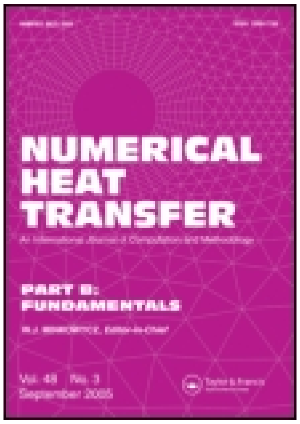


Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/unhb20

ENHANCEMENT OF THE SIMPLE ALGORITHM BY AN ADDITIONAL EXPLICIT CORRECTOR STEP

Ruey-Hor Yen [a] & Chen-Hua Liu [a]
[a] Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan, 10764
Version of record first published: 07 May 2007.

Numerical Heat Transfer, Part B, vol. 24, pp. 127–141, 1993

ENHANCEMENT OF THE SIMPLE ALGORITHM BY AN ADDITIONAL EXPLICIT CORRECTOR STEP

Ruey-Hor Yen and Chen-Hua Liu
Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10764

Due to an assumption made on the pressure–velocity coupling for the SIMPLE algorithm and its variants, the corrected velocity can be obtained from the corrected pressure. However, substituting these quantities into the momentum equations may result in failure to satisfy the momentum equations. Therefore, the equations should be solved iteratively to obtain better velocities, thus giving a more satisfactory solution to the equations. In this article an explicit corrector step is proposed that is imposed on the first corrected velocities, which are obtained from the existing algorithms. This new corrector step has been tested by three flow problems, driven cavity flow, backward-facing step flow, and rectangular tank flow, with different Reynolds numbers. With this additional corrector step imposed on the SIMPLEC and PISO algorithms, the results show that the number of iterations can be reduced drastically due to the much better satisfaction of the momentum equations. Considerable savings in computing effort can be gained.

INTRODUCTION

The tremendous improvement in computer capabilities in the last decade, in memory and speed, has enabled accurate numerical predictions of complex fluid flow problems. However, despite many recent advances in computational fluid mechanics, accurate calculation of practical three-dimensional flows remains a difficult task. Therefore, one of the main goals of research in computational fluid dynamics is to provide reliable, accurate, and economical solutions to the governing equations formulated to represent an industrial flow process.

There are two important factors that determine the accuracy and efficiency of the numerical methods employed. One is the discretization technique; the other is the solution algorithm. The accuracy of the solution is dependent only on the discretization technique; the efficiency, however, is dictated by both of them, since the nature of the discretization equation may control how efficiently they can be solved and the solution algorithm determines the solution procedure. In this article the solution algorithm is investigated. The aim of the present article is to improve the SIMPLE algorithm and its variants.

One of the widely used algorithms for fluid flow problems is the SIMPLE method of Patankar and Spalding [1], which was introduced in 1972. This method and its variants deal primarily with the pressure–velocity coupling of the Navier-Stokes equations. For practical problems, the algebraic equation set is too large for efficient direct solution; an iterative approach is normally adopted that segregates the dependent variables, the governing equations being solved in a sequential manner.

This work was supported by a grant from the National Science Council of the Republic of China.

NOMENCLATURE

$\mathbf{A}$	coefficient matrix in the finite-difference equation, area of a control volume face
b	coefficient in the finite-difference equation
$\mathbf{C}$	coefficient matrix in the finite-difference equation
E	time step multiple
$\mathbf{M}$	coefficient matrix in the finite-difference equation
p	pressure
$\mathbf{p}$	pressure vector
$\mathbf{Q}$	matrix operator in the pressure-correction equation
u, v	velocity components in the x and y directions
$\mathbf{u}, \mathbf{v}$	velocity vectors
$\mathbf{u}_E, \mathbf{v}_E$	velocities obtained by the explicit step
α	relaxation factor
γ_p	residual reduction factor of the pressure-correction equation
ρ	density

Subscripts

e, w, n, s	control-volume boundaries
E, W, N, S	neighboring nodes
nb	neighbor
P	central node
P	central node

Superscripts

(k)	number of time steps or number of iterations
u, v	matrix coefficients of u and v velocity components
$'$	correction quantities
*	quantities in the predictor stage
**	quantities in the first corrector stage
***	quantities in the second corrector stage

The momentum conservation equations reveal that the pressure and velocity are strongly coupled. Many investigators have paid much attention to them in order to improve the computational efficiency. A general pressure-correction method was proposed by Connel and Stow [2]. Two extended pressure-correction techniques using more accurate approximations to deal with the pressure–velocity coupling were derived by them. Van Doormaal and Raithby [3] also presented a similar generalized approach.

There are two kinds of predictor-corrector iteration schemes commonly employed to solve the equations. The first category of methods, which uses a single corrector step, may encounter a slowly converging problem due to the inappropriate treatment of the pressure and velocity coupling in the pressure-correction equation, requiring a heavy underrelaxation. The original SIMPLE method operates in such a manner. A very efficient algorithm, called SIMPLEC [4], has been proposed to improve the pressure–velocity coupling by incorporating a more accurate treatment of neighboring points. In the SIMPLEC algorithm the constant correction is applied to all neighboring points.

In order to remove the assumptions involved in the pressure–velocity coupling, the second category of iteration schemes has been developed. These are two-corrector schemes. Usually, in the first corrector stage, exactly the same step as in the single-corrector approach is followed. The second stage comprises the solution of another pressure-corrector equation. This yields a pressure field that is much closer to the final solution than that obtained by the single-corrector scheme, hence resulting in a substantial improvement in the stability and rate of convergence of the iteration process. SIMPLER [5], FIMOSE [6], and PISO [7] belong to this category. In the SIMPLER and PISO algorithms, iterative or predictive-corrector techniques are used to incorporate the influence of the neighboring points while retaining the tridiagonal nature of the equations in each direction. FIMOSE handles the pressure–velocity coupling in a slightly different manner from the SIMPLE methods. The equations in the second corrector stage are solved by the ADI solution routine for the PISO and FIMOSE methods.

