The segregated algorithm-IDEAL (inner doubly-iterative efficient algorithm for linked-equations) is an efficient and stable algorithm. In this algorithm, there exist inner doubly-iterative processes for pressure equation, which almost completely overcome two approximations in SIMPLE algorithm. Thus the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate and stability of iteration process. In this paper, implementation of the IDEAL algorithm on a 3D collocated grid system is conducted. The interface velocity is calculated by the modified momentum interpolation method (MMIM), by which the converged result is independent of the under-relaxation factor. Finally, five three-dimensional incompressible fluid flow and heat transfer problems are provided to compare the convergence rate and robustness between the IDEAL and three other most widely-used algorithms (SIMPLER, SIMPLEC and PISO). By the comparison it can be concluded that the IDEAL algorithm is more robust and efficient than the three other algorithms.
Van Doormaal and Raithby proposed the SIMPLEC algorithm\cite{cite7}, in which by changing the definition of the coefficients of the pressure-correction equation, the effects of dropping the neighboring grid velocity corrections (the second approximation in the SIMPLE algorithm) are partially compensated. Van Doormaal and Raithby also proposed the SIMPLEX algorithm\cite{cite8,cite9}. In the SIMPLEX algorithm, by solving a set of algebraic equations for the coefficient $d$ in the velocity-correction equation, the effects of dropping the velocity corrections of the neighboring grids are also taken into account to some degree. However, an additional assumption is introduced: the corrections of pressure difference across every interface of the main control volume are the same. PISO algorithm\cite{cite10} was proposed by Issa, which implements two correction steps of pressure correction. Yen and Liu\cite{cite11} proposed the explicit correction step method to accelerate the convergence by making the velocity explicitly satisfy the momentum equation. In summary, more than ten variants of the SIMPLE algorithm are available in the literature, but no one has successfully overcome the two assumptions. And it can be observed that once the pressure correction term, $p'$, is introduced to improve the pressure by adding the correction term to the original one, it will inevitably lead to the drop of the neighboring grid points in the velocity correction equation so as to make the pressure correction equation manageable. Thus, in order to overcome the second approximation of the SIMPLE, we should directly improve the pressure, rather than introducing the pressure correction term. This is the successful point of the algorithm CLEAR\cite{cite12,cite13}. However, the robustness of the CLEAR algorithm is somewhat deteriorated as indicated in\cite{cite14} where a modified algorithm, named by CLEARER was proposed. However, by reintroduction of the correction terms into the algorithm, the fully-implicit character has been destroyed in the CLEARER algorithm. In order to retain the fully-implicit feature while further enhance the robustness and convergence characteristics, on the basis of the CLEAR algorithm, the IDEAL algorithm (Inner Doubly-iterative Efficient Algorithm for Linked-equations)\cite{cite15,cite16} was proposed by the present authors. In the algorithm there exist inner doubly-iterative processes for pressure equation at each iteration level, which almost completely overcome the two approximations in the SIMPLE algorithm. Thus, the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate and stability of solution process.

In this paper, the IDEAL algorithm is extended to a 3D collocated grid system. The interface velocity is calculated by the modified momentum interpolation method (MMIM), by which the converged result is independent of the under-relaxation factor. For the irregular computation domain, the domain extension method\cite{cite17} is applied. Finally, five three-dimensional incompressible fluid flow and heat transfer problems are provided to compare the convergence rate and robustness between the IDEAL and three other most widely-used algorithms (SIMPLER, SIMPLEC and PISO).

