Chinese Science Bulletin

© 2009 SCIENCE IN CHINA PRESS
Springer

Implementation of an efficient segregated algorithm-IDEAL on 3D collocated grid system

SUN DongLiang, QU ZhiGuo, HE YaLing & Tao WenQuan†

State Key Laboratory of Multiphase Flow in Power Engineering, School of Energy & Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China

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.

3D collocated grid, segregated algorithm, IDEAL, inner iteration, convergence rate, robustness

The numerical approaches for solving the Navier-Stokes equations may be broadly divided into two categories[1,2]: density-based and pressure-based. The pressure-based approach, or the primitive-variable approach, though originally was developed for solving incompressible fluid flows, has been successfully extended to compressible flows, and widely adopted in computational fluid dynamics and numerical heat transfer. Among the pressure-based approaches, the pressure-correction method is the most widely-used one because of its simplicity and straight forwardness in physical concept. The first pressure-correction method is the SIMPLE algorithm proposed by Patankar and Spalding in 1972[3]. The major approximations made in the SIMPLE algorithm are: (1) the initial pressure field and the initial velocity field are assumed independently, hence the interconnection between pressure and velocity is neglected, leading to some inconsistency between them; and (2) the effects of the velocity corrections of the neighboring grids are arbitrarily dropped in order to simplify the solution procedure, thus making the algorithm semi-implicit. These two approximations will not affect the final solutions if the solution process converges[4]. However, they do affect the convergence rate and stability. Therefore, since the proposal of the SIMPLE algorithm, a number of variants have been proposed to overcome one or both of the approximations. In 1981, Patankar proposed the SIMPLER algorithm[5], which is the method for overcoming the first approximation in the SIMPLE algorithm. In the SIMPLER algorithm for overcoming the inconsistency between the initial pressure field and the initial velocity field, the initial pressure is determined by a pressure equation. In the CSIMPLER[6] algorithm, the same method is adopted to overcome the first approximation in the SIMPLE. In

Received August 12, 2008; accepted November 8, 2008
doi: 10.1007/s11434-009-0124-4
†Corresponding author (email: wqtao@mail.xjtu.edu.cn)
Supported by the Key Project of National Natural Science Foundation of China (Grant No. 50636050), Fundamental Projects of Research and Development in China (973) (Grant Nos. 2006CB601203 and 2007CB206902)

1984, Van Doormaal and Raithby proposed the SIMPLEC algorithm[7], 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[8,9]. 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[10] was proposed by Issa, which implements two correction steps of pressure correction. Yen and Liu[11] 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[12,13]. However, the robustness of the CLEAR algorithm is somewhat deteriorated as indicated in[14] 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)[15,16] 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[17] 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[18] 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[19] 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)[20], 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.

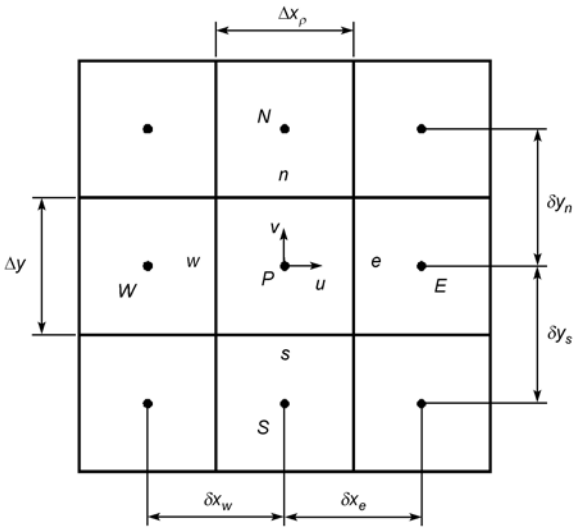

Figure 1 Control volumes on a collocated grid system.

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:

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

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

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

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:

$$\frac{a_P}{\alpha_u}u_P=\sum a_{nb}u_{nb}+b+A_P(p_w-p_e)+\frac{1-\alpha_u}{\alpha_u}a_P u_P^0. \tag{5}$$

The expressions of the coefficients a and source terms b depend on the discretized schemes, and have been well documented in literatures[17]. For the simplicity of presentation, they are not shown here.

From Eq. (5), we can get

$$u_P=\alpha_u\left(\frac{\sum a_{nb}u_{nb}+b}{a_P}\right)+\frac{\alpha_u A_P}{a_P}(p_w-p_e)+(1-\alpha_u)u_P^0. \tag{6}$$

The interface pressures are linearly interpolated from the neighboring nodes:

$$p_w=f_w^+p_P+(1-f_w^+)p_W,\quad p_e=f_e^+p_E+(1-f_e^+)p_P,$$

$$f_w^+=\frac{\Delta x_W}{2\delta_w},\quad f_e^+=\frac{\Delta x_P}{2\delta_e}.$$

We define

$$\overline{u}_P=\frac{\sum a_{nb}u_{nb}+b}{a_P} \tag{7}$$

$$d_P=\frac{\alpha_u A_P}{a_P}. \tag{8}$$

Then we get

$$u_P=\alpha_u\overline{u}_P+d_P(p_w-p_e)+(1-\alpha_u)u_P^0. \tag{9}$$

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

$$u_e=\alpha_u\overline{u}_e+d_e(p_P-p_E)+(1-\alpha_u)u_e^0, \tag{10}$$

where $\overline{u}_e$, d_e are linearly interpolated from the neighboring nodes:

$$\overline{u}_e=f_e^+\overline{u}_E+(1-f_e^+)\overline{u}_P, \tag{11}$$

$$d_e=f_e^+d_E+(1-f_e^+)d_P. \tag{12}$$

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

$$u_e=\tilde{u}_e+d_e(p_P-p_E), \tag{13}$$

where $\tilde{u}_e$ is the pseudo velocity.

$$\tilde{u}_e = \alpha_u \bar{u}_e + (1-\alpha_u)u_e^0. \tag{14}$$

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_P^0, v_P^0, w_P^0, u_e^0, v_n^0, w_t^0.

