Efeito de parâmetros do método Multigrid associado com extrapoladores em problemas de CFD

Anunciação, M. A. M., Pinto, M. A. V., Araki, L. K., Gonçalves, S. F. T.
Universidade Federal do Paraná
Curitiba, Brasil
marcioalexandro@ufpr.br

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

***Resumo* — Neste trabalho foram resolvidos, numericamente, o problema de condução de calor linear bidimensional, governado pela equação de Poisson, com condições de contorno de Dirichlet, empregando o método *Multigrid* geométrico associado aos seguintes métodos de extrapolação: Aitken, Empírico, Mitin, Épsilon (escalar e topológico), Rho (escalar e topológico) e múltiplas extrapolações de Aitken e Mitin; e o problema problema da cavidade quadrada com tampa móvel, governado pela equação de Burgers, com condições de contorno de Dirichlet, empregando o método *Multigrid* geométrico associado ao método de extrapolação Épsilon Topológico durante os ciclos do *Multigrid*. O objetivo deste trabalho foi analisar o comportamento do erro de iteração, tempo de CPU e fatores de convergência. Para ambos os problemas, verificou-se a redução da magnitude do erro de iteração, redução do resíduo adimensionalizado com base na estimativa inicial e redução do fator de convergência, em um tempo praticamente equivalente ao da aplicação do método *Multigrid* puro.**

Palavras-chave* — *métodos de extrapolação; Multigrid; aceleração de convergência; erro de iteração

I. INTRODUÇÃO

No processo de discretização de um modelo matemático em problemas de Dinâmica dos Fluidos Computacional (CFD) é comum a obtenção de sistemas de equações algébricas do tipo Au=f, onde A é a matriz dos coeficientes, f é o vetor independente e u é o vetor incógnita. A solução desses sistemas, em geral, é obtida com o emprego de métodos iterativos, que embora mais rápidos que os métodos diretos, geralmente apresentam convergência lenta para problemas de grande porte. A aceleração de convergência de processos iterativos pode se dar de duas formas: modificando o processo iterativo, ou transformando a sequência que converge lentamente em outra, com melhores propriedades de convergência [1].

Nas últimas décadas, uma alternativa que se tem mostrado muito eficiente na aceleração de processos iterativos é o método *Multigrid* [2,3]. Sua filosofia está baseada no emprego de várias malhas com diferentes graus de refinamento, as quais são percorridas durante o processo iterativo. Outra abordagem que visa acelerar a convergência dos métodos iterativos é associá-los a métodos de extrapolação, cuja finalidade é transformar uma sequência de vetores em uma nova sequência que converge mais rapidamente do que a inicial.

No presente trabalho foram associados métodos de extrapolação ao método *Multigrid*, com o objetivo de acelerar o processo iterativo e reduzir o erro de iteração. Tal formulação mostrou-se atraente, em especial por não se ter na literatura citações sobre tal procedimento.

II. MÉTODOS NUMÉRICOS

O método *Multigrid* se vale das características de suavização do erro por parte dos métodos iterativos clássicos: ela ocorre rapidamente (nas iterações iniciais) para componentes oscilatórias, enquanto que para componentes suaves, para um número grande de iterações tais métodos perdem sua eficiência. Assim, o método *Multigrid* trabalha com um esquema de malhas auxiliares mais grosseiras (com menor número de pontos) nas quais as componentes do erro são rapidamente suavizadas, para então retornar-se à malha original. A informação é transferida entre malhas através de operadores, chamados de operadores de restrição (informações de uma malha fina para a grossa seguinte), representados genericamente por $[I]_h^H$ e definidos por

$$v^H = [I]_h^H \, v^h , \tag{1}$$

ou de prolongação (informações da malha grossa para a fina), representados genericamente por e definidos por

$$v^h = [I]_H^h \, v^H . \tag{2}$$

O operador de restrição utilizado neste trabalho foi o operador de ponderação completa [2,3,4], e o operador de prolongação foi a interpolação bilinear [2,3,4], ambos bastante referenciados na literatura.

Neste trabalho utilizou-se ainda o ciclo V, por ser computacionalmente eficiente [2,3,4]. Para a equação de Poisson, que é um problema linear, utilizou-se o Esquema de Correção (CS – *Correction Scheme*) que transfere apenas o

resíduo para as malhas mais grossas [2,3,4], enquanto para as equações de Burgers, que é um sistema de equações não-lineares, utilizou-se o esquema de aproximação completa (FAS – *Full Approximation Scheme*), onde são transferidos o resíduo e a aproximação da solução para as malhas grossas [3].

A razão de engrossamento, para o caso bidimensional, considerando malhas uniformes, é definida como $r = H/h$, onde h representa o tamanho do espaçamento da malha fina Ω^h (também chamado de dimensão dos elementos da malha fina), H o tamanho do elemento da malha imediatamente mais grossa Ω^H . Neste trabalho, utilizou-se a razão de engrossamento padrão, ou seja, $r = 2$ [2].