1 Momentum interpolation method

The staggered grid is widely employed in computational fluid dynamics/numerical heat transfer literature because it can efficiently guarantee the coupling between velocity and pressure. However, it shows that it is inconvenient for code development in unstructured grid and curvilinear body-fitted grids, especially for 3D computation. On collocated grid, such complication can be greatly alleviated. The crucial issue in using collocated grid is how to eliminate the decoupling between pressure and velocity. In 1983, the momentum interpolation method (MIM) on collocated grid was first presented by Rhie and Chow\cite{cite18} to avoid the decoupling problem. However, MIM method has the weakness that the solution is under-relaxation factor-dependent to some extent. In order to overcome this weakness, an easy technique by Kobayashi and Pereira\cite{cite19} was proposed, in which the under-relaxation factor is set as 1 before momentum interpolation is implemented, but this may decrease the robustness of the algorithm to some extent. Hence, we can see that in order to make a reliable and efficient computation on a collocated grid, the following three aspects should be guaranteed: (1) the algorithm should avoid the checkerboard pressure distribution; (2) the converged result should be independent of the under-relaxation factor; (3) the algorithm should possess the required robustness. The modified momentum interpolation method (MMIM)\cite{cite20}, proposed by Majumdar, possesses the above-mentioned three features, so in this study, MMIM is applied to interpolate the interface velocity.
In the following, a brief description of the governing equations and the discretization procedure will be presented. The system for a collocated grid is shown in Figure 1. For simplicity of presentation, we take incompressible laminar steady flow in Cartesian coordinates as an example. The governing equations are as follows.

**Continuity equation:**

\[
\frac{\partial (\rho u_f)}{\partial x} + \frac{\partial (\rho v_f)}{\partial y} + \frac{\partial (\rho w_f)}{\partial z} = 0.\tag{1}
\]

**Momentum equation:**

\[
\frac{\partial (\rho u_f u)}{\partial x} + \frac{\partial (\rho v_f v)}{\partial y} + \frac{\partial (\rho w_f w)}{\partial z} = -\frac{\partial p}{\partial x} + \eta \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + S_u, \tag{2}
\]

\[
\frac{\partial (\rho u_f v)}{\partial x} + \frac{\partial (\rho v_f v)}{\partial y} + \frac{\partial (\rho w_f w)}{\partial z} = -\frac{\partial p}{\partial y} + \eta \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + S_v, \tag{3}
\]

\[
\frac{\partial (\rho u_f w)}{\partial x} + \frac{\partial (\rho v_f w)}{\partial y} + \frac{\partial (\rho w_f w)}{\partial z} = -\frac{\partial p}{\partial z} + \eta \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + S_w. \tag{4}
\]

where the subscript \(f\) refers to the interface, which is used to distinguish the nodal velocity \(u, v, w\) and the interface velocity \(u_f, v_f, w_f\).

The discretization equation for the \(u\) component takes the following form, into which under-relaxation factor is incorporated:

\[
\frac{u_p}{u_f} = \sum a_{nh} u_{nh} + b + A_p (p_w - p_e) + \frac{1 - \alpha_u}{\alpha_u} a_p u^0_p. \tag{5}
\]

The expressions of the coefficients \(a\) and source terms \(b\) depend on the discretized schemes, and have been well documented in literature \cite{121}. For the simplicity of presentation, they are not shown here.

From Eq. (5), we can get

\[
u_p = \alpha_u \left( \sum a_{nh} u_{nh} + b \right) + \alpha_u A_p (p_w - p_e) + (1 - \alpha_u) u^0_p. \tag{6}
\]

The interface pressures are linearly interpolated from the neighboring nodes:

\[
p_w = f_w^+ p_p + (1 - f_w^+) p_w, \quad p_e = f_e^+ p_e + (1 - f_e^+) p_p,
\]

\[
f_w^+ = \frac{\Delta x_w}{2 \delta_w}, \quad f_e^+ = \frac{\Delta x_e}{2 \delta_e}.
\]

We define

\[
u_p = \sum a_{nh} u_{nh} + b \frac{a_p}{a_p} \tag{7}
\]

\[
d_p = \frac{\alpha_u A_p}{a_p} \tag{8}
\]

Then we get

\[
u_p = \alpha_u \bar{u}_p + d_p (p_w - p_e) + (1 - \alpha_u) u^0_p. \tag{9}
\]

Now, mimicking Equation (9), we express the interface velocity as follows:

\[
u_e = \alpha_e \bar{u}_e + d_e (p_p - p_E) + (1 - \alpha_e) u^0_e, \tag{10}
\]

where \(\bar{u}_e, d_e\) are linearly interpolated from the neighboring nodes:

\[
\bar{u}_e = f_e^+ \bar{u}_E + (1 - f_e^+) \bar{u}_p, \tag{11}
\]

\[
d_e = f_e^+ d_E + (1 - f_e^+) d_p. \tag{12}
\]

Equation (10) is the modified momentum interpolation method (MMIM), in which the under relaxation is based on the interface velocity, \(u_e^0\). Different from the MMIM, the under relaxation is based on the \(f_e^+ u_e^0 + (1 - f_e^+) u_p^0\) in the MIM. By using \(u_e^0\), the converged result is independent of the under-relaxation factor.

