International Journal of Heat and Mass Transfer 120 (2018) 1255–1265

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

A consistent SIMPLE algorithm with extra explicit prediction – SIMPLEPC

Hui Xiao, Junbo Wang, Zhichun Liu, Wei Liu *

School of Energy and Power Engineering, Huazhong University of Science and Technology, Wuhan 430074, PR China

ARTICLE INFO

Article history:
Received 24 September 2017
Received in revised form 15 December 2017
Accepted 22 December 2017

Keywords:
SIMPLE
SIMPLEPC
Computational method
Incompressible flow

ABSTRACT

Some improvements for SIMPLE algorithm have been proposed in the present paper. After the discretized momentum equations are solved implicitly, an extra explicit prediction for velocity will be adopted to eliminate iterative errors introduced in previous implicit prediction step. As for consistent approximation, the velocity-correction formula is amended so that the corrected velocity and corrected pressure are better to satisfy the momentum equations without relaxation and the continuity equation. Moreover, the velocity relaxation factor α is also introduced to velocity-correction formula for obtaining the appropriate velocity correction. In the SIMPLEPC algorithm, letters "P" and "C" appended to SIMPLE represent extra prediction and consistent approximation, respectively. The numerical experiments show that SIMPLEPC can always converge with less time and similar robustness compared to the SIMPLEC algorithm. When the time step multiplier (E) is less than 4 (the corresponding velocity relaxation factor α is less than 0.8), the time consumption ratio (TCR) is approximately less than 0.8. The performance of SIMPLEPC is better than SIMPLEC especially with low time step multiplier.

1. Introduction

The incompressible fluid flow is widespread in production and life, such as heat exchanger design, river flow. In general, liquid and low Mach number gas can both be regarded as incompressible fluid. In recent years, computational fluid dynamics (*CFD*) has become an important method to study fluid mechanics. Numerical simulation can obtain the characteristics of fluid flow and heat transfer quickly so that time can be saved much in the practical engineering. Due to the lack of an evolution differential equation for pressure, the continuity equation is brought into constrain pressure instead. In order to impose the continuity constraint, researchers have developed different algorithms, including Projection Method [1], SIMPLE Algorithm [2], Coupled Method [3–5], Immersed Boundary Method [6], Artificial Compressibility Method [7], Vorticity Stream Function [8], etc. In addition, the Lattice Boltzmann Method [9,10], a mesoscopic approach based on kinetic theory, is also used for simulating incompressible fluid flow in recent years. Among them, SIMPLE algorithm is widely used in steady flows [11–13], and is also extended to unsteady flows and high Mach number flow problems [14]. Owing to the high accuracy and robustness, SIMPLE algorithm is even applied to solve complicated problems, including combustion, turbulence, etc. [15]. It is of great significance to optimize the SIMPLE algorithm so as to accelerate the convergence and improve the robustness.

In 1972, Spalding and Patankar [2] proposed SIMPLE algorithm for heat, mass and momentum transfer. With the SIMPLE algorithm, pressure and velocity are decoupled so as to be solved sequentially. There are two main assumptions in SIMPLE algorithm: First, the initial pressure and velocity are given independently; the second is to neglect the velocity corrections at the neighbors. The two assumptions slow the convergence. In terms of overcoming the first assumption, Patankar [16] proposed the SIMPLER algorithm in 1980. The initial pressure is calculated with an extra discretized pressure equation in which the pressure depends on previous velocity. And then, he systematically elaborated the SIMPLE algorithm in his monograph [17]. For enhancing the SIMPLE algorithm, some researchers focused on the second assumption. In 1984, Raithby [18] introduced a consistent approximation and the SIMPLEC algorithm was provided. In 1986, Issa [19] presented an operator-splitting scheme for implicitly discretized equations called PISO algorithm. Thus, the neighbors' contribution to the velocity correction is under consideration. In 2004, Tao et al. [20] developed a fully implicit algorithm named CLEAR without utilizing the pressure correction equation. The CLEAR algorithm discards the second assumption so that the convergence rate is dramatically increased. However, an unexpected oscillation in the calculation procedure deteriorates the robustness. Hence, they developed the IDEAL [21] algorithm for improving

* Corresponding author.
E-mail address: w_liu@hust.edu.cn (W. Liu).

https://doi.org/10.1016/j.ijheatmasstransfer.2017.12.119
0017-9310/

Nomenclature

a	coefficient in the discretized equation
A	surface area
b	source term in discretized equations
d	coefficient in the velocity-correction formula
E	time step multiplier
g	gravity acceleration
L	characteristic length of cavity
p	pressure
p'	pressure correction
Pr	Prandtl number
r	Radial coordinate
R	radius in case 3
Ra	Rayleigh number in case 2
Re	Reynolds number
Rs	residual
S	source term
t	time
T	temperature
T_H, T_L	highest and lowest temperature in case 2
TCR	time consumption ratio
u, v	velocity components in x, y directions
u_m	mean inlet velocity
u', v'	velocity correction
x, y	Cartesian coordinates
δx, δy	distance of adjacent grid points in x and y directions
Δx, Δy	control volume width in x and y directions
α	velocity relaxation factor
β	coefficient of thermal expansion
Φ	general variable
ν	kinematic viscosity
μ	dynamic viscosity
ρ	density

Subscripts

cv	control volume
nb	neighboring grid points
e, w, n, s	cell faces
E, W, P	grid points
N, S	grid points
M	mass

Superscripts

0	previous iteration
$*$, $**$, $***$	first, second, and third intermediate variable

the robustness. There are also some other methods for accelerating the SIMPLE algorithm [22–28] as well as Fourier analysis of the SIMPLE-serials algorithm [29].

Based on the above previous improvements of the SIMPLE algorithm, the author realizes that, after each outer iteration, the better pressure and velocity simultaneously satisfy the continuity equation and the momentum equations, the better the economy and the robustness are. Therefore, the SIMPLE algorithm can be optimized further. The following paper is consisted of four parts: the first part reviews the organization of SIMPLE algorithm; the second part puts forward the improved strategy; the third part carries on the numerical experiments; and the last part gives the conclusion.

2. The review of SIMPLE algorithm

The essence of the SIMPLE algorithm is the prediction-correction scheme. In the prediction step, the intermediate velocity field is calculated from the discretized momentum equations based on an estimated pressure field. Generally speaking, the intermediate velocity field may not satisfy the continuity equation. In the correction step, the appropriate velocity correction is obtained so that the continuity equation can be satisfied. In this way, one outer iteration is accomplished. After several outer iterations, the velocity and the pressure will become the converged solution of the simultaneous partial differential equations.

