada. As discretizações utilizadas em todos os algoritmos e em todas as variáveis são de segunda ordem, portanto, espera-se que $P_E \rightarrow P_L = 2$.

Para o cálculo da ordem efetiva do erro foi utilizada a equação

$$P_E = \frac{\log\left(\frac{\Phi - \phi_2}{\Phi - \phi_1}\right)}{\log(q)}, \quad (26)$$

onde Φ é a solução analítica do problema, ϕ_2 e ϕ_1 são soluções numéricas de duas malhas e $q = h_2/h_1 = 2$, h_2 representa o comprimento da aresta de cada volume quadrangular da malha grossa e h_1 o comprimento da aresta de cada volume quadrangular na malha fina. Em todos os casos as variáveis analisadas foram (u, v, P) , em $(x, y, t) = (1/2, 1/2, Tf)$, ou seja, a solução no ponto central do domínio no último passo de tempo.

Na Fig. 5 pode-se observar que a ordem (P_E) do erro de todas as variáveis primitivas tende à ordem assintótica esperada ($P_L = 2$).

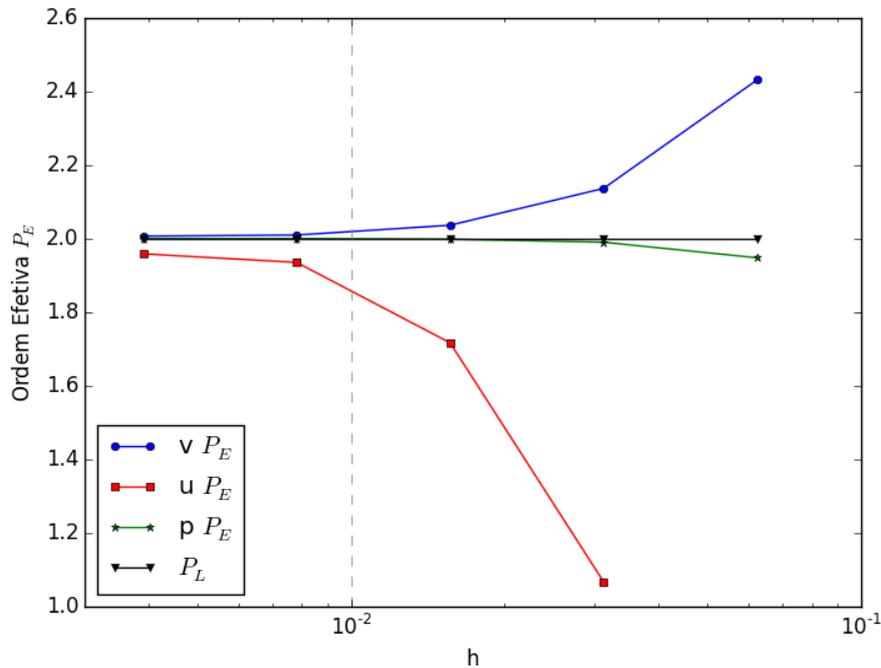


Figura 5. Ordem efetiva do erro para as velocidades e a pressão

Das Figs. 6, 7 e Tab 1 pode-se observar que o método estudado é de segunda ordem, para todas variáveis estudadas, em relação a todas as normas empregadas. Os resultados apresentados na Fig. 6 são importantes no contexto dos métodos de projeção porque a acurácia destes é prejudicada devido à imposição de condições de contorno de Neumann artificiais (não necessariamente obedecidas pelo fluido real) sobre a pressão. Da Fig. 6 conclui-se que os erros introduzidos pela imposição de condições de Neumann pelo método de projeção utilizado é menor, ou de mesma ordem, que os erros de discretização obtidos.

