A Fully-implicit Velocity Pressure coupling Algorithm－IDEAL and Its Applications

Dong-Liang SUN, Zhi-Guo QU, Ya-Ling HE and Wen-Quan TAO*

State Key Laboratory of Multiphase Flow In Power Engineering
School of Energy & Power Engineering
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R.China
*Corresponding author: wqtao@mail.xjtu.edu.cn

Abstract: An efficient segregated algorithm for the coupling of velocity and pressure of incompressible fluid flow, called IDEAL (Inner Doubly-Iterative Efficient Algorithm for Linked-Equations), has been proposed by the present authors. In the algorithm there exist double inner iterative processes for pressure equation at each iteration level, 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 solution process. The performance of the IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems is analyzed and a systemic comparison is made between the algorithm and three other most widely-used algorithms (SIMPLER, SIMPLEC and PISO). It is found that the IDEAL algorithm is the most robust and the most efficient one among the four algorithms compared. This new algorithm is used for the velocity prediction of a new interface capturing method –VOSET ,also proposed by the present author. It is found that the combination of VOSET and IDEAL can appreciably enhance both the interface capture accuracy and convergence rate of computations.

Keywords: segregated algorithm, IDEAL algorithm, SIMPLER algorithm, SIMPLEC algorithm, PISO algorithm, 3D cases, interface capture, VOSET

NOMENCLATURE

a	coefficient in the discretized equation
A	surface area
b	constant term in the discretized equation
d	coefficient in the velocity-correction equation
E	time step multiple
g	gravitational acceleration
N1, N2	inner doubly-iterative times
p	pressure
q_m	reference mass flow rate
Ra	Rayleigh number
Re	Reynolds number
R_{sMass}	relative maximum mass residual
$R_{sU\text{Mom}}$, $R_{sV\text{Mom}}$, $R_{sW\text{Mom}}$	relative maximum u, v, w-component momentum residuals
S	source term
T	temperature
u, v, w	velocity component in x, y, z directions
$\tilde{u}$, $\tilde{v}$, $\tilde{w}$	pseudo-velocity
x, y, z	coordinates
α	under-relaxation factor
β	expansion coefficient
η	dynamic viscosity
ν	kinematic viscosity
ρ	density

Subscripts

e, w, n, s, b, t	cell surface
in	inlet
P, E, N, S, W, B, T	grid point
m	mean
nb	neighboring grid points
u, v, w	referring to u, v, w momentum equations

Superscripts

$PTemp$	temporary value in previous inner iteration step
$Temp$	temporary value in current inner iteration step
0	initial value
*	intermediate value

1. INTRODUCTION

The numerical approaches for solving the Navier-Stokes equations may be broadly divided into two categories[1,2]: density-based and pressure-based. The density-based approach works well for cases of high Mach number, but for low Mach number flow and heat transfer problems it becomes unstable and its convergence rate is greatly deteriorated. 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 straightforward 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 in order 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

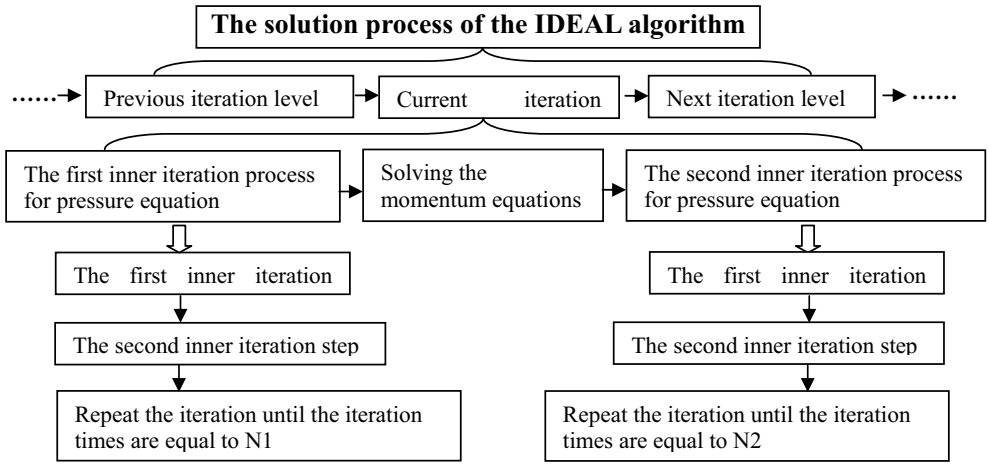


Fig. 1 The framework of the solution process of the IDEAL algorithm.

between the initial pressure field and the initial velocity field, the initial pressure is determined by a pressure equation. In the CSIMPLER[6] and CLEAR[7,8] algorithms, the same method is adopted to overcome the first approximation in the SIMPLE. In 1984, Van Doormaal and Raithby proposed the SIMPLEC algorithm[9], 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[10,11] in 1985. 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[12] was proposed by Issa in 1985, which implements two correction steps of pressure correction. This makes some improvement in the completeness of pressure correction equation of the current iteration level than that obtained by the single correction step. In the FIMOSE algorithm proposed in [13] at one iteration level the momentum and pressure equations are iteratively solved to reduce the effects of the second approximation in the SIMPLE algorithm. Yen and Liu[14] 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 SIMPLE algorithm are available in the literature, but no one has completely overcome the two assumptions in the SIMPLE algorithm except the CLEAR algorithm. In the CLEAR algorithm, the update of pressure and velocity is not conducted by adding a small value of correction, rather, the pressure field is re-solved based on the intermediate velocity, thus the effects of the neighboring grid points can be taken into account, making the algorithm fully implicit. However, the robustness of the CLEAR algorithm is somewhat deteriorated as indicated in [15] where a modified algorithm, named by CLEARER was proposed. However, by re-introduction 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 CLEAR algorithm[7,8] the IDEAL algorithm (Inner Doubly-Iterative Efficient Algorithm for Linked-Equations)[16,17] 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 SIMPLE algorithm. Thus the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate and stability of solution process.

With the development of different coupling algorithms between pressure and velocity, the comparisons between different algorithms have also been extensively conducted. These include: the comparison between FIMOSE, SIMPLER and SIMPLEC algorithms by Latimer and Pollard[13]; the comparison of the PISO, SIMPLER and SIMPLEC algorithms by Jang et al.[18]; the comparison of the PISO and SIMPLE algorithms for steady turbulent flow problems by Wanik and Schnell[19]; the comparison of the SMAC, PISO and ITA schemes for unsteady flows by Kim and Benson[20]; the comparison of SIMPLE with PISO for transient flows by Barton[21]; the comparison study of the convergence characteristics and robustness for SIMPLE, SIMPLER, SIMPLEC and SIMPLEX algorithms at fine grids by Zeng and Tao[22], etc. From the above comparisons, it can be concluded that, globally speaking, the SIMPLER, SIMPLEC and PISO algorithms are relatively better. In [8,17] comparisons were also conducted between SIMPLER, CLEAR and IDEAL algorithms for incompressible fluid flow and heat transfer problems.

