



**Simpósio de Métodos
Numéricos em Engenharia**

25 a 27 de outubro, 2017

Extrapolações de soluções obtidas através do método multigrid aplicadas nas equações de Navier-Stokes

Rutyna, B. B., Pinto, M. A. V., Neundorff, R. L. A.,
Anunciação, M. A. M.
Universidade Federal do Paraná
Curitiba, Brasil
brunorutyna@gmail.com

Martins, M. A.
Universidade Estadual do Centro Oeste - PR
Guarapuava, Brasil

Resumo—Dentre os problemas de Dinâmica dos Fluidos Computacional (Computational Fluid Dynamics, CFD), um que merece destaque é encontrar soluções satisfatórias para as equações de Navier-Stokes, que neste trabalho representam um caso de escoamento laminar bidimensional de um fluido incompressível em regime transiente. Uma opção viável é utilizar métodos computacionais iterativos, dada a dificuldade em obter soluções analíticas. Neste trabalho optou-se por resolvê-las iterativamente através do método de projeção. Ainda assim, as aproximações obtidas durante as iterações podem convergir muito lentamente consumindo, portanto, um tempo indesejado. Logo, torna-se vantajoso utilizar o método multigrid para resolver os sistemas de equações originadas do problema discretizado, que é um poderoso acelerador de convergência e possui baixo custo computacional. É possível obter um ganho ainda maior aplicando técnicas de extrapolações de soluções, que podem ser utilizadas após a realização dos ciclos de multigrid. Este trabalho irá analisar a aplicação de extrapoladores no caso pós-multigrid, levando em conta os seguintes parâmetros: tempo de CPU, pico de memória de armazenamento, norma adimensionalizada do resíduo com base na estimativa inicial, fatores de convergência e normas de erro de iteração. Ao final, espera-se poder afirmar qual extrapolador é mais eficaz.

Palavras-chave—método multigrid; métodos de extrapolação; Navier-Stokes; aceleração de convergência; dinâmica dos fluidos computacional (CFD).

I. INTRODUÇÃO

Os problemas de engenharia exigem métodos cada vez mais rápidos e eficientes para encontrar soluções. Uma das áreas desafiadoras é a Dinâmica dos Fluidos Computacional (CFD), devido à natureza dos problemas. Por exemplo, as equações de Navier-Stokes, cujas soluções analíticas são conhecidas somente em alguns

casos. Isto ocorre pelo fato de se tratarem de equações diferenciais parciais (EDPs) não-lineares, uma área em que a matemática não está suficientemente desenvolvida [10].

Para resolver numericamente o problema, será utilizado o método de volumes finitos (MVF), que particiona um domínio bidimensional $\{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq L_x \text{ e } 0 \leq y \leq L_y\}$ em malhas estruturadas do tipo $N = N_x N_y$, em que N_x e N_y são o número de volumes nas direções coordenadas x e y , e L representa o comprimento do domínio. Estes volumes podem ser ordenados de maneira *lexicográfica*. Assim, por exemplo, uma malha com $N_x = N_y = 6$ fica conforme a Fig 1.

Uma característica da discretização de EDPs através do método de volumes finitos é que obtemos sistemas lineares do tipo

$$Au = f, \quad (1)$$

em que A é a matriz dos coeficientes, f é vetor de termos independentes e u , vetor de incógnitas. A Equação (1) pode ser resolvida através de métodos diretos, como a eliminação de Gauss, ou iterativos, como o método de Jacobi ou Gauss-Seidel. Vale destacar que esta última classe possui um custo computacional menor do que os métodos diretos para sistemas de grande porte [7].

Também há uma maneira mais aprimorada para resolver problemas do tipo (1), utilizando o conceito do *multigrid*. Este método se utiliza de um conjunto de malhas em que as iterações são realizadas em camadas diferentes, desde o nível de maior refinamento (que é a discretização original do problema) até níveis menos refinados, através de operadores que realizam a transferência de informações entre elas [6].