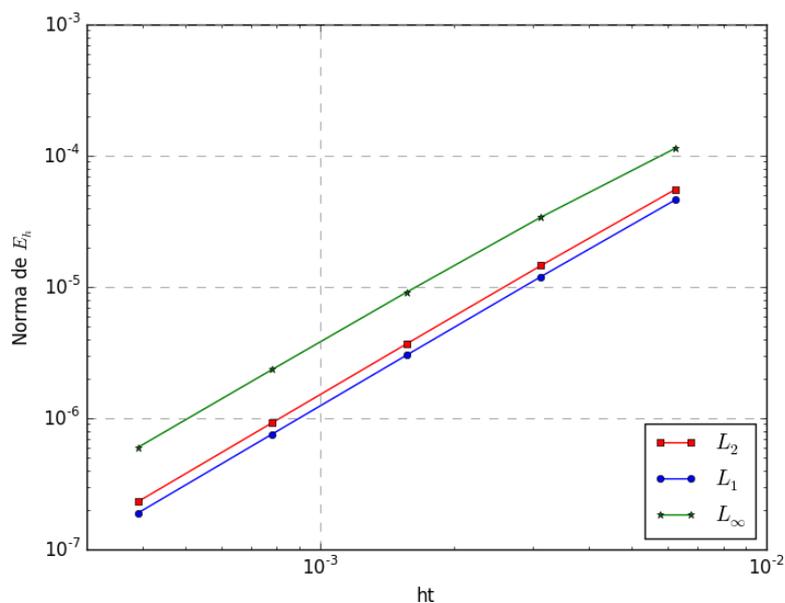


Figura 6. Taxa de convergência das normas em relação ao tempo para pressão

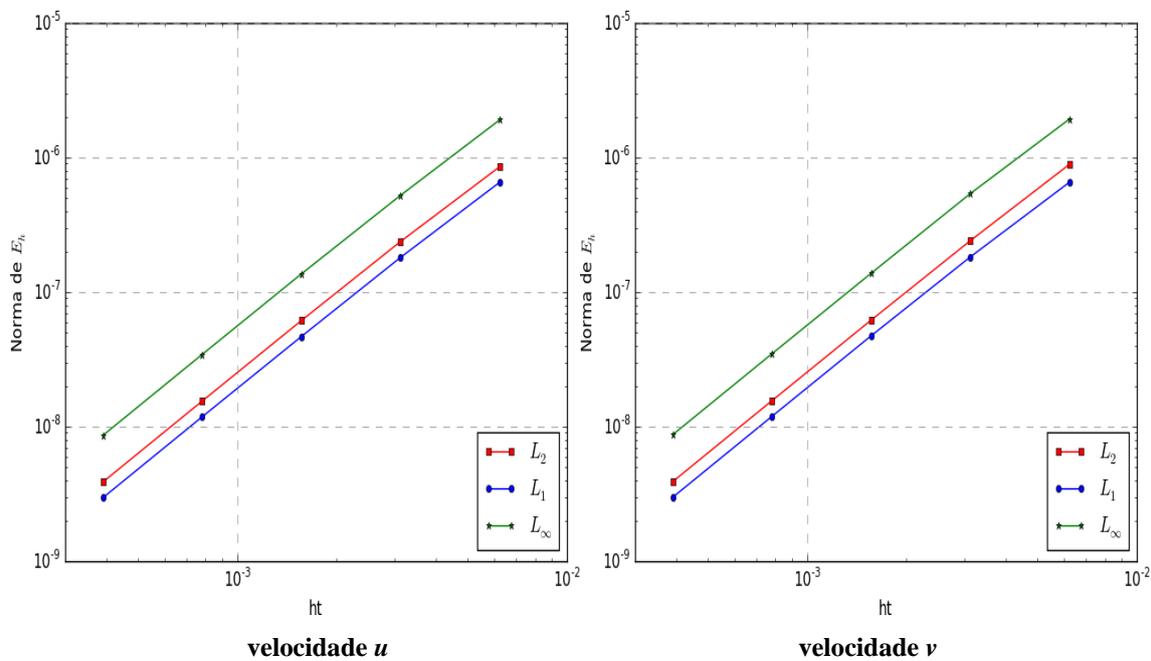


Figura 7. Normas das variáveis primitivas u e v

Tabela 1. Taxas de convergência observadas

	L_1	L_2	L_∞
P	1,995	1,993	1,944
u	1,978	1,980	1,977
v	1,979	1,983	1,984

4.3 Avaliação dos resultados para o Problema 2

Apesar deste problema não ter solução analítica para comparações diretas, em Ghia et al. (1982) podem ser encontrados resultados numéricos para comparação amplamente utilizados na literatura. Como não é conhecido o momento a partir do qual o problema da cavidade com tampa deslizante (problema 2) atinge o regime permanente, não é possível definir um tempo final (T_f) para este problema, logo, para interromper o processo iterativo foi considerado que a maior diferença em módulo entre uma variável no passo de tempo atual e anterior deve ser menor que 10^{-6} , ou seja, $\max(|\varphi^{n+1} - \varphi^n|) \leq 