Numerical simulation of complex fluid flow and heat transfer problems has become an effective tool in scientific research and engineering design and its application range has been widely extended in recent years. One important extension is from two-dimensional flow to three-dimensional case. All of the algorithm comparisons mentioned above are conducted only for two-dimensional fluid flow and heat transfer problems. There is very little information concerning the performance comparisons of different algorithms for three-dimensional fluid flow and heat transfer problems in the literature. The extension of the dimensionality in the simulation of fluid flow problems not only cause to a significant increase in computational effort, but also may drastically change the numerical characteristics of algorithms. It is the authors' experience that almost all of the above mentioned algorithms make no appreciable difference when three dimensional problems are solved. Thus it is a very challenging task and an urgent need to develop an efficient and robust algorithm for solving three-dimensional fluid flow and heat transfer problems. The major purpose of the present

paper is to adopt the IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems and make a systemic comparison between IDEAL and the algorithms of SIMPLER, SIMPLEC and PISO which are probably the three most widely-used algorithms in literature.

In the following the major solution procedure of IDEAL algorithm is first briefly reviewed. Then the comparison conditions and the convergence criterion are described, followed by a systemic comparison of the robustness and convergence rate among the four algorithms for five 3D application examples. Finally, some conclusions are drawn.

2. BRIEF REVIEW OF THE IDEAL ALGORITHM

In [16] the IDEAL algorithm has been proposed for incompressible fluid flow and heat transfer problems, and in [17] comparisons have been made for 2D cases. In the present paper the algorithm is conducted on a staggered system in three-dimensional Cartesian coordinates. For the convenience of further presentation the major points of the IDEAL algorithm are reviewed here.

Fig. 1 shows the framework of the solution process of the IDEAL algorithm in detail. The pressure-based solution method is iterative in nature. In the following we will often use the term "iteration level". By one iteration level we mean that all the computations are completed at the same values of the coefficients of the discretized momentum equations. In the IDEAL algorithm at each iteration level 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 an initial velocity field u^0, v^0 and w^0.

***Step*-2**: Calculate the coefficients *a* and source terms *b* of the discretized momentum Equations (1), (2) and (3), by the initial velocity field. The expressions of the coefficients *a* and source terms *b* depend on the discretized schemes, and have been well documented in literatures[4,23-24]. For the simplicity of presentation, they are not shown here.

--- The first inner iteration process for pressure question ---

***Step*-3**: Calculate the pseudo-velocities $\tilde{u}^0$, $\tilde{v}^0$ and $\tilde{w}^0$ defined in Equations (1), (2) and (3).

$$u_e^{Temp}=\frac{\sum a_{nb}u_{nb}^0+b}{a_e/\alpha_u}+d_e(p_P^{Temp}-p_E^{Temp})=\tilde{u}_e^0+d_e(p_P^{Temp}-p_E^{Temp}) \tag{1}$$

$$v_n^{Temp}=\frac{\sum a_{nb}v_{nb}^0+b}{a_n/\alpha_v}+d_n(p_P^{Temp}-p_N^{Temp})=\tilde{v}_n^0+d_n(p_P^{Temp}-p_N^{Temp}) \tag{2}$$

$$w_t^{Temp}=\frac{\sum a_{nb}w_{nb}^0+b}{a_t/\alpha_w}+d_t(p_P^{Temp}-p_T^{Temp})=\tilde{w}_t^0+d_t(p_P^{Temp}-p_T^{Temp}) \tag{3}$$

where, $d_e=\frac{A_e\alpha_u}{a_e}$, $d_n=\frac{A_n\alpha_v}{a_n}$, $d_t=\frac{A_t\alpha_w}{a_t}$.

***Step*-4**: Solve the pressure Equation (4), and obtain the temporary pressure p^{Temp}

$$\begin{aligned}&\frac{a_P}{\alpha_p}p_P^{Temp}=\sum a_{nb}p_{nb}^{Temp}+b\\&a_P=a_E+a_W+a_N+a_S+a_T+a_B\\&a_E=(\rho Ad)_e,\ a_W=(\rho Ad)_w,\ a_N=(\rho Ad)_n\\&a_S=(\rho Ad)_s,\ a_T=(\rho Ad)_t,\ a_B=(\rho Ad)_b\\&b=(\rho\tilde{u}^0A)_w-(\rho\tilde{u}^0A)_e+(\rho\tilde{v}^0A)_s\\&\quad-(\rho\tilde{v}^0A)_n+(\rho\tilde{w}^0A)_b-(\rho\tilde{w}^0A)_t+(1-\alpha_p)\frac{a_P}{\alpha_p}p_P^{PTemp}\end{aligned} \tag{4}$$

Equation (4) is obtained by substituting Equations (1), (2) and (3) into the discretized continuity equation:

$$(\rho u)_eA_e-(\rho u)_wA_w+(\rho v)_nA_n-(\rho v)_sA_s+(\rho w)_tA_t-(\rho v)_bA_b=0 \tag{5}$$

In the first inner iteration process for pressure equation, the pressure under-relaxation factor α_p is incorporated into the pressure Equation (4). 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 Equation (4) needn't be under-relaxed and the pressure under-relaxation factor α_p is set as 1.

***Step*-5**: Calculate the temporary velocities u^{Temp}, v^{Temp} and w^{Temp} from Equations (1), (2) and (3) by the temporary pressure p^{Temp}. Then one inner iteration step is finished and the next inner iteration step will be started.

***Step*-6**: Regard u^{Temp}, v^{Temp}, w^{Temp} and p^{Temp} calculated in *Step*-4 and *Step*-5 as the temporary velocity and pressure of the previous inner iteration step, denoted by u^{PTemp}, v^{PTemp}, w^{PTemp} and p^{PTemp}. Return to *Step*-3, and then all the superscripts 0 in steps 3 and 4 are replaced by PTemp, and the values of $\tilde{u}^0$, $\tilde{v}^0$ and $\tilde{w}^0$ are updated. Then pressure Equation (4) is re-solved. 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 Equations (6), (7) and (8), by the initial velocity and pressure p^*, and obtain the intermediate velocities u^*, v^* and w^*.

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

$$\frac{a_n}{\alpha_v}v_n^*=\sum a_{nb}v_{nb}^*+b+A_n(p_P^*-p_N^*) \tag{7}$$

$$\frac{a_t}{\alpha_w}w_t^*=\sum a_{nb}w_{nb}^*+b+A_t(p_P^*-p_T^*) \tag{8}$$

--- The second inner iteration process for pressure question---

***Step*-8**: Calculate the pseudo-velocities $\tilde{u}^*$, $\tilde{v}^*$ and $\tilde{w}^*$ defined in Equations (9), (10) and (11).

$$u_e^{Temp}=\frac{\sum a_{nb}u_{nb}^*+b}{a_e/\alpha_u}+d_e(p_P^{Temp}-p_E^{Temp})=\tilde{u}_e^*+d_e(p_P^{Temp}-p_E^{Temp}) \tag{9}$$

$$v_n^{Temp} = \frac{\sum a_{nb} v_{nb}^* + b}{a_n / \alpha_v} + d_n (p_P^{Temp} - p_N^{Temp}) = \tilde{v}_n^* + d_n (p_P^{Temp} - p_N^{Temp}) \tag{10}$$

$$w_t^{Temp} = \frac{\sum a_{nb} w_{nb}^* + b}{a_t / \alpha_w} + d_t (p_P^{Temp} - p_T^{Temp}) = \tilde{w}_t^* + d_t (p_P^{Temp} - p_T^{Temp}) \tag{11}$$

It should be noted that the coefficients a_e, a_n, a_t, a_{nb} and source terms b of Equations (6), (7), (8) and Equations (9), (10), (11) are the same as those of Equations (1), (2) and (3).