Os algoritmos para os esquemas CS e FAS, com ciclo V, para vários níveis de malhas, até se atingir um critério de parada ou alcançar o número máximo de ciclos escolhidos, podem ser encontrados em [2,3,4].

Os métodos de extrapolação têm por finalidade transformar uma sequência que converge lentamente em outra, com melhores propriedades de convergência. Tais extrapoladores podem ser classificados em escalares e vetoriais, de acordo com a forma com que suas informações são manipuladas.

No trabalho de Mitin [5], considera-se um processo iterativo em um espaço de Hilbert, onde os vetores $C_1, C_2, ..., C_k, ..., C_\infty$ são obtidos em passos desse processo. Define-se o vetor $\Delta_k = C_k - C_\infty$. Em geral, tem-se: $\Delta_{k+1} = F(\Delta_k)$, onde F é o operador que define o processo iterativo. Neste mesmo trabalho, chegou-se a uma expressão que servirá de suporte para que sejam definidas as fórmulas dos três primeiros extrapoladores utilizados neste trabalho:

$$C_{\infty,j} = \frac{C_{k+2,j}C_{k,j} - C_{k+1,j}^{\,2}}{C_{k+2,j} - 2C_{k+1,j} + C_{k,j}}, \tag{3}$$

Considerando-se as iterações subsequentes, com $k = 1$, $C_1 = \phi_1$, $C_2 = \phi_2$, $C_3 = \phi_3$ e para todas as j-ésimas componentes satisfazendo a relação descrita pela Eq. (2.47):

$$\phi_\infty^{Aitken} = \frac{\phi_1\phi_3 - \phi_2^{\,2}}{\phi_3 - 2\phi_2 + \phi_1}, \tag{4}$$

que é a fórmula para a extrapolação de Aitken [1,6].

Com base na teoria do estimador Empírico [7], tem-se:

$$\phi_\infty^{Empírico} = \phi_3 + \frac{(\phi_3 - \phi_2)^2}{2\phi_2 - \phi_3 - \phi_1}, \tag{5}$$

que é uma relação com as mesmas propriedades do extrapolador de Aitken.

Considerando-se agora as iterações subsequentes com $k = 1$, $C_1 = \phi_1$, $C_2 = \phi_2$, $C_3 = \phi_3$, $C_4 = \phi_4$, $C_5 = \phi_5$ e para todas as j-ésimas componentes satisfazendo a relação descrita para a Eq. (2.46), pode-se escrever:

$$\phi_\infty^{Mitin} = \frac{\phi_1\phi_5 - \phi_3^{\,2}}{\phi_5 - 2\phi_3 + \phi_1}, \tag{6}$$

que é a fórmula para a extrapolação de Mitin [5].

Outros dois métodos de extrapolação existentes na literatura (conforme serão descritos a seguir) são o algoritmo Épsilon e o algoritmo Rho que apresentam uma semelhança formal, mas diferem significativamente em sua capacidade de acelerar a convergência. As propriedades dos algoritmos Épsilon e Rho são, de certo modo, complementares entre si.

A importância de estudar o algoritmo Épsilon reside em parte no seu potencial para aplicação na aceleração da convergência da solução iterativa de equações diferenciais discretizadas. Entretanto, todos os algoritmos têm seus domínios de validade. O algoritmo Épsilon falha, por exemplo, em sequências logaritmicamente convergentes (que convergem muito lentamente) e não consegue atingir o ponto fixo de geradores de sequências que divergem muito rapidamente. O algoritmo Épsilon geralmente falha para tais sequências e [8,9] demonstraram que não há um acelerador universal para sequências logaritmicamente convergentes [10]. O algoritmo Rho não acelera sequências com convergência linear, mas é muito poderoso para sequências logaritmicamente convergentes [10,11].

Os métodos de extrapolação Rho e Épsilon podem ser generalizados para o caso vetorial, ou seja, envolvendo operações vetoriais. Além disso, tal abordagem pode ser aplicada de forma recursiva o que é conhecido como formulação topológica [1], sendo que tal formulação também foi utilizada no presente trabalho.

Com o objetivo de utilizar um extrapolador escalar para uma sequência de vetores, este deve ser aplicado simultaneamente a cada componente do vetor. No entanto uma desvantagem da utilização de tal técnica é o fato de a ligação entre os componentes serem negligenciadas, em outras palavras, os componentes são tratados como escalares independentes.