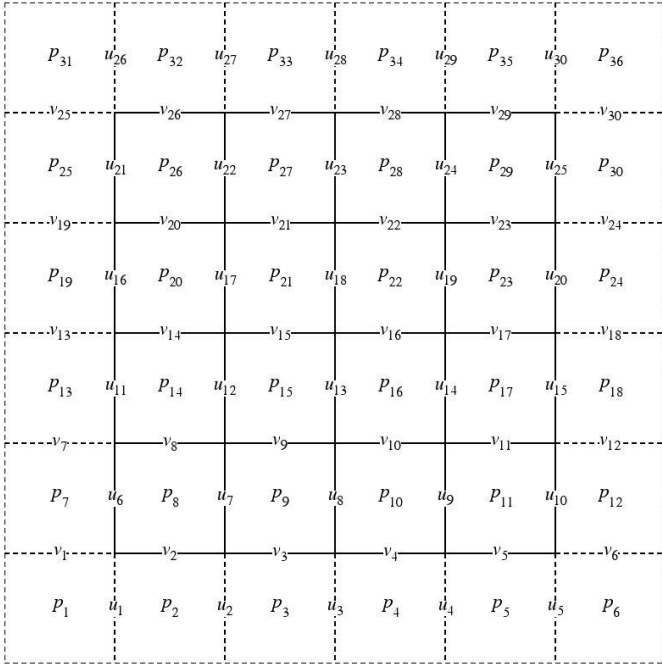


Figura 1: Ordenação lexicográfica

No intuito de melhorar a convergência do método *multigrid*, podemos ainda utilizar técnicas de extrapolação ou de aceleração de convergência associadas aos métodos iterativos [3], obtendo assim soluções numéricas com menor esforço computacional.

II. MODELO MATEMÁTICO

As equações de Navier-Stokes podem ser escritas de muitas formas, dependendo das propriedades do fluido e do escoamento [10]. Escolhemos representar de maneira análoga a de [17], visando facilitar a adaptação e entendimento dos algoritmos utilizados. Portanto, lidaremos com escoamento laminar bidimensional de um fluido incompressível ditado por

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

$$\frac{\partial u}{\partial t} + \mathbf{u} \cdot \nabla u = -\frac{\partial p}{\partial x} + \frac{1}{Re} \nabla^2 u, \quad (3)$$

$$\frac{\partial v}{\partial t} + \mathbf{u} \cdot \nabla v = -\frac{\partial p}{\partial y} + \frac{1}{Re} \nabla^2 v, \quad (4)$$

sendo x e y as coordenadas espaciais; t a coordenada temporal; u e v as coordenadas do vetor velocidade nas direções x e y ; p a pressão do fluido; Re , o número de Reynolds; $\nabla \cdot \mathbf{u}$, $\nabla \mathbf{u}$ e $\nabla^2 \mathbf{u}$ o divergente, gradiente e laplaciano de \mathbf{u} , respectivamente. Para complementar,

$$Re = \frac{\rho v_m D}{\mu}, \quad (5)$$

com ρ sendo a massa específica do fluido; μ , sua viscosidade dinâmica; v_m , sua velocidade média e D , o diâmetro do tubo.

Este trabalho resolve o problema de vórtice de Taylor-Green, cujo domínio é dado por

$$\{(x, y) \in \mathbb{R}^2 : -\pi \leq x \leq \pi \text{ e } -\pi \leq y \leq \pi\}. \quad (6)$$

As soluções analíticas são:

$$u = -(\cos x \cdot \sin y)e^{-2(t+Ti)}, \quad (7)$$

$$v = (\sin x \cdot \cos y)e^{-2(t+Ti)}, \quad (8)$$

$$p = -\frac{1}{4}(\cos 2x + \sin 2y)e^{-4(t+Ti)}, \quad (9)$$

onde Ti é o tempo inicial e $t = k\Delta t$, onde k é o passo de tempo. Neste trabalho, $Ti = 0$. As soluções analíticas geram as condições iniciais e de contorno do problema.

III. MODELOS NUMÉRICOS

A. Método de Projeção

Uma generalização dos métodos de projeção, encontradas em [13], está representada a seguir.

Primeiro passo:

$$\frac{(3\mathbf{u}^t - 4\mathbf{u}^n + \mathbf{u}^{n-1})}{2h_t} = \beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \nu \nabla^2 \mathbf{u}^t + \nabla P^n, \quad (10)$$

$$\mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n,$$

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 a pressão no passo de tempo n e \mathbf{b}^n as condições de contorno no passo de tempo n , β é constante e $\beta \mathbf{g}$ representa os termos convectivos $\mathbf{u} \cdot \nabla \mathbf{u}$. As escolhas de β_1 e β_0 implicam se os termos convectivos serão tratados explicitamente ou implicitamente. Neste trabalho, $\beta_1 = -3/2$ e $\beta_0 = 1/2$ [2].

Segundo passo:

$$\nabla^2 \phi^{n+1} = \frac{3}{2h_t} \nabla \cdot \mathbf{u}^t \quad (11)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{2h_t}{3} \nabla \phi^{n+1} \quad (12)$$

$$P^{n+1} = P^n + \phi^{n+1} - \nu \nabla \cdot \mathbf{u}^t \quad (13)$$

onde ϕ é o termo de correção da pressão e $\nu = 1/Re$. Esta versão do método de projeção tem condições de contorno para a pressão dadas por

$$\partial_n P^{n+1}|_{\partial\Omega} = -\nu(\nabla \times \nabla \times \mathbf{u}^{n+1}) \cdot \mathbf{n}|_{\partial\Omega}, \quad (14)$$

que são condições de contorno consistentes para a pressão. O algoritmo dado pelos passos um e dois é conhecido [14, 15] como correção incremental na pressão na forma rotacional. A Referência [15] mostra que a versão incremental na forma rotacional é de segunda ordem para as velocidades nas normas L_2 , L_1 e L_∞ . Para a pressão, o esquema é de segunda ordem nas normas L_2 e L_1 , e de ordem 3/2 na norma L_∞ .

