CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

COMPARISON BETWEEN EXTRAPOLATION TECHNIQUES ASSOCIATED TO THE MULTIGRID METHOD APPLIED IN THE NAVIER-STOKES EQUATIONS

Bruno B. Rutyna

brunorutyna@gmail.com

Federal University of Paraná, Graduate Program in Numerical Methods in Engineering (PPGMNE)

Centro Politécnico, 81531-980 Curitiba-PR, Brazil

Marcio A. V. Pinto

marcio_villela@ufpr.br

Federal University of Paraná, Department of Mechanical Engineering

Centro Politécnico, 81531-980 Curitiba-PR, Brazil

Réverton L. A. Neundorf

Márcio A. M. Anunciação

revertonluis@gmail.com

marcioalexandro@ufpr.br

Federal University of Paraná, Graduate Program in Numerical Methods in Engineering (PPGMNE)

Centro Politécnico, 81531-980 Curitiba-PR, Brazil

Márcio A. Martins

mandre@unicentro.br

State University of Centro-Oeste, Department of Mathmatics

Camargo Varela St. 3, Vila Carli District, 85040-080 Guarapuava-PR, Brazil

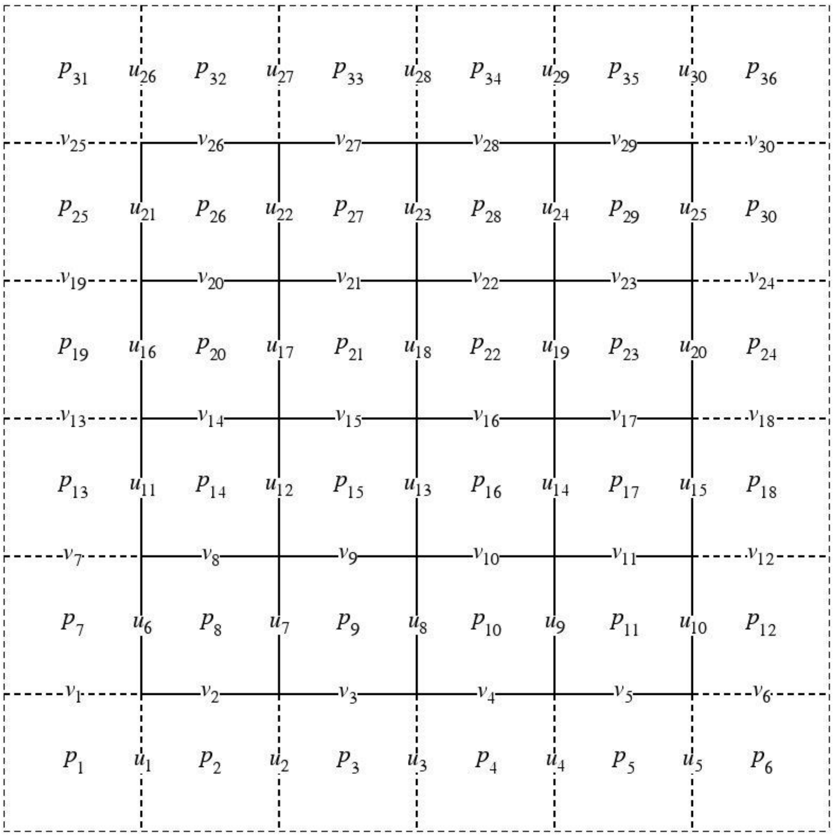
Abstract. *Among the problems of Computational Fluid Dynamics (CFD), the one that deserves attention is finding satisfactory solutions for the Navier-Stokes equations, which in this work they represent a case of two-dimensional laminar flow of a time-dependent incompressible fluid. Given the difficulty to obtain the analytical solutions, to use iterative computational methods is* *a feasible option. In this work it was decided to solve them iteratively through the projection method. Nevertheless, the approximations obtained during the iterations can converge very slowly, thus consuming an undesired computational time. Therefore, it becomes useful to use the multigrid method to solve these systems of equations originated from the discretized problem, which is a powerful convergence accelerator and has low computational cost. Further gain can be achieved by applying solution extrapolation techniques, which can be used after the multigrid cycles or adapted to intercalate these cycles. This work will first analyze the application of extrapolators in the post-multigrid case. The extrapolator that obtained the best performance was used in the second stage of this work, which consists in applying it between the cycles of the multigrid method. In the end, concluding which extrapolator is most effective when used associated with multigrid method.*

Keywords: *Multigrid Method, Extrapolation Methods, Navier-Stokes, Convergence Acceleration, Computational Fluid Dynamics (CFD)*

1 INTRODUCTION

The engineering problems requires even faster and most efficient solving methods. One of the challenging areas is Computational Fluid Dynamics, due to the nature of the problems. As an example, there are the Navier-Stokes equations, whose analytical solutions are known for a few cases, due to the fact that they are nonlinear partial differential equations (PDEs), an area where mathematics is not sufficiently developed (Fortuna, 2000).