Equation (10) can be rewritten as

\[
u_e = \bar{u}_e + d_e (p_p - p_E), \tag{13}
\]

where \(\bar{u}_e\) is the pseudo velocity.
\[ \tilde{u}_e = \alpha_u \bar{\bar{u}}_e + (1 - \alpha_u) u^0_e. \]  

(14)

The momentum discretization form and the procedure for calculating the interface velocity for the \(v\) and \(w\) components are the same as the \(u\) component. Here for the simplicity of presentation, they are not introduced. Thereinafter, we will use the \(u\)-component equations to represent all the equations for the \(u\), \(v\), \(w\) components.

2 Solution procedure of the IDEAL algorithm

Figure 2 shows the framework of the iteration process of the IDEAL algorithm in detail. At each iteration level of the IDEAL algorithm there exist two inner iteration processes, or inner doubly-iterative processes, for pressure field solution. The first inner iteration process for pressure equation almost completely overcomes the first approximation in the SIMPLE algorithm. The second inner iteration process almost completely overcomes the second approximation in the SIMPLE algorithm. The solution procedure of the IDEAL algorithm is presented as follows.

Step-1: Assume initial nodal and interface velocity fields \(u^0_p, v^0_p, w^0_p, u^0_e, v^0_e, w^0_e\).

Step-2: Calculate the coefficients \(a\) and source terms \(b\) of the discretized momentum Eq. (5) for the \(u\), \(v\), \(w\) components by the initial velocity field.

Step-3: Calculate the pseudo-velocities \(\tilde{u}_e^0, v_n^0, \tilde{w}_t^0\) from Eq. (14). The calculation procedure of the pseudo-velocity \(\tilde{u}_e^0\) is as follows. Firstly, \(\bar{\bar{u}}_p^0, \bar{\bar{u}}_E^0\) are calculated from Eq. (7). Then, \(\tilde{u}_e^0\) is linearly interpolated by Eq. (11). Finally, \(\tilde{u}_e^0\) can be obtained by Eq. (14).

Step-4: Solve the pressure Equation (15), and obtain the temporary pressure \(p^{\text{Temp}}\).

\[ \frac{a_p}{\alpha_p} p^{\text{Temp}}_p = \sum a_{nb} p^{\text{Temp}}_{nb} + b, \]

\[ a_p = a_E + a_V + a_N + a_S + a_I + a_B, \]

\[ a_E = (\rho \text{Ad})_e, \quad a_V = (\rho \text{Ad})_V, \quad a_N = (\rho \text{Ad})_N, \]

\[ a_S = (\rho \text{Ad})_S, \quad a_I = (\rho \text{Ad})_I, \quad a_B = (\rho \text{Ad})_B, \]

\[ b = (\rho \tilde{u}_e^0 A)_e - (\rho \tilde{u}_e^0 A)_s + (\rho v^0 A)_s - (\rho v^0 A)_s + (\rho \tilde{w}_t^0 A)_t - (\rho \tilde{w}_t^0 A)_t + (1 - \alpha_p) \frac{d_p}{\alpha_p} p^{\text{Temp}}_p. \]

Equation (15) is obtained by substituting Eq. (13) for the \(u\), \(v\), \(w\) components into the continuity Eq. (1).

Figure 2  The framework of the solution process of the IDEAL algorithm.

\[ \text{SUN DongLiang et al. Chinese Science Bulletin | March 2009 | vol. 54 | no. 6 | 929-942} \]
In the first inner iteration process for pressure equation, the pressure under-relaxation factor \( \alpha_p \) is incorporated into the pressure Eq. (15). The under-relaxation factor is used to make the solution process more stable for some very complicated cases. Generally speaking, the solution process of the IDEAL algorithm is stable enough, so for most cases the pressure in Eq. (15) need not be under-relaxed and the pressure under-relaxation factor \( \alpha_p \) is set as 1.