Due to an assumption made on the pressure–velocity coupling, the corrected velocity can thus be obtained from the corrected pressure. However, substituting these quantities into the momentum equations can result in failure to satisfy the momentum equations. Therefore, the momentum equations should be solved again to obtain better velocities, thus yielding a more satisfactory solution to the equations. In this article an additional corrector step is proposed that is imposed on the first corrected velocities, which are obtained from the existing algorithms. It is an explicit correction step by which considerable savings in computing effort can be gained. This additional correction step applying to existing efficient algorithms, e.g., SIMPLEC and PISO, is tested.

METHOD

For the sake of simplicity, the method is illustrated for a two-dimensional problem; its extension to three-dimensional problems is straightforward. A staggered grid is used for the discretization of equations. The conservation equations for momentum u and v are expressed in finite-difference form and expressed as

$$a_e u_e = \sum a_{nb} u_{nb} + A_e(p_P - p_E) + b^u \tag{1}$$

$$a_n v_n = \sum a_{nb} v_{nb} + A_n(p_P - p_N) + b^v \tag{2}$$

The continuity equation is expressed as

$$(\rho u A)_e - (\rho u A)_w + (\rho u A)_n - (\rho u A)_s = 0 \tag{3}$$

To solve Eqs. (1) and (2), Patankar [8] introduces an iteration procedure with an added underrelaxation factor. This is similar to the time-approach procedure in which the transient term is retained in Eqs. (1) and (2). The resulting equations are

$$a_e\left(1 + \frac{1}{E}\right)u_e^{(k)} = \sum a_{nb} u_{nb}^{(k)} + A_e(p_P^{(k)} - p_E^{(k)}) + b^u + \frac{a_e}{E}u_e^{(k-1)} \tag{4}$$

$$a_n\left(1 + \frac{1}{E}\right)v_n^{(k)} = \sum a_{nb} v_{nb}^{(k)} + A_n(p_P^{(k)} - p_E^{(k)}) + b^v + \frac{a_n}{E}v_n^{(k-1)} \tag{5}$$

where E is called the time step multiple and the superscript (k) denotes the number of time steps or iterations performed. This has been discussed in detail by Van Doormaal and Raithby [4]. The influence of the E factor on the computational performance of the proposed method is studied in this article.

Equations (1) through (3) can be written in matrix notation as

$$\mathbf{A}_e^u \mathbf{u} = \mathbf{A}_{nb}^u \mathbf{u} + \mathbf{C}^u \mathbf{p} + \mathbf{b}^u \tag{6}$$

$$\mathbf{A}_n^v \mathbf{v} = \mathbf{A}_{nb}^v \mathbf{v} + \mathbf{C}^v \mathbf{p} + \mathbf{b}^v \tag{7}$$

$$\mathbf{M}^u \mathbf{u} + \mathbf{M}^v \mathbf{v} = 0 \tag{8}$$

A predictor-corrector iteration proposed by Patankar and Spalding [1] is employed. The predicted quantities are indicated by asterisks and expressed as

$$\mathbf{A}_e^u \mathbf{u}^* = \mathbf{A}_{nb}^u \mathbf{u}^* + \mathbf{C}^u \mathbf{p}^* + \mathbf{b}^u \tag{9}$$

$$\mathbf{A}_n^v \mathbf{v}^* = \mathbf{A}_{nb}^v \mathbf{v}^* + \mathbf{C}^v \mathbf{p}^* + \mathbf{b}^v \tag{10}$$

$$\mathbf{M}^u \mathbf{u}^* + \mathbf{M}^v \mathbf{v}^* = 0 \tag{11}$$

Given an estimated pressure field, denoted by $\mathbf{p}^*$, the corresponding predicted velocities, $\mathbf{u}^*$ and $\mathbf{v}^*$, are obtained for the momentum equations (9) and (10). However, the velocities so obtained may not satisfy the continuity equation. The pressure can be used to effect satisfaction of the both equations through the relation

$$\mathbf{p} = \mathbf{p}^* + \mathbf{p}' \tag{12}$$

where $\mathbf{p}'$ is called the pressure correction. The velocities react to the change in pressure gradients and are corrected in a similar manner:

$$\mathbf{u} = \mathbf{u}^* + \mathbf{u}' \tag{13}$$

$$\mathbf{v} = \mathbf{v}^* + \mathbf{v}' \tag{14}$$

The relation between the pressure corrections and the velocity corrections can be developed by substracting Eq. (9) from Eq. (6) to give

$$\mathbf{A}_e^u \mathbf{u}' = \mathbf{A}_{nb}^u \mathbf{u}' + \mathbf{C}^u \mathbf{p}' \tag{15}$$

$$\mathbf{A}_n^v \mathbf{v}' = \mathbf{A}_{nb}^v \mathbf{v}' + \mathbf{C}^v \mathbf{p}' \tag{16}$$

The velocity correction can be obtained from Eqs. (15) and (16):

$$\mathbf{u}' = (\mathbf{A}_e^u - \mathbf{A}_{nb}^u)^{-1} \mathbf{C}^u \mathbf{p}' \tag{17}$$

$$\mathbf{v}' = (\mathbf{A}_n^v - \mathbf{A}_{nb}^v)^{-1} \mathbf{C}^v \mathbf{p}' \tag{18}$$

