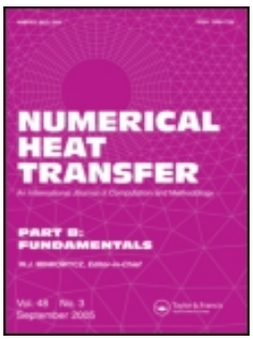

Taylor & Francis
Taylor & Francis Group

Numerical Heat Transfer, Part B: Fundamentals
An International Journal of Computation and Methodology

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: https://www.tandfonline.com/loi/unhb20

Fully Consistent SIMPLE-Like Algorithms on Collocated Grids

Dmitry K. Kolmogorov , Wen Z. Shen , Niels N. Sørensen & Jens N. Sørensen

To cite this article: Dmitry K. Kolmogorov , Wen Z. Shen , Niels N. Sørensen & Jens N. Sørensen (2015) Fully Consistent SIMPLE-Like Algorithms on Collocated Grids, Numerical Heat Transfer, Part B: Fundamentals, 67:2, 101-123, DOI: 10.1080/10407790.2014.949583

To link to this article: https://doi.org/10.1080/10407790.2014.949583

Numerical Heat Transfer, Part B, 67: 101–123, 2015

ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790.2014.949583

Taylor & Francis
Taylor & Francis Group

FULLY CONSISTENT SIMPLE-LIKE ALGORITHMS ON COLLOCATED GRIDS

Dmitry K. Kolmogorov, Wen Z. Shen, Niels N. Sørensen, and Jens N. Sørensen

Department of Wind Energy, Technical University of Denmark, Kgs. Lyngby, Denmark

To increase the convergence rate of SIMPLE-like algorithms on collocated grids, a compatibility condition between mass flux interpolation methods and SIMPLE-like algorithms is presented. Results of unsteady flow computations show that the SIMPLEC algorithm, when obeying the compatibility condition, may obtain up to 35% higher convergence rate as compared to the standard SIMPLEC algorithm. Two new interpolation methods, fully compatible with the SIMPLEC algorithm, are presented and compared with some existing interpolation methods, including the standard methods of Choi [9] and Shen et al. [8]. Numerical results show that the time-step dependence of the standard methods may double the total discretization error at steady state. It is furthermore shown that the new methods are independent of time step and relaxation parameter at convergence. One of the new methods is shown to give a higher accuracy than the standard methods.

1. INTRODUCTION

Originally, SIMPLE-like algorithms were developed for application on staggered grids [1, 2, 3], after which they became widely used on both staggered and collocated grids. Contrary to the staggered grid approaches, collocated grid algorithms need mass flux interpolation to cell faces to ensure pressure–velocity coupling [4]. Originally, to preserve the pressure–velocity coupling, Rhie and Chow [5] proposed a momentum interpolation method, where the pressure gradient contributing to the mass flux at the cell faces is calculated directly from the pressure at the cell centers and not interpolated from the pressure gradient at the cell centers. In this way the Rhie-Chow interpolation avoids successfully the decoupling between velocity and pressure. However, the interpolation method possesses a dependency on the relaxation parameter at steady state. The problem with this dependency was independently solved in [6] and [7]. Nevertheless, it was found later, in [8] and [10], that if the Rhie-Chow interpolation is used for unsteady computations, pressure wiggles appear

Received 9 April 2014; accepted 16 June 2014.

Address correspondence to Dmitry K. Kolmogorov, Department of Wind Energy, Technical University of Denmark, Nils Koppels Allé, Building 403, Room 229, 2800, Kgs. Lyngby, Denmark. E-mail: dkol@dtu.dk

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/unhb.

NOMENCLATURE

A_p, A_{nb}	diagonal and nondiagonal matrix terms accounting for discrete convective and diffusion terms
	$A_p^V = \rho dV_p/\tau$
	$\tilde{A}_p = A_p/\alpha + 1.5A_p^V$
	$\tilde{\tilde{A}}_p = (1/\alpha - \gamma)A_p + 1.5A_p^V$
B_p	Rhie-Chow correction term
CD	drag coefficient
D_p	discrete divergence operator computing divergence using linear interpolation
$d\vec{S}_k$	vector normal to cell face k with size equal to the face area
dV_p	control-volume size
f_k	cell-face mass flux at cell face k
$\tilde{f}_k^n$	$= 2f_k^n - 0.5f_k^{n-1}$
I_p	correction term ensuring pressure–velocity coupling on collocated grids
p	pressure
PI	parameterized and independent of time step and relaxation parameter interpolation
PI^0, PI^∞	two types of PI interpolation methods
PI_E^0, PI_E^∞	SIMPLE-compatible forms of PI^0, PI^∞ interpolations
PI_C^0, PI_C^∞	SIMPLEC-compatible forms of PI^0, PI^∞ interpolations
PS	parameterized standard interpolation
PS_E	SIMPLE-compatible form of PS interpolation
PS_C	SIMPLEC-compatible form of PS interpolation
RHS/LHS	right/left hand side of equation
$\vec{S}_p$	source term containing explicitly treated terms of momentum equations
$\vec{\upsilon}$	velocity vector
$\tilde{\vec{\upsilon}}_p^n$	$= 2\vec{\upsilon}_p^n - 0.5\vec{\upsilon}_p^{n-1}$
$\overline{[\ ]}_k$	operator of linear interpolation from cell centers to cell face k
α	velocity underrelaxation parameter
β, γ	parameters used to ensure compatibility of mass-flux interpolation methods with SIMPLE and SIMPLEC algorithms
ρ	density
τ	time step

Subscripts

p	control-volume center
k	cell face
e, w, n, s	compass notations of cell faces around control volume
E, W, N, S	compass notations of cell centers around control volume

Superscripts

c	notation of corrector step
n	time-step counter
m	subiteration counter
*	notation of prediction step

for small time steps. The reason for the pressure wiggles was given in [8]. An interpolation method for unsteady flow computations free from the pressure wiggles was proposed independently by Choi [9] and Shen et al. [8]. However, it was later found that the methods of Choi and Shen possess a weak dependence of time step and relaxation parameter at convergence [11]. To overcome the difficulty, several interpolation methods independent of time step and relaxation parameter were proposed in [11–14]. Nowadays, in spite of the existence of the time-step-independent methods, the standard methods of Choi and Shen are still widely used, as can be seen, for example, in [15–18].

Among the SIMPLE-like algorithms, the SIMPLE and SIMPLEC algorithms are the most widespread algorithms in engineering applications. On collocated grids there is a common practice to apply both algorithms with the same interpolation methods [4, 19]. An exception from the practice is the SIMPLEC algorithm, proposed by Shen et al. [20], where an interpolation method specially adopted for the SIMPLEC algorithm was employed. Unfortunately, in general, the interpolation

method of Shen et al. [20] possesses dependencies of the time step and relaxation parameter at convergence.

In the current article, to ensure high convergence rate of the SIMPLE-like algorithms on collocated grids, a compatibility condition between the mass-flux interpolation methods and the SIMPLE-like algorithms is presented. Two new interpolation methods, which are independent of time step and relaxation parameter, are presented. Contrary to other time-step-independent methods [11–14], the two methods are fully compatible with the SIMPLEC algorithm. Numerical results of unsteady flow computations, presented later in the article, will show that the fully compatible SIMPLEC algorithm ensures a convergence rate which is up to 35% higher than the standard SIMPLEC algorithm.

The two new interpolation methods will be verified together with other time-step-independent methods from the literature [13–14]. The most accurate method will be appointed and recommended for practical use. In addition, the standard interpolation methods of Choi [9] and Shen et al. [8, 20] will be analyzed. For the standard methods, the magnitude of the dependencies of the time step and relaxation parameter will be estimated and the time-step-independence of the new methods at convergence will be verified. The analysis presented in the article will be supported by various results of steady and unsteady flow computations.

The article is organised as follows. First, the momentum and continuity equations are discretized and a parameterized form of some existing interpolation methods is introduced. Second, two new interpolation methods, which are independent of the length of the time step and the size of the relaxation parameter at convergence, are presented. The methods are shown to satisfy a compatibility condition with the SIMPLEC algorithm, enhancing the fast convergence rate of the algorithm. Third, the compatibility condition is presented and discussed in detail. Finally, several computational experiments of steady and unsteady flows are presented.

2. DISCRETIZATION OF THE GOVERNING EQUATIONS

In this section the discretization of the governing equations on collocated grids for the two-dimensional case is presented. First, the momentum equations are discretized. Second, the discrete continuity equation is presented with a parameterized form of some existing mass flux interpolations. Then, two new interpolations which are time-step-independent at convergence and fully compatible with the SIMPLEC algorithm are presented. Finally, the compatibility condition of interpolation methods with SIMPLE-like algorithms is analyzed in detail.