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.

----------The first inner iteration process----------

Step-3: Calculate the pseudo-velocities $\tilde{u}_e^0$, $\tilde{v}_n^0$, $\tilde{w}_t^0$ from Eq. (14). The calculation procedure of the pseudo-velocity $\tilde{u}_e^0$ is as follows. Firstly, $\bar{u}_P^0$, $\bar{u}_E^0$ are calculated from Eq. (7). Then, $\bar{u}_e^0$ 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^{Temp}.

$$\begin{aligned}
&\frac{a_P}{\alpha_p} p_P^{\mathrm{Temp}} = \sum a_{nb} p_{nb}^{\mathrm{Temp}} + b,\\
&a_P = a_E + a_W + a_N + a_S + a_T + a_B,\\
&a_E = (\rho A d)_e,\ a_W = (\rho A d)_w, a_N = (\rho A d)_n,\\
&a_S = (\rho A d)_s,\ \ a_T = (\rho A d)_t,\ a_B = (\rho A d)_b,\\
&b = (\rho \tilde{u}^0 A)_w - (\rho \tilde{u}^0 A)_e + (\rho \tilde{v}^0 A)_s - (\rho \tilde{v}^0 A)_n,\\
&\quad + (\rho \tilde{w}^0 A)_b - (\rho \tilde{w}^0 A)_t + (1-\alpha_p)\frac{a_P}{\alpha_p} p_P^{P\mathrm{Temp}}.
\end{aligned} \tag{15}$$

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

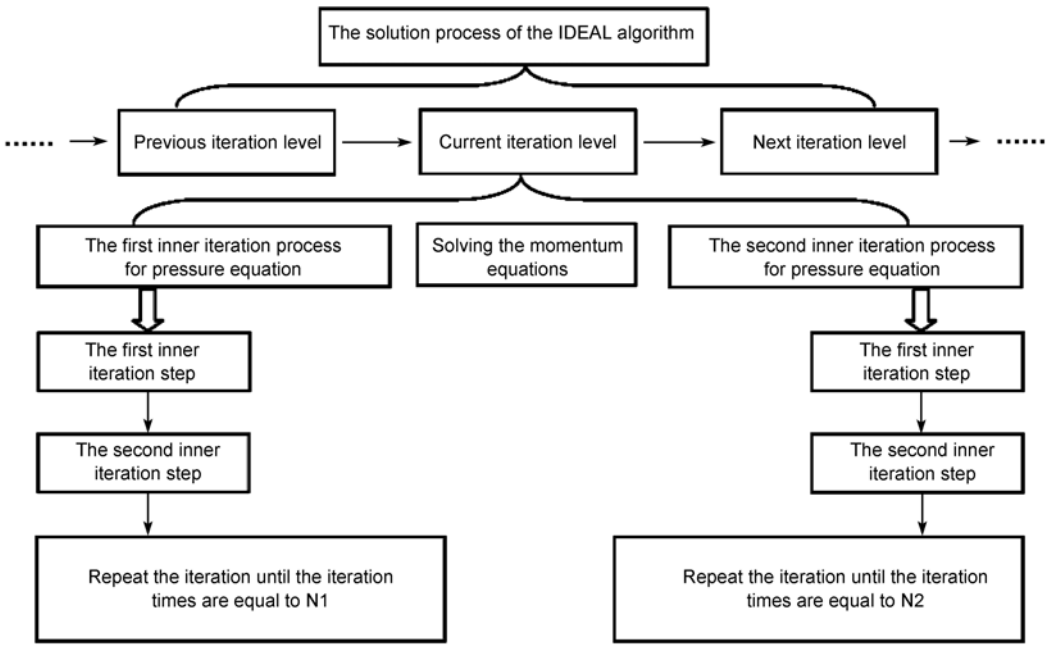

Figure 2 The framework of the solution process of the IDEAL algorithm.

In the first inner iteration process for pressure equation, the pressure under-relaxation factor α_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 α_p is set as 1.

Step-5: Calculate the temporary nodal velocities u_P^{Temp}, v_P^{Temp} and w_P^{Temp} from Eq. (9) by the temporary pressure p^{Temp}.

Step-6: Regard $u_P^{\text{Temp}}, v_P^{\text{Temp}}, w_P^{\text{Temp}}$ and p^{Temp} calculated in Step-4 and Step-5 as the temporary nodal velocity and pressure of the previous inner iteration step, denoted by $u_P^{P\text{Temp}}$, $v_P^{P\text{Temp}}$, $w_P^{P\text{Temp}}$ and $p^{P\text{Temp}}$. Return to Step-3, and then all the superscripts 0 in steps 3 and 4 are replaced by *P*Temp. Repeat such iteration process composed of steps 3, 4 and 5 until the iteration times are equal to the pre-specified times N1. After the first inner iteration process for pressure equation is finished, the final temporary pressure p^{Temp} is regarded as the initial pressure p^*.

Step-7: Solve the momentum Eq. (5) by the initial velocity and pressure p^*, and obtain the intermediate nodal velocities u_P^*, v_P^*, w_P^*.

