

A SIMPLE-based algorithm with enhanced velocity corrections: the COMPLEX method

Horacio J. Aguerre^a, Cesar M. Venier^{a,b}, César I. Pairetti^{b,a}, Santiago Márquez Damián^{a,c}, Norberto M. Nigro^{a,d}

^a*Centro de Investigación de Métodos Computacionales, CONICET-UNL, Santa Fe, Argentina*

^b*Escuela de Ingeniería Mecánica, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Argentina*

^c*Facultad Regional Santa Fe, Universidad Tecnológica Nacional, Santa Fe, Argentina*

^d*Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral, Santa Fe, Argentina*

Abstract

This paper introduces a new pressure-velocity coupling algorithm based on the SIMPLEC method. The new approach considers the neighbour velocity corrections of SIMPLEC as a Taylor series expansion, introducing a first-order term to increase the accuracy of the approximation. The new term includes a velocity gradient which is assumed to be a scalar matrix constrained by means of a mass conservation equation. The stability of the method is analysed via a Fourier decomposition of the error showing a better convergence rate than SIMPLE and SIMPLEC for high relaxation factors. Afterwards, the new method is tested in two incompressible laminar flow problems where the conclusions of the stability analysis are verified. The current proposal sets a theoretical baseline for further improvements and, in view of the results, it is a promising alternative to improve SIMPLE-based algorithms.

Keywords: SIMPLE algorithm, Segregated methods, Collocated grids, OpenFOAM(R), Finite Volume Method

1. Introduction

A major numerical difficulty for solving the incompressible Navier-Stokes equations arises from the coupling between the pressure and velocity fields. Several strategies have been proposed to tackle this problem which can be classified as coupled and segregated methods. The first group solves all the governing equations at once for all the unknowns [1, 2, 3, 4, 5, 6]. Hence, pressure and velocity are assembled together and the non-linearity of the convective term is treated iteratively. Segregated algorithms, in contrast, solve the equations in a sequential fashion for a single variable at a time. The procedure is iterative and ends when the unknowns satisfy a certain convergence criterion [7, 8, 9]. In general, segregated strategies are superior in terms of memory saving and computational performance but their numerical stability may not be guaranteed when solving strongly-coupled physical problems [10, 6]. Within these group of methods, the penalty method [11], the artificial compression method [12] and projection methods [13, 14, 11] are among the most adopted in CFD codes nowadays [10]. In the projection methods, an initial estimation of the velocity is corrected by means of a pressure equation where the velocity field is forced to satisfy the divergence-free condition. In particular, the Fractional Step Methods [15] rely on a temporal splitting on the velocity where a first estimate is obtained with the momentum equation using an initial guess of the pressure. Then, the remaining velocity contribution is computed by solving a pressure equation based on the mass balance.

Under the scope of the projection methods, a new line of strategies was introduced by Patankar and Spalding with the development of the SIMPLE algorithm [16]. In this method, a pressure equation is derived after introducing the momentum equation into the mass balance. Then, the velocity field is updated based on the momentum equation including the new values of the pressure field. This procedure is iterative and employs relaxation factors to improve the stability and convergence rate of the method. Focusing on these last aspects, further improvements have been developed taking over the years the SIMPLE algorithm as a starting point. Among them, the SIMPLER algorithm [17] enhances the pressure-velocity coupling by solving an additional pressure equation. Another relevant improvement on SIMPLE was proposed by Van Doormal and Raithby [18], who included an approximation of the neighbour velocity corrections to build the pressure equation and the velocity update. This algorithm, called SIMPLEC, suppresses the need for relaxing the pressure field to guarantee stability. In order to solve transient problems, Issa proposed the PISO method [19, 20]

which consists on including corrector sub-steps to enforce mass conservation without relaxation. Many variants of the SIMPLE method were developed seeking to enhance the robustness and convergence rate of the standard SIMPLE algorithm [21, 22, 23, 24]. Their particular advantages and drawbacks can be found in the literature [9, 25, 26]. Despite the aforementioned efforts to improve SIMPLE-like algorithms, the errors introduced by the decoupling of the variables cannot be completely eliminated which makes the topic still relevant nowadays [27, 28, 29, 30].

The error introduced by the SIMPLE algorithm comes from neglecting the neighbour velocity to decouple pressure and velocity. In practice, this approximation requires relaxing the pressure to achieve a stable behaviour towards convergence of the iterative process. Regarding this, the SIMPLEC algorithm considers the neighbour velocity correction equal to the one of the current cell. In other words, it considers the velocity correction as a constant field in the proximity of each cell. This process, in most cases, enhances the ranges of stability avoiding the need to relax the pressure field. The SIMPLEC approximation may be seen as an estimation of the neighbour velocities via a zero-order truncation of a Taylor series expansion. In this context, this work proposes to employ this numerical insight by taking into account a first-order term of the same Taylor series expansion. The approach brings up the need for determining the derivative of the velocity correction, which is a topic of discussion of the present work.

This article is organized as follows. Section 2 describes the SIMPLE and SIMPLEC algorithms and then, in Section 3, the basis and theoretical background of the new method are explained. Subsequently, in Section 4, a stability analysis based on Fourier decomposition is performed for the three mentioned algorithms. Next, in Section 5, a set of laminar flow problems are solved to calibrate and investigate the performance of the new proposal. Finally, a discussion of the results and final comments are presented in the last section.

2. Theoretical basis of the SIMPLE-based algorithms

This section presents a general approach for solving the steady-state, incompressible Navier-Stokes equations via segregated pressure-based methods of the SIMPLE family. This is done under the framework of the Finite Volume Method (FVM) on collocated grids with the Rhie-Chow correction [31]. First, the standard SIMPLE algorithm for the pressure and velocity coupling is explained [16]. Then, the SIMPLEC algorithm is introduced [18] following the same formalism and focusing on the approximation of the neighbour velocity corrections [9]. This last aspect is the starting point for the method proposed in this work.

The continuum incompressible Navier-Stokes equations may be written as follows,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \nu \Delta \mathbf{u} + \Phi, \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.2)$$

where $p = (\mathcal{P}/\rho)$, ρ is the mass density field, \mathcal{P} is the pressure field, \mathbf{u} is the velocity field, ν is the kinematic viscosity and Φ is a momentum source term. Based on the FVM and linearizing the convective term, the Eqs. (2.1) and (2.2) may be discretized as:

$$a_P \mathbf{u}_P + \sum_N a_N \mathbf{u}_N = b_P \mathbf{u}_P^0 + \Phi_P - \nabla p_P \quad \forall P \in \Omega, \quad (2.3)$$

$$\sum_f \mathbf{u}_f \cdot \mathbf{S}_f = 0 \quad \forall P \in \Omega, \quad (2.4)$$

where the subscript P refers to a given cell of a Finite Volume (FV) mesh Ω , N refers to the values of neighbour cells of P and f refers to values in the faces of the cell P , $b_P \mathbf{u}_P^0$ is the contribution of the previous time-step or iteration and \mathbf{S}_f is the face-normal vector. Also, a_P and a_N are the diagonal and off-diagonal momentum matrix coefficients respectively and the gradient ∇p_P is computed by linear interpolation of first neighbour values of that field. Further detail can be found in [32, 9, 33].

Eqs. (2.3) and (2.4) are presented in a way that could serve as a starting point for both steady-state and transient algorithm formulations, depending on a_P and b_P definitions [20]. For transient solvers, b_P contains the discretized coefficients of the transient term of the momentum equation (e.g. $b_P = V_P/\Delta t$ for first order schemes, being V_P the

cell volume and Δt the time step) while, for steady-state solvers, it is defined following a numerical relaxation of the momentum equations to enhance the stability of the method [9] (e.g. $b_P = \bar{a}_P (1 - \omega_u) / \omega_u$, where \bar{a}_P includes the current cell contributions of the advective and diffusive terms and ω_u is the relaxation coefficient). Therefore, $a_P = \bar{a}_P + V_P / \Delta t$ for transient solvers and $a_P = \bar{a}_P / \omega_u$ for steady-state solvers. The present work employs on the steady-state formulation.

The system given by Eqs. (2.3) and (2.4) may be rewritten in a more suitable way for the subsequent numerical approach. First, the current cell velocity \mathbf{u}_P may be isolated from Eq. (2.3) as,

$$\mathbf{u}_P = \frac{\mathbf{H}_P - \nabla p_P}{a_P}, \quad (2.5)$$

where \mathbf{H}_P is defined as,

$$\mathbf{H}_P \equiv - \sum_N a_N \mathbf{u}_N + b_P \mathbf{u}_P^0 + \Phi_P. \quad (2.6)$$

Then, Eq. (2.5) is replaced in Eq. (2.4),

$$\sum_f (\mathbf{u}_P)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P - \nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = 0, \quad (2.7)$$

where the $(\cdot)_f$ operator interpolates cell values to the faces. Finally, an equation for the pressure field is obtained,

$$\sum_f \left(\frac{\nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P}{a_P} \right)_f \cdot \mathbf{S}_f. \quad (2.8)$$

Considering that Eqs. (2.5) and (2.8) must be solved for each cell, there are two linear system of equations: one for the velocity and one for the pressure. Now, for the iterative procedure, a subdivision of the velocity into two contributions is considered,

$$\mathbf{u}_P = \mathbf{u}_P^* + \mathbf{u}_P', \quad (2.9)$$

$$\mathbf{u}_N = \mathbf{u}_N^* + \mathbf{u}_N', \quad (2.10)$$

where \mathbf{u}^* is a velocity prediction and \mathbf{u}' is the velocity correction. This subdivision leads to an analogous separation of the variable \mathbf{H}_P ,

$$\mathbf{H}_P = \mathbf{H}_P^* + \mathbf{H}_P', \quad (2.11)$$

where,

$$\mathbf{H}_P^* = - \sum_N a_N \mathbf{u}_N^* + b_P \mathbf{u}_P^0 + \Phi_P, \quad (2.12)$$

$$\mathbf{H}_P' = - \sum_N a_N \mathbf{u}_N'. \quad (2.13)$$

Considering these last expressions, the iterative procedure begins by computing the momentum balance to predict the velocity \mathbf{u}_P^* using the previous iteration value of the pressure p^0 ,

$$\mathbf{u}_P^* = \frac{\mathbf{H}_P^* - \nabla p_P^0}{a_P}. \quad (2.14)$$

After this, the pressure is computed by introducing the Eq. (2.11) in the mass balance of Eq. (2.8),

$$\sum_f \left(\frac{\nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P^* + \mathbf{H}_P'}{a_P} \right)_f \cdot \mathbf{S}_f. \quad (2.15)$$

2.1. The SIMPLE algorithm on collocated grids

Since the correction velocity values \mathbf{u}'_N are not known before solving the pressure equation, an approximation for \mathbf{H}'_P must be proposed. In this sense, the SIMPLE method neglects the influence of the neighbour velocity correction which is equivalent to,

$$\mathbf{H}'_P = 0. \quad (2.16)$$

Thus, introducing Eq. (2.16) into (2.15),

$$\sum_f \left(\frac{\nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P^*}{a_P} \right)_f \cdot \mathbf{S}_f, \quad (2.17)$$

which is a Poisson problem for the pressure.

