SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería



Solution of the Navier-Stokes equations using projection method and preconditioned conjugated gradient with multigrid and ILU solver

Márcio Anunciação¹, Marcio Augusto Villela Pinto¹, Réverton Luis Antunes Neundorf¹

1 Universidade Federal do Paraná

Abstract

Among the main problems in the field of Computational Fluid Dynamics (CFD), there is the two-dimensional laminar flow in a transient regime of an incompressible fluid modeled by Navier-Stokes equations. Among the decoupled solutions for this equation system, that is, solutions for velocity regardless to pressure, there are the Projection methods, which separate the solution in three parts, solved in each time steps applying an iterative process regardless to the iterative process for the other variables. It may result, according to the Projection method applied, in two Reaction-Diffusion equations and one Poisson equation to be solved in each time step. This paper sought to develop an algorithm to solve the Navier-Stokes equation, applying the Finite Volume Method (FVM) with second order approximation scheme (CDS), beside a Projection method with incremental pressurecorrection scheme, so that each Reaction-Diffusion and the Poisson equation are solved efficiently. Therefore, several solvers were tested for each equation, resulting in an algorithm with the combination that achieved the best result for each equation, with the preconditioned Conjugate Gradient method (PCG) with the Multigrid method (MG) and ILU solver (Incomplete LU factorization) being the methodology used in the whole problem solving process. The geometric Multigrid with V cycle, the correction scheme (CS), the full weighting restriction, the prolongation through bilinear interpolation and the maximum number of levels for the studied cases were utilized. The results achieved were satisfactory, since the proposed methodology accelerated the iterative process considerably in relation to the classical methods available in the literature.

🔓 OPEN ACCESS

Published: 10/03/2020

Accepted: 26/12/2019

Submitted: 14/06/2019

DOI: 10.23967/j.rimni.2020.01.003

Keywords:

Navier-Stokes Multigrid Conjugated Gradient Projection methods ILU factorization

1. Introduction

In computational mathematics, determining the solution for linear or nonlinear equation systems is an important problem, since they often occur in the discretization process of mathematical models in Computational Fluid Dynamics (CFD). In order to achieve such solutions, iterative methods are widely applied. They can be categorized as linear, nonlinear, stationary and nonstationary. These approaches may result in difficulties related to the slow convergence of the iterative process applied [1]. In order to improve the convergence of the iterative methods, the Multigrid method has been proving very efficient [2-6]. Its philosophy is based on the application of several grids with several degrees of refinement, which are gone through during the iterative process. Nevertheless, the conditioning of the matrix for the linear system's coefficients may slow the iterative method's convergence. It may be accelerated by modifying this coefficient matrix in order to improve its conditioning. This process is called matrix preconditioning [7]. Among the preconditioning technics, one can mention the incomplete LU factorization (ILU), which consists in changing a matrix *A* into a product of a lower triangular matrix *L* and an upper triangular matrix *U*, attached to a matrix called residual *R*.

In the study of incompressible fluids in transient regime, the Navier-Stokes equations are well established by the conservation laws [8-13]. In order to know the state of a fluid, one has to determine the value of the variables that identify it through the time for each point in space occupied by the fluid. The variables that identify the state of an incompressible fluid in transient and isothermal regime are the velocity and pressure in each point in space.

In order to overcome the dependence between pressure and velocity in the Navier-Stokes equations, several numerical schemes have been developed, remarkably the Projection method [14], which split the solution in three parts, solved in each time step applying a local iterative process. These methods may be categorized in three classes: pressure-correction schemes, velocity-correction schemes and consistent splitting schemes.

In this paper, the problem of a two dimensional laminar flow in a transient regime of an incompressible fluid modeled by the Navier-Stokes equations with Dirichlet boundary conditions for the velocities was solved numerically. A



Projection method with incremental pressure-correction scheme was applied, in which the strategies to speed the solutions of the linear systems from these equations were mixed, such as the preconditioned Krylov Subspace method (Conjugated Gradient method) with Multigrid and ILU solver. The behaviors of some computational parameters that measure the convergence ratio of the methods and norms regarding the errors in numerical solutions were analyzed, in order to produce the algorithm with the best methodology among the ones that were studied.

2. Mathematical model and solution method

The mathematical model in two dimensional Cartesian coordinates for the laminar flow of an incompressible fluid, isothermal and in a transient regime, given by the Navier-Stokes equations [16], in this paper, is represented by the continuity equation that depicts the physical principle of the conservation of mass, and by the equations of dimensionless movement in the directions *x* and *y*:

$$\nabla \cdot u = 0, \tag{1}$$

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

$$\frac{\partial v}{\partial t} + u \cdot \nabla v = -\frac{\partial p}{\partial v} + \frac{1}{\text{Re}} \nabla^2 v, \qquad (3)$$

where *x* and *y* are the coordinates; *t* the temporal coordinate, *u* and *v* the velocity components of the vector **u** in the directions *x* and *y*, respectively; *p* the static fluid pressure and Re the Reynolds number associated to the viscous and the inertia forces. According to Hughes and Brighton, when Re << 1, the viscous forces prevail and when Re >> 1 the inertia forces prevail [17].

For this set of equations, the Taylor-Green Vortices problem [18] was solved, whose domain is given by $\{(x, y) \in \mathbb{R}^2: -\pi \le x \le \pi \text{ and } -\pi \le y \le \pi\}$, and the analytical solution by:

$$u = -(cosx seny)e^{-2(t+Ti)}$$

$$v = -(senx cosy)e^{-2(t+Ti)}$$

$$p = -\frac{1}{4}(cos 2x sen$$

$$2y)e^{-4(t+Ti)}$$

where T_i is the initial time and t > 0.

The initial and boundary conditions are obtained directly from the analytical solution.

The methods for solution of the Navier-Stokes equations can be generally categorized in coupled methods and segregated methods [13,14].