----------The Second inner iteration process----------

Step-8: Calculate the pseudo-velocities $\tilde{u}_e^*, \tilde{v}_n^*, \tilde{w}_t^*$ from Eq. (14).

Step-9: Solve the pressure Eq. (16), and obtain the temporary pressure p^{Temp}.

$$
\begin{aligned}
&a_P p_P^{\text{Temp}} = \sum a_{nb} p_{nb}^{\text{Temp}} + b,\\
&a_P = a_E + a_W + a_N + a_S + a_T + a_B,\\
&a_E = (\rho A d)_e,\ a_W = (\rho A d)_w,\ a_N = (\rho A d)_n,\\
&a_S = (\rho A d)_s,\ \ a_T = (\rho A d)_t,\ a_B = (\rho A d)_b,\\
&b = (\rho \tilde{u}^* A)_w - (\rho \tilde{u}^* A)_e + (\rho \tilde{v}^* A)_s\\
&\qquad - (\rho \tilde{v}^* A)_n + (\rho \tilde{w}^* A)_b - (\rho \tilde{w}^* A)_t.
\end{aligned}
\tag{16}
$$

It should be noted that in the second inner iteration process for pressure equation, the pressure need not be under-relaxed.

Step-10: Calculate the temporary nodal velocities u_P^{Temp}, v_P^{Temp} and w_P^{Temp} from Eq. (9) by the temporary pressure p^{Temp}.

Step-11: Regard u_P^{Temp}, v_P^{Temp}, w_P^{Temp} and p^{Temp} calculated in Step-9 and Step-10 as the temporary nodal velocity and pressure of the previous inner iteration step, denoted by $u_P^{P\text{Temp}}$, $v_P^{P\text{Temp}}$, $w_P^{P\text{Temp}}$ and $p^{P\text{Temp}}$. Return to Step-8, and then all the superscripts * in steps 8 and 9 are replaced by *P*Temp. Repeat such iteration process composed of steps 8, 9 and 10 until the iteration times are equal to the pre-specified times N2. After the second inner iteration process for pressure equation is finished, the final temporary nodal velocities are regarded as the final nodal velocities u_P, v_P and w_P of the current iteration level. Then, the final interface velocities u_e, v_n and w_t can be calculated from Eq. (13) by the final nodal velocities and pressure.

Step-12: Solve the discretization equations of the other scalar variables if necessary.

Step-13: Regard the final nodal and interface velocities as the initial nodal and interface velocities u_P^0, v_P^0, w_P^0, u_e^0, v_n^0, w_t^0 of the next iteration level, then return to Step-2. Repeat such iterative procedure until convergence is reached.

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 N1 and the second inner iteration times N2 (hereafter N1&N2) can be adjusted. N1&N2 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 α by Eq. (17):

$$E = \frac{\alpha}{1-\alpha} \quad (0 < \alpha < 1). \tag{17}$$

Some correspondence between α 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.

Table 1 Some correspondence between α and E

α	0.1	0.5	0.9	0.95	0.96	0.97	0.98	0.99	1
E	0.111	1	9	19	24	32.3	49	99	infinite

(5) Convergence criterion

The adopted convergence criterion requires that both the relative maximum mass residual $\mathrm{Rs}_{\mathrm{Mass}}$, and the relative maximum u, v, w -component momentum residuals $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$, $\mathrm{Rs}_{W\mathrm{Mom}}$ 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—1000 and grid numbers = 32×32×32—82×82×82, and the allowed residuals $\mathrm{Rs}_{\mathrm{Mass}}$, $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ should be all less than 10^{-8}. The Reynolds number is defined by

$$Re = \frac{u_{\mathrm{lid}} H}{\nu}. \tag{18}$$

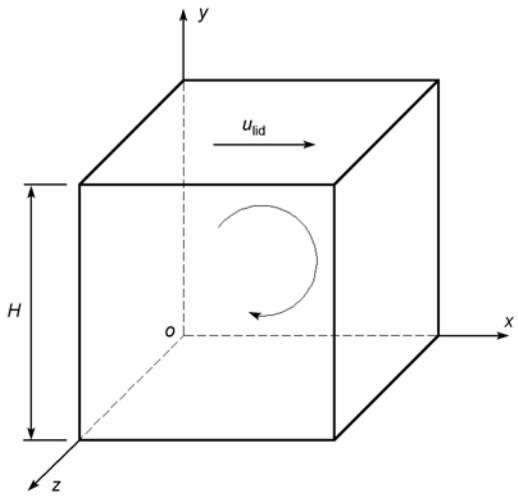

Figure 3 Flow configuration of lid-driven cavity flow in a cubic cavity (problem 1).

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.

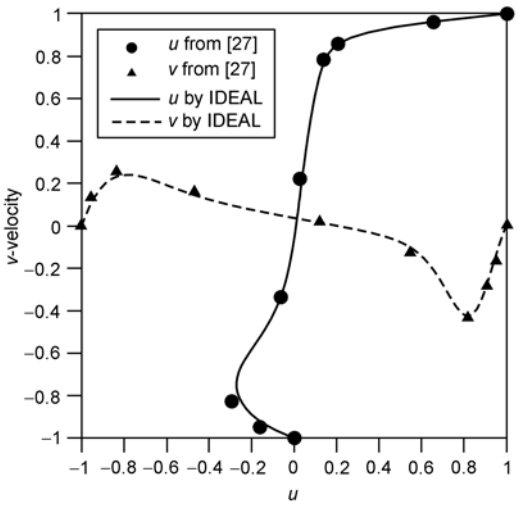

Figure 4 Comparison of velocity profiles u and v along the central axes on plane z=0.5H for Re=1000 (problem 1).