2.1. System of partial differential equations

Continuity equation

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0 \tag{1}$$

Momentum equation:

$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u u)}{\partial x}+\frac{\partial(\rho v u)}{\partial y}=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)+S_u \tag{2}$$

$$\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho u v)}{\partial x}+\frac{\partial(\rho v v)}{\partial y}=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right)+S_v \tag{3}$$

General form:

$$\frac{\partial(\rho\Phi)}{\partial t}+\frac{\partial(\rho u\Phi)}{\partial x}+\frac{\partial(\rho v\Phi)}{\partial y}=\frac{\partial}{\partial x}\left(\Gamma_\Phi\frac{\partial\Phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma_\Phi\frac{\partial\Phi}{\partial y}\right)+S_\Phi \tag{4}$$

2.2. The discretization of above equations

Fig. 1 shows the organization of the staggered grid in the xy-plane. The finite-volume method proposed by Patankar [17] is applied to discretize the general differential equation and the algebraic approximation becomes

$$a_P\Phi_P=a_E\Phi_E+a_W\Phi_W+a_N\Phi_N+a_S\Phi_S+b \tag{5}$$

Similar to the form of Eq. (5), the discretization momentum equation can be written as follows (Taking the u direction as an example and it is the same in the following context).

$$a_e u_e^*=\sum a_{nb}u_{nb}^*+b+(p_P^*-p_E^*)A_e \tag{6}$$

Introducing the relaxation factor α into Eq.(6), the intermediate velocity can be computed by Eq. (7) based on the velocity and the pressure obtained after the previous outer iteration.

$$\frac{a_e}{\alpha}u_e^*=\sum a_{nb}u_{nb}^*+b+(p_P^*-p_E^*)A_e+\frac{1-\alpha}{\alpha}a_e u_e^0 \tag{7}$$

2.3. The velocity-correction formula

In order to satisfy both the momentum and continuity constraint, the modifications to velocity and pressure are implemented in Eqs. (8) and (9).

$$p=p^*+p' \tag{8}$$

$$u=u^*+u' \tag{9}$$

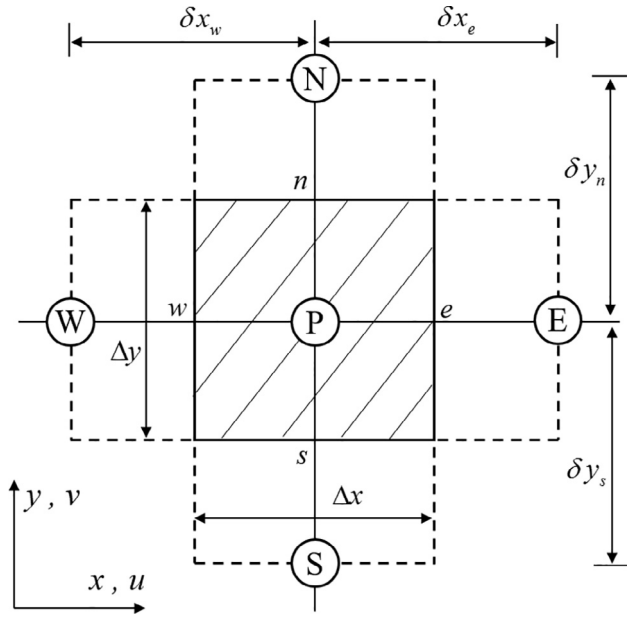

Fig. 1. The staggered grid in the *xy*-plane.

Substituting u and p from Eqs. (8) and (9) respectively into momentum equation, Eq. (10) is obtained as follows.

$$\frac{a_e}{\alpha}(u_e^* + u_e') = \sum a_{nb}(u_{nb}^* + u_{nb}') + b + \left[(p_P^* + p_P') - (p_E^* + p_E')\right]A_e + \frac{1-\alpha}{\alpha}a_e u_e^0 \tag{10}$$

Subtracting Eq.(7) from Eq. (10), Eq. (11) is obtained as

$$\frac{a_e}{\alpha}u_e' = \sum a_{nb}u_{nb}' + (p_P' - p_E')A_e \tag{11}$$

For computational convenience, the term $\sum a_{nb}u_{nb}'$ is boldly dropped and Eq. (11) is simplified to Eq. (12).

$$u_e' = d_e(p_P' - p_E') \tag{12}$$

where

$$d_e = \frac{A_e}{a_e/\alpha}$$

Eq. (12) is the velocity-correction formula and it can also be written as the following form.

$$u_e = u_e^* + d_e(p_P' - p_E') \tag{13}$$

Similarly, correction formulas in other directions can be easily obtained.

2.4. The continuity constraint and the pressure correction equation

The discretized continuity equation is given as

$$\frac{(\rho_P - \rho_P^0)\Delta V}{\Delta t} + (\rho u)_e A_e - (\rho u)_w A_w + (\rho v)_n A_n - (\rho v)_s A_s = 0 \tag{14}$$

Substituting the corrected velocity expressions like Eq. (13) for all the velocity components above, after rearrangement, the pressure correction equation can be written as Eq. (15).

$$a_P p_P' = a_E p_E' + a_W p_W' + a_N p_N' + a_S p_S' + b \tag{15}$$

where

$$a_P = a_E + a_W + a_N + a_S, a_E = (\rho A d)_e, a_W = (\rho A d)_w,$$
$$a_N = (\rho A d)_n, a_S = (\rho A d)_s$$

$$b = \frac{(\rho_P^0 - \rho_P)\Delta V}{\Delta t} + (\rho u^* A)_w - (\rho u^* A)_e + (\rho v^* A)_s - (\rho v^* A)_n$$

As for the boundary condition of the pressure correction equation, Partankar [17] points out that the boundary pressure correction is zero no matter whether the boundary pressure value or the normal velocity at the boundary is given. The treatment of the boundary is deemed to be equivalent to the homogeneous Neumann boundary condition [30].

2.5. The SIMPLEC approximation

Comparing SIMPLE and SIMPLEC, the only difference of both algorithms is the consistent approximation adopted in SIMPLEC. With the consistent approximation, the derivation of the velocity-correction formula is different. The term $\sum a_{nb}u_e'$ is subtracted from both sides of Eq. (11) first. With elementary operations, it yields

$$u_e' = \frac{\sum a_{nb}(u_{nb}' - u_e')}{(a_e/\alpha - \sum a_{nb})} + \frac{A_e}{(a_e/\alpha - \sum a_{nb})}(p_P' - p_E') \tag{16}$$

Neglecting the first term in the right side, the velocity-correction formula becomes

$$u_e = u_e^* + d_e(p_P' - p_E') \tag{17}$$

where

$$d_e = \frac{A_e}{(a_e/\alpha - \sum a_{nb})}$$

Besides, the pressure relaxation factor in SIMPLEC is always 1 due to the consistent approximation. In a word, except velocity-correction formulas and the pressure relaxation factor, SIMPLEC is identical to SIMPLE.

3. The SIMPLEPC algorithm

3.1. Improvements for the SIMPLE algorithm

Though the SIMPLE algorithm is improved in kinds of aspects through previous works, there are still some issues worthy of optimization.

First, generally speaking, the iterative residuals of the momentum equations like Eq. (7) are always not small enough for the purpose of saving time. Thus, we have

$$\frac{a_e}{\alpha}u_e^* \neq \sum a_{nb}u_{nb}^* + b + (p_P^0 - p_E^0)A_e + \frac{1-\alpha}{\alpha}a_e u_e^0 \tag{18}$$

Rearranging the above inequation, the iterative error (IE) is expressed as

$$IE = \sum a_{nb}u_{nb}^* + b + (p_P^0 - p_E^0)A_e + \frac{1-\alpha}{\alpha}a_e u_e^0 - \frac{a_e}{\alpha}u_e^* \tag{19}$$

Subtracting Eq. (19) from Eq. (10), it is obtained as

$$\frac{a_e}{\alpha}u_e' = \sum a_{nb}u_{nb}' + (p_P' - p_E')A_e + IE \tag{20}$$

However, iterative errors are not taken into account when deriving the pressure correction equation in the SIMPLE algorithm. Therefore, iterative errors from the momentum equation will propagate to the corrected pressure and velocity. Thus, the convergence rate could be slowed down.

Second, it is pretty good that the corrected velocity satisfies the continuity equation. But with the application of velocity relaxation factor, the corrected pressure and velocity satisfy the modified momentum equation like Eq. (7) rather than the original momentum equation like Eq. (6). It could also cause a slow convergence.