2.1. Momentum Equations

Using a second-order backward difference scheme in time, the system of the momentum equations on collocated grids can be expressed in a control volume p in the following form:

$$\begin{aligned}(A_p + 1.5A_p^V)u_p^{n+1} + \sum_{E,W,N,S} A_{\mathrm{nb}}u_{\mathrm{nb}}^{n+1} + \nabla_x p_p^{n+1}\, dV_p &= A_p^V(2u_p^n - 0.5u_p^{n-1}) + S_p^{x,n} \\ (A_p + 1.5A_p^V)v_p^{n+1} + \sum_{E,W,N,S} A_{\mathrm{nb}}v_{\mathrm{nb}}^{n+1} + \nabla_y p_p^{n+1}\, dV_p &= A_p^V(2v_p^n - 0.5v_p^{n-1}) + S_p^{y,n}\end{aligned} \tag{1}$$

where A_p, A_{nb} are the diagonal and nondiagonal matrix terms accounting for the discrete convective and diffusion terms, A_p^V is the coefficient of the time derivative and equal to $\rho\, dV_p/\tau$, where τ is time step and $\rho\, dV_p$ is control-volume mass. The source terms $S_p^{x,n}$ and $S_p^{y,n}$ contain all explicitly treated terms in the x and y directions, respectively.

For convenience System (1) can be reformulated in the vector form:

$$(A_p + 1.5A_p^V)\vec{v}_p^{n+1} + \sum_{E,W,N,S} A_{nb}\vec{v}_{nb}^{n+1} + \vec{\nabla} p_p^{n+1}\, dV_p = A_p^V(2\vec{v}_p^n - 0.5\vec{v}_p^{n-1}) + \vec{S}_p^n \quad (2)$$

where the velocity vector $\vec{v}_p = (u_p, v_p)^T$ and the source term vector $\vec{S}_p^n = (S_p^{x,n}, S_p^{y,n})^T$.

The velocity $\vec{v}_p$ in Eq. (2) usually has to be underrelaxed. In this article the underrelaxation is applied only to the spatial term $A_p\vec{v}_p^{n+1}$, similarly to [8, 13, 20]. The resulting momentum equations are written as follows:

$$\tilde{A}_p\vec{v}_p^{m+1} + \sum_{E,W,N,S} A_{nb}\vec{v}_{nb}^{m+1} + \vec{\nabla} p_p^{m+1}\, dV_p = A_p^V(2\vec{v}_p^n - 0.5\vec{v}_p^{n-1}) + \vec{S}_p^m + (1/\alpha - 1)A_p\vec{v}_p^m \quad (3)$$

where $\tilde{A}_p$ is defined by

$$\tilde{A}_p = A_p/\alpha + 1.5A_p^V \quad (4)$$

and α is the velocity underelaxation parameter, m is the subiteration counter, such that the solution at time step $n+1$ is obtained at convergence. In Eq. (3), to compute the flow field at subiteration step $m+1$, the coefficients A_p, A_{nb} and $\tilde{A}_p$ are taken from the former subiteration step m. In the notations of the coefficients, the superscript counter m is dropped for simplicity.

2.2. Discretization of the Continuity Equation: Standard and New Approaches

The discrete continuity equation essentially represents the fact that the sum of mass fluxes through the control-volume faces equals zero:

$$\sum_{k=e,w,n,s} f_k^{m+1} = 0 \quad (5)$$

where the subscript k corresponds to control-volume face positions. On collocated grids the cell-face mass fluxes are not typically available. To identify the fluxes, one of the widely used methods is the momentum interpolation method originally proposed by Rhie and Chow [5].

2.2.1. Standard momentum interpolation methods in a parameterized form. In the literature there exist many modifications of the Rhie-Chow interpolation, among which we consider two methods, proposed by Shen et al. in [8, 20]. Contrary to the other popular interpolation of Choi in [9] for unsteady flow computations, the two methods possess some advantages as described below.

The two methods of Shen can be formulated in a unified parameterized form. For this, one should first subtract a term $\gamma A_p \vec{v}_p^{m+1}$ from the left- and the right-hand sides of Eq. (3), then add and subtract a term $\beta A_p \vec{v}_p^{m}$ from the right-hand side of Eq. (3), where γ and β are constant parameters described below. After these manipulations the cell-centered velocity can be expressed as

$$\vec{v}_p^{m+1} = \frac{A_p^V}{\tilde{\tilde{A}}_p}(2\vec{v}_p^{n} - 0.5\vec{v}_p^{n-1}) + (1/\alpha - 1 - \beta)\frac{A_p}{\tilde{\tilde{A}}_p}\vec{v}_p^{m} + \frac{\vec{h}_p^{m+1}}{\tilde{\tilde{A}}_p} - \vec{\nabla} p_p^{m+1}\frac{dV_p}{\tilde{\tilde{A}}_p} \tag{6}$$

where the term $\tilde{\tilde{A}}_p$ and vector $\vec{h}_p^{m+1}$ are

$$\tilde{\tilde{A}}_p = (1/\alpha - \gamma)A_p + 1.5A_p^V \tag{7}$$

$$\vec{h}_p^{m+1} = -\sum_{E,W,N,S} A_{\mathrm{nb}}\vec{v}_{\mathrm{nb}}^{m+1} - \gamma A_p \vec{v}_p^{m+1} + \beta A_p \vec{v}_p^{m} + \vec{S}_p^{m} \tag{8}$$

Then, applying interpolation similar to the one described in [8], the following expression for the mass flux at some cell face k is obtained:

$$f_k^{m+1} = \overline{\left[\frac{A^V}{\tilde{\tilde{A}}}\right]}_k \tilde{f}_k^{n} + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{\tilde{\tilde{A}}}\right]}_k f_k^{m} + \overline{\left[\frac{\vec{h}^{m+1}}{\tilde{\tilde{A}}}\right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{dV}{\tilde{\tilde{A}}}\right]}_k \bar{\nabla} p_k^{m+1} \cdot d\vec{S}_k \tag{9}$$

where the overbar $\overline{[\,]}_k$ denotes linear interpolation from cell centers to cell face k.[1] The vector $d\vec{S}_k$ indicates the vector normal to cell face k with length equal to the face area. The flux $\tilde{f}_k^{n}$ is equal to $2f_k^{n} - 0.5f_k^{n-1}$, and the fluxes f_k^{n}, f_k^{n-1}, and f_k^{m} are the fluxes available from former time steps n, $n-1$ and subiteration m, respectively. To ensure pressure–velocity coupling on collocated grids, the pressure force term $\vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k$ at cell face k has to be discretized based on $1\,\delta$ pressure difference as in the original Rhie-Chow method [5].

The interpolation method defined in Eqs. (7)–(9) will be called in this article the *parameterized standard* (PS) interpolation. Using the parameters γ and β, the PS method may be used in two forms which correspond to two interpolation methods in the literature:

1. For $\gamma = 0$ and $\beta = 0$, the PS method exactly corresponds to the first method of Shen et al. in [8]. In its turn, Shen's method is similar to the widely used method of Choi [9], but contrary to Choi's method it is based on the second-order backward difference scheme in time.
2. For $\gamma = 1$ and $\beta > 0$, the PS method exactly corresponds to another method of Shen et al. in [20]. The method is rarely met in literature, but as shown later in this article it has several advantages over the first method. According to results shown later in the article, the PS method with $\gamma = 1$ and $\beta = 0.04$ enhances the convergence rate of the SIMPLEC algorithm up to 35%, compared to the PS method in the first form above.

[1]For example, on a uniform grid $\overline{[\phi]}_e = (\phi_E + \phi_P)/2$, where ϕ is a representative variable.

According to Section 2.3.3, the PS method in form 1 and form 2 above is fully compatible with the SIMPLE and SIMPLEC algorithms, respectively. Therefore, the first and the second forms of the PS method will be further referred to as the PS_E and PS_C methods, respectively.[2]

Using the PS_E method, accurate solution on collocated grids can be obtained and the pressure–velocity coupling is ensured for any time step τ, including the infinitely small τ values [8]. The method, in fact, has the disadvantage that at convergence to steady state the corresponding fluxes become weakly dependent on τ and α [10–14]. To get rid of the time-step dependence, several modifications of the PS_E method were developed in [11–14].

The dependence of τ and α in the PS_E method arises from the interpolation of the $\tilde{\tilde{\boldsymbol{A}}}_p$ term, which contains both τ and α at convergence. Indeed, at convergence the cell-centered velocities and cell-face mass fluxes become constant, and Eq. (9) for the PS_E method becomes

$$f_k = \frac{\overline{\left[\vec{h}/\tilde{\tilde{\boldsymbol{A}}}\right]}_k \cdot d\vec{S}_k - \overline{[dV/\tilde{\tilde{\boldsymbol{A}}}]}_k \vec{\nabla} p_k \cdot d\vec{S}_k}{\overline{\left[A/\tilde{\tilde{\boldsymbol{A}}}\right]}_k} \tag{10}$$

where f_k denotes the mass flux at convergence. In notations of the variables in Eq. (10), the superscripts m and n are dropped due to convergence to the steady-state. In Eq. (10) the dependencies of τ and α are maintained in the $\tilde{\tilde{\boldsymbol{A}}}_p$ term, as seen in Eq. (7).