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

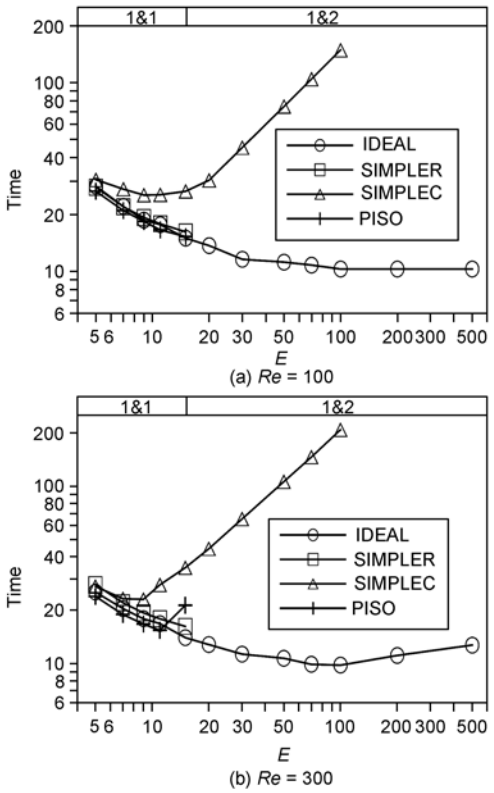

Figure 5 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 1, grid number=32×32×32).

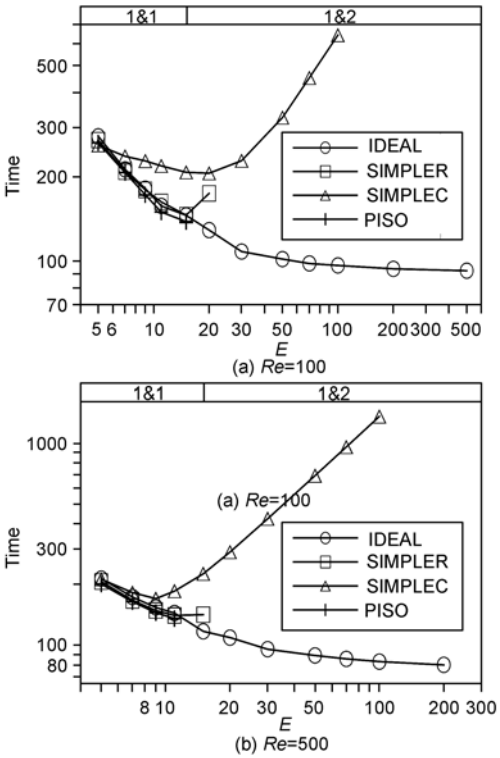

Figure 6 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 1, grid number=52×52×52).

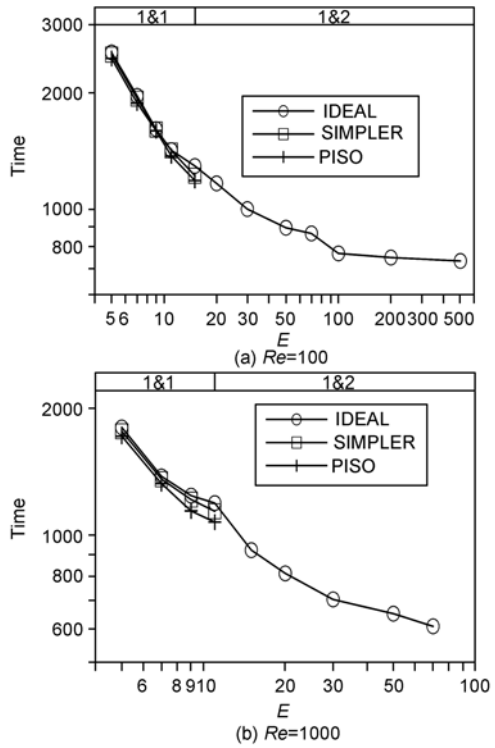

Figure 7 Comparison of computation time and robustness of IDEAL, SIMPLER and PISO algorithms (problem 1, grid number=82×82×82).

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)

Grid Numbers	32×32×32		52×52×52		82×82×82	
Re	100	300	100	500	100	1000
Reducing ratio over SIMPLER	36.4%	33.8%	36.7%	43.0%	39.8%	46.6%
Reducing ratio over SIMPLEC	59.4%	57.4%	55.1%	52.9%	–	–
Reducing ratio over PISO	32.7%	36.4%	33.2%	41.0%	38.1%	43.4%

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.

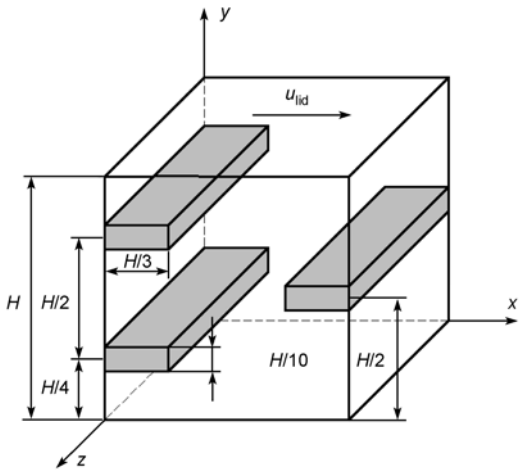

Figure 8 Flow configuration of lid-driven cavity flow in a cubic cavity with complicated structure (problem 2).

