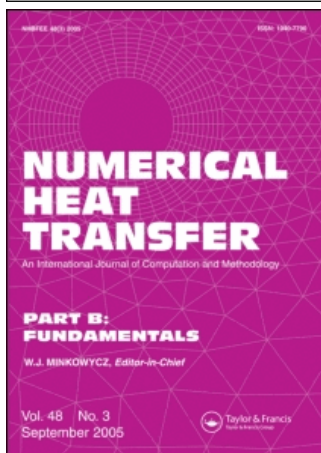


This article was downloaded by:[Xi'an Jiaotong University]
On: 10 April 2008
Access Details: [subscription number 788749276]
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954
Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Numerical Heat Transfer, Part B: Fundamentals

An International Journal of Computation and Methodology

Publication details, including instructions for authors and subscription information:
<http://www.informaworld.com/smpp/title~content=t713723316>

An Efficient Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems - IDEAL (Inner Doubly Iterative Efficient Algorithm for Linked Equations) Part I: Mathematical Formulation and Solution Procedure

D. L. Sun ^a; Z. G. Qu ^a; Y. L. He ^a; W. Q. Tao ^a

^a 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

Online Publication Date: 01 January 2008

To cite this Article: Sun, D. L., Qu, Z. G., He, Y. L. and Tao, W. Q. (2008) 'An Efficient Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems - IDEAL (Inner Doubly Iterative Efficient Algorithm for Linked Equations) Part I: Mathematical Formulation and Solution Procedure', Numerical Heat Transfer, Part B: Fundamentals, 53:1, 1 - 17

To link to this article: DOI: 10.1080/10407790701632543

URL: <http://dx.doi.org/10.1080/10407790701632543>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article maybe used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

AN EFFICIENT SEGREGATED ALGORITHM FOR INCOMPRESSIBLE FLUID FLOW AND HEAT TRANSFER PROBLEMS—IDEAL (INNER DOUBLY ITERATIVE EFFICIENT ALGORITHM FOR LINKED EQUATIONS) PART I: MATHEMATICAL FORMULATION AND SOLUTION PROCEDURE

D. L. Sun, Z. G. Qu, Y. L. He, and W. Q. Tao

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

An efficient segregated solution procedure for incompressible fluid flow and heat transfer problems is proposed. The new algorithm is called IDEAL (Inner Doubly Iterative Efficient Algorithm for Linked Equations). In the new algorithm there exist inner doubly iterative processes for the pressure equation, which almost completely overcome two approximations in the SIMPLE algorithm. Thus the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate and stability of the iteration process. The mathematical formulation and solution procedure of the IDEAL algorithm are described in this article. In Part II, application examples are provided to show the features and feasibility of the new algorithm.

1. INTRODUCTION

The numerical approaches for solving the Navier-Stokes equations may be broadly divided into two categories [1, 2]: density-based and pressure-based. The density-based approach works well for cases of high Mach number, but for low-Mach-number flow and heat transfer problems it becomes unstable and its convergence rate deteriorates greatly. The pressure-based approach, or the primitive-variables approach, though originally developed for solving incompressible fluid flows, has been successfully extended to compressible flows, and is widely adopted in computational fluid dynamics and numerical heat transfer.

The pressure-based approaches may be divided into the direct approach and the segregated approach. In the direct approach, the discretized momentum and continuity equations are solved simultaneously. However, this approach is seldom adopted in present engineering computations because the use of the direct method

Received 29 April 2007; accepted 6 July 2007.

This work was supported by the National Natural Science Foundation of China (50476046, 50636050) and the National Key Project for R&D of China (2007CB206902).

Address correspondence to Wen-Quan Tao, State Key Laboratory of Multiphase Flow in Power Engineering, School of Energy & Power Engineering, Xi'an Jiaotong University, 28 Xian Ning Road, Xi'an, Shaanxi 710049, People's Republic of China. E-mail: wqtao@mail.xjtu.edu.cn

NOMENCLATURE

a	coefficient in the discretized equation	μ	dynamic viscosity
A	surface area	ρ	density
b	constant term in the discretized equation	Subscripts	
d	coefficient in the velocity-correction equation	e, w, n, s	cell surface
N1, N2	inner doubly iterative times	P, E, N, S, W	grid point
p	pressure	nb	neighboring grid points
S	source term	u, v	referring to u, v momentum equation
u, v	velocity component in x, y directions	ϕ	general variable
\tilde{u}, \tilde{v}	pseudo-velocity	Superscripts	
x, y	coordinates	Ptemp	temporary value in previous inner iteration step
α	underrelaxation factor	Temp	temporary value in current inner iteration step
Γ	nominal diffusion coefficient	0	previous iteration
ϕ	general variable	*	intermediate value in iteration
		'	correction

is usually not economic [3]. In the segregated approach, the variables in the momentum and continuity equations are solved sequentially and, because of its efficient and economic characteristics, this approach has long been the dominant one in many engineering applications.

The segregated approaches include the projection method [4], the artificial compressibility method [5], the penalty method [6], and the pressure-correction method [7]. Statistics of published references definitely show that the pressure-correction method is the most widely used one in the literature because of its simplicity and straightforward physical concept. The well-known SIMPLE-family algorithms [7–19] belong to this method. Hereafter we will work in the framework of pressure-correction method.