Similarly to the PS_E method, the PS_C method is also τ- and α-dependent at convergence. But contrary to the PS_E method, it may result in steady-state solution independent of τ in the special case where underrelaxation is not applied. Indeed, in this case the term $\tilde{\tilde{\boldsymbol{A}}}$ becomes equal to $1.5\rho\, dV_p/\tau$ and the steady-state flux in Eq. (9) becomes independent of τ:

$$f_k = \frac{\overline{[\vec{h}/(\rho\, dV)]}_k \cdot d\vec{S}_k - \frac{1}{\rho}\vec{\nabla} p_k \cdot d\vec{S}_k}{\beta\overline{[A/(\rho\, dV)]}_k} \tag{11}$$

The most important advantage of the PS_C method over the PS_E method is satisfaction of the *compatibility condition* with the SIMPLEC algorithm. By satisfying the condition (explained in detail in Section 2.3 and verified in Section 3.1), the convergence rate of the SIMPLEC algorithm may be increased up to 35%. In spite of the advantage in speed, there still exist a few unresolved issues relating to the PS_c method.

First, the method has the mass fluxes at convergence dependent on the additional parameter β, which is employed to ensure the *finiteness* of the fluxes at convergence, as can be seen from Eq. (11). The dependence opens a question: "What is the optimum value of β, ensuring robustness, good accuracy, and high convergence rate of the PS_C method?" The question was partially answered in the work of Shen et al. [20], whereas in this article an exhaustive analysis for the choice of the β will be performed.

[2]For example, in the notation of the PS_E method, the subscript E refers to the last letter of the word "SIMPLE."

Second, it should be noted that in real-life applications, applying under-relaxation with $\alpha < 1$ may become necessary, as the matrix of unsteady momentum equations is diagonally dominant only at small time steps. Moreover, if highly skewed grids are employed, the velocity underrelaxation becomes unavoidable and the PS_C method becomes dependent on both the time step τ and the relaxation parameter α.

Finally, as was discussed above, for both the PS_E and PS_C methods, weak dependencies of time step and relaxation parameter exist. Today, the magnitude of the dependencies in real-life applications is still unknown. In the current article we will perform a deliberate analysis of the dependencies and present two new time-step-independent interpolation methods, which modify the original PS_C method.

2.2.2. On the role of the β parameter in the PS_C method. The PS_C method discussed in the previous section involves the use of an additional positive parameter β to ensure the finiteness of the fluxes at steady-state [see Eq. (11)]. For the PS_C method the requirement for the β parameter to be positive is important, as only in this case the pressure–velocity coupling can be ensured. To see it, one may first rewrite the momentum equations, Eq. (3), at steady-state as

$$-\sum A_{\mathrm{nb}}\vec{\upsilon}_{\mathrm{nb}} - A_p\vec{\upsilon}_p = \vec{\nabla}p_p\, dV_p \tag{12}$$

where the source term $\vec{S}_p$ was dropped for simplicity. Then, using Eq. (8) and Eq. (12), the expression of the steady-state flux in the PS_C method can be rewritten as

$$f_k = \frac{1}{\beta}\frac{-\left[\overline{\left[\frac{dV}{\tilde{\vec{A}}}\right]}_k \vec{\nabla}p_k \cdot d\vec{S}_k - \overline{\left[\frac{dV}{\tilde{\vec{A}}}\vec{\nabla}p\right]}_k \cdot d\vec{S}_k\right]}{\overline{\left[A/\tilde{\vec{A}}\right]}_k} + \frac{\overline{\left[\frac{A}{\tilde{\vec{A}}}\vec{\upsilon}\right]}_k}{\overline{\left[A/\tilde{\vec{A}}\right]}_k}\cdot d\vec{S}_k \tag{13}$$

The meaning of the β parameter can be seen clearly if we approximate Eq. (13) as

$$f_k \approx -\frac{1}{\beta}\overline{[dV/A]}_k\left[\vec{\nabla}p_k - \overline{[\vec{\nabla}p]}_k\right]\cdot d\vec{S}_k + \overline{[\vec{\upsilon}]}_k \cdot d\vec{S}_k \tag{14}$$

Remark, that the term $\overline{[dV/A]}_k\,[\vec{\nabla}p_k - \overline{[\vec{\nabla}p]}_k]\cdot d\vec{S}_k$ in Eq. (14) is the widely known Rhie-Chow correction term preventing the solution from experiencing check-board pressure distributions on collocated grids. Therefore, in Eq. (14), the inverse of the parameter β is seen to play the role of ensuring the pressure–velocity coupling on collocated grids.

The choice of β, which ensures both the finiteness of mass fluxes at convergence and the pressure–velocity coupling on collocated grids, will be presented later, in the results section.

2.2.3. Two new momentum interpolation methods independent of τ and α. Most of the existing interpolation methods, which result in solutions independent of the time step τ and the relaxation parameter α, are constructed based on the interpolation method of Choi or Shen et al. [8–9], referred to as the PS_E

method in this article. The unified form of the PS_E and the PS_C methods [see Eq. (9)] will be used below to develop two new modifications of the PS_C method, which are free from the dependence of both time step τ and the relaxation parameter α at convergence.

The first new interpolation method is obtained from Eq. (9) by using the following approximation:

$$\overline{\left[\frac{\phi}{\tilde{\tilde{A}}}\right]}_k \Leftarrow \overline{\left[\frac{\phi}{A^V}\right]}_k \frac{1}{[1.5 + (1/\alpha - \gamma)\overline{[A/A^V]}_k]}$$

where ϕ is a representative variable. By using the approximation in the expression of the mass flux in Eq. (9), a new method is obtained:

$$f_k^{m+1} = \chi_k[\tilde{f}_k^n - \frac{\tau}{\rho}\vec{\nabla}p_k^{m+1} \cdot d\vec{S}_k + \overline{\left[\frac{\vec{h}^{m+1}}{A^V}\right]}_k \cdot d\vec{S}_k + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{A^V}\right]}_k f_k^m] \quad (15)$$

where χ_k is defined as

$$\chi_k = 1/[1.5 + (1/\alpha - \gamma)\overline{[A/A^V]}_k] \quad (16)$$

The interpolation method, Eq. (15), is independent of time step and relaxation parameter at convergence (see Section 2.2.4 for the proof). It is formulated in the parameterized form using constant parameters γ and β, therefore the interpolation, Eq. (15), will be further referred to as the *parameterized* and *independent* of both time step and relaxation parameter (PI^0) method. The superscript 0 in the method notation is used due to the reasons explained in the next section. Before the PI^0 method is analyzed, a second new interpolation method is introduced below.

The second new interpolation method is obtained from Eq. (9) by using the following approximation:

$$\overline{\left[\frac{\phi}{\tilde{\tilde{A}}}\right]}_k \Leftarrow \overline{\left[\frac{\phi}{A}\right]}_k \frac{1}{[1.5\overline{[A^V/A]}_k + (1/\alpha - \gamma)]}$$

where ϕ is a representative variable. Using the approximation above, Eq. (9) is transformed into the equation for the second new interpolation method:

$$f_k^{m+1} = \psi_k(\overline{\left[\frac{A^V}{A}\right]}_k \tilde{f}_k^n - \overline{\left[\frac{dV}{A}\right]}_k \vec{\nabla}p_k^{m+1} \cdot d\vec{S}_k + \overline{\left[\frac{\vec{h}^{m+1}}{A}\right]}_k \cdot d\vec{S}_k + (1/\alpha - 1 - \beta)f_k^m) \quad (17)$$

where ψ_k is defined as

$$\psi_k = 1/[1.5\overline{[A^V/A]}_k + (1/\alpha - \gamma)] \quad (18)$$

The interpolation method, Eq. (17), is also independent of time step and relaxation parameter at convergence. It will be further referred to as the *parameterized* and *independent* of time step and relaxation parameter (PI^{∞}) method, where the superscript ∞ in the method notation is used due to the reasons explained in the next section.

Similarly to the PS method (see Section 2.2.1), each of the PI^0 and the PI^{∞} methods above has two forms, namely, the form with $\gamma = 0$, $\beta = 0$ and the form with $\gamma = 1$, $\beta > 0$, as described below

1. For $\gamma = 0$, and $\beta = 0$, the PI methods (PI^0 and PI^{∞}) are similar to two interpolation methods of Pascau (see PICTURETWO and PICTURE in [14], respectively), but contrarily to Pascau's methods, the PI methods are based on second-order backward difference in time. In its turn, the PICTURE method of Pascau is similar to Cubero's interpolation in [13].
2. For $\gamma = 1$ and $\beta > 0$, the PI methods are new interpolation methods. As will be seen later in the article, contrary to other time-step-independent methods, the PI methods in this form are fully compatible with the SIMPLEC algorithm. According to the results presented later in Section 3.1.3, the PI^0 method with $\gamma = 1$ and $\beta = 0.04$ may enhance the convergence rate of the SIMPLEC algorithm up to 25%, compared to the time-step-independent methods existing in literature.

According to Section 2.3.3, the PI methods in form 1 and form 2 above are fully compatible with the SIMPLE and SIMPLEC algorithms, respectively. Therefore, the PI methods in the first and the second forms will be further referred to as the PI_E and PI_C methods, respectively.

