Available online at www.sciencedirect.com

SCIENCE DIRECT®

International Journal of Heat and Mass Transfer 49 (2006) 535–545

International Journal of HEAT and MASS TRANSFER

www.elsevier.com/locate/ijhmt

Exact factorization technique for numerical simulations of incompressible Navier–Stokes flows

K.K.Q. Zhang, W.J. Minkowycz, F. Mashayek *

Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, 842 West Taylor Street, 2053 ERF, Chicago, IL 60607, United States

Received 13 August 2005; received in revised form 28 October 2005
Available online 9 January 2006

Abstract

Splitting techniques break an ill-conditioned indefinite system resulting from incompressible Navier–Stokes equations into well-conditioned subsystems, which can be solved reliably and efficiently. Apart from the ambiguity regarding numerical boundary conditions for the pressure (and for intermediate velocities, whenever introduced), splitting techniques usually incur splitting errors which reduce time accuracy. The discrete approach of approximate factorization techniques eliminates the need of numerical boundary conditions and restores time accuracy by an approximate inversion of some matrix in the case of semi-implicit time schemes. For linear implicit, non-linear implicit, and higher-order semi-implicit time schemes, however, approximate factorization techniques are laborious. In this paper, we systematically present a new and straightforward exact factorization technique. The main contributions of this work include: (1) the idea of removing the splitting error or the idea of restoring time accuracy for fully discrete systems, (2) the introduction of the pressure-update type and the pressure-correction type of exact factorization techniques for any time schemes, and (3) an analysis of several established techniques and their relations to the exact factorization technique. The exact factorization technique is implemented with a standard second-order finite volume method and is verified numerically.

Keywords: Exact factorization; Splitting; Numerical boundary conditions; Incompressible Navier–Stokes flows

1. Introduction

The incompressible Navier–Stokes equations in non-dimensional form are described as

$$\frac{\partial u_i}{\partial t} + H_i = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\frac{\partial}{\partial x_j}u_i + f_i, \quad (1)$$

$$\frac{\partial u_j}{\partial x_j} = 0, \quad (2)$$

where $H_i \equiv \frac{\partial}{\partial x_j}(u_j u_i)$, u_i is the velocity component in the x_i direction, f_i is the body force, and Re is the Reynolds number. For decades, a numerical simulation of this system has remained as one of the most interesting topics, and the pressure term is believed by many to be a source for trouble. The indefinite system due to the mixed formulation, in which velocities and pressure are solved simultaneously without any manipulations, is ill-conditioned [9]. In such a system, a relatively small change in some entry of the matrix results in a relatively large change in the solution. Hence, accumulated computer round-off errors or some inherent perturbations of iterative processes make the convergence very hard to achieve. When the size of the system increases or when the physical solutions tend to be more rugged as a consequence of higher Reynolds numbers or discontinuities, the conditioning of the discrete system further deteriorates so that the convergence becomes even more difficult. In some situations, eventually the discrete system becomes singular and no solution can be found. This is why successful simulations of high Re incompressible flows with finite difference methods and finite element

* Corresponding author. Tel.: +1 312 996 1154; fax: +1 312 413 0447.
E-mail address: mashayek@uic.edu (F. Mashayek).

0017-9310/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2005.10.034

methods in mixed formulation are rare. In some situations, pressure stabilized methods help to improve the behavior of the discrete system to some degree. However, these methods often invoke some ad hoc parameters and are too expensive for implementation. Also, none of the stabilized methods could reach high-order spatial accuracy, and in fact frequently only a first-order accuracy could be attained. Furthermore, all stabilized methods fail to decrease the size of the discrete system.

To tackle the pressure term, various artificial compressibility (AC) types of methods were invented, such as Chorin's AC for steady flows [5], the consistent penalty method [2], the generalized AC (the iterative Uzawa algorithm) for transient flows [4], and the reduced integration penalty method [18]. Generally speaking, these methods converge slowly and sometimes fail to converge. Often, the range of appropriate parameters to maintain both reliability and efficiency is narrow. Moreover, some methods in this category are not able to produce accurate results for pressure and some are not able to reduce the size of the discrete system. Vorticity-stream function formulation [12] is very competitive in 2-D and 3-D axisymmetric calculations. In general 3-D calculations the stream function does exist [26]; however, both the vorticity and the stream function have three components. Even in 2-D and 3-D axisymmetric calculations, the derived boundary conditions for the vorticity incurs either loss of accuracy or loss of flexibility of numerical methods.

Splitting methods (also known as projection/operator-splitting/time-splitting/fractional-step methods) remain popular among numerical community. The first two papers on the subject were Harlow and Welch's marker-and-cell (MAC) method [17] and Chorin's projection method [6]. Both methods take a first-order explicit scheme, and both require no initial boundary conditions for pressure which is consistent with the original mathematical system. We would like to call them, including Kim and Moin's [21] second-order semi-implicit fractional-step (splitting) method, pressure-update (PU) methods. In PU methods, the pressure or some variable closely related to the pressure is solved according to a Poisson equation. In contrast, one may solve the change of the pressure from a Poisson equation. Examples are second-order methods by van Kan [29] and Bell et al. [1], and higher-order methods by Karniadakis et al. [20]. We would like to refer to the latter category pressure-correction (PC) methods.

The issue of the boundary conditions for the Poisson equation, as well as boundary conditions for intermediate velocities (whenever introduced), has in the past sparked a considerable debate [17,6,7,22,24,8,21,29,15,1,13,14,20,10,23,27,25,3,16] (in chronic order). According to [27], the accuracy of finite difference schemes "depends critically on the boundary condition for the intermediate velocity." However, the numerical boundary condition for the pressure Poisson equation (PPE) is implied in the system already and actually is not required in practice, as shown in the PC type of approximate factorization technique by Dukowicz and Dvinsky [10] and in the PU type of approximate factorization technique by Perot [23]. The exact factorization technique to be introduced in this paper requires no numerical boundary conditions at all.

The elusive issue of splitting error has also drawn sizable attention, in that many of those papers on the issue of numerical boundary conditions also concern the issue of time accuracy. According to Perot [23], a lower-order splitting-induced term in the momentum equation is pointed out as the source of the trouble. Approximate factorization techniques remove the splitting error through an approximate inversion of some matrix. Quarteroni et al. [25] presented a framework for splitting methods and approximate factorization techniques, including Perot's approach. However, the exact factorization technique presented in this paper takes a different path in terms of restoring time accuracy.

In the next section we progressively introduce the exact factorization technique. We start from a specific time and spatial discretization, discuss the approach of approximate factorization to split the system, and introduce the exact factorization technique. Then, we generalize the technique to any time scheme and introduce another version of exact factorization. After that, we compare the technique with the semi-discrete counterpart and make some additional comments on the exact factorization technique. In the following section, a reduced version of exact factorization technique is discussed and comparisons to several well known techniques are made. Implementation and numerical results make up another full section to support the technique.

2. Exact factorization technique

2.1. Temporal and spatial discretizations

The momentum equation (1) indicates that the pressure gradient should stay at the same site with the time derivative of the velocity, in the four-dimensional space–time coordinates system. This implies that at the same time level, the pressure node should stagger from velocity nodes as on the MAC staggered grid. This also implies that the time level for pressure should stagger from the time level for velocity. Since the pressure, which appears only in the momentum equation, is not initially specified, the momentum equation should be displaced from the initial time level.