In the last several decades the tremendous improvement in computer capabilities, especially in memory and speed, has made accurate numerical predictions of complex, high-Re/Ra or fine-mesh flow cases possible. However, the convergence rate and stability of all the existing pressure-correction algorithms, such as the SIMPLE algorithm, are still not satisfying, even very poor for some flow cases. Therefore, one of the main goals of research in computational fluid dynamics and numerical heat transfer is to search for an algorithm which is efficient and stable not only for the simple, low-Re/Ra or coarse-mesh flow cases, but also for the complex, high-Re/Ra or fine-mesh flow cases. The proposal of SIMPLER, SIMPLEC, SIMPLEX, PISO, FIMOSE, CLEAR, etc. is witness to such continuous efforts in this regard. The authors' research group has made a great effort to improve the pressure-correction algorithm, including the proposal of the CLEAR algorithm [10]. Via extensive numerical practices we have found that even though the CLEAR algorithm can appreciably accelerate the convergence rate for many cases, it has a drawback of less robustness, and for some cases, its acceleration function becomes not significant. This stimulates our research interests in further improving the

pressure-correction algorithm. Based on carefully analyzing the advantages and disadvantages of the existing algorithms, we have finally reached a new algorithm, which is called IDEAL.

In the following the major solution procedure of SIMPLE is first briefly reviewed, followed by a discussion of its variants such as SIMPLER, SIMPLEC, SIMPLEX, PISO, and FIMOSE. Then, on the basis of discussion of the SIMPLE-family algorithms, the IDEAL algorithm is described in detail, and an analysis of the IDEAL algorithm is conducted. In a companion article [20] the IDEAL algorithm is applied to four examples with available numerical solutions, and comparisons are made with computations using the popular SIMPLER algorithm.

2. BRIEF REVIEW OF THE SIMPLE ALGORITHM

In 1972, Patankar and Spalding [7] proposed a calculation procedure called SIMPLE for three-dimensional parabolic flows. For the convenience of further presentation the major points of the SIMPLE algorithm are briefly reviewed here. For simplicity of presentation, the review of the SIMPLE algorithm is conducted only for fluid flow problems in two-dimensional Cartesian coordinates. It should be noted that although the details of the implementations of SIMPLE-like algorithms are well documented in the literature [3], in this article we review the process from quite a different point of view. In addition, this review also sets the stage for the development of the new algorithm, IDEAL.

2.1. Basic Mathematical Formulation

The governing equations of fluid flow in conservative form are expressed as follows.

Continuity equation:

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

Momentum equation:

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

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

where u , v are the velocity components in the x , y directions, p is the pressure, μ is the fluid dynamic viscosity, and ρ is the fluid density. If the dependent variable

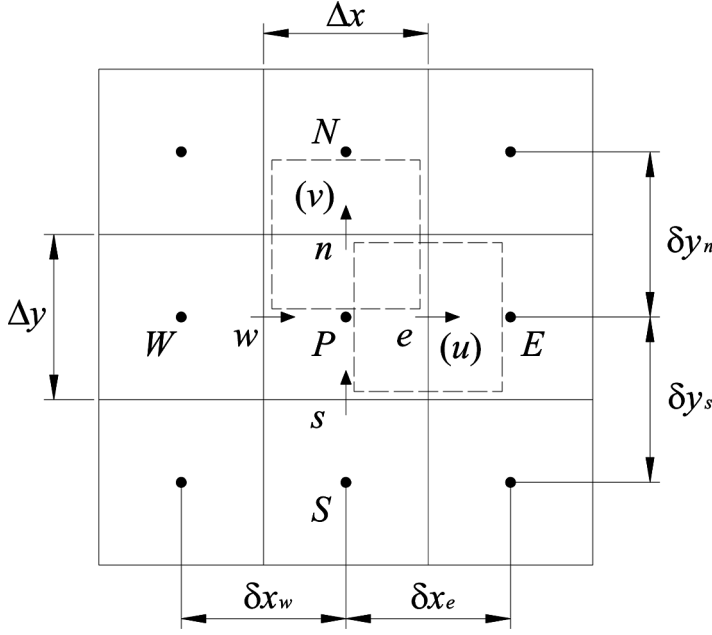


Figure 1. Control volumes on a staggered grid system in 2-D cartesian coordinates.

is denoted by ϕ , the corresponding form of the general differential equation is

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

where Γ is the nominal diffusion coefficient and S_ϕ is the source term. Equation (4) is discretized by the finite-volume method (FVM) [3, 21] on a staggered grid system as shown in Figure 1.

To enhance the convergence rate, the source term S_ϕ is linearized as follows:

$$S_\phi = S_C + S_P \phi_P \quad (5)$$

where $S_P \leq 0$, which guarantees the requirement of diagonal dominant.

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

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (6)$$

Underrelaxation of the dependent variable is incorporated into the solution process of the algebraic equations, so Eq. (6) becomes

$$\frac{a_P}{\alpha} \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b + \frac{1 - \alpha}{\alpha} a_P \phi_P^0 \quad (7)$$

where a_P , a_E , a_W , a_N , and a_S are coefficients of the discretized equation and α is the underrelaxation factor.

