Numerical Heat Transfer, Part B, 45: 1–17, 2004

ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790490254851

A NOVEL SEGREGATED ALGORITHM FOR INCOMPRESSIBLE FLUID FLOW AND HEAT TRANSFER PROBLEMS—CLEAR (COUPLED AND LINKED EQUATIONS ALGORITHM REVISED) PART I: MATHEMATICAL FORMULATION AND SOLUTION PROCEDURE

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

A novel segregated solution procedure for incompressible fluid flow and heat transfer problems is proposed. The new algorithm is called CLEAR (Coupled and Linked Equations Algorithm Revised). It differs from all SIMPLE-like algorithms in that it solves the improved pressure directly, rather than by adding a correction term, and no term is dropped in the derivation of the pressure equation. Thus the effects of the neighboring velocity values are fully taken into account, and the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate of the iteration process. Its robustness is improved by introducing a second relaxation factor. The mathematical formulation and the solution procedure of the CLEAR algorithm are described in detail in this article. Comprehensive discussion is conducted to describe the difference between the CLEAR algorithm and all other existing algorithms of the SIMPLE family. In Part II, six numerical application examples with available numerical solutions are provided to show the feasibility of the new algorithm.

INTRODUCTION

The numerical approaches for solving the Navier–Stokes equations may be broadly divided into two categories [1, 2]: density-based and pressure-based. In the density-based approach the continuity equation serves as an equation for the density and the fluid pressure is solved from the energy and state equations. Although it works well for cases of high Mach number, for low Mach number flow and heat transfer problems it becomes unstable and its convergence rate is greatly deteriorated. On the other hand, the pressure-based approach, or the primitive-variable

Received 21 April 2003; accepted 29 May 2003.
The work reported here is supported by the National Key Project of R & D of China (G2000026303), and the National Natural Science Foundation of China (50076034, 50236010, 50276046).
Address correspondence to W. Q. Tao, School of Energy & Power Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, People's Republic of China. E-mail: wqtao@mail.xjtu.edu.cn

NOMENCLATURE

a_P, a_E, a_W, a_N, a_S	coefficients in the discretized equation	α	underelaxation factor
		β	relaxation factor
A	surface area	Γ	nominal diffusion coefficient
b	constant term in the discretized equation	$\delta x, \delta y$	distance between two adjacent grid points in x and y directions
d_e, d_n	coefficients in the velocity-correction equation	$\Delta x, \Delta y$	control volume width in x and y directions
p	pressure		
p^*	temporary pressure	μ	fluid dynamic viscosity
p'	pressure correction	ρ	fluid density
S_ϕ	source term	ϕ	general variable
u, v	velocity component in x, y directions		
$\widetilde{u}, \widetilde{u^*}$	pseudo-velocity		
$\widetilde{v}, \widetilde{v^*}$	pseudo-velocity	**Subscript**	
x, y	coordinates	nb	neighboring grid points

approach, though originally developed for solving incompressible fluid flows, has been successfully extended to compressible flows in the past two decades [3–7].

The discretization of the Navier-Stokes equations results in a set of algebraic equations that need to be solved numerically. The algebraic equations are nonlinear in the sense that the coefficients of the algebraic equations are based on temporary velocities, which are the dependent variables to be solved, and these coefficients need to be updated in the solution process. There are two different strategies to solve the resulting algebraic equations: the direct approach and the segregated approach [1, 2]. In the direct approach, the discretized momentum and continuity equations are solved simultaneously. This solution technique guarantees a close interconnection between velocities and pressure, and hence no special algorithm is needed to ensure the coupling between pressure and velocity. However, this method is seldom adopted in present engineering computations because of the following two reasons. First, in the direct method the memory required to store the various coefficients at all grid points is often prohibitive [8]; second, as indicated above, the fluid flow problem is nonlinear, and the resulting algebraic equations have to be solved repeatedly with updated coefficients, so the use of the direct method is usually not economical [9]. In the segregated approach, the algebraic equations for different velocity components (u and v for the 2-D case) are solved sequentially with a guessed pressure field or a field determined from a given velocity field. Such a pressure field cannot guarantee that the velocity field predicted from the discretized momentum equations satisfies the mass conservation constraint, hence it should be improved. Since pressure does not have its own governing equation, a problem occurs during the segregated solution process: how to improve the guessed pressure field such that the correspondingly improved velocity satisfies the continuity equation. Only with the process of iteration can the predicted velocity gradually satisfy both the momentum equation and the continuity condition, so we can obtain a converged solution. The pressure-correction method was proposed to accomplish this major task.

It should be noted that within the framework of the pressure-based approach with the segregated solution strategy, a number of numerical methods can be listed:

the fractional step method [10], the artificial compressibility method [11], the penalty method [12], and the pressure-correction method. Statistics of references published in the past three decades definitely show that the pressure-correction method is the most widely used one in the literature. The purpose of the present study is to propose a novel solution algorithm for the pressure-correction method.

The first pressure-correction algorithm was the SIMPLE method, proposed by Patankar and Spalding in 1972 [13]. The acronym SIMPLE stands for Semi-Implicit Method for the Pressure-Linked Equation. The major approximations made in the SIMPLE algorithm are: (1) the initial pressure field and the initial velocity fields are assumed independently, hence the inherent interconnection between pressure and velocity is neglected, leading to some inconsistency between them; and (2) the effects of the pressure corrections of the neighboring grids are arbitrarily dropped in order to simplify the solution procedure, thus making the algorithm semi-implicit. These assumptions will not affect the final solutions if the iterative process converges [9]. However, they do affect the convergence rate. As described in [1], the great simplicity of the SIMPLE algorithm comes from neglecting the terms that couple neighboring velocity values in the equation for the velocity correction. However, this can also cause slow convergence of the SIMPLE algorithm, and it has been found this neglect tends to overpredict the pressure correction so that underrelaxation for the pressure correction has to be resorted to in order to stabilize the iterative procedure. 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 [14–26].