**Step-5:** Calculate the temporary nodal velocities \( u_{p,\text{Temp}}, v_{p,\text{Temp}} \) and \( w_{p,\text{Temp}} \) from Eq. (9) by the temporary pressure \( p_{\text{Temp}} \).

**Step-6:** Regard \( u_{p,\text{Temp}}, v_{p,\text{Temp}}, w_{p,\text{Temp}} \) and \( p_{\text{Temp}} \) calculated in Step-4 and Step-5 as the temporary nodal velocity and pressure of the previous inner iteration step, denoted by \( u_{p,\text{Pre},\text{Temp}}, v_{p,\text{Pre},\text{Temp}}, w_{p,\text{Pre},\text{Temp}} \) and \( p_{\text{Pre},\text{Temp}} \).

Return to Step-3, and then all the superscripts 0 in steps 3 and 4 are replaced by \( P_{\text{Temp}} \). Repeat such iteration process composed of steps 3, 4 and 5 until the iteration times are equal to the pre-specified times \( N_2 \). After the first inner iteration process for pressure equation is finished, the final temporary nodal velocities are regarded as the final nodal velocities \( u_{p,\text{f}}, v_{p,\text{f}} \) and \( w_{p,\text{f}} \) of the current iteration level. Then, the final interface velocities \( u_e, v_n \) and \( w_i \) can be calculated by Eq. (13) by the final nodal velocities and pressure.

**Step-7:** Solve the momentum Eq. (5) by the initial velocity and pressure \( p^* \), and obtain the intermediate nodal velocities \( u^*, v^*, w^* \).

**Step-8:** Calculate the pseudo-velocities \( u_e^*, v_n^*, w_i^* \) from Eq. (14).

**Step-9:** Solve the pressure Eq. (16), and obtain the temporary pressure \( p_{\text{Temp}} \).

\[
a_p P_{\text{Temp}} = \sum a_{nb} P_{nb,\text{Temp}} + b,
\]

\[
a_p = a_E + a_W + a_N + a_S + a_T + a_B,
\]

\[
a_E = (\rho Ad)_e, \quad a_W = (\rho Ad)_w, \quad a_N = (\rho Ad)_n,
\]

\[
a_S = (\rho Ad)_s, \quad a_T = (\rho Ad)_t, \quad a_B = (\rho Ad)_b,
\]

\[
b = (\rho u_e^* A)_e - (\rho u_n^* A)_n - (\rho w_i^* A)_i.
\]

**Step-10:** Calculate the temporary nodal velocities \( u_{p,\text{Temp}}, v_{p,\text{Temp}} \) and \( w_{p,\text{Temp}} \) from Eq. (9) by the temporary pressure \( p_{\text{Temp}} \).

**Step-11:** Regard \( u_{p,\text{Temp}}, v_{p,\text{Temp}}, w_{p,\text{Temp}} \) and \( p_{\text{Temp}} \) calculated in Step-9 and Step-10 as the temporary nodal velocity and pressure of the previous inner iteration step, denoted by \( u_{p,\text{Pre},\text{Temp}}, v_{p,\text{Pre},\text{Temp}}, w_{p,\text{Pre},\text{Temp}} \) and \( p_{\text{Pre},\text{Temp}} \).

It is interesting to note that in the IDEAL algorithm, as in the algorithms SIMPLER and CLEAR, the pressure field used to solve the momentum equations, i.e. \( p^* \), is solved by the pressure equation. Since the algebraic equation is solved iteratively, an initial pressure field is required, and the goodness of this initial field has a profound effect on the solution convergence. The numerical practice provided in ref. [6] revealed this important effect. Our numerical practices show that if the pressure results of the first inner iteration are taken as the initial field for the next level solution, the total solution procedure can be somewhat enhanced.