Substituting Eqs. (17) and (18) into the continuity equation (11) yields

$$\mathbf{Q}\mathbf{p}' = -(\mathbf{M}^u \mathbf{u}^* + \mathbf{M}^v \mathbf{v}^*) \tag{19}$$

where

$$\mathbf{Q} = \mathbf{M}^u(\mathbf{A}^u_e - \mathbf{A}^u_{nb})^{-1} + \mathbf{M}^v(\mathbf{A}^v_n - \mathbf{A}^v_{nb})^{-1} \tag{20}$$

Equation (19) is called the generalized pressure-correction equation. To obtain a corrected pressure for the next iteration, this equation must be solved. Detailed examination of this equation reveals that the operator, $\mathbf{Q}$, contains two matrix inversions, requiring much time to compute. Therefore, various approximations dealing with the matrix inverse have been made to bring about an efficient computational solution. Following is a review of approximations by the SIMPLE, SIMPLEC, and PISO algorithms.

SIMPLE

In the SIMPLE algorithm the velocity correction at a point is not affected by the velocity correction of its neighbors, and thus the inverse of matrix in the operator $\mathbf{Q}$ is reduced to the inverse of $\mathbf{A}^u_e$. Since $\mathbf{A}^u_e$ is a diagonal matrix, finding the inverse of it takes little time. However, this assumption results in the continuity and momentum equations not being satisfied. Thus a heavy relaxation factor has to be used in order to obtain a solution with a better convergence rate.

SIMPLEC

The SIMPLEC algorithm attempts to use a more consistent approximation based on the magnitude of the terms in the velocity-correction equation. This means that the pentadiagonal matrices of $\mathbf{A}^u_{nb}$ and $\mathbf{A}^v_{nb}$ in the operator $\mathbf{Q}$ are approximated by a diagonal matrix whose element is the sum of the corresponding neighboring velocity coefficients. Therefore the inverse of the matrix in the operator $\mathbf{Q}$ becomes an inverse of the diagonal matrix, and this results in a very efficient algorithm. Although the SIMPLEC algorithm satisfies the momentum and continuity equations more adequately than SIMPLE, total satisfaction is not yet reached since the approximation still applies to this algorithm. Nevertheless, the better satisfaction for the SIMPLEC algorithm results in a faster convergence rate than SIMPLE.

PISO

The first pressure-correction equation of PISO is identical to that used by SIMPLE. A second corrector step is done to ensure that the continuity and momentum equations are more satisfied at the end of each iteration than without it. Since the velocity and pressure predicted by SIMPLE cannot satisfy the momentum and continuity equations, as mentioned above, the second corrector step takes full responsibility for doing so. It can be seen that if the first step can satisfy the equation set more successfully, the burden of satisfaction for the second step can be reduced. This means that one can arrive at convergence faster.

Additional Corrector Step

From the above discussion, it is seen that the better the satisfaction of the continuity and momentum equations during iteration, the faster is the convergence. A novel corrector step is proposed based on this general principle. The velocities and pressure are obtained from the predicted step at the kth iteration. They are denoted by $u^{*(k)}, v^{*(k)}$, and $p^{*(k)}$. Using these quantities, the corrected velocities and pressure are thus obtained from the first correction step by either the SIMPLE or the SIMPLEC algorithm. They are denoted by $u^{**(k)}, v^{**(k)}$, and $p^{**(k)}$. Due to the approximation made on the pressure–velocity coupling equation (19), as mentioned previously, substituting these quantities into the momentum equations can result in failure to satisfy the momentum equation. Therefore, the momentum equations have to be solved iteratively to achieve better satisfaction. In this article an explicit procedure is proposed as follows, that is, a point-by-point explicit update using the error vector that results from not satisfying the original equation set is adopted.

$$\mathbf{A}_{\mathrm{e}}^{u}\mathbf{u}_{E}^{(k)} = \mathbf{A}_{\mathrm{nb}}^{u}\mathbf{u}^{**(k)} + \mathbf{C}^{u}\mathbf{p}^{**(k)} + \mathbf{b}^{u} \tag{21}$$

$$\mathbf{A}_{\mathrm{n}}^{v}\mathbf{v}_{E}^{(k)} = \mathbf{A}_{\mathrm{nb}}^{v}\mathbf{v}^{**(k)} + \mathbf{C}^{v}\mathbf{p}^{**(k)} + \mathbf{b}^{v} \tag{22}$$

where the subscript E denotes the explicit procedure. The explicit procedure is used to keep the algorithm much more efficient in obtaining an approximate velocity distribution, rather than using other procedures such as the ADI.

The final corrected velocities and pressure for the next $(k+1)$th iteration are specified by

$$\mathbf{u}^{***(k)} = \mathbf{u}^{**(k)} + \alpha(\mathbf{u}_{E}^{(k)} - \mathbf{u}^{**(k)}) \tag{23}$$

$$\mathbf{v}^{***(k)} = \mathbf{v}^{**(k)} + \alpha(\mathbf{v}_{E}^{(k)} - \mathbf{v}^{**(k)}) \tag{24}$$

$$\mathbf{p}^{***(k)} = \mathbf{p}^{**(k)} \tag{25}$$

where α is the relaxation factor used to accelerate the iterative procedure. It is not necessary to assign the value of a relaxation factor. However, there does exist an optimal value of α, to be discussed later in this article.