Calculations are conducted for Re=100—800 and grid numbers = 52×52×52—82×82×82. The allowed residuals $\mathrm{Rs}_{\mathrm{Mass}}$, $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ should be all less than 10^{-8}. The Reynolds number is defined by

$$Re = \frac{u_{\mathrm{lid}} H}{\nu}. \tag{19}$$

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

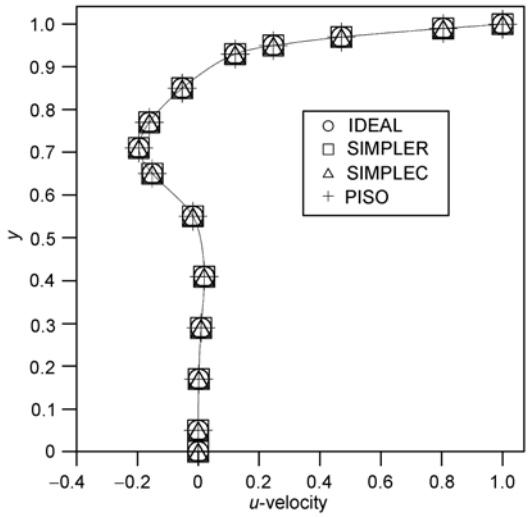

Figure 9 Comparison of velocity profiles u along the central line y on plane $z = 0.5H$ for $Re = 500$ (problem2).

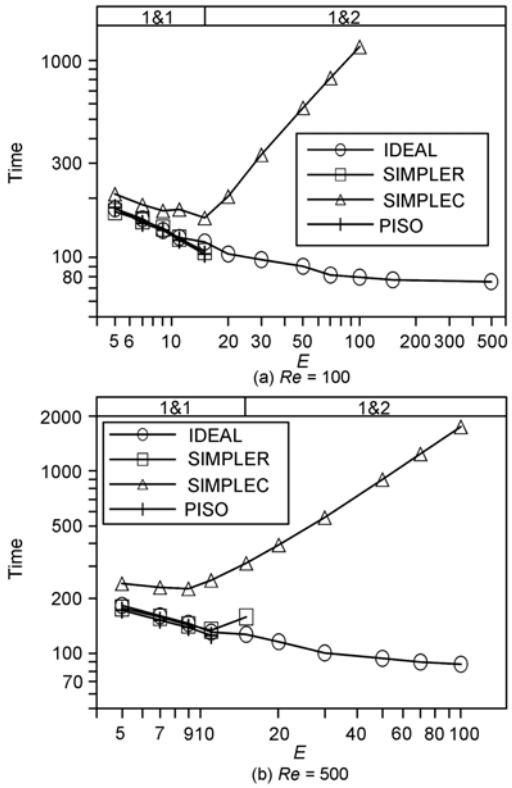

Figure 10 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 2, grid number = 52×52×52).

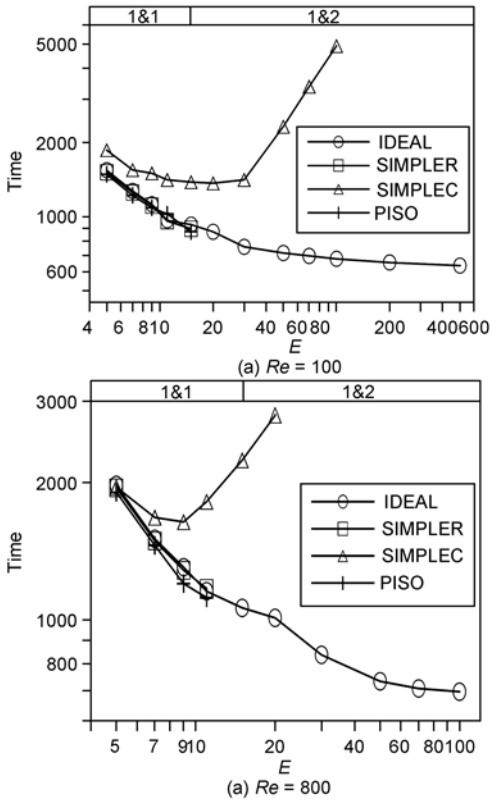

Figure 11 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 2, grid number = 82×82×82).

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

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)

Grid Numbers	52×52×52		82×82×82	
Re	100	500	100	800
Reducing ratio over SIMPLER	29.6%	34.9%	29.0%	40.6%
Reducing ratio over SIMPLEC	52.5%	61.4%	53.6%	57.6%
Reducing ratio over PISO	27.5%	30.5%	27.1%	37.8%

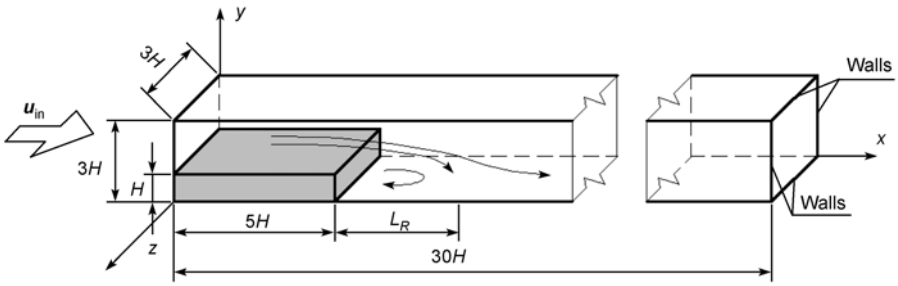

Figure 12 Flow configuration of laminar fluid flow over a backward-facing step (problem 3).