The SIMPLER algorithm [14] successfully overcomes the first approximation, and is widely used in the current CFD/NHT community. Even though there are more than 10 variants of the SIMPLE-like algorithm, the second approximation, i.e., the drop of the neighboring grid effects, have not been successfully resolved so far. In 1984, van Doormaal and Raithby proposed the SIMPLEC algorithm [15], in which, by changing the definition of the coefficients of the velocity-correction equation, the effects of this drop is partially compensated. In the algorithm SIMPLEX [16, 17], by solving a set of algebraic equation for the coefficients in the velocity-correction equations, the effects of dropping the neighboring grids are also taken into account to some degree. Neither SIMPLEC nor SIMPLEX overcomes the approximation totally. This is why the behavior of SIMPLEC or SIMPLEX is not always superior to that of SIMPLE, exhibiting a highly problem-dependent character of the algorithms. In 1985, the PISO method was proposed by Issa [18] to implement two or more correction steps of pressure correction. In 1986 Connel and Stow proposed two variants of the pressure-correction process [19]. Chatwani and Turan [20] proposed a pressure–velocity coupling algorithm in 1991 to determine the underrelaxation factor in the pressure-correction equation based on the minimization of the global mass residual norm. In 1992, Lee and Tzong [21] introduced an artificial source term into the pressure-linked equation to improve the convergence performance. In 1993, Yen and Liu [22] proposed the explicit correction step method to accelerate the convergence by making the velocity explicitly satisfy the momentum equation. For buoyancy-driven fluid flows, Sheng et al. [23] introduced a temperature correction into the velocity-correction equation. In 2001, Yu, Ozoe, and Tao [24] modified the SIMPLER algorithm by artificially changing the underrelaxation term to match the variable to be solved. The revised method was called MSIMPLER. All the above-mentioned algorithms and some others not mentioned above (for example, SIMPLESSEC, SIMPLESSE of [25], and the method proposed in [26]) are usually called SIMPLE-like or SIMPLE-family algorithms. The character common to all these algorithms is that a pressure-correction term is introduced to the segregated solution process to improve the velocity and the effects of the pressure corrections of the neighboring grid points are neglected. Because of this basic feature, the improvement in the convergence rate of the above proposed variants is not very large, usually of the order of tens of percentage. Recently, Moukalled and Darwish [27] made a comprehensive review and reorganization of the express format for all the pressure-correction algorithms.

The function of the pressure-correction term in the SIMPLE family algorithms is to improve the current pressure and velocity by adding their corresponding corrections such that the resulting improved velocity can satisfy the mass conservation condition at each iteration level. And this is of crucial importance to accelerate the iteration convergence, as has been clearly demonstrated in [28]. In this study, the improved velocity and pressure of each iteration level are not determined by adding a correction term to their temporary solution; instead, they are solved directly from the momentum and continuity equations, genuinely avoiding the introduction of a pressure-correction term and a velocity-correction term. Thus the second approximation of the SIMPLE algorithm is totally discarded, making the algorithm fully implicit. The novel algorithm is named CLEAR, standing for Coupled and Linked Equations Algorithm Revised. Because of this key improvement, the convergence rate of the iterative procedure can be drastically increased, and the enhancement ranges from several times to tens of percents.

In the following, the major solution steps of SIMPLER are first briefly reviewed, followed by a detailed description of the CLEAR algorithms. Then discussion of several interesting issues relevant to the CLEAR algorithm is conducted. In a companion article the new algorithm is applied to six examples with available numerical solutions, and comparisons are made with the computations using the SIMPLER algorithm.

REVIEW OF THE SIMPLER ALGORITHM

For simplicity of presentation, all the discussion and computations are conducted for two-dimensional cases; the extension to the three-dimensional situation is straightforward. The review of the SIMPLER algorithm is conducted only for Cartesian coordinates, while in the application part, examples in three two-dimensional orthogonal coordinates are provided.

For a two-dimensional incompressible fluid flow problem in Cartesian coordinates, the governing equations in conservative form are as follows.

Continuity equation:

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

Momentum equation:

$$\frac{\partial(\rho u u)}{\partial x} + \frac{\partial(\rho v u)}{\partial y} = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho vv)}{\partial y}=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

The above three equations can be expressed in the following general form:

$$\frac{\partial}{\partial x}(\rho u\phi)+\frac{\partial}{\partial y}(\rho v\phi)=\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)+S_\phi \tag{4}$$

Equation (4) is discretized by the finite-volume method (FVM, [9, 29]) on a staggered grid system as shown in Figure 1. The source term S_ϕ is linearized as follows [9, 29]:

$$S_\phi = S_C + S_P\phi_P \qquad (\text{with } S_P \le 0) \tag{5}$$

The resulting formulation of the discretization equation takes the following form:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b \tag{6}$$

Underrelaxation of the dependent variables is incorporated into the solution process of the algebraic equations; then Eq. (6) becomes

$$\frac{a_p}{\alpha}\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b + \frac{1-\alpha}{\alpha}a_p\phi_P^0 \tag{7}$$

The denominator of the left-hand-side term and the last term on the right-hand side of Eq. (7) are the outcome of this underrelaxation process.