In collocated grids, solving this system in a straightforward manner may lead to high-frequency oscillations. The Rhie-Chow correction [31] seeks to reduce this effect by introducing a stabilizing term,

$$\sum_f \left(\frac{\nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P^*}{a_P} \right)_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{\nabla p_P}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f \right] \cdot \mathbf{S}_f, \quad (2.18)$$

where ∇p_f is a pressure gradient on a face computed from the pressure values of the adjacent cells. The Eq. (2.18) can be rearranged in to obtain the pressure equation as follows,

$$\sum_f \left(\frac{1}{a_P} \right)_f \nabla p_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}_P^*}{a_P} \right)_f \cdot \mathbf{S}_f. \quad (2.19)$$

To improve the numerical stability of the method, the pressure field is relaxed after Eq. (2.19),

$$p_P = \omega_P \tilde{p}_P + (1 - \omega_P) p_P^0. \quad (2.20)$$

where \tilde{p}_P is the pressure field obtained from Eq. (2.19).

Finally, the velocity value is computed through Eq. (2.5) where the variable \mathbf{H}_P is approximated by \mathbf{H}_P^* (as it was done for the mass balance equation),

$$\mathbf{u}_P = \frac{\mathbf{H}_P^* - \nabla p_P}{a_P}. \quad (2.21)$$

Note that the last equation is explicit since the terms r.h.s. are known. The algorithm continues iteratively from Eq. (2.14), where \mathbf{u}_P^0 and p_P^0 are replaced by the values of \mathbf{u}_P and p_P from the previous iteration, until a certain convergence criterion is reached.

2.2. Improving the neighbour velocity corrections: the SIMPLEC algorithm

The SIMPLEC [18] algorithm proposes an approximation for the neighbour velocity correction \mathbf{u}'_N ,

$$\mathbf{u}'_N = \mathbf{u}'_P, \quad (2.22)$$

and therefore the Eq. (2.13) may be expressed as,

$$\mathbf{H}'_P = \left(- \sum_N a_N \right) \mathbf{u}'_P. \quad (2.23)$$

This expression may be written in terms of the pressure gradient. To do this, an expression for \mathbf{u}'_P is derived by subtracting Eq. (2.14) from Eq. (2.5),

$$\mathbf{u}'_P = \frac{\mathbf{H}'_P - (\nabla p_P - \nabla p_P^0)}{a_P}, \quad (2.24)$$

which, in combination with Eq. (2.23), provides an expression for \mathbf{H}'_P as function of the pressure gradient,

$$\mathbf{H}'_P = \frac{(\tilde{a}_P - a_P)}{\tilde{a}_P} (\nabla p_P - \nabla p_P^0), \quad (2.25)$$

where,

$$\tilde{a}_P = a_P + \sum_N a_N. \quad (2.26)$$

Subsequently, introducing Eq. (2.25) in Eq. (2.15),

$$\sum_f \left(\frac{\nabla p_P}{a_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}'_P}{a_P} \right)_f \cdot \mathbf{S}_f + \sum_f \left[\frac{(\tilde{a}_P - a_P)}{\tilde{a}_P a_P} (\nabla p_P - \nabla p_P^0) \right]_f \cdot \mathbf{S}_f, \quad (2.27)$$

which may be rewritten as,

$$\sum_f \left(\frac{\nabla p_P}{\tilde{a}_P} \right)_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}'_P}{a_P} \right)_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0 \right]_f \cdot \mathbf{S}_f. \quad (2.28)$$

The pressure equation is finally obtained by including an analogous Rhie-Chow correction:

$$\begin{aligned} \sum_f \left(\frac{\nabla p_P}{\tilde{a}_P} \right)_f \cdot \mathbf{S}_f &= \sum_f \left(\frac{\mathbf{H}'_P}{a_P} \right)_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0 \right]_f \cdot \mathbf{S}_f \\ &+ \\ \sum_f \left\{ \left(\frac{\nabla p}{\tilde{a}_P} \right)_f - \left(\frac{1}{\tilde{a}_P} \right)_f \nabla p_f + \left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right)_f \nabla p_f^0 - \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0 \right]_f \right\} \cdot \mathbf{S}_f, \end{aligned} \quad (2.29)$$

which may be simplified as,

$$\sum_f \left(\frac{1}{\tilde{a}_P} \right)_f \nabla p_f \cdot \mathbf{S}_f = \sum_f \left(\frac{\mathbf{H}'_P}{a_P} \right)_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f. \quad (2.30)$$

The velocity is computed by including Eq. (2.25) into Eq. (2.11), and then into Eq. (2.5):

$$\mathbf{u}_P = \frac{\mathbf{H}'_P - \nabla p_P}{a_P} + \frac{(\tilde{a}_P - a_P)}{\tilde{a}_P a_P} (\nabla p_P - \nabla p_P^0). \quad (2.31)$$

where the terms may be reordered to obtain,

$$\mathbf{u}_P = \frac{\mathbf{H}'_P - \nabla p_P}{\tilde{a}_P} + \left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0. \quad (2.32)$$

In contrast to SIMPLE, SIMPLEC does not relax the pressure since it is not needed to ensure stability. The reader may notice that, if \tilde{a}_P is replaced by a_P in Eqs. (2.30) and (2.32), SIMPLEC reduces to SIMPLE.

3. A COupling Method for Pressure Linked Equations based on series eXpansions: COMPLEX

The approximation of the neighbour velocity corrections of SIMPLEC given by Eq. (2.22) can be seen as a zero-order truncation of a Taylor series expansion over \mathbf{u}'_N . This work proposes to enhance this approximation by preserving first order terms of the same expansion,

$$\mathbf{u}'_N = \mathbf{u}'_P + \mathbf{x}_{PN} \cdot \nabla \mathbf{u}'_f, \quad (3.1)$$

where \mathbf{x}_{PN} is a vector that goes from the cell centroid P to the cell centroid N . Thus, \mathbf{H}'_P may be written as,

$$\mathbf{H}'_P = \left(- \sum_N a_N \right) \mathbf{u}'_P + \left(- \sum_N a_N \right) (\mathbf{x}_{PN} \cdot \nabla \mathbf{u}'_f), \quad (3.2)$$

which combined with Eq. (2.24) leads to,

$$\mathbf{H}'_P = \frac{(\tilde{a}_P - a_P)}{\tilde{a}_P} (\nabla p_P - \nabla p_P^0) + \frac{a_P}{\tilde{a}_P} \delta_P, \quad (3.3)$$

where δ_P is defined as the second term of the r.h.s. of Eq. (3.2), which will be referred to as the Complex term,

$$\delta_P \equiv \left(- \sum_N a_N \right) (\mathbf{x}_{PN} \cdot \nabla \mathbf{u}'_f). \quad (3.4)$$

Including Eq. (3.3) in Eq. (2.15) and following an analogous procedure as done from Eq. (2.27) up to Eq. (2.30), the following pressure equation is obtained,

$$\sum_f \left[\left(\frac{1}{\tilde{a}_P} \right)_f \nabla p_f \right] \cdot \mathbf{S}_f = \sum_f \left[\frac{1}{a_P} (\mathbf{H}^*_P) \right]_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f + \sum_f \left(\frac{\delta_P}{\tilde{a}_P} \right)_f \cdot \mathbf{S}_f. \quad (3.5)$$

Then, the velocity is computed including the approximation of Eq. (3.2) in Eq. (2.11) and (2.5),

$$\mathbf{u}_P = \frac{\mathbf{H}^*_P - \nabla p_P}{\tilde{a}_P} + \left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0 + \frac{\delta_P}{\tilde{a}_P}. \quad (3.6)$$

The Eqs. (3.5) and (3.6) are similar to Eqs. (2.30) and (2.32) respectively but for the Complex term δ_P . This term requires to define the face gradient $\nabla \mathbf{u}'_f$. Putting aside this task, the general structure of the SIMPLEC algorithm remains unaltered.

3.1. Computation of the face gradient

In this work, the face gradient $\nabla \mathbf{u}'_f$ is defined as a linear interpolation of a cell-centred gradient $\nabla \mathbf{u}'_P$,

$$\nabla \mathbf{u}'_f = (\nabla \mathbf{u}'_P)_f. \quad (3.7)$$

Now, to calculate $\nabla \mathbf{u}'_P$, a derivation starts by including the subdivision of the velocity of Eq. (2.9) in the mass balance given by Eq. (2.4),

$$\sum_f (\mathbf{u}^*_P + \mathbf{u}'_P)_f \cdot \mathbf{S}_f = \sum_f (\mathbf{u}^*_P)_f \cdot \mathbf{S}_f + \sum_f (\mathbf{u}'_P)_f \cdot \mathbf{S}_f = 0 \quad (3.8)$$

In this last equation, the face values $(\mathbf{u}'_P)_f$ may be expressed as function of $\nabla \mathbf{u}'_P$ by considering the following first-order Taylor expansion,

$$(\mathbf{u}'_P)_f = \mathbf{u}'_P + \mathbf{x}_{Pf} \cdot \nabla \mathbf{u}'_P. \quad (3.9)$$

Now, this may be introduced in Eq. (3.8),

$$\sum_f (\mathbf{u}'_P + \mathbf{x}_{Pf} \cdot \nabla \mathbf{u}'_P) \cdot \mathbf{S}_f = - \sum_f (\mathbf{u}^*_P)_f \cdot \mathbf{S}_f, \quad (3.10)$$

$$\mathbf{u}'_P \cdot \sum_f \mathbf{S}_f + \sum_f (\mathbf{x}_{Pf} \cdot \nabla \mathbf{u}'_P) \cdot \mathbf{S}_f = - \sum_f (\mathbf{u}^*_P)_f \cdot \mathbf{S}_f \quad (3.11)$$

$$\sum_f (\mathbf{x}_{Pf} \cdot \nabla \mathbf{u}'_P) \cdot \mathbf{S}_f = - \sum_f (\mathbf{u}^*_P)_f \cdot \mathbf{S}_f, \quad (3.12)$$

where $\sum_f \mathbf{S}_f = \mathbf{0}$ for convex polyhedra. Then, \mathbf{u}^*_P is known from Eq. (2.14),

$$(\mathbf{u}^*_P)_f = \left[\frac{1}{a_P} (\mathbf{H}^*_P - \nabla p_P^0) \right]_f, \quad (3.13)$$

where a similar form of the Rhie-Chow correction term, given in Eq. (2.18), is included,

$$(\mathbf{u}_P^*)_f = \left[\frac{1}{a_P} (\mathbf{H}_P^* - \nabla p_P^0) \right]_f + \left[\left(\frac{\nabla p_P^0}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0 \right], \quad (3.14)$$

leading to,

$$(\mathbf{u}_P^*)_f = \left(\frac{\mathbf{H}_P^*}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0. \quad (3.15)$$

Then, replacing this last expression in Eq. (3.12),

$$\sum_f (\mathbf{x}_{Pf} \cdot \nabla \mathbf{u}'_P) \cdot \mathbf{S}_f = - \sum_f \left[\left(\frac{\mathbf{H}_P^*}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f. \quad (3.16)$$

This last equation relates the cell gradient $\nabla \mathbf{u}'_P$ with \mathbf{H}_P^* and ∇p_f^0 , which are known values after the momentum predictor. However, the Eq. (3.16) is underdetermined since the gradient $\nabla \mathbf{u}'_P$ is a second-order tensor with nine scalar unknowns. In order to solve this, an additional approximation is proposed,

$$\nabla \mathbf{u}'_P = \alpha_P \mathbf{I}. \quad (3.17)$$

This means that the gradient of the velocity corrections is considered to be a scalar matrix. Therefore, it has only one scalar unknown. Replacing this last definition in Eq. (3.16),

$$\sum_f (\mathbf{x}_{Pf} \cdot \alpha_P \mathbf{I}) \cdot \mathbf{S}_f = - \sum_f \left[\left(\frac{\mathbf{H}_P^*}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f, \quad (3.18)$$

$$\alpha_P \sum_f (\mathbf{x}_{Pf} \cdot \mathbf{I}) \cdot \mathbf{S}_f = - \sum_f \left[\left(\frac{\mathbf{H}_P^*}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f, \quad (3.19)$$

and, isolating α_P ,

$$\alpha_P = \frac{- \sum_f \left[\left(\frac{\mathbf{H}_P^*}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f}{\sum_f \mathbf{x}_{Pf} \cdot \mathbf{S}_f}. \quad (3.20)$$