For u and v components, the pressure-gradient term is usually separated from the source term b . With an initial pressure field p^* , the intermediate velocity solution of the current iteration level can be obtained from Eq. (7) and can be expressed as follows:

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

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

where u^0 , v^0 denote the solutions of the previous iteration level.

The intermediate solutions u^* , v^* shown in Eqs. (8a) and (8b) can only satisfy the momentum equation; the mass conservation condition may not be satisfied. Thus the pressure p^* and the intermediate velocities u^* , v^* have to be modified so that the updated velocity satisfies the continuity equation. As was demonstrated in [22], satisfaction of mass conservation condition is of the most importance for the convergence of iteration in the pressure-correction method. In order to get an improved velocity field, the pressure correction p' and the corresponding velocity corrections u' , v' are introduced. The improved pressure and velocity are expressed as follows:

$$p = p^* + p' \quad (9)$$

$$u_e = u_e^* + u_e' \quad (10)$$

$$v_n = v_n^* + v_n' \quad (11)$$

Equations (9) and (10) are then substituted into the discretized u -component momentum, Eq. (8a), yielding

$$\frac{a_e}{\alpha_u} (u_e^* + u_e') = \sum a_{nb} (u_{nb}^* + u_{nb}') + b + A_e[(p_P^* + p_P') - (p_E^* + p_E')] + \frac{1 - \alpha_u}{\alpha_u} a_e u_e^0 \quad (12)$$

Subtracting Eq. (8a) from Eq. (12), the equation of velocity correction u' can be obtained:

$$\frac{a_e}{\alpha_u} u_e' = \sum a_{nb} u_{nb}' + A_e(p_P' - p_E') \quad (13)$$

Similarly, the equation of velocity correction v' is

$$\frac{a_n}{\alpha_v} v_n' = \sum a_{nb} v_{nb}' + A_n(p_P' - p_N') \quad (14)$$

In the SIMPLE method, the terms $\Sigma a_{nb}u'_{nb}$ and $\Sigma a_{nb}v'_{nb}$ are neglected in order to make the final pressure-correction equation manageable. The final velocity-correction equations are expressed as follows:

$$u'_e = d_e(p'_P - p'_E) \quad (15a)$$

$$v'_n = d_n(p'_P - p'_N) \quad (15b)$$

where

$$d_e = \frac{A_e \alpha_u}{a_e} \quad d_n = \frac{A_n \alpha_v}{a_n}$$

The improved velocity $u = u^* + u'$, $v = v^* + v'$ and the related velocity-correction equations (15a) and (15b) are then substituted into the discretized continuity equation (16):

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

The pressure-correction equation is obtained as

$$a_P p'_P = \sum a_{nb} p'_{nb} + b \quad (17)$$

where

$$a_P = a_E + a_W + a_N + a_S \quad (18a)$$

$$a_E = (\rho A d)_e \quad a_W = (\rho A d)_w \quad a_N = (\rho A d)_n \quad a_S = (\rho A d)_s \quad (18b)$$

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

2.2. Solution Procedure of the SIMPLE Algorithm

The pressure-correction solution method is iterative in nature. In the following we will often use the term “iteration level.” By one iteration level we mean that all the computations are completed at the same values of the coefficients and source term of the discretized momentum and continuum equations.

The solution steps at each iteration level of the SIMPLE algorithm are now summarized as follows.

- Step 1. Assume an initial velocity field u^0 , v^0 and guess the initial pressure field p^* .
- Step 2. Calculate the coefficients in the momentum equations and solve the momentum equations (8a) and (8b) to obtain u^* , v^* .
- Step 3. From u^* , v^* , obtain the mass source b by Eq. (18c) and solve the pressure-correction equation (17), then calculate the pressure p and the velocity u , v from Eqs. (9)–(11) by the pressure correction p' .
- Step 4. Take the improved velocity u , v and improved pressure p as the initial velocity and initial pressure of the next iteration level, and return to Step 2. Repeat until convergence is reached.

2.3. Approximations in the SIMPLE Algorithm

In Step 1, the initial pressure field and the initial velocity field are assumed independently, hence the inherent interconnection between pressure and velocity is neglected, leading to some inconsistency between them. This is the first approximation in the SIMPLE algorithm. This approximation is related to the consistency between initial pressure and velocity. For convenience of presentation, it will hereafter be called the “initial field consistency issue,” or simply the consistency issue.

In Step 3, the effects of the velocity corrections of the neighboring grids are arbitrarily dropped in order to simplify the pressure-correction solution procedure, thus making the algorithm semi-implicit. This is the second approximation in the SIMPLE algorithm. This approximation makes the improved velocity satisfy only the continuity equation, leading to some instability and a slow convergence rate for the iteration process. It has been found that this neglect tends to overpredict the pressure correction so that underrelaxation for the pressure correction has to be resorted in order to stabilize the iterative procedure. This approximation actually concerns the completeness of the pressure-correction equation (or pressure equation), and for convenience of presentation, it will hereafter be called the “pressure equation completeness issue,” or simply the completeness issue.