Coupled methods seek to solve the complete equation system for each computational cycle, coupling the equations of continuity and movement conservation. This is the most immediate way to solve the Navier-Stokes equations, but it renders grater difficulties in its implementation and a higher computational cost, due to the strong influence of the non-linearity convective terms [13].

Segregated methods seek the decoupling of the equations, separating the nonlinear system in simpler problems, which can be solved sequentially. Among these methods, the Projection methods [14] stand out.

Basically, the main idea of the Projection method for the Navier-Stokes Eqs. (1) to (3) is to apply the movement equation to determine a temporary velocity field.

According to Ernst and Gander [19], Eqs. (5) and (6) are Helmholtz equations

$$-(\nabla^2 + k^2)u = f, \tag{5}$$

$$-(\nabla^2 - \eta)u = f, \tag{6}$$

where *k* and η are constants that depend on the studied problem and *f* the source term.

(4)



Equation (6) is common in CFD and describes stationary Reaction-Diffusion phenomena, being the model to be solved in order to obtain the provisional velocity field. Despite the similarities, Eq. (6) is a much greater challenge in the classic iterative methods application [19].

After having obtained this temporary field, for the two dimensional case, it can be shown that a system of three equations and three unknowns (u, v and p) can be obtained, where the pressure appears in only one of the equations. The role of pressure in incompressible flows is to make the velocity field satisfy the continuity equation. Therefore, it is necessary to calculate pressure in the following step. For this purpose, an elliptical Poisson equation is solved for the continuity equation to be satisfied and for the pressure to be determined [14,15].

The Poisson equation is a elliptic partial differential equation. with wide usage in incompressible fluid flows, described by [20]:

$$\nabla^2 u = f. \tag{7}$$

The importance of this equation for pressure is that it links the movement conservation and the continuity equations. Thus, the final velocity field and pressure are calculated.

3. Numerical models

3.1 Utilized grid

In this paper the Finite Volume method (FVM) with staggered grids was applied, as described in [21], also applied by [12], with the velocities placed on the faces and the pressure on the center of the volumes. Applying this strategy avoids the numerical instabilities on the pressure solution [6,22,23]. Such grid and its ordering are represented in Figure 1 (4 to 4 volumes).



In this grid, the volumes with dashed lines are the volumes with ghost volumes (which do not belong to the physical domain of the problem), and it may be noticed that in applying this kind of grid, the boundary conditions for the variable *u* are automatically prescribed in the east and west boundaries, while in the other boundaries some kind of procedure is necessary in order for the ghost volumes to consider the contour information. Here, linear extrapolation was applied, and a similar case for the *v* variable in the south and north contours.



3.2 Spatial discretization by finite volumes

Equations (1) a (3) can be integrated in each control volume described in Figure 2.



Thus, the mass conservation equations, given by Eq. (1), of linear momentum in the direction *x*, Eq. (2), and linear momentum in the direction *y*, Eq. (3), can be written as:

$$\oint_{S} u . n dS,$$
(8)

$$\frac{\partial}{\partial t} \int_{V} u = -\oint_{S} uu \cdot n dS - \oint_{S} p n_{x} dS + \frac{1}{\text{Re}} \oint_{S} \nabla u \cdot n dS$$
⁽⁹⁾

and

$$\frac{\partial}{\partial t} \int_{V} v = -\oint_{S} vu \cdot n dS - \oint_{S} pn_{y} dS + \frac{1}{\text{Re}} \oint_{S} \nabla v \cdot n dS,$$
⁽¹⁰⁾

respectively.

The terms of Eqs. (8), (9) and (10) were discretized applying the FVM with staggered grids, where the integrals are calculated over the control volumes defined in Figure 2, where for a matter of simplicity, the same spatial refinement was applied in all directions, that is, $h_x = h_y = h$ (uniform grids).

In order to discretized the mass conservation, a volume centered in *p* was applied, Figure 2 (b), where there is:

$$\oint_{S} u \cdot n \, dS \approx u_{i+\frac{1}{2},j}^{n+1} - u_{i-\frac{1}{2},j}^{n+1} + v_{i,j+\frac{1}{2}}^{n+1} - v_{i,j-\frac{1}{2}}^{n+1} = 0.$$
⁽¹¹⁾

The pressure terms were discretized applying the volumes centered in the velocity components defined in Figures 2(a) and (c), generating:

$$\oint_{S} p n_x dS \approx (p_{i+1,j} - p_{i,j})h$$
⁽¹²⁾

and



$$\oint_{S} p n_y dS \approx (p_{i,j+1} - p_{i,j})h.$$
⁽¹³⁾

The same control volumes defined in Figures 2(a) and (c) were applied in the discretization of the diffusive terms, where one obtains:

$$\nabla u \cdot n dS \approx u_{i+\frac{3}{2}j}^{n} + u_{i-\frac{1}{2}j}^{n} + u_{i+\frac{1}{2}j+1}^{n} + u_{i+\frac{1}{2}j-1}^{n} - 4u_{i+\frac{1}{2}j}^{n}$$

and

$$\oint_{S} \nabla v \cdot n \, dS \approx v_{i,j+\frac{3}{2}}^{n} + v_{i,j-\frac{1}{2}}^{n} + v_{i+1,j+\frac{1}{2}}^{n} + v_{i-1,j+\frac{1}{2}}^{n} - 4v_{i,j+\frac{1}{2}}^{n}.$$
(15)

Furthermore, applying control volumes centered in the velocity components, the advective terms were discretized, resulting in:

$$\oint_{S} uu . ndS \approx \left[(u^{2})_{i+1,j}^{n} + (u^{2})_{i,j}^{n} + (uv)_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - (uv)_{i+\frac{1}{2},j-\frac{1}{2}}^{n} \right] h$$
⁽¹⁶⁾

and

$$\oint_{S} vu . ndS \approx \left[(v^{2})_{i,j+1}^{n} + (v^{2})_{i,j}^{n} + (uv)_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - (uv)_{i-\frac{1}{2},j+\frac{1}{2}}^{n} \right] h.$$
⁽¹⁷⁾

