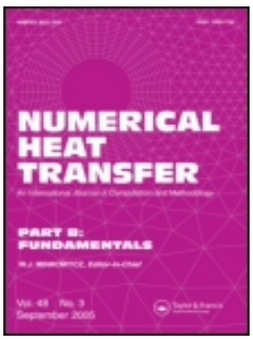


Numerial Heat Transfer

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: http://www.tandfonline.com/loi/unhb20

IMPLEMENTATION OF CLEAR ALGORITHM ON COLLOCATED GRID SYSTEM AND APPLICATION EXAMPLES

Z. G. Qu , W. Q. Tao & Y. L. He

To cite this article: Z. G. Qu , W. Q. Tao & Y. L. He (2004) IMPLEMENTATION OF CLEAR ALGORITHM ON COLLOCATED GRID SYSTEM AND APPLICATION EXAMPLES, Numerial Heat Transfer, 47:1, 65-96, DOI: 10.1080/104077990502998

To link to this article: https://doi.org/10.1080/104077990502998

Published online: 24 Feb 2007.

Submit your article to this journal

Article views: 112

View related articles

Citing articles: 11 View citing articles

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=unhb20

Numerical Heat Transfer, Part B, 46: 65–96, 2005
Copyright © Taylor & Francis Inc.
ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/104077990502998

Taylor & Francis
Taylor & Francis Group

IMPLEMENTATION OF CLEAR ALGORITHM ON COLLOCATED GRID SYSTEM AND APPLICATION EXAMPLES

Z. G. Qu, W. Q. Tao, and Y. L. He
State Key Laboratory of Multiphase Flow in Power Engineering, School of Energy & Power Engineering, Xi'an, Jiaotong University, Xi'an, Shaanxi, People's Republic of China

Implementation of the CLEAR algorithm on a collocated grid system is conducted. Detailed discussion of the previous momentum interoperation method (MIM) is given to analyze the condition to get a unique converged solution that is independent of relaxation factor for steady flow. Six numerical examples on nonstaggered grids of forced-convective fluid flow and natural convection are provided to compare the convergence performance between CLEAR and SIMPLER. The domain extension method, widely used on staggered grids to deal with mildly irregular domains, is further refined to meet the requirement of collocated grids. It is shown that on a collocated grid the CLEAR algorithm can also greatly enhance the convergence rate, based on iteration number and CPU time consumed, compared with the SIMPLER algorithm with similar robustness.

INTRODUCTION

The pressure-correction method that belongs to the segregated algorithm is an approach widely employed for numerically solving the Navier-Stokes equations for fluid flow and heat transfer problems. The SIMPLE algorithm, proposed by Patankar and Spalding [1] in 1972, was the first such algorithm widely used in literature. Since then several variants have been reported to enhance its convergence rate, among which are SIMPLER [2], SIMPLEC [3], SIMPLEX [4, 5], and PISO [6]. In addition, Yen and Liu [7] presented the explicit correction step method to accelerate the convergence rate. Sheng et al. [8] added a term of temperature correction into the velocity correction to enhance convergence rate for buoyancy-driven fluid flows. Recently, Yu et al. [9] modified the SIMPLER algorithm to artificially change the underrelaxation term to match the variable to be solved. The revised method was called MSIMPLER. All the above-mentioned methods are usually called SIMPLE-like or SIMPLE-family algorithms. Moukalled and Darwish [10] did

Received 13 February 2004; accepted 20 May 2004.
The work reported here was supported by the National Key Project of R & D of China (G2000026303), and the National Natural Science Foundation of China (50276046, 50476046) and RFDP200300698015.
Address correspondence to Wen-Quan Tao, School of Energy & Power Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P. R. China. E-mail: wqtao@mail.xjtu.edu.cn

NOMENCLATURE

$a_P, a_E, a_W,$ a_N, a_S	coefficients in the discretized equation
A	surface area
b	source term
B	source term
d_{ef}, d_{nf}	diffusion conductivity of the interface velocity
d_P d_E	diffusion conductivity of the main node velocity
D	diameter
E	time step multiple
flow_{ch}	characteristic (reference) flow rate
g	gravitational acceleration
H_1, H_2	height defined in Figure 18
L	length of square cavity
L_1, L_R, L_2	dimensions defined in Figure 18
L_{in}, L_x	length defied in Figure 15
p	pressure
p^*	temporary pressure
p'	pressure correction
r	radius
R	radius of tube wall
Ra	Rayleigh number
Re	Reynolds number
Rs_{cv}	relative mass flow rate unbalance of control volume
S_ϕ	source term
T	temperature
u, v	velocity component in x and y directions
$\hat{u}, \hat{u}^*$	pseudo-velocity
$\hat{v}, \hat{v}^*$	pseudo-velocity
U, V	dimensionless velocity in two coordinates
U_{lid}	moving velocity of lid
x, y	coordinates
X, Y	dimensionless coordinates
α	underelaxation factor
β	relaxation factor
Γ	nominal diffusion coefficient
δ	gap width
$\delta x, \delta y$	distance between two adjacent grid points in x and y directions
$\Delta x, \Delta y$	control volume width in x and y directions
ε	convergent criteria
η	fluid dynamic viscosity
λ	thermal conductivity coefficient
ρ	fluid density
ϕ	general variable

Superscripts

u, v	refers to u, v momentum equation
0	resolution of the previous iteration

Subscripts

e, w, n, s	cell surface
f	interface
in	inlet; inner
max	maximum
mean	averaged
nb	neighboring grid points
out	outlet
$P, E, N,$ S, W	grid points

a comprehensive review and give a unified reorganization expression form for all the pressure-correction algorithms. The common characteristic of these algorithms is that a pressure-correction term is introduced to the calculation procedure of each segregated solution step to improve the velocity, which is solved from a momentum-discretized equation as the intermediate value. The pressure-correction term in the SIMPLE-family algorithms is used to improve the intermediate velocity solved from the momentum equation such that the modified velocity can satisfy the mass conservation condition for each control volume at each iteration level, which is critical for iteration convergence, as indicated in [11]. In the derivation of the pressure-correction equation, the effects of the pressure corrections of the neighboring grid points are neglected, which will not affect the final solution when the iterative process converges [12], but does affect the convergence rate as described in [13]. Because of this neglect, all the above-mentioned algorithms are of semi-implicit type. Very recently, the present authors [14, 15] proposed a fully implicit segregated algorithm,

CLEAR (Coupled and Linked Equations Algorithm Revised), for incompressible fluid flow and heat transfer. The CLEAR algorithm avoids introducing pressure-correction and velocity-correction terms, discarding the basic assumption of the SIMPLE-series algorithms. The new algorithm is tested with six examples on a staggered grid system, and it is proved that it can enhance convergence rate to a great extent.

In this study, the CLEAR algorithm is extended to a collocated grid system to test its feasibility. To fulfill the fundamental requirement for eliminating the decoupling problem between velocity and pressure and making the solution independent of underrelaxation factor, detailed investigation is made to previous related methods. For some mildly irregular computation domain, the domain extension method [16] is applied, and is further formulated in this study to meet the special requirement on the acollocated grid to treat the solid region in the computational domain. Based on the above preliminary research, the calculation procedures of SIMPLER and CLEAR on acollocated grids are presented for comparison. Six 2-D numerical examples with available solutions are computed to investigate the performance of the CLEAR and SIMPLER algorithms on a collocated grid system.

SOME DISCUSSION ON MIM

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 3-D computation. On nonstaggered grids, such complications can be greatly alleviated. The crucial issue in using nonstaggered grids is how to eliminate the decoupling between pressure and velocity. In the 1980s the momentum interpolation method (MIM) on nonstaggered grid was first presented by Rhie and Chow [17] to avoid the decoupling problem. It was subsequently reformulated by Peric [18] and Majumdar [19]. Later, Majumdar [20] and Miller and Schmidt [21] pointed out that Rhie and Chow's method has the weakness that the solution is underrelaxation factor-dependent to some extent, despite removing the false pressure field. To remedy this unpleasant deficiency, a few schemes have been proposed lately. An easy technique by Kobayashi and Pereira [22] is to set the underrelaxation factor to $\alpha = 1$ 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 nonstaggered grid, the following three aspects should be guaranteed: (1) the algorithm should avoid the checkerboard pressure distribution; (2) the converged resolution should be independent of the underrelaxation factor; (3) the algorithm should possess the required robustness. In order to develop a computation scheme on a collocated grid, which possesses the above-mentioned three features we first briefly review the existing implementation procedures on collocated grids, take advantage of some successful practices, and then propose a new implementation scheme for this study.

In the following, a brief description of the governing equations and the discretization procedure will be presented. For simplicity of presentation, we take 2-D incompressible laminar steady fluid flow in Cartesian coordinates as an example. The

system for a nonstaggered grid is depicted in Figure 1. The governing equations are as follows.
Continuity equation:

$$\frac{\partial(\rho u_f)}{\partial x} + \frac{\partial(\rho v_f)}{\partial u} = 0 \tag{1}$$

Momentum equation:

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

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

Energy equation:

$$\frac{\partial}{\partial}(\rho u_f T) + \frac{\partial}{\partial x}(\rho v_f T) = \frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) + S_T \tag{4}$$

The above four equations can be expressed in a general form:

$$\frac{\partial}{\partial}(\rho u_f \phi) + \frac{\partial}{\partial x}(\rho v_f \phi) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial \phi}{\partial y}\right) + S_\phi \tag{5}$$

where u_f and v_f refer to the interface velocity whose interpolation scheme is the major issue on the nonstaggered grid. Equation (5) is discretized with the finite-volume

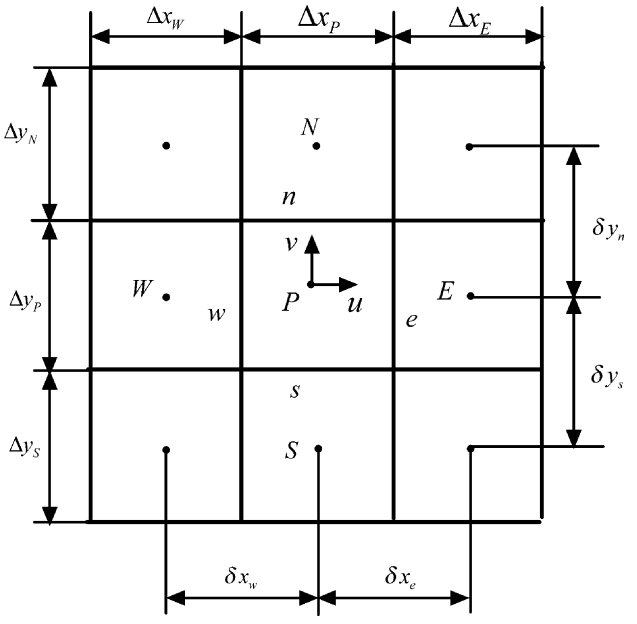


Figure 1. Control volumes of nonstaggered grids in 2-D Cartesian coordinates.