***Step* -9**: Solve the pressure Equation (12), and obtain the temporary pressure p^{Temp}.

$$\begin{aligned} & a_P p_P^{Temp} = \sum a_{nb} p_{nb}^{Temp} + b \\ & a_P = a_E + a_W + a_N + a_S + a_T + a_B \\ & a_E = (\rho A d)_e, \ a_W = (\rho A d)_w, \ a_N = (\rho A d)_n \\ & a_S = (\rho A d)_s, \ \ a_T = (\rho A d)_t, \ a_B = (\rho A d)_b \\ & b = (\rho \tilde{u}^* A)_w - (\rho \tilde{u}^* A)_e + (\rho \tilde{v}^* A)_s - (\rho \tilde{v}^* A)_n + (\rho \tilde{w}^* A)_b - (\rho \tilde{w}^* A)_t \end{aligned} \tag{12}$$

It should be noted that in the second inner iteration process for pressure equation, the pressure needn't be under-relaxed.

***Step*-10**: Calculate the temporary velocities u^{Temp}, v^{Temp} and w^{Temp} from Equations (9), (10) and (11) by the temporary pressure p^{Temp}. Then one inner iteration step is finished and the next inner iteration step will be started.

***Step*-11**: Regard u^{Temp}, v^{Temp}, w^{Temp} and p^{Temp} calculated in *Step*-9 and *Step*-10 as the temporary velocity and pressure of the previous inner iteration step, denoted by u^{PTemp}, v^{PTemp}, w^{PTemp} and p^{PTemp}. Return to *Step*-8, and then all the superscripts * in steps 8 and 9 are replaced by *PTemp*, and the values of $\tilde{u}^*$, $\tilde{v}^*$ and $\tilde{w}^*$ are updated. Then pressure Equation (12) is re-solved. Repeat the iteration 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 velocities u^{Temp}, v^{Temp} and w^{Temp} are regarded as the final velocities u, v and w of the current iteration level.

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

***Step*-13**: Regard the final velocities u, v and w as the initial velocities u^0, v^0 and w^0 of the next iteration level, then return to *Step*-2 of the next iteration level. Repeat such iterative procedure until convergence is reached.

It is interesting to note that in the IDEAL algorithm, as in the algorithm of 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 [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.

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) Hardware and codes

All the calculations in this paper are performed on the computer of CPU 2.01GHz and RAM 2.0GB along with FORTRAN 77 compiler. For the justness of comparison, the codes of SIMPLER, SIMPLEC, PISO and IDEAL algorithms are compiled under the same program structure. In order to reduce the truncated errors, double precision digital is adopted to implement computation in our codes.

(2) Discretization scheme

In order to guarantee the stability and accuracy of the numerical solution, SGSD scheme[25] 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 [26] and latter enhanced in [27].

(3) Solution method of the algebraic equations

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

(4) 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 needn't be under-relaxed at all[9,12]. 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 Equation (14)[9]:

$$E = \frac{\alpha}{1-\alpha} \qquad (0 < \alpha < 1) \tag{14}$$

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.

(5) 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.

(6) Convergence criterion

The adopted convergence criterion requires that both the relative maximum mass and the relative maximum *u, v, w* -component momentum residuals are less than some pre-specified small values.

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

The relative maximum mass residual is expressed as:

$$\mathrm{Rs}_{\mathrm{Mass}}=\frac{\mathrm{MAX}[|(\rho u^{*}A)_{w}-(\rho u^{*}A)_{e}+(\rho v^{*}A)_{s}-(\rho v^{*}A)_{n}|+(\rho w^{*}A)_{b}-(\rho w^{*}A)_{t}]}{q_{m}} \tag{15}$$

where u^{*}, v^{*} and w^{*} are the intermediate velocities of each iteration level, and q_m is the reference mass flow rate. For the open system, we take the inlet mass flow rate as the reference mass flow rate. For the closed system, we make a numerical integration for the mass flow rate along any section in the field to obtain the reference mass flow rate[23].

The relative maximum *u, v, w*-component momentum residuals are expressed as:

$$\mathrm{Rs}_{U\mathrm{Mom}}=\frac{\mathrm{MAX}\{|a_e u_e^0-[\sum_{nb}a_{nb}u_{nb}^0+b+A_e(p_P-p_E)]|\}}{\rho u_m^2} \tag{16}$$

$$\mathrm{Rs}_{V\mathrm{Mom}}=\frac{\mathrm{MAX}\{|a_n v_n^0-[\sum_{nb}a_{nb}v_{nb}^0+b+A_n(p_P-p_N)]|\}}{\rho u_m^2} \tag{17}$$

$$\mathrm{Rs}_{W\mathrm{Mom}}=\frac{\mathrm{MAX}\{|a_t w_t^0-[\sum_{nb}a_{nb}w_{nb}^0+b+A_t(p_P-p_T)]|\}}{\rho u_m^2} \tag{18}$$

where u^0, v^0 and w^0 are the initial velocities of each iteration level, and ρu_m^2 is the reference momentum. For the open system, we take the inlet momentum as the reference one . For the closed system, we make a numerical integration for the momentum along any section in the field to obtain the reference momentum[23].

(7) Double precision computations

Even though our preliminary study in the single precision have also obtained quantitatively the same results, in order to reduce the possible effects of the truncation error and the numerical noise, the double precision is adopted in the comparison computation.

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 are fluid flow problems. 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 coupled heat transfer 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[28].

4.1 Fluid flow problems

4.1.1 Problems of closed system

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

Lid-driven cavity flow in a cubic cavity has served in CFD/NHT as a benchmark problem for testing numerical procedures for three-dimensional fluid flows[29-31]. The flow configuration is shown in Fig. 2. 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

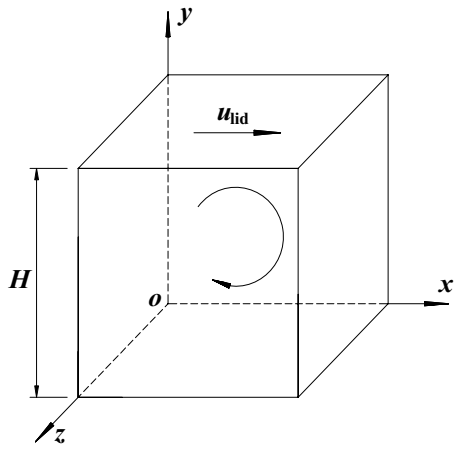


Fig. 2 Flow configuration of lid-driven cavity flow in a cubic cavity.