In light of the above views, we assign time levels for the velocity and for the numerical pressure ϕ as shown in Fig. 1. The idea of numerical pressure was introduced by Kim and Moin [21] and will be further discussed later, but tentatively we may simply regard it as the pressure. The incompressibility is satisfied on time levels for the velocity while momentum equations are satisfied on time levels for pressure. It is noted that in space the incompressibility is satisfied at pressure nodes while momentum equations are satisfied at velocity nodes, just opposite to the

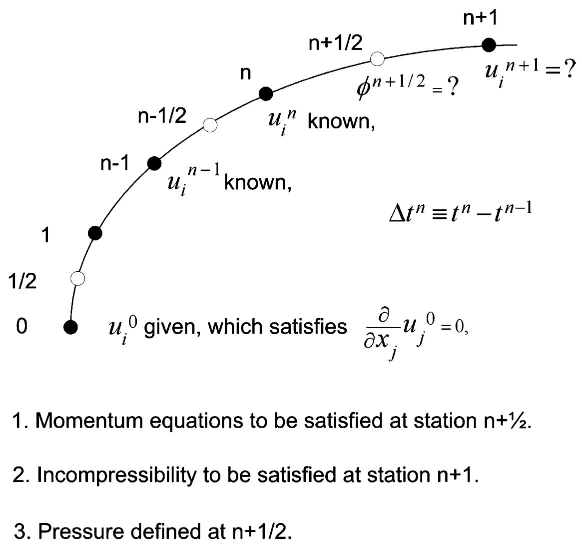

Fig. 1. Schematic for time marching.

case of the time. As indicated in Fig. 1, in a typical step both u_i^n and u_i^{n-1} are known and new $\phi^{n+\frac{1}{2}}$ and u_i^{n+1} are to be sought.

With a semi-implicit scheme, second-order Adams–Bashforth for the convection term and second-order Adams–Moulton for the diffusion term, the continuous system (1) and (2) becomes

$$\frac{u_i^{n+1} - u_i^n}{\Delta t^n} + H_i^{n+\frac{1}{2}} = -\frac{\partial}{\partial x_i} p^{n+\frac{1}{2}} + \frac{1}{2Re} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_j} \left(u_i^n + u_i^{n+1} \right) + f_i^{n+\frac{1}{2}},$$
$$\frac{\partial}{\partial x_j} u_j^{n+1} = 0,$$

where the variable time step $\Delta t^n \equiv t^{n+1} - t^n$ and the convection term

$$H_i^{n+\frac{1}{2}} \equiv \left(1 + \frac{1}{2} \frac{\Delta t^n}{\Delta t^{n-1}} \right) H(u_i^n) - \frac{1}{2} \frac{\Delta t^n}{\Delta t^{n-1}} H(u_i^{n-1}) + \mathcal{O}\{(\Delta t)^2\},$$
$$\text{for } n = 0, 1, 2, \ldots, \tag{3}$$
$$\Delta t^{-1} \equiv \Delta t^0, \quad u_i^{-1} \equiv u_i^0,$$

so that the globally second-order time accuracy for the velocity is maintained. The above system can be recast as

$$\left(\frac{1}{\Delta t^n} - \frac{1}{2Re} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_j} \right) u_i^{n+1} + \frac{\partial}{\partial x_i} p^{n+\frac{1}{2}}$$
$$= \left(\frac{1}{\Delta t^n} + \frac{1}{2Re} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_j} \right) u_i^n - H_i^{n+\frac{1}{2}} + f_i^{n+\frac{1}{2}}, \tag{4}$$
$$\frac{\partial}{\partial x_j} u_j^{n+1} = 0, \tag{5}$$

which is a semi-discrete (discrete in time, continuous in space) description of the original continuous system.

With a typical spatial discretization method such as finite volume method, the semi-discrete system (4) and (5) can always be written in a fully discrete form

$$\left(\frac{1}{\Delta t^n} \underline{\underline{I}} - \frac{1}{2Re} \underline{\underline{L}} \right) \underline{u}^{n+1} + \underline{\underline{G}} \underline{p}^{n+\frac{1}{2}}$$
$$= \left(\frac{1}{\Delta t^n} \underline{\underline{I}} + \frac{1}{2Re} \underline{\underline{L}} \right) \underline{u}^n + \frac{1}{Re} \underline{S}^{l} - \underline{H}^{n+\frac{1}{2}} + \underline{f}^{n+\frac{1}{2}}, \tag{6}$$
$$\underline{\underline{D}} \underline{u}^{n+1} = \underline{S}^{d}. \tag{7}$$

In the above system (6) and (7), matrix $\underline{\underline{L}}$ denotes the discrete Laplace operator with adjustment to Dirichlet or Neumann boundary conditions. Matrix $\underline{\underline{G}}$ represents the discrete gradient operator on the pressure, and this operator is not subjected to modifications in the vicinity of boundaries, if discretized on a MAC staggered grid. Matrix $\underline{\underline{D}}$ denotes the discrete divergence operator with adjustment to Dirichlet or Neumann boundary conditions. Vector $\underline{u}^{n+1}$ contains all discrete unknowns for velocities and vector $\underline{p}^{n+\frac{1}{2}}$ contains all discrete unknowns for the pressure. Vector $\underline{S}^{l}$ stands for the source term due to inhomogeneous Dirichlet boundary conditions on the discrete Laplace operator and vector $\underline{S}^{d}$ stands for the source term due to inhomogeneous Dirichlet boundary conditions on the discrete divergence operator. Finally $\underline{H}^{n+\frac{1}{2}}$ is the discrete form of (3) and it has all relevant boundary conditions included. By defining

$$\underline{S}^{u} \equiv \left(\frac{1}{\Delta t^n} \underline{\underline{I}} + \frac{1}{2Re} \underline{\underline{L}} \right) \underline{u}^n + \frac{1}{Re} \underline{S}^{l} - \underline{H}^{n+\frac{1}{2}} + \underline{f}^{n+\frac{1}{2}} \tag{8}$$

we have a fully discrete system in a compact matrix form

$$\begin{bmatrix} \frac{1}{\Delta t^n} \underline{\underline{I}} - \frac{1}{2Re} \underline{\underline{L}} & \underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^{u} \\ \underline{S}^{d} \end{Bmatrix} \tag{9}$$

Note, by default in the above spatial discretization only discrete unknowns in the interior of the flow domain are contained in $\underline{u}^{n+1}$ and $\underline{p}^{n+\frac{1}{2}}$. If discrete velocities on boundaries are regarded as unknowns and boundary conditions are regarded as additional discrete equations, the format of system (9) has to be modified.

With velocities and the pressure coupled, system (9) is called in mixed form and it is difficult to solve the system directly, as elaborated in Section 1.