10^{-6}$, onde φ representa a pressão ou velocidades u e v . O processo só foi interrompido quando todas as variáveis atingiram a tolerância estabelecida.

Na Fig. 8 são apresentadas as simulações para o Problema 2, Ghia et al. (1982). O refinamento de malha considerado foi de 128×128 e Reynolds variando de 100 até 3200. Pode-se observar boa concordância entre os resultados. Pode-se observar também que há uma forte dependência dos perfis das velocidades em relação ao número de Reynolds. Além disso, foi observado que o tempo necessário para atingir a tolerância aumenta com o número de Reynolds, sendo necessários muitos passos de tempo para Reynolds maiores que 1000. Visto que o tempo de processamento cresce significativamente, para números de Reynolds maiores que 1000, foi utilizada a versão paralelizada do *multigrid*.

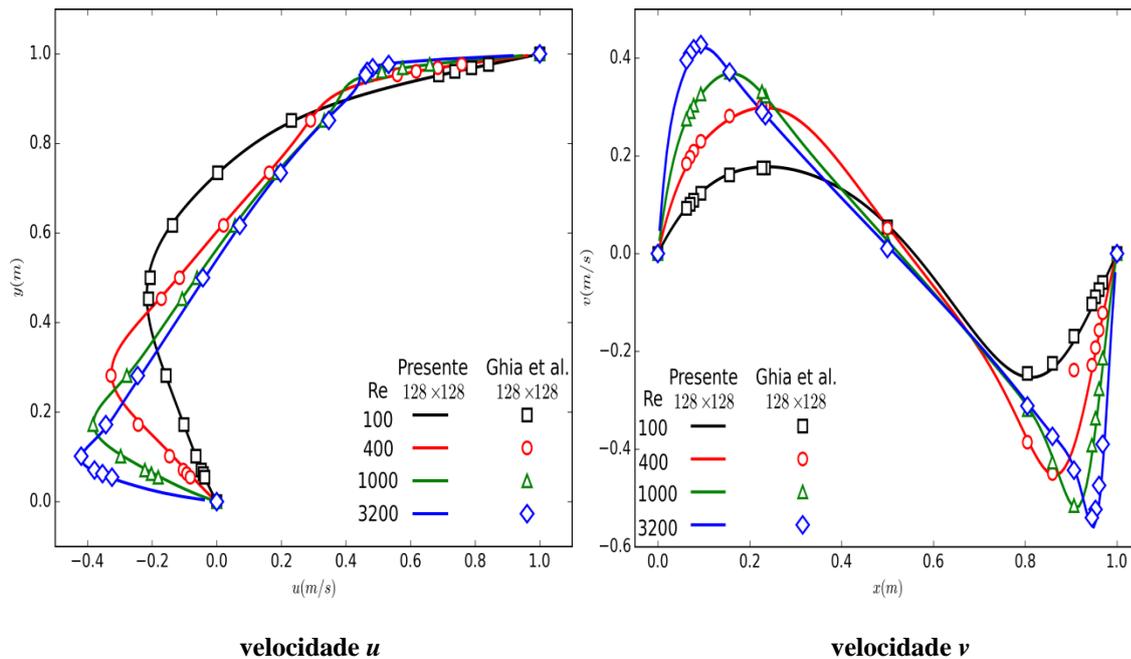


Figura 8. Perfis das velocidades u e v em $x=0,5$ e $y=0,5$ para malhas de 128×128 volumes