In conclusion, the notations for the new and the standard interpolations discussed in the previous sections are shown in Table 1 and Table 2.

2.2.4. Comparison between the PI^0 and PI^{∞} methods at steady-state. The steady-state solutions of the discrete continuity equation, Eq. (5), based on the mass fluxes interpolated using either the PI^0 or PI^{∞} method, are independent of both time step τ and velocity relaxation parameter α.

Indeed, taking the condition that the flow reached the steady-state, the flux for the PI^0 method, Eq. (15), becomes

$$f_k = \frac{\overline{[\vec{h}/(\rho dV)]}_k \cdot d\vec{S}_k - \frac{1}{\rho}\vec{\nabla}p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)\overline{[A/(\rho dV)]}_k} \tag{19}$$

Table 1. Notations for the SIMPLE-compatible forms of PS, PI^0, and PI^{∞} methods

Method	Equation number	Parameter values	Method notation	Similar methods in literature
PS	Equation (9)	$\gamma = 0$, $\beta = 0$	PS_E	Shen [8], Choi [9]
PI^0	Equations (15)–(16)	$\gamma = 0$, $\beta = 0$	PI^0_E	Pascau [14] (PICTURETWO)
PI^{∞}	Equations (17)–(18)	$\gamma = 0$, $\beta = 0$	PI^{∞}_E	Pascau [14] (PICTURE), Cubero [13]

Table 2. Notations for the SIMPLEC-compatible forms of PS, PI^0, and PI^∞ methods

Method	Equation number	Parameter values	Method notation	Similar methods in literature (if existing)
PS	Equation (9)	$\gamma = 1,\ \beta > 0$	PS_C	Shen [20]
PI^0	Equations (15)–(16)	$\gamma = 1,\ \beta > 0$	PI^0_C	
PI^∞	Equations (17)–(18)	$\gamma = 1,\ \beta > 0$	PI^∞_C	

whereas the flux for the PI^∞ method, Eq. (17), is equal to

$$f_k = \frac{\overline{[\vec{h}/A]}_k \cdot d\vec{S}_k - \overline{[dV/A]}_k \vec{\nabla} p_k \cdot d\vec{S}_k}{(1 - \gamma + \beta)} \tag{20}$$

As seen, in Eqs. (19) and (20) the fluxes are independent of both τ and α.

By comparing the PS method, Eq. (9), and the PI methods, Eqs. (15)–(18), the main result of the article may be formulated:

- The fluxes of the PI^0 method at steady-state equal the fluxes of the PS method at steady-state, if the PS method is employed with an *infinitely small* time step, $\tau \to 0$.
- The fluxes of the PI^∞ method at steady-state equal the fluxes of the PS method at steady-state, if the PS method is employed with an *infinitely large* time step, $\tau \to \infty$.

To prove the statements above, one need to consider first the mass fluxes of the PS method, Eq. (9), at steady-state, i.e., when the conditions $n+1 = n = n-1$ and $m+1 = m$ are fulfilled:

$$f_k = \frac{\overline{\left[\vec{h}/\tilde{\tilde{A}}\right]}_k \cdot d\vec{S}_k - \overline{\left[dV/\tilde{\tilde{A}}\right]}_k \vec{\nabla} p_k \cdot d\vec{S}_k}{(1 - \gamma + \beta)\overline{\left[A/\tilde{\tilde{A}}\right]}_k} \tag{21}$$

When $\tau \to 0$, the fluxes, Eq. (21), become equal to the steady-state fluxes of the PI^0 method in Eq. (19). Alternatively, when, the fluxes, Eq. (21), become equal to the steady-state fluxes of the PI^∞ method in Eq. (20).

Using the two statements and knowing that discretization errors drop down at decreasing time steps, we can make the following conclusion:

- The PI^0 method is expected to give higher accuracy than both the PS and the PI^∞ methods.

In support of the statements above, the results in Section 3.2 will show the superiority of the PI^0 method in comparison to both the PI^∞ and the PS methods.

2.3. Compatibility Between Interpolation Methods and SIMPLE-Like Algorithms

In this section, interpolation methods on collocated grids will be classified as being either 1) fully compatible with the SIMPLE algorithm or (2) fully compatible

with the SIMPLEC algorithm. To the authors' knowledge, the compatibility condition, presented below, has not been presented in the literature. We will show later that the SIMPLE-like algorithms, which satisfy the condition, have a faster convergence rate than those not satisfying this condition.

2.3.1. System of Navier-Stokes equations on collocated grids. In this section, the system of the Navier-Stokes equations composed of the momentum equation, Eq. (3), and the continuity equation, Eq. (5), will be reformulated. As an example of momentum interpolation methods, the parameterized standard (PS) interpolation, Eq. (9), will be employed to solve the continuity equation and derive the compatibility condition between the SIMPLE-like algorithms and the momentum interpolations on collocated grids.

First, grouping the explicit terms of Eq. (3) into $\vec{b}_p^m$, the compact form of the momentum equations, Eq. (3), is obtained as

$$\tilde{A}_p \vec{v}_p^{m+1} + \sum_{E,W,N,S} A_{\mathrm{nb}} \vec{v}_{\mathrm{nb}}^{m+1} + \vec{\nabla} p_p^{m+1}\, dV_p = \vec{b}_p^m \tag{22}$$

Second, the discrete continuity equation, Eq. (5), is reformulated. For this, a discrete divergence operator D_p is introduced, which computes the divergence at some control-volume p by linear velocity interpolation:

$$D_p(\vec{v}) = \sum_{k=e,w,n,s} \rho \overline{[\vec{v}]}_k \cdot d\vec{S}_k$$

where $\overline{[\ ]}_k$ is linear interpolation from the cell centers to cell face k. By adding and subtracting $D_p(\vec{v}^{m+1})$ from the discrete continuity equation, Eq. (5), the following discrete continuity equation is obtained:

$$D_p(\vec{v}^{m+1}) + \underbrace{\sum_{k=e,w,n,s} \left[f_k^{m+1} - \rho \overline{[\vec{v}^{m+1}]}_k \cdot d\vec{S}_k \right]}_{I_p} = 0 \tag{23}$$

where the second term in Eq. (23) is denoted as I_p. A concrete form of the term I_p depends on the interpolation method used for determining f_k^{m+1}. Independently of the interpolation method, I_p can be referred to a correction term, ensuring the pressure–velocity coupling on collocated grids. To see this, the PS method, Eq. (9), will be used below.

Using the fluxes of the PS method and the cell-centered velocity, Eq. (6), the term I_p in Eq. (23) can be rearranged as

$$I_p = B_p(p^{m+1}) - d_p^m \tag{24}$$

where the pressure terms are grouped into $B_p(p)$,

$$B_p(p) = \sum_{k=e,w,n,s} \left[\overline{\left[\frac{dV}{\tilde{\vec{A}}} \vec{\nabla} p \right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{dV}{\tilde{\vec{A}}} \right]}_k \vec{\nabla} p_k \cdot d\vec{S}_k \right] \tag{25}$$

and the remaining terms are arranged in d_p^m,

$$d_p^m = \sum_{k=e,w,n,s} \left[\left[\overline{\left[\frac{A^V}{\tilde{\tilde{A}}} \vec{\tilde{\upsilon}}^n \right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{A^V}{\tilde{\tilde{A}}} \right]_k} \tilde{f}_k^n \right] \right] + (1/\alpha - 1 - \beta) \sum_{k=e,w,n,s} \left[\overline{\left[\frac{A}{\tilde{\tilde{A}}} \vec{\upsilon}^m \right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{A}{\tilde{\tilde{A}}} \right]_k} f_k^m \right]$$

where the velocity vector $\vec{\tilde{\upsilon}}^n$ denotes $2\vec{\upsilon}^n - 0.5\vec{\upsilon}^{n-1}$.

The momentum equation, Eq. (22), and the continuity equation, Eq. (23), rearranged using Eq. (24), formulate the system of the Navier-Stokes equations on collocated grids as below:

$$\begin{cases} \tilde{A}_p \vec{\upsilon}_p^{m+1} + \sum\limits_{E,W,N,S} A_{\mathrm{nb}} \vec{\upsilon}_{\mathrm{nb}}^{m+1} + \vec{\nabla} p_p^{m+1} dV_p & = \vec{b}_p^m \\ D_p(\vec{\upsilon}^{m+1}) + B_p(p^{m+1}) & = d_p^m \end{cases} \tag{26}$$

It can be noted that the system of equations, Eqs. (26), is the general system of the Navier-Stokes equations on collocated grids, which may be solved by various iterative methods, such as fractional step, pressure-correction, or other methods. In the system, the term $B_p(p)$ is the Rhie-Chow correction term ensuring pressure–velocity coupling on collocated grids, whereas the term d_p^m is the term ensuring pressure–velocity coupling at small time steps.