2.2. Perot's approximate factorization

To have a context for the exact factorization technique to be analyzed in the next subsection, here Perot's approximate factorization (block LU decomposition) technique [23] is rephrased. We begin with first-order approximate factorization. It is hard to factorize the system (9) exactly. However, the approximated system

$$\begin{bmatrix} \frac{1}{\Delta t^n} \underline{\underline{I}} - \frac{1}{2Re} \underline{\underline{L}} & \left(\frac{1}{\Delta t^n} \underline{\underline{I}} - \frac{1}{2Re} \underline{\underline{L}} \right) \underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \Delta t^n \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^{u} \\ \underline{S}^{d} \end{Bmatrix} \tag{10}$$

can be easily factorized as

$$\begin{bmatrix} \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} & \underline{\underline{0}} \\ \underline{\underline{D}} & -\underline{\underline{D}}\,\underline{\underline{G}} \end{bmatrix} \begin{bmatrix} \underline{\underline{I}} & \underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \Delta t^n \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\rm d} \end{Bmatrix}.$$

By introducing intermediate velocities $\underline{u}^*$ the above LU decomposition is equivalent to the following set:

$$\begin{bmatrix} \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} & \underline{\underline{0}} \\ \underline{\underline{D}} & \underline{\underline{D}}\,\underline{\underline{G}} \end{bmatrix} \begin{Bmatrix} \underline{u}^* \\ \Delta t^n \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\rm d} \end{Bmatrix},$$

$$\begin{bmatrix} \underline{\underline{I}} & \underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \Delta t^n \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{u}^* \\ \Delta t^n \underline{p}^{n+\frac{1}{2}} \end{Bmatrix}.$$

Or, we have

$$\left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{u}^* = \underline{S}^u, \tag{11}$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\left(\Delta t^n \underline{p}^{n+\frac{1}{2}}\right) = \underline{\underline{D}}\,\underline{u}^* - \underline{S}^{\rm d}, \tag{12}$$

$$\underline{u}^{n+1} = \underline{u}^* - \underline{\underline{G}}\left(\Delta t^n \underline{p}^{n+\frac{1}{2}}\right). \tag{13}$$

For the sake of block LU decomposition, the system (11), (13) or equivalently the system (10) is only of first-order accuracy in time, while the system before manipulations is of second-order accuracy. This is because the discrete momentum equation in (9) is approximated by the discrete momentum equation in (10), and the discrepancy is the term $\frac{\Delta t^n}{2Re}\underline{\underline{L}}\,\underline{\underline{G}}\,\underline{p}^{n+\frac{1}{2}}$, which is of first-order accuracy.

To regain the second-order accuracy in time, one may adopt Perot's approach [23], which is rephrased as follows. A system in the format of

$$\begin{bmatrix} \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} & \left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{\underline{B}}\,\underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\rm d} \end{Bmatrix} \tag{14}$$

can be easily decomposed into

$$\begin{bmatrix} \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} & \underline{\underline{0}} \\ \underline{\underline{D}} & -\underline{\underline{D}}\,\underline{\underline{B}}\,\underline{\underline{G}} \end{bmatrix} \begin{bmatrix} \underline{\underline{I}} & \underline{\underline{B}}\,\underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\rm d} \end{Bmatrix}.$$

Choosing $\underline{\underline{B}}$ to be approximate inversion of $\underline{\underline{A}}$ can remove the splitting error. For example, if choosing

$$\underline{\underline{B}} = \Delta t^n\left(\underline{\underline{I}} + \frac{\Delta t^n}{2Re}\underline{\underline{L}}\right),$$

then system (14) differs from system (9) only by a second-order term, that is, the second-order accuracy is recovered.

2.3. Exact factorization technique for semi-implicit schemes

For high-order schemes, Perot's approach to recover accuracy becomes laborious [23]. Here we may play a much simpler trick to regain time accuracy. The first-order approximation (10) only made one modification to the original discrete system (9), and the consequence is an additional term in the momentum equation which causes loss of accuracy in time. If this term is absorbed into the pressure gradient term, the original discrete momentum equation can recover accuracy. Following this idea, we may introduce the *numerical pressure* (or gauge pressure) ϕ and replace the first-order approximation (10) by

$$\begin{bmatrix} \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} & \left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \underline{\phi}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\rm d} \end{Bmatrix}, \tag{15}$$

which is of course a factorizable system because it mimics system (10). If the numerical pressure is *implicitly* defined by

$$\left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}} \equiv \underline{\underline{G}}\,\underline{p}^{n+\frac{1}{2}}, \tag{16}$$

then we notice that for discrete velocities the system (15) and the system (9) are identical. In other words, we have split the system without loss of time accuracy.

As in Perot's approximate factorization, the system (15) can be factorized and rewritten as

$$\left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{u}^* = \underline{S}^u, \tag{17}$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}} = \underline{\underline{D}}\,\underline{u}^* - \underline{S}^{\rm d}, \tag{18}$$

$$\underline{u}^{n+1} = \underline{u}^* - \underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}}. \tag{19}$$

In addition, Eq. (16) may be replaced by

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{p}^{n+\frac{1}{2}} = \underline{\underline{D}}\left(\frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}}\right)\underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}}. \tag{20}$$

We may implicitly (in a coupled way) solve (17) to obtain $\underline{u}^*$, implicitly solve discrete Poisson system (18) for numerical pressure $\underline{\phi}^{n+\frac{1}{2}}$, and finally explicitly update discrete velocities in accordance with (19). The vector $\underline{S}^u$ is defined by (8), which involves the definition of $H_i^{n+\frac{1}{2}}$ by (3). The real pressure is never involved in the time marching procedure and can be recovered from (20) in a post-processing manner. Hence, the essence of the above trick is to absorb into the pressure some first-order numerical error, which is introduced for the sake of breaking the ill-conditioned indefinite system into well-conditioned subsystems, to maintain second-order accuracy for velocities. The whole technique is called *exact factorization*.

Another extremely important feature of the exact factorization technique must be pointed out: since the starting point for the system (17)–(19) is the discrete system (9) which has all the necessary boundary conditions included, no numerical boundary conditions for the numerical pressure ϕ or for the intermediate velocities u^* are needed.

2.4. Exact factorization technique for any time schemes

The exact factorization technique is not limited to semi-implicit time schemes only. With any time schemes and spatial methods, a straightforward discretization of Eqs. (1) and (2) will produce a system of form

$$\begin{bmatrix} \underline{\underline{A}} & \underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{p} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^\mathrm{d} \end{Bmatrix}, \tag{21}$$

where the non-singular matrix $\underline{\underline{A}}$ represents all possible discrete operators on the velocity, matrices $\underline{\underline{G}}$ and $\underline{\underline{D}}$ denote discrete gradient operator and divergence operator respectively, $\underline{u}$ and $\underline{p}$ contain all discrete unknowns, and $\underline{S}^u$ and $\underline{S}^\mathrm{d}$ stand for two known vectors. It is worthwhile to point out that all boundary conditions have been incorporated in the system (21). For instance, $\underline{S}^\mathrm{d}$ is the consequence of application of Dirichlet boundary conditions for the continuity equation. For the *velocity*, the fully discrete indefinite system (21) is equivalent to

$$\begin{bmatrix} \underline{\underline{A}} & \underline{\underline{A}}\,\underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{\phi} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^\mathrm{d} \end{Bmatrix}, \tag{22}$$

if the *numerical pressure* ϕ is defined *implicitly* through

$$\underline{\underline{A}}\,\underline{\underline{G}}\,\underline{\phi} \equiv \underline{\underline{G}}\,\underline{p}. \tag{23}$$