To solve numerically the problem, the finite volume method (FVM) will be used, which partitions a two-dimensional domain $\{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq L_x \; and \; 0 \leq y \leq L_y\}$ in structured meshes with $N = N_x N_y$, in wich N_x and N_y are the number of volumes in coordinated directions x and y, and L represents the length of the domain. These volumes can be ordered *lexicographically*. Thus, for example, a mesh with $N_x = N_y$ = 6 is according to Fig. 1.



p_{31} u_{26} p_{32} u_{27} p_{33} u_{28} p_{34} u_{29} p_{35} u_{30} p_{36}
v_{25} v_{26} v_{27} v_{28} v_{29} v_{30}
p_{25} u_{21} p_{26} u_{22} p_{27} u_{23} p_{28} u_{24} p_{29} u_{25} p_{30}
v_{19} v_{20} v_{21} v_{22} v_{23} v_{24}
p_{19} u_{16} p_{20} u_{17} p_{21} u_{18} p_{22} u_{19} p_{23} u_{20} p_{24}
v_{13} v_{14} v_{15} v_{16} v_{17} v_{18}
p_{13} u_{11} p_{14} u_{12} p_{15} u_{13} p_{16} u_{14} p_{17} u_{15} p_{18}
v_7 v_8 v_9 v_{10} v_{11} v_{12}
p_7 u_6 p_8 u_7 p_9 u_8 p_{10} u_9 p_{11} u_{10} p_{12}
v_1 v_2 v_3 v_4 v_5 v_6
p_1 u_1 p_2 u_2 p_3 u_3 p_4 u_4 p_5 u_5 p_6

Figure 1: Lexicographical ordering

A feature of the discretization of PDEs through the finite volume method is that a linear systems of the type

$$Au = f \tag{1}$$

is obtained, in wich A is the coefficient matrix, f is a vector of independent terms and u, an unknown vector. The Eq. (1) can be solved by direct methods such as Gauss elimination, or iterative, such as Jacobi's method or Gauss-Seidel method. It is worth mentioning that this last class has a lower computational cost than the direct methods for large-sized systems (Burden and Faires, 2008).

There is also an improved way to solve the Eq. (1), using the multigrid method. This method uses a set of meshes in wich the iterations are performed on different layers from the higher level of refinement (which is the original discretization of the problem) to less refined levels, through operators who perform transfer of information between them (Briggs et al., 2000).

An example of the use of these meshes can be seen in the Fig 2.

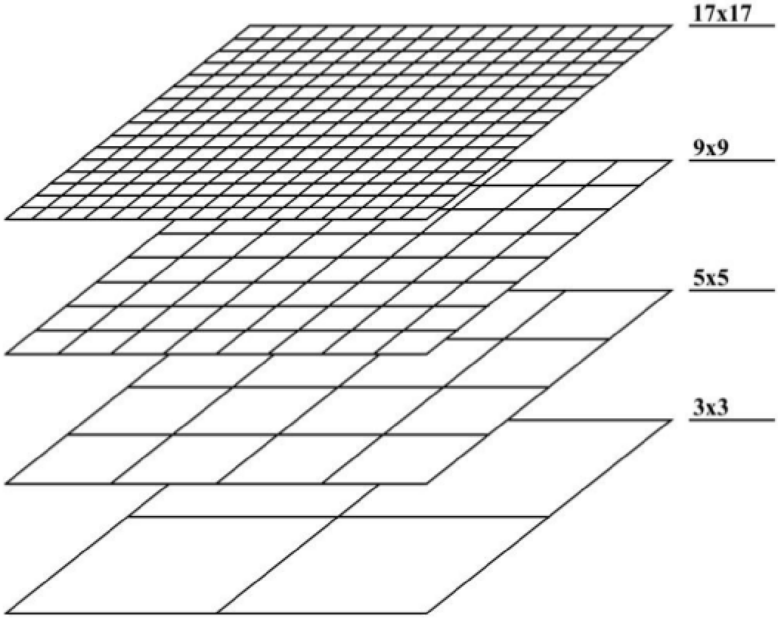


Figure 2: Meshes generation process (adapted from http://www.mgnet.org/mgnet/tutorials/xwb/mg.html)

In order to improve the convergence of the multigrid method, we can also use extrapolation techniques or convergence acceleration associated with iterative methods (Brezinski, 1974), thus obtaining numerical solutions with less computational effort. For exemple, in Anunciação et al. (2016), it is shown that an extrapolation iteration is more efficient than a multigrid cycle and it reduces the operations number when applied to the Burgers equations, thus justifying its use.

2 MATHEMATICAL MODEL

The Navier-Stokes equations can be written in many ways depending on the properties of the fluid and the flow (Fortuna, 2000). It was chosen to represent it in a similar way from Kim and Moin (1985), in order to facilitate the adaptation and understanding of the algorithms used. Therefore, it will handled with two-dimensional laminar flow of an incompressible fluid dictated by

$$\nabla \cdot u = 0, \tag{2}$$

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

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

being x and y the spatial coordinates; t the temporal coordinate; u and v the coordinates of the velocity vector in the directions x and y; p, the fluid pressure; Re, the Reynolds number; ∇u, the gradient of u; $\nabla \cdot u$, the divergence of u and ∇^2, the laplacian operator. To complete,

$$Re = \frac{\rho\, v_m D}{\mu}, \tag{5}$$

with ρ being the fluid specific mass; μ, its dynamic viscosity; v_m, its average speed and D, the tube diameter.

This work solves the Taylor-Green vortex problem (Green and Taylor, 1937), whose domain is given by

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

The analytical solutions are:

$$\begin{aligned} u &= -(\cos x \cdot \sin y)e^{-2(t+Ti)}, \\ v &= (\sin x \cdot \cos y)e^{-2(t+Ti)}, \\ p &= -\frac{1}{4}(\cos 2x + \sin 2y)e^{-4(t+Ti)}, \end{aligned} \tag{7}$$

where Ti is the initial time and $t = k\Delta t$, where k it's the time step. In this work, $Ti = 0$. The analytical solutions generate the initial and boundary conditions of the problem. The reason for knowing an analytical solution in this work is to compare it with the obtained numerical solution and then validate the method used.