As it is known, the discretized momentum equation with under-relaxation factor is shown as Eq. (7). When the iteration reaches convergence, there exists $u_e^* = u_e^0$ so that Eq. (7) becomes Eq. (6). Obviously, the under-relaxation factor vanishes in the converged momentum equation. In consideration for accelerating convergence, the form of discretized momentum equation which consists of corrected velocity and pressure in every iteration step should be consistent with the converged form like Eq. (6). In other words, the under-relaxation factor is redundant so that the corrected velocity and pressure should satisfy the original momentum equation like Eq. (6).

Thirdly, there is not an economical treatment to completely eliminate the effect caused by the omission of the velocity corrections at the adjacent grids. Thus, the velocity-correction formulas have the potential for further optimization.

In order to settle the above three issues, the present paper adopts three corresponding strategies. First, an extra explicit prediction for the intermediate velocity is adopted to eliminate iterative errors introduced in solving momentum equations like Eq. (7). Then, the velocity-correction formula is amended so that the corrected velocity and corrected pressure are better to satisfy the continuity equation and the original momentum equations. Finally, the relaxation factor α is introduced to velocity-correction formula for the purpose of obtaining the appropriate velocity correction and improving the robustness.

3.2. The procedures of SIMPLEPC

The SIMPLEPC algorithm is organized as follows:

Step 1 Estimate the initial velocity field u^0, v^0 and initial pressure field p^0.
Step 2 Evaluate the coefficients of the momentum equations like Eq. (21) based on the initial velocity field u^0, v^0.
Step 3 Get the first intermediate velocity u^* and v^* by solving momentum equations like Eq. (21) where p^0 and u^0 are the results of the previous iteration.

$$\frac{a_e}{\alpha}u_e^* = \sum a_{nb}u_{nb}^* + b + (p_P^0 - p_E^0)A_e + \frac{1-\alpha}{\alpha}a_e u_e^0 \tag{21}$$

Step 4 Evaluate the second intermediate velocity u^{**}, v^{**} and the third intermediate velocity u^{***}, v^{***}. Calculate the source term of the pressure correction equation and then solve it to obtain p'.
Step 5 Get the corrected pressure field p and utilize the velocity-correction formulas to get the corrected velocity u, v.
Step 6 Return to step 2 and start the next iteration with the corrected pressure and corrected velocity until it achieves the convergence.

Remarks

(1) The explicit prediction is applied to eliminate the iterative errors.

The second intermediate velocity is obtained as follows (Taking the u direction as an example and it is the same in the following context).

$$\frac{a_e}{\alpha}u_e^{**} = \sum a_{nb}u_{nb}^* + b + (p_P^0 - p_E^0)A_e + \frac{1-\alpha}{\alpha}a_e u_e^0 \tag{22}$$

Explicit prediction has two aspects. On the one hand, the iteration advances an extra step, which marches the iteration more thoroughly to reduce iterative errors. On the other hand, the explicit prediction helps take iterative errors into consideration in the velocity correction formula. Together, these two aspects eliminate iterative errors.

(2) The velocity-correction formula of the SIMPLPC algorithm.

It is shown the derivation of the velocity-correction formula in the following. The velocity and pressure appended with respective corrected terms should satisfy the following equation.

$$a_e(u_e^{**} + u_e') = \sum a_{nb}(u_{nb}^{**} + u_{nb}') + b + [(p_P^0 + p_P') - (p_E^0 + p_E')]A_e \tag{23}$$

Subtracting Eq. (22) from Eq. (23) and rearranging it, it yields

$$a_e u_e' = -a_e u_e^{**} + \sum a_{nb}u_{nb}^{**} + b + (p_P^0 - p_E^0)A_e + (p_P' - p_E')A_e + \sum a_{nb}u_{nb}' \tag{24}$$

Now, specifying $S = \sum a_{nb}u_{nb}^{**} + b + (p_P^0 - p_E^0)A_e - a_e u_e^{**}$ to simplify Eq. (24), it yields

$$a_e u_e' = S + \sum a_{nb}u_{nb}' + A_e(p_P' - p_E') \tag{25}$$

As we can see, the iterative error term is disappeared in Eq. (25) compared to Eq. (20). Therefore, the iterative error is eliminated. Obviously, iterative errors elimination is mainly contributed by the explicit prediction which transforms Eq. (18) into Eq. (22). For economy considerations, it is a must to make an approximate treatment to the velocity corrections at the adjacent points. This may lead to errors of the velocity correction at the present point. If the velocity correction u_e' is larger than true value, oscillations and even computational divergence may occur. And it deteriorates the robustness. Hence, the under-relaxation factor is necessary for the velocity correction and the value is equal to that in Eq. (21) for simplicity. It is written as

$$\frac{a_e}{\alpha}u_e' = S + \sum a_{nb}u_{nb}' + A_e(p_P' - p_E') \tag{26}$$

Specify $u_e' = u_{e1}' + u_{e2}'$ where u_{e1}' and u_{e2}' satisfy Eqs. (27) and (28), respectively.

$$u_{e1}' = \alpha S/a_e \tag{27}$$

$$\frac{a_e}{\alpha}u_{e2}' = \sum a_{nb}u_{nb}' + A_e(p_P' - p_E') \tag{28}$$

Subtracting the term $\sum a_{nb}u_{e2}'$ from both sides of Eq. (28), similar to the SIMPLEC approximation, it yields

$$\left(\frac{a_e}{\alpha} - \sum a_{nb}\right)u_{e2}' = \sum a_{nb}(u_{nb}' - u_{e2}') + A_e(p_P' - p_E') \tag{29}$$

The above Eq. (29) can be also written as

$$u_{e2}' = \frac{\sum a_{nb}(u_{nb}' - u_{e2}')}{(a_e/\alpha - \sum a_{nb})} + d_e(p_P' - p_E') \tag{30}$$

where d_e is identical to that in Eq. (17) (The same in Eqs. (31), (32), (33), (34), (36)).

Thus, Eq. (26) can be written as

$$u_e' = \frac{\sum a_{nb}u_{nb}' - \sum a_{nb}u_{e2}'}{(a_e/\alpha - \sum a_{nb})} + \alpha S/a_e + d_e(p_P' - p_E') \tag{31}$$

Assuming that $\sum a_{nb}u_{e2}' \approx \sum a_{nb}u_{nb}'$, the first term in the right hand of Eq. (31) can be dropped and Eq. (31) becomes

$$u_e' = \alpha S/a_e + d_e(p_P' - p_E') \tag{32}$$

or

$$u_e = u_e^{**} + u_e' = u_e^{***} + d_e(p_P' - p_E') \tag{33}$$

where

$$u_e^{***} = u_e^{**} + \alpha S/a_e = u_e^{**}(1-\alpha) + (\sum a_{nb}u_{nb}^{**} + b + (p_P^0 - p_E^0)A_e)\alpha/a_e$$

Eq. (33) is the new velocity-correction formula for the SIMPLEPC algorithm. Compared to the SIMPLEC algorithm, the proportion of the neglected term in Eq. (31) is less due to the emergence of the source term $\alpha S/a_e$. Therefore, the consistent approximation in SIMPLEPC is better than that in SIMPLEC. In this way, the corrected velocity and pressure satisfy the momentum equation better, which accelerates convergence.

(3) The corrected pressure and velocity.