The existence of such a ϕ will be discussed shortly. The system (22) can be factorized

$$\begin{bmatrix} \underline{\underline{A}} & \underline{\underline{0}} \\ \underline{\underline{D}} & -\underline{\underline{D}}\,\underline{\underline{G}} \end{bmatrix} \begin{bmatrix} \underline{\underline{I}} & \underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{\phi} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^\mathrm{d} \end{Bmatrix},$$

where $\underline{\underline{I}}$ is the identity matrix. Introducing an auxiliary vector $\underline{u}^*$, the factorized form breaks into the following pair:

$$\begin{bmatrix} \underline{\underline{A}} & \underline{\underline{0}} \\ \underline{\underline{D}} & -\underline{\underline{D}}\,\underline{\underline{G}} \end{bmatrix} \begin{Bmatrix} \underline{u}^* \\ \underline{\phi} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^\mathrm{d} \end{Bmatrix},$$

$$\begin{bmatrix} \underline{\underline{I}} & \underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{\phi} \end{Bmatrix} = \begin{Bmatrix} \underline{u}^* \\ \underline{\phi} \end{Bmatrix},$$

which can be recast into

$$\underline{\underline{A}}\,\underline{u}^* = \underline{S}^u, \tag{24}$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{\phi} = \underline{\underline{D}}\,\underline{u}^* - \underline{S}^\mathrm{d}, \tag{25}$$

$$\underline{u} = \underline{u}^* - \underline{\underline{G}}\,\underline{\phi}. \tag{26}$$

The auxiliary $\underline{u}^*$ needs no boundary conditions since it is a discrete vector instead of a continuous variable. We may first explicitly or implicitly solve Eq. (24) to obtain the auxiliary vector $\underline{u}^*$, then implicitly solve Eq. (25) for the numerical pressure $\underline{\phi}$, and finally explicitly update $\underline{u}$ by expression (26). Eq. (23) can be used to recover the real pressure in a postprocessing manner

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{p} = \underline{\underline{D}}\,\underline{\underline{A}}\,\underline{\underline{G}}\,\underline{\phi}.$$

The whole algorithm does not, different from Perot's approximate factorization, resort to an approximate inversion of matrix $\underline{\underline{A}}$, thus it is referred to exact factorization technique.

The legitimacy of defining ϕ through Eq. (23) is tantamount to the solvability of ϕ through Eq. (25). The non-singularity of $\underline{\underline{A}}$ guarantees that $\underline{u}^*$ can be found through (24), and the operator $\underline{\underline{D}}\,\underline{\underline{G}}$ actually is the discrete counterpart of Laplace operator. Thus, ϕ can be found through (25).

The exact factorization technique presented so far belongs to PU type of methods, where the Poisson operator acts on the numerical pressure. In the PC approach the Poisson operator acts on the change of numerical pressure. It is advised to derive the PC version of exact factorization technique from the PU counterpart.

2.5. Pressure-correction type of exact factorization technique

Assigning a time index to system (24)–(26), we have

$$\underline{\underline{A}}\,\underline{u}^{*,n+1} = \underline{S}^u, \tag{27}$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}} = \underline{\underline{D}}\,\underline{u}^{*,n+1} - \underline{S}^\mathrm{d}, \tag{28}$$

$$\underline{u}^{n+1} = \underline{u}^{*,n+1} - \underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}}, \tag{29}$$

where the time dependency of $\underline{S}^u$ is not indicated because of its irrelevance to the current discussion. In the previous time step, the counterpart for Eq. (28) is

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{\phi}^{n-\frac{1}{2}} = \underline{\underline{D}}\,\underline{u}^{*,n} - \underline{S}^\mathrm{d}. \tag{30}$$

Subtracting Eq. (30) from Eq. (28) we obtain

$$\underline{\underline{D}}\,\underline{\underline{G}}\left(\underline{\phi}^{n+\frac{1}{2}} - \underline{\phi}^{n-\frac{1}{2}}\right) = \underline{\underline{D}}(\underline{u}^{*,n+1} - \underline{u}^{*,n}).$$

By introducing

$$\underline{\psi} \equiv \underline{\phi}^{n+\frac{1}{2}} - \underline{\phi}^{n-\frac{1}{2}},$$

we obtain a PC type algorithm

$$\underline{\underline{A}}\,\underline{u}^{*,n+1} = \underline{S}^u,$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{\psi} = \underline{\underline{D}}(\underline{u}^{*,n+1} - \underline{u}^{*,n}),$$

$$\underline{\phi}^{n+\frac{1}{2}} = \underline{\phi}^{n-\frac{1}{2}} + \underline{\psi},$$

$$\underline{u}^{n+1} = \underline{u}^{*,n+1} - \underline{\underline{G}}\,\underline{\phi}^{n+\frac{1}{2}}.$$

The first time step ($n = 0$) is solved according to the PU type algorithm (27)–(29). $\underline{S}^\mathrm{d}$ enters the system through the first step.

2.6. Relation to Kim and Moin's splitting method

The idea of absorbing numerical error into pressure was first exercised by Kim and Moin [21] in a semi-discrete version. Kim and Moin's splitting method can be rephrased as

$$\frac{\hat{u}_i - u_i^n}{\Delta t} = -\frac{1}{2}(3H_i^n - H_i^{n-1}) + \frac{1}{2}\frac{1}{Re}\frac{\partial^2}{\partial x_j \partial x_j}(\hat{u}_i + u_i^n) + f_i, \tag{31}$$

$$\frac{\partial^2 \phi^{n+1}}{\partial x_j \partial x_j} = \frac{1}{\Delta t}\frac{\partial \hat{u}_j}{\partial x_j}, \tag{32}$$

$$\frac{u_i^{n+1} - \hat{u}_i}{\Delta t} = -\frac{\partial \phi^{n+1}}{\partial x_i}, \tag{33}$$

where ϕ is *numerical pressure* (or *gauge pressure*, *pseudo-pressure*). The real pressure can be calculated in a postprocessing manner

$$p^{n+1} = \phi^{n+1} - \frac{\Delta t}{2Re}\frac{\partial^2 \phi^{n+1}}{\partial x_j \partial x_j}.$$

System (31)–(33) is the semi-discrete analogy of fully discrete system (17)–(19) with $\underline{S}^u$ defined through (8), except some irrelevant differences such as ϕ are defined differently and $\underline{S}^d$ is not explicitly mentioned in the semi-discrete version. When a staggered grid is used, Eq. (33) does not require any boundary condition for the numerical pressure ϕ. However, boundary conditions for ϕ and for $\hat{u}$ must be provided in the Poisson equation (32).

In Kim and Moin's splitting method, a temporal discretization is carried out first, then the semi-discrete system is split, and finally a spatial discretization is conducted. Such an algorithm invites the issue of numerical boundary conditions. If we maintain the temporal discretization as the first phase, but discretize the semi-discrete system into a fully discrete system as the follow-up phase, and split the discrete system finally, then we completely eliminate the need of assigning numerical boundary conditions for the Poisson equation. Because, when the discrete system is about to split all boundary information has been included in matrices and vectors. It is noted that, we can interchange the orders of the temporal discretization and the spatial discretization because there are no actions taken between these two phases.