RESULTS AND DISCUSSION

For the purpose of demonstrating the applicability of the additional explicit step set forth, the SIMPLEC and PISO algorithms with and without the additional explicit correction step were tested to evaluate their relative performances. Three fluid flow problems were selected in the present work, namely, driven cavity flow, backward-facing step flow, and rectangular tank flow, as illustrated in Figs. 1a–1c, respectively. All the calculations were carried out on an IBM PC 486 with MS FORTRAN compiler V.5. All the numerical results were obtained with the same basic code and the criteria for terminating iteration. The power-law discretization scheme is employed in this article. The convergence criterion for the pressure-correction equation is the one proposed by Van Doormaal and Raithby [4]. This

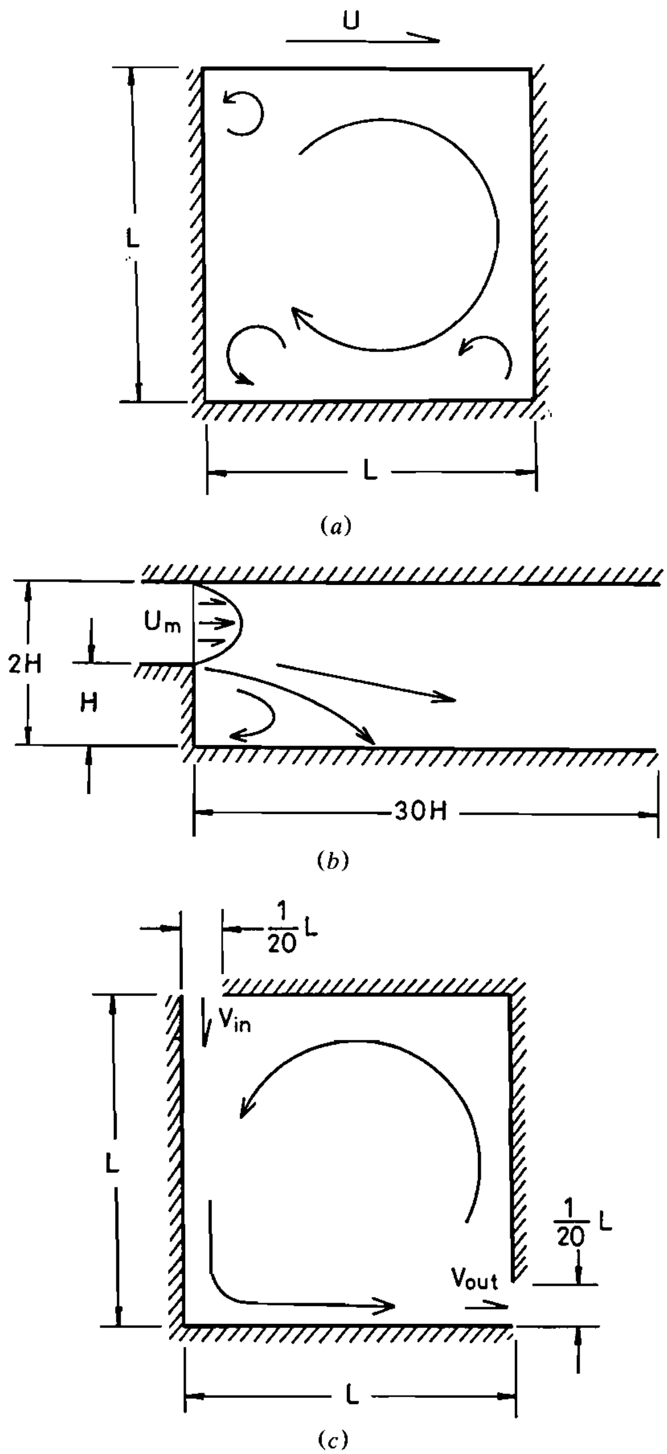


Fig. 1 Geometries of the three test problems.

criterion guarantees that iteration has reduced the residual to at least a fraction γ_p of its initial value. In the present work the value of the residual reduction factor γ_p is 0.05. The convergence criterion for the velocity variables is set as the summation of the absolute values of the differences between the two subsequent iterations over all the grid points, less than 1×10^{-6}. The mass residual convergence criterion is also 1×10^{-6}.

The computational performance for each of the methods in satisfying the above requirement is dependent on the time step multiple E and the relaxation factor α for the explicit corrector step and Reynolds number. The computational performance is defined as the difference of CPU times with and without an explicit corrector step relative to the CPU time with an explicit step. The iteration number is also shown against the time step multiple so that the CPU time savings due to the explicit corrector step can be realized.

Relaxation Factor

Figure 2 shows the optimal relaxation factor for the explicit corrector step for driven cavity flow with a Reynolds number of 100 using the SIMPLEC algorithm. The optimal relaxation factor is that obtained by choosing the relaxation factor with best computational performance for each time step multiple. The value of the optimal relaxation factor increases with an increase of the time step multiple. As shown in Fig. 2, this relationship can be approximated by the straight line $\alpha = 1.0 + 0.1E$. Figure 3 shows the computational performance for the three cases using the optimal relaxation factor, using an approximated relaxation factor, and without using a relaxation factor. It can be seen that even without using the relaxation factor, that is, where $\alpha = 1.0$, the explicit corrector step is always much better than the original SIMPLEC algorithm for all time step multiples. After investigating the optimal relaxation factor for the different flow geometries with