Substituting new velocity-correction formulas for velocity components in the discretized continuity equation, the pressure correction equation is obtained as follows.

$$a_P p'_P = a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S + b \tag{34}$$

where

$$a_P = a_E + a_W + a_N + a_S, a_E = (\rho Ad)_e, a_W = (\rho Ad)_w,$$
$$a_N = (\rho Ad)_n, a_S = (\rho Ad)_s$$

$$b = (\rho u^{***}A)_w - (\rho u^{***}A)_e + (\rho v^{***}A)_s - (\rho v^{***}A)_n$$

Solving the pressure correction equation, the pressure correction p'_P is obtained. The corrected pressure can be calculated by Eq. (35).

$$p_P = p_P^0 + p'_P \tag{35}$$

The corrected velocity can be obtained with the following equation.

$$u_e = u_e^{**} + u'_e = u_e^{***} + d_e(p'_P - p'_E) \tag{36}$$

(4) Comments on the SIMPLEPC algorithm.

The implementation of this algorithm. The procedures and equations of SIMPLEPC is identical to SIMPLEC except step 4. It is obvious that SIMPLEC can extend to SIMPLEPC similar to the extension from SIMPLE to SIMPLEC. Evaluating the second intermediate velocity (from Eq. (22)) and then obtaining the third intermediate velocity (from Eq. (33)) are the only two steps which SIMPLEPC adds to the procedures of SIMPLEC. Compared to the time consumption of a whole outer iteration, the cost of obtaining the third intermediate velocity based on the first intermediate velocity can be neglected. The codes of these two steps are simple and easy to be realized. Thus, SIMPLPC can be easily implemented based on SIMPLEC.

Reasons for the convergence acceleration. The following three reasons have contributed to the convergence acceleration of the SIMPLEPC algorithm: (*a*) Iterative errors are taken into account; (*b*) The velocity relaxation factor does not introduced into Eq. (23); (*c*) The source term in the velocity correction equations like Eq. (31) can suppress the effect of approximate treatment of neighboring velocity corrections. The above three points will promote the satisfaction of the discretized momentum equation with corrected velocity and pressure in every iteration step. Thus, the convergence can be accelerated.

The scalability of this algorithm. For simplicity, this paper only elaborates this algorithm in two dimensional flow, but the extension to three dimensional cases is straightforward. In addition, the main aim of this methodology is to handle the velocity-pressure coupling for incompressible flow. Fortunately, similar to other SIMPLE series algorithms, this methodology can be extended to compressible flow. As for some specific flow problems which are dominated by the source term, future researches can be carried out to incorporate the source term into velocity correction formulas. Furthermore, this method will be extended for the unstructured mesh with the consideration of the least square methods in evaluating the gradients on the surface of the cells in the future work.

4. Numerical experiments

4.1. Preconditions

4.1.1. Numerical computation objective

For the ease of analysis, the numerical experiments in this paper are based on the assumption of incompressible, steady, and laminar flow with constant fluid properties. With years of development, there are many benchmark questions [31–35] in computational fluid dynamics. In this paper, three questions of them are selected to verify this methodology, namely: (*a*) lid-driven flow in a square cavity; (*b*) natural convection in a square cavity; (*c*) sudden expansion flow in the tube. Additionally, the case, heat transfer enhancement with inserts, is also chose for verifying the performance in complicated fluid flow and heat transfer problems. The thrust of this part is to study the economy and robustness of SIMPEPC. The comparison of SIMPEPC and SIMPLEC will be carried out with the grids from coarse to dense, the dimensionless parameter from low to high, and different boundary conditions. Its computational accuracy is consistent with SIMPLEC and SIMPLE so that no additional useless discussion of accuracy is required. Moreover, selecting difference schemes and solutions of algebraic equations is not the objective of this article, either. Also for the sake of convenience, the power law scheme recommended by Partankar is applied and the linear algebraic equations are solved by the alternating direction implicit method (ADI) with auxiliary block correction technique [36]. The staggered grid system is employed and the number of grids is specified in the following specific case. At last, the contrast of economy in both algorithms is carried out based on the number of iterations and time consumption ratio. The time consumption ratio (*TCR*) is defined by

$$TCR = \frac{T_{SIMPLEPC}}{T_{SIMPLEC}} \tag{37}$$

where $T_{SIMPLEPC}$ is the time consumption of SIMPLEPC algorithm and $T_{SIMPLEC}$ is the time consumption of SIMPLEC algorithm.

4.1.2. The E-factor formulation

The time step multiplier *E*, recommended by Van Doormaal and Raithby [18], embodies the consistency of time and step advancement, and is capable of clear physical meaning. The relationship of the time step multiplier and velocity relaxation factor is given by Eq. (38).

$$E = \alpha/(1-\alpha) \quad (0 < \alpha < 1) \tag{38}$$

In this paper, it is found that, the number of iterations of both algorithms is decreased first and then increased with the increase of *E*. When the number of iterations is minimum, the corresponding time step multiplier is called the optimal time step multiplier. The number of iterations will be increased when *E* is greater than the optimal value. Considering the demands of time saving, no research is necessary for those cases. Therefore, in the following tables, the data marked with black box and the blank are useless.

4.1.3. Convergence criterion

The relative residuals can be calculated from Eqs. (39) and (40). When employing relative criterion, the typical convergence criterion is 10^{-3}–10^{-5} [37]. And the appropriate convergence criteria will be specified in different cases.

$$Rs_M = \sum_{cv} b_p/q_m \tag{39}$$

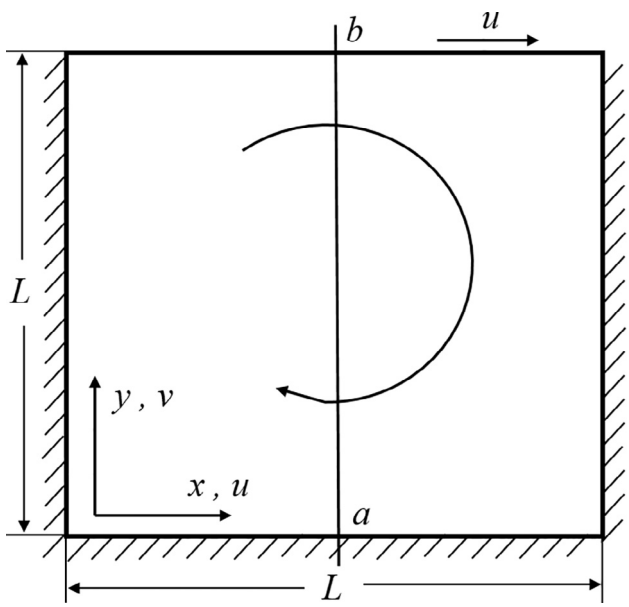

Fig. 2. The diagram of lid-driven flow.

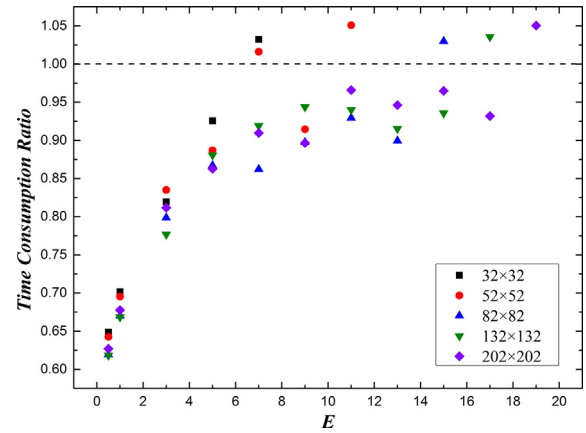

Fig. 4. The *TCR* of case verification 1.

$$Rs_{\Phi} = \left(\sum_{cv}[a_P\Phi_P - \sum a_{nb}\Phi_{nb} - b]^2\right)^{1/2} \Big/ \left(\sum_{cv}(a_P\Phi_P)^2\right)^{1/2} \tag{40}$$

where $\Phi = (u, v, T)$. b_p is the source term of the pressure correction equation, which represents the mass residual of a control volume. q_m denotes the referential mass which can be evaluated by the following equation.