2.7. Some comments on exact factorization technique

Since the 1960s, many techniques have been proposed to resolve the issue of numerical boundary conditions and splitting errors due to splitting methods, which in turn were invented to solve ill-conditioned indefinite systems efficiently, and more importantly, reliably. It turns out that these two issues can be resolved by this simpler exact factorization technique, as well as by two more involved approximate factorization techniques [10,23], in a straightforward and elegant manner. One may notice that the exact factorization approach diverges from the mixed formulation at a very late stage. Computer codes for incompressible flows in mixed formulation can be easily modified to adopt the exact factorization algorithm. As a matter of fact, one may write a linear solver using the exact factorization technique for indefinite systems. It is worthwhile to point out that the exact factorization technique is applicable to many indefinite systems, including incompressible material in solid mechanics, incompressible flows in fluid mechanics, and the divergence-free magnetic flux density in electrodynamics.

The exact factorization technique presented can be implemented in accordance with linear implicit and nonlinear implicit schemes. For a one-step second-order nonlinear implicit scheme using successive substitution, the $\underline{\underline{A}}$ and $\underline{S}^u$ in system (22) are interpreted as

$$\underline{\underline{A}} = \frac{1}{\Delta t^n}\underline{\underline{I}} - \frac{1}{2Re}\underline{\underline{L}} + \frac{1}{2}\underline{\underline{H}}(\underline{u}^{\mathrm{p}}),$$

$$\underline{S}^u = \left(\frac{1}{\Delta t^n}\underline{\underline{I}} + \frac{1}{2Re}\underline{\underline{L}}\right)\underline{u}^n + \frac{1}{Re}\underline{S}^{\mathrm{l}} - \frac{1}{2}\underline{\underline{H}}(\underline{u}^n)\underline{u}^n + \underline{f}^{n+\frac{1}{2}},$$

where $\underline{u}^{\mathrm{p}}$ stands for the predicted solution vector for velocity, and $\underline{\underline{H}}$ in the current context denotes a matrix dependent on discrete velocity vectors. For a two-step linear implicit scheme

$$\underline{u}^{\mathrm{p}} = \left(1 + \frac{\Delta t^n}{\Delta t^{n-1}}\right)u_j^n - \frac{\Delta t^n}{\Delta t^{n-1}}u_j^{n-1}.$$

Approximate techniques [10,23] have to pay attention to the originally used time schemes to maintain time accuracy and avoid excessive operations. In contrast, the time accuracy in exact factorization technique is determined by the original time schemes being selected and remains irrelevant to the technique itself.

If one insists using splitting methods where numerical boundary conditions must be determined, we recommend to use exact factorization technique as a tool to identify the right boundary conditions. For example, if we follow the convention that discrete velocities on boundaries are not included as unknowns, by comparing matrices and vectors generated by Chorin's projection method and the explicit exact factorization finite volume method (EEF-FVM), the right boundary conditions for the intermediate velocity and pressure on solid walls can be identified. It turns out that the boundary conditions for the intermediate velocity and pressure in Chorin's projection method have infinite number of combinations. For simplicity, on solid walls we can impose $u_i^* = 0$ and $\frac{\partial p}{\partial n} = 0$. Boundary conditions on open boundaries can be determined in a similar way.

3. Explicit exact factorization

The system (22) is open for any time scheme. If a first-order explicit time scheme is adopted for some numerical methods (excluding finite element methods), $\underline{\underline{A}} = \frac{1}{\Delta t}\underline{\underline{I}}$ and $\underline{p} = \frac{1}{\Delta t}\underline{\phi}$, we obtain

$$\begin{bmatrix} \frac{1}{\Delta t}\underline{\underline{I}} & \frac{1}{\Delta t}\underline{\underline{G}} \\ \underline{\underline{D}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \frac{1}{\Delta t}\underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\mathrm{d}} \end{Bmatrix}. \tag{34}$$

We may factorize Eq. (34) into

$$\begin{bmatrix} \frac{1}{\Delta t}\underline{\underline{I}} & \underline{\underline{0}} \\ \underline{\underline{D}} & -\underline{\underline{D}}\,\underline{\underline{G}} \end{bmatrix} \begin{bmatrix} \underline{\underline{I}} & \underline{\underline{G}} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \begin{Bmatrix} \underline{u}^{n+1} \\ \Delta t\underline{p}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} \underline{S}^u \\ \underline{S}^{\mathrm{d}} \end{Bmatrix},$$

or equivalently

$$\frac{1}{\Delta t}\underline{u}^* = \underline{S}^u = \frac{1}{\Delta t}\underline{u}^n + \frac{1}{Re}\underline{\underline{L}}\,\underline{u}^n - \underline{\underline{H}}(\underline{u}^n) + \underline{f}^n, \tag{35}$$

$$\underline{\underline{D}}\,\underline{\underline{G}}\left(\Delta t\underline{p}^{n+\frac{1}{2}}\right) = \underline{\underline{D}}\,\underline{u}^* - \underline{S}^{\mathrm{d}}, \tag{36}$$

$$\underline{u}^{n+1} = \underline{u}^* - \underline{\underline{G}}\left(\Delta t\underline{p}^{n+\frac{1}{2}}\right). \tag{37}$$

The system (35) and (37) can be further condensed into

$$\underline{\underline{D}}\,\underline{\underline{G}}\left(\Delta t\underline{p}^{n+\frac{1}{2}}\right) = \underline{\underline{D}}(\Delta t\underline{S}^{u}) - \underline{S}^{d}, \tag{38}$$

$$\underline{u}^{n+1} = \Delta t\underline{S}^{u} - \underline{\underline{G}}\left(\Delta t\underline{p}^{n+\frac{1}{2}}\right), \tag{39}$$

which was also implied in Perot's paper [23]. This reduced system of (38) and (39) is named *explicit exact factorization* (EEF).

3.1. Relation between explicit exact factorization and Chorin's projection method

We first review some key steps in Chorin's projection method [6], which consists of

$$\frac{u_i^* - u_i^n}{\Delta t} = -\frac{\partial}{\partial x_j}(u_j^n u_i^n) + \frac{1}{Re}\frac{\partial^2 u_i^n}{\partial x_j \partial x_j} + f_i, \tag{40}$$

$$\frac{\partial^2 p}{\partial x_j \partial x_j} = \frac{1}{\Delta t}\frac{\partial u_j^*}{\partial x_j}, \tag{41}$$

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{\partial p}{\partial x_i}. \tag{42}$$

Chorin's projection method was derived in a semi-discrete approach. Chorin's original formulation adopted the "convection" form for the non-linear term, which is less advantageous than the conservative "divergence" form presented here. Because the non-conservative form produces a local numerical error, at the same order of the truncation error of the spatial discretization. Such a local numerical error is benign in a smooth flow but may accumulate in case the flow displays large gradients. In the Chorin's projection method, one first explicitly obtains the intermediate velocity field based on Eq. (40), then implicitly solves PPE (41) to obtain the pressure field, and finally the new velocity is updated explicitly by Eq. (42).