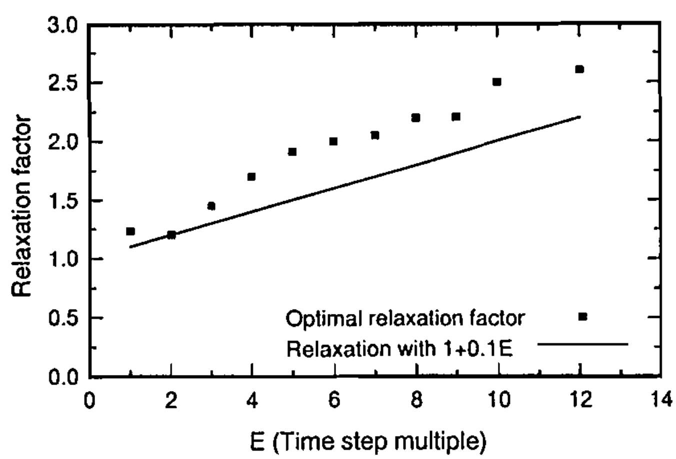


Fig. 2 Optimal relaxation factor and its approximation using SIMPLEC (driven cavity flow, Re = 100, 30 × 30).

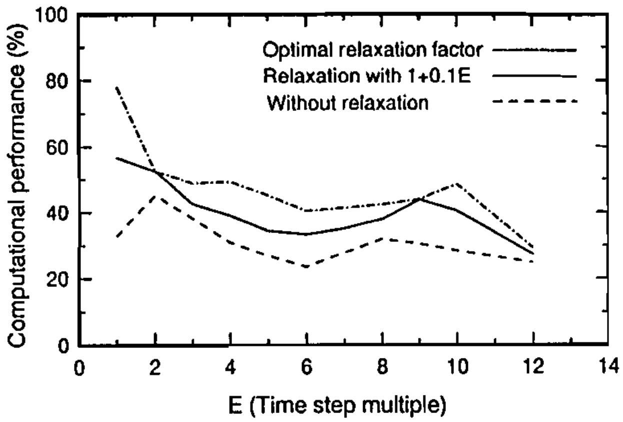


Fig. 3 Comparison of the computational performance using SIMPLEC with different relaxation factors (driven cavity flow, Re = 100, 30 × 30).

different Reynolds number in this article, an approximate relationship between the optimal relaxation factor and the time step multiple can be found to be the same straight line as mentioned above for the SIMPLEC algorithm. Therefore, the approximate optimal relaxation factor is used for all the calculations for the SIMPLEC algorithm. However, for the PISO algorithm, a simple relationship cannot be obtained. Moreover, a slow convergence or even divergence may occur for some values of the relaxation factor. Therefore, computation in this article by the PISO algorithm is implemented by the relaxation factor $\alpha = 1$.

Driven Cavity Flow

The first test problem, as shown in Fig. 1*a*, is the driven cavity problem with Reynolds numbers of 100, 400, and 1000 for grid points 30 × 30. The accuracy of the numerical results has been verified by reproducing the results of Burggraf [9].

Figure 4*a* shows the CPU time required for the SIMPLE, SIMPLEC, and PISO algorithms. The effect of the additional explicit corrector step is also included. In the figure, MSIMPLEC and MPISO denote the curves obtained by the corresponding SIMPLEC and PISO algorithms with the explicit corrector step. Figure 4*b* shows the total number of iterations required for the algorithms. It can be seen that the number of iterations decreases significantly after the additional corrector step is imposed. This is due to better satisfaction of the equations. Figure 4*c* shows that the algorithm with the additional explicit step performs better than the original algorithms. The effect of the number of grid points on the computational performance was also investigated. Figure 5 shows the computer performance for the 43 × 43 grid size, for which the grid points are twice as large as that of 30 × 30. It can be seen that the computational performance is dependent on the grid size. The performance with the explicit corrector step may give tremendous improvement over the original algorithms. Figure 6 shows that better computa-