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 = 80×20×20—160×41×41. The residuals $\mathrm{Rs}_{\mathrm{Mass}}$, $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ are all set to be less than 10^{-7}. The Reynolds number is defined by

$$Re = \frac{u_{\mathrm{in}} H}{\nu} \tag{20}$$

Figures 13 and 14 show the computation time and robustness of the IDEAL, SIMPLER, SIMPLEC 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 SIMPLEC 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 SIMPLEC 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, SIMPLEC 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 SIMPLEC algorithm and by 34.0%—41.6% over PISO algorithm for problem 3.

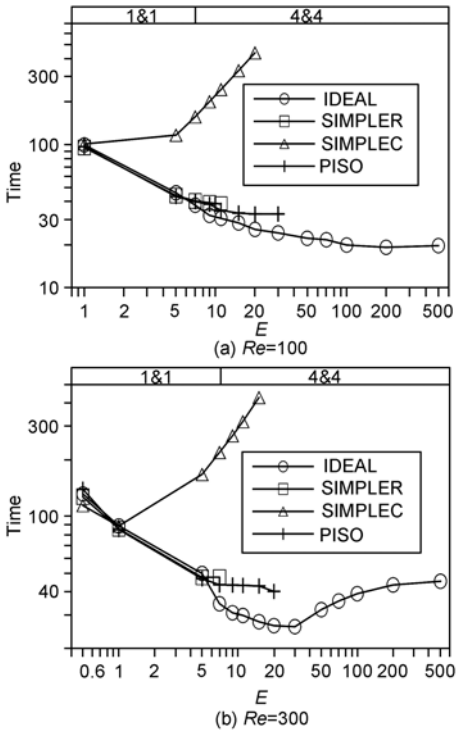

Figure 13 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 3, grid number = 80×20×20).

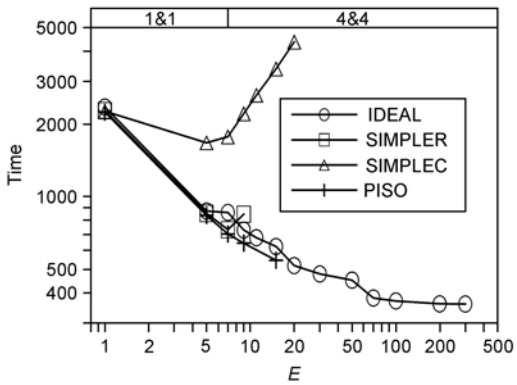

Figure 14 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 3, grid number = 160×41×41, Re = 100).

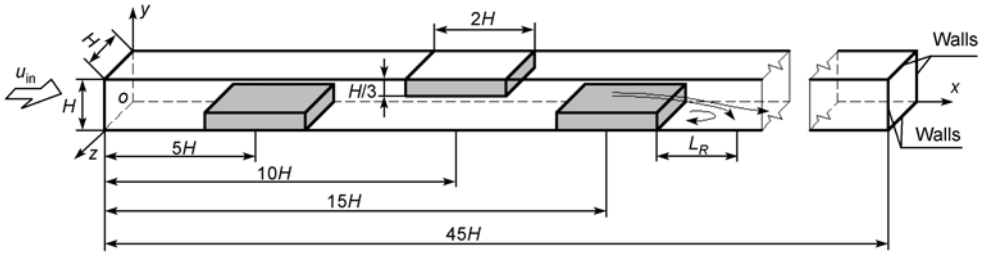

Figure 15 Flow configuration of laminar fluid flow through a duct with complicated structure (problem 4).

Table 4 Reduced ratio of computation time of IDEAL algorithm over the SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples (problem 3)

Grid Numbers	80×20×20		160×41×41
Re	100	300	100
Reducing ratio over SIMPLER	50.1%	45.2%	50.7%
Reducing ratio over SIMPLEC	81.1%	70.7%	78.5%
Reducing ratio over PISO	41.6%	34.9%	34.0%

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 $\mathrm{Rs}_{\mathrm{Mass}}$, $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ are all set to be less than 10^{-7}. The Reynolds number is defined by

$$Re = \frac{u_{\mathrm{in}} H}{\nu}. \tag{21}$$

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

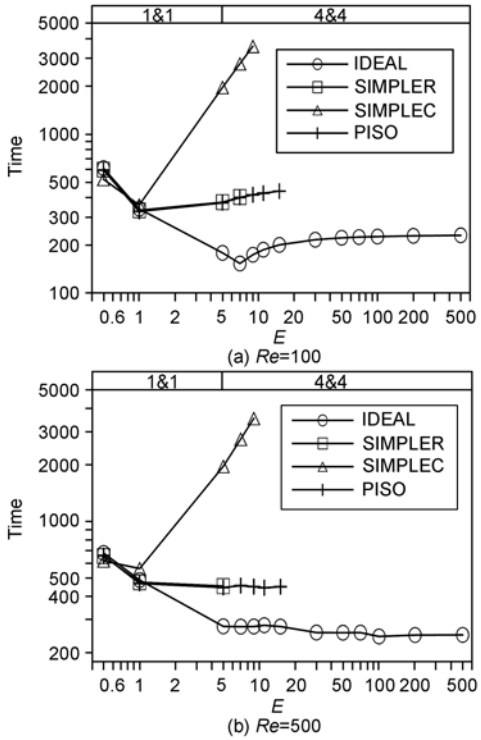

Figure 16 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 4, grid number = 150×20×20).