3 NUMERICAL MODEL

3.1 Projection Methods

The projection method used in this work was described in Timmermans et al. (1996), and it is showed below:

First step:

$$\frac{(3\mathbf{u}^t - 4\mathbf{u}^n + \mathbf{u}^{n-1})}{2h_t} = \beta_1 g_1(\mathbf{u}^n) + \beta_0 g_0(\mathbf{u}^{n-1}) + \nu\nabla^2\mathbf{u}^t + \nabla P^n, \tag{8}$$

$$\mathbf{u}^t|_{\partial\Omega} = \mathbf{b}^n, \tag{9}$$

where $\mathbf{u}^t$ is the auxiliary speed field, $\mathbf{u}^n$ the velocity field in the time step n, h_t is the temporal refinement, P^n is the pressure in the time step n, $\mathbf{b}^n$ the boundary conditions in the time step n, β is constant and $\beta\mathbf{g}$ represents the convective terms $\mathbf{u} \cdot \nabla\mathbf{u}$. The choices of β_1 and β_0 imply whether the convective terms will be treated explicitly or implicitly. In this work, $\beta_1 = -3/2$ and $\beta_0 = 1/2$ (Bell et al., 1989).

Second step:

$$\nabla^2\phi^{n+1} = \frac{3}{2h_t}\nabla \cdot \mathbf{u}^t \tag{10}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^t - \frac{2h_t}{3}\nabla\phi^{n+1} \tag{11}$$

$$P^{n+1} = P^n + \phi^{n+1} - \nu\nabla \cdot \mathbf{u}^t \tag{12}$$

where ϕ is the pressure correction term and $\nu = 1/Re$. This version of the projection method has boundary conditions in the pressure given by

$$\partial_n P^{n+1}|_{\partial\Omega} = -\nu(\nabla \times \nabla \times u^{n+1}) \cdot \mathbf{n}|_{\partial\Omega}, \tag{13}$$

which are consistent boundary conditions for it. The algorithm given by steps one and two is known (Guermond and Shen, 2001, 2004) as incremental correction in pressure in rotational form. Guermond and Shen (2004) show that the incremental version in rotational form is of second order for velocities in the L_2, L_1 and L_∞ norms. For the pressure, the scheme is of second order in the L_2 and L_1 norms, and of order $3/2$ in the L_∞ norm.

The use of the projection method has advantages when compared to other methods, such as the SIMPLE method. The resulting equations avoid coupling in the speed variables and they are treated separately, reducing the complexity of the algorithm (Griffith, 2009). For a generalization of projection methods see Guermond et al. (2006).

3.2 Discretization

This work presents staggered grid, of the type Harlow and Welch (1965), also found in Maliska (1995). The velocities are found on the faces and the pressure in the centers of the volumes, thus avoiding pressure instabilities (Shih et al., 1989).

Again in Fig. 1, the volumes with dashed lines represent the dummy volumes and do not belong to the physical domain of the problem. The boundary conditions for the variable u are prescribed in the west and east contours, using linear extrapolation in the north and south contours. In an analogous way, the variable v in the north and south contours, with linear extrapolation in the other two directions.

The mass conservation equation, Eq. (2), can be rewritten as:

$$\oint_S \mathbf{u} \cdot \mathbf{n} dS = 0, \tag{14}$$

where $\mathbf{n}$ represents the time step.

The equation of linear momentum conservation in the direction x (QML-x), Eq. (3), is:

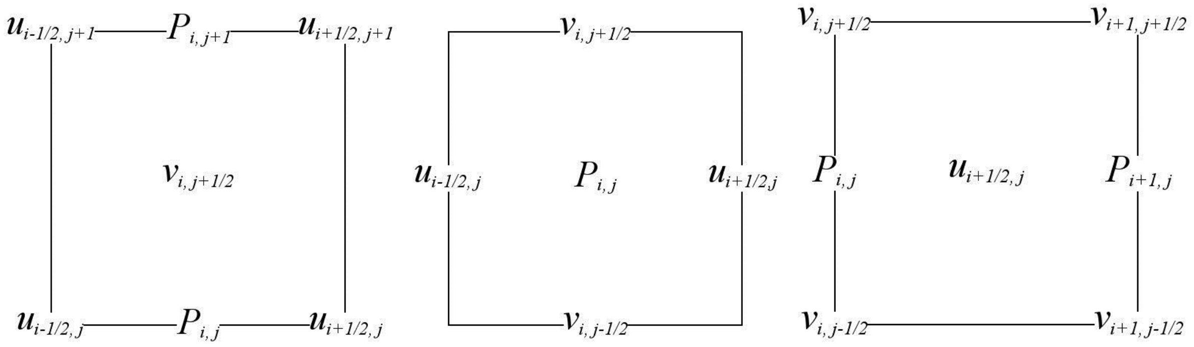
$$\frac{\partial}{\partial t}\int_V u dV = -\oint_S u\mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_x dS + \frac{1}{Re}\oint_S \nabla u \cdot \mathbf{n} dS, \tag{15}$$

and the equation of linear momentum conservation in the direction y (QML-y), Eq. 4:

$$\frac{\partial}{\partial t}\int_V v dV = -\oint_S v\mathbf{u} \cdot \mathbf{n} dS - \oint_S p n_y dS + \frac{1}{Re}\oint_S \nabla v \cdot \mathbf{n} dS. \tag{16}$$

where n_x and n_y are normal vectors in x and y directions, respectively.

The Equations (14) to (16) can be discretized using staggered grid by calculating the integrals on the control volumes shown in Fig. 3. For simplicity, it is assumed that $h_x = h_y = h$ (square mesh). The temporal discretization depends on the projection method used and can be seen in detail in Guermond et al. (2006).