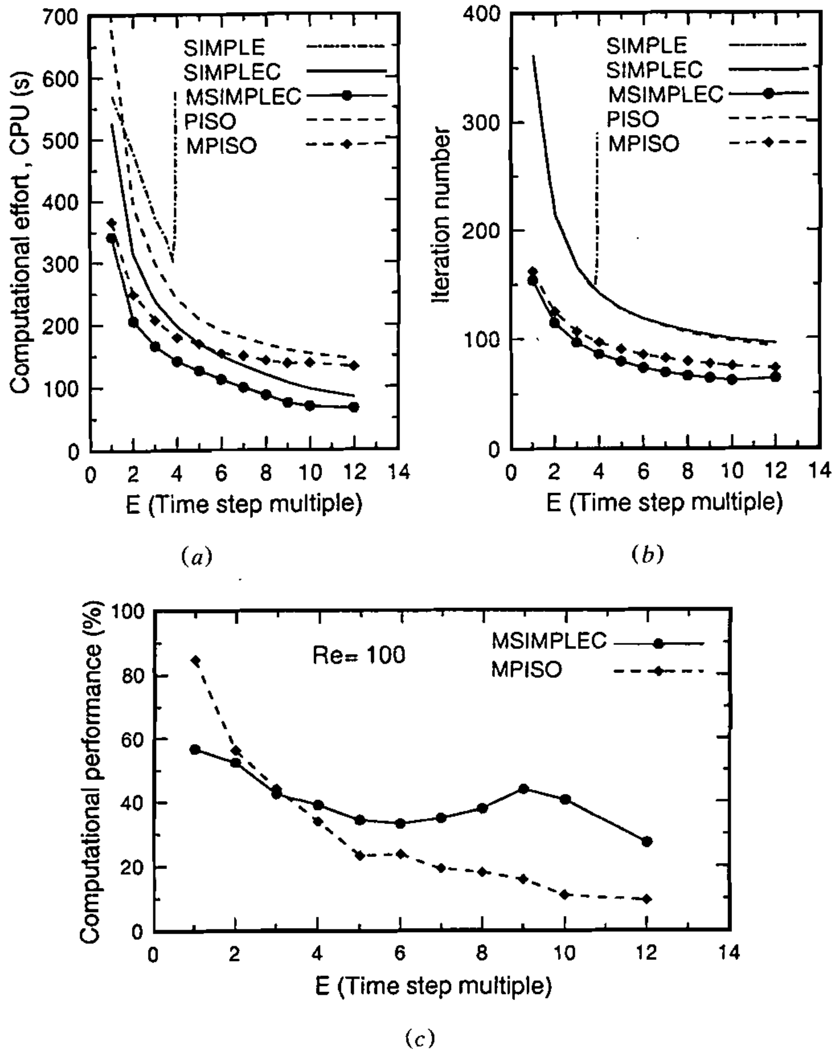


Fig. 4 Computational effort, number of iterations, and computational performance using different algorithms (driven cavity flow, Re = 100, 30 × 30).

tional performance can also be obtained for different Reynolds numbers when the additional explicit step applies to SIMPLEC and PISO.

Backward-Facing Step

The second test problem, as illustrated in Fig. 1*b*, is the backward-facing step problem having Reynolds numbers of 100, 400, and 1000. The results have been verified by comparing the existing data for a Reynolds number of 389, which was obtained numerically by Armaly et al. [10]. Figure 7*a* shows the computational effort, Fig. 7*b* shows the number of iterations, and Fig. 7*c* gives the computational

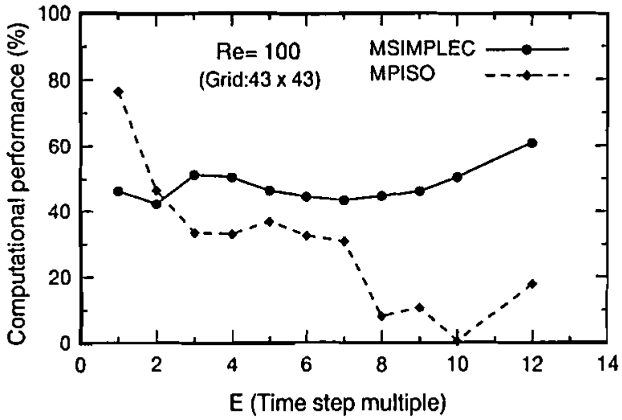
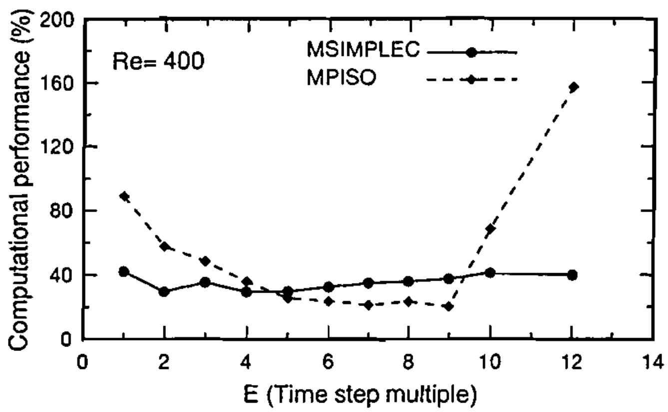
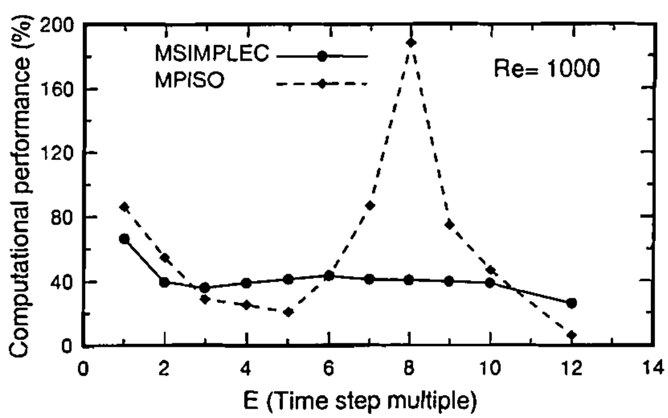


Fig. 5 Computational performance for grid size of 43 × 43 (driven cavity flow, Re = 100).



(*a*)



(*b*)

Fig. 6 Computational performance for Re = 400 and Re = 1000 (driven cavity flow, 30 × 30).