These two approximations will not affect the final solutions if the iterative process converges [3]. However, they do affect the convergence rate and stability. Therefore, since the proposal of the SIMPLE algorithm, a number of variants have been proposed in order to overcome one or both of these approximations.

3. DISCUSSION OF THE SIMPLE-ALGORITHM VARIANTS

3.1. The Algorithms Overcoming the First Approximation in the Simple Algorithm

In 1981, Patankar proposed the SIMPLER algorithm [8], which was the first method for overcoming the first approximation in the SIMPLE algorithm. To overcome the inconsistency between the initial pressure field and the initial velocity field, the initial pressure is determined by a pressure equation, which is derived as follows.

The u -component momentum can be recast into

$$u_e = \frac{\sum a_{nb} u_{nb}^0 + b}{a_e / \alpha_u} + d_e(p_P^* - p_E^*) = \tilde{u}_e^0 + d_e(p_P^* - p_E^*) \quad (19)$$

where the term $[(1 - \alpha_u) / \alpha_u] a_e u_e^0$ has been incorporated into the source term b , and \tilde{u}_e^0 is called pseudo-velocity. Similarly, for the v component we have

$$v_n = \frac{\sum a_{nb} v_{nb}^0 + b}{a_n / \alpha_v} + d_n(p_P^* - p_N^*) = \tilde{v}_n^0 + d_n(p_P^* - p_N^*) \quad (20)$$

If formulas such as Eqs. (19) and (20) for u and v are substituted into Eq. (16), an equation for the pressure p^* can be derived. It is very similar to the pressure-correction

equation and can be written as

$$a_P p_P^* = \sum a_{nb} p_{nb}^* + b \quad (21)$$

where

$$a_P = a_E + a_W + a_N + a_S \quad (22a)$$

$$a_E = (\rho A d)_e \quad a_W = (\rho A d)_w \quad a_N = (\rho A d)_n \quad a_S = (\rho A d)_s \quad (22b)$$

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

In the CSIMPLER [9] and CLEAR [10] algorithms, the same method is adopted to overcome the first approximation in the SIMPLE.

The initial pressure p^* calculated from Eq. (21), and the velocity calculated from Eqs. (19) and (20) satisfies only the explicit momentum equations (19) and (20) and the continuity equation (16). There may be a large difference between the initial pressure p^* and the accurate pressure of the current iteration level. In the SIMPLER algorithm the initial pressure p^* solved by Eq. (21) is regarded as the final result of the current iteration level. This also leads to some slow convergence rates and instability in the iteration process, especially for complex, high-Re/Ra or fine-mesh flow cases. And it is here that some room is left for further improvement.

3.2. The Algorithms Overcoming the Second Approximation in the SIMPLE Algorithm

In 1984, Van Doormaal and Raithby proposed the SIMPLEC algorithm [11], in which, by changing the definition of the coefficients of the pressure-correction equation, the effects of dropping the neighboring-grid velocity corrections (the second approximation in the SIMPLE algorithm) is partially compensated.

The main feature of the SIMPLEC algorithm is now reviewed as follows.

For convenience of discussion, Eq. (13) is copied below:

$$\frac{a_e}{\alpha_u} u'_e = \sum a_{nb} u'_{nb} + A_e (p'_P - p'_E) \quad (23)$$

The term $\sum a_{nb} u'_e$ is subtracted from both sides of Eq. (23). This yields

$$\left(\frac{a_e}{\alpha_u} - \sum a_{nb} \right) u'_e = \sum a_{nb} (u'_{nb} - u'_e) + A_e (p'_P - p'_E) \quad (24)$$

It is expected that the velocity corrections around position e will be of the same order as u'_e , so the term $\sum a_{nb} (u'_{nb} - u'_e)$ on the right-hand side of Eq. (24) can be neglected compared with the second term on the right-hand side of Eq. (24). Then Eq. (24)

becomes

$$u'_e = d_e(p'_P - p'_E) \quad (25)$$

where

$$d_e = \frac{A_e}{\left(\frac{a_e}{\alpha_u} - \Sigma a_{nb}\right)} \quad (26)$$

Similarly, for the v component, we have

$$v'_n = d_n(p'_P - p'_N) \quad (27)$$

where

$$d_n = \frac{A_n}{\left(\frac{a_n}{\alpha_v} - \Sigma a_{nb}\right)} \quad (28)$$

The solution steps of SIMPLEC are mainly the same as those of SIMPLE with d_e, d_n determined by Eqs. (26) and (28), respectively.

Van Doormaal and Raithby also proposed the SIMPLEX algorithm [12, 13] in 1985. In the SIMPLEX algorithm, by solving a set of algebraic equations for the coefficient d in the velocity-correction equation, the effects of dropping the velocity corrections of the neighboring grids are also taken into account to some degree.

Neither SIMPLEC nor SIMPLEX overcomes the second approximation in the SIMPLE completely. Hence the improvement in the completeness of these two algorithms is somewhat limited, and often problem-dependent. The task of enhancing the convergence rate and stability of the iterative solution procedure, especially for the complex, high-Re/Ra or fine-mesh flow cases, still remains.