The velocities that appear in Eqs. (16) and (17) can be approximated by:

$$(u^2)_{i,j}^n = \left[\frac{1}{2} \left(u_{i+\frac{1}{2},j}^n + u_{i-\frac{1}{2},j}^n\right)\right]^2,$$
(18)

$$(uv)_{i+\frac{1}{2}j+\frac{1}{2}}^{n} = \left[\frac{1}{2}\left(u_{i+\frac{1}{2}j}^{n} + u_{i+\frac{1}{2}j+1}^{n}\right)\right] \left[\frac{1}{2}\left(v_{i,j+\frac{1}{2}}^{n} + v_{i+1,j+\frac{1}{2}}^{n}\right)\right]$$
(19)

and

$$(v^{2})_{i,j}^{n} = \left[\frac{1}{2}\left(v_{i,j+\frac{1}{2}}^{n} + v_{i,j-\frac{1}{2}}^{n}\right)\right]^{2}.$$
(20)

3.3 Projection methods

The Projection methods were introduced by [24,25], and they are characterized by the Navier-Stokes solution in two steps. Initially, an auxiliary velocity \mathbf{u}_t is calculated ignoring the incompressibility condition given by Eq. (1) and the pressure term (considering p = 0 in Eqs. (2) and (3)). In this step, a Reaction-Diffusion equation is solved by temporal discretization. In the second step, pressure is calculated by solving a Poisson equation. The most attractive property in the Projection methods is the fact that in each step a sequence of decoupled elliptical equations are solved for pressure and velocities, enabling large scale numerical simulations to be executed efficiently; however, it is not trivial to develop and analyze high order Projection methods [14].

There are several formulations for the Projection methods, differing in the way to calculate the auxiliary velocity \mathbf{u}_t and the projection step. Each of these comes from the method of [24,25], which is a Projection method with a simpler correction scheme for the pressure, which is not yet a first order method (regarding time) for pressure on norm l_2 [14] due to the absence of the pressure gradient on the first step of the method. Goda observed that by adding the pressure gradient [26], Chorin's algorithm presents an improvement in accuracy, an idea applied by Van Kan to develop a incremental pressure-correction scheme of second order [27], which was improved by Timmemans et al. through a modification on the boundary conditions for pressure from the method mentioned previously [28]. This algorithm is known as "incremental correction for pressure on rotational form" and it will be adopted in this paper for

(1 1)



the solution of the Navier-Stokes equations [29]. Such algorithm will be given below and it will be designated as Algorithm 1.

Furthermore, Guermond and Shen show that the incremental version on rotational form is a second order scheme for velocity on l_2 -, l_1 - and l_{∞} -norms [29]. For pressure the scheme is a second order scheme for norms l_2 - and l_1 -norms and 3/2 order for l_{∞} -norms. The method consists on the following steps:

First step:

$$\frac{3u^{t} - 4u^{n} + u^{n-1}}{2vh_{t}} = \frac{\beta g}{v} + \nabla^{2}u^{t} + \frac{\nabla P^{n}}{v}u^{t}|_{\partial\Omega} = b^{n}$$
(21)

where 'ut is the auxiliary velocities field, un is the velocities field on the time step n, ht is the temporal refinement, Pn is pressure on the time step n and bn the boundary conditions on time step n, $\beta g = u \cdot \nabla u$ with constant β and $\nu = 1/\text{Re}$.

Rearranging the terms on Eq. (21), there

$$-\nabla^{2}u^{t} + \frac{3\text{Re}}{2h_{t}}u^{t} = \text{Re}\left(\beta g + \nabla P^{n} + \frac{4u^{n} - u^{n-1}}{2h_{t}}\right)$$

$$u^{t}|_{\partial\Omega} = b^{n}, \qquad (22)$$

which has the form of a Reaction-Diffusion equation (Eq. (6)) for the variable \mathbf{u}_t , with:

$$\eta = -\frac{3\text{Re}}{2h_t} \tag{23}$$

and source term

$$f = \operatorname{Re}(\beta g + \nabla P^{n} + \frac{4u^{n} - u^{n-1}}{2h_{t}}).$$
⁽²⁴⁾

Second step:

$$\nabla^{2} \phi^{n+1} = \frac{3}{2h_{t}} \nabla \cdot u^{t} u^{n+1} = u^{t} - \frac{2h_{t}}{3} \nabla \phi^{n+1}$$

$$P^{n+1} = P^{n} + \phi^{n+1} - v \nabla \cdot$$

$$u^{t},$$
(25)

where ϕ^{n+1} is the pressure correction.

Algorithm 1. Projection method

Start

Estimate initial conditions for u and v: $u^{-1}=u^0$.

Estimate initial conditions for $P: P^0$.

Establish the last step for simulation time: N_t .

For *n* = 0, 1, 2, ..., *N*_t - 1, do:

Step 1: given the velocities u^n , u^{n-1} and the pressure P^n , calculate \mathbf{u}_t , solving the Reaction-Diffusion equation:

$$-\nabla^2 u^t + \frac{3\text{Re}}{2h_t} u^t = \text{Re}\left(\beta g + \nabla P^n + \frac{4u^n - u^{n-1}}{2h_t}\right)$$

Step 2: Given the velocity \mathbf{u}_t , calculate ϕ^{n+1} , solving the Poisson equation:

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

Update u^{n+1} and P^{n+1} by:



M. Anunciação, M. Augusto Villela Pinto and R. Neundorf, Solution of the Navier-Stokes equations using projection method and preconditioned conjugated gradient with multigrid and ILU solver, Rev. int. métodos numér. cálc. diseño ing. (2020). Vol. 36, (1), 17

$$u^{n+1}=u^{t}-\frac{2h_{t}}{3}\nabla\phi^{n+1}$$

$$p^{n+1}=p^{n}+\phi^{n+1}-v\nabla\cdot u^{t}$$
End

If the time step h_t is too large in relation to spatial refinement (h_x and h_y), the simulations can become unstable [30]. The criterion applied in this paper to avoid instabilities is given by [12,30]:

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

Since this paper considers $h_x = h_y = h$, the criterion can be rewritten as

$$h_t < \frac{h^2}{4\text{Re}}.$$
(27)

In [30] other criteria are presented, which support the numerical stability. A common characteristic of the Projection methods is that they impose Neumann boundary conditions for the pressure on the contours [14]. Due to the boundary conditions, the following integrability condition has to be satisfied in order to assure the existence and uniqueness of the solution [5]:

$$\int_{\Omega} \nabla^2 \phi^{n+1} = \int_{\Omega} \frac{1}{h_t} \nabla \cdot u^{n+1} = 0.$$
⁽²⁸⁾

For details on how Eq. (28) is satisfied for each time step, see the procedure proposed by [31].