Nesses casos a abordagem vetorial é recomendada e, entretanto, não acarreta em um nível maior de complexidade. Como exemplo, para generalizar o extrapolador Épsilon escalar no caso vetorial, é necessário definir o inverso de um vetor. Uma possibilidade considerada por Wynn [12] é o uso do inverso definido por

$$z^{-1} = \frac{z}{\|z\|^2}, \; z \in \mathrm{R}^N. \tag{7}$$

Portanto, para sequências vetoriais, o extrapolador Épsilon vetorial é definido por:

$$\begin{aligned} &\varepsilon_{-1}^{(n)} = 0; \; \varepsilon_0^{(n)} = S_n; \; \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \\ &+\left[\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}\right]^{-1}, \; k, n = 0,1,... \end{aligned} \tag{8}$$

Na referência [1] é proposta outra generalização do Épsilon escalar para sequências vetoriais, que se trata de uma forma recursiva do algoritmo Épsilon vetorial, no qual não é necessário o cálculo do inverso de vetores, o chamado Épsilon topológico.

$$\begin{cases} \varepsilon_{-1}^{(n)} = 0; \quad \varepsilon_{0}^{(n)} = S_n; \quad \varepsilon_{2k+1}^{(n)} = \varepsilon_{2k-1}^{(n+1)} + \dfrac{y}{\left\langle y, \Delta\varepsilon_{2k}^{(n)} \right\rangle}; \\ \varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \dfrac{\Delta\varepsilon_{2k}^{(n)}}{\left\langle \Delta\varepsilon_{2k+1}^{(n)}, \Delta\varepsilon_{2k}^{(n)} \right\rangle}, \quad k, n = 0,1,... \end{cases} \tag{9}$$

sendo y um vetor arbitrário, ortogonal à j-ésima componente do resíduo generalizado entre $e_{k,j}\left(S_n\right)$ e $e_{k,j-1}\left(S_n\right)$ e

$$\left\langle \alpha, \beta \right\rangle = \sum_{i=1}^{p} \alpha_i \beta_i \tag{10}$$

o produto interno de dois vetores quaisquer α e β, onde α_i e β_i são as componentes dos vetores α e β, respectivamente e *p* é o número de componentes dos vetores.
O operador de diferença Δ atua sobre o sobrescrito n e tem-se

$$\varepsilon_{2k}^{(n)} = e_k\left(S_n\right) = s \ , \tag{11}$$

e

$$\varepsilon_{2k+1}^{(n)} = \frac{y}{\left\langle y, e_k\left(\Delta S_n\right)\right\rangle}, \ k, n = 0,1,... \ . \tag{12}$$

Além das extrapoladores citados, extrapolações repetidas [6] (extrapolação de dados já extrapolados) foram também feitas para os métodos Aitken e Mitin. Para a extrapolação de Aitken repetida, por exemplo, as cinco últimos soluções obtidas pelo método *Multigrid* são utilizadas $\left(\varphi_{1,0}, \varphi_{2,0}, \varphi_{3,0}, \varphi_{4,0}, \varphi_{5,0}\right)$, onde $\varphi_{g,m}$ representa a g^a solução e o m^o nível de extrapolação. Para o primeiro nível da solução, em que nenhuma extrapolação foi feita, extrapolações são avaliadas usando: $\left(\varphi_{1,0}, \varphi_{2,0}, \varphi_{3,0}\right)$, $\left(\varphi_{2,0}, \varphi_{3,0}, \varphi_{4,0}\right)$ e $\left(\varphi_{3,0}, \varphi_{4,0}, \varphi_{5,0}\right)$, que gera, respectivamente, $\varphi_{3,1}$, $\varphi_{4,1}$ e $\varphi_{5,1}$, quando m = 1 representando o primeiro nível de extrapolação; como um exemplo, $\varphi_{3,1}^{Aitken} = \varphi_{3,1}$ na equação

$$\varphi_{3,1}^{Aitken} = \frac{\varphi_{1,0}\varphi_{3,0} - \varphi_{2,0}^2}{\varphi_{3,0} - 2\varphi_{2,0} + \varphi_{1,0}} \tag{13}$$

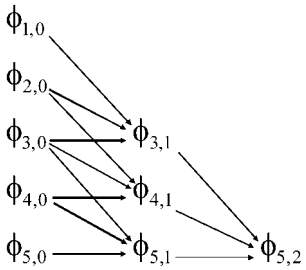

Fig. 1. Representação da Múltipla extrapolação de Aitken com dois níveis.

Depois disso, estas três soluções extrapoladas são utilizados para calcular uma nova extrapolação, gerando a solução no segundo nível de extrapolação. A Figura 1 mostra um esquema da extrapolação de Aitken repetida para dois níveis.

O procedimento para as extrapolações repetidas usando o método de Mitin é análogo ao de Aitken, porém, são necessárias nove soluções no primeiro nível para que a extrapolação do segundo nível seja realizada.