Finally, the face gradient given by Eq. (3.7) may be computed using the face interpolated values of α ,

$$\nabla \mathbf{u}'_f = (\alpha_P)_f \mathbf{I}. \quad (3.21)$$

3.2. Relaxation of the Complex term

The approximation given in Eq. (3.17) provides the mathematical closure of the numerical problem at the cost of reducing the potential of the original proposal. The physical meaning of using a scalar matrix as an estimate of the velocity correction gradient is that it will introduce an isotropic deformation on the fluid. This may not be true once the velocity field is computed with the new pressure values. Taking this into account, a relaxation of the Complex term δ_P is proposed in order to avoid an overestimation of \mathbf{u}'_N . In this sense, a relaxation factor is introduced for the Complex term in Eqs. (3.5) and (3.6),

$$\sum_f \left[\left(\frac{1}{\tilde{a}_P} \right)_f \nabla p_f \right] \cdot \mathbf{S}_f = \sum_f \left[\frac{1}{a_P} (\mathbf{H}_P^*) \right]_f \cdot \mathbf{S}_f + \sum_f \left[\left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right)_f \nabla p_f^0 \right] \cdot \mathbf{S}_f + \kappa \sum_f \left(\frac{\delta_P}{\tilde{a}_P} \right)_f \cdot \mathbf{S}_f. \quad (3.22)$$

$$\mathbf{u}_P = \frac{\mathbf{H}_P^* - \nabla p_P}{\tilde{a}_P} + \left(\frac{1}{\tilde{a}_P} - \frac{1}{a_P} \right) \nabla p^0 + \kappa \frac{\delta_P}{\tilde{a}_P}. \quad (3.23)$$

where κ is the Complex relaxation factor and its value is between 0 and 1.

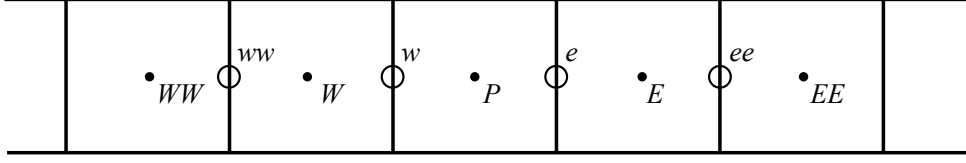


Figure 1: One-dimensional FVM discretization.

4. Stability Analysis

In order to investigate the numerical stability of the new method, an analysis based on a Fourier decomposition of the error is performed. This technique has proven to be a powerful tool to determine the numerical and physical conditions that guarantee a stable solution for linear equations. Moreover, the methodology is extendable to the analysis of multi-variable discrete systems that are addressed by segregation of the variables. The main feature of this approach is to condense the spatial information into a frequency spectrum where only the term of the frequency that produces the highest amplification of the error is taken into account. In this way, the problem is reduced to the analysis of the Fourier coefficients and its evolution in time [34].

On the downside, this technique relies on having constant coefficients in the momentum and pressure equations, which will remain constant until the beginning of the second iteration where a_P and a_N are updated with the new velocity field. That means that a sequence of several SIMPLE iterations cannot be studied via a Fourier decomposition as a whole. Instead, each iteration needs to be studied separately. Another issue of the Fourier analysis is that only periodic boundary conditions can be considered for the analysis. Nevertheless, these restrictions are the price to pay to have an analytical tool that gives valuable information about the ranges of stability and error damping of each method as a function of just a few parameters. Moreover, since this technique is used to compare each coupling method under the same conditions, these issues become irrelevant.

Many examples of the usage of this technique can be found in the literature. Shaw et al. [35] used it to study the error damping properties of the SIMPLE algorithm in staggered grids. Then, Miller [36] extended this analysis to the investigation of the performance of SIMPLE and SIMPLEC on both, collocated and staggered grids. On the other hand, Liu et al. [25] adopted the same technique to study a large number of algorithms of the SIMPLE family. More recently, Miller et al. [37] extended this technique to study Eulerian multiphase flow models, while Venier et al. [38] studied the stability features of the PISO algorithm for addressing transient and steady-state flow problems.

4.1. Fourier analysis for SIMPLE-based algorithms

In this work, a one-dimensional FV discretization is considered to simplify the analysis. A single iteration of the algorithm is studied, which can be separated into three stages: the momentum predictor, the pressure equation and the velocity update. The equations of each stage (for each method) is written in its Fourier decomposed form. In order to do this, first, these equations are written in terms of the errors of the variables (i.e. pressure and velocity), which are defined as,

$$e_{\phi,M}^k = \phi_M^E - \phi_M^k, \quad (4.1)$$

where k is a given stage of the algorithm (0 for the initial values, * for the variables after the momentum predictor, ** for the variables after the pressure equation and *** for the variables after the velocity update), E represents the exact solution of the numerical problem, ϕ is a given variable (pressure or velocity) and M represents the spatial location.

First, the momentum predictor, Eq. (2.14), is rewritten in terms of the error of the variables as follows,

$$\frac{\bar{a}_P}{\omega_u} u_P^* + a_E u_E^* + a_W u_W^* = \frac{1 - \omega_u}{\omega_u} \bar{a}_P u_P^0 - (\nabla p)_P^0 + \Phi_P, \quad (4.2)$$

$$\frac{\bar{a}_P}{\omega_u} e_{u,P}^* + a_E e_{u,E}^* + a_W e_{u,W}^* = \frac{1 - \omega_u}{\omega_u} \bar{a}_P e_{u,P}^0 - \frac{S_f}{2} (e_{p,E}^0 - e_{p,W}^0), \quad (4.3)$$

where P , W and E represent the current cell and neighbour cells respectively of the FVM stencil (see Fig. 1). Now, the errors may be written in terms of the Fourier decomposition as,

$$e_{\phi,M}^k = a_{\phi}^k e^{(i\theta \frac{x_M}{\Delta x})}, \quad (4.4)$$

which, for multivariable systems, become,

$$\begin{bmatrix} e_u \\ e_p \end{bmatrix}^k = \begin{bmatrix} u \\ p \end{bmatrix}^E - \begin{bmatrix} u \\ p \end{bmatrix}^k = \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^k e^{(i\theta \frac{x_M}{\Delta x})}, \quad (4.5)$$

where Δx represents the distance between the centroids of two adjacent cells. Replacing each error on Eq. (4.3) for its corresponding Fourier decomposition, as given by Eq. (4.4), the following equation is obtained:

$$P_u \alpha_u^* = \left(\frac{1 - \omega_u}{\omega_u} \bar{a}_P \right) \alpha_u^0 + \left(-\frac{S_f}{2} \right) (e^{i\theta} - e^{-i\theta}) \alpha_p^0, \quad (4.6)$$

where,

$$P_u = \frac{\bar{a}_P}{\omega_u} + a_E e^{i\theta} + a_W e^{-i\theta}. \quad (4.7)$$

Then, Eq. (4.6) may be rewritten as,

$$\alpha_u^* = a_1 \alpha_u^0 + b_1 \alpha_p^0, \quad (4.8)$$

where,

$$\begin{aligned} a_1 &= \frac{1 - \omega_u}{\omega_u} \bar{a}_P P_u^{-1}, \\ b_1 &= -\left(\frac{S_f}{2} \right) (e^{i\theta} - e^{-i\theta}) P_u^{-1}, \end{aligned} \quad (4.9)$$

and since $p^* = p^0$, the following amplification system is obtained,

$$\begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^* = [A_1] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0, \quad (4.10)$$

where,

$$A_1 = \begin{bmatrix} a_1 & b_1 \\ 0 & 1 \end{bmatrix}. \quad (4.11)$$

This step of the algorithm, the momentum predictor, is the same for each one of the coupling methods previously described. The next steps, the pressure equation and velocity update, depend on the coupling method, so they are derived separately. First, the SIMPLE algorithm with pressure relaxation is presented, followed by the SIMPLEC and COMPLEX algorithms.

The pressure error equation for SIMPLE is obtained from Eq. (2.19), which, in one-dimensional problems, becomes,

$$S_f (e_{p,E}^{**} - 2e_{p,P}^{**} + e_{p,W}^{**}) = -\left(\frac{a_E}{2} \right) (e_{u,EE}^* - e_{u,P}^*) - \left(\frac{a_W}{2} \right) (e_{u,P}^* - e_{u,WW}^*) + \left(\frac{1 - \omega_u}{\omega_u} \right) \left(\frac{\bar{a}_P}{2} \right) (e_{u,E}^0 - e_{u,W}^0), \quad (4.12)$$

and the Fourier decomposition of the error (with pressure relaxation) becomes,

$$\begin{aligned} \alpha_p^{**} &= \left(\frac{\omega_p}{S_f \Psi} \right) \left[\left(-\frac{a_E}{2} \right) (e^{2i\theta} - 1) + \left(-\frac{a_W}{2} \right) (1 - e^{-2i\theta}) \right] \alpha_u^*, \\ &+ \left(\frac{\omega_p}{S_f \Psi} \right) \left[\left(\frac{1 - \omega}{\omega} \right) \left(\frac{\bar{a}_P}{2} \right) (e^{i\theta} - e^{-i\theta}) \right] \alpha_u^0, \\ &+ (1 - \omega_p) \alpha_p^*, \end{aligned} \quad (4.13)$$

where $\alpha_u^{**} = \alpha_u^*$ and,

$$\Psi = e^{i\theta} - 2 + e^{-i\theta}. \quad (4.14)$$

For the velocity update, Eq. (2.21), the equation for the error may be written as,

$$e_{u,P}^{***} = -\frac{\omega_u a_E}{\bar{a}_P} e_{u,E}^* - \frac{\omega_u a_W}{\bar{a}_P} e_{u,W}^* + -\frac{\omega_u S_f}{2\bar{a}_P} (e_{p,E}^{**} - e_{p,W}^{**}) + (1 - \omega_u) e_{u,P}^0, \quad (4.15)$$

which, in Fourier terms, becomes,

$$\alpha_u^{***} = \left[\left(-\frac{\omega_u a_E}{\bar{a}_P} \right) e^{i\theta} + \left(-\frac{\omega_u a_W}{\bar{a}_P} \right) e^{-i\theta} \right] \alpha_u^{**} + \left[\left(-\frac{\omega_u S_f}{2\bar{a}_P} \right) (e^{i\theta} - e^{-i\theta}) \right] \alpha_p^{**} + (1 - \omega_u) \alpha_u^0, \quad (4.16)$$

with $\alpha_p^{***} = \alpha_p^{**}$. Finally, the amplification of the sequence when adding the pressure equation is,

$$\begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^{**} = [A_2^S] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^* + [A_4^S] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0 = ([A_2^S][A_1] + [A_4^S]) \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0, \quad (4.17)$$

and, adding the velocity update amplification contribution, the total amplification matrix (A^S) for the SIMPLE algorithm is given by,

$$\begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^{***} = [A_3^S] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^{**} + [A_5^S] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0 = ([A_3^S][A_2^S][A_1] + [A_3^S][A_4^S] + [A_5^S]) \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0 = [A^S] \begin{bmatrix} \alpha_u \\ \alpha_p \end{bmatrix}^0, \quad (4.18)$$

where,

$$[A_2^S] = \begin{bmatrix} 1 & 0 \\ c_2^S & d_2^S \end{bmatrix} \quad [A_3^S] = \begin{bmatrix} a_3^S & b_3^S \\ 0 & 1 \end{bmatrix} \quad [A_4^S] = \begin{bmatrix} 0 & 0 \\ c_4^S & 0 \end{bmatrix} \quad [A_5^S] = \begin{bmatrix} a_5^S & 0 \\ 0 & 0 \end{bmatrix}, \quad (4.19)$$

and,

$$\begin{aligned} c_2^S &= \left(\frac{\omega_p}{S_f \Psi} \right) \left[\left(-\frac{a_E}{2} \right) (e^{2i\theta} - 1) + \left(-\frac{a_W}{2} \right) (1 - e^{-2i\theta}) \right], \\ d_2^S &= (1 - \omega_p), \\ a_3^S &= \left(-\frac{\omega_u a_E}{\bar{a}_P} \right) e^{i\theta} + \left(-\frac{\omega_u a_W}{\bar{a}_P} \right) e^{-i\theta}, \\ b_3^S &= \left(-\frac{\omega_u S_f}{2\bar{a}_P} \right) (e^{i\theta} - e^{-i\theta}), \\ c_4^S &= \left(\frac{\omega_p}{S_f \Psi} \right) \left(\frac{1 - \omega_u}{\omega_u} \right) \left(\frac{\bar{a}_P}{2} \right) (e^{i\theta} - e^{-i\theta}), \\ a_5^S &= (1 - \omega_u). \end{aligned} \quad (4.20)$$