2.3.2. Prediction-correction form of the system of Navier-Stokes equations. With the system of the reformulated Navier-Stokes equations, Eqs. (26), we may show that to achieve high convergence rate of SIMPLE-like algorithms on collocated grids, a proper choice of interpolation methods has to be made. For this, a prediction-correction form of the Navier-Stokes equations is obtained using the splitting of the velocity and the pressure into two parts, namely, a prediction and a correction part, as follows:

$$\begin{bmatrix} \vec{\upsilon}^{m+1} \\ p^{m+1} \end{bmatrix} = \begin{bmatrix} \vec{\upsilon}^* \\ p^m \end{bmatrix} + \begin{bmatrix} \vec{\upsilon}^c \\ p^c \end{bmatrix} \tag{27}$$

Using Eq. (27), the system of equations, Eqs. (26), is solved in two steps. At the first step the momentum equations in Eqs. (26) are solved[3] for a prediction velocity $\vec{\upsilon}_p^*$ using the pressure p^m available from the previous subiteration m:

$$\tilde{A}_p \vec{\upsilon}_p^* + \sum_{E,W,N,S} A_{\mathrm{nb}} \vec{\upsilon}_{\mathrm{nb}}^* + \vec{\nabla} p_p^m dV_p = b_p^m \tag{28}$$

[3]It can be noted that Eq. (28) is composed of two equations in 2D (or three in 3D) and may be solved for each of the velocity components separately using iterative methods such as Gauss-Seidel, Jacobi, etc.

To obtain an equation for the second step, Eq. (28) is subtracted from Eqs. (26) and the following system for the velocity and pressure corrections is obtained:

$$\begin{cases} \tilde{A}_p \vec{\upsilon}_p^c + \sum\limits_{E,W,N,S} A_{\mathrm{nb}} \vec{\upsilon}_{\mathrm{nb}}^c + \vec{\nabla} p_p^c dV_p & = 0 \\ D_p(\vec{\upsilon}^c) \quad + B_p(p^c) & = \sum\limits_{k=e,w,n,s} f_k^* \end{cases} \tag{29}$$

The prediction part of the continuity equation above was expressed through the mass flux predictions f_k^*, which are determined by the prediction form of Eq. (9):

$$f_k^* = \overline{\left[\frac{A^V}{\tilde{\tilde{A}}}\right]_k} \tilde{f}_k^n + (1/\alpha - 1 - \beta) \overline{\left[\frac{A}{\tilde{\tilde{A}}}\right]_k} f_k^m + \overline{\left[\frac{\vec{h}^*}{\tilde{\tilde{A}}}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{dV}{\tilde{\tilde{A}}}\right]_k} \bar{\nabla} p_k^m \cdot d\vec{S}_k \tag{30}$$

where the prediction term $\vec{h}^*$ is defined as:

$$\vec{h}_p^* = - \sum_{E,W,N,S} A_{\mathrm{nb}} \vec{\upsilon}_{\mathrm{nb}}^* - \gamma A_p \vec{\upsilon}_p^* + \beta A_p \vec{\upsilon}_p^m + \vec{S}_p^m \tag{31}$$

One may verify that the discrete continuity equation in Eqs. (29) is equivalent to the original continuity equation in Eqs. (26). To solve the system of equations, Eqs. (29), a pressure-correction equation has to be obtained in a way as described in the following section.

2.3.3. Pressure-correction equations on collocated grids. It is straightforward to obtain an equation for the pressure correction by expressing the velocity correction from the first equation of the system of equations, Eqs. (29), and then substituting the velocity in the second equation of the system. However, the resulting matrix of pressure-correction equations has a rather complex structure. Below, we will use an alternative approach, similar to the SIMPLE and the SIMPLEC algorithms, where the first equation in the system of equations, Eqs. (29), is replaced by the following simplified equation:

$$L_p \vec{\upsilon}_p^c + \vec{\nabla} p_p^c \, dV_p = 0 \tag{32}$$

where $L_p = \tilde{A}_p$ for the SIMPLE [1] algorithm and $L_p = \tilde{A}_p - A_p$ for the SIMPLEC [3] algorithm.

In spite of the wide spread assumption that the SIMPLE and SIMPLEC algorithms differ only by the L_p [4], [19], we will show below that on collocated grids different interpolation methods have to be employed with the SIMPLE and SIMPLEC algorithms to ensure high efficiency of the algorithms.

By expressing the velocity correction, $\vec{\upsilon}_p^c$, from Eq. (32) and then substituting it in the second equation of the system of equations, Eqs. (29), we obtain the following pressure-correction equation:

$$- \sum_{k=e,w,n,s} \overline{\left[dV/\tilde{\tilde{A}}\right]_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_p((\tilde{\tilde{A}}^{-1} - L^{-1}) \vec{\nabla} p^c \, dV)}_{\text{Compatibility term}} = \sum_{k=e,w,n,s} f_k^* \tag{33}$$

The equation above is the final pressure-correction equation, which is seen to be different from the widely used pressure-correction equations in the literature due to the existence of the second term in the equation. It can be also noted that Eq. (33) was derived for the standard PS interpolation method, whereas the pressure-correction equations for the new PI^0 and PI^∞ interpolations can be derived similarly. Omitting the intermediate steps, these final equations are given below.

The pressure-correction equation based on the PI^0 interpolation has the following form:

$$-\tau/\rho \sum_{k=e,w,n,s} \chi_k \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_p((\tilde{\tilde{A}}^{-1} - L^{-1})\vec{\nabla} p^c \, dV)}_{\text{Compatibility term}} = \sum_{k=e,w,n,s} f_k^* \tag{34}$$

where the prediction mass fluxes f_k^* are defined by the prediction form of Eq. (15).

Alternatively, the pressure-correction equation based on the PI^∞ interpolation is

$$-\sum_{k=e,w,n,s} \psi_k \overline{[dV/A]}_k \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_p((\tilde{\tilde{A}}^{-1} - L^{-1})\vec{\nabla} p^c \, dV)}_{\text{Compatibility term}} = \sum_{k=e,w,n,s} f_k^* \tag{35}$$

where the mass flux predictions f_k^* are determined by the prediction form of Eq. (17).

The second terms in the pressure-correction equations, Eqs. (33)–(35), are not equal to zero in general. A specific condition when these terms are equal to zero will be discussed in the next section.

2.3.4. Compatibility condition on collocated grids. The pressure-correction equations, Eqs. (33)–(35), have identical second terms. These terms, referred to as compatibility terms, are zero if the following condition is fulfilled:

$$L_p^{-1} - \tilde{\tilde{A}}_p^{-1} = 0 \tag{36}$$

As shown below, fulfilling the condition in Eq. (36) may enhance the convergence rate of the SIMPLE-like algorithms.

One may consider different interpolation methods with the SIMPLE or SIMPLEC algorithm. If Eq. (36) is fulfilled for some interpolation with the SIMPLE algorithm, then the compatibility term in the corresponding pressure-correction equation becomes zero. Therefore, in this article such interpolation is referred to as interpolation fully compatible with the SIMPLE algorithm (or SIMPLE-compatible interpolation). Alternatively, if Eq. (36) is fulfilled for the SIMPLEC algorithm, then the interpolation is referred to as fully compatible with the SIMPLEC algorithm (or SIMPLEC-compatible interpolation).

Using the compatibility condition, Eq. (36), it can be verified that the PI_E, PI_E^0,PI_E^∞ interpolations (see Table 1 in Section 2.2.3) and the similar interpolations of Choi [9], Pascau [14], and Cubero [13] are SIMPLE-compatible. The original Rhie-Chow interpolation, and lots of its modifications as presented in [6–7, 10–12, 21–24], are also SIMPLE-compatible. However, in the literature these methods were applied in both the SIMPLE and SIMPLEC algorithms, regardless of the fact that the

interpolations do not satisfy the compatibility condition with the SIMPLEC algorithm. To see this, one may consider the PI_E interpolation, which is a second-order alternative of the widespread interpolation of Choi (see Section 2.2.1). Employing the PS_E interpolation with the SIMPLEC algorithm results in the pressure-correction equation, Eq. (33), with the compatibility term not equal to zero. In this case, Eq. (33) can be solved using several possible approaches.

1. The first approach is to solve the equation exactly. This requires using a solver which can handle matrixes with rather complex structures.
2. The second approach is to treat the compatibility term explicitly and solve the equation iteratively. However, such an approach can be computationally expensive.
3. The third approach is the widely used approach [4, 19], where, instead of Eq. (33), the following equation is solved:

$$-\sum_{k=e,w,n,s} \overline{[dV/L]}_k \vec{\nabla} p'_k \cdot d\vec{S}_k = \sum_{k=e,w,n,s} f^*_k \tag{37}$$

which is a typical pressure-correction equation for the SIMPLEC algorithm on collocated grids [4, 19, 23, 24]. Nevertheless, it can be noted that Eq. (37) approximates the original pressure-correction equation, Eq. (33), which may consequently make the convergence rate of the SIMPLEC algorithm slower.