III. MODELO MATEMÁTICO

O objetivo deste estudo é mostrar como o uso de extrapoladores associados ao método *Multigrid* pode atuar sobre o erro de iteração e o tempo do processo iterativo. Para isso, escolheu-se dois problemas, um mais simples (equação de Poisson), e outro mais complexo (Equação de Burgers), a fim de se utilizar o primeiro problema para analisar especificamente os efeitos desta metodologia, permitindo que fosse detectado qual extrapolador gerou melhores resultados, para que na sequência, pudesse ser observado, como tal extrapolador agiria no segundo problema, um sistema de equações não-lineares.

Assim, neste trabalho, resolveu-se, primeiramente, o problema de condução de calor linear bidimensional em regime permanente descrito pela equação de Poisson, em um domínio quadrado de lado unitário, com condições de contorno de Dirichlet e solução analítica obtida pelo método de soluções fabricadas e dadas em [13].

Em seguida, resolveu-se as equações de Burgers, dadas por um sistema de equações diferenciais parciais não lineares, do tipo advecção-difusão. Consistem em uma forma simplificada das equações de Navier-Stokes, reduzidas às equações QML (Quantidade de Movimento Linear), pois uma vez que o campo de pressões é prescrito, a equação de conservação da massa não é necessária O modelo considerado neste trabalho tem solução analítica e campo de pressões prescrito, dados por [14], sendo que foram consideradas propriedades constantes, regime permanente e coordenadas cartesianas.¸com solução analítica, obtida pelo método das soluções fabricadas dadas em [15].

IV. MODELO NUMÉRICO

O modelo numérico foi obtido discretizando-se a equação de Poisson com o método das diferenças finitas (MDF) [13], com aproximações por diferença central (CDS – Central Difference Scheme). Já o modelo numérico da discretização das equações de Burgers foi obtido utilizando-se o método dos volumes finitos (MVF) [16], onde a discretização do domínio é realizada mediante o uso de malhas estruturadas e uniformes por direção. Neste procedimento, os termos difusivos são aproximados com esquema CDS de segunda ordem. Os termos advectivos, por sua vez são aproximados implicitamente com UDS de primeira ordem, através do procedimento da correção adiada. As condições de contorno, de Dirichlet, são aplicadas de acordo com a técnica dos volumes fictícios. Esta técnica, além de ser de fácil aplicação e respeitar o princípio de conservação para todo domínio [16], facilita a implementação dos procedimentos de restrição e prolongação no método *Multigrid*.

O solver utilizado nas suavizações foi o Gauss-Seidel red-black [12], que beneficia a programação paralela, mas apresenta bons resultados também na serial [17]. O critério de parada para todas as simulações foi a norma adimensionalizada do resíduo com base na estimativa inicial menor que determinada tolerância. A estimativa inicial foi o vetor nulo para os campos de interesse e o número de pontos da malha mais grossa foi 3, sendo que em todas as simulações, o método *Multigrid* partiu da malha mais fina e foi até a mais grossa possível. Para ambos os problemas resolvidos, o número de iterações internas do solver foi $\nu_1 = \nu_2 = 3$, sendo que para as equações de Burgers, foi utilizado o Full *Multigrid* (FMG), que se baseia na ideia de que uma boa estimativa inicial pode reduzir consideravelmente o tempo de CPU, dado que a resolução de um problema com uma boa estimativa inicial requer poucas iterações na malha fina. Sendo assim, no FMG, para se obter uma boa estimativa inicial, interpola-se uma solução da malha grossa para a malha fina. Dessa forma, os erros são suavizados eficientemente, o que garante uma ótima taxa de convergência. De acordo com [3] estas propriedades fazem do FMG a mais eficiente versão do *Multigrid* e [17] e [18] consideram o FMG o método preferido para acelerar o *Multigrid*.

V. RESULTADOS

Para a equação de Poisson foram estudados nove casos, resultantes da combinação dos valores relativos ao dimensionamento da malha mais fina (N = 129x129, N = 1025x1025 e N = 4097x4097)) e ao critério de parada (10^{-6}, 10^{-10} e 10^{-15}). As extrapolações se deram em dois momentos: ao final do processo iterativo e durante o processo iterativo.

Para esta metodologia, cada caso foi resolvido de três maneiras distintas:

(a) usando apenas o método *Multigrid* até atingir o critério de parada (MG);

(b) usando o método *Multigrid* com um ciclo V além daqueles necessários para se atingir o critério de parada (MG + 1 ITE); e

(c) usando o método *Multigrid* até atingir o critério de parada e extrapolando as soluções obtidas nas últimas iterações (MG + Extrapolador).