The PISO algorithm [14] was proposed by Issa in 1985 and implements two correction steps of pressure correction. This makes some improvement in the completeness of the pressure-correction equation of the current iteration level over that obtained by the single correction step. In the FIMOSE algorithm proposed in [15], at one iteration level the momentum and pressure equations are solved iteratively to reduce the effects of the second approximation in the SIMPLE algorithm.

Both the PISO and FIMOSE algorithms adopt inner multistep iterations to overcome the second approximation in the SIMPLE. However, in these two algorithms the first approximation still exists, leading to some inconsistency between the initial pressure and velocity. This initial inconsistency affects the subsequent solutions, leading to some slow convergence rates and instability of the iteration process, especially for the complex, high-Re/Ra or fine-mesh flow cases.

Apart from the above-mentioned algorithms, many other SIMPLE variants have been proposed in order to overcome one or both of the approximations in the SIMPLE algorithm. Connell and Stow [16] proposed two variants of the pressure-correction process. Chatwani and Turan [17] proposed a pressure-velocity coupling algorithm to determine the underrelaxation factor in the pressure-correction equation based on minimization of the global mass residual norm. Yen and Liu [18] proposed the explicit correction step method to accelerate the convergence by making the velocity explicitly satisfy the momentum equation. Yu, Ozoe, and Tao

[19] modified the SIMPLER algorithm by artificially changing the underrelaxation term to match the variable to be solved, and the revised method was called MSIMPLER.

In summary, more than 10 variants of the SIMPLE algorithm are available in the literature, but no one has successfully overcome the two assumptions. And it can be observed that once the pressure-correction term, p' , is introduced to improve the pressure by adding the correction term to the original one, it will inevitably lead to dropping the neighboring grid points in the velocity-correction equation in order to make the pressure-correction equation manageable. Thus, in order to overcome the second approximation of SIMPLE, we should directly improve the pressure, rather than introduce a pressure-correction term. This is the successful point of the algorithm CLEAR [10]. However, as indicated above, the CLEAR algorithm suffers the disadvantage of less robustness. This situation stimulated the present authors to develop a new algorithm in which the improvement of pressure is conducted directly, without any deterioration of the robustness. The new algorithm is now described as follows.

3.3. Features of an Ideal Algorithm

From the discussion above, it can be seen that all the existing pressure-correction algorithms can only partially overcome one or both of the two approximations in the SIMPLE algorithm. In this sense they are not ideal.

An ideal algorithm should simultaneously possess two basic features as follows.

1. The initial pressure field should be as close to the final result of the current iteration level as possible, which can be realized by overcoming the first approximation in a much deeper manner. If the initial pressure is as close to its final result as possible, the intermediate velocity, calculated from the momentum equations by the initial pressure, will deviate little from the conservation of mass. This consistency will greatly help to enhance the convergence rate and stability of the iterative procedure.
2. The final velocity and pressure of each iteration level should satisfy both the momentum and continuity equations as fully as possible, which can be realized by overcoming the second approximation in a more complete manner. That the conservation of mass and momentum is simultaneously satisfied at each iteration level is another ingredient that enhances the stability of the solving process and the convergence rate.

The IDEAL algorithm proposed by the present authors is an ideal algorithm which possesses the features mentioned above. It can almost completely overcome the two approximations in the SIMPLE algorithm. In the following, the IDEAL algorithm is presented in detail.

4. FORMULATION AND SOLUTION PROCEDURE OF THE IDEAL ALGORITHM

Figure 2 shows the framework of the iteration process of the IDEAL algorithm in detail. In the IDEAL algorithm, at each iteration level there exist two inner

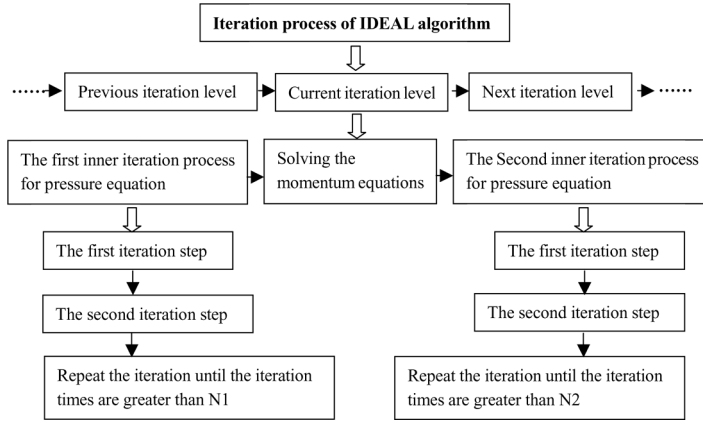


Figure 2. Framework of the iteration process of the IDEAL algorithm.

iteration processes, or inner doubly iterative processes, for pressure-field solution. The first inner iteration process for the pressure equation almost completely overcomes the first approximation in the SIMPLE algorithm. The second inner iteration process almost completely overcomes the second approximation in the SIMPLE algorithm.

4.1. The First Inner Iteration Process

The aim of the first inner iteration process for the pressure equation is to get an initial pressure which is close to the final result of the current iteration level as possible.