Since Eq. (40) is explicit for the intermediate velocity, no numerical boundary condition for u_i^* is needed. On a staggered grid, Eq. (42) does not need any boundary conditions for the pressure. However, in Eq. (41), apart from a numerical boundary condition for pressure, which is typically derived by application of normal component of momentum equation *on* boundaries, a numerical boundary condition for intermediate velocity must also be provided. Eqs. (35)–(37), which are free of numerical boundary condition issue, represent the discrete counterparts of Eqs. (40)–(42) in Chorin's projection method, which demand numerical boundary conditions. Note, by comparing the system (35)–(37), its equivalent system (38) and (39), and the system (40)–(42), it becomes apparent that the intermediate velocity in Chorin's projection method merely serves as a variable for temporary storage.

3.2. Relation between explicit exact factorization and MAC method

The MAC method was introduced in one of the most celebrated papers [17] in computational fluid dynamics (where MAC staggered grid, finite volume method, PPE, and Marker-And-Cell free surface capturing technique were introduced in one single paper), albeit the induction of the issue of numerical boundary condition is a source for debate. The MAC method could have resolved the issue, as we show below while reviewing some key steps of the method.

Harlow and Welch first expressed discrete velocities at time station $n+1$ by discrete velocities at time station n and discrete pressures, symbolically

$$u_i^{n+1} = f(u_i^n, p_j), \tag{43}$$

where we use subscripts to denote all discrete unknowns. The subscript for discrete velocities differs from that for discrete pressures due to the fact that pressure nodes are staggered from velocity nodes in space. At this point, one important aspect of the method must be reminded. The system (43) was intended by Harlow and Welch for *all* discrete velocity nodes, no matter they are well away from boundaries or not.

Then, Harlow and Welch imposed incompressibility at time station $n+1$ using discrete velocities at time station $n+1$

$$D_j(u_i^{n+1}) = 0. \tag{44}$$

Replacing velocities in Eq. (44) by expression (43), Harlow and Welch obtained discrete pressure Poisson equation, which was subsequently in need of the numerical boundary condition.

To avoid the issue of numerical boundary conditions, in the vicinity of boundaries we may supplement Eq. (43) by additional expressions of discrete velocities

$$u_i^{n+1} = g(u_i^n, p_j), \tag{45}$$

where boundary conditions on velocities have been incorporated. With expressions (43) and (45) in derivation of discrete PPE, numerical boundary conditions have no chance to come into play, which results in change of functional structure $f(\ )$ to $g(\ )$. Therefore, in essence our explicit exact factorization finite volume method differs from the original MAC only in respect to the timing of applying boundary conditions.

3.3. Relations among Chorin's projection method, MAC method, and continuous PPE approach

The continuous PPE approach is described as follows. By taking a divergence on the momentum equation

$$\nabla \cdot \left\{\frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u}\vec{u}) = -\nabla p + \frac{1}{Re}\nabla^2 \vec{u} + \vec{f}\right\},$$

and imposing the incompressibility at time level $n+1$

$$\nabla \cdot \vec{u}^{n+1} = 0,$$

we obtain the continuous PPE

$$\nabla^2 p = \nabla \cdot \left\{ \frac{1}{\Delta t} \vec{u}^n - \nabla \cdot (\vec{u}\vec{u})^n + \frac{1}{Re} \nabla^2 \vec{u}^n + \vec{f} \right\}. \tag{46}$$

Discretization of Eq. (46) usually needs boundary conditions for pressure.

Chorin's projection method takes a semi-discrete approach, the MAC method and explicit exact factorization technique take discrete approach, and continuous PPE takes a continuous approach. A discrete approach like MAC method or explicit exact factorization, if wisely implemented, is more advantageous. However, the advantage of MAC method is waived due to the late application of boundary conditions. Consequently, the discrete approach of MAC method [17] turns out to be identical to the continuous PPE approach of Eq. (46). By comparing the first two equations of Chorin's projection method, (40) and (41), and the continuous PPE (46), again we notice that the intermediate velocity merely serves for the temporary storage of some information in the Poisson equation. Hence, in effect Chorin's projection method is also identical to the continuous PPE approach. Thus, the relations among the semi-discrete approach of Chorin's projection method, the fully discrete approach of the MAC method, and the fully continuous PPE approach are revealed. That is, following different methodologies, the three formulations produce identical discrete systems (if the same kind of numerical boundary conditions are adopted). We have to mention that matrices and vectors created in these three formulations have been compared and their agreements support our analysis. A similar view on the relation between MAC method and Chorin's projection method was expressed in [10].

4. Implementation and numerical results

4.1. Implementation

The exact factorization technique involves sparse matrix–matrix multiplications, which only costs $\mathcal{O}(N)$ operations, where N stands for the size of a matrix. Two flows are tested on the standard MAC staggered grid with a second-order semi-implicit exact factorization finite volume method (SIEF-FVM), a second-order explicit exact factorization finite volume method (EEF-FVM), a second-order linear implicit exact factorization finite volume method (LIEF-FVM), and a one-step second-order fully non-linear implicit exact factorization finite volume method (NIEF-FVM). Both pressure-update and pressure-correction formulations are implemented and to reach the same accuracy they do not display sizable difference in performance. In all calculations, results based on SIEF-FVM, EEF-FVM, LIEF-FVM, NIEF-FVM agree with one another extremely well. Two recent non-stationary linear solvers, BiCGStab [28] and GPBiCG(m,l) [30,11] are also employed for the well-conditioned systems but they do not display superior performance over a simple classical stationary linear solver, Gauss-Seidel.

The criterion for convergence of steady solution and for the Gauss-Seidel linear solver is set as

$$\max \left(\frac{|x^{k+1} - x^k|}{1.0 + |x^{k+1}|} \right) < 10^{-7},$$

where k denotes the iteration level. For stability reason in the case of EF-FVM, if a forward Euler is used the time step may satisfy

$$\Delta t < \min \left(\frac{Re}{4} h_x^2, \frac{Re}{4} h_y^2, \frac{4}{Re(|U| + |V|)^2} \right), \tag{47}$$

where h_x and h_y are defined in Fig. 2, and U and V represent local velocity components. Criterion (47) is derived by fixing local convective velocities as constants then using Fourier stability analysis. For flow problems being considered in lid-driven cavity flow, we simply take $|U| = |V| = 1$. In the code, the time step is taken half of the value satisfying criterion (47), since the second-order Adams–Bashforth being used has a smaller convergence radius.

4.2. Cavity flow

Cavity flow possesses corner singularities which result in non-smoothness of the solution in some regions. Our numerical experiments with mixed formulation show that at high Re or with very fine grids the large discrete system fails to produce sensible results, due to deteriorating conditioning number of the indefinite system. On the other hand, the exact factorization technique is designed to solve incompressible flows efficiently, and more importantly, reliably. Hence, a cavity flow under a fine grid is an ideal case to show the success of the exact factorization technique.