Instead of solving Eq. (37), a proper choice of interpolation method for the SIMPLEC algorithm may be used to simplify the corresponding pressure-correction equation. Using the compatibility condition in Eq. (36), it can be verified that each of the PS_C, PI^0_C, and PI^∞_C interpolations (see Table 2 in Section 2.2.3) is SIMPLEC-compatible. For the SIMPLEC algorithm employed with any of these interpolations, the compatibility term in the corresponding pressure-correction equation becomes *inherently* zero. Therefore, no approximation of the pressure-correction equation is required, compared to the approximations needed to obtain the standard pressure-correction equation, Eq. (37). In the results section below, the SIMPLEC algorithm, which is based on the SIMPLEC-compatible interpolations, is shown to possess a convergence rate up to 35% higher compared to the standard SIMPLEC algorithm.

3. NUMERICAL RESULTS

The result-section consists of two parts. First, it is shown that appropriate choice of interpolation method for the SIMPLEC algorithm may increase the convergence rate of the algorithm. Second, the magnitude of time-step dependence of some standard interpolations is estimated and the new SIMPLEC-compatible interpolations, independent of time step and relaxation parameter at convergence, are examined.

The convergence criterion in the computations presented below is for the residual to be less then 10^{-8} for steady computations and less then 10^{-6} for unsteady computations. A velocity underrelaxation parameter of $\alpha = 0.9$ is used for the tests.

3.1. On Compatibility of SIMPLEC Algorithm with Interpolation Methods

In this section, several interpolation methods, which are fully compatible with the SIMPLEC algorithm, will be compared with some standard interpolations. First, as the SIMPLEC-compatible interpolations depend on some positive parameter β (see Table 2 in Section 2.2.3), an optimal value of the parameter will be identified to ensure the robustness of the methods. Second, it will be shown that in unsteady flow computations the SIMPLEC-compatible interpolations enhance the convergence rate of the SIMPLEC algorithm.

Three test cases, lid-driven cavity flows, the Taylor-Green vortex problem, and the roll-up of a shear layer, will be studied. To simplify notation, different SIMPLEC algorithms will be identified by the interpolation methods, which the algorithms are based on. For instance, the SIMPLEC algorithm with the PS_C interpolation will be simply designated as PS_C.

3.1.1. Lid-driven cavity flow. Flow in a lid-driven cavity at $Re = 1000$ is computed on a series of nonuniform grids with different resolutions of from 64^2 to 256^2 cells. The computational effort for computing the flow field is measured in number of multigrid cycles needed for solving the pressure-correction equation. The SIMPLEC-compatible PS_C method (see Table 2 in Section 2.2.3), is used to identify the optimal value of the β parameter. For this, the PS_C method is compared to the PS_E method (see Table 1 in Section 2.2.3), which is independent of β. A work ratio of the methods is computed as $Work^C/Work^E$, where $Work^C$ and $Work^E$ are the workloads of the PS_C and the PS_E methods, respectively. The work ratio is plotted for various grid resolutions in Figure 1a, where it is seen that the PS_C method with β value of [0.04, 0.06] has up to 40% lower workload. To compare the errors of the methods, the minimum vertical velocity along the horizontal line across the center

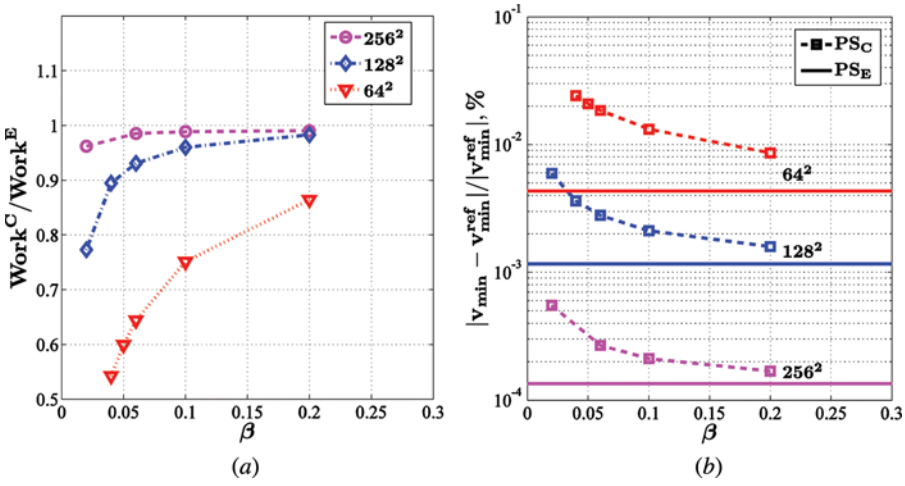

Figure 1. Comparison of workload and relative errors for the PS_C and PS_E methods in a lid-driven cavity flow at $Re = 1{,}000$: (*a*) workload ratio; (*b*) error.

of the cavity is measured and the errors are computed against the reference solution in [25]. As seen in Figure 1b, the SIMPLEC-compatible method gives slightly higher errors than the PS_E method. However, the PS_C method becomes advantageous with respect to both speed and accuracy when it is applied for unsteady flow computations, as will be shown in the following sections.

3.1.2. Taylor-Green vortex. The Taylor-Green vortex is a test case with an exact solution of the Navier-Stokes equations:

$$\begin{aligned} u(x,y,t) &= -\sin(\pi x)\cos(\pi y)e^{-2\pi^2 t/\mathrm{Re}} \\ v(x,y,t) &= \cos(\pi x)\sin(\pi y)e^{-2\pi^2 t/\mathrm{Re}} \\ p(x,y,t) &= 0.25[\cos(2\pi x)+\sin(2\pi y)]e^{-4\pi^2 t/\mathrm{Re}} \end{aligned} \tag{38}$$

The computational domain consists of a square of $[-\pi, \pi]$ with periodic boundary conditions in both directions. The PS_C and the PS_E methods are used to compute the flow field at $\mathrm{Re}=10$. The time integration is performed from $t=0$ to $t=10$. At time $t=10$ the total kinetic energy in the domain has dropped to about 2% of the energy in the initial state. Equidistant grids with different resolutions of from 8×8 to 256×256 cells are used, and the error is computed against the exact solution at $t=10$.

The convergence of the temporal error performed on the grid of 256^2 cells is shown in Figure 2a. It is seen that for both PS_C and PS_E methods the second-order convergence in time is preserved, except for the PS_C method with $\beta=0$, where error stagnation appears. The reason for the error stagnation for $\beta=0$ is that the spatial error contribution becomes *non-negligible*. To check the spatial error, the grid convergence study at a constant and sufficiently small time step $\tau=0.05$ is plotted in

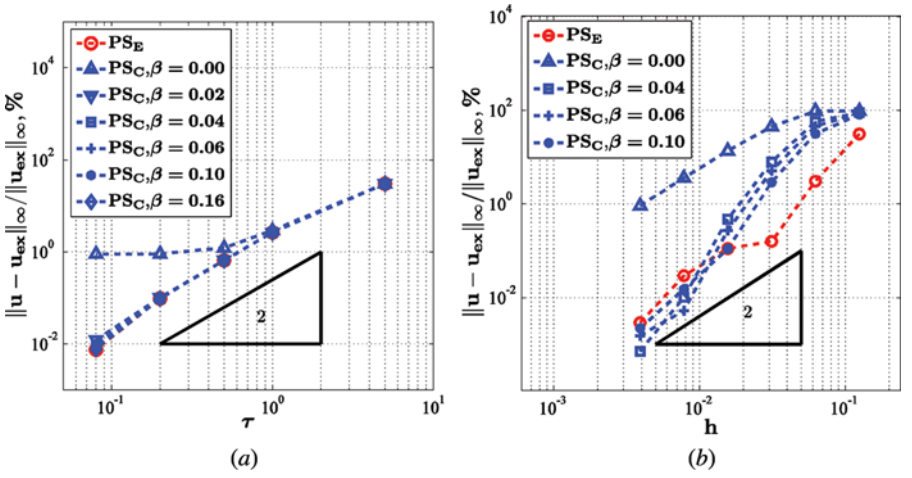

Figure 2. Convergence of temporal and spatial errors for the Taylor-Green problem using PS_E and PS_C methods: (*a*) Temporal error; (*b*) spatial error.

Figure 2*b*. It is seen that the spatial error in the case of $\beta = 0$ is much higher than that for $\beta > 0$.

The workload is measured using the number of multigrid cycles. The efficiency dependence on the β parameter of the PS_C method is shown in Figure 3*a*, where the work ratio of the PS_C and the PS_E methods is plotted. It is seen that within the set of $\beta \in [0, 0.1]$, the efficiency is nearly independent of β. Moreover, for nearly all tested grid resolutions, the SIMPLEC-compatible method with $\beta \in [0, 0.1]$ has a convergence rate about *30% higher* than the PS_E method, as seen in Figure 3*a*.

Although the optimum value of the β parameter is not known in advance, for general applications of the PS_C method, a β value of 0.04 should be used. First, for this β the convergence of the PS_C method is preserved at all grid resolutions. Second, as seen in Figure 3, this value of β ensures that both speed and accuracy of the PS_C method are higher than those of the PS_E method when an asymptotic range is reached. The next test case will show that $\beta = 0.04$ results in fast performance not only for the PS_C, but for the new SIMPLEC-compatible PI_C^0 method.