Os parâmetros analisados foram: tempo de CPU (t_{CPU}) em segundos (s), pico de memória de armazenamento M(MB), norma adimensionalizada do resíduo com base na estimativa inicial ($||R||_2$), fator de convergência empírico $q^{(k)}$ [10], fator de convergência média empírico $\hat{q}^{(k)}$ [13], norma infinito do erro de iteração ($||E_n||_\infty$) e norma euclidiana do erro de iteração ($||E_n||_2$), sendo que o erro de iteração é obtido através da diferença entre a solução obtida na iteração n e a solução obtida levando-se o processo iterativo até o erro de máquina. A Tabela 1 apresenta os resultados para N = 1025x1025 e a tolerância 10^{-15}. Os resultados referentes aos outros casos foram omitidos por serem análogos.

Percebe-se que, para este caso, o uso do extrapolador Épsilon topológico foi a metodologia que apresentou os melhores resultados para a norma do resíduo, para os fatores de convergência e para as normas do erro de iteração. Desta forma, conclui-se que o extrapolador que se mostrou mais eficiente na maioria dos parâmetros foi o Épsilon topológico. Em relação ao tempo de CPU e a memória, os valores para esses dois parâmetros mostraram-se maiores para todos os problemas resolvidos com algum extrapolador. Isso ocorre porque os vetores com as soluções usadas nos cálculos das extrapolações precisam ser armazenados e os cálculos das extrapolações demandam um tempo excedente em comparação com o simples uso do método *Multigrid*. Os resultados para os outros casos foram análogos.

Em seguida, as extrapolações usando o extrapolador Épsilon Topológico foram realizadas durante o *Multigrid*. Nesta abordagem, a cada cinco soluções, estas são combinadas com o extrapolador e geram uma nova solução. Esta solução serve de estimativa inicial para o próximo ciclo *Multigrid*. Ao se obter outras cinco soluções, uma nova extrapolação é realizada e assim por diante, até se atingir o critério de parada estabelecido. Neste ponto do estudo, resolveu-se também o problema modelado pelas equações de Burgers.

A Tabela 2 apresenta os resultados para N = 4097x4097 e tolerância 10^{-15} para a equação de Poisson e os resultados para N = 1025x1025 e tolerância 10^{-12} para as equações de Burgers. Os resultados referentes aos outros casos foram omitidos por serem análogos. Nesta tabela apenas as metodologias *Multigrid* (MG) e extrapolador Épsilon topológico durante o *Multigrid* (MG + Épsilon topológico) são comparadas.

Analisando este caso, pode-se perceber que o tempo de CPU do uso do extrapolador foi levemente maior quando comparado ao uso do *Multigrid* puro (*Multigrid* sem qualquer tipo de extrapolação) para a equação de Poisson e as equações de Burgers (entre 3% e 4% a mais). Quanto à norma adimensionalizada do resíduo, o uso do extrapolador mostrou-se mais eficiente do que o *Multigrid* puro para ambos os problemas. O mesmo pode ser percebido para o fator de convergência médio. Quanto ao erro de iteração, houve uma redução de sua magnitude em relação às duas normas, e esta redução chegou a aproximadamente 95% para a equação de Poisson e a 98% para as equações de Burgers.

TABELA 1. RESULTADOS PARA OS PARÂMETROS ESTUDADOS; EXTRAPOLAÇÕES NO FIM DO *MULTIGRID*.

Metodologia	t_{CPU} **[s]**	M **[MB]**	$\|\|\boldsymbol{R}\|\|_2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\|\|\boldsymbol{E}_n\|\|_\infty$	$\|\|\boldsymbol{E}_n\|\|_2$
MG	21.871	119.640	$6.272\text{x}10^{-16}$	$4.238\text{x}10^{-2}$	$4.149\text{x}10^{-2}$	$3.672\text{x}10^{-17}$	$1.816\text{x}10^{-17}$
MG + 1 ITE	23.775	119.636	$2.663\text{x}10^{-17}$	$4.246\text{x}10^{-2}$	$4.157\text{x}10^{-2}$	$1.553\text{x}10^{-18}$	$7.749\text{x}10^{-19}$
MG + Aitken	119.076	218.468	$3.145\text{x}10^{-14}$	$5.014\text{x}10^{+1}$	$7.495\text{x}10^{-2}$	$4.118\text{x}10^{-18}$	$7.301\text{x}10^{-20}$
MG + Empírico	119.076	218.468	$3.145\text{x}10^{-14}$	$5.014\text{x}10^{+1}$	$7.495\text{x}10^{-2}$	$4.118\text{x}10^{-18}$	$7.301\text{x}10^{-20}$
MG + Mitin	119.029	251.360	$8.701\text{x}10^{-14}$	$1.387\text{x}10^{+2}$	$8.158\text{x}10^{-2}$	$9.630\text{x}10^{-18}$	$3.349\text{x}10^{-19}$
MG + Épsilon Escalar	29.874	251.556	$3.044\text{x}10^{-11}$	$4.854\text{x}10^{+4}$	$1.329\text{x}10^{-1}$	$5.152\text{x}10^{-15}$	$7.111\text{x}10^{-18}$
MG + Rho Escalar	29.921	251.564	$4.511\text{x}10^{-12}$	$7.193\text{x}10^{+3}$	$1.133\text{x}10^{-1}$	$2.066\text{x}10^{-14}$	$1.000\text{x}10^{-14}$
MG + Épsilon Topológico	121.525	399.444	$1.648\text{x}10^{-19}$	$2.628\text{x}10^{-4}$	$2.721\text{x}10^{-2}$	$1.213\text{x}10^{-21}$	$5.165\text{x}10^{-22}$
MG + Rho Topológico	31.169	399.656	$3.498\text{x}10^{-13}$	$5.577\text{x}10^{+2}$	$9.161\text{x}10^{-2}$	$2.066\text{x}10^{-14}$	$9.998\text{x}10^{-15}$
MG + Aitken repetida	29.952	251.576	$4.716\text{x}10^{-12}$	$7.519\text{x}10^{+3}$	$1.137\text{x}10^{-1}$	$7.045\text{x}10^{-16}$	$1.856\text{x}10^{-18}$
MG + Mitin repetida	30.405	317.368	$4.702\text{x}10^{-10}$	$7.497\text{x}10^{+5}$	$1.669\text{x}10^{-1}$	$6.402\text{x}10^{-14}$	$1.098\text{x}10^{-16}$