Para uma generalização dos métodos de projeção ver [13].

B. Discretização

Este trabalho apresenta malhas desencontradas, *staggered grid*, do tipo [16], também encontrado em [18]. As velocidades encontram-se nas faces e a pressão nos centros dos volumes, assim evitando instabilidades da pressão [25].

Novamente em Fig. 1, os volumes com linhas tracejadas representam os volumes fictícios e não pertencem ao domínio físico do problema. As condições de contorno para a variável u são prescritas nos contornos oeste e leste, utilizando extrapolação linear nos contornos norte e sul. De maneira análoga encontra-se a variável v nos contornos norte e sul, com extrapolação linear nas outras duas direções.

A equação da conservação da massa, Eq. 2, pode ser reescrita como:

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS = 0, \quad (15)$$

onde n representa o passo de tempo.

A Equação (3), da conservação da quantidade de movimento linear na direção x (QML- x), fica:

$$\frac{\partial}{\partial t} \int_V u dV = - \oint_S u \mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_x dS + \frac{1}{Re} \oint_S \nabla u \cdot \mathbf{n} dS, \quad (16)$$

e a equação da conservação da quantidade de movimento linear na direção y (QML- y):

$$\frac{\partial}{\partial t} \int_V v dV = - \oint_S v \mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_y dS + \frac{1}{Re} \oint_S \nabla v \cdot \mathbf{n} dS. \quad (17)$$

onde n_x e n_y são vetores normais às direções x e y , respectivamente.

As Equações (15) a (17) podem ser discretizadas com uso de malhas "desencontradas" calculando as integrais sobre os volumes de controle mostrados nas Figs. 2 a 4. Para simplificar, assume-se $h_x = h_y = h$ (malhas quadrangulares). A discretização temporal depende do método de projeção utilizado e pode ser visto detalhadamente em [13].

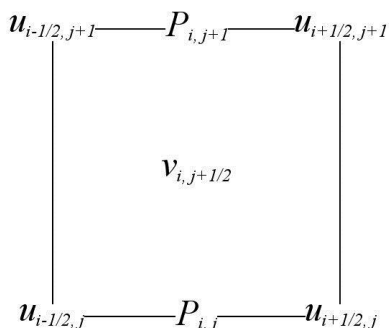


Figura 2: Volume centrado em v

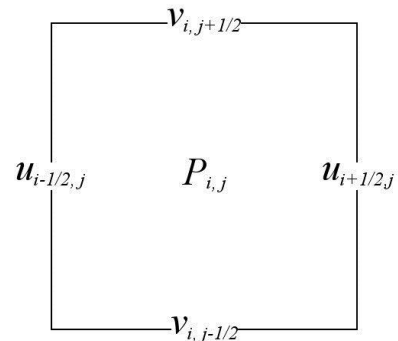


Figura 3: Volume centrado em P

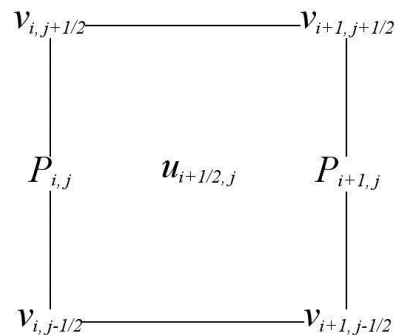


Figura 4: Volume centrado em u

A Eq. 15 é discretizada utilizando um volume centrado em P , conforme a Fig. 3:

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS \approx u_{i+1/2, j}^{n+1} - u_{i-1/2, j}^{n+1} + v_{i, j+1/2}^{n+1} - v_{i, j-1/2}^{n+1} = 0. \quad (18)$$

Os termos das pressões são discretizados utilizando os volumes mostrados nas Figs. 2 e 4 da seguinte maneira:

$$\oint_S p n_x dS \approx (p_{i+1, j} - p_{i, j})h, \quad (19)$$

$$\oint_S p n_y dS \approx (p_{i, j+1} - p_{i, j})h. \quad (20)$$

Com os volumes de controle mostrados nas Figs. 2 e 4 pode-se discretizar os termos difusivos e advectivos. Estes passos são vistos em [22].