For velocity components, the pressure gradient term is usually separated from the source term b. With a pressure field solved from the velocity of the previous

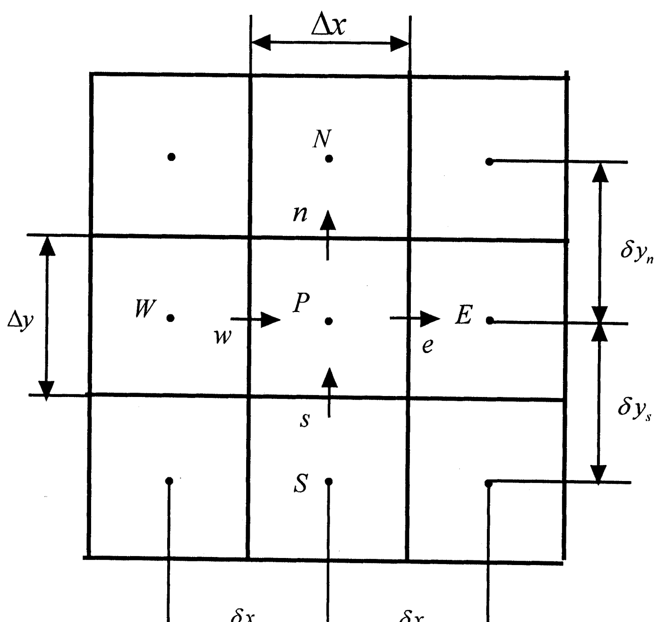

Figure 1. Control volumes in 2-D Cartesian coordinates.

iteration, the temporary or intermediate velocity solution of the current iteration, u^*, v^*, can be expressed by Eqs. (8*a*) and (8*b*):

$$\frac{a_e}{\alpha_u} u_e^* = \sum a_{\mathrm{nb}} u_{\mathrm{nb}}^* + b + A_e(p_P^* - p_E^*) + \frac{1-\alpha_u}{\alpha_u} a_e u_e^0 \tag{8a}$$

$$\frac{a_n}{\alpha_v} v_n^* = \sum a_{\mathrm{nb}} v_{\mathrm{nb}}^* + b + A_n(p_P^* - p_N^*) + \frac{1-\alpha_v}{\alpha_v} a_n v_n^0 \tag{8b}$$

where u^0, v^0 denoting the solutions of u and v of the previous iteration.

Based on the intermediate value, velocity and pressure have to be modified so that the updated velocity satisfies the discretized continuity equation. In order to get an improved velocity field, velocity-correction terms, denoted by u', v', are introduced, and for this purpose, a corresponding pressure-correction term, denoted by p', is also introduced. The improved pressure and velocity are expressed as follows:

$$p = p^* + p' \tag{9}$$

$$\begin{aligned} u_e &= u_e^* + u_e' \\ v_n &= v_n^* + v_n' \end{aligned} \tag{10}$$

The improved pressure and velocities are then substituted into the discretized momentum equation, Eq. (8*a*), yielding

$$\frac{a_e}{\alpha_u}(u_e^* + u_e') = \sum a_{\mathrm{nb}}(u_{\mathrm{nb}}^* + u_{\mathrm{nb}}') + b + A_e\left[(p_P^* + p_P') - (p_E^* - p_E')\right] + A_e \frac{1-\alpha_u}{\alpha_u} a_e u_e^0 \tag{11}$$

Subtracting Eq. (8*a*) from Eq. (11), the equation of velocity correction u_e' is obtained:

$$\frac{a_e}{\alpha_u} u_e' = \sum u_{\mathrm{nb}} u_{\mathrm{nb}}' + (p_P' - p_E') \tag{12}$$

Similarly, for the v component we have

$$\frac{a_e}{\alpha_v} v_n' = \sum v_{\mathrm{nb}} v_{\mathrm{nb}}' + A_n(p_P' - p_N') \tag{13}$$

From Eq. (12) and Eq. (13), it can be found that the velocity-correction term includes two parts, the velocity corrections in the vicinity of control volumes and the difference in pressure correction of two adjacent grid points. In SIMPLE-like methods, the term $\sum a_{\mathrm{nb}} u_{\mathrm{nb}}'$ is neglected in order to make the final pressure-correction equation manageable [9]. The final velocity-correction terms are expressed in the following forms:

$$u_e' = d_e(p_P' - p_E') \tag{14a}$$

$$v_n' = d_n(p_P' - p_N') \tag{14b}$$

where

$$d_e = \frac{A_e \alpha_u}{a_e} \tag{15a}$$

$$d_n = \frac{A_n \alpha_v}{a_v} \tag{15b}$$

The improved velocity, $u = u^* + u', v = v^* + v'$, is substituted into the discretized continuity equation (16),

$$(\rho u)_e A_e - (\rho u)_w A_w + (\rho v)_n A_n - (\rho v)_s A_s = 0 \tag{16}$$

the final pressure-correction equation is expressed as

$$a_P p'_P = \sum a_{\mathrm{nb}} p'_{\mathrm{nb}} + b \tag{17}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{18a}$$

$$a_E = (\rho A d)_e \qquad a_W = (\rho A d)_w \qquad a_N = (\rho A d)_n \qquad a_S = (\rho A d)_s \tag{18b}$$

$$b = (\rho u^* A)_w - (\rho u^* A)_e + (\rho v^* A)_s - (\rho v^* A)_n \tag{18c}$$