method [16, 23] (FVM) on a nonstaggered grid system and the source term S_ϕ is linearized as

$$S_\phi = S_C + S_P \phi_P \qquad (\text{with } S_P \le 0) \tag{6}$$

By taking out the pressure gradient from the S_ϕ in Eq. (5) for the u component, the final discretization equation for the u component takes the following form, into which underrelaxation is incorporated:

$$\frac{a_P}{\alpha_u} u_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + b_P + \Delta y (p_w - p_e)_P + \frac{1-\alpha_u}{\alpha_u} a_P u_P^0$$

$$= a_E u_E + a_W u_W + a_N u_N + a_S u_S + B_P + \Delta y (p_w - p_e)_P \tag{7}$$

where

$$B_P = b_P + \frac{1-\alpha_u}{\alpha_u} a_P u_P^0 \qquad b_P = S_C \Delta x \Delta y \tag{8}$$

The two terms $(p_w)_P$ and $(p_e)_P$ are linearly interpolated from the neighboring nodes:

$$(p_w)_P = f_w^+ p_P + (1 - f_w^+) p_W \tag{9}$$

$$(p_e)_P = f_e^+ p_E + (1 - f_e^+) p_P \tag{10}$$

where

$$f_w^+ = \frac{\Delta x_W}{2\delta x_w} \qquad f_e^+ = \frac{\Delta x_P}{2\delta x_e} \tag{11}$$

Similar expressions can be formulated for velocity component v_P and the related terms $(p_n)_P$, $(p_s)_P$, and the interpolation coefficients f_s^+, f_n^+. From Eq. (7), we can get

$$u_P = \alpha_u \left(\frac{\sum a_{\text{nb}} u_{\text{nb}} + B_P}{a_P} \right)_P + \frac{\alpha_u \Delta y (p_w - p_e)_P}{(a_P)_P} \tag{12}$$

$$u_E = \alpha_u \left(\frac{\sum a_{\text{nb}} u_{\text{nb}} + B_P}{a_p} \right)_E + \frac{\alpha_u \Delta y (p_w - p_e)_E}{(a_p)_E} \tag{13}$$

To eliminate the checkerboard pressure field, Rhie and Chow [17] proposed that the interface velocity should be determined by the same format equation as the main-grid point velocity. Now, mimicking Eqs. (12) and (13), we express the interface velocity at the cell surface as follows:

$$u_e = \alpha_u \left(\frac{\sum a_{\text{nb}} u_{\text{nb}} + B_P}{a_P} \right)_e + \frac{\alpha_u \Delta y (p_P - p_E)}{(a_P)_e} \tag{14}$$

The notable feature of Eq. (14) is that the $1 - \delta$ difference of pressure, i.e., $p_P - p_E$, is introduced as in the staggered grid, and this is the reason why coupling between pressure and velocity can be ensured. If a checkerboard pressure distribution occurs, use of an equation such as Eq. (14) for the interface velocities will produce some

pseudo sources or sinks of mass flow rate of each control volume and the continuity equation will reject it as an acceptable solution. The terms

$$\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+B_P}{a_P}\right)_e \quad \text{and} \quad \frac{1}{(a_P)_e}$$

are linearly interpolated from the corresponding values of Eqs. (12) and (13), that is,

$$\begin{aligned}\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+B_P}{a_P}\right)_e &= f_e^+\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+B_P}{a_P}\right)_E+(1-f_e^+)\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+B_P}{a_P}\right)_P\\ &= f_e^+\left\{\frac{\sum a_{\mathrm{nb}}u_{nb}+b_P+[(1-\alpha_u)/\alpha_u]\ a_Pu_P^0}{a_P}\right\}_E\\ &\quad +(1-f_e^+)\left\{\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P+[(1-\alpha_u)/\alpha_u]a_Pu_P^0}{a_P}\right\}_P\\ &= f_e^+\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_E+f_e^+\frac{1-\alpha_u}{\alpha_u}u_E^0\\ &\quad +(1-f_e^+)\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_P+(1-f_e^+)\frac{1-\alpha_u}{\alpha_u}u_P^0 \qquad (15)\end{aligned}$$

$$\frac{1}{(a_P)_e}=f_e^+\frac{1}{(a_P)_E}+(1-f_e^+)\frac{1}{(a_P)_P} \qquad (16)$$

Substituting Eqs. (15) and (16) into Eq. (14), we obtain

$$\begin{aligned}u_e=\alpha_u\Bigg\{&f_e^+\left(\frac{\sum \alpha_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_E+(1-f_e^+)\left(\frac{\sum \alpha_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_P\\ &+\left[f_e^+\frac{1}{(a_P)_E}+(1-f_e^+)\frac{1}{(a_P)_P}\right]\Delta y(p_P-p_E)\Bigg\}\\ &+(1-\alpha_u)\left[f_e^+u_E^0+(1-f_e^+)u_P^0\right] \qquad (17)\end{aligned}$$

We define

$$\begin{aligned}u_{\mathrm{MIM}^*} &= f_e^+\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_E+(1-f_e^+)\left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_P\\ &\quad +\left[f_e^+\frac{1}{(a_P)_E}+(1-f_e^+)\frac{1}{(a_P)_P}\right]\Delta y(p_P-p_E)\\ &= \left(\frac{\sum a_{\mathrm{nb}}u_{\mathrm{nb}}+b_P}{a_P}\right)_e+\frac{\Delta y(p_P-p_E)}{(a_P)_e} \qquad (18)\end{aligned}$$

$$u_{\mathrm{LN}} = f_e^+ u_E^0 + (1 - f_e^+) u_P^0 \tag{19}$$

where u_{MIM^*} represents the momentum interpolation, which does not include the underrelaxation factor of velocity, hence it is obviously independent of the underrelaxation factor. The term u_{LN} is the liner interpolated one. Then we get

$$u_e = \alpha_u u_{\mathrm{MIM}^*} + (1 - \alpha_u) u_{\mathrm{LN}} \tag{20}$$

Thus it can be seen that when the underrelaxation factor is incorporated into the solution procedure of the momentum equations, the interpolation proposed by Rhie and Chow, Eq. (14), is actually a mixing interpolation composed of momentum and linear interpolation terms that is obviously affected by the underrelaxation factor. To develop an appropriate interpolation scheme which can discard this weakness of Rhie and Chow's scheme, we reformulate Eqs. (12) and (13) as follows:

$$u_P = \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}} + b_P}{a_P} \right)_P + \frac{\alpha_u \Delta_y (p_w - p_e)_P}{(a_P)_P} + (1 - \alpha_u) u_P^0 \tag{21}$$

$$u_E = \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}} + b_P}{a_P} \right)_E + \frac{\alpha_u \Delta_y (p_w - p_e)_E}{(a_P)_E} + (1 - \alpha_u) u_E^0 \tag{22}$$

Hence the interface value u_e can be expressed as

$$\begin{aligned} u_e &= \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}} + b_P}{a_P} \right)_e + \frac{\alpha_u \Delta y (P_p - p_E)}{(a_P)_e} + (1 - \alpha_u) u_e^0 \\ &= \alpha_u u_{\mathrm{MIM}^*} + (1 - \alpha_u) u_e^0 \end{aligned} \tag{23a}$$

It is worth noting that by reformulating Eqs. (12) and (13) into the form of Eqs. (21) and (22), respectively, the terms $(1 - \alpha_u) u_P^0$ and $(1 - \alpha_u) u_E^0$ are separated explicitly, and the mimicking of Eqs. (21) and (22) leads to Eq. (23*a*), which is different from the original MIM of Rhie and Chow, in which the term u_{LN} is replaced by u_e^0. When iteration converges, u_e and u_e^0 approach the same value, and Eq. (23*a*) is equivalent to $u_e = u_{\mathrm{MIM}^*}$, which is independent of the underrelaxation factors. For comparison, Eq. (23*a*) is rewritten as follows:

$$\begin{aligned} u_e &= \alpha_u u_{\mathrm{MIM}^*} + \alpha_y u_{\mathrm{LN}} + (1 - \alpha_u) u_e^0 - \alpha_u u_{\mathrm{LN}} \\ &= \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}} + B_P}{a_P} \right)_e + \frac{\alpha_u \Delta y (p_P - p_E)}{(a_P)_e} \\ &\quad + (1 - \alpha_u) u_e^0 - \alpha_u \left[f_e^+ u_E^0 + (1 - f_e^+) u_P^0 \right] \end{aligned} \tag{23b}$$

Equation (23*a*) or (23*b*) is the modified momentum interpolation method (MMIM) by Majumdar [20], which is later reprovided by Choi [24] for unsteady flow. Yu et al. [25] gave a detailed discussion of the role of interface velocity, and recommended that all the interface velocities should be obtained with the momentum interpolation method in the overall calculation process of u_f and v_f, and Eq. (23*a*) will be used in this study for the interpolation.

GENERAL REVIEW OF SIMPLER ALGORITHM

For the implementation of the SIMPLER algorithm on the collocated grid, the discretized momentum equation is expressed by Eq. (7). In order to derive the pressure equation, Eq. (23*a*) can be rewritten for the u component as

$$u_e = \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u^0_{\mathrm{nb}} + b_P}{a_P} \right)_e + (1 - a_u) u^0_e + \frac{\alpha_u \, \Delta y (p_P - p_E)}{(a_P)_e}$$

$$= \widehat{\mathbf{u}}^{\mathbf{0}}_{\boldsymbol{e}} + d_{ef}(p_P - p_E) \tag{24}$$

where

$$\widehat{\mathbf{u}}^{\mathbf{0}}_{\boldsymbol{e}} = \alpha_u \left(\frac{\sum a_{\mathrm{nb}} u^0_{\mathrm{nb}} + b_P}{a_P} \right)_e + (1 - \alpha_u) u^0_e$$

$$= \alpha_u \left[f^+_e \left(\frac{\sum a_{\mathrm{nb}} u^0_{\mathrm{nb}} + b_P}{a_P} \right)_E + (1 - f^+_e) \left(\frac{\sum a_{\mathrm{nb}} u^0_{\mathrm{nb}} + b_P}{a_P} \right)_P \right] + (1 - \alpha_u) u^0_e \tag{25a}$$

$$d_{ef} = \frac{\alpha_u \, \Delta y}{(a_P)_e} = \left[f^+_e \frac{1}{(a_P)_E} + (1 - f^+_e) \frac{1}{(a_P)_P} \right] \alpha_u \, \Delta y \tag{25b}$$

Similarly, we can get the following equations for the v component:

$$v_n = \alpha_v \left(\frac{\sum a_{\mathrm{nb}} v^0_{\mathrm{nb}} + b_P}{a_P} \right)_n + (1 - \alpha_v) v^0_n + \frac{\alpha_u \, \Delta x (p_P - p_N)}{(a\hat{p})_n}$$

$$= \widehat{\boldsymbol{v}}^{\mathbf{0}}_{\boldsymbol{n}} + d_{nf}(p_P - p_N) \tag{26}$$

where

$$\widehat{\boldsymbol{v}}^{\mathbf{0}}_{\boldsymbol{n}} = \alpha_v \left(\frac{\sum a_{\mathrm{nb}} v^0_{\mathrm{nb}} + b_P}{a_P} \right)_n + (1 - \alpha_v) v^0_n$$

$$= \alpha_v \left[f^+_n \left(\frac{\sum a_{\mathrm{nb}} v^0_{\mathrm{nb}} + b_P}{a_P} \right)_N + (1 - f^+_n) \left(\frac{\sum a_{\mathrm{nb}} v^0_{\mathrm{nb}} + b_P}{a_P} \right)_P \right]$$

$$+ (1 - \alpha_v) v^0_n \tag{27a}$$

$$d_{nf} = \frac{\alpha_v \, \Delta x}{(a_P)_n} = \left[f^+_n \frac{1}{(a_P)_N} + (1 - f^+_n) \frac{1}{(a_P)_P} \right] \alpha_v \, \Delta x \tag{27b}$$

