INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2007; **72**:1490–1512
Published online 2 May 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2054

A bridge between projection methods and SIMPLE type methods for incompressible Navier–Stokes equations

Ming-Jiu Ni*, † and Mohamed A. Abdou

Mechanical and Aerospace Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095, U.S.A.

SUMMARY

A bridge is built between projection methods and SIMPLE type methods (Semi-Implicit Method for Pressure-Linked Equation). A general second-order accurate projection method is developed for the simulation of incompressible unsteady flows by employing a non-linear update of pressure term as $\Theta^n \nabla p^{n+1} + (I - \Theta^n)\nabla p^n$, where Θ^n is a coefficient matrix, which may depend on the grid size, time step and even velocity. It includes three- and four-step projection methods. The standard SIMPLE method is written in a concise formula for steady and unsteady flows. It is proven that SIMPLE type methods have second-order temporal accuracy for unsteady flows. The classical second-order projection method and SIMPLE type methods are united within the framework of the general second-order projection formula. Two iteration algorithms of SIMPLE type methods for unsteady flows are described and discussed. In addition, detailed formulae are provided for general projection methods by using the Runge–Kutta technique to update the convective term and Crank–Nicholson scheme for the diffusion term. Copyright © 2007 John Wiley & Sons, Ltd.

Received 9 December 2005; Revised 22 February 2007; Accepted 27 February 2007

KEY WORDS: Navier–Stokes equations; projection method; SIMPLE method; temporal accuracy

1. INTRODUCTION

The dimensionless unsteady incompressible Navier–Stokes equations in primitive variables can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} \tag{1}$$

*Correspondence to: Ming-Jiu Ni, Chinese Academy of Sciences, Beijing 100049, China.
†E-mail: jiuzuiren@gmail.com, mjni@ucla.edu

Contract/grant sponsor: The U.S. Department of Energy; contract/grant number: DE-FG03-86ER52123

and the conservation of mass equation for incompressible flows is given as

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where $\mathbf{u}$, p are the non-dimensional velocity vector and kinetic pressure, respectively, and Re is the Reynolds number. A key issue in the design of numerical methods for Navier–Stokes equations is the development of an appropriate discrete form of the incompressibility constraint. Famous primitive variable numerical methods include the MAC method [1], projection methods [2–13], and the SIMPLE type method (Semi-Implicit Method for Pressure-Linked Equation) [14–18]. PISO [19] is also a SIMPLE type method. They all have been extensively used and have served well. MAC is an explicit transient algorithm, and it has been extensively applied to do DNS (Direct Numerical Simulation) of turbulence [20] and interfacial flows [21]. The projection and SIMPLE methods are implicit or semi-implicit algorithms. Implicit and semi-implicit methods are attractive as a means of avoiding restrictions on the explicit time step. Projection methods have been extensively applied to unsteady flows such as DNS in [5, 22, 23], and interfacial flows in [3, 24, 25], whereas SIMPLE type methods have been successfully applied to steady flows, especially for the numerical calculations of heat transfer. It has also been applied to unsteady flows [26], unsteady interfacial flows [27–29] and even to DNS for turbulence research [30] but without analysis of the temporal accuracy of the method. The analysis of a connection among the above primitive algorithms, especially between the projection methods and SIMPLE methods, is of current interest.

The standard SIMPLE method was developed in [16]. SIMPLE type methods include SIMPLER [15], SIMPLEC [18], PISO [19] and so forth. Recent development of SIMPLE method can be found in [31]. The pioneering work of Chorin [6] introduced a projection method for unsteady incompressible Navier–Stokes equations, in which the diffusion term is updated in a semi-implicit manner and time step size is greatly enlarged as compared to the MAC method. Kim and Moin [9] improved Choin's first-order accurate projection method to second order, which formulation has been extensively applied in DNS and LES (Large Eddy Simulation) simulations for turbulence. Bell *et al.* [2] employed Godunov's scheme for the update of the convective term, which is robust and has been one of the most popular projection methods for the calculation of variable density interfacial flows [3, 25]. Gresho [8] presented a detailed discussion, which contains first-, second- and third-order temporal accurate methods. The boundary conditions for intermediate velocities were also discussed in [8]. Dukowicz and Dvinsky [7] designed a general high-order projection method based on the application of the approximate factorization (AF) technique on the Navier–Stokes system. By employing the block LU decomposition, Perot [11, 12] analysed the solution accuracy of the projection method and the influence of boundary conditions on it. Choi and Moin [5] presented their four-step projection method, in which a Poisson equation for pressure and not pressure difference is solved. Ni and Komori [10] also presented a general second-order projection method. Brown *et al.* [4] developed a fully second-order projection method, where the pressure term is computed to second-order accuracy. However, the analysis in all of the above projection methods was based on a constant coefficient update of the pressure term, and the connection between projection methods and SIMPLE type methods was not demonstrated.

This paper presents a general second-order projection method for the incompressible Navier–Stokes equation in Section 2. The formulation includes three- and four-step projection methods. In Section 3, a concise formula is given for SIMPLE type methods, and the temporal accuracy analysis is done for the same. A bridge between the projection and SIMPLE methods is built up. The detailed formulae for the three- and four-step projection methods are given in Section 4. These

DOI: 10.1002/nme

are based on the update of the convective term using a Runge–Kutta technique and of the diffusion term using the Crank–Nicholson scheme. Two iteration methods of SIMPLE type algorithms are discussed for the calculation of unsteady flows, and high-order SIMPLE methods are also listed in this section. Section 5 gives numerical validation. The conclusion is given in Section 6.

2. GENERAL SECOND-ORDER PROJECTION FORMULAE

The straightforward discretization of Equations (1) and (2) can be written as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}+\Theta^n\nabla p^{n+1}=\frac{1}{Re}(\nabla^2\mathbf{u})^{n+1/2}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1/2}-(\mathbf{I}-\Theta^n)\nabla p^n \tag{3}$$

$$\nabla\cdot\mathbf{u}^{n+1}=0 \tag{4}$$

Here, backward-Euler technique is employed for the update of the time derivative term. $\mathbf{u}$ and p are the unknown discrete velocity vector and pressure, $(1/Re)(\nabla^2\mathbf{u})^{n+1/2}$ and $(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1/2}$ are the temporal update of the diffusion and convective terms, respectively. The pressure term is updated by $\Theta^n\nabla p^{n+1}+(I-\Theta^n)\nabla p^n$. $\Theta^n=\mathrm{DIAG}(\theta_x^n,\theta_y^n,\theta_z^n)$ is a diagonal coefficient matrix with the diagonal elements $\theta_x^n,\theta_y^n,\theta_z^n\in[0,1]$, which may depend on the grid size, time step and even velocity as shown later. We obtain the trapezoidal update of the pressure term when $\Theta^n=\frac{1}{2}\mathbf{I}$, and the fully implicit update when $\Theta^n=\mathbf{I}$. Here $\mathbf{I}$ is the unit identity matrix operator. The solution of Equations (3) and (4) is not easy to obtain because it involves simultaneous solution for the velocity and pressure. Projection methods first solve the convection–diffusion equation to predict the intermediate velocity, which is then projected onto the space of divergence-free field. The discretized equations (3) and (4) can be written as the matrix form following [7, 11]:

$$\begin{pmatrix}\mathbf{A} & \Theta^n\mathbf{G}\\ \mathbf{D} & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}^{n+1}\\ p^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{r}-(\mathbf{I}-\Theta^n)\mathbf{G}p^n\\ 0\end{pmatrix} \tag{5}$$

here $\mathbf{A}$, $\mathbf{G}$ and $\mathbf{D}$ are the sub-matrices, the right-hand-side $\mathbf{r}$ vector contains all those quantities that are already known. Considering the explicit update of the convective term for simplicity and semi-implicit Crank–Nicholson update of the diffusion term for stability, the discretized Navier–Stokes equation (3) and the continuity equation (4) can be written in matrix form (5) with

$$\mathbf{A}=\frac{1}{\Delta t}\left(\mathbf{I}-\frac{\Delta t}{2Re}\mathbf{L}\right) \tag{6}$$

$$\mathbf{r}=\frac{1}{\Delta t}\left(\mathbf{I}+\frac{\Delta t}{2Re}\mathbf{L}\right)\mathbf{u}^n-\mathbf{N}^{n+1/2}(\mathbf{u}) \tag{7}$$

where $\mathbf{N}^{n+1/2}(\mathbf{u})$ is the discrete convective operator, $\mathbf{L}$ is the discrete Laplacian operator, $\mathbf{D}$ is the discrete divergence operator, and $\mathbf{G}$ is the discrete gradient operator.

DOI: 10.1002/nme

The block matrix form of Equation (5) can be factored as

$$\begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{D} & -\mathbf{D}\mathbf{A}^{-1}\Theta^n\mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\Theta^n\mathbf{G} \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} - (\mathbf{I} - \Theta^n)\mathbf{G}p^n \\ 0 \end{pmatrix} \tag{8}$$

The above equation can be reduced to the following set of equations:

$$\mathbf{A}\tilde{\mathbf{u}} = \mathbf{r} - (\mathbf{I} - \Theta^n)\mathbf{G}p^n \tag{9}$$

$$\mathbf{D}\mathbf{A}^{-1}\Theta^n\mathbf{G}p^{n+1} = \mathbf{D}\tilde{\mathbf{u}} \tag{10}$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \mathbf{A}^{-1}\Theta^n\mathbf{G}p^{n+1} \tag{11}$$

which is referred to as the Uzawa method [13]. However, the Uzawa method is computationally expensive since matrix $\mathbf{A}$ must effectively be inverted for every iteration of the discrete pressure Poisson equation. The projection method in [6] approximates Equation (5) and significantly reduces the computational complexity by assuming that $\mathbf{A}^{-1} = \Delta t$ as

$$\begin{pmatrix} \mathbf{A} & \Delta t\mathbf{A}\Theta^n\mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} - (\mathbf{I} - \Theta^n)\mathbf{G}p^n \\ 0 \end{pmatrix} \tag{12}$$

The pressure gradient terms in the momentum equations have been altered. The approximation has only first-order temporal accuracy with an error term as $\Delta t(\mathbf{I}/\Delta t - \mathbf{A})\Theta^n\mathbf{G}p^{n+1}$ since $\mathbf{I}/\Delta t - \mathbf{A}$ is a quantity independent of the time step. Different update techniques for the convective and diffusion terms will produce different formulations of $\mathbf{A}$ and $\mathbf{r}$. For example, when the Crank–Nicholson and explicit techniques are employed for the update of the diffusion term and the convective term, respectively, we have formulations (6) and (7) of $\mathbf{A}$ and $\mathbf{r}$, and the error term is $(\Delta t/2Re)\mathbf{L}(\Theta^n\mathbf{G})p^{n+1}$.