It should be noted that the second inner iteration process seems to be the continuation of the first inner iteration process. In fact, these two inner iteration processes are different in essence. The first inner iteration process for pressure equation is for getting an initial pressure to solve the momentum equation. The second inner iteration process for pressure equation is for getting the final velocity of the current iteration level.
In the IDEAL algorithm, the first inner iteration times \( N_1 \) and the second inner iteration times \( N_2 \) (hereafter \( N_1 \& N_2 \)) can be adjusted. \( N_1 \& N_2 \) should be increased with the increase of the velocity under-relaxation factor. At a larger velocity under-relaxation factor the solution process may become very unstable, therefore, the inner iteration times need to be increased to ensure the convergence of solution process and to enhance the robustness.

3 Comparison conditions and convergence criterion

For making meaningful comparisons of the four algorithms, numerical comparison conditions and convergence criterion should be specified. In our study, the comparison conditions and convergence criterion include:

1. Discretization scheme

   In order to guarantee the stability and accuracy of the numerical solution, SGSD scheme [21] is adopted, which is at least of second-order accuracy and absolutely stable. For stability of the solution process, the deferred-correction method is adopted, which was proposed in ref. [22] and later enhanced in ref. [23].

2. Solution method of the algebraic equations

   The algebraic equations are solved by the alternative direction implicit method (ADI).

3. Under-relaxation factor

   In the SIMPLER and IDEAL algorithms, the pressure under-relaxation factor is set as 1.0. In the SIMPLEC and PISO algorithms, the pressure need not be under-relaxed at all. For the four algorithms, the same value is adopted for the velocity and temperature under-relaxation factors. For the convenience of presentation, the time step multiple \( E \) is used in the following presentation, which relates to the under-relaxation factor \( \alpha \) by Eq. (17):

   \[
   E = \frac{\alpha}{1 - \alpha} \quad (0 < \alpha < 1).
   \]  

   Some correspondence between \( \alpha \) and \( E \) is presented in Table 1. It can be seen that with the time step multiple, we have a much wider range to show the performance of the algorithm in the high-value region of the under-relaxation factor.

4 Grid system

   For each problem the same uniform grid system is used for execution of the four algorithms. The details of each grid system will be presented individually.

5 Convergence criterion

   The adopted convergence criterion requires that both the relative maximum mass residual \( R_{\text{Mass}} \), and the relative maximum \( u \), \( v \), \( w \)-component momentum residuals \( R_{\text{Mom}} \), \( R_{\text{Mom}} \), \( R_{\text{Mom}} \) are less than some pre-specified small values [17].

4 Numerical comparisons

In the following, comprehensive comparisons are made among the SIMPLER, SIMPLEC, PISO and IDEAL algorithms for five three-dimensional problems of fluid flow and heat transfer, which are: (1) lid-driven cavity flow in a cubic cavity (problem 1); (2) lid-driven cavity flow in a cubic cavity with complicated structure (problem 2); (3) laminar fluid flow over a backward-facing step (problem 3); (4) laminar fluid flow through a duct with complicated structure (problem 4); (5) natural convection in a cubic cavity (problem 5).

Problem 1 to problem 4 is fluid flow problem. Among these four problems, problem 1 and problem 2 belong to closed system; problem 3 and problem 4 belong to open system. Problem 5 is a velocity- temperature coupling problem. All of the five problems are based on the following assumptions: laminar, incompressible, steady-state, and constant fluid property. For the fifth problem, the Boussinesq assumption was adopted [24].

4.1 Problem 1: Lid-driven cavity flow in a cubic cavity

Lid-driven cavity flow in a cubic cavity is served in CFD/NHT as a benchmark problem for testing numerical procedures for three-dimensional fluid flows [25–27]. The flow configuration of lid-driven cavity flow is shown in Figure 3. Calculations are conducted for \( Re = 100 \rightarrow 1000 \) and grid numbers = \( 32 \times 32 \times 32 \rightarrow 82 \times 82 \times 82 \), and the allowed residuals \( R_{\text{Mass}} \), \( R_{\text{Mom}} \), \( R_{\text{Mom}} \) and \( R_{\text{Mom}} \) should be all less than \( 10^{-8} \). The Reynolds number is defined by

\[
Re = \frac{u_{\text{lid}} H}{v}.
\]
In Figure 4 the velocity profiles along the central lines on the plane \( z = 0.5H \) are presented. As shown in this figure, the results calculated by IDEAL algorithm are in excellent agreement with those reported by Tang et al. [27]. This comparison gives some support to the reliability of the proposed 3D IDEAL algorithm and the developed code.