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 $\mathrm{Rs}_{\mathrm{Mass}}$,

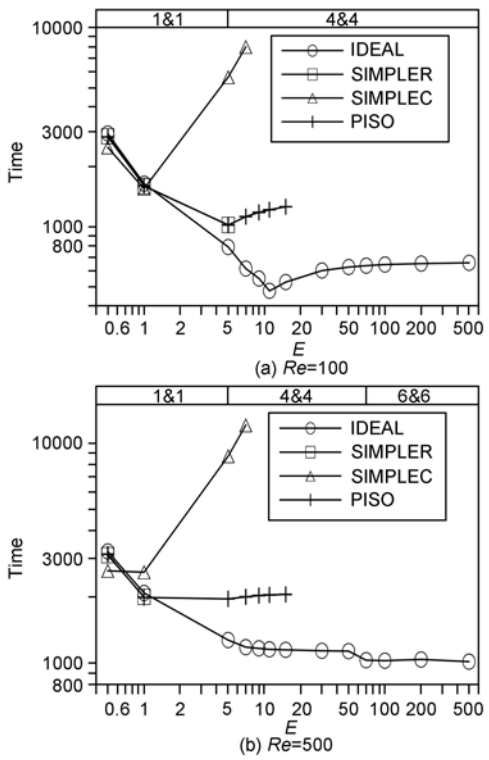

Figure 17 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 4, grid number = 190×29×29).

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)

Grid Numbers	150×20×20		190×29×29	
Re	100	300	100	500
Reducing ratio over SIMPLER	53.7%	46.0%	53.3%	49.2%
Reducing ratio over SIMPLEC	57.2%	56.4%	69.3%	60.8%
Reducing ratio over PISO	52.9%	44.9%	53.2%	48.1%

$\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ being all less than 10^{-7}. The Rayleigh number is defined by

$$Ra = \frac{\rho g \beta H^3 (T_H - T_C)}{a\mu}. \quad (22)$$

In Table 6, a comparison is given between the solutions from the IDEAL algorithm and the results from[28,29]. The comparison concerns the mean Nusselt numbers *Nu*. The present results agree very well with the solutions reported by Fusegi et al.[28] and Wakashima and Saitoh[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.

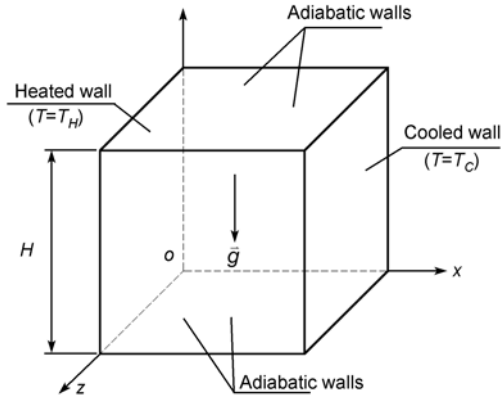

Figure 18 Flow configuration of natural convection in a cubic cavity (problem 5).

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

Ra	10^4	10^5
Fusegi et al.[28]	2.1000	4.3610
Wakashima et al.[29]	2.0814	4.4309
IDEAL	2.0773	4.4016

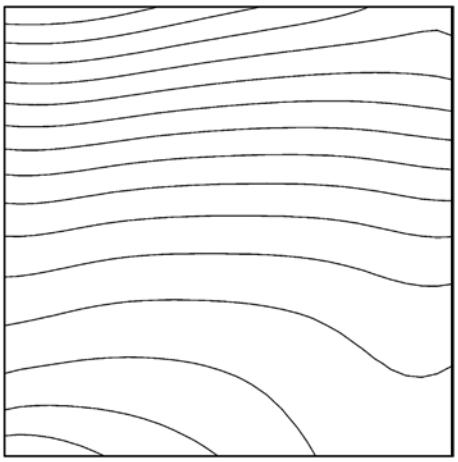

Figure 19 The pressure field on the plane z=0.5H for Ra=10^4 by IDEAL algorithm (problem 5).

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

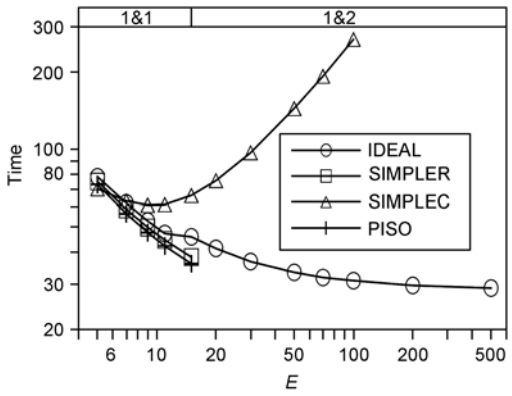

Figure 20 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 5, grid number=30×30×30, Ra=10^4).