4.4 Multigrid versus singlegrid

Nesta seção são apresentados os resultados do *multigrid versus singlegrid*. Basicamente foi utilizado o método iterativo Gauss-Seidel *red-black* para o *singlegrid* e como suavizador para o método *multigrid*.

Para gerar o tempo de CPU foi resolvido o Problema 1 para Reynolds igual a 100 com ambos os métodos e considerado apenas o primeiro passo de tempo (lembrando que h_t depende do refinamento espacial utilizado de acordo com as restrições ao passo de tempo). A

solução numérica depende do tempo e diferentes refinamentos de tempo podem gerar diferentes números de passos de tempo, logo, para que a comparação tenha sentido foi realizado apenas um passo de tempo para vários tamanhos de malha. Foram realizadas pelo menos 5 simulações (repetições) de cada experimento numérico com as respectivas coletas de tempo de CPU para cada tamanho de malha. O tempo apresentado na Fig. 9 é uma média destes tempos (esta figura está na escala log log). Além disso, foi utilizado o mesmo critério de parada para ambos os métodos: norma L_2 do resíduo da iteração atual dividida pela norma L_2 do resíduo na primeira iteração deve ser menor ou igual a uma tolerância, isto é $\|\mathbf{r}^{(i)}\|_2 / \|\mathbf{r}^{(0)}\|_2 \leq \varepsilon$. Neste trabalho foi utilizado $\varepsilon = 10^{-9}$.

Na Tab. 2 são apresentados os tempos de CPU do *multigrid* (serial), do *singlegrid* e o fator *speed-up* (aceleração do *multigrid* em relação ao *singlegrid*). Pode-se observar um fator *speed-up* significativo do *multigrid* em relação ao *singlegrid*. Pode-se observar ainda que o fator *speed-up* aumenta com o refinamento da malha, uma propriedade desejável.