Substituting Eqs. (24) and (26) into the following discretized form of the continuous equation,

$$(\rho u)_e A_e - (\rho u)_w A_w + (\rho v)_n A_n - (\rho v)_s A_s = 0 \tag{28}$$

we have the following equations for pressure:

$$a_P p^*_P = \sum a_{\mathrm{nb}} p^*_{\mathrm{nb}} + b \tag{29}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{30}$$

$$a_E = (\rho Ad)_{ef} \qquad a_W = (\rho Ad)_{wf} \qquad a_N = (\rho Ad)_{nf} \qquad a_s = (\rho Ad)_{sf} \tag{31a}$$

$$b = (\rho \widehat{u^0} A)_w - (\rho \widehat{u^0} A)_e + (\rho \widehat{v^0} A)_s - (\rho \widehat{v^0} A)_n \tag{31b}$$

The coefficients (a_E, a_W, a_N, a_S) are determined by the previous interface value u_f^0, v_f^0. In the actual implementation, the pressure equation can be incorporated with pressure underrelaxation factor α_P, and the final pressure equation becomes

$$\frac{a_P}{\alpha_P} p_P^* = \sum a_{\mathrm{nb}} p_{\mathrm{nb}}^* + b + \frac{1 - \alpha_P}{\alpha_P} a_p p_P^0 \tag{32}$$

Similar to the derivation process in the staggered grid of the corrector step, the improved velocity can be expressed as

$$u_e = u_e^* + d_{ef}(p'_P - p'_E) \tag{33}$$

$$v_n = v_n^* + d_{nf}(p'_P - p'_N) \tag{34}$$

and the interface velocity-correction terms are

$$u'_e = d_{ef}(p'_P - p'_E) \tag{35}$$

$$v'_n = d_{nf}(p'_P - p'_N) \tag{36}$$

Substituting Eqs. (33) and (34) into the discretized continuous equation, Eq. (28), yields

$$a_P p'_P = \sum a_{\mathrm{nb}} p'_{\mathrm{nb}} + b \tag{37}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{38}$$

$$a_E = (\rho Ad)_{ef} \qquad a_W = (\rho Ad)_{wf} \qquad a_N = (\rho Ad)_{nf} \qquad a_S = (\rho Ad)_{sf} \tag{39a}$$

$$b = (\rho u^* A)_w - (\rho u^* A)_e + (\rho v^* A)_s - (\rho v^* A)_n \tag{39b}$$

It is noted that Eq. (39*a*) is identical to Eq. (31*a*), while in Eq. (39*b*), the intermediate velocity is used instead of the initial pseudo-velocity in Eq. (31*b*).

Similarly, the velocity in the main-grid point is also corrected, and the improved value is expressed as

$$u_P = u_p^* + d_P^u (p'_w - p'_e)_P \tag{40}$$

$$v_P = v_E^* + d_P^v (p'_s - p'_n)_P \tag{41}$$

where

$$d_P^u = \frac{\alpha_u \Delta y}{(a_P)_P} \tag{42}$$

$$d_P^v = \frac{\alpha_v \Delta y}{(a_P)_P} \tag{43}$$

The pressure-correction values in the interface are linearly interpolated as

$$(p'_w)_P = f_w^+ p'_P + (1 - f_w^+) p'_W \tag{44}$$

$$(p'_e)_P = f_e^+ p'_E + (1 - f_e^+) p'_P \tag{45}$$

$$(p'_s)_P = f_s^+ p'_P + (1 - f_s^+) p'_S \tag{46}$$

$$(p'_n)_P = f_n^+ p'_N + (1 - f_n^+) p'_P \tag{47}$$

The computational steps of the SIMPLER algorithm on the collocated grid are summarized as follows.

Step 1. Assume the initial field of the main node and interface velocities u_P^0, v_P^0, u_f^0, v_f^0.

Step 2. Calculate the descritization coefficient of the momentum equation and pseudo-velocity $\widehat{u_e^0}$ [Eq. (25*a*)] and $\widehat{v_n^0}$ [Eq. (27*a*)] to determine the source term for the pressure equation, Eq. (32), based on the previous interface and main-grid point velocities u_P^0, v_P^0, u_f^0, v_f^0.

Step 3. Calculate the coefficients d_{ef} [Eq. (25*b*)] and d_{nf} [Eq. (27*b*)] of the pressure equation, Eq. (32), with u_f^0, v_f^0.

Step 4. Solve the discretized forms of the pressure equation and obtain the pressure field p^*.

Step 5. Based on p^*, solve the discretized forms of the momentum equation, obtaining the intermediate velocity field u_P^*, v_P^*.

Step 6. Calculate the interface velocity with MMIM based on u_P^*, v_P^*, p^*, and the previous discretized momentum equation coefficients derived from u_f^0, v_f^0, obtaining the intermediate interface velocities u_f^*, v_f^* to determine the source term of the pressure-correction equation.

Step 7. Solve the pressure-correction equation, Eq. (37), obtaining the pressure correction value p'.

Step 8. Correct the interface velocity with Eqs. (33) and (34), and correct the main-grid point velocities with Eqs. (40) and (41).

Step 9. Solve the discretization equations of the other scalar variables if necessary.

Step 10. Return to step 2 and repeat until convergence is reached.

From the above procedure we can see that a checkerboard pressure field, if it occurs, will be damped out due to the introduction of MMIM, and the solution will be underrelaxation-independent.

MATHEMATICAL FORMULATION OF CLEAR ALGORITHM

As indicated in [14, 15], the main difference between SIMPLER and CLEAR is the way to obtain the improved velocity, which satisfies the mass conservation condition. In the SIMPLER algorithm, this improved velocity is obtained by adding a correction term to the intermediate velocity, while in the CLEAR algorithm the improved velocity is obtained directly from an improved pressure field.

Similar to the expression for the revised velocity on the staggered grid of the CLEAR algorithm [14, 15], the improved velocities on a collocated grid are expressed as

$$u_e = \beta_u \left(\frac{\sum a_{\mathrm{nb}} u^*_{\mathrm{nb}} + b_P}{a_P} \right)_e + (1 - \beta_u) u^*_e + \frac{\beta_u \, \Delta y (p_P - p_E)}{(a_P)_e} \tag{48}$$

$$v_n = \beta_v \left(\frac{\sum a_{\mathrm{nb}} v^*_{\mathrm{nb}} + b_P}{a_P} \right)_n + (1 - \beta_v) v^*_n + \frac{\beta_v \, \Delta x (p_P - p_N)}{(a_P)_n} \tag{49}$$

The terms

$$\left(\frac{\sum a_{\mathrm{nb}} u^*_{\mathrm{nb}} + b_P}{a_P} \right)_e \qquad \text{and} \qquad \left(\frac{\sum a_{\mathrm{nb}} v^*_{\mathrm{nb}} + b_P}{a_P} \right)_n$$

in the interface are interpolated from the neighboring main-grid points. The parameter β is called the second relaxation factor as defined in [14, 15], and its value is recommended as

$$\beta_{\hat{u}} = \beta_{\hat{v}} = \beta = \begin{cases} 0.5 & 0 < \alpha \le 0.5 \\ 1 & 0.5 < \alpha \le 1 \end{cases} \tag{50}$$

It is noted that Eqs. (48) and (49) have the same form as the MMIM, Eq. (23*a*), so the interface velocities defined by Eqs. (48) and (49) are also independent of relaxation factor β when iteration converges.

In order to derive the equation for improved pressure on a collocated grid, we define:

$$\begin{aligned} \widehat{u^*_e} &= \beta_u \left(\frac{\sum a_{\mathrm{nb}} u^*_{\mathrm{nb}} + b_P}{a_P} \right)_e + (1 - \beta_u) u^*_e \\ &= \beta_u \left[f_e^+ \left(\frac{\sum a_{\mathrm{nb}} u^*_{\mathrm{nb}} + b_P}{a_P} \right)_E + (1 - f_e^+) \left(\frac{\sum a_{\mathrm{nb}} u^*_{\mathrm{nb}} + b_P}{a_P} \right)_P \right] + (1 - \beta_u) u^*_e \end{aligned} \tag{51a}$$

$$d_{ef} = \frac{\beta_u \, \Delta y}{(a_P)_e} = \left[f_e^+ \frac{1}{(a_P)_E} + (1 - f_e^+) \frac{1}{(a_P)_P} \right] \beta_u \, \Delta y \tag{51b}$$

$$\begin{aligned} \widehat{v^*_n} &= \beta_v \left(\frac{\sum a_{\mathrm{nb}} v^*_{\mathrm{nb}} + b_P}{a_P} \right)_n + (1 - \beta_v) v^*_n \\ &= \beta_v \left[f_n^+ \left(\frac{\sum a_{\mathrm{nb}} v^*_{\mathrm{nb}} + b_P}{a_P} \right)_N + (1 - f_n^+) \left(\frac{\sum a_{\mathrm{nb}} v^*_{\mathrm{nb}} + b_P}{a_P} \right)_P \right] + (1 - \beta_v) v^*_n \end{aligned} \tag{52a}$$

$$d_{nf} = \frac{\beta_v \Delta x}{(a_P)_n} = \left[f_n^+ \frac{1}{(a_P)_N} + (1 - f_n^+) \frac{1}{(a_P)_P} \right] \beta_v \Delta x \tag{52b}$$

Equations (48) and (49) are rewritten for simplicity as

$$u_e = \widehat{u_e^*} + d_{ef}(p_P - p_E) \tag{53}$$

$$v_n = \widehat{v_n^*} + d_{nf}(p_P - p_N) \tag{54}$$

Equations (53) and (54) are substituted into the continuous equation, Eq. (28) to get

$$a_P p_P = \sum a_{\mathrm{nb}} p_{\mathrm{nb}} + b \tag{55}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{56}$$

$$a_E = (\rho A d)_{ef} \qquad a_W = (\rho A d)_{wf} \qquad a_N = (\rho A d)_{nf} \qquad a_S = (\rho A d)_{sf} \tag{57a}$$

$$b = (\rho \widehat{u^*} A)_w - (\rho \widehat{u^*} A)_e + (\rho \widehat{v^*} A)_s - (\rho \widehat{v^*} A)_n \tag{57b}$$

The coefficients a_E, a_W, a_N, a_S are calculated based on the intermediate field u_f^*, v_f^*. In the numerical implementation for the present study, it is found that the improved pressure equation should also be underrelaxed to enhance the robustness. The equation is then expressed as

$$\frac{a_P}{\alpha_P} p_P = \sum a_{\mathrm{nb}} p_{\mathrm{nb}} + b + \frac{1 - \alpha_P}{\alpha_P} a_P p_P^* \tag{58}$$

The parameter β is the relaxation factor for the updated interface pseudo-velocity. The larger the value of β, the larger the variation of the interface pseudo-velocity, and the smaller the relative change between the improved and the previous pressure. And for a case with high value of α_u, the improved pressure should be highly underrelaxed, to guarantee the convergence of the solution procedure, and this can be done by setting a larger value of β, sometimes even greater than 1. Hence, due to the good coupling of velocity and pressure in the CLEAR algorithm, the improved pressure should be underrelaxed. There are two approaches to underrelax the improved pressure: one is to give a larger second relaxation factor β, the other is to incorporate a pressure underrelaxation factor α_P to the improved pressure equation. Our practice shows that the value α_P may be taken within the range 0.7–1.0.