Figures 5—7 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Reynolds numbers of problem 1. The inner iteration times N1\&N2 in the IDEAL algorithm are displayed at the top of these figures. For example 1\&1 and 1\&2 at the top of

**Figure 3** Flow configuration of lid-driven cavity flow in a cubic cavity (problem 1).

**Figure 4** Comparison of velocity profiles \( u \) and \( v \) along the central axes on plane \( z=0.5H \) for \( Re=1000 \) (problem 1).

**Figure 5** Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 1, grid number=32×32×32).

**Figure 6** Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 1, grid number=52×52×52).

**Figure 7** Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 1, grid number=64×64×64).
Figure 7 Comparison of computation time and robustness of IDEAL, SIMPLER and PISO algorithms (problem 1, grid number=82×82×82).

Figure 5(a) show that in the two ranges of $E$ the two inner iterative times are 1&1 and 1&2, respectively. From the three figures, the following three features may be noted. First, N1 & N2 increase with the increase of time step multiple, i.e., with the under-relaxation factor. Second, among the four algorithms compared, the IDEAL algorithm is far more robust than the SIMPLER, SIMPLEC and PISO algorithms, and it can converge almost at any time step multiple for any case in problem 1. Third, for the consumed computation time, the SIMPLEC algorithm needs the largest, even in the case of grid number=82×82×82 it diverges and can not acquire the converged result. The SIMPLER and PISO algorithms come next. The IDEAL algorithm needs the least.

Table 2 Reduced ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples (problem 1)

<table>
<thead>
<tr>
<th>Grid Numbers</th>
<th>32×32×32</th>
<th>52×52×52</th>
<th>82×82×82</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Re$</td>
<td>100</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLER</td>
<td>36.4%</td>
<td>33.8%</td>
<td>36.7%</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLEC</td>
<td>59.4%</td>
<td>57.4%</td>
<td>55.1%</td>
</tr>
<tr>
<td>Reducing ratio over PISO</td>
<td>32.7%</td>
<td>36.4%</td>
<td>33.2%</td>
</tr>
</tbody>
</table>

Table 2 shows the reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples for different cases of problem 1. When each method works at its own optimal time step multiple, the IDEAL algorithm can reduce the computation time by 33.8%−46.6% over the SIMPLER algorithm, by 55.1−59.4 over the SIMPLEC algorithm and by 32.7%−43.4% over PISO algorithm for problem 1.

4.2 Problem 2: Lid-driven cavity flow in a cubic cavity with complicated structure

Problem 1 belongs to the simple closed system. The IDEAL algorithm shows its significant advantages over the SIMPLER, SIMPLEC and PISO algorithms for this simple closed system. In order to show the better performance of the IDEAL algorithm superior to the other three algorithms for a complicated closed system, problem 2 is especially designed. The flow configuration of problem 2 is shown in Figure 8. Three blocks of baffle plates are inserted into the cubic cavity to make the flow configuration more complicated.

Calculations are conducted for $Re=100−800$ and grid numbers = 52×52×52−82×82×82. The allowed residuals $Rs_{\text{Mass}}$, $Rs_{\text{Mom}}$, $Rs_{\gamma\text{Mom}}$ and $Rs_{\#\text{Mom}}$ should be all less than $10^{-8}$. The Reynolds number is defined by

$$Re = \frac{u_{\text{lid}}H}{\nu}. \quad (19)$$

In Figure 9 the velocity profiles $u$ along the central line $y$ on the plane $z = 0.5H$ from the four algorithms are
presented. The results calculated by the IDEAL algorithm are in excellent agreement with those calculated by the other three algorithms. Figures 10 and 11 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Reynolds numbers of problem 2. From these two figures, we can find that the relative performances of different algorithms in the complicated closed system are almost the same as those in the simple closed system. Thus, the IDEAL algorithm also shows its advantages for complicated closed systems.