For each method, the total amplification matrix is defined as the product of the matrices of each step following the next rule,

$$[A^m] = ([A_3^m][A_2^m][A_1] + [A_3^m][A_4^m] + [A_5^m]), \quad (4.21)$$

where the superscript m indicates the coupling method being considered (S for SIMPLE, C for SIMPLEC and X for COMPLEX).

Looking at Eqs. (2.29) and (2.31), and taking into account the relation between ∇p_p^* and the velocity field after the momentum predictor given by Eq. (2.14), the amplification of the error for the pressure equation and the velocity update of SIMPLEC have a few differences compared to the SIMPLE amplification matrices,

$$\begin{aligned}
c_2^C &= c_2^S + \left(\frac{a_E + a_W}{2} \right) (e^{i\theta} - e^{-i\theta}), \\
d_2^C &= 1, \\
a_3^C &= \left(\frac{\bar{a}_P}{\omega_u \tilde{a}_P} \right) a_3^S + \frac{a_E + a_W}{\tilde{a}_P}, \\
b_3^C &= \left(\frac{\bar{a}_P}{\omega_u \tilde{a}_P} \right) b_3^S, \\
c_4^C &= \frac{1}{\omega_p} c_4^S, \\
a_5^C &= \left(\frac{\bar{a}_P}{\omega_u \tilde{a}_P} \right) a_5^S.
\end{aligned} \tag{4.22}$$

The same procedure may be applied to the COMPLEX method based on Eqs. (3.22) and (3.23). Here, it should be noticed that defining the gradient of the velocity correction $\nabla \mathbf{u}'_p$ using a single scalar unknown α , is no longer an approximation for one-dimensional problems. Eq. (3.17) now provides mathematical closure for the problem without the need for any assumption, since the gradient of a vector field in one-dimensional problems is, in fact, a scalar value. The amplification matrix elements for COMPLEX now become,

$$\begin{aligned}
c_2^X &= c_2^C + \left(\frac{a_E}{8} \right) (e^{3i\theta} + e^{2i\theta} - 2e^{i\theta} - 2 + e^{-i\theta} + e^{-2i\theta}) + \left(-\frac{a_W}{8} \right) (e^{2i\theta} + e^{i\theta} - 2 - 2e^{-i\theta} + e^{-2i\theta} + e^{-3i\theta}), \\
d_2^X &= d_2^C, \\
a_3^X &= a_3^C + \left(\frac{a_E}{4} \right) (e^{2i\theta} + e^{i\theta} - 1 - e^{-i\theta}) + \left(-\frac{a_W}{4} \right) (e^{i\theta} + 1 - e^{-i\theta} - e^{-2i\theta}), \\
b_3^X &= b_3^C, \\
c_4^X &= c_4^C, \\
a_5^X &= a_5^C.
\end{aligned} \tag{4.23}$$

The von Neumann stability criterion [34] is applied to determine the range of stability of each method. For multivariable problems, this criterion may be expressed as,

$$\beta(\theta) = \max_i \{ |\lambda_i(\theta)| \} \leq 1 \tag{4.24}$$

where β is the amplification factor and λ_i are the eigenvalues of the amplification matrix $[A^m]$. If Eq. (4.24) is verified for each frequency, then the method will be considered numerically stable (under the hypothesis of the Fourier analysis). Moreover, the value of β gives a notion about the error damping or convergence rate of the method. If the values are below but close to unity, then the method will have a slow convergence rate, while the opposite happens when the values are close to zero.

4.2. Stability results

The von Neumann stability criterion is evaluated for the different methods studied in this work. Fig. 2 shows the amplification factor for each method for low and high mesh-Reynolds numbers, ($Re_h = \Delta x U/\nu$), and using the following relaxation factors: $\omega_u = 0.7$, $\omega_p = 1$ for SIMPLEC and COMPLEX, and $\omega_p = 1 - \omega_u$ for SIMPLE. It is observed that SIMPLE has a higher amplification factor (slower convergence rate) for each frequency and for each Re_h considered. On the other hand, the SIMPLEC method gets slightly affected with the change of Re_h while COMPLEX does not show any significant variation. All methods show a stable behaviour under the Fourier conditions, while COMPLEX and SIMPLEC show lower amplification factors than those of SIMPLE. Fig. 3 shows the sensitivity of the amplification factor to the momentum relaxation factor for all frequencies and for the three coupling methods. The results are obtained for a $Re_h = 1$ and always considering the rule $\omega_p = 1 - \omega$ for SIMPLE. While the SIMPLE algorithm does not show any

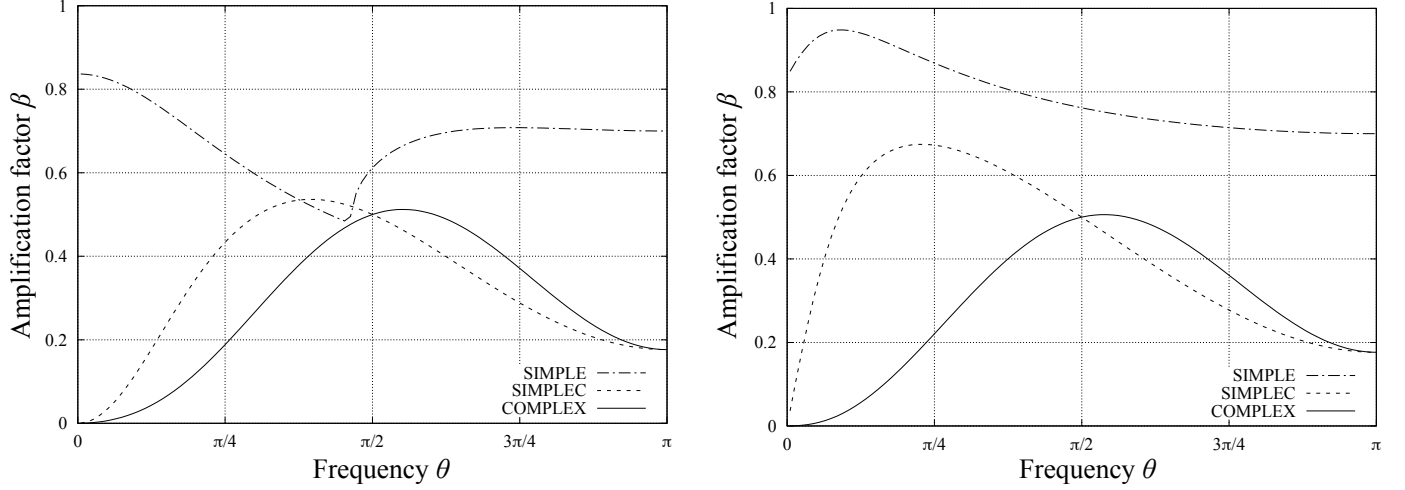


Figure 2: Amplification factor for $Re_h = 0.001$ (left) and $Re_h = 1000$ (right)

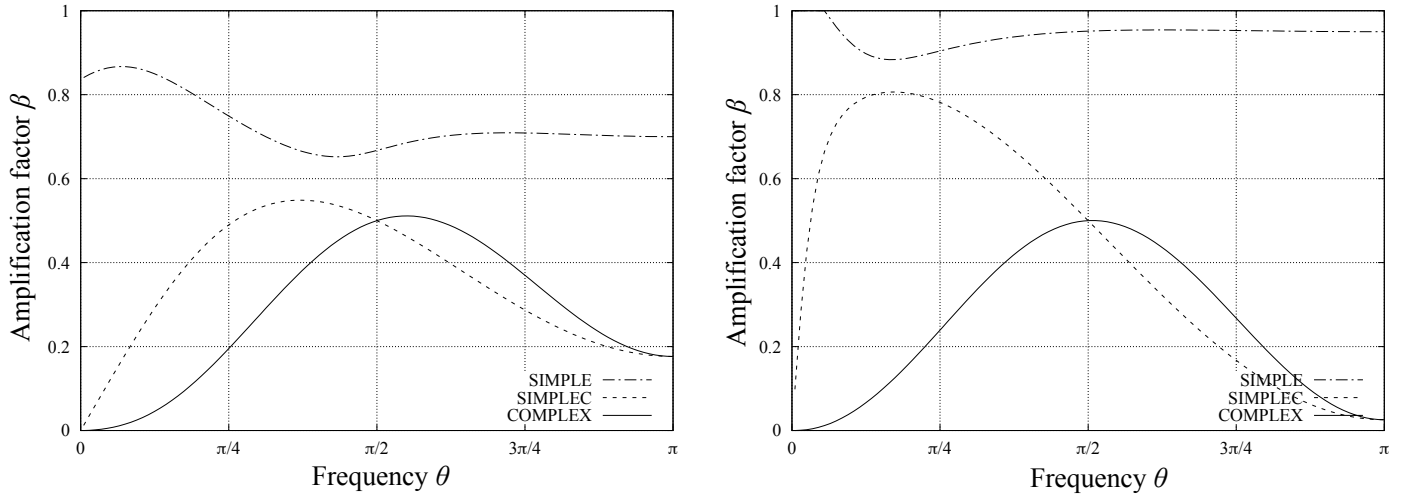


Figure 3: Amplification factor for $\omega_u = 0.7$ (left) and $\omega_u = 0.95$ (right)

significant improvement for higher values of ω (even showing a range of instability for low frequencies at $\omega_u = 0.95$ and $\omega_p = 0.05$), the SIMPLEC algorithm shows a depletion in its performance for high momentum relaxation factors (tending to unity for $\theta = 0$ as ω_u tends to 1). In this aspect, the COMPLEX algorithm shows a clear advantage over the other two methods by preserving the overall error damping, even for high momentum relaxation factors. This theoretical advantage is very important since it indicates that the good performance of the method holds up even when the convergence rate to the steady-state solution is accelerated by increasing the momentum relaxation factors.