Exemplos de aproximações para as componentes das velocidades podem ser escritas como

$$(u^2)_{i, j}^n = \left(\frac{1}{2} (u_{i+1/2, j}^n + u_{i-1/2, j}^n) \right)^2, \quad (21)$$

$$(v^2)_{i, j}^n = \left(\frac{1}{2} (v_{i, j+1/2}^n + v_{i, j-1/2}^n) \right)^2. \quad (22)$$

Para garantir a estabilidade das soluções, o passo de tempo h_t

deve respeitar o critério [18]:

$$h_t < \frac{1}{2Re} \frac{h_x^2 h_y^2}{h_x^2 + h_y^2}. \quad (23)$$

Em geral, os métodos de projeção fazem com que as condições de contorno sobre a pressão sejam do tipo de Neumann. Para garantir existência e unicidade da solução, deve-se ter [6]:

$$\int_{\Omega} \nabla^2 \phi^{n+1} = \int_{\Omega} \frac{3}{2h_t} \nabla \cdot \mathbf{u}^t = 0. \quad (24)$$

Mais detalhes sobre métodos de projeção podem ser encontrados em [22] e [13].

C. Multigrid Method

Os *solvers* básicos, como os métodos de Jacobi ou Gauss-Seidel, possuem a propriedade de reduzir rapidamente os modos de alta frequência do erro, restando apenas os modos suaves. Além disso, os modos suaves passam a ser mais oscilatórios em malhas mais grossas. Logo, é possível utilizar uma estratégia que tira proveito destas duas características, que neste trabalho é feito com a aplicação do método *multigrid* [6].

O método *multigrid* se aplica a sistemas de equações do tipo (1), com um domínio discretizado Ω_h . Para iniciá-lo, basta aplicar um *solver* iterativo, como Jacobi-ponderado ou Gauss-Seidel, porém apenas um número reduzido de iterações. Feito isso, calculamos o resíduo, dado por:

$$r^h = f^h - A^h \mathbf{v}^h \quad (25)$$

em que \mathbf{v}^h é a aproximação de \mathbf{u} obtida na malha Ω^h e h é a distância entre dois pontos subsequentes do domínio [6].

Uma vez realizado o procedimento acima, transferimos o resíduo para a malha subsequentemente mais grossa utilizando um operador de *restrição*. A razão de *engrossamento* da malha será igual a 2. Assim, temos o sistema:

$$A^{2h} e^{2h} = r^{2h}, \text{ em } \Omega^{2h}, \quad (26)$$

sendo e o erro a ser estimado.

Após obtermos uma aproximação do erro na malha Ω^{2h} , iniciamos o processo de correção da aproximação \mathbf{v} . Primeiro utilizamos um operador que transfere o erro para a malha fina (operador de *prolongação*), então realizamos a seguinte operação:

$$\mathbf{v}^h \leftarrow \mathbf{v}^h + e^{2h}. \quad (27)$$

Por fim, aplicamos novamente o *solver* escolhido na aproximação \mathbf{v}^h obtida. Mais detalhes sobre os operadores de *restrição* e *prolongação* são encontrados em [28], [27] e [26].

Este procedimento pode ser realizado utilizando outras malhas mais grossas. A cada vez que retornamos à malha mais fina com a devida correção, temos o que chamamos de um ciclo-V, conforme ilustrado na Fig 5.

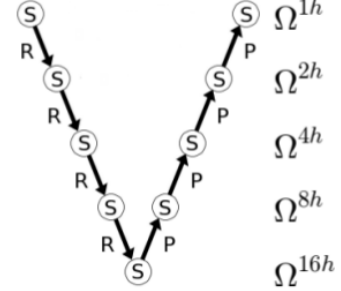


Figura 5: Diagrama de um ciclo V aplicado em 5 malhas.

S = suavização, R = restrição, P = prolongação [21]

O número de vezes a realizar ciclos-V deve ser estabelecido por uma quantidade fixada pelo usuário ou através de critérios de parada que envolvem a convergência da solução.

D. Métodos de Extrapolação

A fim de acelerar a convergência de métodos iterativos, pode-se utilizar técnicas de extrapolações, que se baseiam em transformações de sequências. Isto é, se uma sequência converge lentamente, é possível transformá-la em outra, desde que se mantenha o limite da original, e esta nova sequência pode convergir mais rapidamente que a primeira.

Um exemplo de transformação de uma sequência (S_n) de números reais ou complexos que converge para s , de acordo com [5], é:

$$T_n = \frac{S_n + S_{n+1}}{2}, n = 0, 1, 2, \dots, \quad (28)$$

ou

$$T_n = \frac{S_n S_{n+2} + S_{n+1}^2}{S_{n+2} - 2S_{n+1} + S_n}, n = 0, 1, 2, \dots, \quad (29)$$

onde T_n é a nova sequência gerada, que também pode ser chamada de extrapolador.