According to the error analysis result of the first-order approximation, here we consider the following approximation to system of Equation (12) as

$$\begin{pmatrix} \mathbf{A} & \Delta t\mathbf{A}\Theta^n\mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} - (\mathbf{I} - \Theta^n)\mathbf{G}p^n - (\mathbf{I} - \Delta t\mathbf{A})\Theta^n\mathbf{G}p^n \\ 0 \end{pmatrix} \tag{13}$$

where the right-hand-side vector has been augmented by an extra term of $(\mathbf{I} - \Delta t\mathbf{A})\Theta^n\mathbf{G}p^n$. The above approximation can be further written as

$$\begin{pmatrix} \mathbf{A} & \Delta t\mathbf{A}\Theta^n\mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} + (\Delta t\mathbf{A}\Theta^n - \mathbf{I})\mathbf{G}p^n \\ 0 \end{pmatrix} \tag{14}$$

Equation (14) is an approximation to system of Equation (5) with an error term of $(\Delta t)^2(\mathbf{I}/\Delta t - \mathbf{A})\Theta^n\mathbf{G}((p^{n+1} - p^n)/\Delta t)$. As we have noted before, $\mathbf{I}/\Delta t - \mathbf{A}$ is a quantity independent of the time step and Equation (14) is a second-order accurate in time approximation to system of Equation (5) when Θ^n is a constant or a variable independent of time step. For example, this error term is $((\Delta t)^2/2Re)\mathbf{L}\Theta^n\mathbf{G}((p^{n+1} - p^n)/\Delta t) = ((\Delta t)^2/2Re)\mathbf{L}\Theta^n\mathbf{G}(\partial p^n/\partial t + o(\Delta t))$ and has second-order temporal accuracy when the diffusion term is updated using the Crank–Nicholson scheme.

DOI: 10.1002/nme

The coefficient matrix of Equation (14) can be factored into the block decomposition as

$$\begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{D} & -\Delta t \mathbf{D}\Theta^n \mathbf{G} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} + (\Delta t \mathbf{A}\Theta^n - \mathbf{I})\mathbf{G}p^n \\ 0 \end{pmatrix} \tag{15}$$

$$\begin{pmatrix} \mathbf{I} & \Delta t \Theta^n \mathbf{G} \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{u}} \\ p^{n+1} \end{pmatrix} \tag{16}$$

Equations (15) and (16) can be written in the series of operation as

$$\mathbf{A}\tilde{\mathbf{u}} = \mathbf{r} + (\Delta t \mathbf{A}\Theta^n - \mathbf{I})\mathbf{G}p^n \tag{17}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G} p^{n+1} = \mathbf{D}\tilde{\mathbf{u}} \tag{18}$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t \Theta^n \mathbf{G} p^{n+1} \tag{19}$$

The projection method has second-order temporal accuracy, and Equation (17) can be reformulated as

$$\mathbf{A}(\tilde{\mathbf{u}} - \Delta t \Theta^n \mathbf{G} p^n) = \mathbf{r} - \mathbf{G}p^n \tag{20}$$

Let $\hat{\mathbf{u}} = \tilde{\mathbf{u}} - \Delta t \Theta^n \mathbf{G} p^n$, the general four-step projection method for incompressible flows can be acquired as

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{r} - \mathbf{G}p^n \tag{21}$$

$$\tilde{\mathbf{u}} = \hat{\mathbf{u}} + \Delta t \Theta^n \mathbf{G} p^n \tag{22}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G} p^{n+1} = \mathbf{D}\tilde{\mathbf{u}} \tag{23}$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t \Theta^n \mathbf{G} p^{n+1} \tag{24}$$

Equations of (21)–(24) can also be formulated as a three-step projection step method as

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{r} - \mathbf{G}p^n \tag{25}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G}(p^{n+1} - p^n) = \mathbf{D}\hat{\mathbf{u}} \tag{26}$$

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \Delta t \Theta^n \mathbf{G}(p^{n+1} - p^n) \tag{27}$$

The general four-step projection method, described in Equations (21)–(24) will form the Choi and Moin method [5] with $\Theta^n = \mathbf{I}$. Choi and Moin's method is the first four-step projection method. In [5], the semi-implicit second-order Crank–Nicholson scheme is employed for the update of both convective and diffusion terms to improve the stability. It should be noted that the implicit update of the convective term does not favour the implementation of higher order spatial schemes. A higher order spatial scheme can improve the resolution of small vortex in DNS [22]. The second-order central difference scheme is utilized to conduct the spatial discretization of the convective term in [5].

DOI: 10.1002/nme

The second-order accurate Dukowicz and Dvinsky method [7] is often employed to do the DNS of turbulent flows. It requires solution of a Poisson equation for pressure difference. In fact, the general three-step projection method of Equations (25)–(27) with $\Theta^n = \frac{1}{2}\mathbf{I}$ forms the method in [7]. The original second-order method of [7] was designed for solving the Stokes flow, in which the Crank–Nicholson scheme was employed to update the diffusion term with good stability.

Bell *et al.* [2] developed a projection method, in which the Crank–Nicholson scheme is employed for update of the diffusion term and the Godunov's scheme for update of the convective term. This projection method is robust, and the variable density version of this projection method [3] has been successfully applied to the simulation of unsteady interfacial flows incorporating the volume of fluid (VOF) [25] and the level set [32] approaches. This projection method is also a special case of the general three-step projection method with $\Theta^n = \mathbf{I}$ in Equations (25)–(27).

Traditionally, Θ^n is a constant for projection methods, such as the projection methods in [2, 5, 7]. Θ^n may not be a constant and may depend on time step, spatial size and even velocity. When the variable Θ^n is not a constant, the accuracy can be analysed as follows:

$$\left|(\Delta t)^2\left(\frac{\mathbf{I}}{\Delta t}-\mathbf{A}\right)\Theta^n\mathbf{G}\left(\frac{p^{n+1}-p^n}{\Delta t}\right)\right| \leqslant \left|(\Delta t)^2\left(\frac{\mathbf{I}}{\Delta t}-\mathbf{A}\right)\mathbf{G}\left(\frac{p^{n+1}-p^n}{\Delta t}\right)\right||\Theta^n| \underbrace{\leqslant}_{\theta_x,\theta_y,\theta_z\in(0,1]} \left|(\Delta t)^2\left(\frac{\mathbf{I}}{\Delta t}-\mathbf{A}\right)\mathbf{G}\left(\frac{p^{n+1}-p^n}{\Delta t}\right)\right| \quad (28)$$

It is second order since $\mathbf{I}/\Delta t - \mathbf{A}$ is a quantity independent of the time step. This means that the temporal accuracy of the general formula for four-step projection method (Equations (20)–(24)) and the three-step projection method (Equations (25)–(27)), is second order. This holds true irrespective of Θ^n being a constant matrix or a variable.

To our knowledge, a projection method with a non-linear coefficient matrix Θ^n has not been developed. In the next section, it will be demonstrated that SIMPLE type methods can be regarded as a special case of the general projection methods. The standard SIMPLE method has a time step, grid size and even velocity-dependent coefficient matrix, which can be an example of the general projection method with a non-linear coefficient matrix.

The projection methods in [2, 9] are also analysed in [4]. It has been shown that velocity of this projection method has second-order accuracy, while the pressure is only first-order accurate. A modified version of BCG (Bell, Colela and Glaz) projection method has been presented in [4] by utilizing a correct pressure update. This update can also be employed for the general four-step and three-step projection methods to improve the accuracy of pressure. For the three-step general projection method, we have the following fully second-order accurate formulation as

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{r} - \mathbf{G}p^n \quad (29)$$

$$\Delta t\mathbf{D}\Theta^n\mathbf{G}(\Phi^{n+1}) = \mathbf{D}\hat{\mathbf{u}} \quad (30)$$

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \Delta t\Theta^n\mathbf{G}(\Phi^{n+1}) \quad (31)$$

$$p^{n+1} = p^n + \Delta t\mathbf{A}\Phi^{n+1} \quad (32)$$

DOI: 10.1002/nme

For the four-step general projection method, we have the following fully second-order accurate formulation:

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{r} - \mathbf{G}p^n \tag{33}$$

$$\tilde{\mathbf{u}} = \hat{\mathbf{u}} + \Delta t \Theta^n \mathbf{G}p^n \tag{34}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G}p^* = \mathbf{D}\tilde{\mathbf{u}} \tag{35}$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t \Theta^n \mathbf{G}p^* \tag{36}$$

$$p^{n+1} = p^n + \Delta t \mathbf{A}(p^* - p^n) \tag{37}$$

These modified general multi-step projection methods described in Equations (29)–(32) and (33)–(37) have complete second-order temporal accuracy.

3. CONCISE FORMULA OF SIMPLE METHOD AND ITS RELATIONSHIP WITH PROJECTION METHOD

The fundamental concept of the SIMPLE method is to derive a pressure correction equation by enforcing mass continuity over each cell. Using the notation of Patankar's in [14], assuming a fully implicit technique is employed to update the convective and diffusion terms, a fully discretized momentum equation can be acquired based on Equation (3) with $\Theta^n = 0$ as

$$\left(\frac{1}{\Delta t}\mathbf{I} + \mathbf{A}_P^n\right)\hat{\mathbf{u}}_P = \frac{1}{\Delta t}\mathbf{u}_P^n + \sum_M \mathbf{A}_M^n \hat{\mathbf{u}}_M - \mathbf{G}p^n \tag{38}$$