The analysis is deepened by considering a continuous range of relaxation factors from 0 to 1 for the three methods, as shown in Fig. 4. These figures show a clear advantage of SIMPLEC over SIMPLE for almost every relaxation factor considered, and a clear advantage of COMPLEX over SIMPLEC for high values of the amplification factor. Here it may be noted that, for very low values of ω_u , all methods tend to the same stability behaviour.

To quantify the overall error damping for a general problem, where the initial solution is formed by an equally distributed amount of modes (i.e. assuming that there is no frequency that is clearly dominant over the others in the initial values of u and p fields), an average amplification factor over all the frequencies is considered. This is shown in Fig. 5, where the mean value of β over all frequencies is plotted for all the possible values of the momentum relaxation factor. Here it is also clear that, while all methods are stable under these conditions, the SIMPLEC and COMPLEX algorithms are clearly more efficient in terms of the error damping, and the COMPLEX algorithm outperforms SIMPLEC when the relaxation factor is increased.

5. Test cases

In this section, the COMPLEX method is analysed in incompressible, laminar and stationary Newtonian flow problems. The numerical performance of the method is compared with SIMPLE and SIMPLEC by measuring the total number of iterations required to achieve a convergence criterion. The details of each problem are explained followed by the definition of the convergence criterion adopted. Two different analysis over the COMPLEX method are performed: first, a series of simulations are carried out to investigate the Complex factor κ , and then, a comparison of the current proposal with SIMPLE and SIMPLEC is presented for various conditions.

5.1. Problems description

The evaluation of the COMPLEX method is performed by solving two tests for incompressible and stationary flows. The first of them is a cubic cavity test and the second one is a Backward Facing Step (BFS) channel.

All the simulations use the same numerical setup. The convective terms are discretised with an upwind scheme and the pressure gradient operator is computed with a Gauss-based formula using linear interpolation to define the face values. The diffusive terms are discretised with a Gauss-based formula where the face gradient is assembled with a first-neighbour stencil. The linear system of equations are solved until the residuals for both momentum and mass balances decrease two orders of magnitude. The momentum relaxation factor ω_u is a variable of the parametric study. The pressure relaxation factor $\omega_p = 1$ for COMPLEX and SIMPLEC, while for SIMPLE is defined as $\omega_p = 1 - \omega_u$.

5.1.1. Cubic cavity

A graphical scheme of this problem is depicted in Fig. 6. Here, a tangential fixed velocity of 1 m/s is imposed on the upper face of the cubic domain of length 1 m. The remaining faces of the cavity are walls with a non-slip condition. The case is solved using two different values of the kinematic viscosity, $\nu = 2.5 \times 10^{-3} \text{ m}^2/\text{s}$ ($Re = 400$) and $\nu = 2.5 \text{ m}^2/\text{s}$ ($Re = 0.4$). For the purposes of this work, the cavity is discretised with hexahedral cells using three different refinement levels: a coarse mesh with 25 divisions per side (15625 cells), a medium mesh with 50 divisions per side (1.25×10^5 cells) and a fine mesh with 100 divisions per side (10^6 cells).

5.1.2. Backward Facing Step

The domain and dimensions of this test are shown in Fig. 7. At the inlet, a constant normal velocity of 1 m/s is imposed. The outlet has a constant pressure and the remaining faces of the channel are walls with a non-slip condition. Analogously to the cavity case, the problem is simulated with two different viscosity values, $\nu = 5.141 \times 10^{-3} \text{ m}^2/\text{s}$ ($Re = 389$) and $\nu = 5.141 \text{ m}^2/\text{s}$ ($Re = 0.389$). The domain is discretized with hexahedral cells using three different mesh

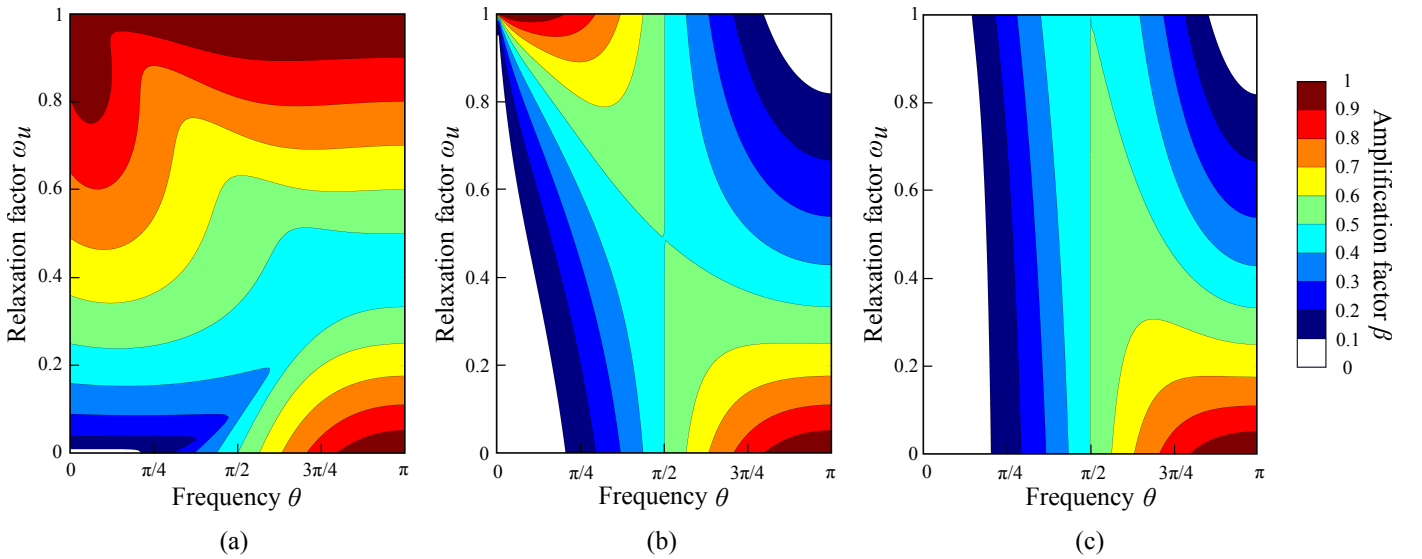


Figure 4: Amplification values for SIMPLE (a), SIMPLEC (b) and COMPLEX (c) for different relaxation factors and frequencies

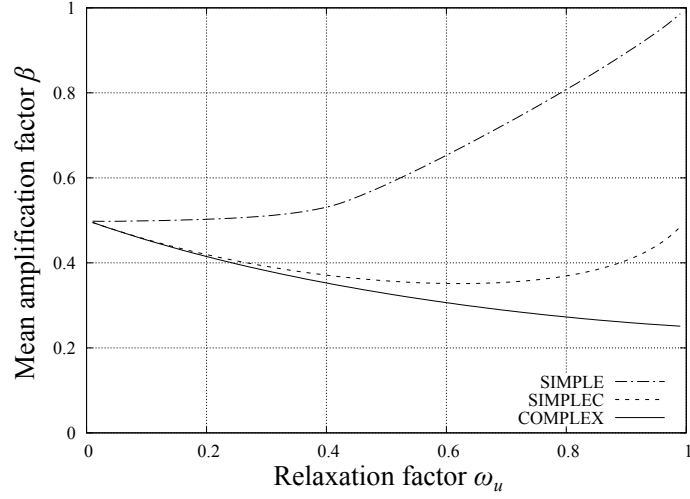


Figure 5: Mean amplification factor as a function of the momentum relaxation factor ω_u

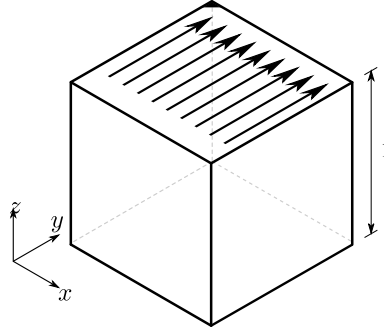


Figure 6: The cubic cavity problem where a fixed tangential velocity is imposed on the upper face.

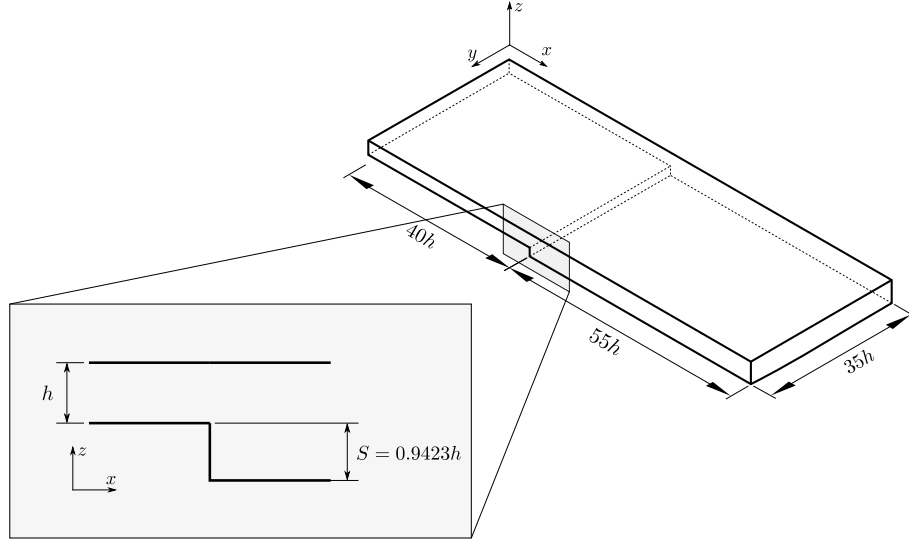


Figure 7: Description of the BFS domain where its dimensions are defined relative to the inlet height $h = 1$ m.

sizes as defined in Table 1 with a variable cell size as shown in Fig. 8, where the regions near the wall and the step are more refined than the rest of the domain.

5.2. Error measurement

To analyse the rate of convergence of the different algorithms, an error norm is defined. This norm is the residual field of the discretized Navier-Stokes equations, which are computed for each cell P of the FV mesh Ω . Then, the momentum residual is expressed as,

$$R_1(P) = a_P \mathbf{u}_P + \sum_N a_N \mathbf{u}_N - b_P \mathbf{u}_P^0 - \Phi_P + \nabla p_P \quad \forall P \in \Omega, \quad (5.1)$$

and, analogously, the mass residual is computed as follows,

$$R_2(P) = \sum_f \mathbf{u}_f \cdot \mathbf{S}_f \quad \forall P \in \Omega, \quad (5.2)$$

where the face velocity value \mathbf{u}_f is replaced in terms of the momentum equation as it was done in Eq. (2.8),

$$R_2(P) = \sum_f \left(\frac{1}{a_P} \right)_f \nabla p_f \cdot \mathbf{S}_f - \sum_f \left(\frac{\mathbf{H}_P}{a_P} \right)_f \cdot \mathbf{S}_f \quad \forall P \in \Omega. \quad (5.3)$$

Then, the Root Mean Square (RMS) of the R_1 and R_2 fields are computed,

$$\bar{R}_1 = \sqrt{\sum_{P \in \Omega} R_1(P)^2}, \quad (5.4)$$

$$\bar{R}_2 = \sqrt{\sum_{P \in \Omega} R_2(P)^2}. \quad (5.5)$$

Regarding \bar{R}_1 and \bar{R}_2 , the following convergence criterion is proposed: a numerical simulation will be considered converged if both values, \bar{R}_1 and \bar{R}_2 are below a tolerance of 10^{-9} ,

$$\begin{cases} \bar{R}_1 \leq 10^{-9} \\ \bar{R}_2 \leq 10^{-9} \end{cases} . \quad (5.6)$$

Mesh Level	Pre-step divisions			Post-step divisions			Total Cells
	x -axis	y -axis	z -axis	x -axis	y -axis	z -axis	
Coarse	20	40	8	26	40	16	23040
Medium	40	80	15	52	80	30	172800
Fine	80	160	30	104	160	60	1382400

Table 1: Scheme of divisions for the BFS case. The pre-step and post-step are the upstream and downstream regions of the step transversal section.