For the improved velocity on the main-grid point, Eq. (21) is rewritten with intermediate values as

$$u_P = \beta_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}}^* + b_P}{a_P} \right)_P + \frac{\beta_u \Delta y (p_w - p_e)_P}{(a_P)_P} + (1 - \beta_u) u_P^* \tag{59}$$

Defining:

$$\widehat{u_P^*} = \beta_u \left(\frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}}^* + b_P}{a_P} \right)_P + (1 - \beta_u) u_P^* \tag{60a}$$

$$d_P^u = \frac{\beta_u \Delta y}{(a_P)_P} \tag{60b}$$

then Eq. (59) can be recast into

$$u_P = \widehat{u_P^*} + d_P^u (p_w - p_e)_P \tag{61}$$

Similarly, we can get the improved main-grid point velocity for the v component:

$$v_P = \widehat{v_P^*} + d_P^v (p_s - p_n)_P \tag{62}$$

where

$$\widehat{v_P^*} = \beta_v \left(\frac{\sum a_{\mathrm{nb}} v_{\mathrm{nb}}^* + b_P}{a_P} \right)_P + (1 - \beta_v) v_P^* \tag{63a}$$

$$d_P^v = \frac{\beta_v \Delta y}{(a_P)_P} \tag{63b}$$

The solution procedure of the CLEAR algorithm on a collocated grid is summarized as follows.

Step 1. Assume the initial field of the main node and interface velocities u_P^0, v_P^0, u_f^0, v_f^0.

Step 2. Calculate the descritization coefficient of the momentum equation and pseudo-velocities $\widehat{u_e^0}$ [Eq. (25a)] and $\widehat{v_n^0}$ [Eq. (27a)] to determine the source term for the pressure equation, Eq. (32), based on the previous interface and main node velocities u_P^0, v_P^0, u_f^0, v_f^0.

Step 3. Calculate the coefficients d_{ef} [Eq. (25b)] and d_{nf} [Eq. (27b)] of the pressure equation with u_f^0, v_f^0.

Step 4. Solve the pressure equation and obtain pressure field p^*.

Step 5. Based on p^*, solve the discritized forms of the momentum equation, obtaining the intermediate velocity field u_P^*, v_P^*.

Step 6. Calculate the interface velocity with MMIM based on u_P^*, v_P^*, p^*, and the previous discretized momentum equation coefficient, which is derived from u_f^0, v_f^0, obtaining the intermediate interface velocity u_f^*, v_f^*.

Step 7. Based on the intermediate interface velocities u_f^*, v_f^*, recalculate the momentum equation coefficients and obtain the improved pseudo-velocities $\widehat{u_e^*}$ [Eq. (51a)] and $\widehat{v_n^*}$ [Eq. (52a)] by the intermediate main node velocity to determine the source term of the improved pressure equation.

Step 8. Calculate the coefficients d_{ef} [Eq. (51b)] and d_{nf} [Eq. (52b)] of the improved pressure equation, Eq. (58), with u_f^*, v_f^*.

Step 9. Solve the improved pressure equation and obtain the updated pressure field p.
Step 10. Update the interface velocity with Eqs. (53) and (54), correcting the main-grid point velocities with Eqs. (61) and (62).
Step 11. Solve the discretization equations of the other scalar variables if necessary.
Step 12. Return to step 2 and repeat until convergence is reached.

From the above solution procedure, we can see that the first six steps are the same for the two algorithms. In the CLEAR algorithm the pressure equation is solved twice—the first time it is solved to determine the source term of momentum equation, and the second time it is solved to gain the improved pressure to correct the intermediate velocity—so the number of the solved equations is equal to SIMPLER, while in step 7, the coefficient of discretized moment equation is recalculated without solving, which leads to a bit longer CPU time for the new algorithm in one iteration.

Before going into the details of application examples, three issues will be addressed in regard to the CLEAR algorithm. First, the meaning of the adopted word "couple" is explained. It is a widely accepted concept that in incompressible flow the momentum and continuity equations, or the velocity and pressure, are coupled [26–28], and it is in this sense that the word is used in the acronym of CLEAR, even though the word "couple" may have another meanings—for example, the coupling between momentum and energy equations—the simultaneous solutions of the discretized velocity and pressure equations, etc. In our context, however, it refers to the inherent interrelationship between velocity and pressure. Second, as indicated above, the CLEAR algorithm differs from all SIMPLE-like algorithms in that no correction terms are introduced in the solution procedure of CLEAR. Taking the PISO algorithm as an example, PISO is actually an extension of SIMPLE [28, 29] in that PISO is a two-step algorithm: in the first step the calculation procedure is exactly the same as SIMPLE, and in the second step a further correction in pressure is introduced to make the satisfaction of the continuity condition better. Thus it is quite clear that CLEAR and PISO are totally different in this sense. Third, as far as the theoretical aspect is concerned, the solution oscillation on the collocated grids may be analyzed by the Fourier analysis method [30]. The focus of this article, however, is in the implementation of a collocated grid. Therefore such analysis is not conducted. As far as the extensions of CLEAR to 3-D and curvilinear coordinates are concerned, study is now underway in the authors' group and the results will be reported elsewhere.

SIX TYPICAL NUMERICAL EXAMPLES

To verify the feasibility of the CLEAR algorithm on the collocated grid, six typical numerical examples with available solutions are computed. The iteration number and CPU time consumed are compared between the SIMPLER and CLEAR algorithms with variation of time-step multiple. The time-step multiple is defined by Eq. (64) [23]:

$$E = \frac{\alpha}{1-\alpha} \qquad (0 < \alpha < 1) \tag{64}$$

The convergence criterion is

$$\mathrm{Rs}_{\mathrm{cv}} = \underset{\mathrm{cv}}{\mathrm{MAX}} \left[\frac{(\rho u^* A)_w - (\rho u^* A)_e + (\rho v^* A)_s - (\rho v^* A)_n}{\mathrm{flow}_{\mathrm{ch}}} \right] \leq \varepsilon \tag{65}$$

where $\mathrm{Rs}_{\mathrm{cv}}$ is the maximum mass residual of each control volume and $\mathrm{flow}_{\mathrm{ch}}$ is the characteristic flow rate defined in [15]. The value of ε is individually defined in each example. The pressure underrelaxation factor is 0.85 for the first four examples and 0.95 for the last two as the default value. For cases where a larger value of β and or a smaller α_P is used, special description will be provided.

Tests of Forced-Convective Fluid Flow

Example 1: Lid-driven cavity flow. The calculations are carried out for $\mathrm{Re} = 100$. A uniform grid of 52×52 is applied with $\varepsilon = 5.0 \times 10^{-8}$. The Reynolds number is defined by

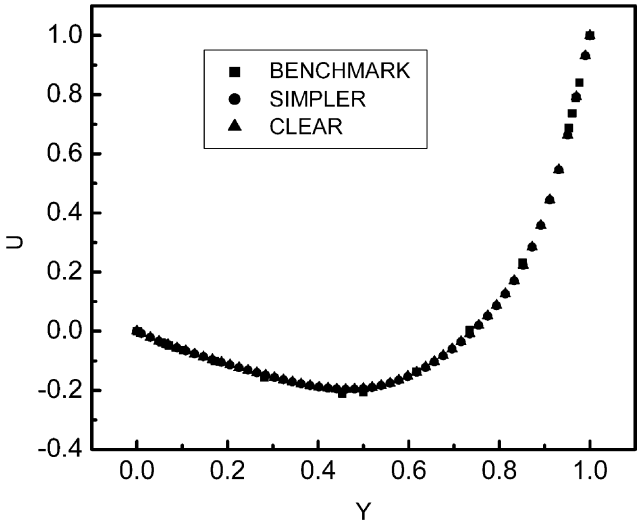
$$\mathrm{Re} = \frac{U_{\mathrm{lid}} L}{\nu} \tag{66}$$

The velocity distribution along the two centerlines compared with the benchmark solution from [31] are presented in Figure 2 to verify the accuracy of the new algorithm, where X and Y are nondimensional coordinates, normalized by the cavity height. It can be seen that the solutions from SIMPLER and CLEAR are almost identical. The iteration number and the CPU time consumed are displayed in Figure 3. In Figure 4, the ratios of iteration number and CPU time of CLEAR over that of SIMPLER varied with time-step multiple are presented. In these figures, the part shown by the dashed lines is obtained by taking $\beta = 1.3$, $\alpha_P = 0.8$ for a velocity underrelaxation factor of 0.9. Here underrelaxation of the improved pressure is treated simultaneously in two ways: adding the second relaxation factor and decreasing the pressure underrelaxation factor. It can be seen that the ratio of iteration number ranges from 0.16 to 0.65 and that of CPU time ranges from 0.19 to 0.77. The saving of iteration number and CPU time is appreciable.

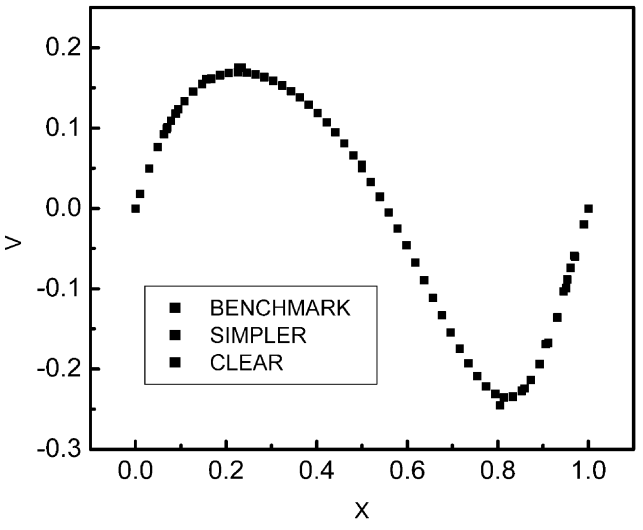
Example 2: Lid-driven cavity flow in a polar cavity. Figure 5 shows the configuration of the polar cavity ($\theta = 1$ rad, $\delta/R_{\mathrm{in}} = 1$). Fuchs and Tillmark [32] have studied the problem experimentally and numerically. The Reynolds number is defined as

$$\mathrm{Re} = \frac{U_{\mathrm{lid}} \delta}{\nu} \tag{67}$$

where U_{lid} is the circumferential velocity of the moving lid, $U_{\mathrm{lid}} = R_0 \times \omega$. A uniform grid system, 52×52, is used in our computation ($\varepsilon = 5.0 \times 10^{-8}$). The predicted streamfunctions for a Re number of 350 from the two algorithms are nearly the same, as shown in Figure 6. In Figures 7 and 8, the comparisons are



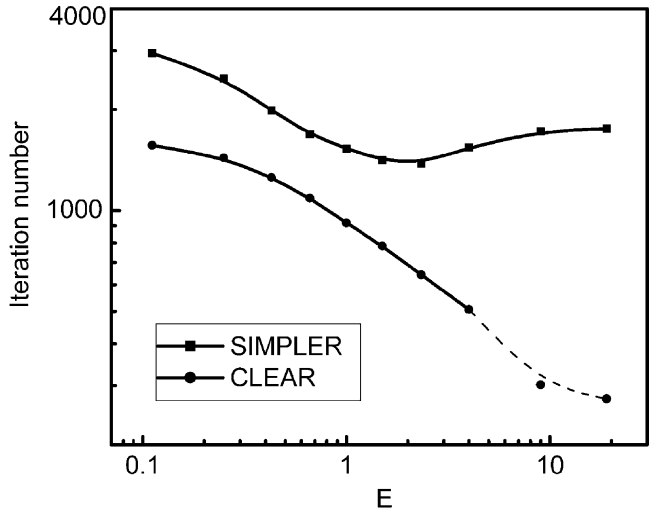
(a) U component distribution comparison along X=0.5