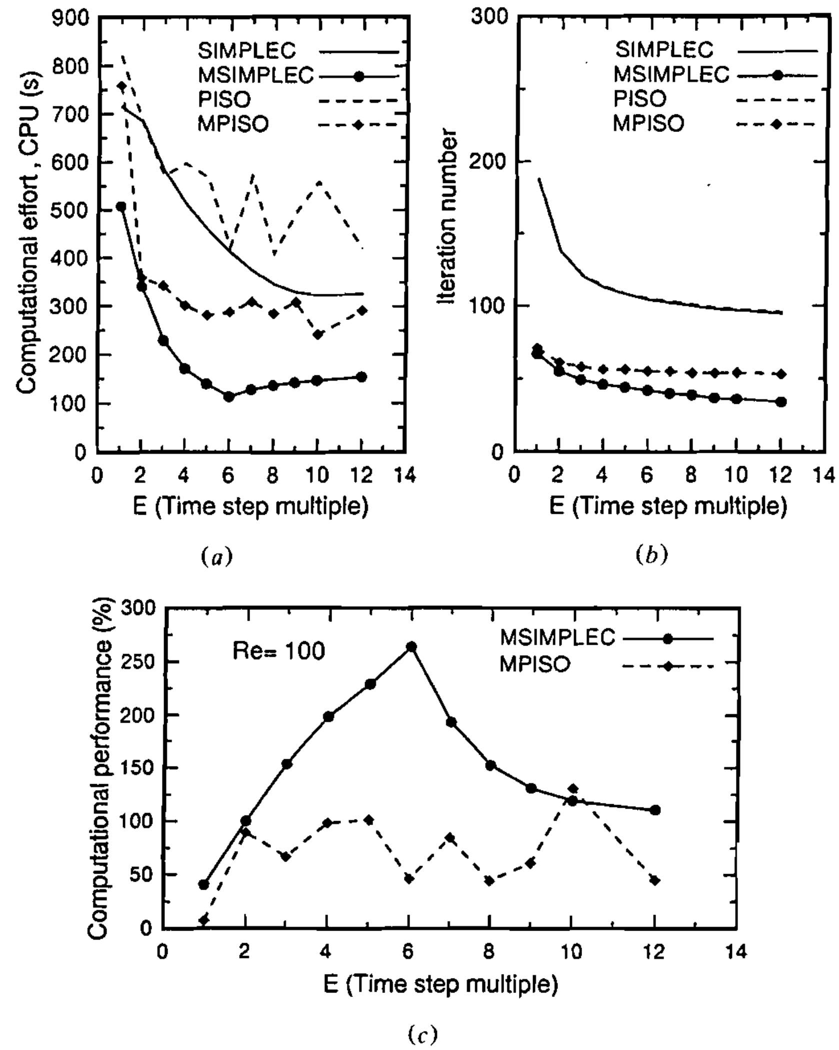


Fig. 7 Computational effort, number of iterations, and computational performance using different algorithms (backward-facing step flow, Re = 100).

performance for SIMPLEC and PISO with and without the additional corrector step. It appears that results similar to the driven cavity flow problem can be obtained as discussed above. It should be noted that the computational performance for this flow problem is even better than for the previous flow problem. The oscillation of performance with time step multiple for PISO is due to the nature of PISO in the second corrector stage [6]. Figure 8 shows that better computational performance can also be obtained for different Reynolds numbers when the additional explicit step applies to the SIMPLEC and PISO algorithms. It can be seen that the best computational performance can even reach 270% over that of SIMPLEC.

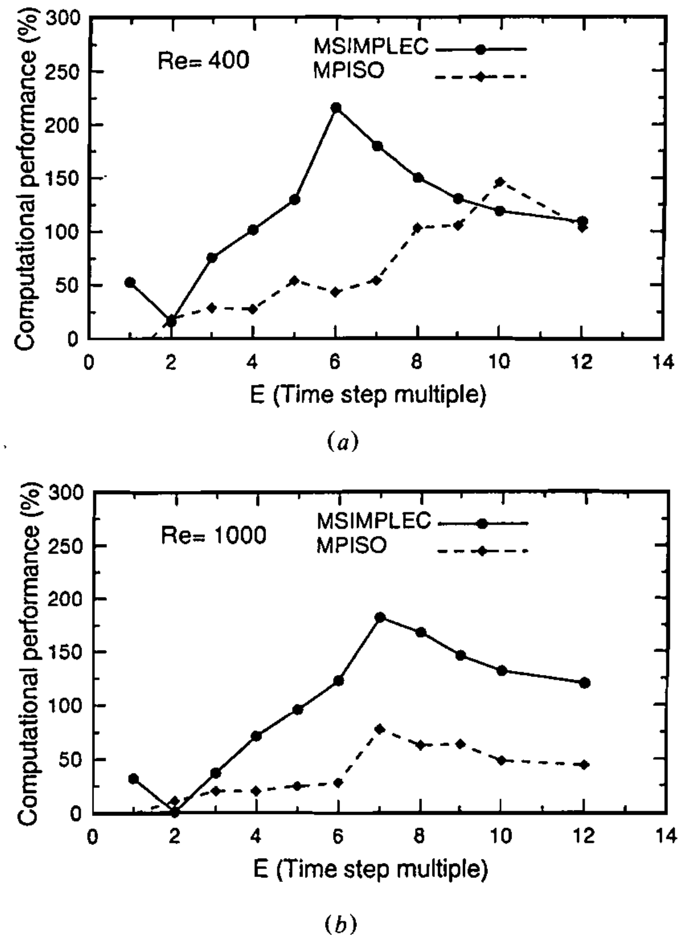


Fig. 8 Computational performance for Re = 400 and Re = 1000 (backward-facing step flow).

Rectangular Tank Flow

The third test problem, as illustrated in Fig. 1*c*, is the complex flow field established in a rectangular box when flow enters from one of the corners and exits from an opposite corner. Reynolds numbers of 500 and 2000 were tested against computation performance. The results are shown in Figs. 9*a*, 9*b*, and 9*c*. It is seen that the performance of the algorithm with the additional explicit corrector step is better than without it. Figure 10 that shows the computational performance for two difference Reynolds numbers is better without the explicit step. This leads to the same conclusion as for the previous two test problems.