(a) Volume centered on v (b) Volume centered on P (c) Volume centered on u

Figure 3: Control Volumes

The Equation (14) is discretized using a volume centered in P, according to Fig. 3(b):

$$\oint_S \mathbf{u}\cdot\mathbf{n}dS \approx u^{n+1}_{i+1/2,j} - u^{n+1}_{i-1/2,j} + v^{n+1}_{i,j+1/2} - v^{n+1}_{i,j-1/2} = 0. \tag{17}$$

The terms of the pressures are discretized using the volumes shown in Figs. 3(a) and 3(c) in the following way:

$$\oint_S pn_x dS \approx (p_{i+1,j} - p_{i,j})h, \tag{18}$$
$$\oint_S pn_y dS \approx (p_{i,j+1} - p_{i,j})h.$$

With the control volumes shown in Figs. 3(a) and 3(c) the diffusive terms can be discretized by

$$\oint_S \nabla u\cdot\mathbf{n}dS \approx u^n_{i+3/2,j} + u^n_{i-1/2,j} + u^n_{i+1/2,j+1} + u^n_{i+1/2,j-1} - 4u^n_{i+1/2,j}, \tag{19}$$
$$\oint_S \nabla v\cdot\mathbf{n}dS \approx v^n_{i,j+3/2} + v^n_{i,j-1/2} + v^n_{i+1,j+1/2} + v^n_{i-1,j+1/2} - 4v^n_{i,j+1/2}. \tag{20}$$

The advective terms are discretized with the volumes centered on the components of the speeds according to the Figs. 3(a) and 3(c):

$$\oint_S u\mathbf{u}\cdot\mathbf{n}dS \approx \left((u^2)^n_{i+1,j} - (u^2)^n_{i,j} + (uv)^n_{i+1/2,j+1/2} - (uv)^n_{i+1/2,j-1/2}\right)h, \tag{21}$$
$$\oint_S v\mathbf{u}\cdot\mathbf{n}dS \approx \left((v^2)^n_{i,j+1} - (v^2)^n_{i,j} + (uv)^n_{i+1/2,j+1/2} - (uv)^n_{i-1/2,j+1/2}\right)h, \tag{22}$$

With Eqs. (21) and (22), examples of approximations for velocity components can be written as

$$(u^2)^n_{i,j} = \left(\frac{1}{2}(u^n_{i+1/2,j} + u^n_{i-1/2,j})\right)^2, \tag{23}$$
$$(v^2)^n_{i,j} = \left(\frac{1}{2}(v^n_{i,j+1/2} + v^n_{i,j-1/2})\right)^2, \tag{24}$$

$$(uv)^n_{i+1/2,j+1/2} = \left(\frac{1}{2}(u^n_{i+1/2,j} + u^n_{i+1/2,j+1})\right)\left(\frac{1}{2}(v^n_{i,j+1/2} + v^n_{i+1,j+1/2})\right). \tag{25}$$

To guarantee the stability of the solutions, the time step h_t must respect the criterion (Maliska, 1995):

$$h_t < \frac{1}{2Re}\frac{h_x^2 h_y^2}{h_x^2 + h_y^2}. \tag{26}$$

In general, projection methods cause the boundary conditions on the pressure to be of the Neumann type. In order to guarantee the existence and uniqueness of the solution, should have (Briggs et al., 2000):

$$\int\limits_\Omega \nabla^2 \phi^{n+1} = \int\limits_\Omega \frac{3}{2h_t}\nabla \cdot \mathbf{u}^t = 0. \tag{27}$$

More details on projection methods can be found at Neundorf et al. (2016) and Guermond et al. (2006).

3.3 Multigrid Method

The basic solvers, such as the Jacobi method or the Gauss-Seidel method, have the property of rapidly reducing the high frequency modes of error, leaving only the soft modes. In addition, soft modes become more oscillating in coarsener meshes. Thus, it is possible to use a strategy that takes advantage of these two behaviors, which in this work is done with the multigrid method (Briggs et al., 2000).

The multigrid method is applied to systems of equations of type Eq. (1), with a discretized domain Ω_h. To start it, simply apply an iterative solver, also called smoother, such as Jacobi-weighted or Gauss-Seidel, but only a reduced number of iterations. After this, the residue can be calculated , given by:

$$r^h = f^h - A^h \mathbf{v}^h \tag{28}$$

in wich $\mathbf{v}^h$ is the approximation of **u** obtained in the mesh Ω^h and h is the distance between two subsequent points of the domain (Briggs et al., 2000).

After carrying out the above procedure, the residue is transferred to the subsequently coarsener mesh using a restriction operator. The coarsening ratio of the mesh will be equal to 2. Thus, obtaining the system:

$$A^{2h}e^{2h} = r^{2h}\ , in\ \Omega^{2h}, \tag{29}$$

being e the error to be estimated.

After obtaining an approximation of the error in the mesh Ω^{2h}, it begins the process of correction of the approximation **v**. First we use an operator that transfers the error to the a more refined mesh (prolongation operator), then we perform the following operation:

$$\mathbf{v}^h \leftarrow \mathbf{v}^h + e^{2h}. \tag{30}$$

Finally, we apply again the chosen solver in the approximation $\mathbf{v}^h$ obtained. More details about restriction and prolongation operators are found in Wesseling (1995), Villar (2007) and Trottenberg et al. (2000).

This procedure can be performed using other coarser meshes. Each time we return to the refined mesh with the correction, we have what we call a V-cycle, as shown in Fig. 4.

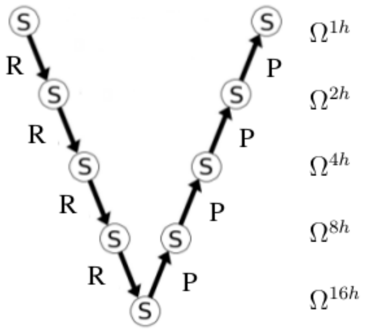


Figure 4: One cycle V Diagram applied in 5 meshes. S = smoothing, R = restriction, P = prolongation (Neundorf, 2016).