Table 3 shows the reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples for different cases of problem 2. When each method works at its own optimal time step multiple, the IDEAL algorithm can reduce the computation time by 29.0% – 40.6% over the SIMPLER algorithm, by 52.5% – 61.4% over the SIMPLEC algorithm and by 27.1% – 37.8% over the PISO algorithm.

Table 3  Reduced ratio of computation time of IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples (problem 2)

<table>
<thead>
<tr>
<th>Grid Numbers</th>
<th>52×52×52</th>
<th>82×82×82</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re</td>
<td>100</td>
<td>500</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLER</td>
<td>29.6%</td>
<td>34.9%</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLEC</td>
<td>52.5%</td>
<td>61.4%</td>
</tr>
<tr>
<td>Reducing ratio over PISO</td>
<td>27.5%</td>
<td>30.5%</td>
</tr>
</tbody>
</table>
27.1%—37.8% over PISO algorithm for problem 2.

4.3 Problem 3: Laminar fluid flow over a backward-facing step

Laminar fluid flow over a backward-facing step shown in Figure 12 belongs to simple open system. It is another typical configuration widely adopted in computational fluid dynamics study. Calculations are conducted for $Re = 100 − 300$ and grid numbers $= 80 \times 20 \times 20 − 160 \times 41 \times 41$. The residuals $R_{\text{Mass}}, R_{\text{Mom}}$, $R_{\text{Mom}}$ and $R_{\text{Mom}}$ are all set to be less than $10^{-7}$. The Reynolds number is defined by

$$Re = \frac{u_{\text{in}} H}{v} \quad (20)$$

Figures 13 and 14 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEX and PISO algorithms for different grid numbers and different Reynolds numbers of problem 3. As shown in these two figures, the SIMPLER algorithm has the worst robustness, and the robustness of the SIMPLEX and PISO algorithms is a bit better. The IDEAL algorithm is the best. From Figures 13 and 14 we can find that the IDEAL algorithm can converge almost at any time step multiple for any case of problem 3. As far as the consumed computation time is concerned, the SIMPLEX algorithm needs the largest, and the SIMPLER and PISO algorithms come next. The IDEAL algorithm needs the least.

Table 4 shows the reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEX and PISO algorithms at their own optimal time step multiples for different cases of problem 3. When each method works at its own optimal time step multiple, the IDEAL algorithm can reduce the computation time by 45.2%—50.7% over the SIMPLER algorithm, by 70.7%—81.1% over the SIMPLEX algorithm and by 34.0%—41.6% over PISO algorithm for problem 3.
4.4 Problem 4: Laminar fluid flow through a duct with complicated structure

Laminar fluid flow through a duct with complicated structure belongs to complicated open system. This problem is adopted to examine whether the IDEAL algorithm is still superior to the SIMPLER, SIMPLEC and PISO algorithms in a complicated open system. The flow configuration of problem 4 is shown in Figure 15. Three blocks of baffle plates are inserted into the duct to make the flow configuration more complicated.

Calculations are conducted for \( Re = 100 - 500 \), grid numbers = 150×20×20 – 190×29×29, and the residuals \( R_{\text{Mass}} \), \( R_{\text{Mom}} \), \( R_{\text{MOM}} \), \( R_{\text{Mom}} \) and \( R_{\text{MOM}} \) are all set to be less than \( 10^{-7} \). The Reynolds number is defined by

\[
Re = \frac{u_L H}{\nu}.
\]

Figures 16 and 17 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Reynolds numbers of problem 4. From these two figures, we can find that the IDEAL algorithm also shows its advantages for complicated open systems.

Table 5 shows the reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples for different cases of problem 4. When each method uses its own optimal time step multiple, the IDEAL algorithm can reduce the computation time by 46.0% – 53.7% over the SIMPLER algorithm, by 56.4% – 69.3% over the SIMPLEC algorithm and by 44.9% – 53.2% over PISO algorithm for problem 4.