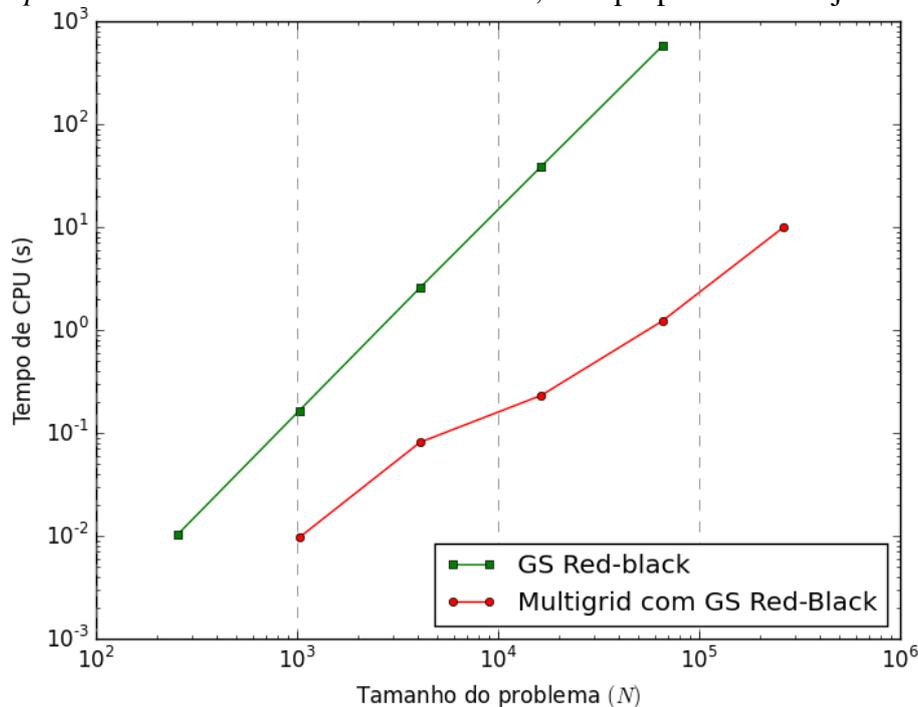


Figura 9. Tempo de CPU para *multigrid* versus *singlegrid* para Reynolds igual a 1

Tabela 2. Tempo de CPU *Multigrid* (MG) versus *singlegrid* (SG) para Reynolds igual 1

Malha	<i>Multigrid</i>	<i>Singlegrid</i>	<i>Speed-up</i>
32×32	0,01	0,16	16
64×64	0,08	2,58	32
128×128	0,23	38,8	169
256×256	1,22	572,94	470
512×512	9,91	---	---

Para calcular o esforço computacional (ordem de complexidade) foi realizado o ajuste geométrico dos dados da Tab. 2 considerando a função

$$T_{CPU}(N) = cN^p, \quad (27)$$

onde p representa a ordem do *solver* utilizado (*multigrid* ou *singlegrid*) ou a inclinação das curvas apresentadas na Fig. 9. No caso do *multigrid*, quanto mais próximo p estiver de 1 melhor o desempenho do algoritmo testado. A Tab. 3 apresenta o valor de p calculado utilizando a Eq. (27) e os dados da Tab. 2.

Tabela 3. Esforço computacional dos algoritmos *multigrid* e *singlegrid*

	<i>Multigrid</i>	<i>Singlegrid</i>
p	1,196	1,978

4.5 Avaliação do desempenho paralelo

Nesta seção são apresentados os resultados da paralelização do método *multigrid*. Para a geração dos tempos computacionais foram simulados 10 passos de tempo de forma que os tempos totais de CPU medidos ficaram acima de 1s e, portanto, acima da precisão da função OMP_GET_WTIME() do módulo OMP da linguagem FORTRAN. Além disso, foram feitas pelo menos 5 simulações (repetições) de cada experimento numérico com as respectivas coletas de tempo de CPU para cada tamanho de problema e *thread* utilizada. O tempo de CPU foi considerado como a média destes tempos.

A Fig. 10 mostra o fator *speed-up* para várias malhas. Nela são apresentados o número de processadores no eixo x e o fator *speed-up* (aceleração da versão paralela em relação à versão serial do *multigrid*) no eixo y . Além disso, são apresentadas as curvas de *speed-up* teórico máximo previsto, pela Eq. (25), para programas que têm, hipoteticamente, frações paralelas de 90%, 95% e 99%.

Uma análise da Fig. 10 a), b), c) e d) mostra que o fator *speed-up* encontra-se consideravelmente abaixo do pico máximo teórico de 14. As principais justificativas para estes resultados podem ser resumidas nas seguintes constatações:

1) Na versão do método *multigrid* utilizada, muitos processadores ficam ociosos durante o processamento da malha mais grossa. Para os experimentos realizados, na malha mais grossa, 12 (de 13) processadores ficam ociosos enquanto esperam que 1 processador termine suas tarefas.

2) No método Gauss-Seidel *red-black* apenas metade do domínio pode ser processado em paralelo.

3) O método de particionamento do domínio, apesar do ótimo balanceamento de carga não é ideal, pois produz subdomínios com perímetros consideráveis elevando os custos de comunicação entre os processadores.