The number of times to perform V-cycles must be set by a user fixed quantity or by parameters that involve the convergence of the solution that involve the convergence of the solution.

3.4 Extrapolation Methods

In order to accelerate the convergence of iterative methods, extrapolation techniques can be used, which are based on transformations of sequences. That is, if one sequence converges slowly, it is possible to transform it into another, as long as the original limit is maintained, and this new sequence may converge faster than the first.

An example of a sequence transformation (T_n) of real or complex numbers converging to s, according to Brezinski and Zaglia (2013), is:

$$T_n = \frac{S_n + S_{n+1}}{2}, n = 0, 1, 2, ..., \tag{31}$$

or

$$T_n = \frac{S_n S_{n+2} + S_{n+1}^2}{S_{n+2} - 2S_{n+1} + S_n}, n = 0, 1, 2, ..., \tag{32}$$

where T_n is the new generated sequence, also called extrapolator.

This paper presents seven extrapolators, being five of the scalar type and two of the vector type. Whereas $C_1, C_2, ..., C_k, ..., C_\infty$ are obtained through an iterative process and $\phi_1 = C_1(k)$, $\phi_2 = C_2(k)$, $\phi_i = C_i(k)$ are the scalars for all k-th components of the vectors C_i, then the following extrapolators are defined:

$$\phi_\infty^{Aitken} = \frac{\phi_1\phi_3 - \phi_2^2}{\phi_3 - 2\phi_2 + \phi_1}, \tag{33}$$

Aitken extrapolator (Burden and Faires, 2008).

Based on the Empirical Estimator proposed in Martins and Marchi (2008),

$$\phi_{\infty}^{Empirical} = \phi_3 + \frac{(\phi_3 - \phi_2)^2}{2\phi_2 - \phi_3 - \phi_1}, \tag{34}$$

and

$$\phi_{\infty}^{Mitin} = \frac{\phi_1\phi_5 - \phi_3^2}{\phi_5 - 2\phi_3 + \phi_1}, \tag{35}$$

which is the formula used in Mitin extrapolation (Mitin, 1985).

The other two extrapolators used in this work, Epsilon and Rho, have a formal similarity between them, but they differ on convergence capabilities. The Epsilon algorithm is a powerful accelerator of iterative methods that solve discrete differential equations but for sequences that converge logically, it fails (Delahaye and Germain-Bonne, 1982; Delahaye, 2012). Conversely, Rho algorithm does not accelerate sequences with linear convergence, but is very efficient for logarithmically convergent sequences (Gao et al., 2010; Graves-Morris et al., 2000).

For a sequence (S_n), the Epsilon Scalar algorithm is given by (Brezinski, 1974):

$$\epsilon_{-1}^{(n)} = 0, \;\; \epsilon_0^{(n)} = S_n, \;\; n = 0, 1, ..., \tag{36}$$

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}}, \;\; n, k = 0, 1, ..., \tag{37}$$

for denominators other than zero. The superscript of $\tilde{\epsilon}$ represents the element and, the subscript, the iteration. Very simillar, the Rho Scalar algorithm is given by (Wynn, 1956):

$$\rho_{-1}^{(n)} = 0, \;\; \rho_0^{(n)} = S_n, \;\; n = 0, 1, ..., \tag{38}$$

$$\rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{k+1}{\rho_k^{(n+1)} - \rho_k^{(n)}}, \;\; n, k = 0, 1, ... \tag{39}$$

Both algorithms, Epsilon and Rho, have scalar and topological orientations. While the first preserves characteristics of each component of the vector individually, a second one treats to take into account the properties of the vector as a whole, and those two variations are analyzed in this work. Epsilon Topological algorithm (Tan, 1987) can be seen below:

$$\tilde{\epsilon}_{-1}^{(n)} = 0, \;\; n = 0, 1, ..., \tag{40}$$

$$\tilde{\epsilon}_0^{(n)} = S_n, \;\; n = 0, 1, ..., \tag{41}$$

$$\tilde{\epsilon}_{2k+1}^{(n)} = \tilde{\epsilon}_{2k-1}^{(n+1)} + \frac{y}{< y, \tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)} >}, \;\; k, n = 0, 1, ..., \tag{42}$$

$$\tilde{\epsilon}_{2k+2}^{(n)} = \tilde{\epsilon}_{2k}^{(n+1)} + \frac{\tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)}}{< \tilde{\epsilon}_{2k+1}^{(n+1)} - \tilde{\epsilon}_{2k+1}^{(n)}, \tilde{\epsilon}_{2k}^{(n+1)} - \tilde{\epsilon}_{2k}^{(n)} >}, \;\; k, n = 0, 1, ..., \tag{43}$$

where, as S_n are elements of E, so y is an arbitrary vector such that $y \in E^*$ (the algebric dual space of E) (Brezinski and Redivo-Zaglia, 2014) and the operator $< .,. >$ represents an inner product. Also the Topological Rho algorithm can be given by (Osada, 1991):

$$\tilde{\rho}_{-1}^{(n)} = 0, \;\; n = 0, 1, ..., \tag{44}$$

$$\tilde{\rho}_{0}^{(n)} = S_n, \;\; n = 0, 1, ..., \tag{45}$$

$$\tilde{\rho}_{2k+1}^{(n)} = \tilde{\rho}_{2k-1}^{(n+1)} + \frac{(2k+1)y}{< y, \tilde{\rho}_{2k}^{(n+1)} - \tilde{\rho}_{2k}^{(n)} >}, \;\; k, n = 0, 1, ..., \tag{46}$$

$$\tilde{\rho}_{2k+2}^{(n)} = \tilde{\rho}_{2k}^{(n+1)} + \frac{(2k+2)\tilde{\rho}_{2k}^{(n+2)} - \tilde{\rho}_{2k}^{(n+1)}}{< \tilde{\rho}_{2k}^{(n+2)} - \tilde{\rho}_{2k}^{(n+1)}, \tilde{\rho}_{2k+1}^{(n+1)} - \tilde{\rho}_{2k+1}^{(n)} >}, \;\; k, n = 0, 1, ... \tag{47}$$