Here $\mathbf{A}_P^n = \mathrm{DIAG}(A_{PX}^n, A_{PY}^n, A_{PZ}^n)$, $\mathbf{A}_M^n = \mathrm{DIAG}(A_{MX}^n, A_{MY}^n, A_{MZ}^n)$ are diagonal matrices with elements meeting with the conditions of $A_{PX}^n = \sum_M A_{MX}^n$, $A_{PY}^n = \sum_M A_{MY}^n$ and $A_{PZ}^n = \sum_M A_{MZ}^n$. Same as in [14], a linear treatment of the convective terms is applied in the above discretization, which means $\mathbf{N}(\mathbf{u}^{n+1}) = \mathbf{u}^{n+1}\cdot\nabla\mathbf{u}^{n+1} \approx \mathbf{u}^n\cdot\nabla\mathbf{u}^{n+1}$. The A coefficient terms contain the contribution of the convective and diffusion terms, and may depend on grid size and velocity. Index M is a grid identifier referring to all nodes surrounding the pole nodes that are involved in the formulation of the finite difference representation of spatial fluxes. Subscript P denotes the pole node. As an example, in a uniform grid system, we have $\mathbf{A}_P = 6(Reh^2)^{-1}\mathbf{I}$ and $\mathbf{A}_P = ((|u| + |v| + |w|)h^{-1} + 6(Reh^2)^{-1})\mathbf{I}$ when the central difference and first-order upwind difference schemes are employed for the convective term, respectively. Here u, v, w are the velocity variables corresponding to x, y, z co-ordinates, h is the grid size. Since it satisfies the momentum and continuity equations of (3) and (4) with $\Theta^n = \mathbf{I}$, using the same update technique and linear treatment as for Equation (38), we have

$$\left(\frac{1}{\Delta t}\mathbf{I} + \mathbf{A}_P^n\right)\mathbf{u}_P^{n+1} = \frac{1}{\Delta t}\mathbf{u}_P^n + \sum_M \mathbf{A}_M^n \mathbf{u}_M^{n+1} - \mathbf{G}p^{n+1} \tag{39}$$

By subtracting Equation (38) from Equation (39), a velocity difference equation can be acquired:

$$\left(\frac{1}{\Delta t}\mathbf{I} + \mathbf{A}_P^n\right)(\mathbf{u}_P^{n+1} - \hat{\mathbf{u}}_P) = \underline{\sum_M \mathbf{A}_M^n(\mathbf{u}_M^{n+1} - \hat{\mathbf{u}}_M)} - \mathbf{G}(p^{n+1} - p^n) \tag{40}$$

DOI: 10.1002/nme

By neglecting the underlined term in Equation (40), a velocity correction equation can be written as

$$\mathbf{u}_P^{n+1} = \hat{\mathbf{u}}_P - \Delta t(\mathbf{I} + \mathbf{A}_P^n \Delta t)^{-1}\mathbf{G}(p^{n+1} - p^n) \tag{41}$$

Since $\nabla \cdot \mathbf{u}_P^{n+1} = 0$, the pressure correction can be acquired through the solution of the following Poisson equation for pressure difference:

$$\Delta t\mathbf{D}((\mathbf{I} + \mathbf{A}_P^n \Delta t)^{-1}\mathbf{G}(p^{n+1} - p^n)) = \mathbf{D}\hat{\mathbf{u}}_P \tag{42}$$

Here $(\mathbf{I}+\mathbf{A}_P^n \Delta t)^{-1} = \mathrm{DIAG}(1/(1+A_{PX}^n \Delta t), 1/(1+A_{PY}^n \Delta t), 1/(1+A_{PZ}^n \Delta t))$ is a diagonal matrix.

Equations (38), (41) and (42) form the standard SIMPLE method [14, 16]. For the SIMPLE method, the pressure correction from the Poisson equation (42) is utilized to correct the velocity by (41). In the general three-step projection method of Equations (25)–(27), we can obtain the standard SIMPLE method by setting $\Theta^n = (\mathbf{I}+\mathbf{A}_P^n \Delta t)^{-1}$, $\mathbf{r} = \mathbf{u}_P^n/\Delta t$, $\mathbf{A}\hat{\mathbf{u}} = ((1/\Delta t)\mathbf{I}+\mathbf{A}_P^n)\hat{\mathbf{u}}_P - \sum_M \mathbf{A}_M^n \hat{\mathbf{u}}_M$. It has been shown the temporal accuracy of the general projection method of Equations (25)–(27) is second-order irrespective of Θ^n being a constant or a variable, dependent on or independent of the time step. Hence the SIMPLE method with a variable matrix $\Theta^n = (\mathbf{I} + \mathbf{A}_P^n \Delta t)^{-1}$ is also second-order accurate, which can be further shown as follows.

According to the analysis in the above section for the general second-order projection method, the error term of SIMPLE method can be written as

$$(\Delta t)^2 \left(\frac{\mathbf{I}}{\Delta t} - \mathbf{A}\right) \Theta^n \mathbf{G}(\delta p^{n+1}) = -(\Delta t)^2 \left(\mathbf{A}_P^n \Theta_P^n \mathbf{G}_P(\delta p^{n+1}) - \sum_M \mathbf{A}_M^n \Theta_M^n \mathbf{G}_M(\delta p^{n+1})\right) \tag{43}$$

Here $\delta p^{n+1} = (p^{n+1} - p^n)/(\Delta t)$. Since $|\Theta_P^n| = |(\mathbf{I} + \mathbf{A}_P^n \Delta t)^{-1}| \leqslant 1$, $|\Theta_M^n| = |(\mathbf{I} + \mathbf{A}_M^n \Delta t)^{-1}| \leqslant 1$, we have

$$\left|(\Delta t)^2 \left(\frac{\mathbf{I}}{\Delta t} - \mathbf{A}\right) \Theta^n \mathbf{G}(\delta p^{n+1})\right| \leqslant (\Delta t)^2 \left(|\mathbf{A}_P^n||\mathbf{G}_P(\delta p^{n+1})| + \sum_M |\mathbf{A}_M^n||\mathbf{G}_M(\delta p^{n+1})|\right) \tag{44}$$

Apparently, it is second-order accurate.

Approximating $\mathbf{u}_M^{n+1} - \hat{\mathbf{u}}_M$ by $\mathbf{u}_P^{n+1} - \hat{\mathbf{u}}_P$ instead of neglecting the underlined term in Equation (40), and considering the fact of $\mathbf{A}_P^n = \sum_M \mathbf{A}_M^n$, we have the velocity correction equation and pressure difference Poisson equation as

$$\mathbf{u}_P^{n+1} = \hat{\mathbf{u}}_P - \Delta t\mathbf{G}(p^{n+1} - p^n) \tag{45}$$

Since $\nabla \cdot \mathbf{u}_P^{n+1} = 0$, we obtain the pressure difference Poisson equation as

$$\Delta t\mathbf{D}(\mathbf{G}(p^{n+1} - p^n)) = \mathbf{D}\hat{\mathbf{u}}_P \tag{46}$$

Equations (38), (45) and (46) form the traditional three-step projection method with $\Theta^n = \mathbf{I}$ in the general three-step projection formula of Equations (25)–(27). This idea has been employed to form the SIMPLEC method in [18].

DOI: 10.1002/nme

In SIMPLE type methods, underrelaxation is usually employed to solve both the momentum equation and pressure Poisson equation. Employing an underrelaxation technique for the momentum equation of Equation (38) for the standard SIMPLE method, we have

$$\left(\frac{1}{\Delta t}\mathbf{I}+\Lambda^{-1}\mathbf{A}_P^n\right)\hat{\mathbf{u}_P}=\frac{1}{\Delta t}\mathbf{u}_P^n+(\Lambda^{-1}-\mathbf{I})\mathbf{A}_P^n\hat{\mathbf{u}}_P^n+\sum_M\mathbf{A}_M^n\hat{\mathbf{u}}_M-\mathbf{G}p^n \tag{47}$$

$$\Delta t\mathbf{D}((\mathbf{I}+\Delta t\Lambda^{-1}\mathbf{A}_P^n)^{-1}\mathbf{G}(p^{n+1}-p^n))=\mathbf{D}\hat{\mathbf{u}}_P \tag{48}$$

$$\mathbf{u}_P^{n+1}=\hat{\mathbf{u}}_P-\Delta t(\mathbf{I}+\Delta t\Lambda^{-1}\mathbf{A}_P^n)^{-1}\mathbf{G}(p^{n+1}-p^n) \tag{49}$$

Here the diagonal matrix of $\Lambda=\mathrm{DIAG}(\lambda_x,\lambda_y,\lambda_z)$ is the underrelaxation factor matrix. $\lambda_x,\lambda_y,\lambda_z$ are the underrelaxation factors for the momentum equations in x, y, z co-ordinates, respectively, which can be used to improve the solution stability of the momentum equation (47). In [33], the solution stability criterion has been acquired for single and multi-dimensional steady convective diffusion equations using the underrelaxation technique. One important point has to be considered while using the underrelaxation method for the calculation of unsteady flows. The variable $\hat{\mathbf{u}}_P^n$ introduced in Equation (47) is not equal to $\mathbf{u}_P^n$ and is also different with $\hat{\mathbf{u}}_P$. $\mathbf{u}_P^n$ represents the velocity at previous time step, which is divergence free. While $\hat{\mathbf{u}}_P$ represents the intermediate velocity used to get the pressure difference by solving Equation (48). This intermediate velocity is further used to get the corrected velocity using Equation (49). $\hat{\mathbf{u}}_P^n$ can be named as sub-intermediate velocity. When the convergent solution of momentum equation at the predictor step of Equation (47) is obtained, we should have $\hat{\mathbf{u}}_P=\hat{\mathbf{u}}_P^n$. However before the attainment of a convergent solution at the predictor step, the sub-intermediate velocity is different with the intermediate velocity. For unsteady flows, both the sub-intermediate velocity $\hat{\mathbf{u}}_P^n$ and the intermediate velocity $\hat{\mathbf{u}}_P$ are not equal to $\mathbf{u}_P^n$. Also, they are not divergence free. For steady flow, when convergent solution is acquired, the three velocities should be same and divergence free.

Approximating $\mathbf{u}_M^{n+1}-\hat{\mathbf{u}}_M$ using $\mathbf{u}_P^{n+1}-\hat{\mathbf{u}}_P$ in a velocity difference equation, the SIMPLEC method can be acquired based on the above formula as

$$\left(\frac{1}{\Delta t}\mathbf{I}+\Lambda^{-1}\mathbf{A}_P^n\right)\hat{\mathbf{u}}_P=\frac{1}{\Delta t}\mathbf{u}_P^n+(\Lambda^{-1}-\mathbf{I})\mathbf{A}_P^n\hat{\mathbf{u}}_P^n+\sum_M\mathbf{A}_M^n\hat{\mathbf{u}}_M-\mathbf{G}p^n \tag{50}$$

$$\Delta t\mathbf{D}((\mathbf{I}+\Delta t(\Lambda^{-1}-\mathbf{I})\mathbf{A}_P^n)^{-1}\mathbf{G}(p^{n+1}-p^n))=\mathbf{D}\hat{\mathbf{u}}_P \tag{51}$$

$$\mathbf{u}_P^{n+1}=\hat{\mathbf{u}}_P-\Delta t(\mathbf{I}+\Delta t(\Lambda^{-1}-\mathbf{I})\mathbf{A}_P^n)^{-1}\mathbf{G}(p^{n+1}-p^n) \tag{52}$$

When $\Lambda=\mathbf{I}$, the SIMPLEC method of Equations (50)–(52) is in fact a traditional three-step projection method with $\Theta^n=\mathbf{I}$ in Equations (25)–(27). Δt is a time step for the calculation of unsteady flows, Δt term can also be regarded as an underrelaxation term for steady flows calculation. For steady flow, we do not need to emphasize the difference between the velocities of $\hat{\mathbf{u}}_P^n$, $\hat{\mathbf{u}}_P$ and $\mathbf{u}_P^n$.