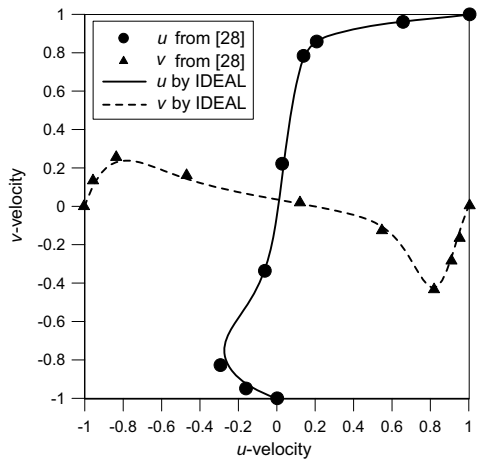
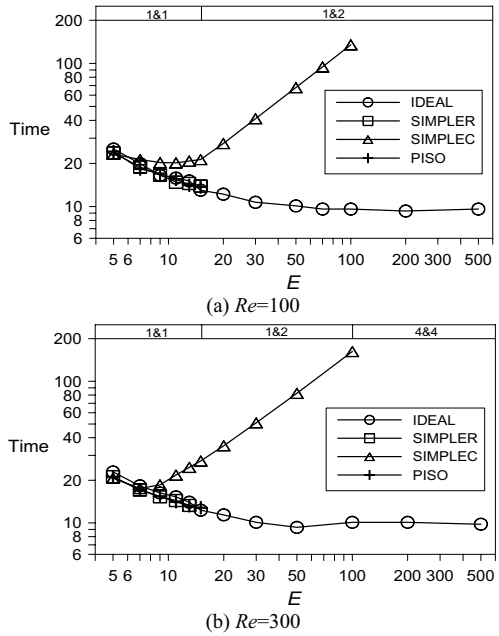


Fig. 3 Comparison of velocity profiles *u* and *v* along the central axes on plane *z*=0.5*H* for *Re*=1000.

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

In Fig. 3 the velocity profiles along the central lines on the plane *z*=0.5*H* are presented. As shown in this figure, the results calculated by IDEAL algorithm are in excellent agreement with those reported by Tang et al.[31]. This comparison gives some support to the reliability of the proposed 3D IDEAL algorithm and the developed code. From following comparisons with other well-documented algorithms (SIMPLER, SIMPLEC and PISO) further strong support to the present algorithm and code will be provided.

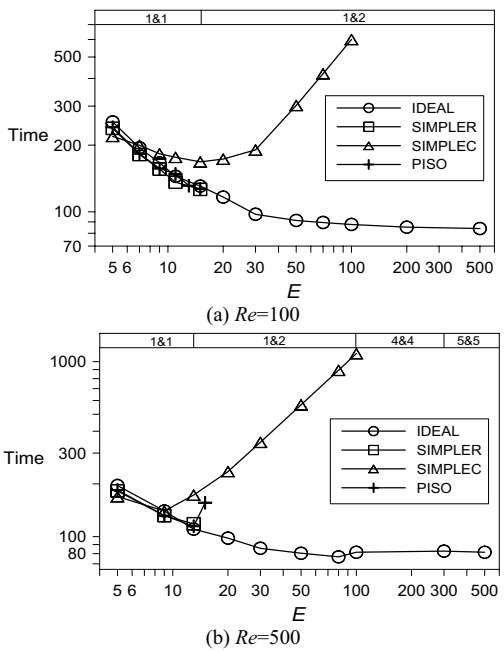
Figs. 4, 5 and 6 show the computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Reynolds numbers of



(a) *Re*=100

(b) *Re*=300

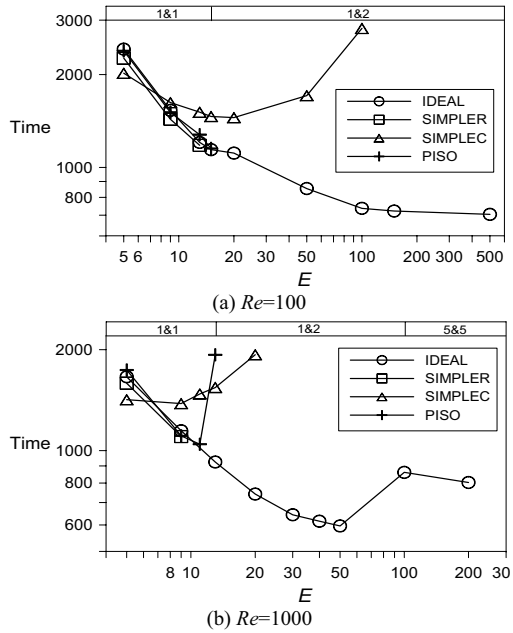
Fig. 4 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=300 with grid number=32×32×32 of problem 1.



(a) *Re*=100

(b) *Re*=500

Fig. 5 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=500 with grid number=52×52×52 of problem 1.



(a) *Re*=100

(b) *Re*=1000

Fig. 6 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=1000 with grid number=82×82×82 of problem 1.

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 Fig. 4(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, 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 SIMPLER, SIMPLEC and PISO algorithms, and it can converge almost at any time step multiple for any case in problem 1. The SIMPLER and PISO algorithms have the worst robustness and the SIMPLEC algorithm is something in between. Third, for the consumed computation time the SIMPLEC algorithm needs the largest, and the SIMPLER and PISO algorithms come next. The IDEAL algorithm needs the least.

Table 2 shows the reduced ratio of computation time of IDEAL algorithm over 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 30.1-45.9% over SIMPLER algorithm, by 45.3-56.9% over SIMPLEC algorithm and by 27.3-43.0% over PISO algorithm for problem 1.

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

Grid number	32×32×32		52×52×52		82×82×82	
Re	100	300	100	500	100	1000
Reducing ratio over SIMPLER	33.1%	30.1%	33.5%	35.3%	40.3%	45.9%
Reducing ratio over SIMPLEC	54.0%	46.9%	50.0%	45.3%	51.4%	56.9%
Reducing ratio over PISO	32.1%	27.3%	33.8%	33.2%	38.8%	43.0%

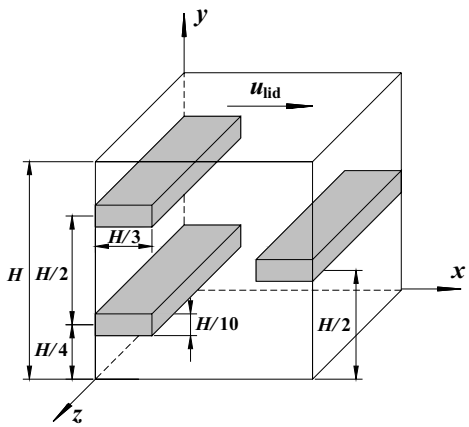


Fig. 7 Flow configuration of lid-driven cavity flow in a cubic cavity with complicated structure.

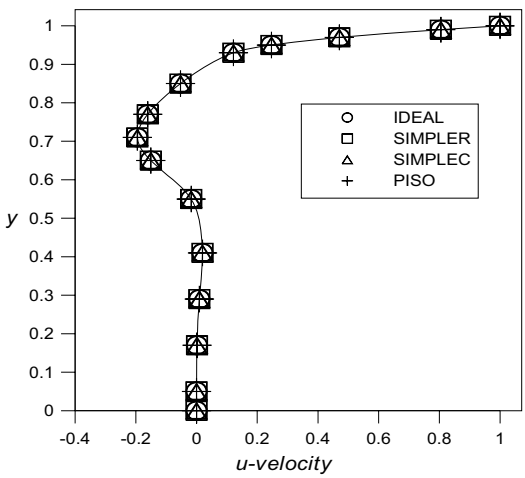


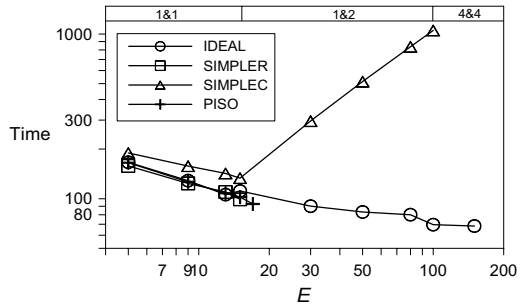
Fig. 8 Comparison of velocity profiles u along the central axes y on plane z=0.5H for Re=500.