Details of the Epsilon extrapolator are found in Wynn (1956) and Brezinski (1974), and of the Rho extrapolator in Brezinski and Zaglia (2013). Applications of extrapolators are seen in Anunciação (2013).

4 RESULTS

The extrapolation methods of this work were applied to the speed variables u and v for each time step t. It is possible to divide the experiment into two stages. In the first one, the extrapolator was executed after the cycles of the multigrid method and, in the second one, between their cycles. Meshes were used with $N = 128 \times 128$, $N = 512 \times 512$ and $N = 1024 \times 1024$, and stop criteria ε equals to 10^{-6}, 10^{-8} and 10^{-10}. It is expected that the use of extrapolators applied in the Navier-Stokes equations will be as advantageous as applied in the Burgers equations, as shown in Anunciação et al. (2016).

4.1 Implementation Data

The multigrid method of this work performs V-cycles, with mesh coarsening ratio equal to 2 (standard coarsening), until it reaches the coarsest mesh (Briggs et al., 2000). The associated solver is the Gauss-Seidel red-black, with pre and post-smoothing numbers equal to 3 (Oliveira, 2010).

At each step of time velocities and pressure are solved numerically. The stop criterion used in the iterative method is the dimensionless Euclidean-norm of the residue by Euclidean-norm of the residue in the initial guess (Briggs et al., 2000; Trottenberg et al., 2000):

$$||R||_2 = \frac{||R^{it}||_2}{||R^0||_2}, \tag{48}$$

where R^{it} represents the residue of the current iteration. When the Eq. (48) reaches values less than a ε tolerance, the process is terminated.

The algorithms were implemented in the Fortran 2003 language with double precision. The tests were performed in a computer with Intel processor of 2.50 GHz x 4, 4GB of RAM and 64-bit Windows operating system.

4.2 Use of post multigrid extrapolators

At this stage, the multigrid method was executed until it reached an established tolerance ε. Then, one of these two paths were taken:

- was performed a new cycle of multigrid, called "MG + 1 ITE";
- it was applied an extrapolation method, called "MG + Extrapolator".

Table 1 shows how many approximations are used for each extrapolator.

Table 1: Number of solutions used for extrapolations

Extrapolator	Number of solutions
MG + Aitken	3
MG + Empirical	3
MG + Mitin	5
MG + Scalar Epsilon	5
MG + Scalar Rho	5
MG + Topological Epsilon	5
MG + Topological Rho	5

The results were measured with the following parameters: processing time or CPU time (t_{cpu}), in seconds $[s]$; the memory peak (M) reached during the execution of the algorithm, in Megabytes [MB]; the dimensionless residual based on the initial guess ($||R||_2$), given by Eq. (48); the mean empirical convergence factor ($\hat{q}^{(k)}$), defined by (Trottenberg et al., 2000):

$$\hat{q}^{(k)} = \sqrt[k]{\frac{||R^{it}||_2}{||R^0||_2}}, \tag{49}$$

where k represents the k-th iteration. Values of $\hat{q}^{(k)}$ considered good are close to zero (Briggs et al., 2000; Burden and Faires, 2008). Finally, infinite-norms are also measured ($||E_n||_\infty$) and Euclidian-norms ($||E_n||_2$) of variables u, v and p, with E_n (also called iteration errors) being the difference obtained between the approximation in the n-th step of time and the analytical solution.

Table 2 contains the results obtained in a mesh with $N = 512 \times 512$, tolerance $\varepsilon = 10^{-8}$ and steps of time $t = 10$.

Measurements $\hat{q}^{(k)}$ e $||R||_2$ for the variable v were made at each instant of time, and can be seen in Figs. 5 and 6.

It is worth emphasizing that the parameters $\hat{q}^{(k)}$ and $||R||_2$, when measured to u, presented results practically equals as in Figs. 5 and 6.

With the tests carried out, it is noted that the Epsilon Topological extrapolator is better than the others in infinite norm parameters and Euclidean-norm of the iteration error of the variables u and v, empirical mean convergence factor and norm of the residue. The memory used is higher than the other methods, and this is justified by the number of auxiliary vectors used in

Table 2: Extrapolations in the end of Multigrid

Methodology	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 + Empirical	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 + Scalar Epsilon	60.2	116.240	1,529E-09	7,492E-08	1,369E-09	7,490E-08	4,459E-07	1,055E-04
MG + Scalar Rho	58.3	116.220	1,746E-06	3,747E-04	1,746E-06	3,747E-04	4,086E-05	5,086E-03
MG + Topological Epsilon	59.5	134.684	5,961E-10	7,465E-08	5,961E-10	7,465E-08	4,460E-07	1,055E-04
MG + Topological Rho	58.6	134.676	1,746E-06	3,748E-04	1,746E-06	3,748E-04	4,069E-05	5,066E-03