The SIMPLE type methods can also be written as a four-step fractional step formula. For example, we can write the standard three-step SIMPLE method of (38), (41) and (42) as the

DOI: 10.1002/nme

following four-step algorithm:

$$\left(\frac{1}{\Delta t}\mathbf{I}+\mathbf{A}_P^n\right)\hat{\mathbf{u}}_P=\frac{1}{\Delta t}\mathbf{u}_P^n+\sum_M \mathbf{A}_M^n\hat{\mathbf{u}}_M-\mathbf{G}p^n \tag{53}$$

$$\tilde{\mathbf{u}}_P=\hat{\mathbf{u}}_P+\Delta t(\mathbf{I}+\mathbf{A}_P^n\Delta t)^{-1}\mathbf{G}p^n \tag{54}$$

$$\mathbf{u}_P^{n+1}=\tilde{\mathbf{u}}_P-\Delta t(\mathbf{I}+\mathbf{A}_P^n\Delta t)^{-1}\mathbf{G}p^{n+1} \tag{55}$$

$$\Delta t\mathbf{D}((\mathbf{I}+\mathbf{A}_P^n\Delta t)^{-1}\mathbf{G}p^{n+1})=\mathbf{D}\tilde{\mathbf{u}}_P \tag{56}$$

It has second-order temporal accuracy if the temporal term is updated using a second-order temporal scheme. CLEAR (Coupled and Linked Equation Algorithms Revised) [17] can be regarded as a special case of the four-step SIMPLE method of Equations (53)–(56), which has been successfully applied for simulations of steady flows.

The modification of Brown, Cortez and Minion for the general second-order projection method can also be employed to improve the accuracy of pressure in SIMPLE type methods, in the calculation of unsteady flows.

In this section, it has been demonstrated that SIMPLE type methods can be regarded as the projection methods with non-linear coefficient matrices, which are dependent on time step, grid size and velocities. SIMPLE type methods have second-order temporal accuracy. In other words, the bridge between projection methods and SIMPLE type methods has been built up based on the general second-order projection methods presented in Section 2.

4. TEMPORAL UPDATE OF THE PROJECTION METHODS AND SIMPLE METHODS

4.1. For the general projection methods

A simple Fourier stability analysis of the Adams–Bashforth method for the convective term as applied to the 1-D linear hyperbolic equation shows that it is unstable for all CFL numbers. The implicit techniques for the convective term in Navier–Stokes equations will make it difficult to solve the discretized algorithm equations, especially for the higher-order spatial discretization schemes. The Runge–Kutta method seems more suitable for the convective term because of their stability and simplicity. Rai and Moin [23] employed an explicit low-storage Runge–Kutta method, which has the additional advantage that it requires the minimum amount of computer run-time memory, for the convective term and the Crank–Nicholson technique for the diffusion term. The overall accuracy of the method is second-order in time. The explicit third-order temporal accurate Runge–Kutta procedure, coupled with an explicit high-order spatial accurate upwind bias difference scheme for the convective term, has demonstrated the highly desirable characteristic of strong numerical stability [23].

Now we use the Runge–Kutta technique to update the convective term and the Crank–Nicholson scheme to update the diffusion term based on the general second-order formula of the projection methods (21)–(24) and (25)–(27). The detailed three-stage four-step projection method can be

DOI: 10.1002/nme

expressed as

$$\mathbf{A}^m(\hat{\mathbf{u}}^m - \mathbf{u}^{m-1}) = \frac{2\gamma^m}{Re}\mathbf{L}(\mathbf{u}^{m-1}) + \mathbf{r}_N^m - \mathbf{G}(\alpha^m p^{m-1} + \beta^m p^{m-2}) \tag{57}$$

$$\tilde{\mathbf{u}}^m = \hat{\mathbf{u}}^m + \Delta t \Theta^n \mathbf{G}(\alpha^m p^{m-1} + \beta^m p^{m-2}) \tag{58}$$

$$\mathbf{u}^m = \tilde{\mathbf{u}}^m - \Delta t \Theta^n \mathbf{G}(\alpha^m p^m + \beta^m p^{m-1}) \tag{59}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G}(\alpha^m p^m) = \mathbf{D}\tilde{\mathbf{u}}^m - \Delta t \mathbf{D}\Theta^n \mathbf{G}(\beta^m p^{m-1}) \tag{60}$$

where $\alpha^m = \langle 8/15, 5/12, 3/4\rangle$, $\beta^m = \langle 0, 17/60, -5/12\rangle$, $\gamma^m = \langle 4/15, 1/15, 1/6\rangle$ and $\mathbf{A}^m = \frac{1}{\Delta t}[\mathbf{I} - (\Delta t \gamma^m / Re)\mathbf{L}]$, $\mathbf{r}_N^m = -\alpha^m \mathbf{N}(\mathbf{u}^{m-1}) - \beta^m \mathbf{N}(\mathbf{u}^{m-2})$. $\mathbf{N}(\mathbf{u})$ represents the discretization formula of the non-linear term. The velocity components and pressure in the intermediate velocities equation at the first sub-stage are $\mathbf{u}^{-1} = 0$, $p^{-1} = 0$ $(m-2) = -1$ and $\mathbf{u}^0 = \mathbf{u}^n$, $p^0 = p^n$ $(m-1) = 0$. At the third stage $\mathbf{u}^3 = \mathbf{u}^{n+1}$, $p^3 = p^{n+1}$ which are the updated velocities and pressure for the next time level.

And the detailed three-stage three-step projection method can be given as

$$\mathbf{A}^m(\hat{\mathbf{u}}^m - \mathbf{u}^{m-1}) = \frac{2\gamma^m}{Re}\mathbf{L}(\mathbf{u}^{m-1}) + \mathbf{r}_N^m - \mathbf{G}(\alpha^m p^{m-1} + \beta^m p^{m-2}) \tag{61}$$

$$\mathbf{u}^m = \hat{\mathbf{u}}^m - \Delta t \Theta^n \mathbf{G}(\alpha^m (p^m - p^{m-1}) + \beta^m (p^{m-1} - p^{m-2})) \tag{62}$$

$$\Delta t \mathbf{D}\Theta^n \mathbf{G}(\alpha^m (p^m - p^{m-1})) = \mathbf{D}\hat{\mathbf{u}}^m - \Delta t \mathbf{D}\Theta^n \mathbf{G}(\beta^m (p^{m-1} - p^{m-2})) \tag{63}$$

For the three-stage three-step projection method, the Poisson equation (63) will be solved to get the pressure difference, which will be further used to correct the velocity by (62). Both the three-stage three-step and three-stage four-step projection methods need to solve the Poisson equations for the pressure or pressure difference in every sub-stage. We can employ multi-grid and Krylov subspace method, such as GMRES and CG, to accelerate the convergence.

For projection methods, it is important to present high-order accurate boundary conditions for the intermediate velocities. For the three-stage four-step and three-stage three-step projection methods, the second-order temporal boundary conditions for the intermediate velocities can be given as

$$\hat{\mathbf{u}}^m - \mathbf{u}^{m-1} = (\alpha^m + \beta^m)(\mathbf{u}^{n+1} - \mathbf{u}^n) \tag{64}$$

$$\mathbf{u}^m = \mathbf{u}^n + \sum_{l=1}^{m} (\alpha^l + \beta^l)(\mathbf{u}^{n+1} - \mathbf{u}^n) \tag{65}$$

4.2. Two iteration algorithms of the SIMPLE method for unsteady flows

In Section 3, we have concluded that SIMPLE type methods have second-order temporal accuracy for unsteady flows if a second-order temporal technique is used to update the temporal term in momentum equations at the predictor step. However, due to the linear treatment of the convective terms in SIMPLE type methods, coefficients $\mathbf{A}_P^n$ and $\mathbf{A}_M^n$ are dependent on the velocity for any implicit update of the convective term. Iteration is needed to obtain the intermediate velocity or the velocity at the next time step. Following, we consider two iteration algorithms for the application of

DOI: 10.1002/nme

SIMPLE type methods to calculations of unsteady flows. Based on the standard SIMPLE method of Equations (38), (41) and (42) and using the first-order fully implicit technique to update the temporal term, we have SIMPLE-A algorithm as

$$\begin{cases} \mathbf{A}_P^n = \mathbf{A}_P(\mathbf{u}^n), \quad \mathbf{A}_M^n = \mathbf{A}_M(\mathbf{u}^n) \\ DO \quad i = 1, L \\ \quad \begin{cases} \mathbf{A}_P^{n+\frac{i-1}{L}} = \mathbf{A}_P(\mathbf{u}^{n+\frac{i-1}{L}}), \ \mathbf{A}_M^{n+\frac{i-1}{L}} = \mathbf{A}_M(\mathbf{u}^{n+\frac{i-1}{L}}) \\ \left(\dfrac{1}{\Delta t}\mathbf{I} + \mathbf{A}_P^{n+\frac{i-1}{L}}\right)\hat{\mathbf{u}}_P^{n+\frac{i}{L}} = \dfrac{1}{\Delta t}\mathbf{u}_P^n + \sum_M \mathbf{A}_M^{n+\frac{i-1}{L}}\hat{\mathbf{u}}_M^{n+\frac{i}{L}} - \mathbf{G}p^{n+\frac{i-1}{L}} \\ \Delta t\mathbf{D}\left(\left(\mathbf{I} + \Delta t\mathbf{A}_P^{n+\frac{i-1}{L}}\right)^{-1}\mathbf{G}\left(p^{n+\frac{i}{L}} - p^{n+\frac{i-1}{L}}\right)\right) = \Delta t\mathbf{D}\hat{\mathbf{u}}_P^{n+\frac{i}{L}} \\ \mathbf{u}_P^{n+\frac{i}{L}} = \hat{\mathbf{u}}_P^{n+\frac{i}{L}} - \Delta t\left(\mathbf{I} + \Delta t\mathbf{A}_P^{n+\frac{i-1}{L}}\right)^{-1}\mathbf{G}\left(p^{n+\frac{i}{L}} - p^{n+\frac{i-1}{L}}\right) \end{cases} \\ END \quad DO \end{cases} \tag{66}$$