Este trabalho apresenta sete extrapoladores, sendo cinco do tipo escalar e dois vetoriais. Considerando que $C_1, C_2, \dots, C_k, \dots, C_\infty$ são vetores obtidos através de um processo iterativo e $\phi_1 = C_1(k)$, $\phi_2 = C_2(k)$, $\phi_i = C_i(k)$ são os escalares para todos k -ésimos componentes dos vetores C_i , então definem-se os seguintes extrapoladores:

$$\phi_\infty^{Aitken} = \frac{\phi_1 \phi_3 - \phi_2^2}{\phi_3 - 2\phi_2 + \phi_1}, \quad (30)$$

extrapolador de Aitken [7].

Baseado no Estimador Empírico proposto em [19],

$$\phi_\infty^{Empirico} = \phi_3 + \frac{(\phi_3 - \phi_2)^2}{2\phi_2 - \phi_3 - \phi_1}, \quad (31)$$

e

$$\phi_{\infty}^{Mitin} = \frac{\phi_1 \phi_5 - \phi_3^2}{\phi_5 - 2\phi_3 + \phi_1}, \quad (32)$$

que é a fórmula utilizada na extrapolação de Mitin [20].

Os outros dois extrapoladores utilizados neste trabalho, Épsilon e Rho, possuem uma semelhança formal entre si, porém diferem quanto às capacidades de convergência. O algoritmo Épsilon é um poderoso acelerador de métodos iterativos que resolvam equações diferenciais discretizadas, porém para sequências que converjam logaritmicamente, ele falha [8, 9]. Opostamente, o algoritmo Rho não acelera sequências com convergência linear, mas é muito eficiente para sequências logaritmicamente convergentes [11, 12].

Para uma sequência (S_n) , o algoritmo Épsilon Escalar é dado por [3]:

$$\epsilon_{-1}^{(n)} = 0, \quad \epsilon_0^{(n)} = S_n, \quad n = 0, 1, \dots, \quad (33)$$

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}}, \quad n, k = 0, 1, \dots, \quad (34)$$

para denominadores diferentes de zero. O sobrescrito de $\tilde{\epsilon}$ representa o elemento e, o subscrito, a iteração. Muito semelhante, o algoritmo de Rho Escalar é dado por [29]:

$$\rho_{-1}^{(n)} = 0, \quad \rho_0^{(n)} = S_n, \quad n = 0, 1, \dots, \quad (35)$$

$$\rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{k+1}{\rho_k^{(n+1)} - \rho_k^{(n)}}, \quad n, k = 0, 1, \dots \quad (36)$$

Ambos os algoritmos, Épsilon e Rho, possuem versões escalar e topológica. Enquanto a primeira preserva as características de cada componente do vetor individualmente, a segunda trata de levar em consideração as propriedades do vetor como um todo, e as duas variações são analisadas neste trabalho. O algoritmo Épsilon Topológico pode ser visto abaixo:

$$\tilde{\epsilon}_{-1}^{(n)} = 0, \quad n = 0, 1, \dots, \quad (37)$$

$$\tilde{\epsilon}_0^{(n)} = S_n, \quad n = 0, 1, \dots, \quad (38)$$

$$\tilde{\epsilon}_{2k+1}^{(n)} = \tilde{\epsilon}_{2k-1}^{(n+1)} + \frac{y}{\langle y, \tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)} \rangle}, \quad k, n = 0, 1, \dots, \quad (39)$$

$$\tilde{\epsilon}_{2k+2}^{(n)} = \tilde{\epsilon}_{2k}^{(n+1)} + \frac{\tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)}}{\langle \tilde{\epsilon}_{2k+1}^{(n+1)} - \tilde{\epsilon}_{2k+1}^{(n)}, \tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)} \rangle},$$

$$k, n = 0, 1, \dots,$$

onde, se S_n são elementos em E , então y é um vetor arbitrário tal que $y \in E^*$ (o espaço dual algébrico de E) [4] e o operador $\langle \cdot, \cdot \rangle$ representa um produto interno. E o algoritmo de Rho Topológico é dado por [24]:

$$\tilde{\rho}_{-1}^{(n)} = 0, \quad n = 0, 1, \dots, \quad (40)$$

$$\tilde{\rho}_0^{(n)} = S_n, \quad n = 0, 1, \dots, \quad (41)$$

$$\tilde{\rho}_{2k+1}^{(n)} = \tilde{\rho}_{2k-1}^{(n+1)} + \frac{(2k+1)y}{\langle y, \tilde{\rho}_{2k}^{(n+1)} - \tilde{\rho}_{2k}^{(n)} \rangle}, \quad k, n = 0, 1, \dots, \quad (42)$$

$$\tilde{\rho}_{2k+2}^{(n)} = \tilde{\rho}_{2k}^{(n+1)} + \frac{(2k+2)\tilde{\rho}_{2k}^{(n+2)} - \tilde{\rho}_{2k}^{(n+1)}}{\langle \tilde{\rho}_{2k}^{(n+2)} - \tilde{\rho}_{2k}^{(n+1)}, \tilde{\rho}_{2k+1}^{(n+1)} - \tilde{\rho}_{2k+1}^{(n)} \rangle},$$

$$k, n = 0, 1, \dots$$

Detalhes do extrapolador Épsilon são encontrados em [29] e [3], e do extrapolador Rho em [5]. Aplicações de extrapoladores são vistas em [1].