TABELA 2. RESULTADOS PARA OS PARÂMETROS ESTUDADOS. EXTRAPOLAÇÕES DURANTE OS CICLOS *MULTIGRID*.

Equação de Poisson						
Metodologia	t_{CPU} **[s]**	M **[MB]**	$\|\|\boldsymbol{R}\|\|_2$	$\hat{q}^{(k)}$	$\|\|\boldsymbol{E}_n\|\|_\infty$	$\|\|\boldsymbol{E}_n\|\|_2$
MG Puro	351.579	1.844	$6.270\text{x}10^{-16}$	$4.148\text{x}10^{-2}$	$3.673\text{x}10^{-17}$	$1.818\text{x}10^{-17}$
MG + Épsilon Topológico	364.699	3.946	$2.320\text{x}10^{-16}$	$1.831\text{x}10^{-2}$	$1.912\text{x}10^{-18}$	$8.251\text{x}10^{-19}$

Equação de Burgers										
Metodologia	t_{CPU} **[s]**	M **[MB]**	u				v			
			$\|\|\boldsymbol{R}\|\|_2$	$\hat{q}^{(k)}$	$\|\|\boldsymbol{E}_n\|\|_\infty$	$\|\|\boldsymbol{E}_n\|\|_2$	$\|\|\boldsymbol{R}\|\|_2$	$\hat{q}^{(k)}$	$\|\|\boldsymbol{E}_n\|\|_\infty$	$\|\|\boldsymbol{E}_n\|\|_2$
MG Puro	3.25	290.1	$1.385\text{x}10^{-13}$	$2.681\text{x}10^{-3}$	$3.392\text{x}10^{-12}$	$6.697\text{x}10^{-13}$	$2.241\text{x}10^{-10}$	$1.175\text{x}10^{-3}$	$1.545\text{x}10^{-12}$	$6.322\text{x}10^{-13}$
MG + Épsilon Topológico	3.349	596.1	$5.941\text{x}10^{-14}$	$6.246\text{x}10^{-3}$	$2.947\text{x}10^{-13}$	$1.338\text{x}10^{-13}$	$7.208\text{x}10^{-11}$	$2.040\text{x}10^{-2}$	$9.654\text{x}10^{-14}$	$1.278\text{x}10^{-14}$

A Fig. 2 apresenta o comportamento da norma do resíduo (em escala logarítmica) em função do número de iterações para a equação de Poisson.

De acordo com a Fig. 2, percebe-se que até a quinta iteração, o resíduo cai igualmente para as duas metodologias. Porém, como na próxima iteração ocorre a primeira extrapolação, a norma do resíduo da metodologia com extrapolador fica com magnitude menor que a metodologia com *Multigrid* puro (em torno de duzentas vezes menor). Com toda esta redução, o critério de parada é atingido na nona iteração com MG + Épsilon topológico, enquanto são necessárias 11 iterações para interromper o processo quando somente o *Multigrid* é utilizado para resolver o problema.