4. Method for linear systems solution

The discretization for the partial differential equations to be solved results on the following algebraic equations system:

$$Ab = f, \tag{29}$$

where *b* and $f \in \mathbb{R}^n$ where *b* can be *u*, *v* or ϕ , according to steps 1 and 2 on Algorithm 1.

4.1 Gauss-Seidel method (GS)

Matrix *A* being divided in the form:

$$A = P - N, \tag{30}$$

with non-singular *P*, and *P* is called preconditioning matrix or preconditioner.

To solve Eq. (29), there is the following method, which is called basic iterative method:

$$Pb^{(k+1)} = Nb^{(k)} + f, (31)$$

where the superscript of *b* is the representation of the iterations.

Considering the method on the following mode:

$$b^{(k+1)} = Bb^{(k)} + P^{-1}f, (32)$$

where $B = P^{-1}N$, N = P - A, and B is called the iteration matrix of the method given by Eq. (32).

If the diagonal entries of *A* are non-zeros, the unknown corresponding value can be isolated in each equation, resulting in the Gauss-Seidel method, given an initial estimate $b^{(0)}$:



$$b_i^{(k+1)} = \frac{1}{a_{ii}} \left[f_i - \sum_{j=1}^n a_{ij} b_j^{(k+1)} - \sum_{j=1+1}^n a_{ij} b_j^{(k)} \right], i = 0, \dots, n,$$
(33)

4.2 Preconditioned Conjugated Gradient (PCG)

Descent methods are based on obtaining a solution *b* for a system of the type given by Eq. (29) by the minimization property of an appropriate norm for the error. From an initial estimate $b^{(0)}$, these methods determine a solution for the problem by diminishing the error.

The vectorial norm being given by $\|b\|_2 = \langle b, Sb \rangle$, where $S \in \mathbb{R}^{n \times n}$ is a positive defined symmetrical matrix. Considering the residual equation $Ae^{(k)} = r^{(k)}$, it can be deduced that:

$$\| e^{(k)} \|_{2} = \langle e^{(k)}, Se^{(k)} \rangle = \langle r^{(k)}, Rr^{(k)} \rangle,$$
(34)

where $R = A^{-T}SA^{-1}$, which means that minimizing the error equals minimizing the residue in the same norm [7].

On the Descent method one starts from an initial estimate $b^{(0)}$ for the solution and a succession of vectors $b^1, \dots, b^{(k)}$, \dots is generated, moving successively towards diminishing the error, making this succession converge to b.

One of the ways to choose the best direction in each point $b^{(k)}$ would be to choose the direction of $p^{(k)}$ which is lined up with the direction of the function gradient, but in the opposite orientation (Gradients method).

Thus, the iterative expression of the Gradients methods is given by [7]:

$$b^{(k+1)} = G_k b^{(k)} + c_k$$
(35)

where $G_k = I - \alpha_k S$ and $c_k = \alpha_k S A^{-1} f$, α_k being a constant.

Conjugate Gradient method (CG) is an iterative descent method applied in cases where the iteration matrix A is a symmetric positive definite matrix. This method demands a more careful choice of the descent directions $p^{(k)}$.

In order to exemplify it, consider a two-dimensional problem. The lines $|| e^{(k)} ||_2 = || r^{(k)} ||_2 = \text{constante, are concentric ellipses with the center in$ *b*, as illustrated in Figure 3.





Supposing there is an initial estimate $b^{(0)}$, according to the Gradient method, one can start from this point and research the minimum of $||e^{(k)}||_2$, across the direction $r^{(0)}$, obtaining $b^{(1)}$. Observing Figure 3, it can be verified that the best progression of $b^{(1)}$ would not be the more inclined direction, but the direction $p^{(1)}$ which points to the center of the ellipse.

In fact, the new minimizing point $b^{(2)}$ would coincide with the exact solution of *b*, and thus the iterative process would end with just two iterations. Therefore a direction $p^{(1)}$ can be chosen as:

$$p^{(1)} = b - b^{(1)}. (36)$$

As it can be observed, complying with the orthogonality of the successive residues verified previously, there is

$$0 = \langle r^{(1)}, r^{(0)} \rangle = \langle r^{(1)}, p^{(0)} \rangle = \langle f - Ab^{(1)}, p^{(0)} \rangle = \langle A(b - b^{(1)}), p^{(0)} \rangle,$$
(37)

in such a way that

$$\langle p^{(1)}, A p^{(0)} \rangle = 0.$$
 (38)

The sought directions are

$$p^{(k+1)} = r^{(k+1)} + \omega_k p_k, \tag{39}$$

where ω is given by

$$\omega_k = \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle}.$$
(40)

The iterations of the CG method satisfy the following error estimate [7]:

$$\| e^{(k)} \| \leq \left(\frac{(cond_2A)^{\frac{1}{2}} - 1}{(cond_2A)^{\frac{1}{2}} + 1} \right) \| e^{(0)} \|,$$
⁽⁴¹⁾

where $cond_2A = ||A||_2 \cdot ||A^{-1}||_2$.