3.1.3. Roll-up of a shear-layer vortex. In this section a shear layer roll-up test case [26] is computed at $Re = 100$. The new SIMPLEC-compatible PI_C^0 method (see Table 2 in Section 2.2.3) with $\beta = 0.04$ is used to enhance the convergence rate of the SIMPLEC algorithm. To compare with the new method, the PI_E^0 interpolation is employed as a representative of Pascau's interpolation [14], which is non-fully compatible with the SIMPLEC algorithm (see Table 1 in Section 2.2.3).

The flow is initialized in the domain of $[0, \pi]$ with periodic boundary conditions,

$$u \begin{cases} \tanh\left(\frac{y-\pi/2}{\delta}\right), & y \le \pi \\ \tanh\left(\frac{3\pi/2-y}{\delta}\right), & y > \pi \end{cases} \qquad \begin{array}{l} v = \varepsilon \sin(x) \\ \delta = \pi/15, \quad \varepsilon = 0.05 \end{array} \tag{39}$$

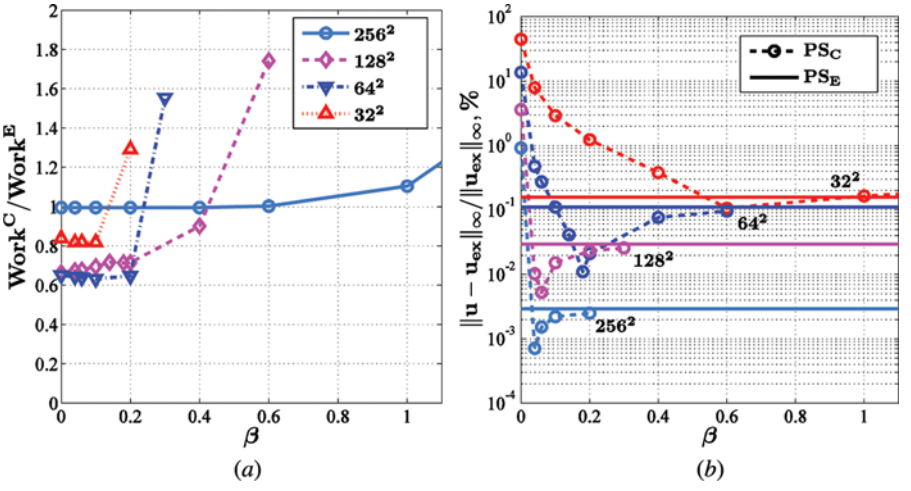

Figure 3. Dependence of workload and relative error on β for the PS_C and the PS_E methods on grids with 32^2, 64^2, 128^2, and 256^2 cells for the Taylor-Green problem: (*a*) workload ratio; (*b*) error.

The solutions are computed at $t = 4$ and the errors are measured in comparison to the reference solution, which is obtained using the PI_C^0 method on a fine grid of 512^2 cells and a small time step $\tau = 0.005$. The computations are performed on a sequence of the successively coarsened grids. The largest CFL number measured on the finest grid equals 0.2. For each of the coarse grids, the error is computed by interpolating the reference solution on the coarse grid and subtracting the solution from that grid. As seen from Figure 4*a*, second-order spatial accuracy is obtained and the error tolerances of the new method and the PI_E^0 method are nearly identical.

To compare the efficiencies of the PI_C^0 and PI_E^0 methods, the corresponding workloads, measured in CPU seconds, are found first as the functions of accuracy. Then the ratio of the workloads of the PI_C^0 method and the PI_E^0 method is plotted in Figure 4*b*. As seen in the figure, the workload of the new method is up to 25% lower compared to the workload of the PI_E^0 method. These results correlate well with the discussions in Section 2.3.4, where the SIMPLEC-compatible interpolations (such as PS_C, PI_C^0, and PI_C^∞) were supposed to enhance the convergence rate of the SIMPLEC algorithm.

3.2. Time-Step and Relaxation Parameters Dependencies and Solution Accuracy

In this section the PI^0 and the PI^∞ methods (see Sections 2.2.3–2.2.4) are evaluated with respect to their independence of both time step τ and relaxation parameter α at steady-state. In addition, the magnitude of the dependencies on τ and α of the PS method (see Section 2.2.1) is estimated. It is shown that at convergence to steady-state, the PI^0 method is the most accurate compared to both the PI^∞ and the PS methods. To evaluate the methods the steady flow around a circular cylinder and the lid-driven cavity flows are studied using the SIMPLEC algorithm.

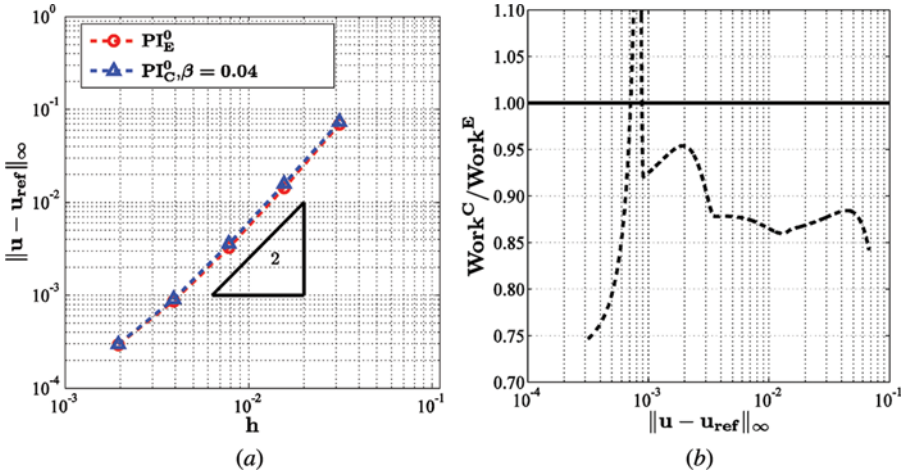

Figure 4. Comparison of spatial error convergence and workloads measured in CPU seconds for the PI_C^0 and the PI_E^0 methods in the shear-layer roll-up test case: (*a*) error; (*b*) workload ratio.

3.2.1. Steady flow around a circular cylinder. As the fourth test case, the steady flow around a circular cylinder is computed at $\mathrm{Re} = 40$. The standard $\mathrm{PS_C}$ method and the two new $\mathrm{PI_C^0}$ and $\mathrm{PI_C^\infty}$ methods (see Table 2 in Section 2.2.3) are used with $\beta = 0.04$. The drag error dependence of τ and α on a grid with 64×32 cells is plotted in Figure 5, where the drag value of 1.532 computed on the grid with 512×256 cells is used as the reference solution. It is seen that the solution of the $\mathrm{PS_C}$ method asymptotically tends to the solutions of the $\mathrm{PI_C^0}$ and the $\mathrm{PI_C^\infty}$ methods for infinitely small and large time steps, respectively. This behavior is in exact agreement with the analytical conclusions in Section 2.2.4. It is also seen that the solution dependence of the standard method on the relaxation parameter α is less significant than that on the time step τ.

As seen from Figures 5 and 6, the dependence on time step of the standard method may contribute up to 50% of the error. Contrary to the standard method, the $\mathrm{PI_C^0}$ and $\mathrm{PI_C^\infty}$ methods result in solutions which are independent of τ and α, but, as seen in the figures, the accuracy of the $\mathrm{PI_C^0}$ method is higher than that of the other methods. These results are in good agreement with the conclusion in Section 2.2.4 that the $\mathrm{PI^0}$ method results in the highest accuracy compared to both the PS and the $\mathrm{PI^\infty}$ methods.

3.2.2. Lid-driven cavity flows. In this section it is shown that the $\mathrm{PI^0}$ method is the most accurate method not only globally, but also locally. For this the flow in a lid-driven cavity is computed at $\mathrm{Re} = 1{,}000$ on a nonuniform grid with resolution of 128^2 cells. The $\mathrm{PI^0}$, PS, and $\mathrm{PI^\infty}$ methods are used in the SIMPLE-compatible forms, which are similar to Choi's interpolation and the two interpolations of Pascau (see Table 1 in Section 2.2.3). The convergence properties of the methods are compared by the errors of the vertical velocity along the horizontal line across the center of the cavity. The reference solution is computed on a grid with 512^2 cells. The steady solutions on the grid with 128^2 cells are computed using the time step $\tau = 1$. The error is computed by interpolating the reference solution on the coarse grid and subtracting the solution from that grid. From the error distribution along the

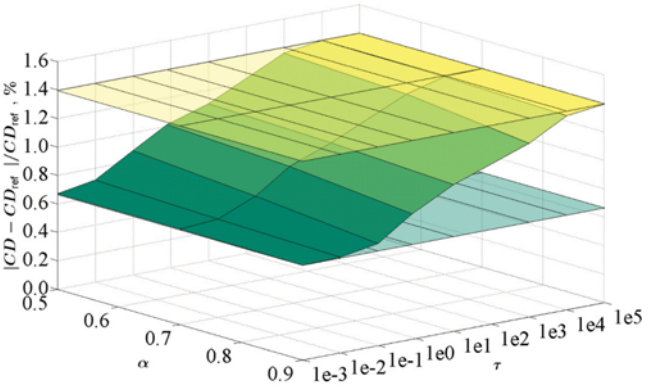

Figure 5. Dependence of relative error on both time step τ and relaxation parameter α for flows around a circular cylinder using three interpolation methods: $\mathrm{PS_C}$ (curved surface), $\mathrm{PI_C^0}$ (bottom plane), and $\mathrm{PI_C^\infty}$ (top plane).

