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

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...
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 algo-
rithm, the author realizes that, after each outer iteration, the better
pressure and velocity simultaneously satisfy the continuity equa-
tion and the momentum equations, the better the economy and
the robustness are. Therefore, the SIMPLE algorithm can be opti-
mized 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 interme-
diate 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

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0
\]  

(1)

Momentum equation:

\[
\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) + S_u
\]  

(2)

General form:

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

(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 algo-
braic approximation becomes

\[
a_p \Phi_i = a_i \Phi_i + a_w \Phi_w + a_s \Phi_s + b
\]  

(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).

\[
a_w u_c = \sum a_{wb} u_{wb} + b (p_w - p_e) A_e
\]  

(6)

Introducing the relaxation factor \( \alpha \) 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.

\[
a_v u_c = \sum a_{vb} u_{vb} + b (p_v - p_e) A_e + \frac{1 - \alpha}{\alpha} a_v u_v^0
\]  

(7)

2.3. The velocity-correction formula

In order to satisfy both the momentum and continuity con-
straint, the modifications to velocity and pressure are imple-
ment in Eqs. (8) and (9).

\[
p = p^0 + p'
\]  

(8)

\[
u = u' + u^0
\]  

(9)
Ae de u ae sure correction equation can be written as Eq. (15).

where

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

Substituting the corrected velocity expressions like Eq. (13) for

and

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

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

where

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

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

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

However, iterative errors are not taken into account when deriving the pressure correction equation in the SIMPLEC 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 \( u_\varepsilon^* = u_\varepsilon^0 \) 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 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.

\[ a_e (u_\varepsilon^* + u_\varepsilon^c) = \sum a_{ib} (u_{ib}^* + u_{ib}^c) + b + [(p_b^0 - p_b^c) / A_e \times (p_b^0 - p_b^e)] \]

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

\[ a_e u_\varepsilon^c = S \sum a_{ib} u_{ib}^c + b + (p_b^0 - p_b^e) / A_e \times (p_b^0 - p_b^e) \]

Now, specifying \( S = \sum a_{ib} u_{ib}^c + A_e (p_b^0 - p_b^e) \) to simplify Eq. (24), it yields

\[ a_e u_\varepsilon^c = S + \sum a_{ib} u_{ib}^c + A_e (p_b^0 - p_b^e) \]

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 \( u_\varepsilon^c \) 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

\[ a_e u_\varepsilon^c = S + \sum a_{ib} u_{ib}^c + A_e (p_b^0 - p_b^e) \]

Specify \( u_\varepsilon^c = u_{\varepsilon 1} + u_{\varepsilon 2} \) where \( u_{\varepsilon 1} \) and \( u_{\varepsilon 2} \) satisfy Eqs. (27) and (28), respectively.

\[ u_{\varepsilon 1} = \alpha S / a_e \]

\[ u_{\varepsilon 2} = \frac{\sum a_{ib} u_{ib}^c + A_e (p_b^0 - p_b^e)}{a_e / \alpha - \sum a_{ib}} \]

Subtracting the term \( \sum a_{ib} u_{ib}^c \) from both sides of Eq. (28), similar to the SIMPLEC approximation, it yields

\[ \frac{a_e}{\alpha} u_{\varepsilon 2} = \sum a_{ib} \left( u_{ib}^c - u_{ib}^c \right) + A_e (p_b^0 - p_b^e) \]

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

\[ u_{\varepsilon 2} = \frac{\sum a_{ib} \left( u_{ib}^c - u_{ib}^c \right)}{a_e / \alpha - \sum a_{ib}} + \frac{d_e \left( p_b^0 - p_b^e \right)}{a_e / \alpha - \sum a_{ib}} \]

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

\[ u_\varepsilon^c = \frac{\sum a_{ib} u_{ib}^c - \sum a_{ib} u_{ib}^c}{a_e / \alpha - \sum a_{ib}} + \alpha S / a_e + d_e \left( p_b^0 - p_b^e \right) \]

Assuming that \( \sum a_{ib} u_{ib}^c \approx \sum a_{ib} u_{ib}^c \), the first term in the right hand of Eq. (31) can be dropped and Eq. (31) becomes

\[ u_\varepsilon^c = \alpha S / a_e + d_e \left( p_b^0 - p_b^e \right) \]

or

\[ u_\varepsilon^c = u_\varepsilon^* + u_\varepsilon^c = u_\varepsilon^* + d_e \left( p_b^0 - p_b^e \right) \]
where
\[ u_t^{**} = u_t^{**} + zS/\alpha_t = u_t^{**}(1 - x) + \left( \sum_{ab} u_{ab}^{**} + b + (p_A - p_A^0)/\alpha_t \right)/\alpha_t. \]