Thus, it can be noticed that the CG method, theoretically, calculates the exact solution in *n* iterations, which is not verified in practice due to the application of this method in finite precision arithmetic's and the introduction of rounding errors which destroy the exact conjugacy of the direction $p^{(k)}$ and the exact orthogonality of the residues $r^{(k)}$. Thus, this method takes a typical iterative form.

One way to make this iterative method faster consists in transforming the system Ab = f into an equivalent system, where the matrix has a more favorable conditioning number. According to Eq. (41), if it is possible to reduce $cond_2A$, consequently a fast convergence is achieved.

The transformation of the pre- and post-multiplication of the matrix *A* by the matrixes *P* and *Q* being invertible by the following manner:

$$(PAQ)(Q)^{-1}b = Pf.$$
 (42)

The system now corresponds, therefore, to $\overline{A} \ \overline{b} = \overline{f}$, with $\overline{A} = PAQ$, $\overline{b} = Q^{-1}b$ and $\overline{f} = Pf$, the matrixes *P* and *Q* having to be chosen in the way that provides the best convergence properties to the matrix in relation to the original matrix *A*. This procedure is known as preconditioning of the original equations system, *P* and *Q* being known by the name of preconditioning matrixes.

Considering that in the CG method the transformed matrix has to keep being a symmetric positive definite matrix, the preconditioning will also have to preserve this property. In fact, it can be obtained making P = S and $Q = P^T = S^T$, and

https://www.scipedia.com/public/Anunciacao_et_al_2019a



in this manner, it can be concluded that $\overline{A} \overline{b} = \overline{f}$, $\overline{A} = SAS^T$, $\overline{b} = S^{-T}b$ and $\overline{f} = Sf$.

Thus, it can be verified that $\bar{p}^{(0)} = \bar{r}^{(0)} = \bar{f} - \bar{A} \bar{b}^{(0)} = Sr^{(0)}$. On the other hand, choosing $p^{(0)} = S^T \bar{p}^{(0)} = S^T \bar{r}^{(0)} = S^T Sr^{(0)}$, there is $\langle \bar{p}^{(0)}, \bar{A} \bar{p}^{(0)} \rangle = \langle p^{(0)}, A p^{(0)} \rangle$, the relation $\langle \bar{r}^{(0)}, \bar{r}^{(0)} \rangle = \langle \bar{r}^{(0)}, r^{(0)} \rangle$ also being valid, with

$$W\bar{r}^{(0)} = r^{(0)} \tag{43}$$

and

$$W = (S^T S)^{-1}.$$
 (44)

From that, there is:

$$\alpha_0 = \langle \bar{r}^{(0)}, r^{(0)} \rangle / \langle p^{(0)}, A p^{(0)} \rangle, \tag{45}$$

$$b^{(1)} = b^{(0)} + \alpha_0 p^{(0)}, \tag{46}$$

$$r^{(1)} = r^{(0)} - \alpha_0 A p^{(0)}, \tag{47}$$

$$\beta_0 = \langle \bar{r}^{(1)}, r^{(1)} \rangle / \langle \bar{r}^{(0)}, r^{(0)} \rangle \tag{48}$$

and

$$p^{(1)} = r^{(1)} + \beta_0 p^{(0)}, \tag{49}$$

keeping these expressions for the following iterations.

Algorithm 2 presents the preconditioned CG method (PCG), where it can be noticed that at each iteration it is necessary to solve the auxiliary system:

$$W\bar{r}^{(j+1)} = r^{(j)}$$
. (50)

and it is desirable that its resolution is not very laborious.

Algorithm 2. Preconditioned Conjugated Gradient method (PCG)

Start
Read A, f
Estimate
$$b^{(0)}$$

Determine a tolerance γ
Calculate $r^{(0)}=f-Ab^{(0)}$
Solve $W\tilde{r}^{(0)}=r^{(0)}$
Attribute $p^{(0)}=\tilde{r}^{(0)}$
For $j = 0, 1, ...$ until convergence
 $a_j = (\tilde{r}^{(j)}, r^{(j)})/(Ap^{(j)}, p^{(j)})$ $b^{(j+1)}=b^{(j)}+a_j p^{(j)}$
 $r^{(j+1)}=r^{(j)}-a_j Ap^{(j)}$
Solve $W\tilde{r}^{(j+1)}=r^{(j)}$
 $\beta_j = (\tilde{r}^{(j+1)}, r^{(j+1)})/(\tilde{r}^{(j)}, r^{(j)})$ $p^{(j+1)}=\tilde{r}^{(j+1)}+\beta_j p^{(j)}$
End
End

4.3 Incomplete LU decomposition (ILU)

The incomplete LU decomposition (ILU) consists in decomposing a matrix A in an incomplete manner, as the name suggests. Such decomposition is in the A = LU - R form, where L and U are, respectively, inferior and superior triangular



matrixes *A* and *R* represents the residue or decomposition error. On the matrix *R*, conditions are imposed such as having null terms in determined positions in order to maintain certain characteristics of the original matrix *A*.

A general algorithm for ILU can be obtained applying the Gaussian Elimination [7] and having some elements in predetermined positions. For this, a set of points is determined $P \subset \{(i,j) | i \neq j, \text{ com } 1 \le i, j \le n\}$ where the elements of the matrix will be null. In this case, there can be several variations of the ILU decomposition, for example: ILU(0), where the structure of the LU decomposition is identical to the structure of matrix A, that is, it does not allow extra fillings; ILU(1), where the filling of one additional diagonal is allowed in each LU factor, etc. In this paper, ILU(1) was chosen, which will be called 7-points ILU (since the stencil of the original problem has 5 points).

Thus, given the incomplete LU decomposition, it is solved in an iterative process composed of the following steps:

(1) $Lz^{(k)} = r^{(k)}$ to obtain $z^{(k)}$, $r^{(k)}$ being the residue vector

(2) $U\delta^{(k)} = z^{(k)}$ to obtain $\delta^{(k)}$.