For convenience of discussion, the equations mentioned in the SIMPLER algorithm will be copied below, while the superscripts will be changed. At the first iteration step of the first inner iteration process, the initial velocity u^0, v^0 is used to calculate the coefficients of the discretized momentum equation and the pseudo-velocity, denoted by \tilde{u}^0, \tilde{v}^0 . Thus the discretized momentum equations can be expressed as follows:

$$u_e^{\text{Temp}} = \tilde{u}_e^0 + d_e(p_P^{\text{Temp}} - p_E^{\text{Temp}}) \quad (29a)$$

$$v_n^{\text{Temp}} = \tilde{v}_n^0 + d_n(p_P^{\text{Temp}} - p_N^{\text{Temp}}) \quad (29b)$$

where

$$\tilde{u}_e^0 = \frac{\sum a_{nb} u_{nb}^0 + b}{a_e / \alpha_u} \quad (30a)$$

$$\tilde{v}_n^0 = \frac{\sum a_{nb} v_{nb}^0 + b}{a_n / \alpha_v} \quad (30b)$$

In Eqs. (29a) and (29b), p^{Temp} , u^{Temp} , and v^{Temp} stand for the temporary pressure and velocity. Again substituting Eqs. (29a) and (29b) into the continuity equation (16), we obtain the pressure equation as

$$a_P p_P^{\text{Temp}} = \sum a_{nb} p_{nb}^{\text{Temp}} + b \quad (31)$$

where

$$a_P = a_E + a_W + a_N + a_S \quad (32a)$$

$$a_E = (\rho A d)_e \quad a_W = (\rho A d)_w \quad a_N = (\rho A d)_n \quad a_S = (\rho A d)_s \quad (32b)$$

$$b = (\rho \tilde{u}^0 A)_w - (\rho \tilde{u}^0 A)_e + (\rho \tilde{v}^0 A)_s - (\rho \tilde{v}^0 A)_n \quad (32c)$$

By solving the pressure equation (31), we can get the temporary pressure field p^{Temp} . Substituting the temporary pressure p^{Temp} into Eqs. (29a) and (29b), the temporary velocity u^{Temp} , v^{Temp} can be obtained. Then the first iteration step is over. It can be found that the temporary pressure and velocity satisfy only the explicit momentum equations (29a) and (29b) and the continuity equation (16). By explicit momentum equation we mean that the velocities of the neighboring grids in the pseudo-velocity, Eqs. (30a) and (30b), are obtained from the initial velocity. Hence, there may be a large difference between the temporary pressure and the final pressure. To reduce this difference, the next iteration step should be conducted.

In the second iteration step, the velocities of the neighboring grids in the pseudo-velocity, Eqs. (30a) and (30b), are replaced with the temporary velocity calculated by the previous iteration step, denoted by $u^{P\text{Temp}}$, $v^{P\text{Temp}}$. The coefficients of the discretized momentum equation remain the same as those used in the previous iteration step. Thus the pseudo-velocity can be expressed as follows:

$$\tilde{u}_e^{P\text{Temp}} = \frac{\sum a_{nb} u_{nb}^{P\text{Temp}} + b}{a_e / \alpha_u} \quad (33a)$$

$$\tilde{v}_n^{P\text{Temp}} = \frac{\sum a_{nb} v_{nb}^{P\text{Temp}} + b}{a_n / \alpha_v} \quad (33b)$$

By the updated pseudo-velocity $\tilde{u}_e^{P\text{Temp}}$, $\tilde{v}_n^{P\text{Temp}}$, the corresponding pressure and velocity of the current iteration step can be obtained. Then the second iteration step is over. In the second iteration step the pseudo-velocity is updated by the temporary velocity of the previous iteration step, thus the change of the velocities of the neighboring grids in the pseudo-velocity is considered. Hence, the temporary pressure of the second iteration step is closer to the final pressure than that of the first iteration step.

Repeat the above iterative calculation until the iteration times are greater than the prespecified times, denoted by N1. Once the first inner iteration process ends, the difference between the temporary pressure and the final pressure is quite small. Then this temporary pressure is regarded as the initial pressure p^* . Thus the intermediate

velocity u^* , v^* can be calculated from Eqs. (8a) and (8b) by the initial velocity u^0, v^0 and the initial pressure p^* calculated by the first inner iteration process.

4.2. The Second Inner Iteration Process

The aim of the second inner iteration process for the pressure equation is to get the final velocity and pressure of the current iteration level, which almost fully satisfy both the momentum and continuity equations at the iteration level.

The second inner iteration process for the pressure equation is the same as the first inner iteration process except that in the first iteration step of the second inner iteration process, the pseudo-velocity is calculated by the intermediate velocity u^* , v^* rather than the initial velocity u^0, v^0 . That is, when equations similar to Eqs. (30a) and (30b) are used to determine the pseudo-velocity, in the right term of Eqs. (30a) and (30b) u^0, v^0 should be replaced by u^*, v^* . In the process of the second inner iteration, the pseudo-velocity is updated continuously by the temporary velocity of the previous iteration step. From the structure of Eqs. (30a) and (30b) it can be clearly observed that the change of the velocities of the neighboring grids is fully taken into account without dropping any term, making the algorithm fully implicit.