CONCLUSION

Due to an assumption made on the pressure–velocity coupling for the SIMPLE algorithm and its variants, the corrected velocity can be obtained from

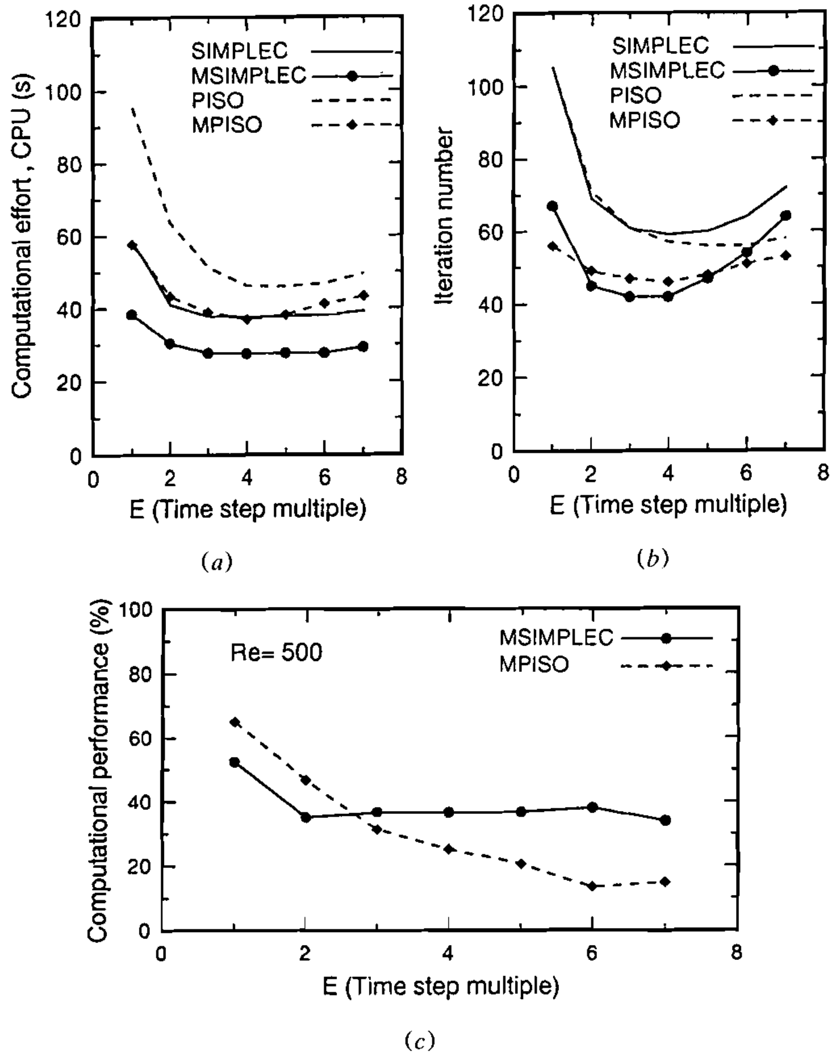


Fig. 9 Computational effort, number of iterations, and computational performance using different algorithms (rectangular tank flow, Re = 500).

the corrected pressure. However, substituting these quantities into the momentum equations may result in not satisfying the momentum equations. Therefore, the momentum equations should be solved iteratively to obtain better velocities, thus providing a better solution to the equations.

In this article an explicit corrector step is proposed that is imposed on the first corrected velocities that are obtained from the existing algorithms. This new corrector step has been tested by applying it to three flow problems: driven cavity flow, backward-facing step flow, and rectangular tank flow with different Reynolds numbers. With this additional corrector step imposed on the SIMPLEC and PISO algorithms, the results show that the number of iterations can be reduced drastically due to the much better satisfaction of the momentum equations. Considerable

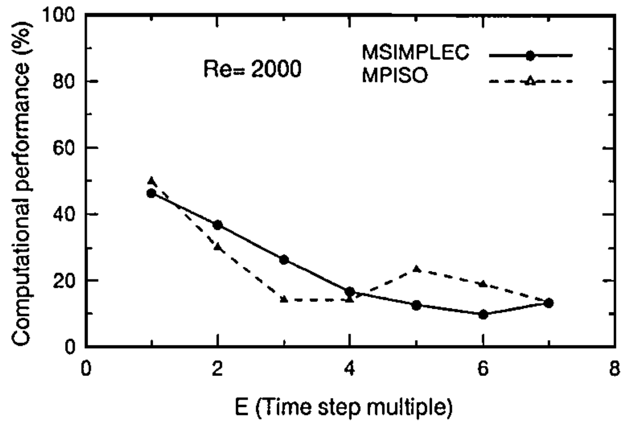


Fig. 10 Computational performance for Re = 2000 (rectangular tank flow).

savings in computing effort can be gained, and the computational performance can be as high as 270% employing the present method.

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, pp. 1787–1806, 1972.
2. S. D. Connel and P. Stow, The Pressure Correction Method, *Computers and Fluids*, vol. 14, pp. 1–10, 1986.
3. J. P. Van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, ASME Heat Transfer Conference, Denver, August 1985, Paper 85-HT-9.
4. J. P. Van Doormaal and G. D. Raithby, Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
5. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
6. B. R. Latimer and A. Pollard, Comparison of Pressure–Velocity Coupling Solution Algorithms, *Numer. Heat Transfer*, vol. 8, pp. 635–652, 1985.
7. R. I. Issa, Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting, *J. Comput. Phys.*, vol. 62, pp. 40–65, 1985.
8. S. V. Patankar, *Numerical Heat Transfer and Fluid Flows*, p. 67, Hemisphere, Washington, DC, 1980.
9. O. R. Burggraf, Analytical and Numerical Studies of the Structure of Steady Separated Flows, *J. Fluid Mech.*, vol. 24, pp. 113–151, 1966.
10. B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schonung, Experimental and Theoretical Investigation of Backward-Facing Step Flow, *J. Fluid Mech.*, vol. 127, pp. 473–496, 1983.

Received 21 September 1992
Accepted 9 March 1993

Address correspondence to Ruey-Hor Yen.