Já se encontra em fase de estudos, a solução dos problemas elencados nas constatações 1 e 3 acima. Para o problema elencado pela constatação 2 acima, será necessário utilizar outro suavizador cujo grau de paralelismo seja superior ao Gauss-Seidel *red-black* e com propriedades de suavização igual ou superior. O método de Jacobi, por exemplo, não se enquadra como substituto do método Gauss-Seidel *red-black* pois, apesar de ter um grau de

paralelismo superior (todo o domínio pode ser resolvido em paralelo), sua propriedade de suavização é muito inferior (Trottenberg et al., 2001) e portanto conduz a tempos de CPU maiores.

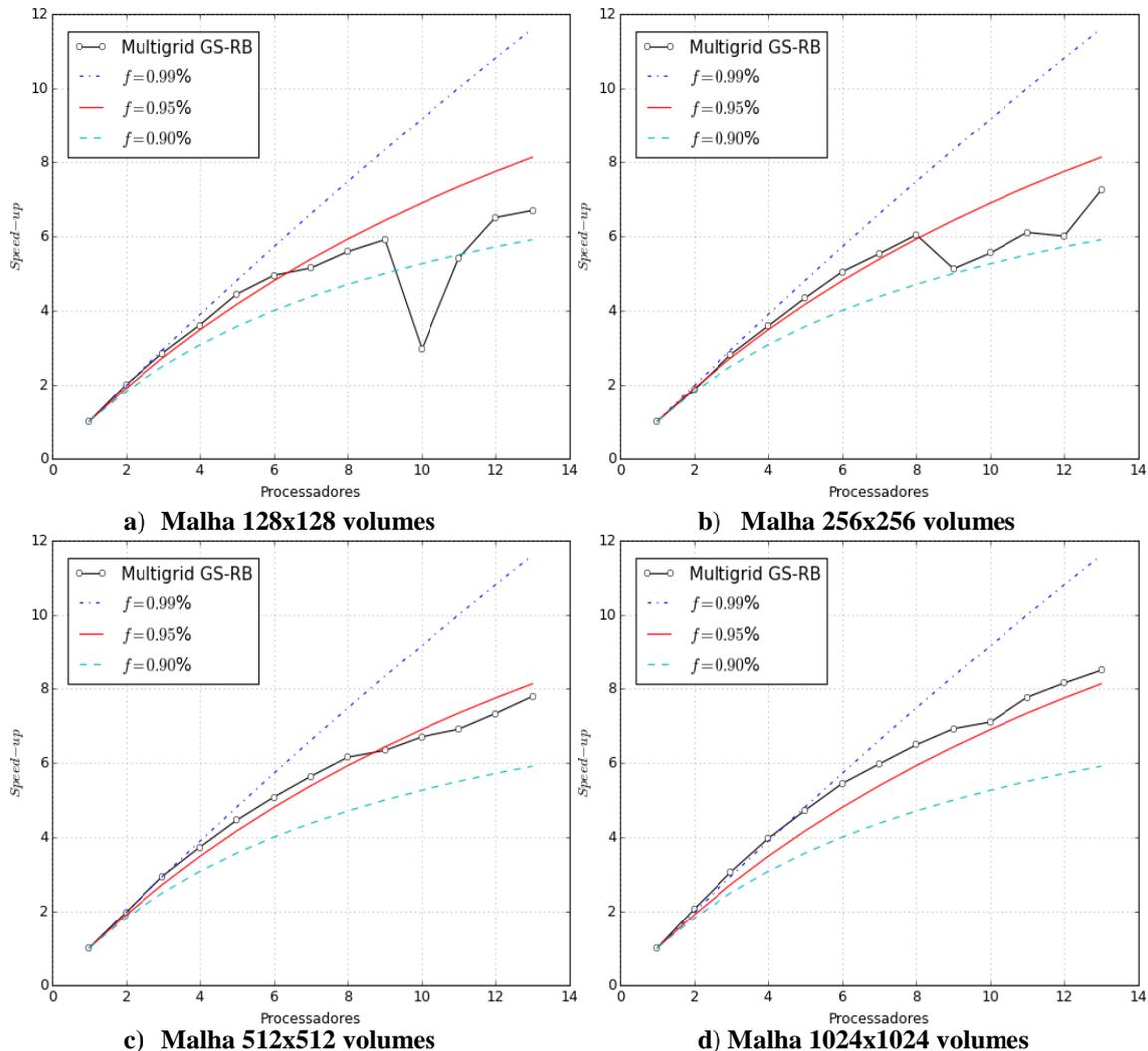


Figura 10. Fator *speed-up* para várias malhas.