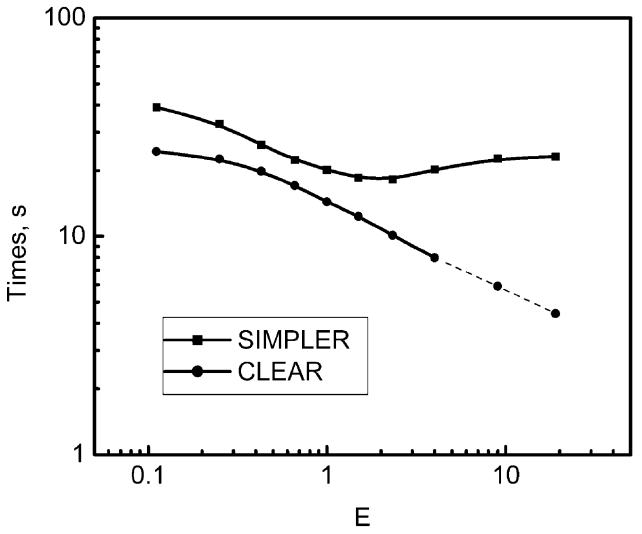
(b) V component distribution comparison along Y=0.5

Figure 2. Predicted velocity distributions for Re = 100 of Example 1.

presented. Again, for the high underrelaxation factor (0.9), the β value is greater than 1 (here it is 1.3), and the corresponding pressure underrelaxation factor is 0.75, presented by the dashed lines. It can be found that the new algorithm performs superiorly to SIMPLER. The iteration number and CPU time of the CLEAR are 14–67% and 19–90%, respectively, of that of the SIMPLER for Re = 350.



(a) Iteration number



(b) CPU time

Figure 3. Comparison of iteration numbers and CPU time for Re = 100 of Example 1.

Tests with Natural Convection

Example 3: Natural convection in a square cavity. The square cavity has two adiabatic walls (top and bottom), and its two vertical walls are maintained at constant but different temperatures. We use a uniform grid of 82×82 for computation with $\mathrm{Ra} = 10^4$, $\varepsilon = 2.0 \times 10^{-7}$ based on the Boussinesq assumption.

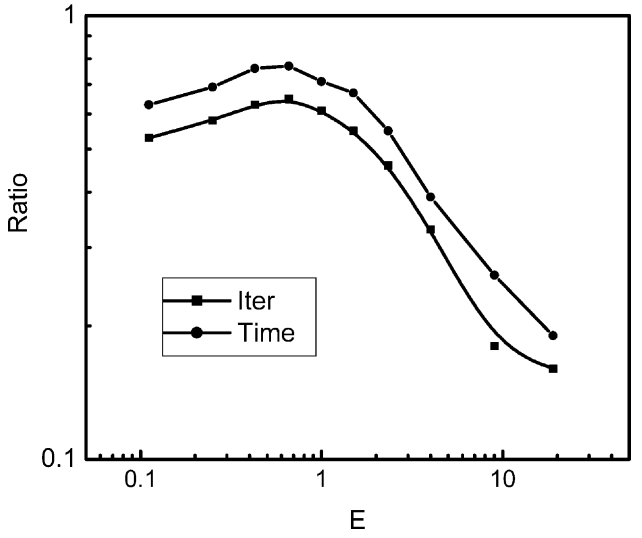


Figure 4. Ratios of the iteration number and the CPU time of CLEAR versus SIMPLER.

The second relaxation factor $\beta = 1.2$ for $\alpha = 0.9$. The Rayleigh number is defined by

$$\mathrm{Ra} = \frac{\rho g \beta L^3 \Delta T}{a\mu} \tag{68}$$

The cavity average Nusselt number is 2.238 from the benchmark solution of [33]. The value from the present study is 2.24, which is in good agreement with the benchmark solution. Figure 9 shows the variation of the iteration number and CPU time with the time-step multiple for the two algorithms. The ratio of the iteration

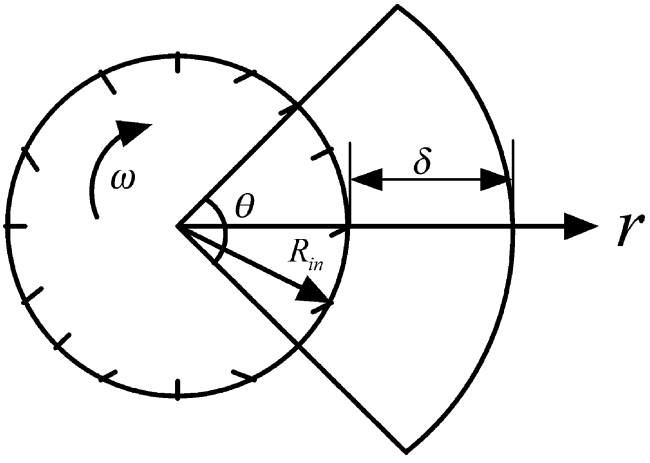
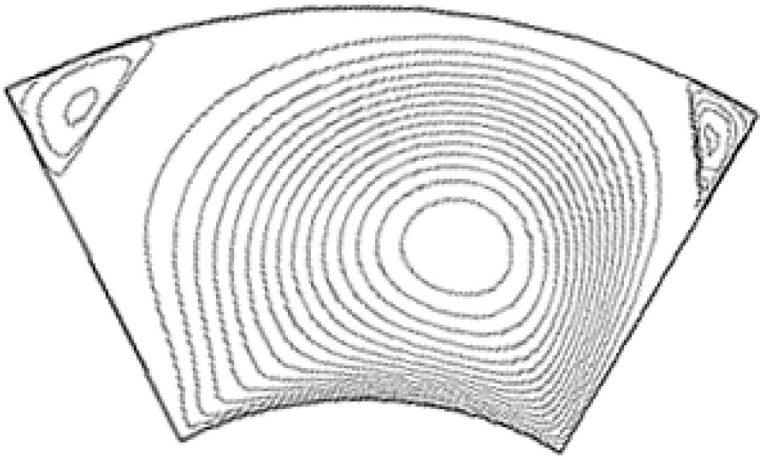
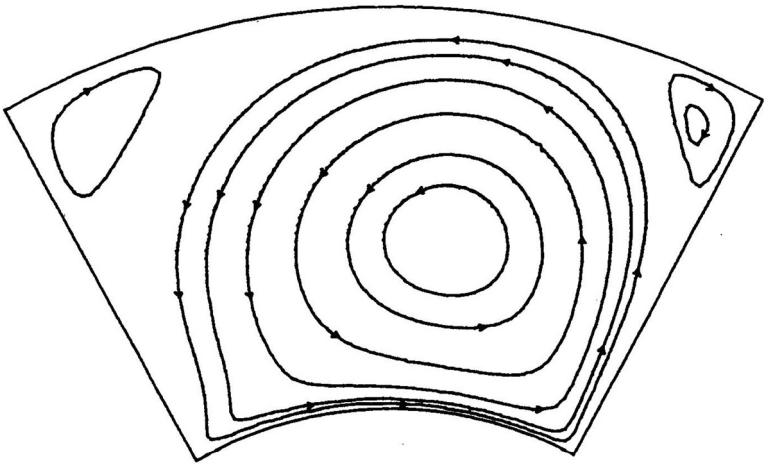


Figure 5. Lid-driven cavity in a polar cavity.



(a) Prediction in the present study



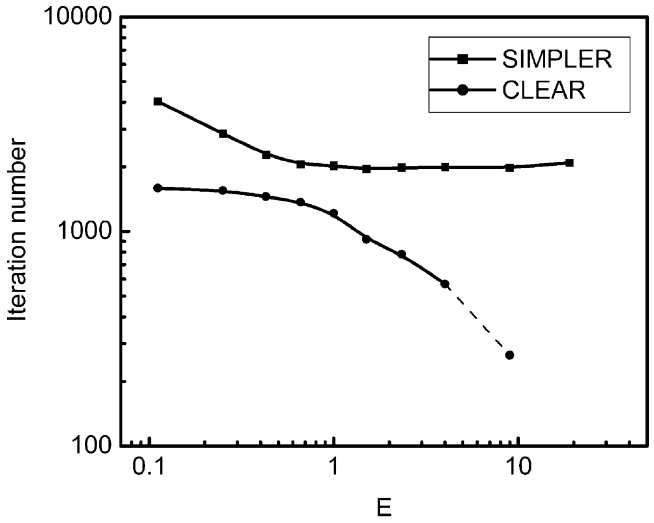
(b) Results of [32]

Figure 6. Predicted stream function of Example 2 (Re = 350).

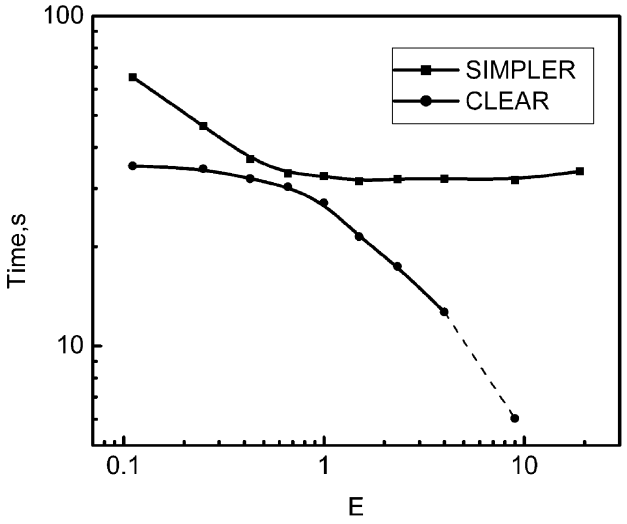
number ranges from 0.31 to 0.39 and the value of the CPU time is from 0.37 to 0.45 (Figure 10).

Example 4: Natural convection in an annular enclosure. The diagram of the laminar natural convection between two horizontal concentric cylinders is depicted in Figure 11. The case studied is for $\mathrm{Ra} = 10^4$, where Rayleigh number is defined by

$$\mathrm{Ra} = \frac{\rho g \beta \delta^3 \Delta T}{a\mu} \tag{69}$$



(a) Iteration number



(b) CPU tim

Figure 7. Comparison of iteration number and CPU time for Re = 350 of Example 2.

Computations were conducted on a uniform grid system with 42×32 meshes with $\varepsilon = 2.0 \times 10^{-7}$. Figure 12 shows the numerical results including flow field and temperature field with the CLEAR algorithm, where the results of [34] are also provided for comparison. The performance comparison results are displayed in Figures 13 and 14, and the dashed line in the figure represents $\beta = 1.3$ for $\alpha = 0.9$. The ratio of the iteration number varies from 0.1 to 0.3 and the CPU time ratio varies from 0.12 to 0.38. Obviously, the saving in CPU time is enormous.