In the SIMPLER algorithm, the pressure correction is used only to modify velocity. The pressure is determined by a pressure equation, which is derived as follows. The u-momentum equation can be recast into

$$u_e = \frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}} + b}{a_e} + d_e(p_P - p_E) = \widetilde{u}_e + d_e(p_P - p_E) \tag{19}$$

where $\widetilde{u_e}$ is called pseudo-velocity. Similarly, for the v component we have

$$v_n = \widetilde{v}_n + d_n(p_P - p_N) \tag{20}$$

Again substituting Eqs. (19) and (20) into the continuity equation (16), we obtain the pressure equation as

$$a_P p^*_P = \sum a_{nb} p^*_{nb} + b \tag{21}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{22a}$$

$$a_E = (\rho A d)_e \qquad a_W = (\rho A d)_w \qquad a_N = (\rho A d)_n \qquad a_S = (\rho A d)_s \tag{22b}$$

$$b = (\rho \widetilde{u^0} A)_w - (\rho \widetilde{u^0} A)_e + (\rho \widetilde{v^0} A)_s - (\rho \widetilde{v^0} A)_n \tag{22c}$$

From the above derivation, we can see that the intermediate values u^*, v^* satisfy the momentum equation, and the improved values u_e, v_n satisfy the continuity equation. The improved values are taken as the solution of the current iteration level to start the next iteration. Then the consistency condition is satisfied for the singular coefficient matrix of velocity, and the convergence rate can be accelerated [28]. The converged solution we are searching for is the one which satisfies both the momentum equation and the continuity equation.

MATHEMATICAL FORMULATION OF THE CLEAR ALGORITHM

A New Expression for Improved Velocity

In the correction stage, the temporary solution of the current iteration in the SIMPLER algorithm is expressed as

$$u_e = u_e^* + d_e(p'_P - p'_E) \tag{23}$$

Equation (23) is similar to Eq. (19), where the pseudo-velocity is introduced. And for the convenience of discussion Eq. (19) is rewritten as follows:

$$u_e = \widetilde{u}_e + d_e(p_P - p_E) \tag{24}$$

Here $\widetilde{u}_e$ and u_e^* are at the same position, and the terms $(p_P - p_E)$ and $(p'_P - p'_E)$ play a similar role in the two equations. Hence we may assume that in the corrector step, the improved velocities, u and v, and the improved pressure, p, can be related by the same type of equation:

$$u_e = \widetilde{u_e^*} + d_e(p_P - p_E) \tag{25a}$$

$$v_n = \widetilde{v_n^*} + d_n(p_P - p_N) \tag{25b}$$

where the pseudo-velocities, $\widetilde{u^*}, \widetilde{v^*}$, are based on the temporary solution u^*, v^*, and can be determined after the momentum equations have been solved. Equations (25*a*) and (25*b*) are the expressions for the improved velocity in the new algorithm. As will be seen later, it is these new expressions that avoid neglecting some terms in deriving the equation for the improved pressure.

New Expression for the Updated Pseudo-Velocity

In order to set an extra access for controlling the convergence process, in the determination of the new (or updated) pseudo-velocity an extra relaxation factor, β, is introduced, and the improved velocity is rewritten as

$$u_e = \frac{\sum a_{\text{nb}} u_{\text{nb}}^* + b + [(1-\beta_u)/\beta_u] a_e u_e^*}{a_e/\beta_u} + d_e(p_P - p_E) = \widetilde{u_e^*} + d_e(p_P - p_E) \tag{26a}$$

$$v_n = \frac{\sum a_{\text{nb}} v_{\text{nb}}^* + b + [(1-\beta_v)/\beta_v] a_n v_n^*}{a_n/\beta_v} + d_n(p_P - p_N) = \widetilde{v_n^*} + d_n(p_P - p_N) \tag{26b}$$

Hereafter β is called the second relaxation factor.

Equation for the Improved Pressure

The improved velocity should satisfy the mass conservation condition. Thus, substituting Eqs. (26*a*), (26*b*) into Eq. (16), we get the equation for the improved pressure:

$$a_P p_P = \sum a_{\mathrm{nb}} p_{\mathrm{nb}} + b \tag{27}$$

where

$$a_P = a_E + a_W + a_N + a_S \tag{28a}$$

$$a_E = (\rho A d)_e \qquad a_W = (\rho A d)_w \qquad a_N = (\rho A d)_n \qquad a_S = (\rho A d)_s \tag{28b}$$

$$b = (\rho \widetilde{u^*} A)_w - (\rho \widetilde{u^*} A)_e + (\rho \widetilde{v^*} A)_s - (\rho \widetilde{v^*} A)_n \tag{28c}$$

the coefficients d_e, d_w, d_n, d_s are calculated based on the intermediate field u^*, v^* with the same expressions shown in Eq. (15*a*), (15*b*).

Once the improved pressure is solved, the improved velocities can be determined by Eqs. (26*a*), (26*b*), which satisfy the continuity condition. In the above derivation, we do not neglect any term, making the solution algorithm fully implicit. Compared with the SIMPLE-like algorithms, it can be stated that the effects of the neighboring grid points are totally taken into account by introducing the updated pseudo-velocity based on u^* and v^*. The above solution algorithm is called CLEAR.