In algorithm (66), L steps of iterations are needed to get the velocity and pressure $(\mathbf{u}^{n+1}, p^{n+1})$ from $(\mathbf{u}^n, p^n)$. The pressure difference Poisson equation needs to be solved in every iteration step of this algorithm. We know that the solution of pressure Poisson equation will usually take much more time than the solution of the momentum equation itself. So, algorithm (66) is not economic. However, most of applications of SIMPLE type methods for unsteady flows employ this kind of algorithm. Now we present another algorithm, SIMPLE-B algorithm:

$$\begin{cases} \hat{\mathbf{u}}^n = \mathbf{u}^n \\ DO \quad i = 1, L \\ \quad \begin{cases} \hat{\mathbf{A}}_P^{n+\frac{i-1}{L}} = \mathbf{A}_P(\hat{\mathbf{u}}^{n+\frac{i-1}{L}}), \ \hat{\mathbf{A}}_M^{n+\frac{i-1}{L}} = \mathbf{A}_M(\hat{\mathbf{u}}^{n+\frac{i-1}{L}}) \\ \left(\dfrac{1}{\Delta t}\mathbf{I} + \hat{\mathbf{A}}_P^{n+\frac{i-1}{L}}\right)\hat{\mathbf{u}}_P^{n+\frac{i}{L}} = \dfrac{1}{\Delta t}\mathbf{u}_P^n + \sum_M \hat{\mathbf{A}}_M^{n+\frac{i-1}{L}}\hat{\mathbf{u}}_M^{n+\frac{i}{L}} - \mathbf{G}p^n \end{cases} \\ END \quad DO \\ \hat{\mathbf{A}}_P^{n+1} = \mathbf{A}_P(\hat{\mathbf{u}}^{n+1}), \ \hat{\mathbf{A}}_M^{n+1} = \mathbf{A}_M(\hat{\mathbf{u}}^{n+1}) \\ \Delta t\mathbf{D}\left(\left(\mathbf{I} + \Delta t\mathbf{A}_P^{n+1}\right)^{-1}\mathbf{G}\left(p^{n+1} - p^n\right)\right) = \Delta t\mathbf{D}\hat{\mathbf{u}}_P^{n+1} \\ \mathbf{u}_P^{n+1} = \hat{\mathbf{u}}_P^{n+1} - \Delta t\left(\mathbf{I} + \Delta t\mathbf{A}_P^{n+1}\right)^{-1}\mathbf{G}\left(p^{n+1} - p^n\right) \end{cases} \tag{67}$$

For SIMPLE-B of algorithm (67), L steps of iterations are only done for momentum equations to obtain the intermediate velocity $\hat{\mathbf{u}}_P^{n+1}$, which will be used to acquire the pressure and velocity corrections. Apparently, algorithm (67) will greatly save computational time compared to algorithm (66).

According to the analysis in Sections 2 and 3, both algorithms should be able to get second-order temporal accurate results supposing that second-order update techniques are employed for both the convective and diffusion terms. However, in (66) and (67), the first-order fully implicit technique

DOI: 10.1002/nme

is employed, so they have first-order temporal accuracy. To keep the SIMPLE method accurate, the second-order semi-implicit Crank–Nicholson scheme can be employed to update the convective and diffusion terms in the standard SIMPLE method. The diffusion term and the convective term are updated as

$$\left(\frac{\nabla^2\mathbf{u}}{Re}\right)^{n+1/2} = \frac{1}{2}\left(\left(\frac{\nabla^2\mathbf{u}}{Re}\right)^{n+1} + \left(\frac{\nabla^2\mathbf{u}}{Re}\right)^{n}\right), \quad (\mathbf{u}\cdot\nabla\mathbf{u})^{n+1/2} = \tfrac{1}{2}((\mathbf{u}\cdot\nabla\mathbf{u})^{n+1} + (\mathbf{u}\cdot\nabla\mathbf{u})^{n})$$

respectively. We then have a second-order SIMPLE Algorithm A:

$$\begin{cases} \mathbf{A}_P^n = \mathbf{A}_P(\mathbf{u}^n), \quad \mathbf{A}_M^n = \mathbf{A}_M(\mathbf{u}^n) \\ DO \quad i = 1, L \\ \quad \begin{cases} \mathbf{A}_P^{n+\frac{i-1}{L}} = \mathbf{A}_P(\mathbf{u}^{n+\frac{i-1}{L}}), \ \mathbf{A}_M^{n+\frac{i-1}{L}} = \mathbf{A}_M(\mathbf{u}^{n+\frac{i-1}{L}}) \\ \left(\frac{1}{\Delta t}\mathbf{I} + \frac{1}{2}\mathbf{A}_P^{n+\frac{i-1}{L}}\right)\hat{\mathbf{u}}_P^{n+\frac{i}{L}} = \left(\left(\frac{1}{\Delta t}\mathbf{I} + \frac{1}{2}\mathbf{A}_P^{n}\right)\mathbf{u}_P^n + \frac{1}{2}\sum_M \mathbf{A}_M^n\mathbf{u}_M^n\right) \\ \qquad\qquad + \frac{1}{2}\sum_M \mathbf{A}_M^{n+\frac{i-1}{L}}\hat{\mathbf{u}}_M^{n+\frac{i}{L}} - \mathbf{G}p^{n+\frac{i-1}{L}} \\ \Delta t\mathbf{D}\left(\left(\mathbf{I} + \frac{1}{2}\Delta t\mathbf{A}_P^{n+\frac{i-1}{L}}\right)^{-1}\mathbf{G}\left(p^{n+\frac{i}{L}} - p^{n+\frac{i-1}{L}}\right)\right) = \Delta t\mathbf{D}\hat{\mathbf{u}}_P^{n+\frac{i}{L}} \\ \mathbf{u}_P^{n+\frac{i}{L}} = \hat{\mathbf{u}}_P^{n+\frac{i}{L}} - \Delta t\left(\mathbf{I} + \frac{1}{2}\Delta t\mathbf{A}_P^{n+\frac{i-1}{L}}\right)^{-1}\mathbf{G}\left(p^{n+\frac{i}{L}} - p^{n+\frac{i-1}{L}}\right) \end{cases} \\ END \quad DO \end{cases} \tag{68}$$

and Algorithm B as

$$\begin{cases} \hat{\mathbf{u}}^n = \mathbf{u}^n \\ DO \quad i = 1, L \\ \quad \begin{cases} \hat{\mathbf{A}}_P^{n+\frac{i-1}{L}} = \mathbf{A}_P(\hat{\mathbf{u}}^{n+\frac{i-1}{L}}), \ \hat{\mathbf{A}}_M^{n+\frac{i-1}{L}} = \mathbf{A}_M(\hat{\mathbf{u}}^{n+\frac{i-1}{L}}) \\ \left(\frac{1}{\Delta t}\mathbf{I} + \frac{1}{2}\hat{\mathbf{A}}_P^{n+\frac{i-1}{L}}\right)\hat{\mathbf{u}}_P^{n+\frac{i}{L}} = \left(\left(\frac{1}{\Delta t} - \frac{1}{2}\mathbf{A}_P^{n}\right)\mathbf{u}_P^n + \frac{1}{2}\sum_M \mathbf{A}_M^n\mathbf{u}_M^n\right) \\ \qquad\qquad + \frac{1}{2}\sum_M \hat{\mathbf{A}}_M^{n+\frac{i-1}{L}}\hat{\mathbf{u}}_M^{n+\frac{i}{L}} - \mathbf{G}p^n \end{cases} \\ END \quad DO \\ \hat{\mathbf{A}}_P^{n+1} = \mathbf{A}_P(\hat{\mathbf{u}}^{n+1}), \ \hat{\mathbf{A}}_M^{n+1} = \mathbf{A}_M(\hat{\mathbf{u}}^{n+1}) \\ \Delta t\mathbf{D}\left(\left(\mathbf{I} + \frac{1}{2}\Delta t\mathbf{A}_P^{n+1}\right)^{-1}\mathbf{G}\left(p^{n+1} - p^n\right)\right) = \Delta t\mathbf{D}\hat{\mathbf{u}}_P^{n+1} \\ \mathbf{u}_P^{n+1} = \hat{\mathbf{u}}_P^{n+1} - \Delta t\left(\mathbf{I} + \frac{1}{2}\Delta t\mathbf{A}_P^{n+1}\right)^{-1}\mathbf{G}\left(p^{n+1} - p^n\right) \end{cases} \tag{69}$$

DOI: 10.1002/nme

Also, the multi-stage Runge–Kutta technique can be used to update the convective term in the SIMPLE method for simplicity and the Crank–Nicholson scheme can be employed for the update of the diffusion term for stability. We, then have a similar formulation of a SIMPLE type method as the general projection method for unsteady flows.

5. NUMERICAL VALIDATION

5.1. Two-dimensional vortex flow

The proposed schemes and boundary conditions for the four-step projection method of Equations (57)–(60) and SIMPLE type methods are tested in computing the following 2-D unsteady flow of decaying vortices [9], which has the following exact solution:

$$u(x, y, t) = -\cos(x)\sin(y)\exp(-2t) \tag{70}$$

$$v(x, y, t) = \sin(x)\cos(y)\exp(-2t) \tag{71}$$

$$p(x, y, t) = -\tfrac{1}{4}(\cos(2x) + \cos(2y))\exp(-4t) \tag{72}$$

Computations are carried out in the domain $0 \leqslant x, y \leqslant \pi$. In the computation, uniform collocated meshes were used. On the boundaries, the exact solution is imposed. The same CFL number is used for all of the tested algorithms, which means the time-step size is proportional to the grid size. The error is the absolute value of the difference between the numerical and analytical solutions. Figure 1 plots the maximum error in u, at different time levels as a function of mesh refinement. y-axis represents the maximum error in u, while x-axis represents the mesh points. Similar results are obtained for v. For the SIMPLE methods A and B in Equations (66) and (67), Figure 1(a) and (b) shows that they have first-order accuracy since the first-order fully implicit technique is employed to update both the convective and diffusion terms. Using the high-order SIMPLE method, in which the temporal term is discretized using a second-order temporal discretization scheme, Figure 1(c) shows that the slope is greater than 2 and the method has a second-order temporal accuracy. This validation shows that SIMPLE methods have second-order temporal accuracy if a second-order update is employed for the temporal term. Figure 1(d) shows that the projection methods with the update of the convective term and diffusion term using the Runge–Kutta technique and the Crank–Nicholson scheme, respectively, are of second-order temporal accuracy since the slopes in Figure 1(d) are greater than or equal to 2. The result indirectly proves that the connection between SIMPLE and projection methods built up in Section 3 is correct.