5 CONCLUSÃO

Neste trabalho foi apresentado um método de projeção com correção incremental na pressão e convergência temporal de segunda ordem para a resolução das equações de Navier-Stokes na variáveis primárias. Foi utilizado o método dos volumes finitos em malhas cartesianas e com variáveis desencontradas para a discretização das variáveis. Os sistemas lineares gerados foram resolvidos com o método *multigrid* geométrico com esquema CS e ciclo V. O método *multigrid* foi usado em conjunto com técnicas de paralelização. Foram realizados testes numéricos com resultados satisfatórios para um problema com solução analítica conhecida e o problema da cavidade com tampa móvel (Ghia et al., 1982). Foi possível testar a superioridade do *multigrid* em relação à utilização do método de malha única (*singlegrid*). Finalmente, com a paralelização do método *multigrid* foi possível reduzir em até 8 vezes, utilizando 14 processadores, o tempo de CPU necessário para se obter as soluções das equações de Navier-Stokes. Os resultados podem ser considerados positivos, pois

verificou-se que, além do tempo de processamento ter sido reduzido significativamente empregando o *multigrid* paralelizado, este diminuiu à medida que aumentou-se o número de processadores utilizados.

AGRADECIMENTOS

Os autores agradecem o apoio do Sistema Meteorológico do Paraná – SIMEPAR, do PPGMNE (Programa de Pós-Graduação em Métodos Numéricos em Engenharia) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq. O primeiro autor é bolsista do Sistema Meteorológico do Paraná – SIMEPAR.

REFERÊNCIAS

- Amdahl, G. M., 1967. Validity of the single processor approach to achieving large scale. AFIPS spring joint computer conference. IBM Sunnyvale, California: [s.n.], pp. 483–485.
- Bell, J. B., Collela, P., & Glaz, H. M., 1989. A second order projection for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, v. 85, n.2, pp. 257-283.
- Briggs, W., L., Henson, V., E., & McCormick, S., 2000. A Multigrid Tutorial, Second Edition, SIAM, Philadelphia.
- Brown, D. L., Cortez, R., & Minion, M. L., 2001. Accurate projection methods for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, v. 168, n. 2, pp. 464-499.
- Chan, T. F., & Saad, Y., 1986. Multigrid Algorithms on the Hypercube Multiprocessor, *IEEE Trans. Comput*, v. 35, pp. 969-977.
- Chan, T. F., & Schreiber, R., 1985. Parallel Networks for Multigrid Algorithms: Architecture and Complexity, *SIAM J. Sci. Comput*, v. 6, pp. 698-711.
- Chapman, B., Jost, G., & Van Der Paas, A. R., 2008. Using OpenMP - Portable Shared Memory Parallel Programming, The MIT PRESS. Massachusetts (EUA).
- Chorin, A. J., 1968. Numerical solution of the Navier-Stokes equations, *Mathematics of Computation*, v. 22, n. 104, pp. 745–762.
- Ferziger, J. H., & Peric, M., 2002. Computational Methods for Fluid Dynamics, 3 ed., Berlin: Springer.
- Ghia, U., Ghia, N., Shin, C., 1982. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, v. 48, pp. 387–411.
- Goda, K., 1979. A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows, *Journal of Computational Physics*, v. 30, n. 1, pp. 76-95.
- Guermond, J. L., & Shen, J., 2003. Velocity-correction projection methods for incompressible flows, *SIAM J.*, v. 41, n. 1, pp. 112-134.
- Guermond, J. L., Mineev, P., & Shen, J., 2006. An overview of projection methods for incompressible flows, *Computer Methods in Applied Mechanics and Engineering*, v. 195, n. 44-47, pp. 6011-6045.
- Harlow, F. H., & Welch, J. E., 1965. Numerical calculation of Time-Dependent of Viscous Incompressible Flow of Fluids with Free Surfaces, *Physics of Fluids*, v. 8, n. 12, pp. 251-263.