Solution Procedure of the CLEAR Algorithm

The solution procedure of the CLEAR algorithm is now summarized as follows.

Step 1. Assume an initial velocity field u^0, v^0.

Step 2. Calculate the coefficient of the discretized momentum equation and pseudo-velocity $\widetilde{u^0}$, $\widetilde{v^0}$:

$$\widetilde{u_e^0} = \frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}}^0 + b + [(1-\alpha_u)/\alpha_u] a_e u_e^0}{a_e/\alpha_u} \tag{29a}$$

$$\widetilde{v_n^0} = \frac{\sum a_{\mathrm{nb}} v_{\mathrm{nb}}^0 + b + [(1-\alpha_v)/\alpha_v] a_n v_n^0}{a_n/\alpha_v} \tag{29b}$$

Step 3. Solve the pressure equation (21) and obtain pressure field p^*.

Step 4. Based on p^*, solve the momentum equations (8*a*), (8*b*), obtaining the intermediate velocity field u^*, v^*.

Step 5. Recalculate the coefficient of momentum equation and the pseudo-velocity $\widetilde{u^*}$, $\widetilde{v^*}$ based on the intermediate velocity solution u^*, v^*:

$$\widetilde{u_e^*} = \frac{\sum a_{\mathrm{nb}} u_{\mathrm{nb}}^* + b + [(1-\beta_u)/\beta_u] a_e u_e^*}{a_e/\beta_u} \tag{30a}$$

$$\widetilde{v_n^*} = \frac{\sum a_{\mathrm{nb}} v_{\mathrm{nb}}^* + b + [(1-\beta_v)/\beta_v] a_n v_n^*}{a_n/\beta_v} \tag{30b}$$

Step 6. Solve the pressure equation (27) for the improved field p.

Step 7. Improve the velocity with Eqs. (26*a*), (26*b*) to obtain the solution of the present iteration.

Step 8. Solve the discretization equations of the other scalar variables if necessary.

Step 9. Return to step 2 and repeat until convergence is reached.

DISCUSSION OF THE CLEAR ALGORITHM

The Difference between SIMPLER and CLEAR

From the above presentation, we can see that the first four steps of the solution procedure of the CLEAR algorithm are the same as those of the SIMPLER algorithm. However, in steps 5 and 6, we do not solve the pressure-correction equation; instead, we introduce a new expression for the improved velocity, calculate the updated pseudo-velocities $\widetilde{u^*}$, $\widetilde{v^*}$, and resolve the pressure equation to get the improved pressure, from which the improved velocities can be determined by Eqs. (26*a*), (26*b*). In such a way, in the entire solution process of one iteration, we do not neglect any term, making the solution method fully implicit. This is the key difference between CLEAR and any SIMPLE-family algorithm.

It can be seen that in the CLEAR algorithm, the pressure equation is solved twice: in step 3 and in step 6. In step 3, the pressure equation is solved to provide a source term for solving the momentum equation. In step 6, the pressure equation is solved to improve the intermediate velocity. It is to be noted that in step 5 we calculate the coefficients of the momentum equation to evaluate the updated pseudo-velocities, $\widetilde{u^*}$, $\widetilde{v^*}$, but we do not resolve the momentum equation thereafter. Thus the computational effort in one iteration of the CLEAR algorithm increases only a bit compared with the SIMPLER algorithm. However, the coupling between velocity and pressure in the CLEAR algorithm is much better than that in the SIMPLER algorithm, leading to an appreciable improvement in the convergence rate, as can be seen from the companion article.

The Second Relaxation Factor

In the determination of the updated pseudo-velocity of step 5, we introduce a second relaxation factor β. This is based on the following consideration. As indicated above, the improved velocity and pressure are fully consistent. The good coupling between the pressure and velocity in the CLEAR algorithm can appreciably enhance the convergence rate. This implies that the changes of the velocity solution between two successive iterations are usually larger than those of the SIMPLER algorithm. For the iteration solution procedure of a nonlinear problem, experience shows that too large a variation of the dependent variables between two successive iterations may lead to diverge of the iteration process [29]. Therefore the second relaxation factor β is introduced in step 5 to present an extra access for controlling the iteration process. From Eqs. (30*a*), (30*b*) it can be observed that β appears in both the denominator and nominator. However, the relaxation part is usually not dominated compared to the other terms. Thus a larger value of β will lead to a larger updated value of the pseudo-velocity, hence alleviating the burden of the pressure gradient term, and reduces the variation rate between two successive iterations. Therefore the second relaxation factor may take values varying in a wide range. If $\beta > 1$, the updated pseudo-velocity is overrelaxed, while the pressure is somewhat underrelaxed; if $\beta < 1$, the situation is the opposite. For simplicity, we set the following relation between the two relaxation factors β and α in the computations of the six examples presented in the companion article:

$$\beta = \begin{cases} 0.5 & 0 < \alpha \leq 0.5 \\ 1 & 0.5 < \alpha \leq 1 \end{cases} \tag{31}$$