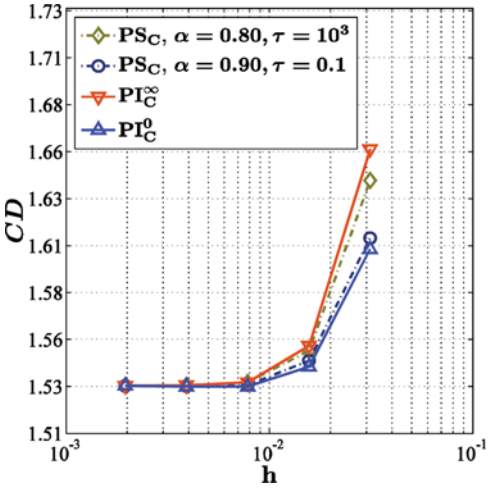

Figure 6. Dependence of the drag coefficient (CD) on grid resolution for flows around a circular cylinder at $\mathrm{Re} = 40$.

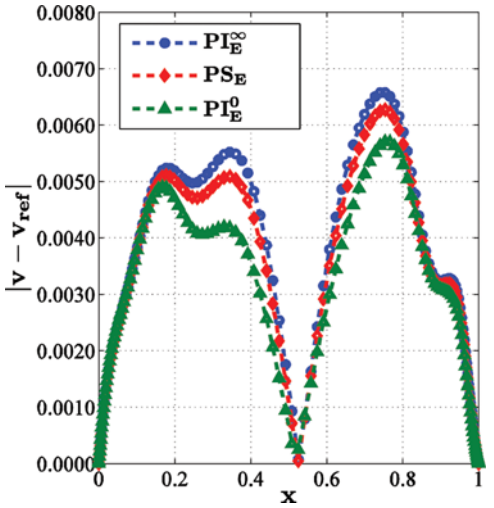

Figure 7. Velocity error distribution along the horizontal line at $y = 0.5$ across the lid-driven cavity on grid with resolution of 128^2 cells.

horizontal line, shown in Figure 7, it is seen that the $\mathrm{PI}_{\mathrm{E}}^{0}$ method is the most accurate method compared to both the $\mathrm{PI}_{\mathrm{E}}^{\infty}$ and the PS_{E} methods, which correlates well with the conclusion in Section 2.2.4.

4. SUMMARY AND CONCLUSIONS

In order to increase the convergence rate of SIMPLE-like algorithms on collocated grids, a compatibility condition between the algorithms and the mass flux interpolations has been presented. Two new SIMPLEC-compatible interpolations

have been developed. For the interpolations, the optimal value of β was found to be equal to 0.04. Numerical results of unsteady flow computations shown that employing the SIMPLEC algorithm with the SIMPLEC-compatible interpolations may result in a convergence rate up to 35% higher compared to the standard SIMPLEC algorithm. Numerical results also show that the two new interpolations are independent of both time step and relaxation parameter at convergence. The interpolations were compared with the standard interpolations of Shen and Choi. It was shown that for the standard methods the dependencies of the relaxation parameter are negligible, whereas the dependencies of the time step may contribute up to 50% of the error when coarse grids are employed. One of the new interpolations was shown to result in higher accuracy than the standard interpolations.

FUNDING

The present work was funded by the Danish Council for Strategic Research (DSF), under contract 2104-09-0026, Center for Computational Wind Turbine Aerodynamics and Atmospheric Turbulence.

REFERENCES

1. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Taylor & Francis, Washington, DC, 1980.
2. R. I. Issa, Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting, *J. Comput. phys.*, vol. 62, pp. 40–65, 1986.
3. J. P. Van Doormaal and G. D. Raithby, Enhancements of the Simple Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
4. J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, Springer-Valley, Berlin, 2002.
5. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, vol. 21, pp. 1525–1532, 1983.
6. T. F. Miller and F. W. Schmidt, Use of a Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier-Stokes Equations on a Nonstaggered Grid System, *Numerical Heat Transfer A*, vol. 14, pp. 213–233, 1988.
7. S. Majumdar, Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Nonstaggered Grids, *Numer. Heat Transfer*, vol. 13, pp. 125–132, 1988.
8. W. Z. Shen, J. A. Michelsen, N. N. Sørensen, and J. N. Sørensen, Improved Rhie-Chow Interpolation for Unsteady Flow Computations, *AIAA J*, vol. 39, pp. 2406–2409, 2001.
9. S. K. Choi, Note on the Use of Momentum Interpolation Method for Unsteady Flows, *Numer. Heat Transfer*, vol. 36, pp. 545–550, 1999.
10. B. Yu, W. Q. Tao, J. J. Wei, Y. Kawaguchi, T. Tagawa, and H. Ozoe, Discussion on Momentum Interpolation Method for Collocated Grids of Incompressible flow, *Numer. Heat Transfer B*, vol. 42, pp. 141–166, 2002.
11. B. Yu, Y. Kawaguchi, W. Q. Tao, and H. Ozoe, Checkerboard Pressure Predictions due to the Underrelaxation Factor and Time Step Size for a Nonstaggered Grid with Momentum Interpolation Method, *Numer. Heat Transfer B*, vol. 41, pp. 85–94, 2002.
12. F. S. Lien and M. A. Leschziner, A General Non-orthogonal Collocated Finite Volume Algorithm for Turbulent Flow at All Speeds Incorporating Second-Moment Turbulence-Transport Closure, Part 1: Computational Implementation, *Comput. meth. Appl. Mech. Eng.*, vol. 114, pp. 123–148, 1994.

13. A. Cubero and N. Fueyo, A Compact Momentum Interpolation Procedure for Unsteady Flows and Relaxation *Numer. Heat Transfer B*, vol. 52, pp. 507–529, 2007.
14. A. Pascau, Cell Face Velocity Alternatives in a Structured Collocated Grid for the Unsteady Navier–Stokes Equations, *Int. J. Numer. Meth.*, vol. 65, pp. 812–833, 2011.
15. R. S. Kamakoti, W. Shyy, S. Thakur and B. Sankar, Time Dependent RANS Computation for an Aeroelastic Wing, AIAA 886, 2004.
16. J. Mencinger and I. Žun, On the Finite Volume Discretization of Discontinuous Body Force Field on Collocated Grid: Application to VOF Method, *J. Comput. Phys.*, vol. 222, pp. 524–538, 2007.
17. R. Kamakoti and W. Shyy, Fluid–Structure Interaction for Aeroelastic Applications, *Prog. Aerospace Sci.*, vol. 40, pp. 535–558, 2004.
18. R. Abbasi, A. Ashrafizadeh, and A. Shadaram, A Comparative Study of Finite Volume Pressure-Correction Projection Methods on Collocated Grid Arrangements, *Comput. Fluids*, vol. 81, pp. 68–84, 2013.
19. P. Wesseling, *Principles of Computational Fluid Dynamics*, vol. 29, Springer, Heidelberg, New York, 2009.
20. W. Z. Shen, J. A. Michelsen, N. N. Sørensen, and J. N. Sørensen, An Improved SIMPLEC Method on Collocated Grids for Steady and Unsteady Flow Computations, *Numer. Heat Transfer B*, vol. 43, pp. 221–239, 2003.
21. M. Ijaz, and N. K. Anand, Co-located Variables Approach Using Implicit Runge-kutta Methods for Unsteady Incompressible Flow Simulation, *Numer. Heat Transfer B*, vol. 54, pp. 291–313, 2008.
22. V. Kazemi Kamyab, A. H. Van Zuijlen, and H. Bijl, Higher Order Implicit Time Integration Schemes to Solve Incompressible Navier-Stokes on Collocated Grids Using Consistent Unsteady Rhie-Chow, Vienna Institute of Technology, ECCOMAS, 2012.
23. P. Johansson and L. Davidson, Modified Collocated SIMPLEC Algorithm Applied to Buoyancy-Affected Turbulent Flow using a Multigrid Solution Procedure, *Numer. Heat Transfer B*, vol. 28, pp. 39–57, 1995.
24. J. Papageorgakopoulos, G. Arampatzis, D. Assimacopoulos, and N. C. Markatos, Enhancement of the Momentum Interpolation Method on Non-staggered Grids, *Int. J. Numer. Meth. Fluids*, vol. 33, pp. 1–22, 2000.
25. O. Botella and R. Peyret, Benchmark Spectral Results on the Lid-Driven Cavity Flow, *Comput. & Fluids*, vol. 27, pp. 421–433, 1998.
26. M. L. Minion and D. L. Brown, Performance of Under-resolved Two-Dimensional Incompressible Flow Simulations, II, *J. Comput. Phys.*, vol. 138, pp. 734–765, 1997.