Thus, the solution in the iteration k + 1 is such that:

$$b^{(k+1)} = b^{(k)} + \delta^{(k)} . (51)$$

4.4 Multigrid method (MG)

The MG method [5,6,32] makes use of the error smoothing characteristics from the classic iterative methods: it occurs fast (in the initial iterations) for oscillatory components, while for smooth components, for a high number of iterations, such methods lose their efficiency. Thus, the MG method works with a system of auxiliary coarser grids (with a lower number of points) in which the error components are quickly smoothed, in order to return to the original grid. The information is transferred between grids through operators, called restriction operators (information from a fine to a following coarser grid), generically represented by $[I]_h^H$ and defined by

$$v^H = [I]_h^H v^h, \tag{52}$$

or prolongation ones (information from the coarse grid to a finer one), represented generically by $[I]_{H}^{h}$ and defined by

$$v^h = [I]^h_H v^H. ag{53}$$

The restriction operator applied in this paper was the complete weighting operator and the prolongation operator was the bilinear interpolation [5,6,32,33].

In this paper, moreover, the V-Cycle was applied, for its computational efficiency. The equations of Poisson and Reaction-Diffusion being linear, the correction scheme (CS) was applied, which transfers only the residues to the coarser grids [5,6]. The coarsening ratio, considering uniform grids, is defined as r = H/h, where h represents the size of the spacing of the fine grid, Ω^h , (also called the dimension of the fine grid elements), and H the size at the element of the next coarser grid, Ω^H .

Figure 4 illustrates the cycle-V with CS for five grids. The superscripts h, 2h, 4h, ... indicate the grid where the vectors and matrixes are defined, the smoothing numbers (S) being achieved in the restriction process (R) (pre-smoothing) and prolongation (P) (post-smoothing) being represented, respectively, by v_1 and v_2 .





In this paper, the MG with GS and ILU solvers was applied. Such iterative methods were also applied as preconditioners for the PCG method, that is, they were used to solve the linear system given by Eq. (50) at each iteration of the CG method.

5. Results

5.1 Methodology and computational details

Computational tests were done to calculate the values of p, u and v in the discretized model problem, Eqs. (22) and (25), applying the following methodology:

- 1) Multigrid with Gauss-Seidel solver (MG-GS).
- 2) Multigrid with ILU solver (MG-ILU).
- 3) Preconditioned Conjugated Gradient with Multigrid and Gauss-Seidel solver (PCG-MG-GS).

4) Preconditioned Conjugated Gradient with Multigrid and ILU solver (PCG-MG-ILU).

In all simulations and for all the variables of interest, the stop criterion applied was the nondimensionalized residue I_2 -norm based on the initial estimate, given by:

$$\frac{r^{(k)}}{r^{(0)}}\|_{2}^{2} < \epsilon ,$$
(54)

where $r^{(k)}$ is the residue on the current iteration, $r^{(0)}$ is the residue of the initial estimate, and ε is the tolerance.

In this paper the null vector was utilized as the initial estimate for all the variables of interest and with the use of double precision. The standard coarsening ratio was utilized, that is , r = 2 [5], where the Multigrid method started from the finest grid and got to the coarsest grid possible, that is, the grid with N = 2x2 volumes. The number of unknowns in the finest grids was: $N = 16 \times 16$, 32×32 , 64×64 , 128×128 , 256×256 , 512×512 and 1024×1024 . The number of pre- and post-smoothing was equals to $v = v_1 = v_2 = 3$.

The computer utilized for the simulations has an Intel Core i7-5500 processor, a 2.40 Ghz CPU, 16.0 Gb memory and a 64 bits Windows 10 operational system.

Four parameters were analyzed in order to compare the four methodologies:



- l_{∞} -norm for numerical error (||E|| $_{\infty}$);
- l_2 -norm for nondimensionalized residue based on the initial estimate (Eq. (54));
- Empirical convergence factor, defined as [6]:

$$q^{(k)} = \frac{\|r^{(k)}\|_2}{\|r^{(k-1)}\|_2}$$
(55)

and

CPU time.

5.2 Analysis of the pressure

This phase it is intended to determine the best method to solve the Poisson equation, which models pressure, determining a method to solve the velocities equations. In this case, MG-GS, MG-ILU, PCG-MG-GS and PCG-MG-ILU were utilized for pressure and MG-GS for the velocities.

The behavior of the l_{∞} -norm was initially evaluated for the four studied methods, where it could be noticed that all of them presented a reduction in the numerical error norm with the increase in the size of the problem; a desirable characteristic in a convergent numerical method (Figure 5).



Figure 6 presents the number of iterations necessary in order to reach the stop criterion for the problem with $N = 1024 \times 1024$. Based on this figure, it can be noticed that PCG-MG-ILU reached the stop criterion in a lower number of iterations.





Figure 7 illustrates the behavior of the empirical convergence factor with the increase of the size of the problem. Based on this figure, it can be claimed again that the PCG-MG-ILU methodology presented the best results, since it is the one that generated the best values for the convergence factors, that is, values near to zero [5,34].





Figure 8 illustrates the behavior of the CPU time (in seconds) of the simulations with the increase of the size of the problem. It can be noticed that the computational times became very close, with a slight advantage for the PCG-MG-ILU methodology, which can be confirmed in sequence.





The speed-up is a measure utilized to determine the increase in velocity obtained during the execution of a program utilizing an algorithm "A" in relation to its execution utilizing an algorithm "B" [6]. It is given by:

$$S_P = \frac{t_{CPU}(\text{algoritmo A})}{t_{CPU}(\text{algoritmo B})} .$$
(56)

Table 1 shows the Speed-up of the three methods MG-GS, MG-ILU and PCG-MG-GS in relation to the PCG-MG-ILU method.