IV. RESULTADOS

Os métodos de extrapolação deste trabalho foram aplicados nas variáveis de velocidade u e v para cada instante de tempo t . O extrapolador foi executado após os ciclos do método *multigrid* e foram usadas malhas com $N = 128 \times 128$, $N = 512 \times 512$ e $N = 1024 \times 1024$, e critérios de parada ϵ iguais a 10^{-6} , 10^{-8} e 10^{-10} .

A. Dados de Implementação

O método *multigrid* deste trabalho realiza ciclos-V, com razão de engrossamento das malhas igual a 2 (engrossamento padrão), até que se atinja a malha mais grossa possível [6]. O solver associado é o Gauss-Seidel *red-black*, com números de pré e pós-suavização iguais a 3 [23].

A cada passo de tempo são resolvidas numericamente as velocidades e a pressão. O critério de parada utilizado no método iterativo é a norma euclidiana do resíduo adimensionalizada pela norma euclidiana do resíduo na estimativa inicial [6, 26]:

$$\|R\|_2 = \frac{\|R^{it}\|_2}{\|R^0\|_2}, \quad (43)$$

onde R^{it} representa o resíduo da iteração atual. Quando a Eq. (43) atinge valores menores que uma tolerância ϵ , o processo é encerrado.

Os algoritmos foram implementados na linguagem Fortran 2003 e utilizam precisão dupla. Os testes foram realizados em um computador com processador Intel de 2.50 GHz x 4, 4GB de memória RAM e sistema operacional Windows 64-bits.

B. Uso de extrapoladores pós-multigrid

Nesta etapa, o método *multigrid* foi executado até que atingisse uma tolerância ϵ estabelecida. Em seguida, foi tomado um desses dois caminhos:

- foi realizado um novo ciclo de multigrid, denominado "MG + 1 ITE";
- foi aplicado um método de extrapolação, denominado MG + Extrapolador".

A Tabela I mostra quantas aproximações são utilizadas para cada extrapolador.

Tabela I: Número de soluções utilizadas para extrapolações

Extrapolador	Número de soluções
MG + Aitken	3
MG + Empírico	3
MG + Mitin	5
MG + Épsilon escalar	5
MG + Rho escalar	5
MG + Épsilon topológico	5
MG + Rho topológico	5

As medições dos resultados foram feitas com os seguintes parâmetros: tempo de processamento ou tempo de CPU (t_{cpu}), em segundos [s]; o pico de memória (M) atingido durante a execução do algoritmo, em Megabytes [MB]; a norma do resíduo adimensionalizada pela norma do resíduo na estimativa inicial ($\|R\|_2$), dada pela Eq. (43); o fator de convergência média empírico ($\hat{q}^{(k)}$), definido por [26]:

$$\hat{q}^{(k)} = \sqrt[k]{\frac{\|R^{it}\|_2}{\|R^0\|_2}}, \quad (44)$$

onde k representa a k -ésima iteração. Valores de $\hat{q}^{(k)}$ considerados bons são próximos de zero [6, 7]. Por fim, também são medidas as normas infinito ($\|E_n\|_\infty$) e euclidiana ($\|E_n\|_2$) das variáveis u , v e p , com E_n (também chamado erro de iteração) sendo a diferença obtida entre a aproximação no n -ésimo passo de tempo e a solução analítica.

A Tabela II contém os resultados obtidos em uma malha com $N = 512 \times 512$, tolerância $\varepsilon = 10^{-8}$ e passos de tempo $t = 10$.

As medições $\hat{q}^{(k)}$ e $\|R\|_2$ para a variável v foram feitas a cada instante de tempo, e podem ser vistas nas Figs. 6 e 7.