- Hempel, R., & Schüller, A., 1988. Experiments with Parallel Multigrid using the SUPRENUM Communications Library, GMD-Studie, n. 141.
- Herbin, R., Gerbi, S., & Sonnad, V., 1988. Parallel Implementation of a Multigrid Method on the Experimental ICAP Supercomputer, *Appl. Math. Comput.*, v. 27, pp. 281-312.
- Karniadakis, G. E., Israeli, M., & Orszag, S. A., 1991. High-order splitting methods for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, v. 97, n. 2, pp. 414-443.
- Kim, J., & Moin, P., 1985. Application of a fractional-step method to incompressible Navier-Stokes equations, *Journal of Computational Physics*, v. 59, n.2, pp. 308-323.
- Maliska, C. R., 2004. Transferência de Calor e Mecânica dos Fluidos Computacional, Rio de Janeiro: LTC, 2 ed.
- Marchi, C. R., 2001. Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos, Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, SC.
- McBryan, O. A., Frederickson, P. O., Linden, J., Shuller, A., Solchenbach, K., Stuben, K., Thole, C. A., & Trottenberg, U., 1990. Multigrid Methods on Parallel Computers – a Survey on Recent Developments.
- Oliveira, F., 2010. Efeitos de malhas anisotrópicas bidimensionais sobre o desempenho do método Multigrid geométrico, Tese (Doutorado em Engenharia Mecânica), Universidade Federal do Paraná, Curitiba, PR.
- Orszag, S. A., Israeli, M., & Deville, M., 1986. Boundary conditions for incompressible flows, *Journal of Scientific Computing*, v. 1, n.1, pp. 75-111.
- Pearson, C. E., 1964. A computational method for time dependent two dimensional incompressible viscous flow problems, Sperry-Rand Research Center, Sudbury, Mass., Report No. SRRC-RR-64-17.
- Shih, T. M., Tan, C. H., & Hwang, B. C, 1989. Effects of grid staggering on numerical schemes, *Int. J. Num. Meth. Fld.*, vol. 9, pp. 193-212.
- Sun, X. H., & Chen, Y., 2010. Reevaluating Amdahl's law in the multicore era, *Journal of Parallel and Distributed Computing*, v. 70, pp. 183-188.
- Tannehill, J. C., Anderson, D. A, Pletcher, R. H., 1997. Computational Fluid Mechanics and Heat Transfer, 2 ed. Washington: Taylor & Francis.
- Temam, R., 1969. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des fractionnaires, *Archive for Rational Mechanics and Analysis*, v. 33, n. 5, pp. 377-385.
- Thole, C. A., 1985. Experiments with Multigrid Methods on the CalTech-hypercube, GMD-Studie, n. 103.
- Timmermans, L. J., P., Mineev, P. D., & Van De Vosse, F., N., 1996. An approximate projection scheme for incompressible flow using spectral elements, *International Journal for Numerical Methods in Fluids*, v. 22, n. 7, pp. 673-688.
- Trottenberg, U., Oosterlee, C., & Schüller, A., 2001. Multigrid, Academic Press, London.
- Van Kan, J., 1986. A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput*, v. 7, n. 3, pp. 870-891.
- Villar, M. M., 2007. Análise Numérica Detalhada de Escoamentos Multifásicos Bidimensionais. Tese (Doutorado), Universidade Federal de Uberlândia.