The implicit meaning of Eq. (31) is further explained as follows. When the momentum equation is solved with a small value of underrelaxation factor α, the relative change between the temporary velocity and the previous value is usually mild, hence the improved pressure may have a relatively large variation, and in some sense it can be overrelaxed (a small value of β meets this requirement). If the momentum equation is solved with a larger underelaxation factor, then the change between the temporary velocity and the previous solution is probably large, and the variation of pressure is better to be mildly or even relatively small, then a large value of β may be adopted. Our numerical practices show that when the value of α is in the vicinity of 0.9, the value of β sometimes should be larger than 1 in order to get a converged solution. Therefore the two relaxation factors have different functions. The parameter α is the underelaxation factor for the solution of the temporary velocity, and the larger the value of α, the larger is the change of velocities between two successive iterations. The parameter β is the relaxation factor for the updated is the change of the pseudo-velocity, and the larger the value of β, the larger is the pseudo-velocity, and the smaller the relative change between the improved and the previous pressure. Thus, as far as the improved pressure is concerned, the larger the value of β, the more appreciable is the underrelaxation function. It is worth noting that in the SIMPLE-like algorithms, the underelaxation is executed directly for the pressure-correction term, since, when the converged solution is reached, the pressure-correction approaches zero, hence the underrelaxation will not affect the final solution. In the CLEAR algorithm, however, the under–or overrelaxation of the pressure term can only be implemented indirectly via the pseudo-velocity, since in Eqs. (26*a*) and (26*b*), and the pressure in Eq. (27), because what we solve is the improved pressure itself, rather than its correction term.

Boundary Condition for the Two Pressure Equations

For the two pressure equations, (21) and (27), their boundary condition treatment method is the same, i.e., setting the related coefficient of the pressure equation to zero, so no information is needed at the related boundary point. This can be briefly demonstrated by referring to the control volume adjacent to the boundary shown in Figure 2. In engineering computations there are usually two situations: either the boundary pressure is known or the normal velocity component is specified. For these two situations we can analyze as follows: if u_e is known, then in the discretized form of the continuity equation for control volume P, the known value of u_e is adopted, and there is no need to introduce the term $(p_P - p_E)$. This implies that in the discretized form of the continuity equation of control volume P, the term $a_E p_E$ does not appear. This is equivalent to setting $a_E = 0$. If the boundary pressure p_E is specified, then it can be introduced directly into the pressure equation for the control volume P. This known term, $a_E p_E$, should be incorporated into the b

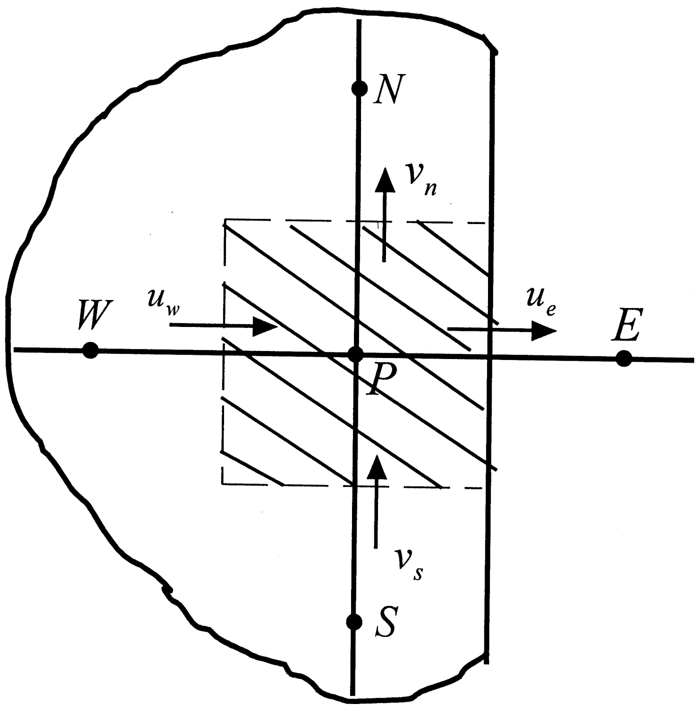

Figure 2. Illustration describing the boundary condition treatment of the pressure equation.

term when the algebraic equations are to be solved, and this is equivalent to setting $a_E = 0$ in the final algebraic equation of the pressure.

It is to be noted that such simple treatment of the outflow boundary condition for the pressure equation in the pressure-correction method once was disputed in the literature [30]. Now it is widely accepted that such a treatment is equivalent to adopting the homogenous Neumann boundary condition for the pressure equation [1, 31]. This is because, for incompressible flow, velocity field can be uniquely determined with a specified pressure gradient, rather than pressure itself. Thus such simple treatment of the pressure boundary condition does not suffer from any ambiguity.

Difference between CLEAR and FIMOSE

In 1985, Latimer and Pollard proposed an algorithm called FIMOSE (Fully Implicit Method for Operator-Split Equation) [32]. FIMOSE is fully implicit in the sense that no assumptions were made during the development of the algorithm and no pressure-correction equation is solved. In this regard, CLEAR and FIMOSE have something in common. However, the way by which the two algorithms realize the fully implicit result is totally different. The major solution steps of FIMOSE [32] can be best illustrated by applying FIMOSE to the flow over a backward-facing step (Figure 3).