Velocity profiles on the vertical midplane in a *square* cavity flow are calculated on the configuration illustrated

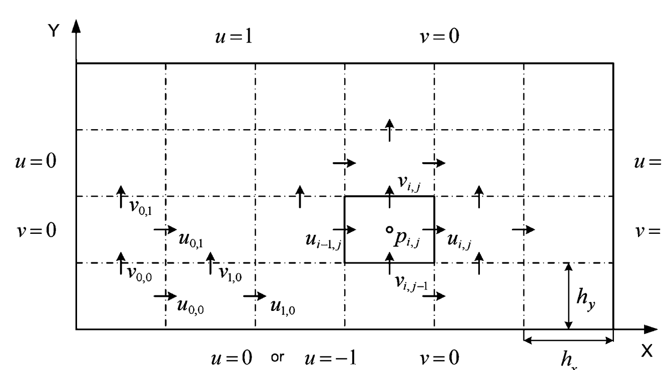

Fig. 2. Computational domain for the cavity flow simulation.

Table 1
Spatial convergence rate of square cavity flow at $Re = 400$

Grid	32 × 32	64 × 64	128 × 128
u at $y = 0.015625$	−0.05407	−0.05804	−0.05909

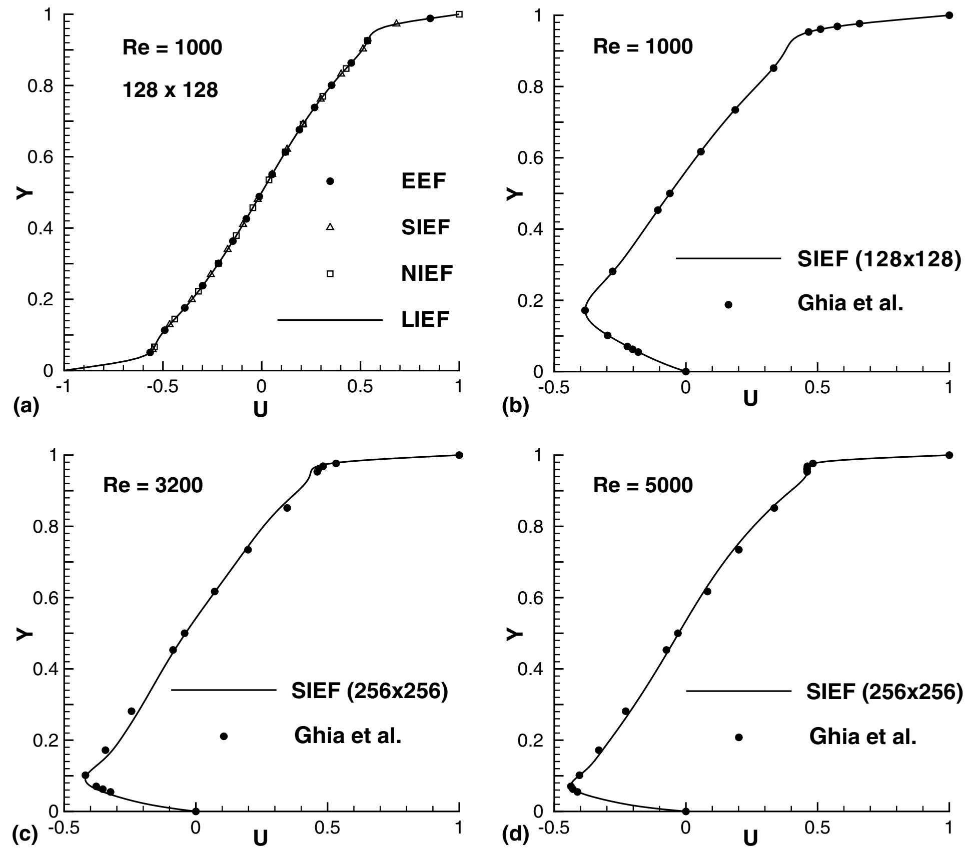

Fig. 3. Comparisons of mid-plane x-component velocity for the square cavity flow for (a) two-sided, (b)–(d) one-sided.

in Fig. 2. The spatial convergence rate is checked at a point close to the lower (at $y = 0.015625$) wall as displayed in Table 1, where u_1 is the result based on a 32×32 grid and likewise u_2 and u_3 are based on two other grids. The results show that $\frac{|u_1-u_2|}{|u_2-u_3|} = 4.0$, which indicates second-order convergence rate. Fig. 3(a) shows that our results based on four different time schemes are in excellent agreement with each other. Fig. 3(b) and (d) compares the results based on SIEF-FVM with those from a multigrid finite difference method in vorticity-streamfunction (VS) formulation [12]. At $Re = 1000$, these results match each other very well. At $Re = 3200$ and 5000 our results based on SIEF-FVM (on 256×256) differ from Ghia's (on 128×128) to some extent. There are two possible reasons for these slight discrepancies. As Re increases, a finer grid becomes more necessary, while the grid used in VS formulation [12] for $Re = 3200$ and 5000 remains the same for all calculations. Also, the numerical boundary conditions for vorticity, which is unavoidable in VS formulation, may introduce some appreciable errors.

4.3. Periodic cavity flow

We consider a transient Navier–Stokes flow inside a unity square cavity with all boundaries fixed, but under the influence of a supplied body force. The time-periodic exact solution is prescribed similar to [19]

$$u(x,y,t) = -\sin t\ \sin^2 \pi x \sin \pi y \cos \pi y,$$
$$v(x,y,t) = \sin t\ \sin \pi x \cos \pi x \sin^2 \pi y,$$
$$p(x,y,t) = \sin t\ \sin \pi x \cos \pi y,$$

which vanish at $t = 0$ and vanish on four boundaries $x = 0$, $x = 1$, $y = 0$, and $y = 1$. The appropriate body force functions can be derived by substituting exact solutions into the momentum equation (1),

$$\begin{aligned} f_x = & -\cos t \sin^2 \pi x \sin \pi y \cos \pi y + \pi \sin^2 t \sin^3 \pi x \cos \pi x \sin^2 \pi y \\ & + \pi \sin t \cos \pi x \cos \pi y \\ & - \frac{\pi^2}{Re} \sin t\,(6 \sin^2 \pi x - 2\cos^2 \pi x) \sin \pi y \cos \pi y, \\ f_y = & \cos t \sin \pi x \cos \pi x \sin^2 \pi y + \pi \sin^2 t \sin^2 \pi x \sin^3 \pi y \cos \pi y \\ & - \pi \sin t \sin \pi x \sin \pi y \\ & + \frac{\pi^2}{Re} \sin t \sin \pi x \cos \pi x\ (6 \sin^2 \pi y - 2\cos^2 \pi y). \end{aligned}$$

Comparisons between numerical solutions and exact solutions in Fig. 4 show excellent agreements.