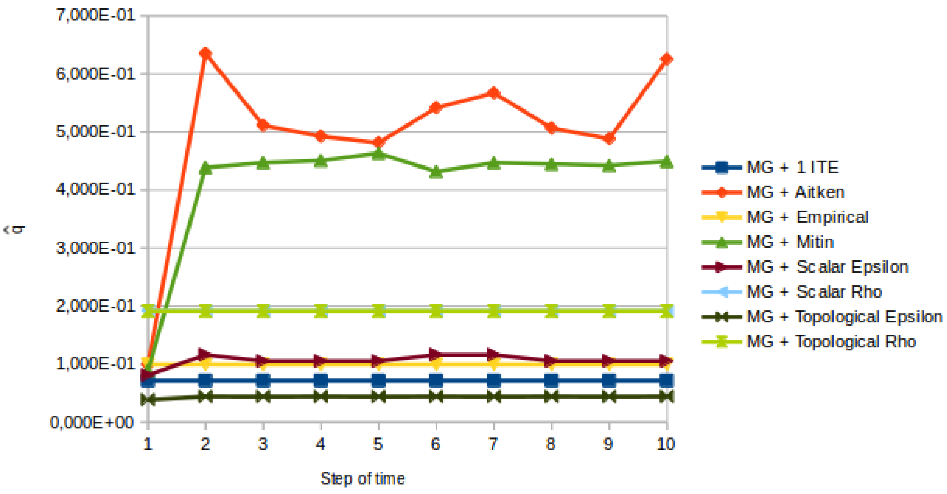


Figure 5: The mean empirical convergence factor for each step of time

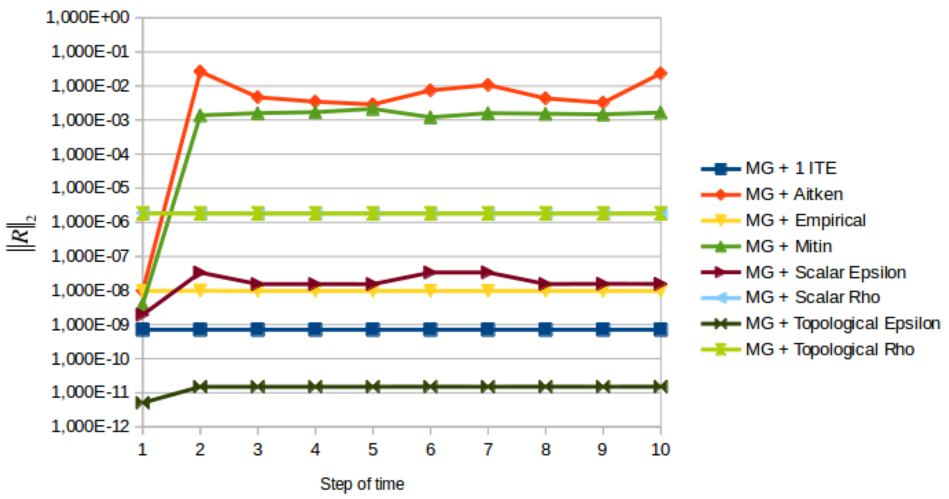


Figure 6: $\|R\|_2$ for each step of time

the extrapolation. The norms of the variable p reach values close to the best results of these columns. All this leads to conclude that the Topological Epsilon has the potential to accelerate the convergence of the presented problem in a significant way.

4.3 Extrapolators use between cycles

In this section the Topological Epsilon extrapolator was applied between the multigrid cycles. This occurs as follows: given the initial guess for the problem, five cycles of the multigrid are performed, enough for extrapolation, which is then fulfilled. Subsequently, five multigrid cycles and a new extrapolation are performed, and further, until the stopping criterion is reached.

Table 3 shows the comparison between the pure multigrid method (multigrid without the use of the extrapolator, MG) and the multigrid using Topological Epsilon extrapolator (MG + Topological Epsilon). The mesh used has $N = 1024 \times 1024$, stopping criterion $\varepsilon = 10^{-10}$ and time steps $t = 10$.

Table 3: Extrapolation between the multigrid cycles

Methodology	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	273.2	378.816	7,444E-11	7,443E-11	1,193E-07	3,171E-08	3,171E-08	5,376E-05
MG + Topological Epsilon	271.1	504.124	3,922E-11	3,916E-11	1,178E-07	9,674E-09	9,675E-09	5,370E-05

Figure 7 shows the measurements $\hat{q}^{(k)}$ e $\|R\|_2$ for the variable v at each instant of time. The variable u has similar results to the Fig. 7.