5.2. DNS of a fully developed turbulence

A DNS of the fully developed thermal field in a fully developed turbulent channel flow of air is carried out by the above developed projection method with the Runge–Kutta technique employed for update of the convective term and the Crank–Nicholson for the diffusion term. Periodic boundary conditions are applied in streamwise and spanwise directions, while the no slip wall conditions are used for the wall-normal direction. A fourth-order fully conservative central difference scheme [34] was employed to discretize the convective term. The FFT (Fast Fourier Transformation) method is employed to solve the PPE. For this computation, the turbulent Reynolds number, based on

DOI: 10.1002/nme

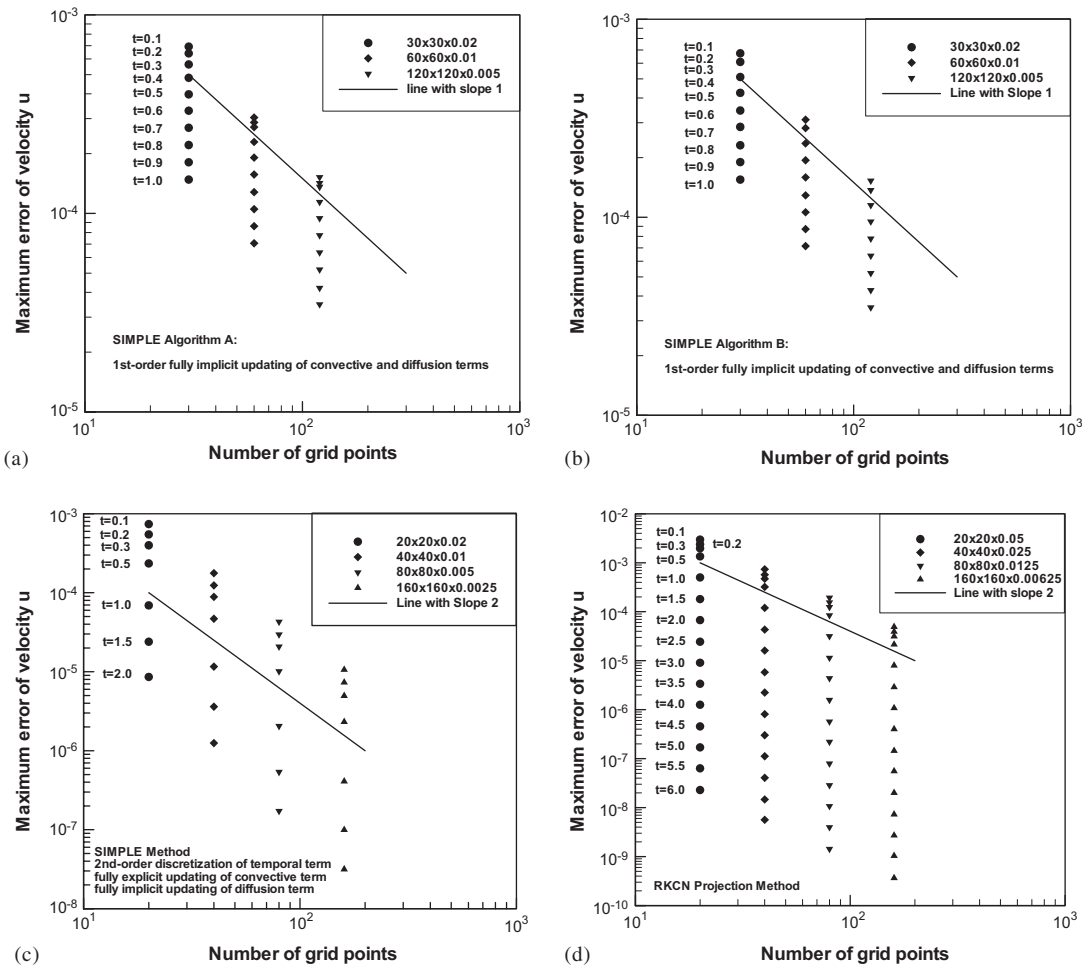

Figure 1. The maximum error of velocity u over the refined meshes: (a) by SIMPLE Algorithm A using first-order fully implicit update of convective and diffusion terms; (b) by SIMPLE Algorithm B using first-order fully implicit update of convective and diffusion terms; (c) by SIMPLE algorithm using second-order update of temporal term; and (d) by classical four-step projection method with a Runge–Kutta technique used.

wall friction velocity u_τ and width of channel δ, is set at 150. The computational periods are chosen to be $2.5\pi\delta$ and $\pi\delta$ in the streamwise and spanwise directions, respectively. Staggered grid system is employed to avoid the checkerboard phenomena. Non-uniform meshes of 128 points with hyperbolic tangent distribution are used with $\Delta y^+ = u_\tau \Delta y/\nu = 0.50 \sim 7.51$ in the wall-normal direction, while uniform meshes with spacing $\Delta x^+ = u_\tau \Delta x/\nu = 18.4$, $\Delta z^+ = u_\tau \Delta z/\nu = 7.36$ in streamwise and spanwise directions. In this paper, $()^+$ means the normalization by the wall variables, u_τ(friction velocity), ν(kinetic viscosity), and T_τ(friction temperature).

The dimensionless mean velocity and temperature profiles are shown as a function of $y^+ = u_\tau y/\nu$ in Figure 2(a). The root-mean-square velocity and temperature fluctuations, normalized by the

DOI: 10.1002/nme

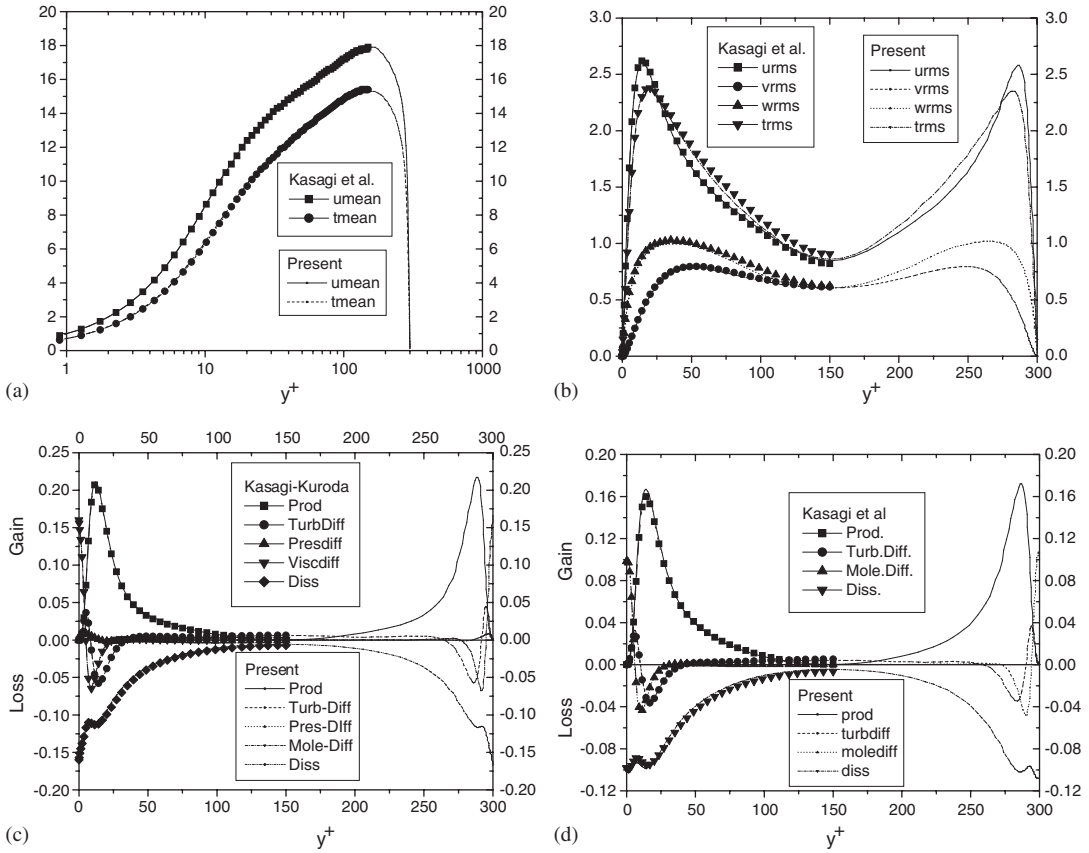

Figure 2. The maximum error of velocity u over the refined meshes: (a) dimensionless mean velocity and temperature; (b) distribution of rms velocity and temperature; (c) budget of turbulence kinetic; and (d) budget of temperature variance $k_\theta = \frac{1}{2}(\overline{\theta'})^2$.

friction velocity and friction temperature, respectively, are illustrated in Figure 2(b). Figure 2(c) and (d) showed the budgets of turbulent kinetic energy and temperature variance. From Figure 2, we can see all of the results agree closely with the results of Kasagi *et al.* [35], which were acquired by the high accurate spectral method. This illustrates that the general projection method with a Runge–Kutta technique for the update of the convective term and the Crank–Nicholson technique for the diffusion term can be applied on non-uniform meshes to get high accurate results. By employing high-order spatial difference scheme, it can be used to do the DNS, LES research of the turbulence.