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 \( xS/\alpha_t \). 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.

\[ a_p p'_p = a_p p'_p + a_p p'_w + a_p p'_n + a_p p'_w + b \] (34)

where
\[ a_p = a_e + a_w + a_n + a_t \times (\rho Ad), a_w = (\rho Ad)_w, \]
\[ a_n = (\rho Ad)_n, a_t = (\rho Ad)_t, \]
\[ b = (p u)\tau (p u)\tau - (p u)\tau (p v)\tau + (p v)\tau (p v)\tau - (p v)\tau (p u)\tau \]

Solving the pressure correction equation, the pressure correction \( p_c \) is obtained. The corrected pressure can be calculated by Eq. (35).

\[ p_p = p_p^0 + p_p^c \] (35)

The corrected velocity can be obtained with the following equation.

\[ u_p = u_p^{**} + u_p^{**} + d_e(p_p - p_p^0) \] (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, SIMPLEPC 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 SIMPLEPC 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

\[ TCR = \frac{T_{SIMPLEC}}{T_{SIMPLEC}} \] (37)

where \( T_{SIMPLEC} \) is the time consumption of SIMPLEC 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).

\[ E = \frac{x}{1 - x} \quad (0 < x < 1) \] (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 \( 10^{-3} \text{–} 10^{-5} \) [37]. And the appropriate convergence criteria will be specified in different cases.

\[ R_{SM} = \sum_{i=1}^{n} d_i/q_m \] (39)
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.

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

<table>
<thead>
<tr>
<th>( a )</th>
<th>0.333</th>
<th>0.500</th>
<th>0.750</th>
<th>0.833</th>
<th>0.875</th>
<th>0.900</th>
<th>0.917</th>
<th>0.925</th>
<th>0.938</th>
<th>0.944</th>
<th>0.950</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.5</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>202×202</td>
<td>SIMPLEC</td>
<td>19242</td>
<td>9606</td>
<td>3201</td>
<td>1892</td>
<td>1456</td>
<td>1173</td>
<td>950</td>
<td>890</td>
<td>812</td>
<td>756</td>
</tr>
<tr>
<td></td>
<td>SIMPLEPC</td>
<td>10703</td>
<td>5774</td>
<td>2305</td>
<td>1448</td>
<td>1175</td>
<td>933</td>
<td>814</td>
<td>747</td>
<td>695</td>
<td>625</td>
</tr>
<tr>
<td>132×132</td>
<td>SIMPLEC</td>
<td>10287</td>
<td>5139</td>
<td>1728</td>
<td>1046</td>
<td>764</td>
<td>617</td>
<td>526</td>
<td>488</td>
<td>437</td>
<td>398</td>
</tr>
<tr>
<td></td>
<td>SIMPLEPC</td>
<td>5725</td>
<td>3091</td>
<td>1208</td>
<td>829</td>
<td>632</td>
<td>524</td>
<td>445</td>
<td>402</td>
<td>368</td>
<td>371</td>
</tr>
<tr>
<td>82×82</td>
<td>SIMPLEC</td>
<td>4817</td>
<td>2408</td>
<td>816</td>
<td>486</td>
<td>374</td>
<td>301</td>
<td>268</td>
<td>249</td>
<td>217</td>
<td>218</td>
</tr>
<tr>
<td></td>
<td>SIMPLEPC</td>
<td>2683</td>
<td>1451</td>
<td>586</td>
<td>379</td>
<td>290</td>
<td>243</td>
<td>224</td>
<td>199</td>
<td>201</td>
<td>201</td>
</tr>
<tr>
<td>52×52</td>
<td>SIMPLEC</td>
<td>2233</td>
<td>1119</td>
<td>381</td>
<td>230</td>
<td>169</td>
<td>155</td>
<td>136</td>
<td>150</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>SIMPLEPC</td>
<td>1245</td>
<td>675</td>
<td>276</td>
<td>177</td>
<td>149</td>
<td>123</td>
<td>124</td>
<td>142</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>32×32</td>
<td>SIMPLEC</td>
<td>946</td>
<td>476</td>
<td>167</td>
<td>104</td>
<td>91</td>
<td>112</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>SIMPLEPC</td>
<td>529</td>
<td>288</td>
<td>118</td>
<td>83</td>
<td>81</td>
<td>104</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