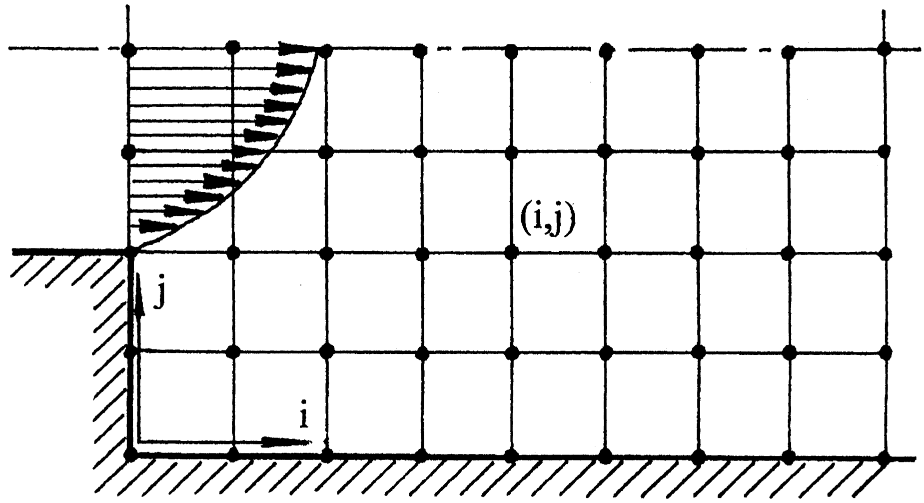

Figure 3. Illustration for execution of FIMOSE algorithm.

In the FIMOSE algorithm two integral conservation equations are introduced to maintain a global balance, i.e., coupling, between the velocity and pressure. These two integral equations are derived as follows.

From the specified inlet velocity field, the flow rate at the inflow boundary can be determined:

$$m_{\text{in}} = \sum_{j}^{\text{inflow}} \rho u_{\text{in},j} A_{\text{in},j} \tag{32}$$

The temporary velocities u^*, v^* may not satisfy the mass conservation constraint, so a line-averaged velocity correction for the u velocity may be determined by the following equation:

$$\overline{u'} = \frac{\dot{m} - \sum_j \rho u^*_{i,j} A_{i,j}}{\rho \times (\text{flow area})} \tag{33}$$

This is the first integral equation to get the line-average velocity correction. When this line-averaged velocity correction is added to each temporary velocity along a line, the velocity field will satisfy the continuity in a global manner. The adjustments to the velocity field must be supported by a change in the overall pressure gradient. The average pressure gradient along that line can be obtained as follows. Introducing the velocity-correction equation (23) into Eq. (32),

$$\dot{m}_{\text{in}} = \sum_j \rho [u^*_e + d_e(p'_P - p'_E)]_{i,j} A_{i,j} \tag{34a}$$

Replacing the local pressure-correction difference by an average one, denoted by $\overline{\Delta p'_i}$, we have

$$\overline{\Delta p_i'} = \frac{m_{in} - \sum_j \rho u_{i,j}^* A_{i,j}}{\sum_j \rho (d_e)_{i,j} A_{i,j}} \tag{34b}$$

The change in pressure should be added to all grid points located downstream of the line computed to maintain the mass flow rate. Equation (34*b*) is the second integral equation introduced in the FIMOSE algorithm.

The solution steps of the FIMOSE algorithm is now summarized as follows.

1. Specify the inlet velocity profile, initializing the velocity and pressure field, denoted by u^0, v^0 and p^0.
2. Calculate the coefficients and source term of the discretized momentum equations.
3. Solve the momentum equations based on the specified pressure filed to get u^*, v^*.
4. Apply the two integral equations, Eqs. (33), (34*b*) along each line of the domain to ensure the coupling between velocity and pressure.
5. Calculate the pseudo-velocity based on u^*, v^*, and solve the pressure equation to get p^*.
6. Based on p^*, re-solve the momentum equation to get u^{**}, v^{**}.
7. Apply the integral conservation equations to each grid line.
8. Recalculate the pseudo-velocity and re-solve the pressure equation to get p^{**}.
9. Based on p^{**}, re-solve the momentum equation to get u^{***}, v^{***}.
10. Apply the integral conservation equations to each grid line.
11. Return to step 2, treating u^{***}, v^{***} and p^{**} as the initial velocity and pressure fields. Iterate until convergence is reached.

From the above description, it can be seen that the FIMOSE algorithm differs from CLEAR in several fundamental ways. First, in FIMOSE the coupling between the velocity and pressure is enhanced by introducing two integral conservation equations, and these two equation are applied three times during one iteration. Second, in FIMOSE the momentum equations are solved thee times and the pressure equations are solved twice, leading to considerable increase in computational effort. Third, even though the pressure-correction equation is not solved, the pressure-correction term and the velocity-correction term are still kept in the FIMOSE algorithm. The discarding of the approximation of dropping the effects of the neighboring grid points is realized at the cost of introducing two more integral equations and repeatedly solving the pressure and momentum equations. Finally, as indicated in [33], only when the flow situation contains one predominant flow direction is the use of the two integral equations appropriate. Because of the above reasons, FIMOSE does not exhibit an obvious advantage compared with other existing algorithms. In the CLEAR algorithm, by introducing a new expression for the improved velocity fields based on the temporary solution u^*, v^*, the improved pressure is solved directly and the relevant improved velocity filed is obtained by explicit equations. Computational effort is greatly saved, while the coupling between the velocity and pressure is fully ensured.

CONCLUSIONS

In this article a novel pressure-correction algorithm, CLEAR, is proposed. The major features of the new algorithm are as follows.