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

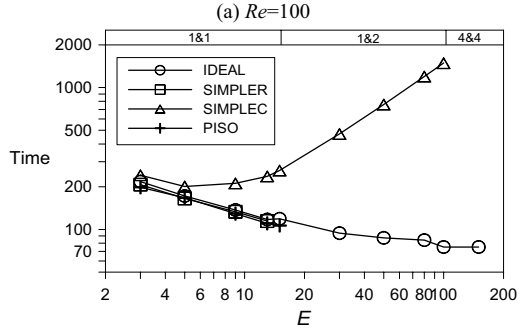
Problem 1 belongs to the simple closed system. The IDEAL algorithm shows its significant advantages over the SIMPLER, SIMPLEC and PISO algorithms for this simple closed system. In order to show the better performance of the IDEAL algorithm superior to the other three algorithms for a complicated closed system, problem 2 is especially designed. The flow configuration of problem 2 is shown in Fig. 7. Three blocks of baffle plates are inserted into the cubic cavity to make the flow configuration more complicated. The domain extension method[23] is applied for this irregular computation domain. i.e, the three blocks are supposed to be the fluids with very large viscosity and computations are conducted for the entire cubic.

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

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

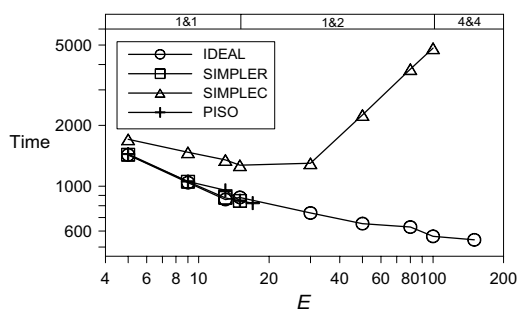


(a) Re=100

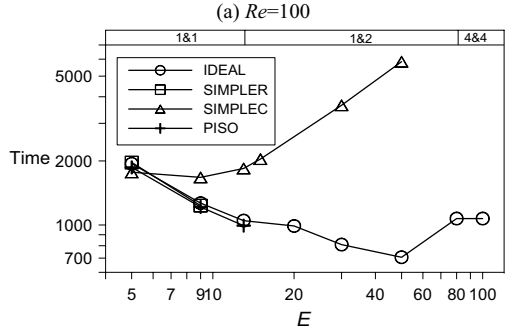


(b) Re=500

Fig. 9 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) Re=100 and (b) Re=500 with grid number=52×52×52 of problem 2.



(a) Re=100



(b) Re=800

Fig. 10 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) Re=100 and (b) Re=800 with grid number=82×82×82 of problem 2.

In Fig. 8 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 IDEAL algorithm are in excellent agreement with those calculated by the other three algorithms. Figs. 9 and 10 show the computation time and robustness of 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 IDEAL algorithm over 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 31.1-42.8% over SIMPLER algorithm, by 48.8-62.6% over SIMPLEC algorithm and by 26.5-34.9% over PISO algorithm for problem 2.

4.1.2 Problems of open system

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

Laminar fluid flow over a backward-facing step shown in Fig. 11 belongs to simple open system. It is another typical configuration widely adopted in computational fluid dynamics study. Again the domain extension method is used to deal with the solid step and solutions are performed for the entire region with 2H × 8H × 25H.

Calculations are conducted for Re=100~300 and grid numbers=127×32×63~202×52×63. The inflow velocity distribution is taken from Shah and London[32], and the fully-developed boundary condition is used at the outflow boundary. 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

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

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

Grid number	52×52×52		82×82×82	
Re	100	500	100	800
Reducing ratio over SIMPLER	31.1%	34.4%	36.5%	42.8%
Reducing ratio over SIMPLEC	48.8%	62.6%	57.7%	57.8%
Reducing ratio over PISO	26.5%	29.5%	34.9%	28.9%

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

Grid number	127×32×63		202×52×63
Re	100	300	100
Reducing ratio over SIMPLER	25.2%	31.2%	33.1%
Reducing ratio over SIMPLEC	66.5%	73.4%	54.4%
Reducing ratio over PISO	27.3%	28.3%	26.2%

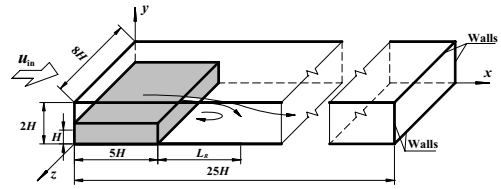


Fig. 11 Flow configuration of laminar fluid flow over a backward-facing step.

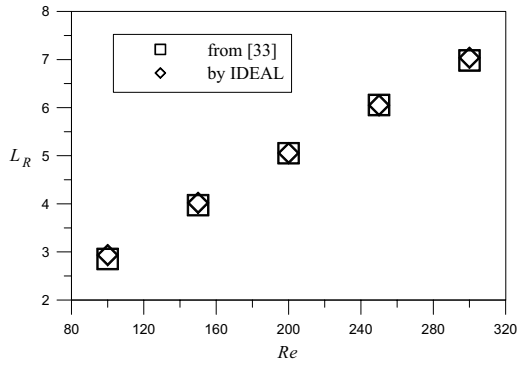


Fig. 12 The predicted reattached lengthen, L_R, on plane z=4H obtained by IDEAL algorithm and from reference[33].

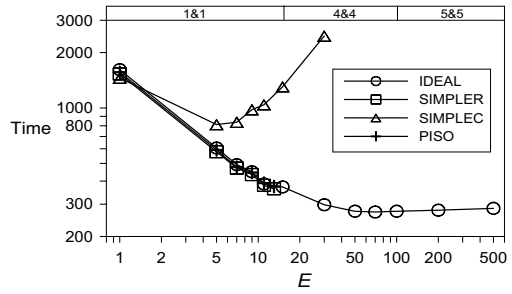
Fig. 12 shows the predicted reattachment lengths, L_R, on plane z=4H obtained, respectively, by IDEAL algorithm and from reference[33]. The results calculated by the IDEAL algorithm agree very well with those from[33]. Figs. 13 and 14 show the computation time and robustness of IDEAL, SIMPLER, SIMPLEC 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 PISO and SIMPLEC algorithms is a bit better. The IDEAL algorithm is the best. From Figs. 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 IDEAL algorithm over 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 25.2-33.1% over SIMPLER algorithm, by 54.4-73.4% over SIMPLEC algorithm and by 26.2-28.3% over PISO algorithm for problem 3.

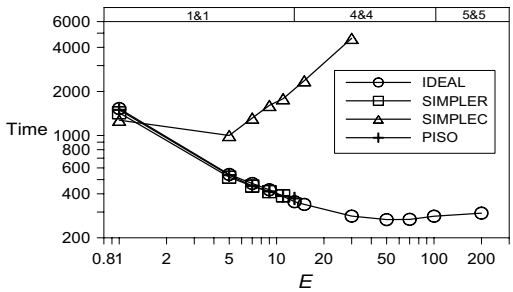
Problem 4: Laminar fluid flow through a duct with complicated structure

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 SIMPLER, SIMPLEC and PISO algorithms in a complicated open system. The flow configuration of problem 4 is shown in Fig. 15. Three blocks of baffle plates are inserted into the duct to make the flow configuration more complicated. The three solid blocks are treated by the domain extension method.