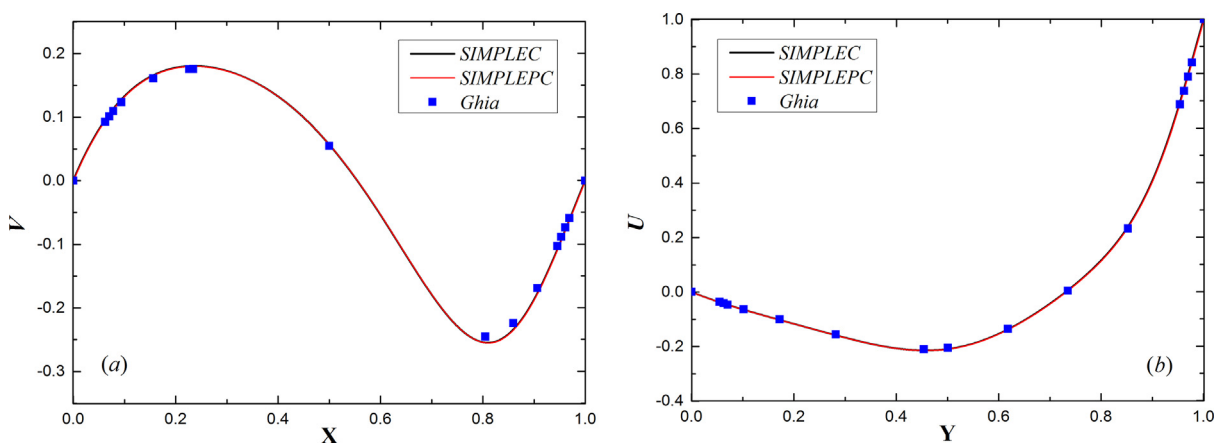

Fig. 3. The distributions of dimensionless velocity along centerlines: (*a*) Y = 0.5; (*b*) X = 0.5.

Table 1
The comparison of iteration numbers between SIMPEPC and SIMPLEC.

	α	0.333	0.500	0.750	0.833	0.875	0.900	0.917	0.929	0.938	0.944	0.950
	E	0.5	1	3	5	7	9	11	13	15	17	19
202×202	SIMPLEC	19242	9606	3201	1892	1456	1173	950	890	812	756	676
	SIMPLEPC	10703	5774	2305	1448	1175	933	814	747	695	625	630
132×132	SIMPLEC	10287	5139	1728	1046	764	617	526	488	437	398	402
	SIMPLPC	5725	3091	1208	829	632	524	445	402	368	371	327
82×82	SIMPLEC	4817	2408	816	486	374	301	268	249	217	218	–
	SIMPLPC	2683	1451	586	379	290	243	224	199	201	201	–
52×52	SIMPLEC	2233	1119	381	230	169	155	136	150	–	–	–
	SIMPLPC	1245	675	276	177	149	123	124	142	–	–	–
32×32	SIMPLEC	946	476	167	104	91	112	–	–	–	–	–
	SIMPLPC	529	288	118	83	81	104	–	–	–	–	–

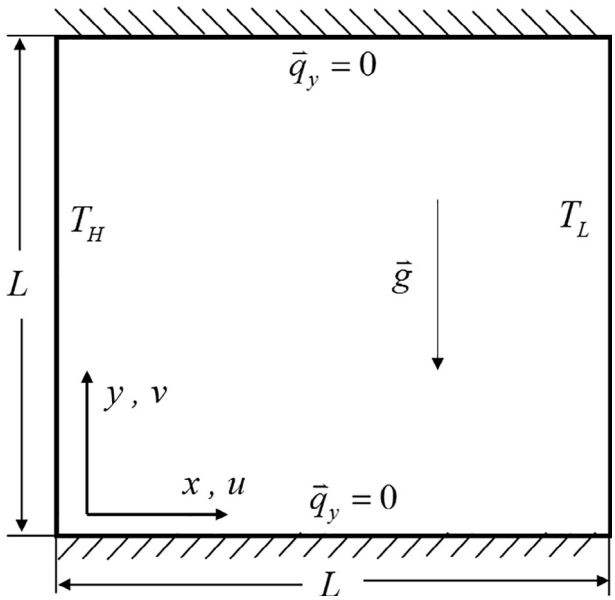

Fig. 5. The diagram of natural convection in a square cavity.

Table 2
Mean Nusselt number at hot wall.

Ra	10^4	10^5	10^6
Barakos [35]	2.245	4.510	8.806
SIMPLEC	2.261	4.606	9.004
SIMPLEPC	2.257	4.605	9.004

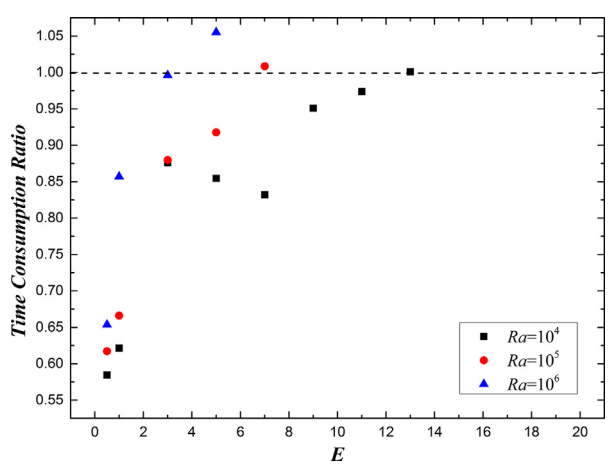

Fig. 6. The *TCR* of case verification 2.

$$q_m = \int_a^b \rho |u_m| dy \tag{41}$$

4.2. Case verification 1: lid-driven flow in a square cavity

The lid-driven flow in a square cavity is presented in Fig. 2. The Reynolds number (*Re*) is kept at 100 and given by Eq. (42).

$$Re = \rho L u / \mu \tag{42}$$

The different uniform grids systems 32 × 32, 52 × 52, 82 × 82, 132 × 132, and 202 × 202 are employed to analyze the calculation economy with the grids from coarse to dense. The convergence criteria are set as $Rs_M \le 10^{-7}$, $Rs_u \le 10^{-5}$, $Rs_v \le 10^{-5}$ and the convergence is achieved. The distributions of dimensionless velocity along two centerlines are displayed in Fig. 3. As it is shown, the predicted distributions of SIMPLEC and SIMPLEPC agree fairly well with the benchmark results provided by Ghia et al.[33]. It demonstrates that the code is correct, and that accurate results can be obtained by these two algorithms.

The comparison of iterations between SIMPEPC and SIMPLEC is listed in Table 1. A general law can be concluded from this table that the iteration number is decreased with the increase of *E* and the decrease of grids number. The iteration number of SIMPLEPC is always less than that of SIMPLEC within the range of research. Also, the robustness is weakened with the decrease of grids number and it is gratifying that the robustness of SIMPLEPC is similar to SIMPLEC. A clear description of time consumption ratio is showed in Fig. 4. The time consumption ratio is approximately from 0.6 to 1. It is also demonstrated that SIMPLEPC performs better when low time step multiplier is applied.