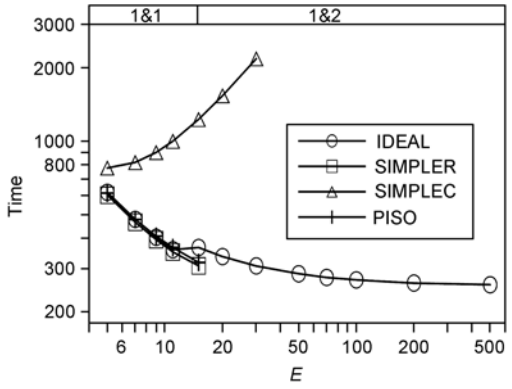

Figure 21 Comparison of computation time and robustness of the IDEAL, SIMPLER, SIMPLEC and PISO algorithms (problem 5, grid number = 50×50×50, Ra = 10^5).

multiples for different cases of problem 5. When each method uses its own optimal time step multiple, the 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)

Grid Numbers	30×30×30	50×50×50
Re	10^4	10^5
Reducing ratio over SIMPLER	24.3%	16.3%
Reducing ratio over SIMPLEC	52.4%	66.8%
Reducing ratio over PISO	19.7%	19.0%

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.

1 Shyy W, Mittal R. Solution methods for the incompressible Navier-Stokes equations. In: Johnson R W, ed. The Handbook of Fluid Dynamics. Boca Raton: CRC Press, 1998, 31.1−31.33
2 Tao W Q. Recent Advances in Computational Heat Transfer. Beijing: Science Press, 2000
3 Patankar S V, Spalding B. A calculation procedure for heat mass and momentum transfer in three dimensional parabolic flows. Int J Heat Mass Transfer, 1972, 15: 1787−1806
4 Patankar S V. Numerical heat transfer and fluid flow. Washington, DC: Hemisphere, 1980
5 Patankar S V. A calculation procedure for two-dimensional elliptic situations. Numer Heat Transfer, 1981, 4: 409−425
6 Liu X L, Tao W Q, He Y L. A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems. Engineering Computations, 2005, 22: 921−939
7 Van Doormaal J P, Raithby G D. Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numer Heat Transfer, 1984, 7: 147−163
8 Van Doormaal J P, Raithby G D. An evaluation of the segregated approach for predicting incompressible fluid flow. ASME Paper 85-HT-9, 1985
9 Raithby G D, Schneider G E. Elliptic system: finite difference method II. In: Minkowycz W J, Sparrow E M, Pletcher R H, Schneider G E, eds. Handbook of Numerical Heat Transfer. New York: Wiley, 1988, 241−289
10 Issa R I. Solution of implicitly discretized fluid flow equation by operator-splitting. J. Comput Phys, 1985, 62: 40−65
11 Yen R H, Liu C H. Enhancement of the SIMPLE algorithm by an additional explicit corrector step. Numer Heat Transfer, Part B, 1993, 24: 127−141
12 Tao W Q, Qu Z G, He Y L. A novel segregated algorithm for incom-

pressible fluid and heat transfer problems-CLEAR (Coupled and Linked Equations Algorithm Revised) Part I: mathematical formulation and solution procedure. Numer. Heat Transfer, Part B, 2004, 45: 1－17

13 Tao W Q, Qu Z G, He Y L. A novel segregated algorithm for incompressible fluid and heat transfer problems-CLEAR (Coupled and Linked Equations Algorithm Revised) Part II: application examples. Numer. Heat Transfer, Part B, 2004, 45: 19－48

14 Cheng Y P, Lee T S, Low H T, Tao W Q. An efficient and robust numerical scheme for the SIMPLER algorithm on non-orthogonal curvilinear coordinates: CLEARER. Numerical Heat Transfer, B, 2007, 51: 433－461

15 Sun D L, Qu Z G, He Y L, et al. 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, 2008, 53: 1－17

16 Sun D L, Qu Z G, He Y L, et al. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (Inner Doubly-Iterative Efficient Algorithm for Linked-Equations) Part II: application examples. Numer Heat Transfer. Part B, 2008, 53: 18－38

17 Tao W Q. Numerical Heat Transfer, 2nd. Xi'an: Xi'an Jiaotong University Press, 2001

18 Rhie C M, Chow W L. Numerical study of the turbulent flow past an airfoil with trading edge separations. AIAA J, 1983, 21(11): 1525－1535

19 Kobayashi M H, Pereira J C F. Numerical comparison of momentum interpolation methods and pressure-velocity algorithm using non-staggered grids. Commun Appl Numer Meth, 1991, 7: 173－196

20 Majumdar S. Roles of under-relaxation in momentum interpolation for calculation of flow with non-staggered grids. Numer Heat Transfer, 1988, 13: 125－132

21 Li Z Y, Tao W Q. A new stability-guaranteed second-order difference scheme. Numer. Heat Transfer, Part B, 2002, 42: 349－365

22 Khosla P K, Rubin S G. A diagonally dominant second order accurate implicit scheme. Comput Fluids, 1974, 2: 207－209

23 Haysae T, Humphery J A C, Grief A R. A consistently formulated QUICK scheme for fast and stable convergence using finite volume iterative calculation procedures. J Comput Phys, 1992, 93: 108－118

24 Gray D D, Giorgin A. The validity of the Boussinesq approximation for liquids and gases. Int J Heat Mass Transfer, 1976, 19: 545－551

25 Ku H C, Hirsh R S, Taylor T D. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. J Comput Phys, 1987, 70: 439－462

26 Guj G, Stella F. A vorticity-velocity method for the numerical solution of 3D incompressible flows. J Comput Phys, 1993, 106: 286－298

27 Tang L Q, Cheng T, Tsang T T H. Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method. Int J Numer Methods Fluids, 1995, 21: 413－432

28 Fusegi T, Hyun J M, Kuwahara K, et al. A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure. Int J Heat Mass Transfer, 1991, 34: 1543－155

29 Wakashima S, Saitoh T S. Benchmark solutions for natural convection in a cubic cavity using the high-order time-space method. Int J Heat Mass Transfer, 2004, 47: 853－864