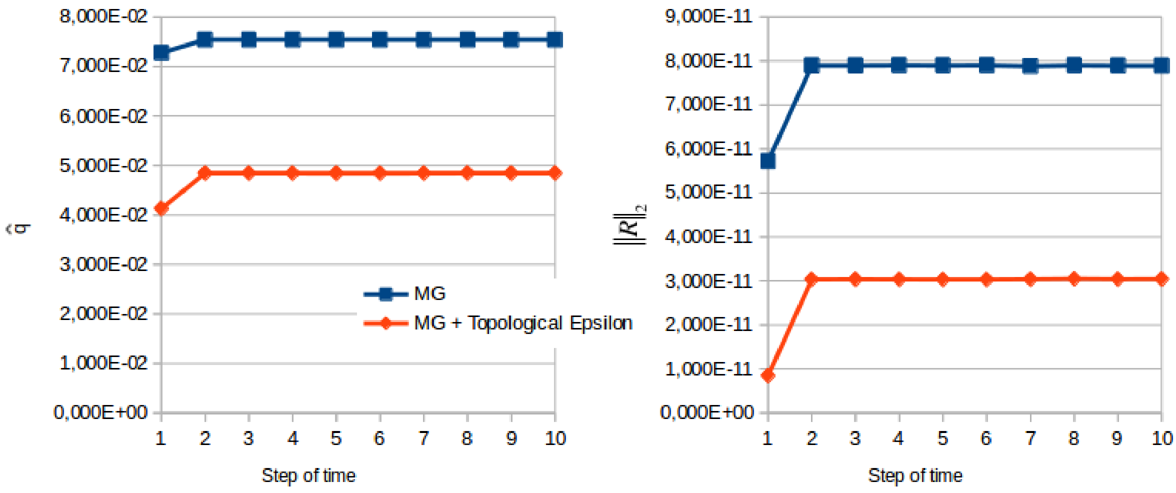


Figure 7: $\hat{q}^{(k)}$ and $\|R\|_2$ at each instant of time

It is possible to note that the Topological Epsilon extrapolator has great advantages over the pure multigrid method, overcoming in almost all the questions and losing only for the memory. It is also necessary to add the information that pure multigrid, all iterative calls reached convergence after nine cycles. However, with the extrapolator, considering the extrapolation as being also a cycle, the convergence was reached with only eight. All this allows us to conclude that the applied topological Epsilon extrapolation between cycles is very advantageous and can overcome the use of pure multigrid method.

5 CONCLUSION

With the results obtained in this work, it is observed that between seven methods of extrapolation analyzed, the Topological Epsilon is what stands out to solve the problem involving the proposed two-dimensional laminar flow of a time-dependent incompressible fluid, modeled by Navier-Stokes equations. It is also noted that it is the only one to have great advantages over the use of pure multigrid, even surpassing it in different applications, especially in the application between cycles.

In order to deal with the high memory usage of the Topological Epsilon, it is worth mentioning that an optimization in the algorithm can contribute to improve the performance, since the effort during the implementation of the extrapolator was given with another bias.

The next work should be in the sense of exploring variants of the Topological Epsilon application, optimizing memory usage and cover new applications for this extrapolator.

Acknowledgements

The first author acknowledges the financial support of Capes (Coordination for the Improvement of Higher Education Personnel). The third author acknowledges the financial support of SIMEPAR - Paraná Meteorological System. The authors also would acknowledge the infrastructure provided by the Graduate Program in Numerical Methods in Engineering (PPGMNE) of UFPR (Federal University of Paraná).

Bibliography

Anunciação, M. A. M. d. (2013). Redução do erro de iteração e aceleração do método multigrid com uso de extrapoladores.

Anunciação, M. A. M., Pinto, M. A. V., Martins, M. A. and Gonçalves, S. d. F. T. (2016). Effect of parameters of multigrid method associated with extrapolators in CFD problems.

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.

Brezinski, C. (1974). *Forme confluente de l'ε-algorithme topologique*, Vol. 23, Springer.

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.

Brezinski, C. and Zaglia, M. R. (2013). *Extrapolation methods: theory and practice*, Vol. 2, Elsevier.

Briggs, W. L., Henson, V. E. and McCormick, S. F. (2000). *A multigrid tutorial*, SIAM.

Burden, R. and Faires, J. (2008). *Análise numérica*.

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.

Delahaye, J.-P. (2012). *Sequence transformations*, Vol. 11, Springer Science & Business Media.

Fortuna, A. d. O. (2000). *Técnicas computacionais para dinâminca dos fluidos: conceitos básicos e aplicações*, Edusp.

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.

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.

Green, A. and Taylor, G. (1937). Mechanism of the production of small eddies from larger ones, **158**: 499–521.

Griffith, B. E. (2009). An accurate and efficient method for the incompressible navier–stokes equations using the projection method as a preconditioner, *Journal of Computational Physics* **228**(20): 7565–7595.

Guermond, J.-L., Minev, P. and Shen, J. (2006). An overview of projection methods for incompressible flows, *Computer methods in applied mechanics and engineering* **195**(44): 6011–6045.

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.

Guermond, J. and Shen, J. (2004). On the error estimates for the rotational pressure-correction projection methods, *Mathematics of Computation* **73**(248): 1719–1737.

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.

Kim, J. and Moin, P. (1985). *Application of a fractional-step method to incompressible Navier-Stokes equations*, Vol. 59, Elsevier.

Maliska, C. R. (1995). *Transferência de calor e mecânica dos fluidos computacional*.

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.

Mitin, A. (1985). Linear extrapolation in an iterative method for solving systems of equations, *USSR Computational Mathematics and Mathematical Physics* **25**(2): 1–6.

Neundorf, R. L. A. (2016). Aplicação do método multigrid paralelizado em GPUs no modelo WRF.

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.

Oliveira, F. d. (2010). *Efeito de malhas anisotrópicas bidimensionais sobre o desempenho do método multigrid geométrico*, PhD thesis.

Osada, N. (1991). Acceleration methods for vector sequences, *Journal of Computational and Applied Mathematics* **38**(1-3): 361–371.

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.

Tan, R. C. (1987). Computing derivatives of eigensystems by the topological ε-algorithm, *Applied numerical mathematics* **3**(6): 539–550.

Timmermans, L., Minev, P. and Van De Vosse, F. (1996). An approximate projection scheme for incompressible flow using spectral elements, *International journal for numerical methods in fluids* **22**(7): 673–688.

Trottenberg, U., Oosterlee, C. W. and Schuller, A. (2000). *Multigrid*, Academic press.

Villar, M. M. (2007). *Análise numérica detalhada de escoamentos multifásicos bidimensionais*, PhD thesis.

Wesseling, P. (1995). *Introduction To Multigrid Methods.*

Wynn, P. (1956). On a procrustean technique for the numerical transformation of slowly convergent sequences and series, **52**(4): 663–671.