5.3. Multi-fluid flows

Multi-phase flows are of importance in fluid engineering. The general projection methods have been employed for the simulation of multi-fluid flows with the level set method [32] and VOF method [36] employed to capture the interface. The dam breaking problem is calculated to validate the

DOI: 10.1002/nme

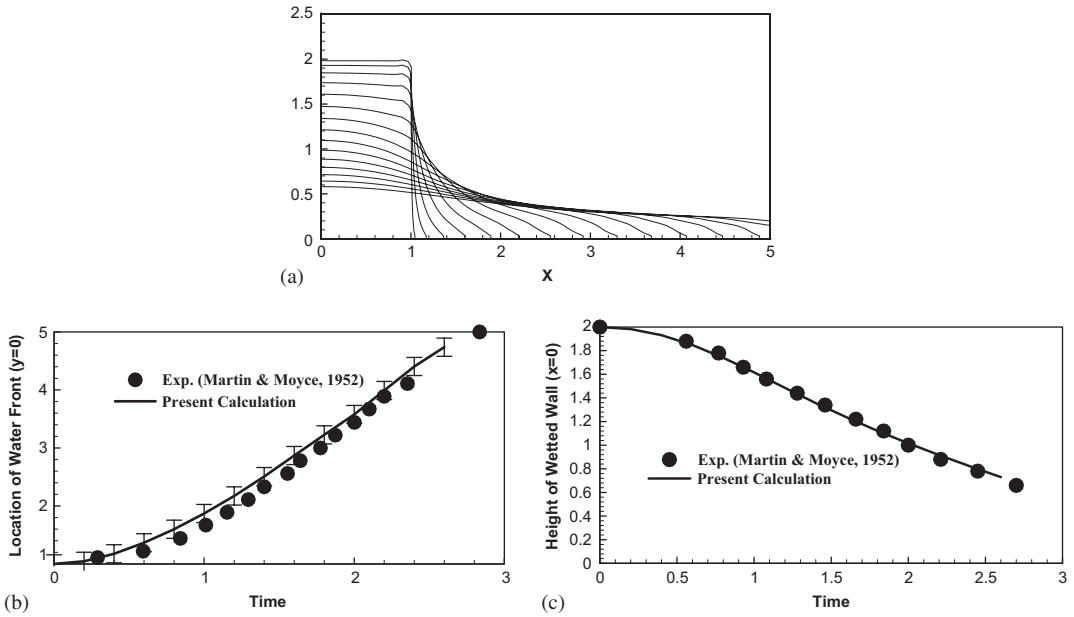

Figure 3. History of water front location on solid surfaces in the dam break: (a) zero level set contour from $time = 0.2$–3.0 with time interval 0.2; (b) location of water front at $y = 0$; and (c) height of wetted wall at $x = 0$.

method for multi-fluid flows. In this case, the level set method [32] is employed to automatically capture the interface. A 81×41 uniform Cartesian grid is used with an initial water column height to width ratio of 2. $\rho_{\text{water}}/\rho_{\text{back}} = 1000$, $\mu_{\text{water}}/\mu_{\text{back}} = 1000$, and $Re = 1000$. At the outlet boundary, the Neumann boundary condition is set for velocity. At all other boundaries, slip wall boundary conditions are applied for the velocities. The variable time step method developed in [24] is employed for the discretization of the reinitialization equation to reduce the mass loss. Figure 3(a) illustrates the free surface profiles between $time = 0.2$ and 3.0 with time interval 0.2. The water surface evolves in a smooth shape and no oscillation occurs at the interface near the solid wall. Figure 3(b) shows the history of the water front marching along the ground surface ($y = 0$) and Figure 3(c) shows the transient height of the wetted wall along the vertical surface ($x = 0$). Also the error bar in Figure 3(b) shows the interface height in this calculation. The experimental results [37] are also shown in Figure 3. The numerical results match the experimental data well.

We also applied the developed method and code for the simulation of a 3-D rising bubble in a channel, for which experimental data exist. Numerical results are described in terms of the dimensionless parameters Morton number ($Mo = g\mu_{\text{L}}^4(\rho_{\text{L}} - \rho_{\text{G}})/\rho_{\text{L}}^2\sigma^3$), and Eötvös number ($E\ddot{o} = g(\rho_{\text{L}} - \rho_{\text{G}})d_0^2/\sigma$) since both Mo and $E\ddot{o}$ do not include the terminal velocity, which can only be obtained after reaching a steady state. Here, g represents the gravity acceleration, ρ_{L} and ρ_{G} denote the density of the liquid and gas, respectively, μ_{L} and μ_{G} denote the viscosity of liquid and gas, respectively, σ is the surface tension coefficient, and d_0 is the initial bubble diameter.

In this paper, a second-order VOF method [36] is used to capture the interface. The general projection method with a non-linear coefficient matrix is used to do the direct simulation of

DOI: 10.1002/nme

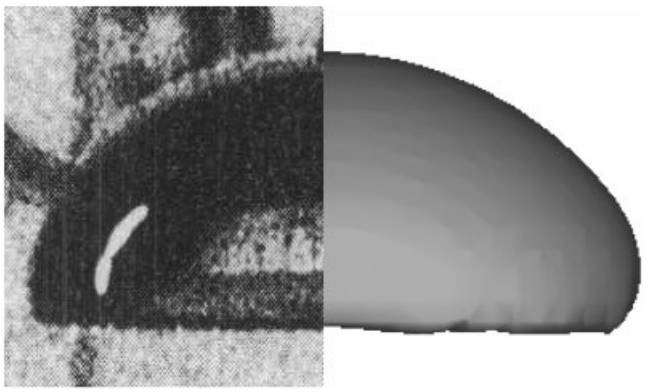

Figure 4. Comparison of bubble shapes.

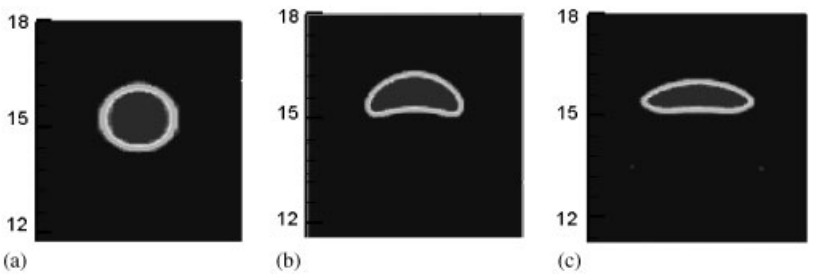

Figure 5. The shapes are consistent with Grace's generalized correlation figure: (a) $E\ddot{o}=3.6$, $Mo=9\times10^{-5}$; (b) $E\ddot{o}=36$, $Mo=9\times10^{-2}$; and (c) $E\ddot{o}=36$, $Mo=9\times10^{-5}$.

a rising bubble in a stagnant liquid. A uniform grid of $60\times60\times180$ is employed for the simulation region with the size of $3d_0\times3d_0\times9d_0$. The bubble rises up due to buoyancy. The rising bubble deforms and reaches a steady state. We simulate the case of $Mo=1.31$ and $E\ddot{o}=161$. Figure 4 shows that our computational interface shape is very consistent with the experimental result from Bhaga and Weber [38]. Figure 5 further shows that our computational shapes in a big region of Mo (from $Mo=9\times10^{-2}$ to $Mo=9\times10^{-5}$) and $E\ddot{o}$ (from $E\ddot{o}=3.6$ to $E\ddot{o}=36$). We have the ellipsoidal shape for the case of $E\ddot{o}=3.6$, $Mo=9\times10^{-5}$ as shown in Figure 5(a), the dimpled shape for the case of $E\ddot{o}=36$, $Mo=9\times10^{-2}$ as shown in Figure 5(b), and the sphere cap for the case of $E\ddot{o}=36$, $Mo=9\times10^{-5}$ as shown in Figure 5(c). These are very consistent with Grace's graphical correlation [39]. Figure 6 shows the standing vortex formed after a rising bubble. In this figure, We denotes the Weber number. Both We and Re numbers are defined based on the terminal velocity of a rising bubble. The numerical work of Ryskin and Leal [40] reports on the formation of an attached, or standing eddy behind a strongly deformed bubble. Ryskin and Leal concluded the key condition for existence of standing eddies is the generation of vorticity at a sufficient rate

DOI: 10.1002/nme

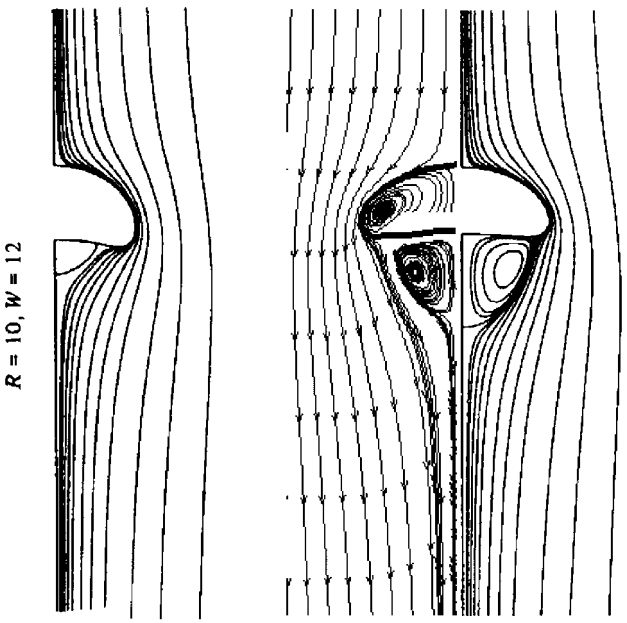

Figure 6. Vortexes and shapes comparison.

at the deformed bubble surface with the existence of sufficiently large curvature. Ryskin and Leal's numerical results clearly demonstrate the flow separation from a smooth free surface. Ryskin and Leal's results are also shown in Figure 6. From this comparison, we can see that the standing vortex from our computation is also very consistent with Ryskin and Leal's simulation results. The inner circulation is also shown in our computational result.

5.4. Magnetohydrodynamics

The simulation of magnetohydrodynamic (MHD) flows at high Hartmann numbers has been a topic of great interest in the development of a fusion reactor blanket [41]. To design self-cooled liquid metal blankets for fusion reactors, one must know about the behaviour of MHD flows at high Hartmann numbers. The square of the Hartmann number Ha is the ratio between the electromagnetic and the viscous forces. It is a measure of the magnetic field strength for a given fluid in a duct of a given scale. The dimensionless thickness of the Hartmann layers scales with Ha^{-1} and is very thin. Similarly, the side layers parallel to the magnetic field scale with $Ha^{-1/2}$ and are much thicker than the Hartmann layers at high Hartmann numbers. At high Hartmann number, a very fine mesh is required to resolve the Hartmann layer and the side layer. The general projection method is extended to MHD with the Lorentz force included as a source term. A consistent and conservative scheme [42] is employed for the calculation of the current density and the Lorentz force based on the electrical potential formula.

DOI: 10.1002/nme

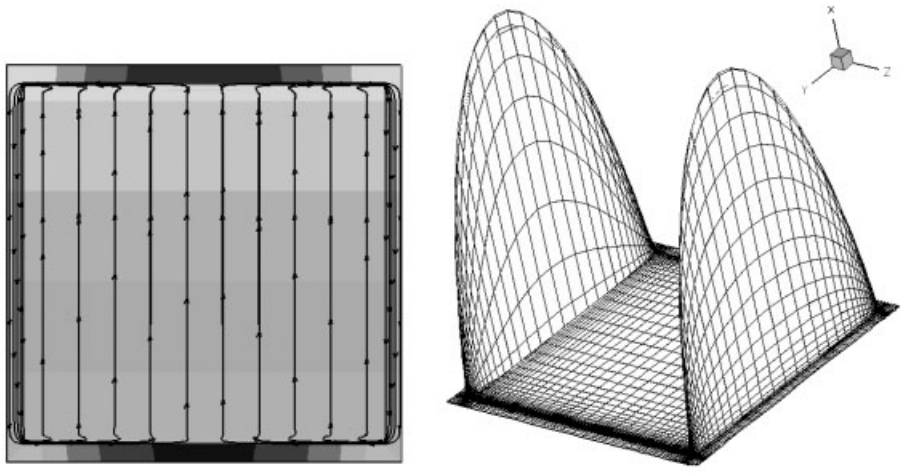

Figure 7. Current lines and velocity profile for fully developed flow at $Ha = 10k$: (a) current distribution and (b) velocity distribution.