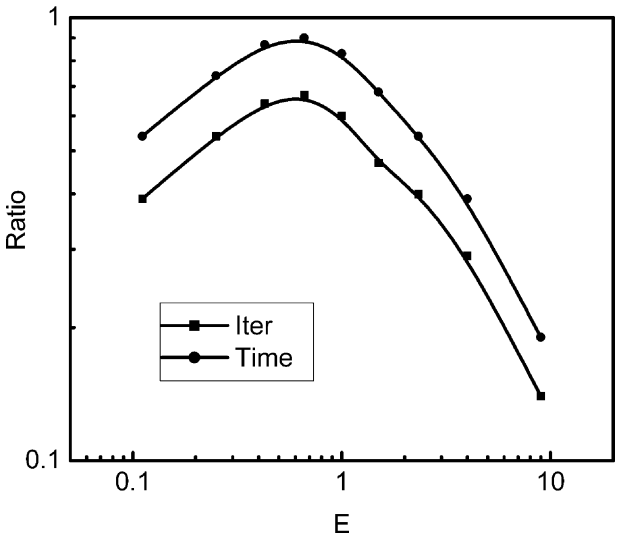
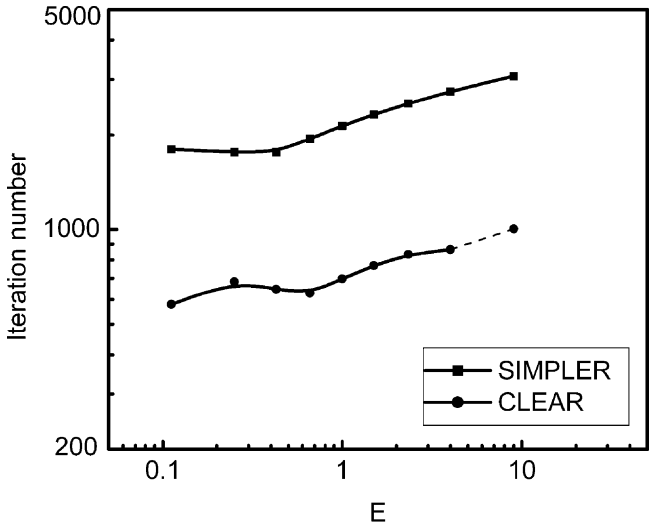


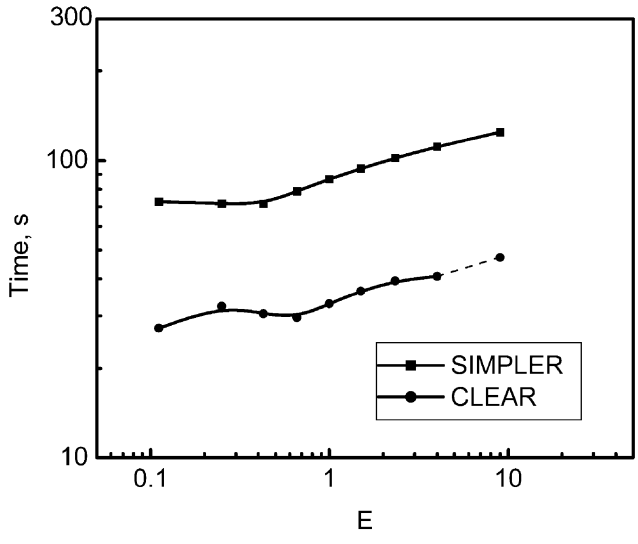
Figure 8. Ratios of iteration numbers and CPU time of Example 2.

Tests of Fluid Flow with Solid Region inside in the Computation Domain

On a staggered grid, it is easy to deal with heat transfer and fluid flow problems of conjugated type, in which the solid exists in the computation domain. If the solid geometry is not very complicated and not isolated from the domain, the domain extension method [16] can be used to treat the solid part by giving the viscosity coefficient a very large value, while for the case with isolated solid region, the large coefficient method [35] can be employed. On a non-staggered grid, however, the above-mentioned numerical techniques are not enough to get a converged solution. This is because on the staggered grid, the interface of the main grid is the location where the velocity component is positioned, and for the conjugated case, the velocity component value at the interface can be easily treated as zero by the above-mentioned techniques. In the collocated grid, the velocity components sit at the same locations as the main grid points, and to guarantee that the velocity at the control volume interface, i.e., the solid/fluid interface, is zero, additional steps must be adopted. In the present study, a refined domain extension method on the collocated grid is proposed by which the solid region can be treated in a uniform way no matter whether it is isolated or not from the computational boundary and the velocity components in the solid region, including at its boundaries, will always be zero. There are two approaches for numerical computation of a conjugated problem: (1) separate computations of the fluid and solid regions and then coupling at the interface boundary; (2) simultaneous computation of both the fluid and solid regions by treating the solid region as a special part of the fluid. Numerical practices often show that the second approach is computationally more efficient [16]. In order to implement the second approach successfully, there are three aspects that should be fully taken into account: (1) It is important to make the velocity in the solid



(a) Iteration number



(b) CPU time

Figure 9. Comparison of iteration number and CPU time for $Ra = 10^6$ of Example 3.

region zero (or almost zero) in the whole iteration process. (2) The interface velocity between solid and fluid should also be always kept at zero in the iteration. (3) In the conjugated computation, the pressure values in the solid region are meaningless and they should not affect the fluid pressure field. In calculating the source term of the momentum equation of the fluid control volume adjacent to the interface, the interface pressure value is interpolated from the fluid side.

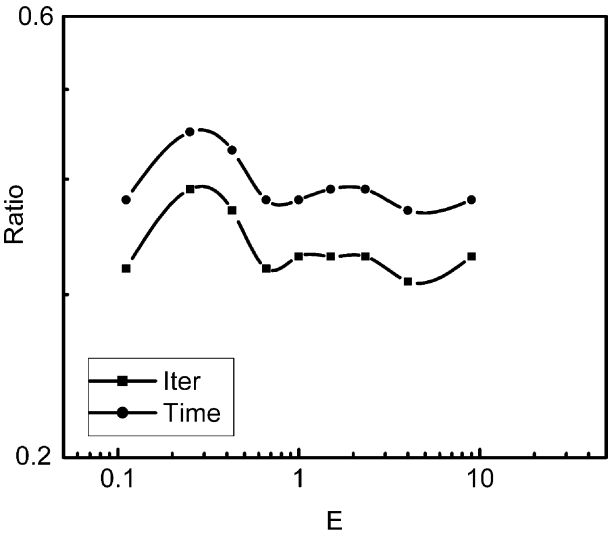


Figure 10. Ratios of iteration numbers and CPU time of Example 3.

The following two numerical examples are concerned with this kind of problem; and the solid part is treated by the above-mentioned techniques to get the converged solution.

Example 5: Laminar fluid flow over an annular backward step. The computation domain is depicted in Figure 15. The specific dimensions are

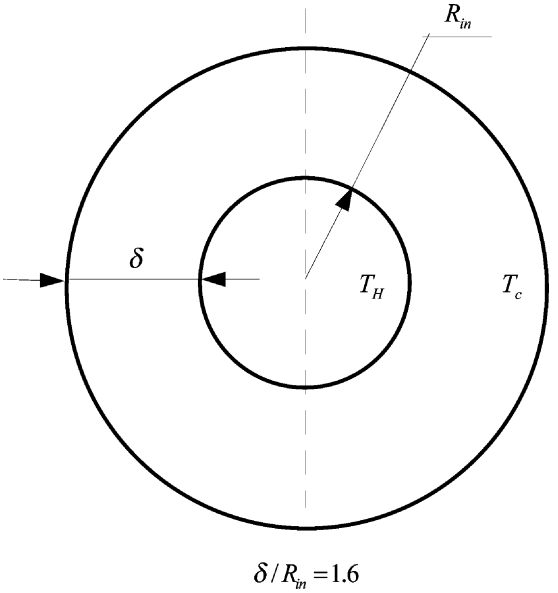
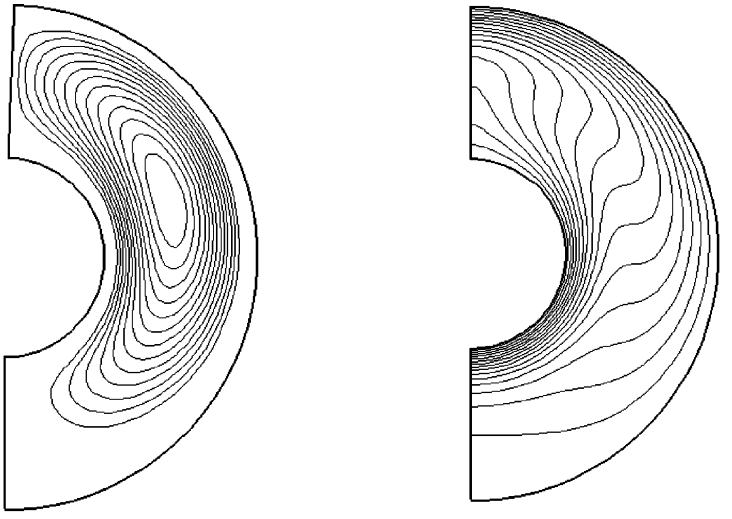
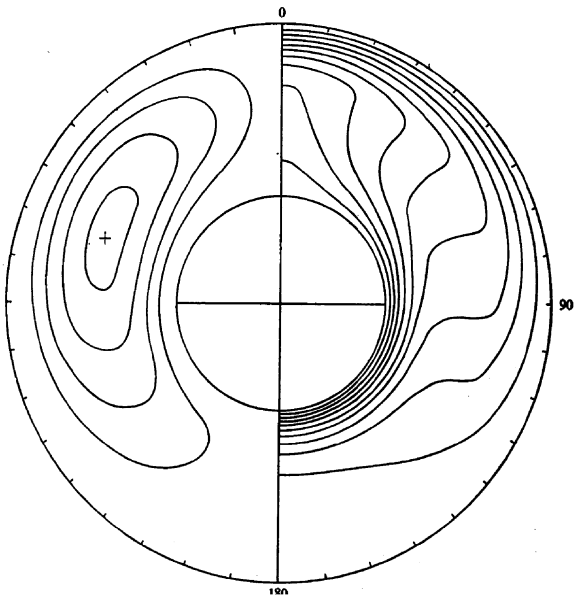


Figure 11. Natural convection in an annular space.