Buscando resultados mais precisos em relação ao tempo de CPU, foram realizados testes adicionais para a equação de Poisson, com N variando de 33x33 a 8193x8193 e tolerância de 10^{-20} e para as equações de Burgers, com N variando de 5x5 a 1025x1025 e tolerância de 10^{-12}. Nestes testes, pode-se perceber que o uso do extrapolador mostra-se mais vantajoso com o aumento de N, pois reduz o tempo de CPU em relação

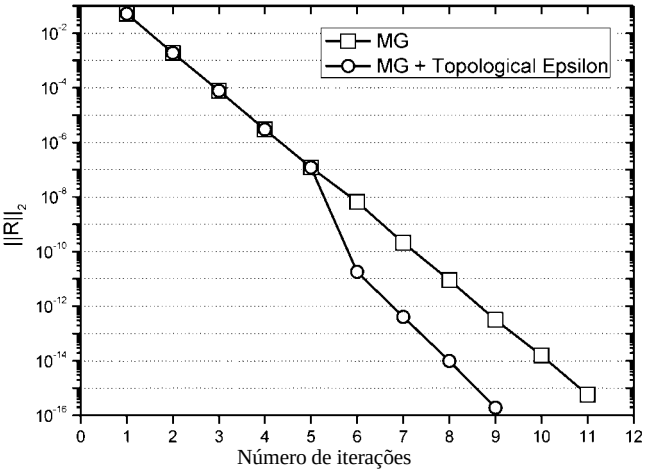

Figura 2. $||R||_2$ versus número de iterações para N = 4097x4097.

ao uso do *Multigrid*. Para confirmar tal melhora, foi calculado o speed-up da metodologia MG em relação à metodologia MG + Épsilon topológico. O speed-up (S_p) é definido como a razão entre os tempos de CPU de dois algoritmos [19].

$$S_p = \frac{t_{CPU}\left(\text{algoritmo A}\right)}{t_{CPU}\left(\text{algoritmo B}\right)} \ . \qquad (14)$$

Na Tabela 3 aparece o speed-up da metodologia MG (algoritmo A) em relação à metodologia MG + Épsilon topológico (algoritmo B) para para a equação de Poisson e para as equações de Burgers.

Com base na Tabela 3, percebe-se que a partir de , para a equação de Poisson, o valor do speed-up passa a ser maior que 1, indicando que MG + Épsilon topológico passa a ser mais rápido do que MG.

Foi feito ainda um ajuste de curvas por mínimos quadrados para

$$t_{CPU}\left(N\right) = cN^p \,, \qquad (15)$$

onde p descreve o grau de complexidade do algoritmo. Estes resultados são apresentados na Tabela 4.

De acordo com os dados da Tabela 4, pode-se afirmar que as duas metodologias são praticamente equivalentes, sendo que o uso do extrapolador Épsilon topológico acelera o método *Multigrid* com o aumento do número de incógnitas em ambos os problemas.

Pode-se perceber pela Fig. 3 que os valores de memória aumentam com o uso dos extrapoladores, mas o aumento percentual tem caráter assintótico, o que é um bom sinal, pois se busca resolver problemas em malhas bem refinadas.

TABELA 3. SPEED-UP DO *MULTIGRID* PURO EM RELAÇÃO AO MG + ÉPSILON TOPOLÓGICO; $\varepsilon = 10^{-20}$ PARA A EQUAÇÃO DE POISSON E $\varepsilon = 10^{-12}$ PARA A EQUAÇÃO DE BURGERS.

N	***Sp* (Poisson)**	***Sp* (Burgers)**
5x5	-	1.000
9x9	-	1.000
17x17	-	0.500
33x33	0.966	0.500
65x65	0.929	0.714
129x129	0.908	0.769
257x257	0.930	0.900
513x513	0.938	0.958
1025x1025	1.013	0.970
2049x2049	1.011	-
4097x4097	1.011	-
8193x8193	1.011	-

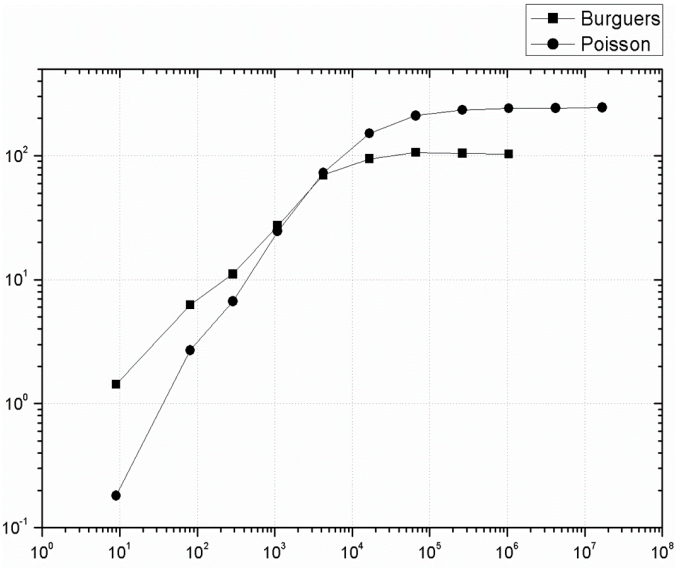

Figura 3. Incremento relativo da exigência de memória RAM em função do número de incógnitas dos problemas.