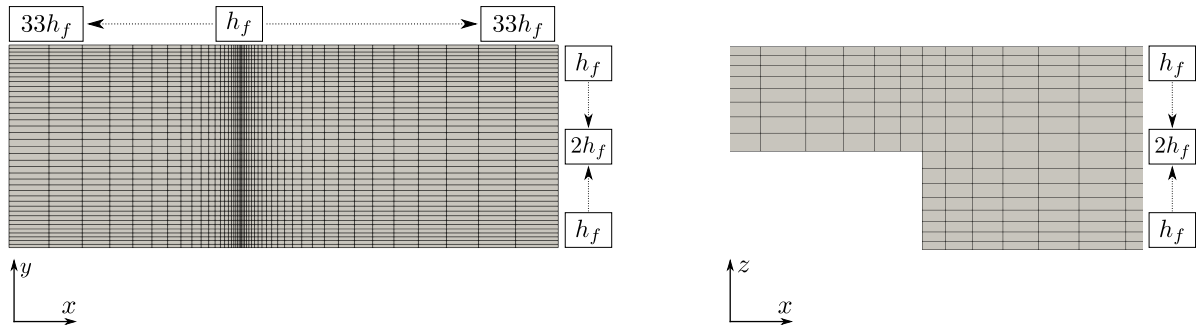


Figure 8: Variable cell refinement in the BFS. On the left figure, the cell size grows from the step point up to the inlet and outlet boundaries and from the lateral walls up to the middle longitudinal section. On the right, the cell size in the post-step regions grows from the bottom and upper walls up to the middle height section of the channel.

It is important to highlight that the residuals are only function of the velocity, pressure, and of the momentum equation coefficients which are equally assembled for the three methods studied. Thus, the computation of \bar{R}_1 and \bar{R}_2 are independent of the coupling method.

5.3. Study of the Complex factor κ

In this section, the influence of κ on the convergence rate is analysed in combination with the relaxation factor for the momentum equation ω_u measuring the number of iterations to reach convergence as defined in Eq. (5.6). Here, a series of simulations of the cavity and BFS problems are solved using the medium refinement level. For each case, two different physical configurations are addressed according to the proposed Reynolds numbers. Each one of the four simulations are solved varying the values of κ and ω_u . The Complex factor ranges from 0 to 1 with a step of 0.05 (21 values) and the relaxation factor ranging from 0.8 up to 1 with a step of 0.1 (21 values). Altogether, this parametric study consists of 1764 simulations.

In order to compare the performance of the methods, the relative number of iterations (RI) is defined as follows,

$$RI(\omega_u, \kappa) = \frac{NI(\omega_u, \kappa) - NI(\omega_u, 0)}{NI(\omega_u, 0)} \times 100\%, \quad (5.7)$$

where $NI(\omega_u, \kappa)$ is the number of iterations required for convergence using the relaxation value ω_u and the Complex factor κ . In this context, $NI(\omega_u, 0)$ is equivalent to the number of iterations required for convergence of the SIMPLEC algorithm. Fig. 9 shows RI values for all the studied cases.

The results of the simulations are presented in the Tables 2, 3, 4 and 5, and in Fig. 9. Here, the required number of iterations to reach the convergence as function of ω_u and κ is shown. The tables focus on the results that are closer to the optimal values of ω_u (between 0.92 and 1) using a κ step of 0.1 instead of 0.05.

The Tables 2, 3, and the upper plots of Fig. 9 present the results for relative low Reynolds simulations, while Tables 4, 5, and the lower plots of Fig. 9 present the results for relative high Reynolds simulations.

In general, it is concluded that the addition of the Complex term does not influence the convergence when using low relaxation factors. On the contrary, with high values of ω_u , the addition of the Complex term reduces the number of iterations to approximately 20 %. The best results are obtained with values of the Complex factor $\kappa \leq 0.5$, while $\kappa \geq 0.5$ values may induce instabilities. In particular, the BFS problem at $Re = 389$ presents an extended divergence zone for the highest values of ω_u , followed by a slow convergence zone (coloured in grey) for lower values of ω_u . Despite this behaviour, the simulations are stable and have a maximum convergence rate for low values of κ .

In view of these results, a Complex factor $\kappa = 0.2$ will be used in the following analysis since it is adequate to speed up the convergence without losing stability.

5.4. Comparison of COMPLEX with SIMPLE and SIMPLEC

In this section, a comparison of the COMPLEX strategy against SIMPLE and SIMPLEC is performed. As in the previous study, the number of iterations for convergence is measured. The numerical problems defined in the previous

$\omega_u \setminus \kappa$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	div	div	div	div	div	div	div	div	div	div	div
0.99	1993	1702	1742	1754	1747	1742	1761	div	div	div	div
0.98	991	842	845	856	866	870	870	868	865	863	864
0.97	658	551	559	560	565	570	574	575	575	575	573
0.96	491	425	414	416	418	421	424	426	427	428	428
0.95	391	348	327	330	332	333	335	337	338	339	339
0.94	373	373	373	373	373	373	373	373	373	373	373
0.93	438	438	438	438	438	438	438	438	438	438	438
0.92	504	504	504	504	504	504	504	504	504	504	504

Table 2: Required number of iterations at convergence for the Cavity case with $Re = 0.4$

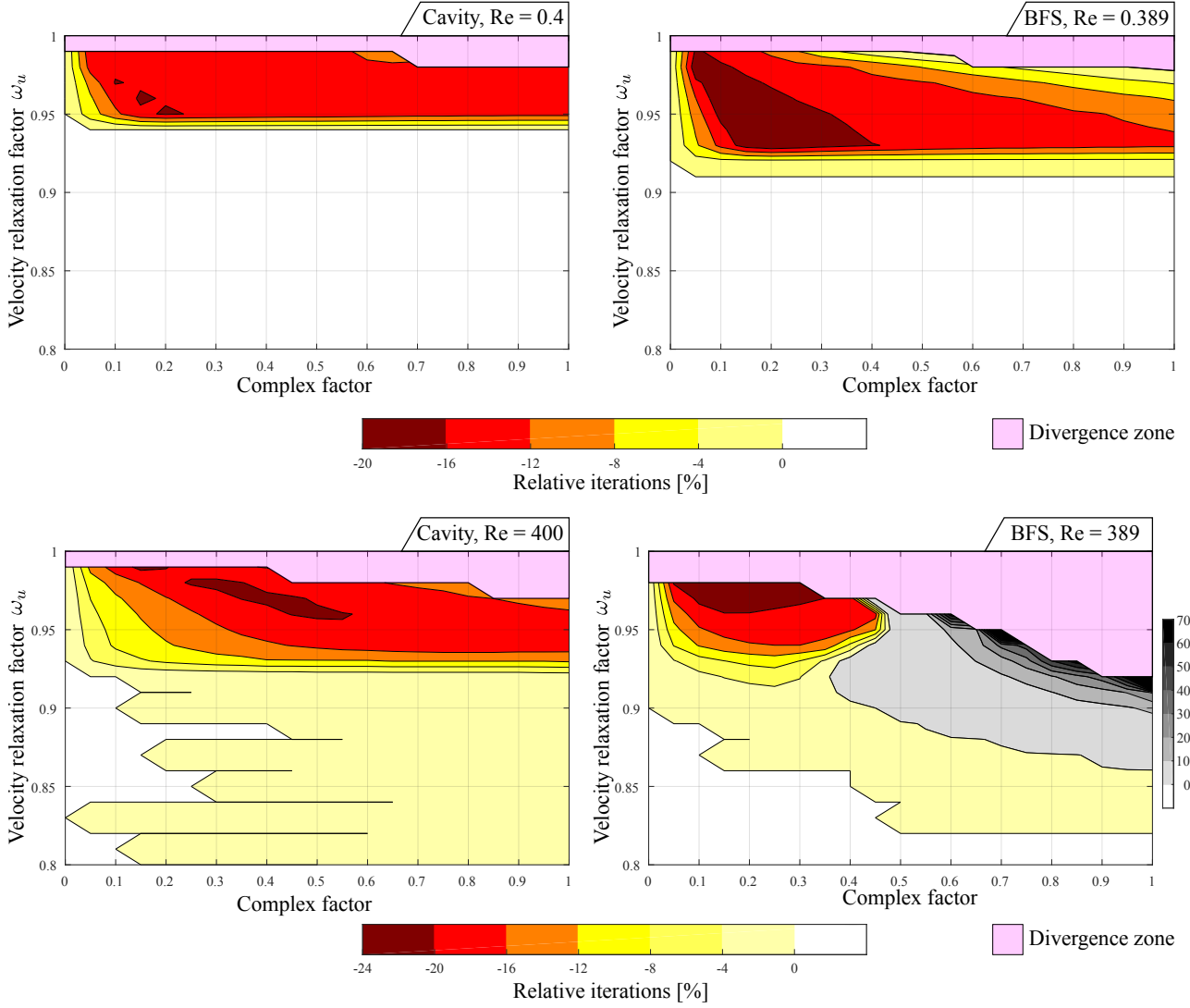


Figure 9: Isolines of RI as function of the momentum relaxation factor, the Complex factor for the Cavity and BFS cases, and the values of the Reynolds number employed respectively.

section are solved using the coarse, medium and fine mesh levels. Moreover, the effect of momentum relaxation is included considering ω_u values from 0.8 to 1 using a step of 0.1.

The results are presented for to the three mesh refinement levels in Figs. 10, 11 and 12. Each figure includes six bar

$\omega_u \setminus \kappa$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	div	div	div	div	div	div	div	div	div	div	div
0.99	1974	1697	1753	1865	1941	1993	div	div	div	div	div
0.98	981	816	844	854	874	904	929	949	966	980	992
0.97	651	528	550	560	565	571	581	594	606	618	627
0.96	486	392	403	413	418	421	424	428	435	442	449
0.95	387	320	317	324	330	333	335	336	339	342	346
0.94	320	272	259	266	271	274	276	277	278	280	282
0.93	273	237	220	225	229	232	234	235	236	237	238
0.92	240	233	233	233	233	233	233	233	233	233	233

Table 3: Required number of iterations at convergence for the BFS case with $Re = 0.389$

$\omega_u \setminus \kappa$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	div	div	div	div	div	div	div	div	div	div	div
0.99	1611	1321	1286	1333	1357	div	div	div	div	div	div
0.98	802	693	652	635	648	662	671	678	682	div	div
0.97	532	471	448	431	423	427	435	441	446	449	452
0.96	398	357	342	330	321	317	319	323	327	331	333
0.95	318	288	277	269	262	257	255	256	258	261	263
0.94	266	243	235	229	225	221	218	216	215	216	217
0.93	239	225	218	214	211	211	211	210	210	210	209
0.92	243	243	242	242	242	241	241	241	241	241	240

Table 4: Required number of iterations at convergence for the Cavity case with $Re = 400$

$\omega_u \setminus \kappa$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	div	div	div	div	div	div	div	div	div	div	div
0.99	div	div	div	div	div	div	div	div	div	div	div
0.98	818	624	632	637	div	div	div	div	div	div	div
0.97	540	427	422	428	435	div	div	div	div	div	div
0.96	403	329	323	327	329	478	3086	div	div	div	div
0.95	322	272	264	266	268	351	712	4714	div	div	div
0.94	269	236	227	226	250	320	483	1151	div	div	div
0.93	240	225	218	221	246	299	397	608	1427	div	div
0.92	242	235	230	232	256	292	351	457	696	1087	8863

Table 5: Required number of iterations at convergence for the BFS case with $Re = 389$

plots arranged in three rows and two columns. The left column presents the results for the cubic cavity case while the right one presents the BFS ones. The number of iterations at convergence of the COMPLEX method, $NI_X(\omega_u)$, is shown in the first row. The second and third row display the results for SIMPLEC and SIMPLE in relative terms using the RI index,

$$RI(\omega_u) = \frac{NI_{(S,C)}(\omega_u) - NI_X(\omega_u)}{NI_X(\omega_u)} \times 100\%, \quad (5.8)$$

where $NI_{(S,C)}(\omega_u)$ is the absolute number of iterations needed at convergence for the SIMPLE or SIMPLEC methods for a given relaxation factor ω_u . In all plots, the blue bars represent the low Reynolds simulations and the red ones represent the high Reynolds simulations.