Calculations are conducted for Re=100~500, grid numbers=150×20×20~190×29×29. The inflow velocity is



(a) *Re*=100



(b) *Re*=300

Fig. 13 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=300 with grid number=127×32×63 of problem 3.

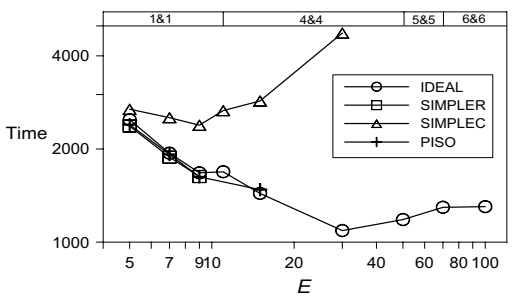


Fig. 14 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for *Re*=100 with grid number=202×52×63 of problem 3.

uniform, and the fully-developed boundary condition is used at the outflow. 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

$$Re = \frac{u_{\mathrm{in}} H}{\nu} \tag{22}$$

Table 5 shows the predicted reattachment lengths, L_R, on plane z=0.5H obtained by the four different algorithms. The results computed by the IDEAL algorithm are almost the same as those by the other three algorithms. Figs. 16 and 17 show the computation time and robustness of 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 SIMPLEC algorithm in the complicated open system becomes less robust and less effective than in the simple open system, the performances of SIMPLER, SIMPLEC are in the middle and the IDEAL algorithm is the most robust and efficient.

Table 6 shows the reduced ratio of computation time of IDEAL algorithm over 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 40.7-52.7% over SIMPLER algorithm, by 49.7-67.0% over SIMPLEC algorithm and by 41.4-53.1% over PISO algorithm for problem 4.

4.2 Velocity-temperature coupling problems

Problem 5: Natural convection in a cubic cavity

Natural convection in a cubic cavity is a velocity-temperature coupling problem, which is a classical fluid flow and heat transfer problem widely adopted in computational heat transfer community[34,35]. The flow configuration of problem 5 is shown in Fig. 18. The cubic cavity has four adiabatic walls with two vertical walls being maintained at constant but different temperatures.

Calculations are conducted for Ra=10^4~10^6 and grid numbers=30×30×30~80×80×80 with the residuals $\mathrm{Rs}_{\mathrm{Mass}}$, $\mathrm{Rs}_{U\mathrm{Mom}}$, $\mathrm{Rs}_{V\mathrm{Mom}}$ and $\mathrm{Rs}_{W\mathrm{Mom}}$ being all less than 10^{-7}. The Rayleigh number is defined by

$$Ra = \frac{\rho g \beta H^3 (T_H - T_C)}{a\eta} \tag{23}$$

In Table 7, a comparison is given between the solutions from the IDEAL algorithm and the results from[34, 35]. The comparison concerns the mean Nusselt, Nu_{m}, which is defined as

$$Nu_{\mathrm{m}} = \frac{\iint (Nu_{\mathrm{Local}}(y,z)|_{x=0} + Nu_{\mathrm{Local}}(y,z)|_{x=H}) dydz}{2H^2} \tag{24}$$

where,

$$Nu_{\mathrm{Local}}(y,z)|_{x=0 \text{ or } x=\mathrm{H}} = \frac{-(\partial T/\partial x)H}{T_H - T_C} \tag{25}$$

Table 5 Predicted reattachment lengths on plane z=0.5H in problem 4.

Re	IDEAL	SIMPLER	SIMPLEC	PISO
100	0.9725	0.9728	0.9730	0.9725
300	2.1099	2.1095	2.1100	2.1095
500	3.2426	3.2495	3.2525	3.2423

Table 6 Reduced ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples in problem 4.

Grid number	150×20×20		190×29×29	
Re	100	300	100	500
Reducing ratio over SIMPLER	49.4%	52.7%	40.7%	50.5%
Reducing ratio over SIMPLEC	49.7%	58.6%	67.0%	58.9%
Reducing ratio over PISO	48.8%	53.1%	41.4%	47.3%

Table 7 Comparison of solutions with previous works for different *Ra*-values of Problem 5.

Ra	10^4	10^5	10^6
Fusegi et al.[34]	2.1000	4.3610	8.770
Wakashima et al.[35]	2.0814	4.4309	8.8681
IDEAL	2.0842	4.4048	8.8005

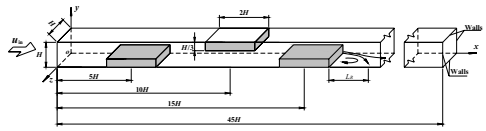
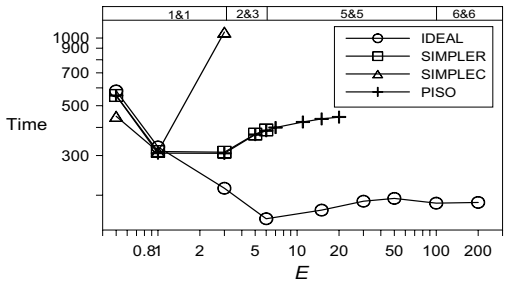
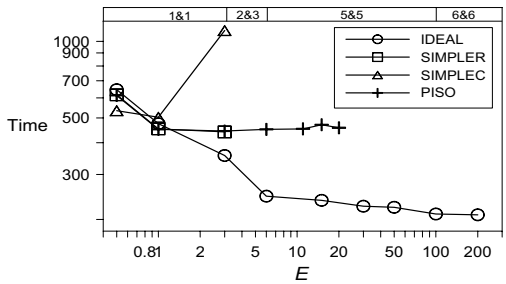


Fig. 15 Flow configuration of laminar fluid flow through a duct with complicated structure.

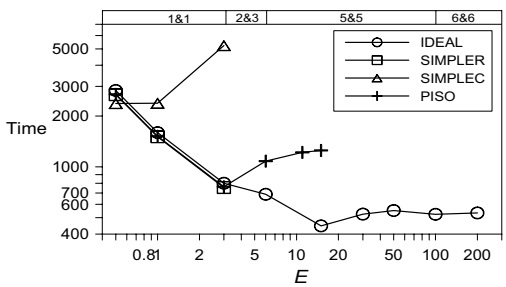


(a) *Re*=100

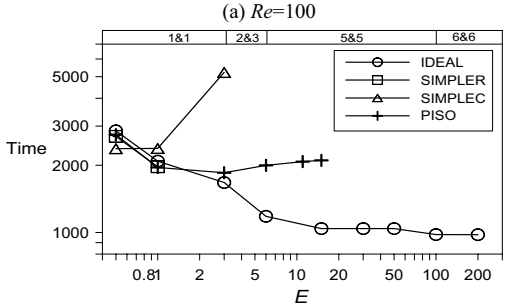


(b) *Re*=300

Fig. 16 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=300 with grid number =150×20×20 of problem 4.



(a) *Re*=100



(b) *Re*=500

Fig.17 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for (a) *Re*=100 and (b) *Re*=500 with grid number =190×29×29 of problem 4.