VI. CONCLUSÃO

Com base nos resultados obtidos neste trabalho, verificou-se que:

1) O uso dos extrapoladores ao final do *Multigrid*, para a equação de Poisson, fez o tempo de CPU aumentar em relação a um ciclo adicional do *Multigrid*.

2) A memória de armazenamento é maior quando se usa um extrapolador.

3) O extrapolador Épsilon topológico usado tanto ao final, quanto durante o *Multigrid* foi a metodologia que mais reduziu a norma adimensionalizada do resíduo, os fatores de convergência e a magnitude do erro de iteração para a equação de Poisson.

4) Os tempos de CPU das extrapolações realizadas durante o *Multigrid* são praticamente iguais aos obtidos apenas com *Multigrid*, sendo que a metodologia com extrapolador passa a ser levemente mais rápida com o aumento do número de incógnitas do problema.

REFERÊNCIAS

[1] C. Brezinski, R. M. Zaglia, A Review of Vector Convergence Acceleration Methods, With Applications to Linear Algebra Problems. Int. J. Quantum Chem., 109 (2008) 1631–1639.

[2] W. L. Briggs, V. E. Henson, S. F. McCormick, A *Multigrid* Tutorial, 2 ed. Philadelphia: SIAM, 2000.

TABELA 4. VALORES DE *p* E *c* DA EQ. (15), PARA CADA ALGORITMO DAS METODOLOGIAS ESTUDADAS.

Metodologia	**Poisson**		**Burgers**	
	c	***p***	***c***	***p***
Pure MG	0.0755	1.03258	1.3547×10^{-6}	1.05962
MG + Topological Epsilon	0.0850	1.02957	3.599×10^{-6}	0.99137

[3] U. Trottenberg, C. Oosterlee, A. Schüller, *Multigrid*, St Augustin, Germany: Academic Press, 2001.

[4] W. Hackbush, *Multigrid* Methods and Applications. Berlin: Springer-Verlag, 1985.

[5] A. V. Mitin, Linear extrapolation in an iterative method for solving systems of equations, U.S.S.R. Comput. Math. Math.+, 25-2 (1985) 1-6.

[6] R. L. Burden, J. D. Faires, Numerical analysis, eighth edition, Cengage Learning – Brooks Cole, Florence, 2005.

[7] M. A. Martins, C. H. Marchi. Estimate of iteration errors in Computational Fluid Dynamics, Num. Heat Tr. B-Fund., 53 (2008) 234-245.

[8] J. P. Delahaye, B. Germain-Bonne, The set of logarithmically convergent sequences cannot be accelerated. Philadelphia: SIAM, 1982.

[9] J. P. Delahaye, Sequence Transformations. Berlin: Springer, 1988.

[10] P. R. Graves-Morris, D. E. Roberts, A. Salamc, The epsilon algorithm and related topics, J. Comput. Appl. Math., 122 (2000) 51-80.

[11] Q. Gao, Z. Jiang, T. Liao, K. Song, Application of the vector ε and ρ extrapolation methods in the acceleration of the Richardson–Lucy algorithm, Opt. Commun., 283 (2010) 4224–4229.

[12] P. Wynn, Acceleration techniques for iterated vector and matrix problems. Math. Comput., 16 (1962) 301-322.

[13] J. C. Tannehill, D. A. Anderson, R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Washington: Taylor & Francis, 1997.

[14] T. M. Shih, C. H. Tan, B. C. Hwang, Effects of grid staggering on numerical scheme. Int. J. Num. Met. Fluids, 9 (1989) 193-212

[15] C. J. Roy, Review of Code and Solution Verification Procedures for Computational Simulation. J. Comput. Phys., 205 (2005) 131-156

[16] C. R. Maliska, Computational Heat Transfer and Fluid Mechanics [in Portuguese], 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004.

[17] J. Zhang, Acceleration of Five-Point Red-Black Gauss-Seidel in *Multigrid* for Poisson equation. Appl. Math. Comput., 80 (1996) 73-93.

[18] A. Thekale, T. Grandl, K. Klamroth, U. Rüde, Optimizing the number of *Multigrid* cycles in full *Multigrid* algorithm. Numer. Linear Algebr., 17 (2010) 199-210.

[19] G. Galante, *Multigrid* Parallel Methods on Non-structured grids Applied to Simulation of Computational Fluid Dynamic and Heat Transfer Problems (in Portuguese). PhD Thesis. Federal University of Rio Grande do Sul, Porto Alegre, RS, 2006.