Repeat the above iterative calculation until the iteration times are greater than the prespecified times, denoted by N2. Once the second inner iteration process ends, the temporary velocity and pressure mostly fully satisfy both the momentum and continuity equations at the current iteration level; then the temporary velocity and pressure are regarded as the final velocity and pressure of the current iteration level.

It should be noted that the second inner iteration process seems to be a continuation of the first inner iteration process. In fact, these two inner iterations are different in essence. The first inner iteration process for the pressure equation is for getting an initial pressure to solve the momentum equations (8a) and (8b), hence to obtain the intermediate solutions u^*, v^* . After obtaining the intermediate solutions u^*, v^* , the second inner iteration process for the pressure equation starts, which is for improving the coupling between velocity and pressure and to get the final velocity and pressure of the current iteration level. This difference can be noticed from following description of the solution procedure of IDEAL.

4.3. Solution Procedure of the IDEAL Algorithm

The solution procedure of the IDEAL algorithm in one iteration level is now summarized as follows.

- Step 1. Assume an initial velocity field u^0, v^0 .
- Step 2. Calculate the coefficients of the discretized momentum equation and pseudo-velocity \tilde{u}^0, \tilde{v}^0 .
- Step 3. Solve the pressure equation (31) and obtain the temporary pressure p^{Temp} .
- Step 4. Calculate the temporary velocity $u^{\text{Temp}}, v^{\text{Temp}}$ from Eqs. (29a) and (29b).
- Step 5. Update the pseudo-velocity by the temporary velocity obtained from Step 4 and then return to Step 3. Repeat the iteration until the iteration times are greater than the prespecified times N1. Then the temporary pressure

- p^{Temp} of the first inner iteration process is regarded as the initial pressure p^* .
- Step 6. Solve the momentum equations (8a) and (8b) by the initial velocity and pressure and obtain the intermediate velocity u^*, v^* .
- Step 7. Calculate the pseudo-velocity \tilde{u}^*, \tilde{v}^* by the intermediate velocity u^*, v^* .
- Step 8. Solve the pressure equation (31) and obtain the temporary pressure p^{Temp} .
- Step 9. Calculate the temporary velocity $u^{\text{Temp}}, v^{\text{Temp}}$ from Eqs. (29a) and (29b).
- Step 10. Update the pseudo-velocity by the temporary velocity obtained from Step 9 and then return to Step 8. Repeat the iteration until the iteration times are greater than the prespecified times N2. After the second inner iteration process ends, the temporary velocity $u^{\text{Temp}}, v^{\text{Temp}}$ and the temporary pressure p^{Temp} are regarded as the final results of velocity and pressure of the current iteration level.
- Step 11. Regard the final velocity u, v as the initial velocity u^0, v^0 of the next iteration level, then return to Step 2.

4.4. Analysis of the IDEAL Algorithm

As indicated above, the IDEAL algorithm is brought forward on the basis of the CLEAR algorithm [10], which was proposed by the authors' group in 2004. In the CLEAR algorithm, at each iteration level the pressure equation is also solved in two separate steps. The first step for the pressure solution is to overcome the first approximation in SIMPLE; the second step is to compensate for the second approximation in SIMPLE. However, in the first step of the CLEAR algorithm the velocities of the neighboring grids in the pseudo-velocity take the values of the previous iteration and the pressure equation is solved only once; similarly, in the second step of the CLEAR algorithm, from the immediate velocity u^*, v^* the improvement in velocity is conducted only once. Therefore, the updated pressure and velocity satisfy only the explicit momentum equation and the continuity equation. In order to improve the consistency of initial fields and the completeness of the discretized pressure equation, the IDEAL algorithm carries out multiple iterations in each step, hence greatly improves the consistency and completeness of the algorithm. It is believed that the final velocity and pressure can simultaneously satisfy both the momentum and continuity equations at each iteration level. Therefore, the IDEAL algorithm is very stable and efficient not only for the simple, low-Re/Ra or coarse-mesh flow cases, but also for the complex, high-Re/Ra or fine-mesh flow cases.

Because the conservation condition of mass and momentum is almost fully satisfied at each iteration level, the velocity and pressure underrelaxation factors may be set to quite large values and need not be adjusted. Considering the efficiency and stability of the iteration process, we set a value of 0.9 for both the velocity and pressure underrelaxation factors ($\alpha_{u,v,p} = 0.9$), and this value works well in our four examples, as will be shown in detail in the companion article [20]. Here the pressure underrelaxation factor α_p is incorporated into the solution process of the discretized pressure equations solved in the inner doubly iterative processes. The large, fixed underrelaxation factors guarantee a rather fast convergence rate for any flow case. However, the inner doubly iterative times N1, N2 need to be adjusted so that the conservation of mass and momentum can be almost fully guaranteed at each

iteration level. In our study, the inner doubly iterative times $N1$ and $N2$ are usually set as 4; for some situations, such as very high-Re/Ra or very fine-mesh flow cases, larger values may be needed. Therefore, in the IDEAL algorithm the convergence and stability of iteration process can be controlled by adjusting the inner doubly iterative times $N1$, $N2$ so that the conservation of mass and momentum at each iteration level can be almost fully guaranteed. This is the essential feature of IDEAL which is different from all previous methods.