4.3. Case verification 2: Natural convection in a square cavity

The problem of natural convection in a square cavity is depicted in Fig. 5.

The Rayleigh number (*Ra*) is the dimensionless parameter and defined by

$$Ra = \frac{g\beta L^3 \Delta T}{\nu^2} \cdot Pr \tag{43}$$

where $\Delta T = T_H - T_L$. With *Ra* from low to high, the analysis of economy is implemented. The value of *Ra* is 10^4, 10^5, and 10^6. The Boussinesq approximation is applied. The grids system is uniform and set as 202 × 202. The convergence criteria are $Rs_M \le 10^{-7}$, $Rs_u \le 10^{-5}$, $Rs_v \le 10^{-5}$, $Rs_T \le 10^{-7}$ and the convergence is achieved. Based on the converged temperature field, the mean Nusselt number at high temperature wall is calculated and listed in Table 2. The calculated results are in excellent agreement with the benchmark results, which demonstrates the results is reliable in this case.

Table 3
The iteration number of natural convection in a square cavity.

	α	0.333	0.500	0.750	0.833	0.875	0.900	0.917	0.929	0.938	0.944	0.950
	E	0.5	1	3	5	7	9	11	13	15	17	19
Ra=10^4	SIMPLEC	7216	3584	1150	878	770	686	684	678	687	–	–
	SIMPLEPC	4002	2113	956	712	608	619	632	644	658	–	–
Ra=10^5	SIMPLEC	3901	1928	1081	1029	986	1002	–	–	–	–	–
	SIMPLPC	2160	1152	853	847	892	923	–	–	–	–	–
Ra=10^6	SIMPLEC	2037	1414	1039	901	1101	–	–	–	–	–	–
	SIMPLPC	1168	1063	908	834	1014	–	–	–	–	–	–

Table 3 shows that the iteration number of SIMPLEPC is less. With the increase of *Ra*, the flow becomes more complicated so that the calculation divergences more easily. The robustness of both algorithms are similar. The time consumption ratio is depicted in Fig. 6. Most points in the figure are below the dash line, which means the time consumption of SIMPLEPC is always less than that of SIMPLEC. Thus, SIMPLEPC can also behave better with the variation of dimensionless parameter.

4.4. Case verification 3: Sudden expansion flow in the tube

The sudden expansion flow in the tube is showed in Fig. 7. It can be known in Ref. [31] that the sudden expansion flow in the tube is axisymmetric when *Re* is low. The Reynolds number in this case is always less than 200 so that the cylindrical coordinate is employed.

The Reynolds number is given by Eq. (44). When the laminar flow is fully developed, outlet velocity should satisfy the theoretic profile [38] as Eq. (45).

$$Re = \frac{2\rho u_m r}{\mu} \tag{44}$$

$$u = \frac{1}{2}u_m(1-(r/R)^2) \tag{45}$$

This case is used to observe the computational economy when given different boundaries. The inlet velocity boundary is uniform, parabolic, and linear, respectively. The uniform grids 22×32 is applied. The convergence criteria are $Rs_M \le 10^{-7}$, $Rs_u \le 10^{-5}$ and the convergence is achieved. It is displayed in Fig. 8 that the calculated outlet velocity profile is in accordance with respective theoretic profile whatever the inlet velocity is given uniform, linear, or parabolic.

The iteration number is listed in Table 4 and time consumption ratio is demonstrated in Fig. 9. And the results are similar with those of case verification 1 and case verification 2.

4.5. Application in complicated case: Convective heat transfer enhancement with inserts

Based on three numerical experiments above, some conclusions can be found. The iteration number of SIMPLEPC is always less. The time consumption is also less with applying SIMPLEPC when the time step multiplier *E* is low, about less than 10. If the time step multiplier is less than 4, the time consumption ratio is approximately less than 0.8. If the value of *E* is beyond 10, the time consumption of both algorithms are almost the same. The robustness of both algorithms are similar. In conclusion, the performance

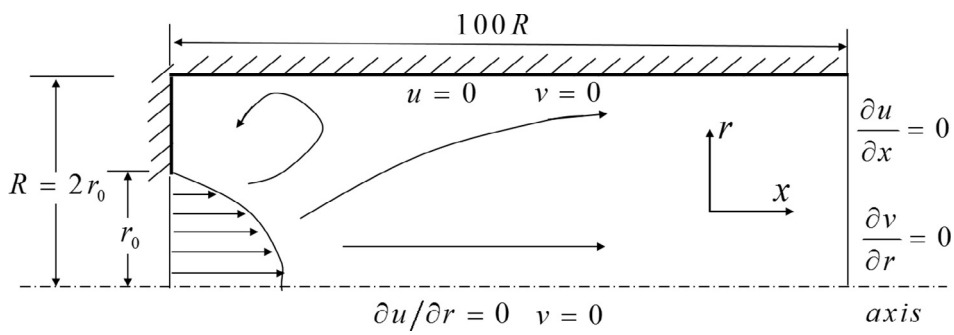

Fig. 7. The schematic diagram of sudden expansion flow in the tube.

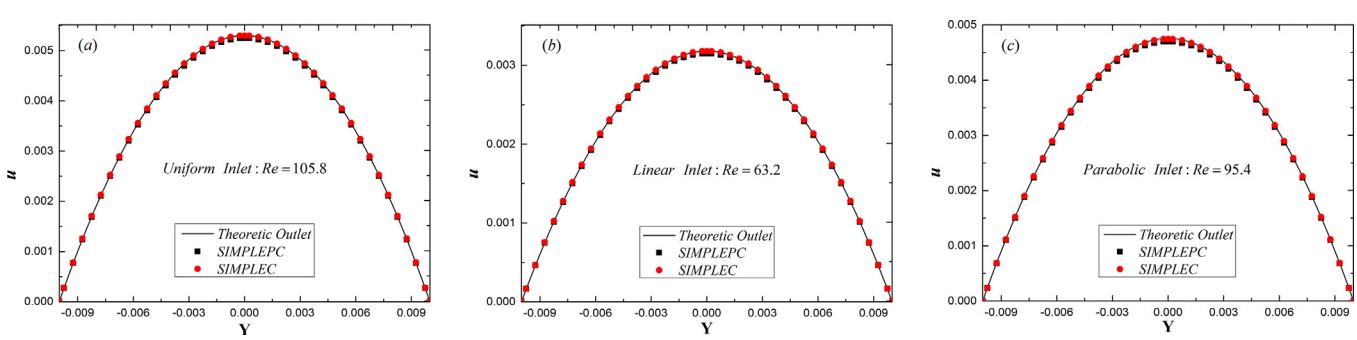

Fig. 8. The verification of outlet velocity profile calculated by SIMPLEC and SIMPLEPC.

Table 4
The iteration number of sudden expansion in the tube.