4.5 Problem 5: Natural convection in a cubic cavity

Natural convection in a cubic cavity is a velocity-temperature coupling problem, and is also a classical fluid flow and heat transfer problem. The flow configuration of problem 5 is shown in Figure 18. The cubic cavity has four adiabatic walls with two vertical walls being maintained at constant but different temperatures. Calculations are conducted for \( Ra = 10^4 - 10^5 \) and grid numbers = 30×30×30 – 50×50×50 with the residuals \( R_{\text{Mass}} \),
Figure 17 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 4, grid number = 190x29x29).

Table 5 Reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples (problem 4)

<table>
<thead>
<tr>
<th>Grid Numbers</th>
<th>150x20x20</th>
<th>190x29x29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLER</td>
<td>53.7%</td>
<td>46.0%</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLEC</td>
<td>57.2%</td>
<td>56.4%</td>
</tr>
<tr>
<td>Reducing ratio over PISO</td>
<td>52.9%</td>
<td>44.9%</td>
</tr>
</tbody>
</table>

The Rayleigh number is defined by

$$Ra = \frac{\rho g \beta H^3(T_H - T_C)}{\alpha \mu}.$$  \hspace{1cm} (22)

In Table 6, a comparison is given between the solutions from the IDEAL algorithm and the results from \cite{28,29}. The comparison concerns the mean Nusselt numbers $Nu$. The present results agree very well with the solutions reported by Fusegi et al. \cite{28} and Wakashima and Saitoh \cite{29}. Figure 19 shows the pressure field on the plane $z = 0.5H$ for $Ra = 10^4$, which is calculated by the IDEAL algorithm. From the figure we can find that the modified momentum interpolation method (MMIM) can avoid the checkerboard pressure distribution.

Table 6 Comparison of solutions with previous works for different $Ra$-values (Problem 5)

<table>
<thead>
<tr>
<th>$Ra$</th>
<th>$10^4$</th>
<th>$10^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusegi et al. \cite{28}</td>
<td>2.1000</td>
<td>4.3610</td>
</tr>
<tr>
<td>Wakashima et al. \cite{29}</td>
<td>2.0814</td>
<td>4.4309</td>
</tr>
<tr>
<td>IDEAL</td>
<td>2.0773</td>
<td>4.4016</td>
</tr>
</tbody>
</table>

Figures 20 and 21 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Rayleigh numbers of problem 5. From these two figures, we can find that the performances of different algorithms in velocity-temperature coupling problems are almost the same as those in fluid flow problems. The IDEAL algorithm is the most robust and most efficient one among the four algorithms compared.

Table 7 shows the reduced ratio of computation time of the IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step.
IDEAL algorithm can reduce the computation time by 16.3%—24.3% over the SIMPLER algorithm, by 52.4%—66.8% over the SIMPLEC algorithm and by 19.0%—19.7% over the PISO algorithm for problem 5.

Table 7 Reducing ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples (problem 5)

<table>
<thead>
<tr>
<th>Grid Numbers</th>
<th>$30 \times 30 \times 30$</th>
<th>$50 \times 50 \times 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Re_{10^4}$</td>
<td>$10^4$</td>
<td>$10^5$</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLER</td>
<td>24.3%</td>
<td>16.3%</td>
</tr>
<tr>
<td>Reducing ratio over SIMPLEC</td>
<td>52.4%</td>
<td>66.8%</td>
</tr>
<tr>
<td>Reducing ratio over PISO</td>
<td>19.7%</td>
<td>19.0%</td>
</tr>
</tbody>
</table>

5 Conclusions

In the present paper, the IDEAL algorithm has been extended to the 3D collocated grid system, and the performance of the IDEAL algorithm has been analyzed by a systemic comparison with three other most widely-used algorithms (SIMPLER, SIMPLEC and PISO). The main conclusions are as follows.

(1) The IDEAL algorithm is the most robust and most efficient one among the four algorithms compared.

(2) The IDEAL algorithm can converge almost at any time step multiple for the five problems studied.

(3) When each algorithm works at its own optimal time step multiple, the IDEAL algorithm can reduce the computation time by 16.3%—53.7% over the SIMPLER algorithm, by 52.4%—81.1% over the SIMPLEC algorithm and by 19.0%—53.2% over the PISO algorithm.

12 Tao W Q, Qu Z G and He Y L. A novel segregated algorithm for in-