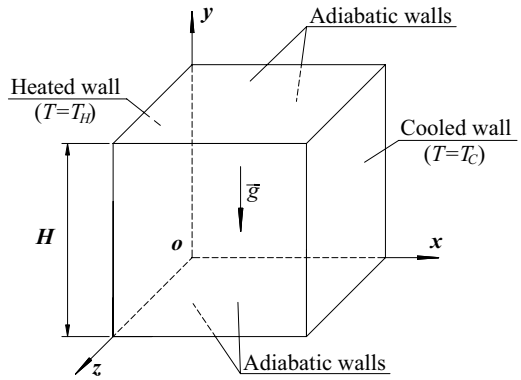
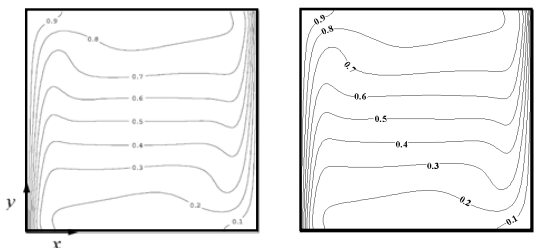


Fig.18 Flow configuration of natural convection in a cubic cavity.



(a) (b)

Fig. 19 Temperatures at the plane z=0.5H for Ra=10^6, obtained (a) from reference[35], (b) by IDEAL algorithm.

In Fig. 19, the temperatures at the plane z=0.5H for Ra=10^6 are shown. From the figure, we can find that the temperature field calculated by IDEAL algorithm agrees very well with that from the reference[35].

Figs. 20, 21 and 22 show the computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for different grid numbers and different Rayleigh numbers of problem 5. From these figures, we can find that the performances of different algorithms in velocity-temperature coupling problems are almost the same as those in decoupled fluid flow problems, for example, problem 1 and problem 2. The IDEAL algorithm is the most robust and most efficient one among the four algorithms compared.

Table 8 shows the reduced ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step 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 12.9-22.4% over SIMPLER algorithm, by 48.3-67.9% over SIMPLEC algorithm and by 10.7-13.7% over PISO algorithm for problem 5.

Table 8 Reduced ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms at their own optimal time step multiples in problem 5.

Grid number	30×30×30	50×50×50	80×80×80
Ra	10^4	10^5	10^6
Reducing ratio over SIMPLER	22.4%	12.9%	13.7%
Reducing ratio over SIMPLEC	48.3%	63.7%	67.9%
Reducing ratio over PISO	13.7%	10.7%	11.9%

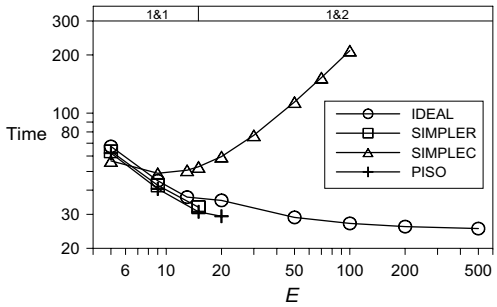


Fig. 20 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for Ra=10^4 with grid number=30×30×30 of problem 5.

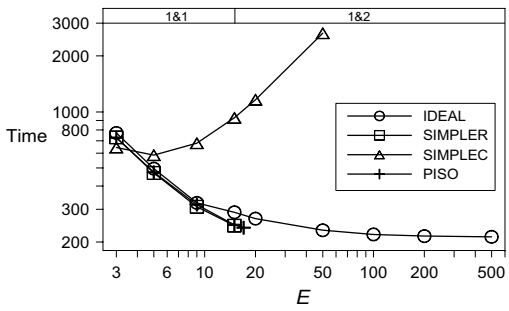


Fig. 21 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for Ra=10^5 with grid number=50×50×50 of problem 5.

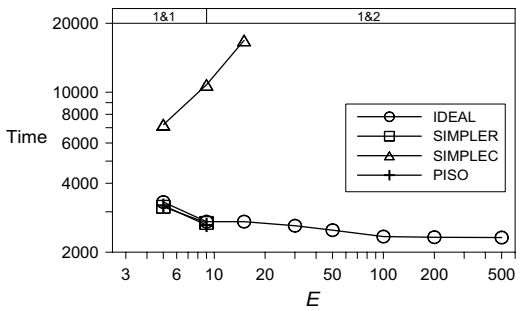


Fig. 22 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms for Ra=10^6 with grid number=80×80×80 of problem 5.

4.3 Interface capture by VOSET in conjunction with IDEAL algorithm

Recently we have proposed an improved interface capture method, called VOSET[35], which is an appropriate combination of the well-know VOF[36] and Level-Set[37] methods. VOSET and accurately predict the curvature of an interface (advantage of Level-Set method) while still keep mass conservation characteristics (advantage of VOF). In addition the computational efficiency of VOSET is quite high. To predict the interface of two-phase flow the velocity fields should be first predicted and usually the fractional step method is used. However, for complicated geometry this method is not convenient. The SIMPLE–series algorithms provide another choice. The drawback is slow convergence rate. We have incorporated the IDEAL algorithm with Bi-CGSTAB[38] for the solution of the algebraic equation for solving the velocity field and adopt VOSET for the interface capture. Very successful results are obtained in both the acceleration of the convergence rate and the extension of parameter range for the interface capturing. For the upward moving of a single bubble in a static liquid volume, some comparisions are provided in Table 9, where the dimensionless parameters Eo an M are number of Morton and Eotvos, respectively, and defined by:

$$Eo = gd_e^2(\rho_l - \rho_g)/\sigma; \qquad M = g\eta_l^4/\rho_l\sigma^3 \tag{26}$$

From Table 9 it can be seen that the new prediction method combined by IDEAL/VOSET and Bi-CGSATB is very efficient.

4. CONCLUSIONS

In the present paper the performance of the IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems 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 12.9-52.7% over SIMPLER algorithm, by 45.3-73.4% over SIMPLEC algorithm and by 10.7-53.1% over PISO algorithm.
(4) The combination of IDEAL-VOSET-Bi-CGSTAB provides an very efficient interface capturing method.

The extensions of the 3D IDEAL algorithm to non-orthogonal curvilinear systems and to unstructured grid systems are now underway in the authors' group and will be reported elsewhere.

ACKNOWLEDGMENTS

This work was supported by the Key Project of the National Natural Science Foundation of China (50636050) and the Fundamental Projects of R & D in China (973) (G2006CB601203).

Table 9 CPU time comparisons.

(a) Case 1: Eo=1.0, M=0.001, grid system 150×50

N1 & N2 of IDEAL	1&1	2&2	3&3	4&4
Ratio of CPU time reduction compared with SIMPLER+ADI	78.9%	84.7%	83.5%	80.7%

(b) Case 2: Eo=10.0, M=0.1, grid system 150×50

N1 & N2 of IDEAL	1&1	2&2	3&3	4&4
Ratio of CPU time reduction compared with SIMPLER+ADI	84.7%	88.5 %	88.1%	85.9%

(c) Case 3: Eo=100.0, M=1000.0, grid system 150×50

N1 & N2 of IDEAL	1&1	2&2	3&3	4&4
Ratio of CPU time reduction compared with SIMPLER+ADI	82.2%	86.5%	84.8%	82.2%

REFERENCES

[1] 1998, Shyy, W. and Mittal, R., *Solution Methods for the Incompressible Navier-Stokes Equations, in R. W. Johnson (ed.), Handbook of Fluid Dynamic,* CRC Press, Boca Raton.