1. A new expression for improving the velocity is introduced by mimicking the velocity-correction equation in the SIMPLE algorithm and the momentum equation with the pseudo-velocity and pressure gradient term in the SIMPLER algorithm. It is this new expression that avoids neglecting the terms that couples neighboring velocity values in the equation for the improved velocity, making the algorithm fully implicit.
2. In one iteration, the pressure equation is solved twice, while the momentum equation is only solved once. The computational effort of one iteration in the CLEAR algorithm is only a bit larger than that of the SIMPLER algorithm. However, the coupling between the pressure and velocity is fully taken into account in the CLEAR algorithm, hence the convergence rate can be greatly enhanced. Application examples will be provided in a companion article.
3. Two relaxation factors are introduced in the CLEAR algorithm: one for the temporary velocity when solving the discretized momentum equation (α), and the other for the updated pseudo-velocity when solving the improved pressure (β). A larger value of β is equivalent to underrelaxing the improved pressure. When the value of α is in the vicinity of its upper limit (0.9–1.0), it is recommended to take a large value of β, near or greater than 1.0.
4. Detailed discussion is provided in the article, showing that the CLEAR algorithm differs from all the existing variants of SIMPLE-like algorithms. However, for those existing codes based on the SIMPLE or SIMPLER algorithm, the incorporation of the CLEAR algorithm is very easy. It is thus expected that the CLEAR algorithm will be widely adopted in computations of incompressible fluid flow and heat transfer problems. Extensions to a collocated grid system and to compressible fluid flow cases are now underway in the authors' group.

REFERENCES

1. 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, 1998.
2. W. Q. Tao, *Recent Advances in Computational Heat Transfer*, Science Press, Beijing, 2000.
3. J. P. van Doormal, G. D. Raithby, and B. H. McDonalds, The Segregated Approach to Predicting Viscous Compressible Flows, *ASME J. Turbomachinery*, vol. 109, pp. 265–277, 1987.
4. K. C. Karki and S. V. Patankar, Pressure Based Calculation Procedure for Viscous Flows at all Speeds in Arbitrary Configurations, *AIAA J.*, vol. 27, pp. 1167–1174, 1989.
5. W. Shyy, M. H. Chen, and C.-S. Sun, Pressure Based Multigrid Algorithm for Flows at All Speeds, *AIAA J.*, vol. 30, pp. 2660–2669, 1992.
6. C. H. March and C. R. Maslika, A Non-orthogonal Finite-Volume Method for the Solution of All Speed Flows Using Collocated Variables, *Numer Heat Transfer*, vol. 26, pp. 293–311, 1994.

7. A. W. Date, Solutions of Navier-Stokes Equations on Nonstaggered Grid of All Speeds, *Numer. Heat Transfer B*, vol. 33, pp. 451–467, 1998.
8. M. E. Braaten, Development and Evaluation of Iterative and Direct Methods for the Solution of Equations Governing Recirculating Flows, Ph.D. thesis, University of Minnesota, Minneapolis, MN; 1985.
9. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
10. J. Kim, and P. Moin, Application of a Fractional Step Method to Incompressible Navier-Stokes Equations, *J. Comput. Phys.*, vol. 59, pp. 308–323, 1985.
11. S. L. Lee and R. Y. Tzong, Artificial Pressure for Pressure-Linked Equation, *Int. J. Heat Mass Transfer*, vol. 35, pp. 2705–2716, 1992.
12. M. E. Braaten and W. Shyy, Comparison of Iterative and Direct Method for Viscous Flow Calculations in Body-Fitted Coordinates, *Int. J. Numer. Meth. Fluids*, vol. 6, pp. 325–349, 1986.
13. 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.
14. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
15. 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.
16. 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.
17. 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.
18. R. I. Issa, Solution of Implicitly Discretized Fluid Flow Equation by Operator-Splitting, *J. Comput. Phys.*, vol. 62, pp. 40–65, 1985.
19. S. D. Connell and P. Stow, The Pressure Correction Methods, *Comput. Fluids*, vol. 14, pp. 1–10, 1986.
20. A. U. Chatwani and A. Turan, Improved Pressure–Velocity Coupling Algorithm Based on Global Residual Norm, *Numer. Heat Transfer B*, vol. 20, pp. 115–123, 1991.
21. L. S. Lee and R. Y. Tzong, Artificial Pressure for Pressure Linked Equation, *Int. J. Heat Mass Transfer*, vol. 35, pp. 2705–2716, 1992.
22. 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.
23. 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.
24. 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.
25. T. Gjesdal and M. E. H. Lossius, Comparison of Pressure Correction Smoothers for Multigrid Solution of Incompressible Flow, *Int. J. Numer. Meth. Fluids*, vol. 25, pp. 393–405, 1997.
26. X. Wen and D. B. Ingham, A New Method for Accelerating the Rate of Convergence of the SIMPLE-like Algorithm, *Int. J. Numer. Meth. Fluids*, vol. 17, pp. 385–400, 1993.
27. 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.
28. E. Blosch and W. Shyy, The Role of Mass Conservation in Pressure-Based Algorithms, *Numer. Heat Transfer B*, vol. 24, pp. 415–429, 1993.

29. W. Q. Tao, *Numerical Heat Transfer*, 2d ed., Xi'an Jiaotong University Press, Xi'an, China, 2001.
30. P. M. Gresho, A Simple Question to SIMPLE Users, *Numer. Heat Transfer A*, vol. 20, p. 123, 1991.
31. R. L. Sani and P. M. Gresho, Resume and Remarks on the Open Boundary Condition Minisymposium, *Int. J. Numer. Meth. Fluids*, vol. 18, pp. 983–1008, 1994.
32. B. R. Latimer and A. Pollard, Comparison of Pressure-Velocity Coupling Solution Algorithms, *Numer. Heat. Transfer*, vol. 8, pp. 635–652,1985.