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

$$Re = \frac{\rho u L}{\mu}$$

The different uniform grids systems $32 \times 32$, $52 \times 52$, $82 \times 82$, $132 \times 132$, and $202 \times 202$ are employed to analyze the calculation economy with the grids from coarse to dense. The convergence criteria are set as $R_{SM} \leq 10^{-7}$, $R_{SU} \leq 10^{-7}$, $R_{SV} \leq 10^{-5}$ 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

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

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 $202 \times 202$. The convergence criteria are $R_{SM} \leq 10^{-7}$, $R_{SU} \leq 10^{-7}$, $R_{SV} \leq 10^{-5}$, $R_{ST} \leq 10^{-7}$ 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.

$\overline{q}_y = 0$

$$\overline{q}_y = 0$$

$q_m = \int_a^b \rho |u_m| \, dy$ (41)

$\overline{q}_y = 0$

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

<table>
<thead>
<tr>
<th>Table 2 Mean Nusselt number at hot wall.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ra$</td>
</tr>
<tr>
<td>Barakos [35]</td>
</tr>
<tr>
<td>SIMPLEC</td>
</tr>
<tr>
<td>SIMPLEPC</td>
</tr>
</tbody>
</table>

Fig. 6. The TCR of case verification 2.

<table>
<thead>
<tr>
<th>Table 3 The iteration number of natural convection in a square cavity.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ra=10^4$</td>
</tr>
<tr>
<td>SIMPLEC</td>
</tr>
<tr>
<td>SIMPLEPC</td>
</tr>
<tr>
<td>$Ra=10^5$</td>
</tr>
<tr>
<td>SIMPLEC</td>
</tr>
<tr>
<td>SIMPLEPC</td>
</tr>
<tr>
<td>$Ra=10^6$</td>
</tr>
<tr>
<td>SIMPLEC</td>
</tr>
<tr>
<td>SIMPLEPC</td>
</tr>
</tbody>
</table>
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).

\[
Re = \frac{2 \mu u_m r}{\mu} \quad (44)
\]

\[
u = \frac{1}{2} u_m (1 - (r/R)^2) \quad (45)
\]

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 RsM ≤ 10^{-7}, Rsu ≤ 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

---

**Table 4**

The iteration number of sudden expansion in the tube.

<table>
<thead>
<tr>
<th>a</th>
<th>0.333</th>
<th>0.500</th>
<th>0.750</th>
<th>0.833</th>
<th>0.875</th>
<th>0.900</th>
<th>0.917</th>
<th>0.929</th>
<th>0.938</th>
<th>0.944</th>
<th>0.950</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIMPLEC</td>
<td>SIMPLEPC</td>
<td>SIMPLEC</td>
<td>SIMPLEPC</td>
<td>SIMPLEC</td>
<td>SIMPLEPC</td>
<td>SIMPLEC</td>
<td>SIMPLEPC</td>
<td>SIMPLEC</td>
<td>SIMPLEPC</td>
<td>SIMPLEC</td>
</tr>
<tr>
<td>Uniform</td>
<td>434</td>
<td>224</td>
<td>76</td>
<td>48</td>
<td>36</td>
<td>29</td>
<td>25</td>
<td>24</td>
<td>21</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>Linear</td>
<td>473</td>
<td>226</td>
<td>78</td>
<td>52</td>
<td>39</td>
<td>30</td>
<td>26</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>Parabolic</td>
<td>443</td>
<td>223</td>
<td>78</td>
<td>49</td>
<td>38</td>
<td>30</td>
<td>26</td>
<td>23</td>
<td>21</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

---

**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.
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.

\[
Re = \frac{\rho u m H}{\mu}
\]

\[
u = \frac{6 u m}{H^2} (Hy - y^2)
\]

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 \times 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].

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

where \( \mu_0 = 0.001792 \text{ kg/(m s)} \), \( T_0 = 273.16 \text{ K} \). Other physical parameters such as density (\( \rho \)), conductivity (\( \lambda \)), and specific heat (\( c_p \)) are constant and depend on the characteristic temperature. The
convergence criteria are $R_M < 10^{-7}$, $R_u < 10^{-5}$, $R_v < 10^{-5}$, $R_T < 10^{-7}$ 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 SIMPLE 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 SIMPLE, 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 SIMPLEPC 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