 Table 1. Speed-up values for MG-GS, MG-ILU and PCG-MG-GS in relation to PCG-MG-ILU

Ν	MG-GS	MG-ILU	PCG-MG-GS
16×16	1	1	1
32×32	0.833	1	1
64×64	0.948	1.136	1.109
128×128	1.036	1.156	1.109
256×256	1.033	1.153	1.110
512×512	1.055	1.139	1.160
1024×1024	1.074	1.194	1.167

According to the data presented in this table, it can be confirmed that the CPU times are close among themselves, but there is a slight advantage with the use of PCG-MG-ILU (Speed-up above 1), as described in [35].

It all leads us to conclude that the CGP-MG-ILU method has the potential to accelerate the convergence of the presented problem.



5.3 Analysis of the velocities

In this phase, it is intended to determine the best method to solve the Reaction-Diffusion equations, which model the velocities u and v, determining the best method for pressure presented in the previous section, which was PCG-MG-ILU. Since the results for the velocities u and v for all the parameters are similar, it was chose to display only the results for u.

Figure 9 shows the behavior of the numerical error *versus* the size of the problem, for the several solution methods. It can be noticed that the four studied methods presented an expected characteristic for the convergent iterative methods, that is, the decrease in the numerical error with the increase in the size of the problem. It can also be noticed that the presented error have values which are very close among themselves.



Figure 10 displays the behavior of $_2$ -norm of the nondimensionalized residue by l_2 -norm of the initial estimate in function of the number of iterations for the problem with $N = 1024 \times 1024$ volumes. Based on this figure, it can be noticed that the PCG-MG-ILU and the MG-ILU methods were the ones which achieved the stop criterion in the lowest number of iterations.





Figure 11 illustrates the behavior of the empirical convergence factor in function of the size of the problem. Based on this figure, it can be concluded that the PCG-MG-ILU method presented the best results, since it generated the best values for the convergence factors (closest values to zero), besides decreasing its value with the increase of the size of the problem.





Figure 12 shows the CPU times (in seconds) of the simulations with the increase of the size of the problem. Based on this figure, it can be noticed that the computational times were very close, with a slight advantage for the PCG-MG-ILU methodology, which can be confirmed in the results presented in the sequence.





Table 2 presents the Speed-up (Eq.(56)) of the MG-GS, MG-ILU and PCG-MG-GS methods in relation to the PCG-MG-ILU method. In this table, it can be confirmed that the CPU times are close among themselves, but there is an advantage with the use of PCG-MG-ILU, that is, the method that most accelerated the convergence of the iterative process for the presented problem.

Table 2. Values of Speed-up for MG-GS, MG-ILU and PCG-MG-GS in relation to PCG-MG-ILU

Ν	MG-GS	MG-ILU	PCG-MG-GS
16×16	1	1	1
32×32	1.5	1	1.5
64×64	1.428	1.214	1.571
128×128	1.436	1.218	1.490
256×256	1.425	1.242	1.495
512×512	1.443	1.230	1.524
1024×1024	1.337	1.228	1.535

5.4 Algorithm

After the tests of the two previous phases (sections 5.2 and 5.3), whose objective was to identify the best methods for the solution of the equations for the variables p (Poisson equation) and u and v (Reaction-Diffusion equations), we get to the point where the best methods were applied simultaneously to solve the Navier-Stokes equations.

The results presented below are compared the classic Gauss-Seidel (GS) method in its singlegrid version (only grid method) so that the gains can be measured when using the proposed algorithm (adaptation of the Projection method presented in Algorithm 1), that is, using the PCG-MG-ILU solver in the solution of the linear systems given in steps 1 and 2 of Algorithm 1.

Figure 13 presents the behavior of l_2 -norm of the nondimensionalized residue by the initial estimate in function of the



number of iterations for the problem with $N = 128 \times 128$ volumes for the variables u, v and p. Based on this figure, it can be noticed that the PCG-MG-ILU method reached the stop criterion $\varepsilon = 10^{-10}$ in an extremely lower number of iterations, both for p and for u and v.



Figure 14 illustrates the behavior of the empirical convergence factor with the increase of the size of the problem. Based on this figure, it can be concluded that the PCG-MG-ILU method presented much better results for the convergence factors for the three variables of interest, comparing with the classic GS method.





Figure 15 contains the values of the CPU times (in seconds) of the simulations with the increase of the size of the problem. Based on this figure, it can be noticed that the PCG-MG-ILU methodology was the one that converged faster, being quite evident the difference in CPU time between the proposed methodology and the classic GS method.





Table 3 presents the Speed-up (Eq.(56)) on the GS method in relation to the PCG-MG-ILU method. According to the data on this table, the methodology that uses the PCG-MG-ILU method proved much more efficient than the methodology that utilizes the GS method, reaching a 105 times higher velocity for the biggest problem compared in this table. It can also be noticed that the Speed-up increases as the size of the problem increases, which is a highly desirable property.

Table 3. Values of GS Speed-up in relation to PCG-MG-ILU

Ν	Sp
8×8	3.013
16×16	6.504
32×32	11.785
64×64	31.181
128×128	105.242

Another complementary study can be done through the comparative of complexities of algorithms utilizing the different methods. For complexity analysis of such algorithms, the exponent *p* was calculated; obtaining the by least square fitting, for the function given by

$$t_{CPU}(N) = c N^p , \qquad (57)$$

where *p* represents the order of the solver associated with the employed method and *c* is a coefficient that depends on each method and each solver, *N* is the number of unknowns of the system and t_{CPU} , the CPU time.

For the ideal MG method, p = 1, meaning that the computational effort increases linearly with the size of the grid [3,6]. Thus, for a given hardware and compilator, the closer p is to the unity, the more efficient the algorithm is and the lower the value of c, the faster it is.

Table 4 presents the values of *p* and *c* for the curve adjustments obtained for the studied methods (data from Figure



15). In this table, it can be observed that the value of $p \approx 1$ for PCG-MG-ILU, being $p \approx 2$ for GS, as expected from literature [6]. Besides that, the lower value for *c* appears for PCG-MG-ILU.