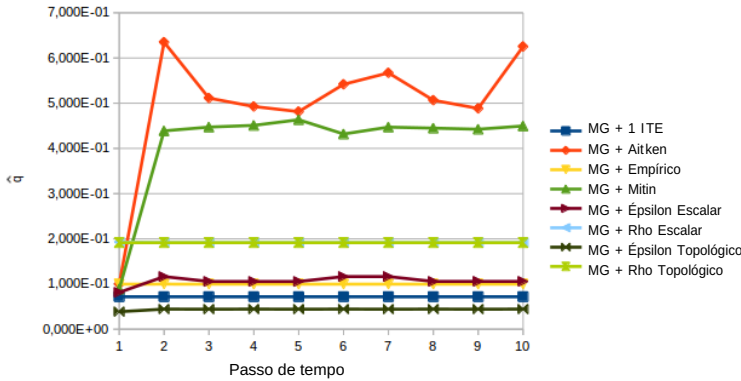


Figura 6: Fator de convergência média empírico para cada instante de tempo

Vale ressaltar que os parâmetros $\hat{q}^{(k)}$ e $\|R\|_2$, quando medidos para u , apresentaram resultados praticamente iguais aos das Figs. 6 e 7.

Com os testes realizados, nota-se que o extrapolador Épsilon Topológico é melhor que os demais nos parâmetros normas infinito

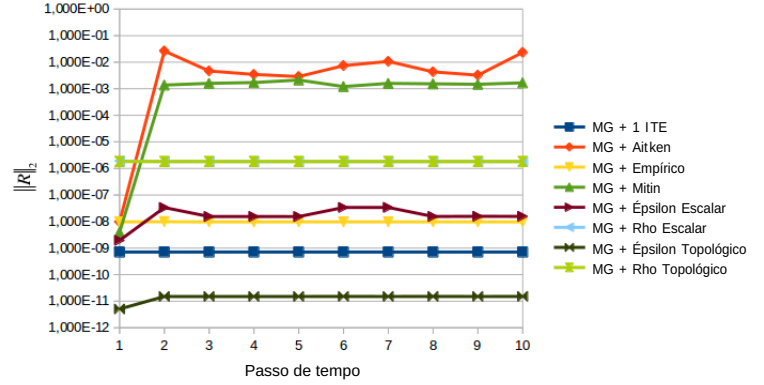


Figura 7: $\|R\|_2$ para cada instante de tempo

e euclidiana do erro de iteração das variáveis u e v , fator de convergência média empírico e norma do resíduo. A memória utilizada é superior que os demais métodos, e isto se justifica pelo número de vetores auxiliares utilizados nas operações da extrapolação. As normas da variável p atingem valores próximos dos melhores resultados destas colunas. Tudo isto leva a concluir que o Épsilon Topológico possui potencial para acelerar a convergência do problema apresentado de maneira significativa.

V. CONCLUSÃO

Com os resultados obtidos neste trabalho, observa-se que dentre os sete métodos de extrapolação analisados, o Épsilon Topológico é o que se destaca para resolver o problema proposto de escoamento laminar bidimensional de um fluido incompressível em regime transiente, regido pelas equações de Navier-Stokes. Nota-se também que ele é o único a ter grandes vantagens em relação ao uso do multigrid puro, até mesmo o superando em diversos quesitos.

Para tratar o alto uso de memória do Épsilon Topológico, vale ressaltar que uma otimização no algoritmo pode contribuir para melhorar a performance, já que o esforço durante a implementação do extrapolador foi dado com outro viés.

Os próximos trabalhos devem ser no sentido de explorar variantes da aplicação do Épsilon Topológico, otimizando o uso de memória e abranger novas aplicações para o extrapolador.

AGRADECIMENTOS

O primeiro autor agradece o apoio financeiro da Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). Os autores também agradecem a infraestrutura oferecida pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da UFPR (Universidade Federal do Paraná).

Tabela II: Extrapolações ao final dos ciclos *multigrid*

Metodologia	t_{cpu} [s]	M (MB)	$\ E_n\ _\infty (u)$	$\ E_n\ _2 (u)$	$\ E_n\ _\infty (v)$	$\ E_n\ _2 (v)$	$\ E_n\ _\infty (p)$	$\ E_n\ _2 (p)$
MG + 1 ITE	60,2	101,464	5,982E-10	8,229E-08	5,967E-10	8,228E-08	4,493E-07	1,054E-04
MG + Aitken	52,5	116,204	7,225E-06	1,626E-05	7,434E-06	1,506E-05	8,552E-03	1,106E-02
MG + Empírico	55,8	116,200	9,733E-09	2,083E-06	9,732E-09	2,083E-06	6,924E-07	1,122E-04
MG + Mitin	51,1	116,232	2,592E-07	1,363E-06	2,642E-07	1,354E-06	5,617E-04	9,646E-04
MG + Épsilon escalar	60,2	116,240	1,529E-09	7,492E-08	1,369E-09	7,490E-08	4,459E-07	1,055E-04
MG + Rho escalar	58,3	116,220	1,746E-06	3,747E-04	1,746E-06	3,747E-04	4,086E-05	5,086E-03
MG + Épsilon topológico	59,5	134,684	5,961E-10	7,465E-08	5,961E-10	7,465E-08	4,460E-07	1,055E-04
MG + Rho topológico	58,6	134,676	1,746E-06	3,748E-04	1,746E-06	3,748E-04	4,069E-05	5,066E-03