The results indicate that the difference between COMPLEX and the other methods start becoming relevant when COMPLEX is reaching its minimum number of iterations at convergence (which is the lowest of the three methods). Moreover, it is observed that the convergence for the highest values of ω_u is better handled by the COMPLEX method. An hypothesis for this behaviour is that higher relaxation factors leads to velocity predictions that are less consistent with the divergence-free condition. For this, the COMPLEX strategy includes a numerical correction which is introduced in the pressure equation.

For finer meshes, the point of minimum iterations at convergence occurs at a higher value of ω_u while, for coarser meshes, the range of ω_u in which COMPLEX converges faster than the other methods is wider. The Reynolds number also affects the convergence of the methods. It is noted that for the relative high values, less iterations are required to reach the convergence criterion with COMPLEX. This advantage is also observed in the comparison with SIMPLE and SIMPLEC, where the relative differences are higher for the relative high Reynolds number.

The COMPLEX algorithm was implemented using the OpenFOAM(R) libraries for the FVM [39]. The current implementation is compared with the OpenFOAM(R) version of SIMPLEC. The analysis shows that each COMPLEX iteration takes 5% more time than SIMPLEC for both problems of these section. The maximum computational performance is obtained using high values of the momentum relaxation factor. In these conditions, the COMPLEX method saves between 10 and 20% of iterations which, in the end, leads to a higher computational efficiency.

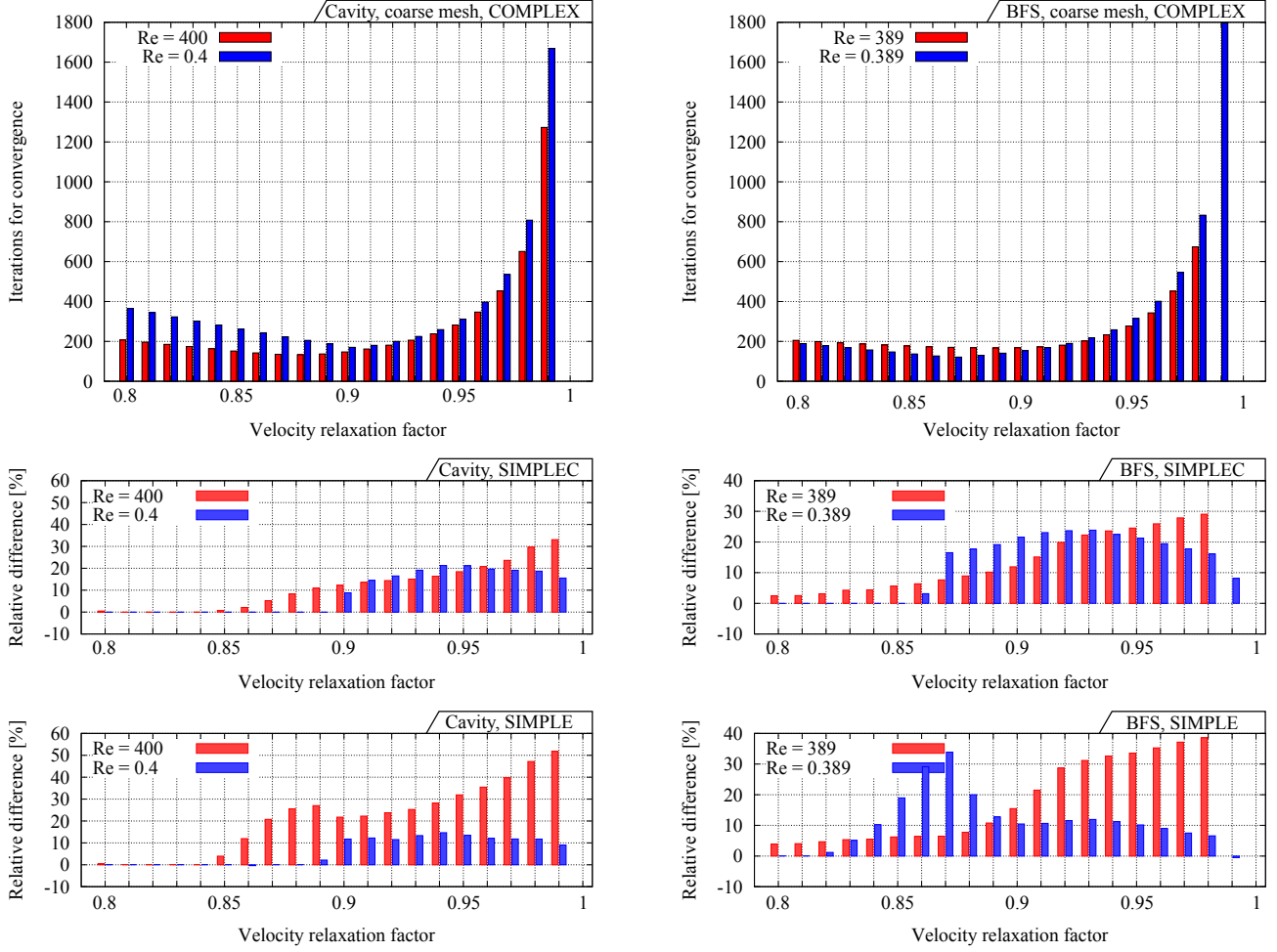


Figure 10: Comparison of the number of iterations required for convergence between COMPLEX, SIMPLEC and SIMPLE algorithms using the coarse mesh refinement level. On the left are the results of the cavity case and on the right, those respective to the BFS.

6. Conclusions

This paper presented a new segregated algorithm to solve incompressible and stationary flow problems in the context of the FVM, called COMPLEX. It is based on the SIMPLEC method where the neighbour velocity correction is enhanced using a first-order Taylor series expansion. The current procedure requires computing a velocity derivative which is approximated as a scalar matrix defined by a mass balance equation. The contribution of the first-order term of the Taylor expansion is included in the pressure equation and can be relaxed by the so-called Complex factor.

The stability of the COMPLEX method was compared against SIMPLE and SIMPLEC via a Fourier decomposition of the error in one-dimensional cases. This study showed that, for low momentum relaxation factors, both SIMPLEC and COMPLEX behave similarly but, for high momentum relaxation factors, the COMPLEX method has a better performance in terms of error damping. Thus, under the hypothesis of the Fourier analysis, the current method has a higher convergence rate.

Two three-dimensional problems were solved: a cubic cavity and a BFS. First, the optimal value of the Complex relaxation factor was investigated. To do this, an exhaustive study of more than 1500 simulations was performed where the number of iterations at convergence was measured for different values of the momentum relaxation factor, the Complex relaxation factor and the Reynolds number. Based on the results, a Complex relaxation factor of $\kappa = 0.2$ was chosen as the most suitable for the problems under study. Secondly, the performance of the COMPLEX method was compared with SIMPLE and SIMPLEC. Here, each problem was solved using three different mesh refinements for the low and high values of the Reynolds number. The results proved that the COMPLEX method has a better convergence rate for high values of the momentum relaxation factor which verifies the results of the Fourier analysis. The advantages

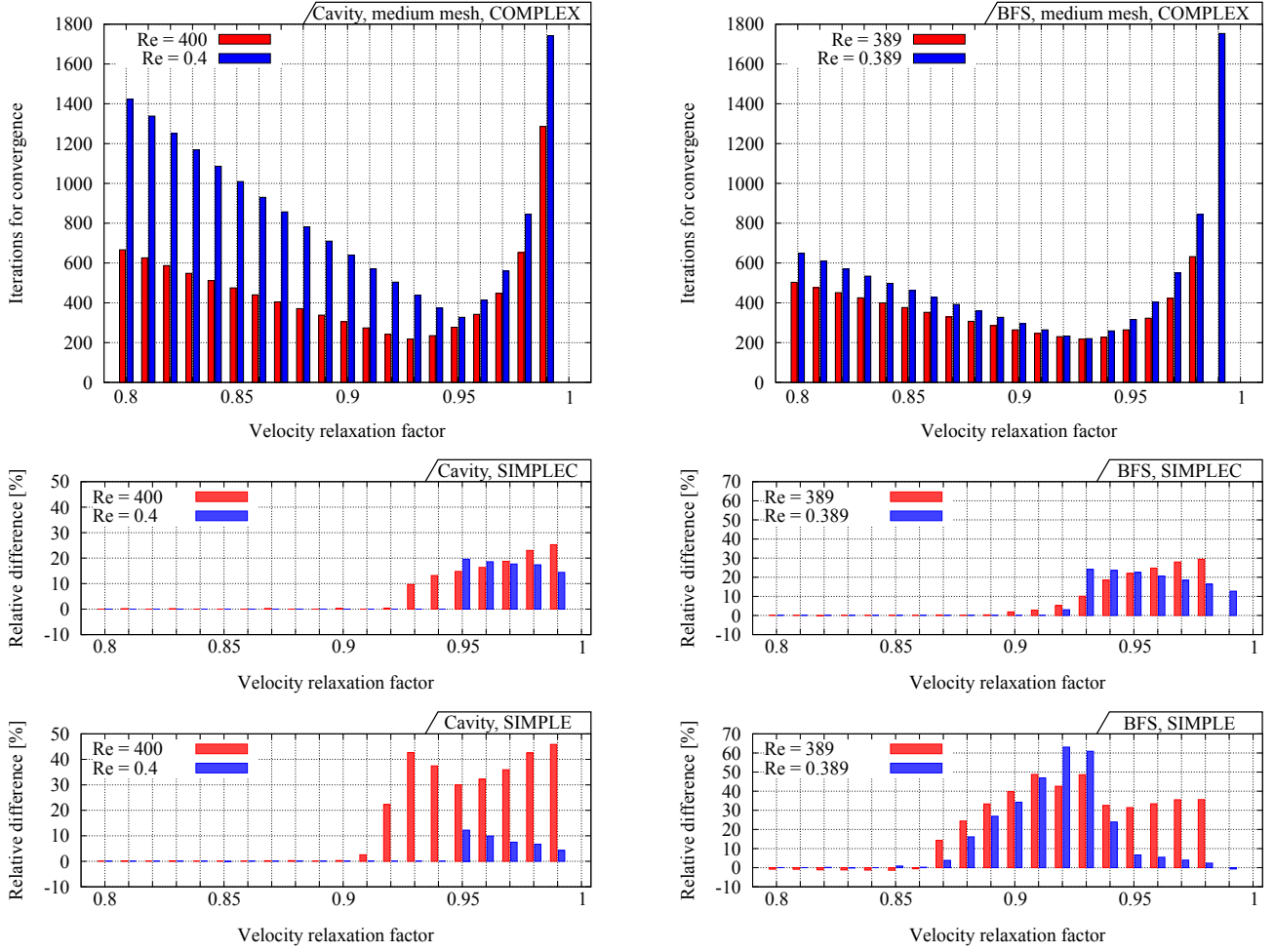


Figure 11: Comparison of the number of iterations required for convergence between COMPLEX, SIMPLEC and SIMPLE algorithms using the medium mesh refinement level. On the left are the results of the cavity case and on the right, those respective to the BFS.

are more notorious in the coarse meshes and for the problems with relative high Reynolds numbers.