Table 4. Values of *p* and *c* for the algorithms of the studied methodologies

Metodology	С	р
GS	5.332E-04	1.855
PCG-MG-ILU	4.272E-05	0.984

6. Conclusion

In this paper, a problem of a two dimensional laminar flow of an incompressible, isothermal fluid in a transient regime given by the Navier-Stokes equations was solved utilizing a Projection method with the intention of obtaining the primary variables *u*, *v*, and *p*. The Finite Volume Method in staggered grids in the discretization process was used, generating systems of linear equations, which were solved by applying the Geometric Multigrid method and some combinations with different solvers. Numerical tests were performed for the Taylor-Green vortices problems, which have a known analytical solution. Based on the obtained results, it was verified that the proposed algorithm, the algorithm that contemplates the PCG-MG-ILU method on the solution systems arising from pressure and velocities, presented results better than the methodologies compared in this paper, since it converges with few iterations, has the best empirical convergence factor, Speed-up and complexity factors values. Therefore, the computational time was reduced, for example, in up to 105 times for the problem with 128 × 128 in relation to a classic method, and an improvement tendency of this factor (Speed-up) was noticed with the increase of the size of the problem, a quite desirable property.

Acknowledgements

The authors would like to thank the Graduate Program in Numerical Methods in Engineering (PPGMNE) of the Federal University of Paraná (Brazil). The third author thanks the scholarship of Meteorological System of Paraná (SIMEPAR). This study was financied partly by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) -Finance Code 001.

References

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

[2] Fedorenko R.P. On the speed of convergence of an iteration process. U.S.S.R. Comput. Math. And Math. Phys., 4(3):227-235, 1964.

[3] Brandt A. Multi-level adaptive solutions to boundary-value problems. Mathematics of Computation, 31:333-390, 1977.

[4] Pletcher R., Tannehill J., Anderson D. Computational fluid mechanics and heat transfer. Series in Computational and Physical processes in Mechanics and Thermal Sciences, 2nd ed., Taylor & Francis, 1997.

[5] Briggs W.L., Henson V.E., McCormick S.F. A multigrid tutorial. 2nd ed., SIAM, 2000.

[6] Trottenberg U., Oosterlee C., Schüller A. Multigrid. Academic Press, 2001.

[7] Saad Y. Iterative methods for sparse linear systems. SIAM, 2003.

[8] Batchelor G. An introduction to fluid dynamics. Cambridge University Press, 1970.

[9] Peyret R., Taylor T.D. Computational methods for fluids flow. Springer-Verlag, 1983.

[10] Panton R.L. Incompressible flow. John Wiley & Sons, 1984.

[11] Fletcher C.A.J. Computational methods for fluid dynamics. Vol. 1, 2nd ed., Springer, 1992.

[12] Maliska C.R. Transferência de calor e mecânica dos fluidos computacional. 2nd ed., LTC, 2004.

[13] Ferziger J.H., Peric M. Computational methods for fluid dynamics. Springer-Verlag, 2002.

[14] Guermond J.L., Minev P., Shen J. An overview of projection methods for incompressible flows. Computer Methods in Applied Mechanics and Engineering, 195(44):6011-6045, 2006.

[15] Griffith B.E. 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, 2009.

[16] Kim J., Moin P. Application of a fractional-step method to incompressible Navier Stokes equations. Journal of Computational Physics, 59:308-323, 1985.

[17] Hughes W.F., Brighton J.A. Dinâmica dos fluidos. McGraw Hill, 1967.

[18] Pearson C.E. A computational method for time dependent two dimensional incompressible viscous flow problems. Sperry-Rand Research Center, 1964.

[19] Ernst O.G., Gander M.J. Why it is difficult to solve Helmholtz problems with classical iterative methods? Computational Science and Engineering, 83:325-363, 2012.

[20] Incropera F.P., Dewitt D.P. Bergman T.L., Lavine, A.S. Fundamentals of heat and mass transfer. 6 ed., John Wiley & Sons, 2007.

[21] Harlow F.H., Welch J.E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. The Physics of Fluids, 8:2182-2189, 1965.
 [22] Shih T.M., Tan C.H., Hwang B.C. Effects of grid staggering on numerical scheme. Int. J. Num. Met. Fluids, 9:193-212, 1989.



[23] Devendran D., Corona E. Projection nethods, computational fluid dynamics reading group. Courant Institute of Mathematical Sciences of New York University, New York, 2009.

[24] Chorin A.J. Numerical solution of the Navier-Stokes equations. Mathematics of Computation, 22(104):745-762, 1968.

[25] Temam R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des fractionnaires. Archive for Rational Mechanics and Analysis, 33(5):377-385, 1969.

[26] Goda K. A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows. Journal of Computational Physics, 30(1):76-95, 1979.

[27] Van Kan J. A second-order accurate pressure-correction scheme for viscous incompressible flow. SIAM J. Sci. Stat. Comput., 7(3):870-891, 1986.

[28] Timmemans L.J.P., Minev P.D., Van de Vosse F.N. An approximate projection scheme for incompressible flow using spectral elements. International Journal for Numerical Methods in Fluids, 22(7):673-688, 1996.

[29] Guermond J.L., Shen J. On the error estimates for the rotational pressure-correction projection methods. Math. Comput., 73(248):1719–1737, 2004.

[30] Kalland K.A. Navier-stokes solver for single and two-phase flow. Master Thesis, University of Oslo, 2008.

[31] Villar M.M. Análise numérica detalhada de escoamentos multifásicos bidimensionais. Thesis UFU, Uberlandia, Brazil, 2007.

[32] Wesseling P. An introduction to multigrid methods. John Wiley & Sons, 1992.

[33] Hackbush W. Multigrid methods and applications. Springer-Verlag, 1985.

[34] Burden R.L., Faires J.D. Análise numérica. Pioneira Thomson Learning, 2007.

[35] Anunciação M.A.M., Pinto M.A.V., Neundorf R.L. Análise de parâmetros do método do Gradiente Conjugado pré-condicionado com Multigrid para a equação de Poisson bidimensional. In XXXVIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE), Proceedings Florianópolis, 2017.