α		0.333	0.500	0.750	0.833	0.875	0.900	0.917	0.929	0.938	0.944	0.950
E		0.5	1	3	5	7	9	11	13	15	17	19
Uniform	SIMPLEC	434	224	76	48	36	29	25	24	21	20	19
	SIMPLEPC	287	150	53	37	29	24	21	23	19	17	17
Linear	SIMPLEC	473	226	78	52	39	30	26	23	22	21	20
	SIMPLPC	272	133	54	41	32	25	22	22	20	18	18
Parabolic	SIMPLEC	443	223	78	49	38	30	26	23	21	20	20
	SIMPLPC	245	146	57	37	29	25	22	20	21	18	17

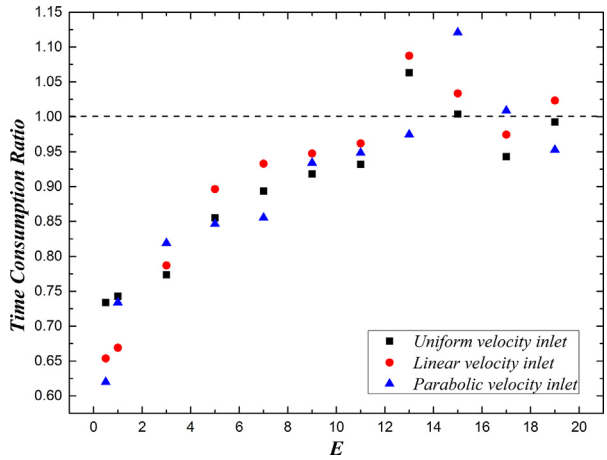

Fig. 9. The *TCR* of case verification 3.

of SIMPLEPC is better than SIMPLEC especially with low time step multiplier.

In order to verify the SIMPLEPC algorithm further, this paper carry out a complicated case of convective heat transfer enhancement with inserts in the following. The diagram of heat transfer enhancement with inserts in the two dimensional plate is depicted in Fig. 10.

The Reynolds number is 100 and the velocity inlet is defined as parabolic profile while local unidirectional assumption is employed in the outlet.

$$Re = \rho u_m H/\mu \tag{46}$$

$$u = \frac{6u_m}{H^2}(Hy - y^2) \tag{47}$$

As shown in Fig. 10, inserts are immersed in the flow region, just like isolated islands. Such isolated island is dealt with by the method recommended by Tao [36]. The grids system is 122 × 1002 which is uniform in u direction and denser near wall in v direction. The working fluid is water. Fluid viscosity is dependent on local temperature, which satisfies the formula recommended by white [38].

$$\ln\left(\frac{\mu}{\mu_0}\right) \approx -1.94 - 4.80\left(\frac{T_0}{T}\right) + 6.74\left(\frac{T_0}{T}\right)^2 \tag{48}$$

where μ_0 = 0.001792 kg/(m·s) T_0 = 273.16 K. Other physical parameters such as density (ρ), conductivity (λ), and specific heat (c_p) are constant and depend on the characteristic temperature. The

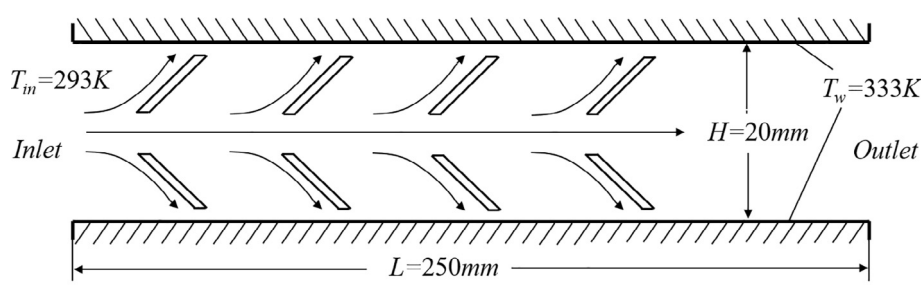

Fig. 10. The diagram of convective heat transfer enhancement with inserts.

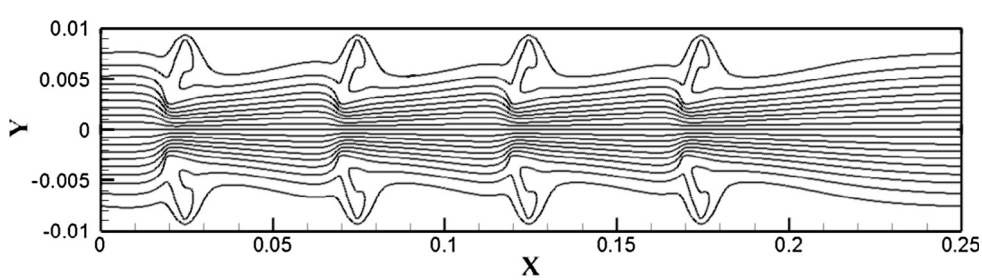

Fig. 11. Streamlines in the case of convective heat transfer enhancement with inserts.

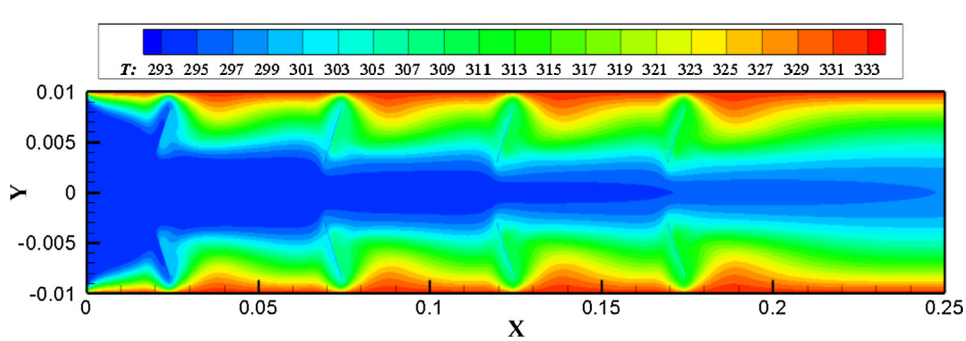

Fig. 12. Temperature field in the case of convective heat transfer enhancement with inserts.

Table 5
The iteration results of convective heat transfer enhancement with inserts.

α	0.333	0.5	0.75	0.833	0.875	0.9	0.917	0.929	0.938	0.944	0.95
E	0.5	1	3	5	7	9	11	13	15	17	19
Iteration Results											
SIMPLEC	12,617	6819	2982	2221	1888	1714	1640	1608	1595	1580	1575
SIMPLEPC	7145	4242	2269	1837	1635	1517	1487	1476	1474	1474	1483
Time Consumption Ratio											
TCR	0.625	0.686	0.839	0.913	0.955	0.976	1.000	1.013	1.020	1.029	1.039

convergence criteria are $Rs_M \leq 10^{-7}$, $Rs_u \leq 10^{-5}$, $Rs_v \leq 10^{-5}$, $Rs_T \leq 10^{-7}$ and the convergence is achieved.

Since the code has been verified in the previous cases, there is no need to carry out redundant code verification work. The calculated results of SIMPLEC and SIMPLEPC are exactly the same. The streamlines and temperature field of SIMPLEPC are depicted in Figs. 11 and 12, respectively. The iteration results of these two algorithms are listed in Table 5. Though the fluid flow becomes more complicated, the SIMPLEPC algorithm retains its edge when velocity relaxation factor is low.

5. Conclusion

For the purpose of accelerating the calculation convergence of steady incompressible flow, this paper provides a new algorithm named SIMPLEPC with two suggestions on enhancing SIMPLEC algorithm. On the one hand, an extra explicit prediction for velocity is applied. And on the other hand, the velocity-correction formulas are modified so that the corrected velocity and pressure can satisfy the original momentum equations like Eq. (6) rather than Eq. (7). These two suggestions can be implemented easily. Based on SIMPLEC, only two lines of code are added for the calculation of the second and third intermediate velocity.