It can be observed that in the IDEAL algorithm the pressure equation is solved in multiple manners, usually eight times, at each iteration level, which is much more than in the other algorithms mentioned in this article. Hence, the computation time used at each iteration level in the IDEAL algorithm is appreciably longer than is needed for other algorithm, usually about twice more than the others from the authors' numerical practice. However, the coupling between velocity and pressure in the IDEAL algorithm is much better than that in the other algorithms, leading to a significant improvement in the convergence rate and stability, which greatly decreases the iteration numbers and the total computation time of the iteration process. From the four examples shown in the companion article [20], the saving in total computation time compared to the SIMPLER algorithm is usually about 50%, while the robustness of IDEAL is much better than that of SIMPLER. For example, for the lid-driven cavity flow at $Re = 10,000$ and a grid system of 260×260 , the IDEAL algorithm can obtain a converged solution, while no matter how small the underrelaxation factor is, the SIMPLER algorithm always diverges.

5. CONCLUSIONS

In this article, the IDEAL algorithm has been proposed. The main features of the new algorithm are as follows.

1. In the IDEAL algorithm the inner doubly iterative processes for the pressure equation are used to almost completely overcome the two approximations in the SIMPLE algorithm, thus making the IDEAL algorithm fully implicit.
2. The coupling between velocity and pressure is almost fully guaranteed, greatly enhancing the convergence rate and stability of the iteration process not only for the simple, low-Re/Ra or coarse-mesh flow cases, but also for the complex, high-Re/Ra or fine-mesh flow cases.
3. In the IDEAL algorithm two adjustable parameters, the inner doubly iterative times $N1$, $N2$, are introduced to control the convergence of the iteration process instead of the underrelaxation factors, which can be set as large as 0.9 ($\alpha_{u,v,p} = 0.9$) and need not be adjusted. The large fixed underrelaxation factors guarantee a rather fast convergence rate for any flow case.
4. For those existing codes based on the SIMPLE or SIMPLER algorithm, the incorporation of the IDEAL algorithm is very easy. Thus it is expected that the IDEAL algorithm will be widely adopted in computations of incompressible fluid flow and heat transfer problems.

Extensions to a collocated grid system, to compressible fluid flow, and to multiphase 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, FL, 1998.
2. W. Q. Tao, *Recent Advances in Computational Heat Transfer*, Science Press, Beijing, 2000.
3. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
4. J. B. Bell, P. Colelia, and J. M. Glaz, A Second-Order Projection Method for the Incompressible Navier-Stokes Equations, *J. Comput. Phys.*, vol. 85, pp. 257–283, 1989.
5. S. L. Lee and R. Y. Tzong, Artificial Pressure for Pressure-linked Equation, *Int. J. Heat Mass Transfer*, vol. 35, pp. 2705–2716, 1992.
6. 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.
7. 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.
8. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
9. X. L. Liu, W. Q. Tao, and Y. L. He, A Simple Method for Improving the SIMPLER Algorithm for Numerical Simulations of Incompressible Fluid Flow and Heat Transfer Problems, *Eng. Comput.*, vol. 22, pp. 921–939, 2005.
10. W. Q. Tao, Z. G. Qu, and Y. L. He, A Novel Segregated Algorithm for Incompressible Fluid and Heat Transfer Problems—CLEAR (Coupled and Linked Equations Algorithm Revised) Part I: Mathematical Formulation and Solution Procedure, *Numer. Heat Transfer B*, vol. 45, pp. 1–17, 2004.
11. 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.
12. 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.
13. 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.
14. R. I. Issa, Solution of Implicitly Discretized Fluid Flow Equation by Operator-Splitting, *J. Comput. Phys.*, vol. 62, pp. 40–65, 1985.
15. B. R. Latimer and A. Pollard, Comparison of Pressure-Velocity Coupling Solution Algorithms, *Numer. Heat Transfer*, vol. 8, pp. 635–652, 1985.
16. S. D. Connell and P. Stow, The Pressure Correction Methods, *Comput. Fluid*, vol. 14, pp. 1–10, 1986.
17. 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.
18. R. H. Yen and C. H. Liu, Enhancement of the SIMPLE Algorithm by an Additional Explicit Corrector Step, *Numer. Heat Transfer B*, vol. 24, pp. 127–141, 1993.
19. 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.
20. D. L. Sun, W. Q. Tao, and Z. G. Qu, An Efficient Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems—IDEAL (Inner Doubly Iterative Efficient

- Algorithm for Linked Equations) Part II: Application Examples, *Numer. Heat Transfer B*, vol. 53, pp. 2007.
21. W. Q. Tao, *Numerical Heat Transfer*, 2nd ed., Xi'an Jiaotong University Press, Xi'an, China, pp. 135–183, 2001.
 22. E. Bloesch, W. Shyy, and R. Smith, The Role of Mass Conservation in Pressure-Based Algorithm, *Numer. Heat Transfer B*, vol. 24, pp. 415–429, 1993.