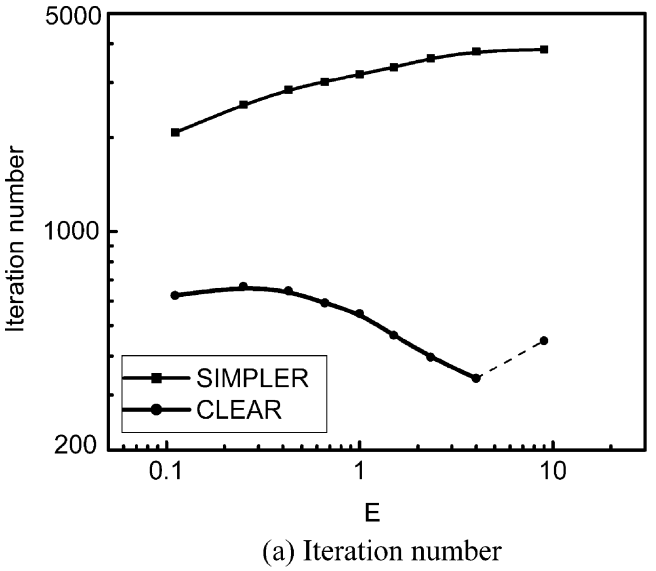


(a) Stream functions (b) Isothermals

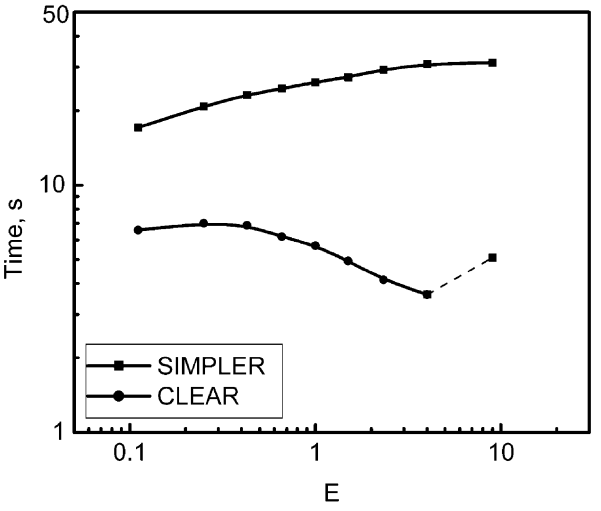


(c) Stream functions and isothermals of [34]

Figure 12. Predicted isothermals and stream functions for $Ra = 10^4$.



(a) Iteration number



(b) CPU time

Figure 13. Comparison of iteration number and CPU time for $\mathrm{Ra} = 10^4$ of Example 4.

$$L_x/D_{\mathrm{in}} = 30 \qquad L_{\mathrm{in}}/D_{\mathrm{in}} = 5 \qquad D_{\mathrm{out}}/D_{\mathrm{in}} = 2 \tag{70}$$

The inlet velocity distribution is supposed to be fully developed, which satisfies

$$u = u_{\max}\left(1 - \frac{r^2}{R_{\mathrm{in}}^2}\right) \qquad R_{\mathrm{in}} = \frac{D_{\mathrm{in}}}{2} \qquad u_{\max} = 2u_{\mathrm{mean}} \tag{71}$$

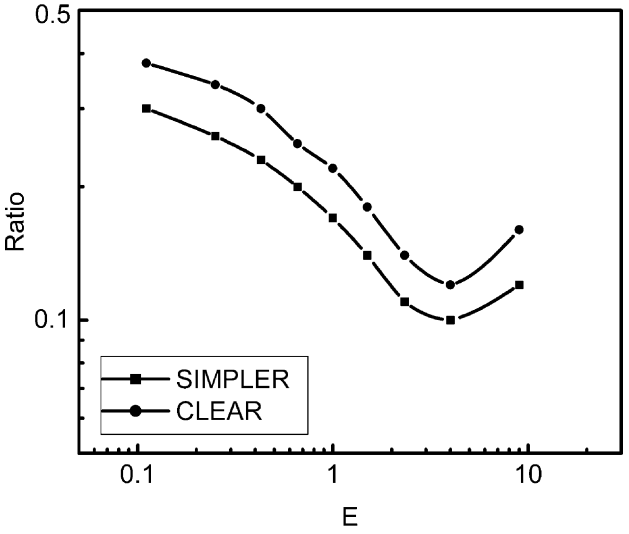


Figure 14. Ratios of iteration numbers and CPU time of Example 4.

The outflow boundary condition is assumed to be fully developed. The result obtained by Macagno and Hung [36] by both experimental and numerical methods is the ratio of the reattachment length over inlet diameter, L_R/D_{in}, as 8.8 for $\text{Re} = 200$. In the present study, we adopt a uniform 202×42 grid ($\varepsilon = 1.0 \times 10^{-6}$). The pressure-equation underrelaxation factor is 0.95 for the low convergent rate of the case. The present predicted ratio of L_R/D_{in} from the two algorithms is identical (8.5) for the predicted Re number. The iteration number and CPU time of the two algorithms are shown in Figure 16. The ratio of iteration number and CPU time are presented in Figure 17. The variation range of the iteration number ratio is from 0.07 to 0.76, and from 0.1 to 0.96 for CPU time. The maximum saving in CPU time is up to 90%.

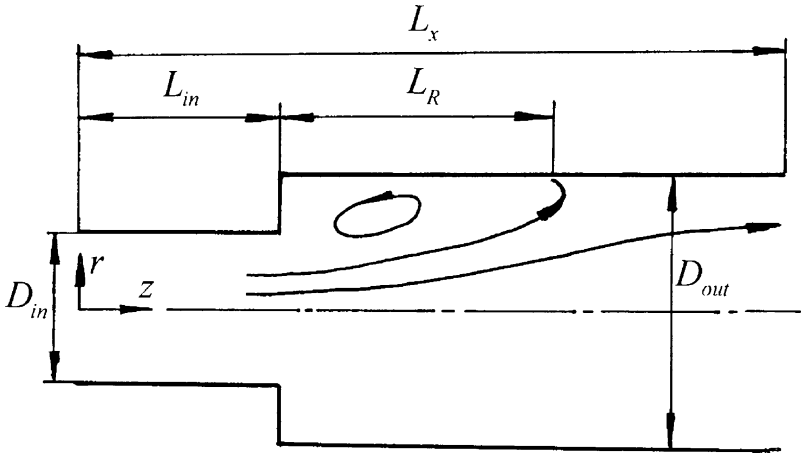


Figure 15. Laminar flow over an annular backward step.

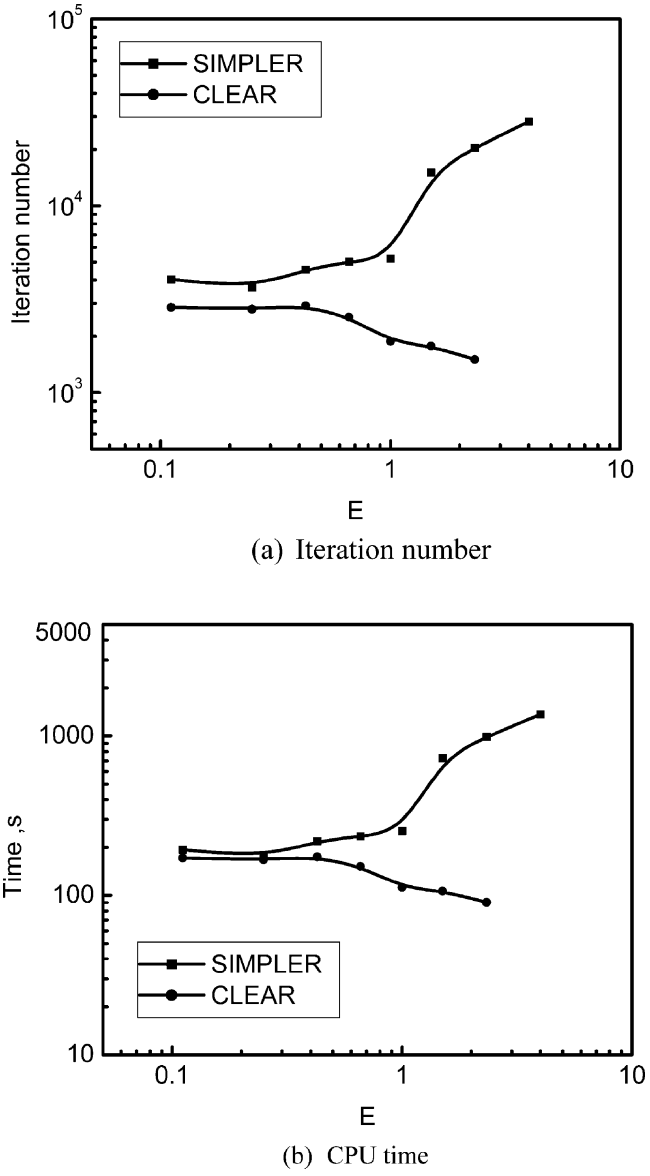


Figure 16. Comparison of iteration number and CPU time for Re = 200 of Example 5.

Example 6: Laminar fluid flow over a rectangular backward step. The problem is presented schematically in Figure 18. The fluid flow for Re = 300 is simulated numerically. The geometry parameters are taken from Kondoh et al. [37], and include

$$\frac{H_2}{H_1} = 2 \qquad \frac{L_1}{H_1} = 5 \qquad \frac{L_2}{H_1} = 30 \tag{72}$$

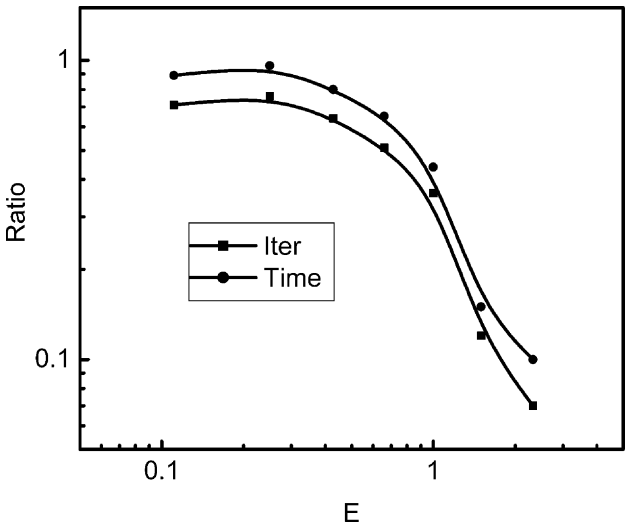


Figure 17. Ratios of iteration numbers and CPU time of Example 5.

The inlet velocity distribution is fully developed in dimensionless form:

$$X = 0 \quad 1 < Y < \frac{H_1 + H_2}{H_1} \quad U = 1.5\left\{1 - \left[\frac{Y - 0.5\left(\frac{H_2}{H_1}\right) - 1}{0.5\left(\frac{H_2}{H_1}\right)}\right]\right\}^2 \quad V = 0 \qquad (73)$$

At the outflow boundary, fully developed condition is assumed. The Reynolds number is defined as $\mathrm{Re} = u_{\mathrm{mean}} H_1/\nu$. In the present study, the grid number is