In order to test the performance of SIMPLEPC, four experiments are carried out. It can be concluded from the numerical experiments that the performance of SIMPLEPC is better than that of SIMPLEC. In most cases in this paper, the iteration number of SIMPLEPC is less as well as time consumption. And particularly, SIMPLEPC behaves better when time step multiplier is lower. When the time step multiplier is less than 4 (the corresponding velocity relaxation factor is less than 0.8), the time consumption ratio is approximately less than 0.8. In other words, time consumption of SIMPLEC can be saved when employing SIMPLEPC. It is particularly gratifying that SIMPLEPC and SIMPLEC are similar in robustness with the change of grid number, dimensionless parameter, boundary conditions and the time step multiplier.

Conflict of interest

We wish to confirm that there are no known conflicts of interest associated with this publication.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 51736004 and 51776079).

References

[1] A. Guittet, M. Theillard, F. Gibou, A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees, J. Comput. Phys. 292 (2015) 215–238.
[2] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (10) (1972) 1787–1806.
[3] M. Darwish, I. Sraj, F. Moukalled, A coupled finite volume solver for the solution of incompressible flows on unstructured grids, J. Comput. Phys. 228 (1) (2009) 180–201.
[4] D. Shirokoff, R.R. Rosales, An efficient method for the incompressible Navier-Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary, J. Comput. Phys. 230 (23) (2011) 8619–8646.
[5] C.-N. Xiao, F. Denner, B.G.M. van Wachem, Fully-coupled pressure-based finite-volume framework for the simulation of fluid flows at all speeds in complex geometries, J. Comput. Phys. 346 (Supplement C) (2017) 91–130.
[6] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, Commun. Appl. Math. Comput. Sci. 11 (1) (2016) 79–141.
[7] P. Nithiarasu, C.B. Liu, An artificial compressibility based characteristic based split (CBS) scheme for steady and unsteady turbulent incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (23) (2006) 2961–2982.
[8] M. Napolitano, G. Pascazio, L. Quartapelle, A review of vorticity conditions in the numerical solution of the ζ–ψ equations, Comput. Fluids 28 (2) (1999) 139–185.
[9] Q. Li, K.H. Luo, Q.J. Kang, Y.L. He, Q. Chen, Q. Liu, Lattice Boltzmann methods for multiphase flow and phase-change heat transfer, Prog. Energy Combust. Sci. 52 (2016) 62–105.
[10] L. An, T.S. Zhao, Transport phenomena in alkaline direct ethanol fuel cells for sustainable energy production, J. Power Sources 341 (2017) 199–211.
[11] Z. Qu, H. Xu, W. Tao, Numerical simulation of non-equilibrium conjugate heat transfer in tubes partially filled with metallic foams, J. Therm. Sci. Technol. 7 (1) (2012) 151–165.
[12] P. Liu, N. Zheng, F. Shan, Z. Liu, W. Liu, Numerical study on characteristics of heat transfer and friction factor in a circular tube with central slant rods, Int. J. Heat Mass Transf. 99 (2016) 268–282.
[13] N. Zheng, P. Liu, F. Shan, Z. Liu, W. Liu, Heat transfer enhancement in a novel internally grooved tube by generating longitudinal swirl flows with multi-vortexes, Appl. Therm. Eng. 95 (2016) 421–432.
[14] F.M.M. Darwish, A unified formulation of the segregated class of algorithms for fluid flow at all speeds, Numer. Heat Transfer, Part B: Fundam. 37 (1) (2000) 103–139.
[15] R.W. Johnson, Handbook of Fluid Dynamics, second ed., CRC Press, Boca Raton, Florida, 2016.
[16] S.V. Patankar, A calculation procedure for two-dimensional elliptic situations, Numer. Heat Transfer 4 (4) (1981) 409–425.
[17] S.V. Patankar, Numerical HeatTransfer and Fluid Flow, Hemisphere, Washington D.C., 1980.
[18] J.P. Van Doormaal, G.D. Raithby, Enhancements of the simple method for predicting incompressible fluid flows, Numer. Heat Transfer 7 (2) (1984) 147–163.
[19] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1) (1986) 40–65.
[20] W. Tao, Z. Qu, Y. He, A novel segregated algorithm for incompressible fluid flow and heat transfer problems—clear (coupled and linked equations algorithm revised). Part I: Mathematical formulation and solution procedure, Numer. Heat Transfer, Part B: Fundam. 45 (1) (2004) 1–17.
[21] 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, Numer. Heat Transfer, Part B: Fundam. 53 (1) (2008) 1–17.
[22] I. Demirdzic, A.D. Gosman, R.I. Issa, M. Peric, A calculation procedure for turbulent flow in complex geometries, Comput. Fluids 15 (3) (1987) 251–273.
[23] E. Blosch, W. Shyy, R. Smith, The role of mass conservation in pressure-based algorithms, Numer. Heat Transfer, Part B: Fundam. 24 (4) (1993) 415–429.
[24] R.-H. Yen, C.-H. Liu, Enhancement of the SIMPLE algorithm by an additional explicit corrector step, Numer. Heat Transfer, Part B Fundam. 24 (1) (1993) 127–141.
[25] W.Q. Tao, H. Ozoe, A modified pressure-correction scheme for the SIMPLER method MSIMPLER, Numer. Heat Transfer: Part B: Fundam. 39 (5) (2001) 435–449.
[26] S.C. Xue, G.W. Barton, Implementation of boundary conditions and global mass conservation in pressure-based finite volume method on unstructured grids for fluid flow and heat transfer simulations, Int. J. Heat Mass Transf. 55 (19) (2012) 5233–5243.
[27] J. Li, Q. Zhang, Z.-Q. Zhai, An efficient SIMPLER-revised algorithm for incompressible flow with unstructured grids, Numer. Heat Transfer, Part B: Fundam. 71 (5) (2017) 425–442.

[28] W. Liu, J.B. Wang, Z.C. Liu, A method of fluid dynamic analysis based on Navier-Stokes equation and conservation equation on fluid mechanical energy, Int. J. Heat Mass Transf. 109 (2017) 393–396.
[29] Y. Liu, F. Wang, Y.T. Li, Fourier analysis of the SIMPLE serials, Numer Heat Tranf. B-Fundam. 69 (3) (2016) 197–216.
[30] R.L. Sani, P.M. Gresho, Résumé and remarks on the open boundary condition minisymposium, Int. J. Numer. Meth. Fluids 18 (10) (1994) 983–1008.
[31] E.O. Macagno, T.-K. Hung, Computational and experimental study of a captive annular eddy, J. Fluid Mech. 28 (1) (1967) 43–64.
[32] C. Prakash, S.V. Patankar, Combined free and forced convection in vertical tubes with radial internal fins, J. Heat Transfer 103 (3) (1981) 566–572.
[33] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (3) (1982) 387–411.
[34] T. Kondoh, Y. Nagano, T. Tsuji, Computational study of laminar heat transfer downstream of a backward-facing step, Int. J. Heat Mass Transf. 36 (3) (1993) 577–591.
[35] G. Barakos, E. Mitsoulis, D. Assimacopoulos, Natural convection flow in a square cavity revisited: laminar and turbulent models with wall functions, Int. J. Numer. Meth. Fluids 18 (7) (1994) 695–719.
[36] W.Q. Tao, Numerical Heat Transfer, second ed., Xi'an Jiaotong University Press, Xi'an, 2001.
[37] B. Latimer, A. Pollard, Comparison of pressure-velocity coupling solution algorithms, Numer. Heat Transfer 8 (6) (1985) 635–652.
[38] F.M. White, Fluid Mechanics, seventh ed., McGraw-Hill, New York, 2010.