Based on the results of the present work, the current strategy is a promising technique to speed up the convergence of SIMPLE-based algorithms. Its computational implementation is simple and does not introduce significant computational costs compared to the SIMPLEC method. A challenge for future works consists on improving the prediction of the velocity derivative, Eq. (3.21), by allowing an anisotropic deformation of the velocity correction.

References

- [1] Z. Mazhar, A procedure for the treatment of the velocity-pressure coupling problem in incompressible fluid flow, *Numerical Heat Transfer: Part B: Fundamentals* 39 (2001) 91–100.
- [2] Z. Mazhar, An enhancement to the block implicit procedure for the treatment of the velocity-pressure coupling problem in incompressible fluid flow, *Numerical Heat Transfer: Part B: Fundamentals* 41 (2002) 493–500.
- [3] Z. Chen, A. Przekwas, A coupled pressure-based computational method for incompressible/compressible flows, *Journal of Computational Physics* 229 (2010) 9150–9165.
- [4] M. Darwish, I. Sraj, F. Moukalled, A coupled finite volume solver for the solution of incompressible flows on unstructured grids, *Journal of Computational Physics* 228 (2009) 180–201.
- [5] A fully coupled Navier-Stokes solver for fluid flow at all speeds, author=Darwish, M and Moukalled, F, *Numerical Heat Transfer, Part B: Fundamentals* 65 (2014) 410–444.
- [6] T. Uroić, H. Jasak, Block-selective algebraic multigrid for implicitly coupled pressure-velocity system, *Computers & Fluids* 167 (2018) 100–110.
- [7] J. H. Ferziger, M. Peric, *Computational methods for fluid dynamics*, Springer Science & Business Media, 2012.
- [8] H. K. Versteeg, W. Malalasekera, *An introduction to Computational Fluid Dynamics: the Finite Volume Method*, Pearson Education, 2007.

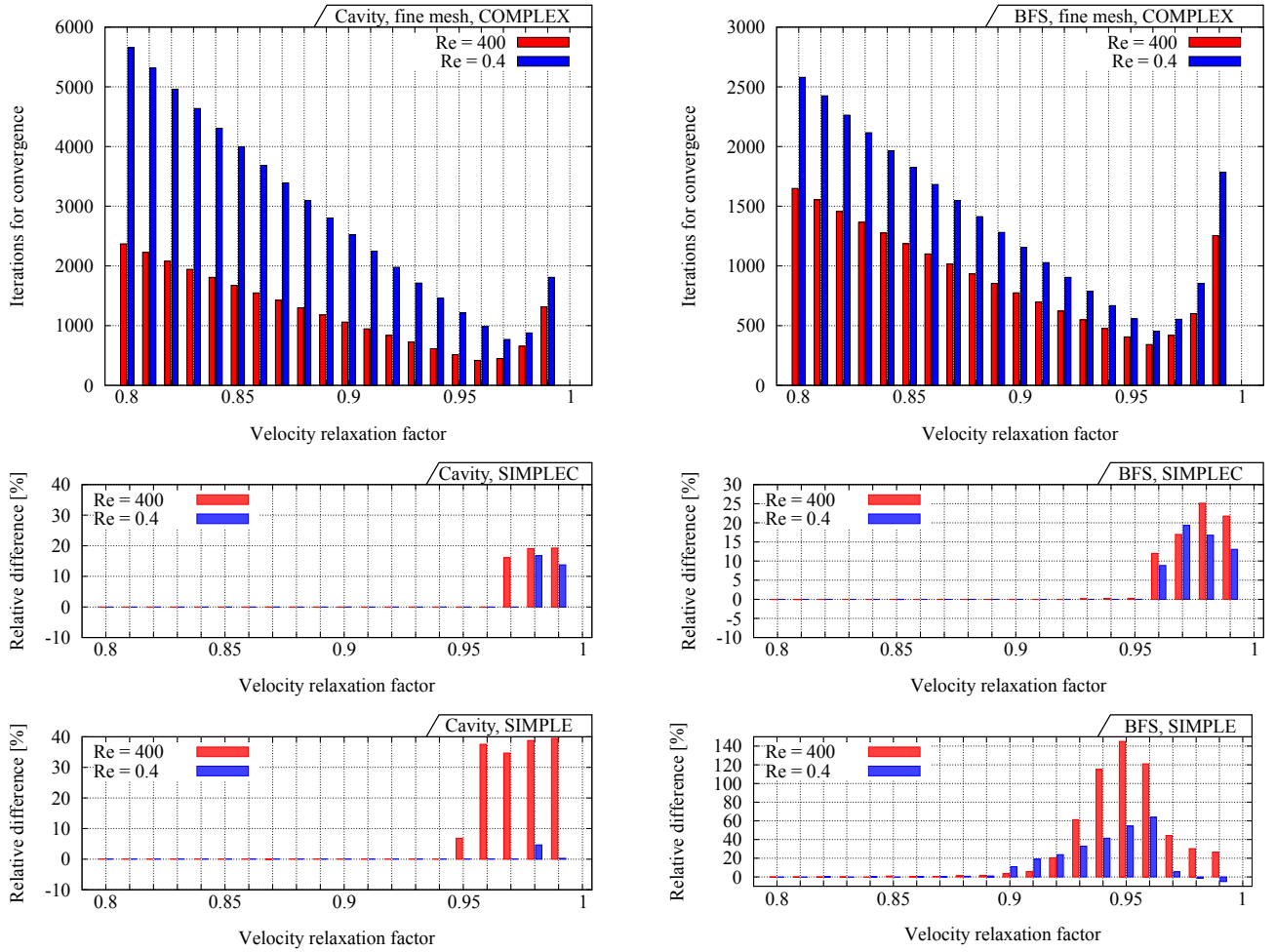


Figure 12: Comparison of the number of iterations required for convergence between COMPLEX, SIMPLEC and SIMPLE algorithms using the fine mesh refinement level. On the left are the results of the cavity case and on the right, those respective to the BFS.

- [9] F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab, volume 113, Springer, 2015.
- [10] H. Wang, H. Wang, F. Gao, P. Zhou, Z. J. Zhai, Literature review on pressure–velocity decoupling algorithms applied to built-environment CFD simulation, Building and Environment (2018).
- [11] R. Temam, Une méthode d’approximation de la solution des équations de Navier-Stokes, Bull. Soc. Math. France 98 (1968) 115–152.
- [12] A. Harten, The artificial compression method for computation of shocks and contact discontinuities. I. Single conservation laws, Communications on Pure and Applied Mathematics 30 (1977) 611–638.
- [13] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, Journal of computational physics 2 (1967) 12–26.
- [14] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Mathematics of computation 22 (1968) 745–762.
- [15] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, Journal of computational physics 59 (1985) 308–323.
- [16] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer 15 (1972) 1787–1806.
- [17] S. Patankar, Numerical heat transfer and fluid flow, CRC press, 1980.
- [18] J. Van Doormaal, G. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, Numerical Heat Transfer 7 (1984) 147–163.
- [19] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1986) 40–65.
- [20] R. I. Issa, A. Gosman, A. Watkins, The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme, Journal of Computational Physics 62 (1986) 66–82.
- [21] W.-Q. Tao, Hiroyuki Ozoe, A modified pressure-correction scheme for the SIMPLER method, MSIMPLER, Numerical Heat Transfer: Part B: Fundamentals 39 (2001) 435–449.
- [22] Z. Qu, W. Tao, Y. He, Implementation of CLEAR algorithm on collocated grid system and application examples, Numerical Heat Transfer 47 (2004) 65–96.

- [23] Y. Cheng, T. Lee, H. Low, W. Tao, An efficient and robust numerical scheme for the SIMPLER algorithm on non-orthogonal curvilinear coordinates: CLEARER, *Numerical Heat Transfer, Part B: Fundamentals* 51 (2007) 433–461.
- [24] D. Sun, Z. Qu, Y. He, W. Tao, An efficient segregated algorithm for incompressible fluid flow and heat transfer problems—IDEAL (inner doubly iterative efficient algorithm for linked equations) Part I: Mathematical formulation and solution procedure, *Numerical Heat Transfer, Part B: Fundamentals* 53 (2008) 1–17.
- [25] Y. Liu, F. Wang, Y.-t. Li, Fourier analysis of the SIMPLE series, *Numerical Heat Transfer, Part B: Fundamentals* 69 (2016) 197–216.
- [26] F. Wang, L. Zheng, Y. T. Li, Fourier analysis of the effect of non-constant terms on the convergence properties of SIMPLE algorithm formulated on a staggered grid, in: *Applied Mechanics and Materials*, volume 204, Trans Tech Publ, pp. 4688–4693.
- [27] J. Li, Q. Zhang, Z.-Q. Zhai, An efficient SIMPLER-revised algorithm for incompressible flow with unstructured grids, *Numerical Heat Transfer, Part B: Fundamentals* 71 (2017) 425–442.
- [28] A consistent SIMPLE algorithm with extra explicit prediction—SIMPLEPC, author=Xiao, Hui and Wang, Junbo and Liu, Zhichun and Liu, Wei, *International Journal of Heat and Mass Transfer* 120 (2018) 1255–1265.
- [29] J. Aoussou, J. Lin, P. F. Lermusiaux, Iterated pressure-correction projection methods for the unsteady incompressible Navier–Stokes equations, *Journal of Computational Physics* 373 (2018) 940–974.
- [30] H. J. Aguerre, C. I. Pairetti, C. M. Venier, S. M. Damián, N. M. Nigro, An oscillation-free flow solver based on flux reconstruction, *Journal of Computational Physics* 365 (2018) 135–148.
- [31] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, *AIAA Journal* 21 (1983) 1525–1532.
- [32] H. Jasak, Error analysis and estimation for the Finite Volume Method with applications to fluid flows, Ph.D. thesis, Imperial College of Science, Technology and Medicine, 1996.
- [33] S. Márquez Damián, An extended mixture model for the simultaneous treatment of short and long scale interfaces, Ph.D. thesis, Universidad Nacional del Litoral, 2013.
- [34] C. Hirsch, *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*, Butterworth-Heinemann, 2007.
- [35] G. Shaw, S. Sivaloganathan, On the smoothing properties of the SIMPLE pressure-correction algorithm, *International Journal for Numerical Methods in Fluids* 8 (1988) 441–461.
- [36] T. F. Miller, Fourier analysis of the SIMPLE algorithm formulated on a collocated grid, *Numerical Heat Transfer* 30 (1996) 45–66.
- [37] T. F. Miller, D. J. Miller, A Fourier analysis of the IPSA/PEA algorithms applied to multiphase flows with mass transfer, *Computers & fluids* 32 (2003) 197–221.
- [38] C. M. Venier, C. I. Pairetti, S. M. Damian, N. M. Nigro, On the stability analysis of the PISO algorithm on collocated grids, *Computers & Fluids* 147 (2017) 25–40.
- [39] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Computers in Physics* 12 (1998) 620–631.