We also employ the periodic cavity flow to examine time accuracy. The mid-plane x-component velocity is calculated at $t = 0.64$ with three different time steps, 0.0004, 0.0002, and 0.0001. Four different rules are used for time accuracy study and they are displayed in Table 2, and

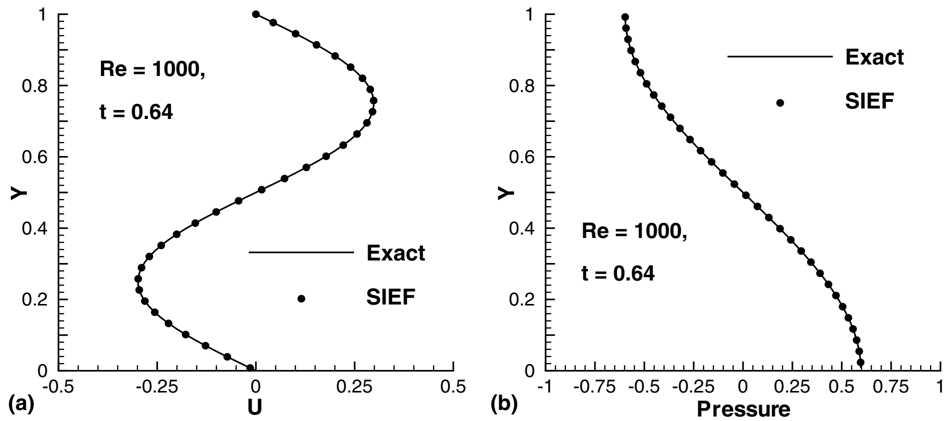

Fig. 4. Comparisons of numerical results (on a 64×64 grid) with exact solutions for the square periodic cavity flow: (a) mid-plane x-component velocity and (b) mid-plane pressure.

Table 2
Temporal convergence rate of periodic cavity flow at $Re = 1000$ on a 64×64 grid

	Rule II: $\lambda \equiv \frac{\|\|u - u_{\text{exact}}\|\|}{\|\|u_{\text{exact}}\|\|}$	Rule III: $\lambda \equiv \max\left(\frac{\|u - u_{\text{exact}}\|}{1 + \|u_{\text{exact}}\|}\right)$	Rule IV: $\lambda \equiv u - u_{\text{exact}}$
λ for $\Delta t = 0.0004$	5.795×10^{-4}	2.143×10^{-4}	1.587×10^{-6}
λ for $\frac{1}{2}\Delta t$	1.479×10^{-4}	5.38×10^{-5}	7.237×10^{-6}
λ for $\frac{1}{4}\Delta t$	4.22×10^{-5}	1.36×10^{-5}	8.474×10^{-6}
$\frac{\|\lambda(\Delta t) - \lambda(\frac{1}{2}\Delta t)\|}{\|\lambda(\frac{1}{2}\Delta t) - \lambda(\frac{1}{4}\Delta t)\|}$	4.08	3.99	4.57

In Rule IV, u and u_{exact} are evaluated at $y = 0.0078125$.

$$||u|| \equiv \frac{1}{k}\sqrt{\sum_k u_k^2},$$

where k denotes number of points being evaluated along the mid-plane. Rule I is a self-checking of the convergence rate and does not refer to the exact solution, with the result $\frac{||u(\Delta t) - u(\frac{1}{2}\Delta t)||}{||u(\frac{1}{2}\Delta t) - u(\frac{1}{4}\Delta t)||} = 3.97$. In contrast, Rules II, III, and IV refer to the exact solution hence a discrepancy between the numerical solution and the exact solution can be displayed in Table 2 for each time step size. In rule IV of the pointwise convergence rate examination, a point very close to boundary ($y = 0.0078125$) is chosen to make sure the second-order accuracy is maintained even in the boundary layer. Note, in rule IV we may take an equivalent definition of $\lambda \equiv u$ so that, as in rule I, the exact solution is no longer required. Clearly, Table 2 and the result from rule I show that the exact factorization technique can break up the indefinite system into well-conditioned smaller ones without sacrificing the original time accuracy, which is second-order in our implementation.

5. Conclusions

In this paper, we have presented the technique of exact factorization to treat indefinite systems, especially for those arising from incompressible flows. The technique was particularly inspired by works documented in [21,10,23]. As a matter of fact, the idea of introducing a numerical pressure [21] and the idea of fully discretizing continuous systems before any further operations [10,23] should be regarded as two cornerstones of the current technique. The technique presented here is independent of selected time schemes and does not involve approximate inversion of matrices. The technique is free of numerical boundary condition issue simply because the splitting happens *after* the temporal and spatial discretizations. In addition, some well-known techniques are compared and their close relation to the explicit exact factorization technique is elucidated. Finally, some numerical evidences are provided to support analysis made in this paper. Implementation of the exact factorization technique follows the mixed formulation methods till a very late stage; therefore, codes written in traditional finite element methods, spectral methods, or other spatial methods can be easily converted into the exact factorization manner. However, some challenges related to the pressure still remain, such as the need of a rapid Poisson solver and a unified approach to tackle both incompressible and compressible flows.

Acknowledgement

The support for this work was in part provided by grant CTS-0237951 from the National Science Foundation.

References

[1] J.B. Bell, P. Colella, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.

[2] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1982) 199–259.

[3] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.

[4] J. Cahouet, J.P. Chabard, Some fast 3D finite element solvers for the generalized stokes problem, Int. J. Numer. Methods Fluids 8 (1988) 869–895.

[5] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[6] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[7] A.J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, Math. Comput. 23 (1969) 341–353.

[8] M. Deville, L. Kleiser, F. Montigny-Rannou, Pressure and time treatment for Chebyshev spectral solution of a stokes problem, Int. J. Numer. Methods Fluids 4 (1984) 1149–1163.

[9] M.O. Deville, P.F. Fischer, E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, 2002.

[10] J.K. Dukowicz, A.S. Dvinsky, Approximate factorization as a high-order splitting for the implicit incompressible flow equations, J. Comput. Phys. 102 (1992) 336–347.

[11] S. Fujino, GPBiCG(m,l): a hybrid of BiCGStab and GPBiCG methods with efficiency and robustness, Appl. Numer. Math. 41 (2002) 107–117.

[12] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[13] P.M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: theory, Int. J. Numer. Methods Fluids 11 (1990) 587–620.

[14] P.M. Gresho, Incompressible fluid dynamics: some fundamental formulation issues, Ann. Rev. Fluid Mech. 23 (1991) 413–453.

[15] P.M. Gresho, R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 7 (1987) 1111–1145.

[16] J.L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, J. Comput. Phys. 192 (2003) 262–276.

[17] F.H. Harlow, J.F. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[18] T.J.R. Hughes, W.K. Liu, A. Brooks, Finite element analysis of incompressible viscous flows by the penalty function formulation, J. Comput. Phys. 30 (1979) 1–60.

[19] H. Johnston, J.G. Liu, Finite difference schemes for incompressible flow based on local pressure boundary conditions, J. Comput. Phys. 180 (2002) 120–154.

[20] G.E.M. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414–443.

[21] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.

[22] P. Moin, J. Kim, On the numerical solution of time-dependent viscous incompressible fluid flows involving solid boundaries, J. Comput. Phys. 35 (1980) 381–392.

[23] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.

[24] R. Peyret, D. Taylor, Computational Methods for Fluid Flow, Springer-Verlag, Berlin, 1983.

[25] A. Quarteroni, F. Saleri, A. Veneziani, Factorization methods for the numerical approximation of Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 188 (2000) 505–526.

[26] P.G. Saffman, Vortex Dynamics, Cambridge University Press, 1992 (Chapter one).

[27] J.C. Strikwerda, Y.S. Lee, The accuracy of the fractional step method, SIAM J. Numer. Anal. 37 (1999) 37–47.

[28] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.

[29] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.

[30] S.L. Zhang, GPBi-CG: generalized product-type methods based on Bi-CG for solving non-symmetric linear systems, SIAM J. Sci. Comput. 18 (1997) 537–551.