REFERÊNCIAS

- [1] Anuniação, M. A. M. d. [2013]. Redução do erro de iteração e aceleração do método multigrid com uso de extrapoladores.
- [2] Bell, J. B., Colella, P. and Glaz, H. M. [1989]. A second-order projection method for the incompressible Navier-Stokes equations, *Journal of Computational Physics* **85**(2): 257–283.
- [3] Brezinski, C. [1974]. *Forme confluente de l' ϵ -algorithme topologique*, Vol. 23, Springer.
- [4] Brezinski, C. and Redivo-Zaglia, M. [2014]. The simplified topological ϵ -algorithms for accelerating sequences in a vector space, *SIAM Journal on Scientific Computing* **36**(5): A2227–A2247.
- [5] Brezinski, C. and Zaglia, M. R. [2013]. *Extrapolation methods: theory and practice*, Vol. 2, Elsevier.
- [6] Briggs, W. L., Henson, V. E. and McCormick, S. F. [2000]. *A multigrid tutorial*, SIAM.
- [7] Burden, R. and Faires, J. [2008]. *Análise numérica*.
- [8] Delahaye, J. and Germain-Bonne, B. [1982]. The set of logarithmically convergent sequences cannot be accelerated, *SIAM Journal on Numerical Analysis* **19**(4): 840–844.
- [9] Delahaye, J.-P. [2012]. *Sequence transformations*, Vol. 11, Springer Science & Business Media.
- [10] Fortuna, A. d. O. [2000]. *Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações*, Edusp.
- [11] Gao, Q., Jiang, Z., Liao, T. and Song, K. [2010]. Application of the vector ϵ and ρ extrapolation methods in the acceleration of the Richardson–Lucy algorithm, *Optics Communications* **283**(21): 4224–4229.
- [12] Graves-Morris, P., Roberts, D. and Salam, A. [2000]. The epsilon algorithm and related topics, *Journal of Computational and Applied Mathematics* **122**(1): 51–80.
- [13] Guermond, J.-L., Mineev, P. and Shen, J. [2006]. An overview of projection methods for incompressible flows, *Computer methods in applied mechanics and engineering* **195**(44): 6011–6045.
- [14] Guermond, J.-L. and Shen, J. [2001]. Quelques résultats nouveaux sur les méthodes de projection, *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* **333**(12): 1111–1116.
- [15] Guermond, J. and Shen, J. [2004]. On the error estimates for the rotational pressure-correction projection methods, *Mathematics of Computation* **73**(248): 1719–1737.
- [16] Harlow, F. H. and Welch, J. E. [1965]. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *The physics of fluids* **8**(12): 2182–2189.
- [17] Kim, J. and Moin, P. [1985]. *Application of a fractional-step method to incompressible Navier-Stokes equations*, Vol. 59, Elsevier.
- [18] Maliska, C. R. [1995]. *Transferência de calor e mecânica dos fluidos computacional*.
- [19] Martins, M. A. and Marchi, C. H. [2008]. Estimate of iteration errors in computational fluid dynamics, *Numerical Heat Transfer, Part B: Fundamentals* **53**(3): 234–245.
- [20] Mitin, A. [1985]. Linear extrapolation in an iterative method for solving systems of equations, *USSR Computational Mathematics and Mathematical Physics* **25**(2): 1–6.
- [21] Neundorf, R. L. A. [2016]. Aplicação do método multigrid paralelizado em GPUs no modelo WRF.
- [22] Neundorf, R. L. A., Pinto, M. A. V., Araki, L. K. and Calvetti, L. [2016]. Desempenho de um algoritmo multigrid paralelo aplicado às equações de Navier-Stokes.
- [23] Oliveira, F. d. [2010]. *Efeito de malhas anisotrópicas bidimensionais sobre o desempenho do método multigrid geométrico*, PhD thesis.
- [24] Osada, N. [1991]. Acceleration methods for vector sequences, *Journal of Computational and Applied Mathematics* **38**(1-3): 361–371.
- [25] Shih, T., Tan, C. and Hwang, B. [1989]. Effects of grid staggering on numerical schemes, *International Journal for numerical methods in fluids* **9**(2): 193–212.
- [26] Trottenberg, U., Oosterlee, C. W. and Schuller, A. [2000]. *Multigrid*, Academic press.
- [27] Villar, M. M. [2007]. *Análise numérica detalhada de escoamentos multifásicos bidimensionais*, PhD thesis.
- [28] Wesseling, P. [1995]. *Introduction To Multigrid Methods*.
- [29] Wynn, P. [1956]. On a procrustean technique for the numerical transformation of slowly convergent sequences and series, **52**(4): 663–671.