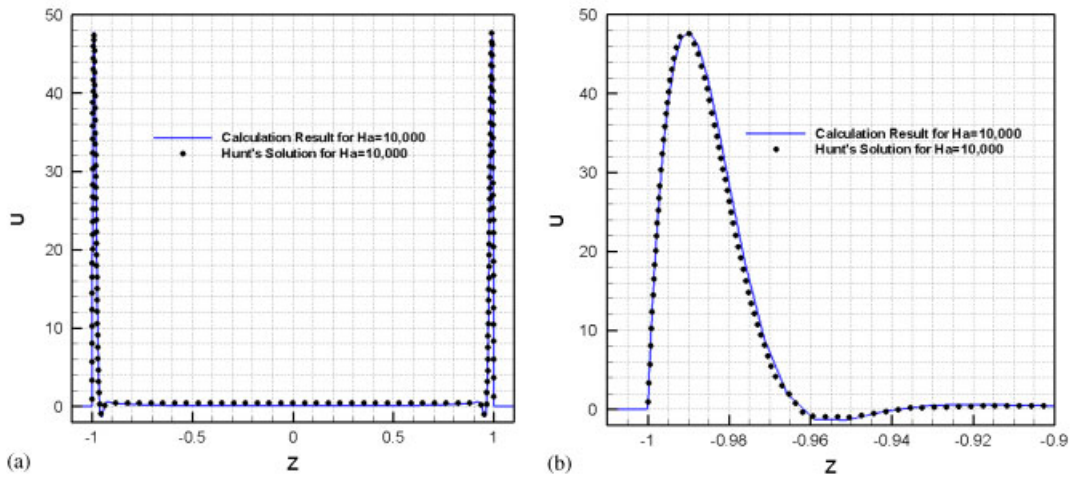

Figure 8. Results for fully developed flow with $Ha = 10\,000$: (a) comparing of velocity with Hunt's solution and (b) comparing of velocity in a side layer with Hunt's solution.

We present here the comparison between the calculation and analytical solution of Hunt's [43] at $Ha = 10\,000$ fully developed flow in a square duct. This is a large Hartmann number by the standard of MHD simulation. A channel of dimensions 2×2 was padded with walls with a thickness of 0.1 on all sides. The mesh has 67×67 non-uniform cells in the cross-section with four cells in

DOI: 10.1002/nme

the Hartmann layer. Magnetic field with a value of 1 is applied. The Hartmann walls are assumed to be conducting with a wall conductance ratio $c_w = 0.05$ and the side walls are insulated. The fully developed flow simulations used a zero velocity at time $t = 0$, with a pressure gradient of -1000 to drive the flow. A mass flow rate value of 0.208 was observed in the fully developed flow situations. This compares well with Hunt's solution, which gives a flow rate of 0.203. The convergent current distribution (Figure 7(a)), velocity profile (Figure 7(b)) are illustrated in Figure 7. Figure 8(a) gives the comparison between our results and Hunt's analytical results. Figure 8(b) zooms in the side layer. The numerical results on a collocated non-uniform grid match very well with Hunt's solution. The figure clearly shows the M-shape formed by the Lorentz force. In fact, we can see that two jets form in the side layers.

6. CONCLUDING REMARKS

A general second-order projection formula is developed based on a matrix analysis method by employing a pressure update of $\Theta^n \mathbf{G} p^{n+1} + (\mathbf{I} - \Theta^n)\mathbf{G} p^n$. Θ^n is a coefficient matrix, which may depend on grid size, time step and even velocity. Also, a concise formula is derived for SIMPLE like methods for unsteady flows. The SIMPLE type methods have second-order temporal accuracy for unsteady flows. We built up a bridge between SIMPLE type methods and second-order three-step projection methods. The standard SIMPLE method can be acquired from the general three-step projection formula. While the idea for the construction of SIMPLEC can be used to construct the traditional second-order projection method. The general projection method has been applied to simulate a 2-D Taylor vortex flow, DNS of a fully developed turbulence, multi-fluid flows with the surface tension included using the level set method and VOF method to capture the interface. It is also extended to the simulation of MHD incorporating with the consistent and conservative scheme for the calculation of the current density and the Lorentz force based on the electrical potential formula. Numerical examples validate the analysis in this paper.

REFERENCES

1. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1965; **8**(12):2182–2189.
2. Bell JB, Colella, P, Glaz HM. A second-order projection method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257–283.
3. Bell JB, Marcus DL. A second-order projection method for variable density flows. *Journal of Computational Physics* 1992; **101**:334–348.
4. Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2001; **168**:464–499.
5. Choi H, Moin P. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics* 1994; **113**:1–4.
6. Chorin AJ. Numerical solution of Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
7. Dukowicz JK, Dvinsky AS. Approximation factorization as a high-order splitting for implicit incompressible flow equations. *Journal of Computational Physics* 1992; **102**:336–347.
8. Gresho P. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduce a nearly consistent mass matrix. Part 1: Theory. *International Journal for Numerical Methods in Fluids* 1990; **11**:587–620.
9. Kim J, Moin P. Application of fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
10. Ni M-J, Komori S. General second-order projection formulas for unsteady flows. *AIAA Journal* 2002; **40**(7): 1464–1467.

DOI: 10.1002/nme

11. Perot JB. An analysis of the fractional step method. *Journal of Computational Physics* 1993; **108**:51–58.
12. Perot JB. Letter to the editor: comments on the fractional step method. *Journal of Computational Physics* 1995; **121**:190–191.
13. Temam R. *Navier–Stokes Equations*: *Theory and Numerical Analysis*. North-Holland: New York, 1979.
14. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
15. Patankar SV. A calculation procedure for two-dimensional elliptic situations. *Numerical Heat Transfer* 1981; **4**:409–425.
16. Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal Heat and Mass Transfer* 1972; **15**:1787–1803.
17. Tao W-Q, Qu Z-G, He Y-L. 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. *Numerical Heat Transfer B* 2003; **45**:1–17.
18. Van Doormaal JP, Raithby GD. Enhancement of the SIMPLE method for predicting incompressible fluid flows. *Numerical Heat Transfer* 1984; **7**:147–163.
19. Issa RI. Solution of the implicitly discretized fluid flow equations by operator-splitting. *Journal of Computational Physics* 1985; **62**:40–65.
20. Komori S, Nagaosa R, Murakami Y, Chiba S, Ishii K, Kuwahara K. Direct numerical simulation of 3D open-channel flow with zero-shear gas–liquid interface. *Physics of Fluids A* 1993; **5**:115–125.
21. Kothe DB, Mjolsness RC. RIPPLE: a new model for incompressible flows with free surfaces. *AIAA Journal* 1992; **30**(11):2694–2700.
22. Moin P, Mahesh K. Direct numerical simulation: a tool in turbulence research. *Annual Reviews in Fluid Mechanics* 1998; **30**:539–578.
23. Rai MM, Moin P. Direct simulation of turbulent flow using finite difference schemes. *Journal of Computational Physics* 1991; **96**:15–53.
24. Ni M-J, Komori S, Morley NB. Direct simulation of falling droplet in a closet channel. *International Journal of Heat and Mass Transfer* 2006; **49**:366–376.
25. Pucket EG, Almgren AS, Bell JB, Marcus DL, Rider WJ. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics* 1997; **130**:269–282.
26. Mohamad AA. Benchmark solution for unsteady state CFD problems. *Numerical Heat Transfer*, *Part A* 1998; **34**:653–672.
27. Chen L, Garimella SV, Reizes JA, Leonardi E. The development of a bubble rising in a viscous liquid. *Journal of Fluid Mechanics* 1999; **387**:61–96.
28. Udaykumar HS, Kan H-C, Shyy W, Tran-Son-Tay R. Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. *Journal of Computational Physics* 1997; **137**:366–405.
29. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
30. Suzuki H, Usui H. Direct numerical simulation of a turbulent duct flow by high-ordered scheme for SIMPLE. *The 13th Japanese Symposium of Computational Fluid Dynamics* (*JSCFD*), Tokyo, Japan, 1999; B04-3.
31. Shyy W. *Computational Modeling for Fluid Flow and Interfacial Transport*. Elsevier: Amsterdam, 1994.
32. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–154.
33. Ni M-J, Tao W-Q, Wang S-J. Stability analysis for discretized steady convective–diffusion equation. *Numerical Heat Transfer*, *Part B* 1999; **35**:369–388.
34. Morinishi Y, Lund TS, Vasilyev OV, Moin P. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics* 1998; **143**:90–124.
35. Kasagi N, Tomita Y, Kurda A. Direct numerical simulation of passive scalar field in a turbulent channel flow. *Journal of Heat Transfer* 1992; **114**:598–606.
36. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
37. Martin JC, Moyce WJ. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London*, *Series A*: *Mathematical and Physical Sciences* 1952; **A244**:312–324.
38. Bhaga D, Weber ME. Bubbles in viscous liquids: shapes, wakes and velocities. *Journal of Fluid Mechanics* 1981; **105**:61–85.
39. Grace JR. Shapes and velocities of bubbles rising in infinite liquids. *Transactions of the Institution of Chemical Engineers* 1973; **51**:116–120.

DOI: 10.1002/nme

40. Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. Part II: Buoyancy-driven motion of a gas bubble through a quiescent liquid. *Journal of Fluid Mechanics* 1984; **148**:19–35.
41. Abdou MA, The APEX Team. On the exploration of innovative concepts for fusion chamber technology fusion. *Fusion Engineering and Design* 2001; **54**:181–247.
42. Ni M-J, Abdou MA. Consistent and conservative schemes for incompressible MHD flows at low magnetic Reynolds numbers. Part II. A strong conservative formula of the Lorentz force on an arbitrary collocated mesh. *Journal of Computational Physics* 2007, under review.
43. Hunt JCR. Magnetohydrodynamic flow in rectangular ducts. *Journal of Fluid Mechanics* 1965; **21**:577–590.

DOI: 10.1002/nme