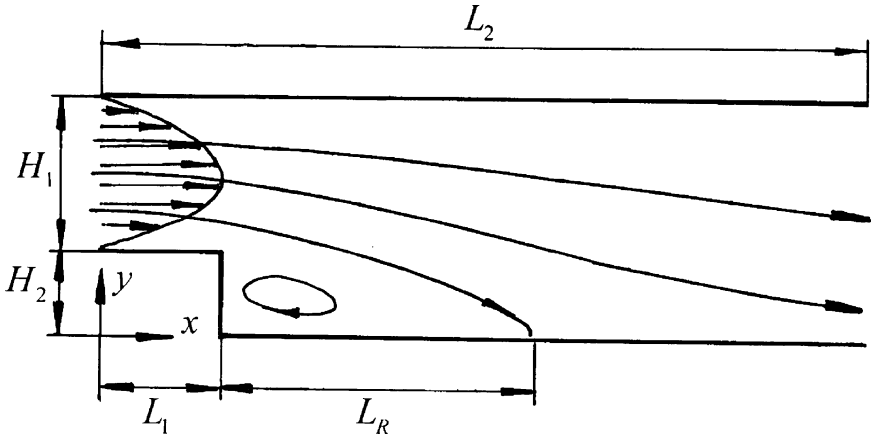
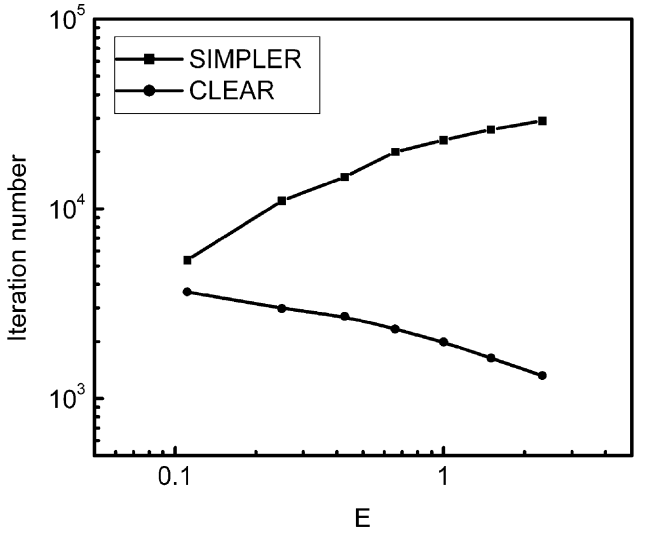
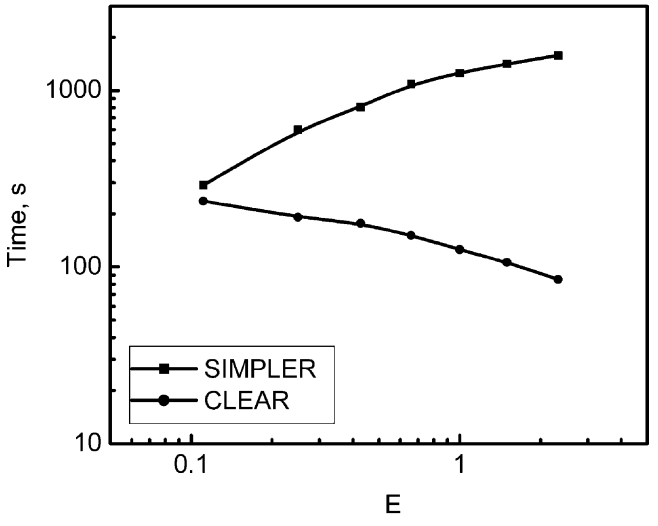


Figure 18. Flow over a rectangular backward step.



(a) Iteration number



(b) CPU time

Figure 19. Comparison of iteration number and CPU time for Re = 300 of Example 6.

162×62 with $\varepsilon = 1.0 \times 10^{-6}$. In the solid region of $0 < X < L_1/H_1$, $0 < Y < 1$, the domain extension method is used to deal with solid parts. The compared results shown in Figures 19 and 20. Again we can see that CLEAR perform better than SIMPLER. The ratios of iteration number and CPU time are from 0.045 to 0.68 and from 0.054 to 0.81, respectively.

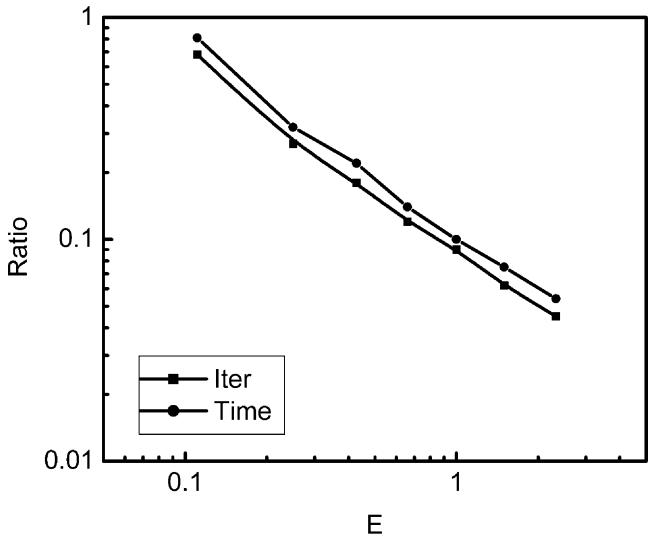


Figure 20. Ratios of iteration numbers and CPU time of Example 6.

CONCLUSION

In this article, the implementation of fully implicit algorithm CLEAR is proposed on the collocated grid system. Six typical numerical examples with available solutions are applied to compare the CLEAR and SIMPLER algorithms. It is revealed that the CLEAR algorithm possesses good convergent performance on collocated grids with similar robustness compared to the SIMPLER algorithm for various applications, showing the feasibility of the new algorithm.

For conjugated problems solved by the domain extension method, special care must be taken on the collocated gird to guarantee the zero velocity in the solid region and at the boundary of the solid part.

REFERENCES

1. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, vol. 15, p. 1787, 1972.
2. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
3. J. P. van Doormaal and G. D. Raithby, Enhancement of SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
4. J. P. van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flow, ASME Paper 85-HT-9, 1985.
5. G. D. Raithby and G. E. Schneider, 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*, pp. 241–289, Wiley, New York, 1988.
6. R. I. Issa, Solution of Implicitly Discretized Fluid Flow Equation by Operator-Splitting, *J. Comput. Phys.*, vol. 62, pp. 40–65, 1985.

7. R. H. Yen and C. H. Liu, Enhancement of the SIMPER Algorithm by an Additional Explicit Corrector Step, *Numer. Heat Transfer B*, vol. 24, pp. 127–141, 1993.
8. Y. Sheng, M. Shoukri, G. Sheng, and P. Wood, A Modification to the SIMPLE Method for Buoyancy-Driven Flows, *Numer. Heat Transfer B*, vol. 33, pp. 65–78, 1998.
9. B. Yu, H. Ozoe, and W. Q. Tao, A Modified Pressure-Correction Scheme for the SIMPLER Method, MSIMPLER, *Numer. Heat Transfer B*, vol. 39, pp. 439–449, 2001.
10. F. Moukalled and M. Darwish, A Unified Formulation of the Segregated Class of Algorithm for Fluid Flow at All Speeds, *Numer. Heat Transfer B*, vol. 37, pp. 103–139, 2000.
11. E. Blosch and W. Shyy, The Role of Mass Conservation in Pressure-Based Algorithms, *Numer. Heat Transfer B*, vol. 24, pp. 415–429, 1993.
12. S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington, DC, 1980.
13. W. Shyy and R. Mittal, Solution Methods for the Incompressible Navier-Stokes Equations, in R. W. Johnson (ed.), *Handbook of Fluid Dynamics*, pp. 31.1–31.33, CRC Press, Boca Raton, FL, 1998.
14. W. Q. Tao, Z. G. Qu, and Y. L. He, A Novel Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems—CLEAR (Coupled & Linked Equations Algorithm Revised), Part I: Mathematical Formulation and Solution Procedure, *Numer. Heat Transfer B*, vol. 45, pp. 1–17, 2004.
15. W. Q. Tao, Z. G. Qu, and Y. L. He, A Novel Segregated Algorithm for Incompressible Fluid Flow and Hear Transfer Problems—CLEAR (Coupled & Linked Equations Algorithm Revised), Part II: Application Examples, *Numer. Heat Transfer B*, vol. 45, pp. 19–48, 2004.
16. W. Q. Tao, *Numerical Heat Transfer*, 2nd ed., Xi'an Jiaotong University Press, Xi'an, China, 2001.
17. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trading Edge Separations, *AIAA J.*, vol. 21, no. 11, pp. 1525–1535, 1983.
18. M. Peric, A Finite Volume Method for the Predictions of Three-Dimensional Fluid Flow in Complex Ducts, Ph.D. thesis, University of London, London, UK, 1985.
19. S. Majumdar, Development of a Finite-Volume Procedure for Prediction Fluid Flow Problems with Complex Irregular Boundaries, Rep. 210/T/29, SFB210, University of Karlsrule, Karlsrule, Germany, 1986.
20. S. Majumdar, Roles of Under-relaxation in Momentum Interpolation for Calculation of Flow with Non-staggered Grids, *Numer. Heat Transfer*, vol. 13, pp. 125–132, 1988.
21. T. F. Miller and F. W. Schmidt, Use of a Pressure-Weighted Interpolation Method for the Solution of Incompressible Navier-Stokes Equations on a Non-staggered Grid System, *Numer. Heat Transfer*, vol. 14, pp. 213–233, 1988.
22. M. H. Kobayashi and J. C. F. Pereira, Numerical Comparison of Momentum Interpolation Methods and Pressure-Velocity Algorithm Using Nonstaggered Grids, *Commun. Appl. Numer. Meth.*, vol. 7, pp. 173–196, 1991.
23. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
24. S. K. Choi, Note on the Use of Momentum Interpolation Method for Unsteady Flows, *Numer. Heat Transfer A*, vol. 36, pp. 545–550, 1999.
25. Bo Yu, Wen-Quan Tao, Jin-Jia Wei, Y. Kawaguchi, T. Tagawa, and H. Ozoe, Discussion on Momentum Interpolation Method for Collocated Grids of Incompressible Flow, *Numer. Heat Transfer B*, vol. 42, pp. 141–166, 2002.
26. C. J. Chen, R. Bernatz, D. Carlson, and W. L. Lin, *Finite Analytic Method in Flows and Heat Transfer*, p. 219, Taylor & Francis, New York, 2000.

27. W. Shyy, S. S. Thakur, H. Ouyang, J. Liu, and E. Blosch. *Computational Techniques for Complex Transport Phenomena*, p. 38, Cambridge University Press, Cambridge, UK, 1997.
28. H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, pp. 135, 150, Longman Scientific & Technical, Harlow, UK, 1995.
29. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, p. 166, Springer-Verlag, Berlin, 1996.
30. T. F. Miller, Fourier Analysis of the SIMPLE Algorithm Formulation on a Collocated Grid, *Numer. Heat Transfer B*, vol. 30, pp. 45–66, 1996.
31. U. Ghie, K. N. Ghie, and C. T. Shin, High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. Comput. Phys.*, vol. 48, pp. 387–411, 1982.
32. L. Fuchs and N. Tillmark, Numerical and Experimental Study of Driven Flow in a Polar Cavity, *Int. J. Numer. Meth. Fluids*, vol. 5, pp. 311–329, 1985.
33. G. Barakos and E. Mitsoulis, Natural Convection Flow in a Square Cavity Revisited: Laminar and Turbulent Methods with Wall Functions, *Int. J. Numer. Meth. Fluids*, vol. 18, no. 7, pp. 695–719, 1994.
34. T. H. Kuehn and R. J. Goldstein, An Experimental and Theoretical Study of Natural Convection in the Annulus between Horizontal Concentric Cylinders, *J. Fluid Mech.*, vol. 74, pp. 695–715, 1969.
35. M. Yang, W. Q. Tao, and Q. J. Wu, Numerical Study of Natural Convection Heat Transfer in a Cylindrical Envelope with Internal Concentric Slotted Hollow Cylinder, *Numer. Heat Transfer A*, vol. 22, pp. 289–305, 1992.
36. E. O. Macagno and T. K. Hung, Computation and Experimental Study of Captive Annular Eddy, *J. Fluid Mech.*, vol. 28, pp. 43–64, 1967.
37. T. Kondoh, Y. Nagano, and T. Tsuji, Computational Study of Laminar Heat Transfer Downstream of a Backward-Facing Step, *Int. J. Heat Mass Transfer*, vol. 36, no. 3, pp. 577–591, 1993.