[2] 2000, Tao, W.Q., *Recent Advances in Computational Heat Transfer*. Science Press, Beijing.

[3] 1972, Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat Mass and Momentum Transfer in Three Dimensional Parabolic Flows," *International Journal of Heat and Mass Transfer*, Vol.15, pp.1787-1806.

[4] 1980, Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC.

[5] 1981, Patankar, S.V., "A Calculation Procedure for Two-Dimensional Elliptic Situations," *Numerical Heat Transfer*, Vol.4, pp.409-425.

[6] 2005, Liu, X.L., Tao, W.Q. and He, Y.L., "A Simple Method for Improving the SIMPLER Algorithm for Numerical Simulations of Incompressible Fluid Flow and Heat Transfer Problems," *Engineering Computations*, Vol.22, pp.921-939.

[7] 2004, Tao, W.Q., Qu, Z.G. and He, Y.L., "A Novel Segregated Algorithm for Incompressible Fluid and Heat Transfer Problems-CLEAR (Coupled and Linked Equations Algorithm Revised) Part I: Mathematical Formulation and Solution Procedure," *Numerical Heat Transfer, Part B*, Vol.45, pp.1-17.

[8] 2004, Tao, W.Q., Qu, Z.G. and 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," *Numerical Heat Transfer, Part B*, Vol.45, pp.19-48.

[9] 1984, Van Doormaal, J.P. and Raithby, G.D., "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, Vol.7, pp.147-163.

[10] 1985, Van Doormaal, J.P. and Raithby, G.D., "An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flow," *ASME Paper 85-HT-9*.

[11] 1988, Raithby, G.D. and Schneider, G.E., *Elliptic System: Finite Difference Method II, in W. J. Minkowycz, E. M. Sparrow, R. H. Pletcher, and G. E. Schneider (eds.), Handbook of Numerical Heat Transfer*, Wiley, New York.

[12] 1985, Issa, R.I., "Solution of Implicitly Discretized Fluid Flow Equation by Operator-Splitting," *Journal of Computational Physics*, Vol.62, pp.40-65.

[13] 1985, Latimer, B.R. and Pollard, A., "Comparison of Pressure-Velocity Coupling Solution Algorithms," *Numerical Heat Transfer*, Vol.8, pp.635-652.

[14] 1993, Yen, R.H. and Liu, C.H., "Enhancement of the SIMPLE Algorithm by an Additional Explicit Corrector Step," *Numerical Heat Transfer, Part B*, Vol.24, pp.127-141.

[15] 2007, Cheng, Y.P., Lee, T.S., Low, H.T. and Tao, W.Q., "An efficient and robust numerical scheme for the SIMPLER algorithm on non-orthogonal curvilinear coordinates: CLEARER." *Numerical Heat Transfer, Part B*, Vol.51, pp.433-461.

[16] 2008, Sun, D.L., Tao, W.Q. and Qu, Z.G., "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," *Numerical Heat Transfer, Part B*, Vol.53, pp.1-17.

[17] 2008, Sun, D.L., Tao, W.Q. and Qu, Z.G., "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," *Numerical Heat Transfer, Part B* , Vol.53, pp.18-38.

[18] 1986, Jang, D.S., Jetli, R. and Acharya, S., "Comparison of the PISO, SIMPLER, and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems," *Numerical Heat Transfer*, Vol.10, pp.209-228.

[19] 1989, Wanik, A. and Schnell, U., "Some Remarks on the PISO and SIMPLE Algorithms for Steady Turbulent Flow Problems," *Computers &. Fluids*, Vol.17, pp.535-570.

[20] 1992, Kim, S.W. and Benson, T.J., "Comparison of the SMAC, PISO and Iterative Time-Advancing Schemes for Unsteady Flows," *Computers &. Fluids*, Vol.21, pp.435-454.

[21] 1998, Barton, I.E., "Comparison of SIMPLE- and PISO-Type Algorithms for Transient Flows," *International Journal for Numerical Methods in Fluids*, Vol.26, pp.459-483.

[22] 2003, Zeng, M. and Tao, W.Q., "A comparison study of the convergence characteristics and robustness for four variants of SIMPLE-family at fine grids," *Engineering Computations*, Vol.20, pp.320-341.

[23] 2001, Tao, W.Q., *Numerical Heat Transfer. 2nd*, Xi'an Jiaotong University Press, Xi'an, China.

[24] 1995, Versteeg, H.K. and Malalasekera, W., *An introduction to computational fluid dynamics. The finite volume method*, Essex, Longman Scientific & Technical.

[25] 2002, Li, Z.Y. and Tao, W.Q., "A new stability-guaranteed second-order difference scheme," *Numerical Heat Transfer, Part B*, Vol.42, pp.349-365.

[26] 1974, Khosla, P.K. and Rubin, S.G., "A diagonally dominant second order accurate implicit scheme," *Computers &. Fluids*, Vol.2, pp.207-209.

[27] 1992, Haysae, T., Humphery J.A.C. and Grief, A.R., "A consistently formulated QUICK scheme for fast and stable convergence using finite volume iterative calculation procedures," *Journal of Computational Physics*, Vol.93, pp.108-118.

[28] 1976, Gray, D.D. and Giorgini, A., "The validity of the Boussinesq approximation for liquids and gases," *International Journal of Heat and Mass Transfer*, Vol.19, pp.545-551.

[29] 1987, Ku, H.C., Hirsh, R.S. and Taylor, T.D., "A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations," *Journal of Computational Physics*, Vol.70, pp.439-462.

[30] 1993, Guj, G. and Stella, F., "A vorticity-velocity method for the numerical solution of 3D incompressible flows," *Journal of Computational Physics*, Vol.106, pp.286-298.

[31] 1995, Tang, L.Q., Cheng, T. and Tsang, T.T.H., "Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method," *International Journal for Numerical Methods in Fluids*, Vol.21, pp.413-432.

[32] 1978, Shah, R.K. and London, A.L., *Laminar forced convection in D*, Academic Press, New York.

[33] 2003, Nie, J.H. and Armaly, B.F., "Reattachment of three-dimensional flow adjacent to backward-facing step," *ASME Journal of Heat Transfer*, Vol.125, pp.422-428.

[34] 1991, Fusegi, T., Hyun, J.M., Kuwahara, K. and Farouk, B., “A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure,” *International Journal of Heat and Mass Transfer*, Vol.34, pp.1543–1557.
[35] 2004, Wakashima, S. and Saitoh, T.S., “Benchmark solutions for natural convection in a cubic cavity using the high-order time–space method,” *International Journal of Heat and Mass Transfer*, Vol.47, pp.853–864.
[36] Sun, D.L., Li, Z.Y., He, Y.L. and Tao, W.Q., “A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows,” Submitted for review.
[37] 1981, Hirt, C.W. and Nichols, B.D., “Volume of fluid (VOF) method for the dynamics of free boundary,” *J. Comput. Phys*., Vol.39, pp.201-225.
[38] 1988, Osher, S. and Sethian, J.A., “Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *J. Comput. Phys*, Vol.79, pp.12-49.
[39] 1992, van der Vorst, H.A., “Bi-CGSTAB, a more smoothly converging variant of Bi-CG for the solution of non-symmetric systems pf linear equations,” *SIAM Journal of Scientific and Statistical Computations*, Vol.13, pp.631-644.