Numer Algor (2014) 65:43–68
DOI 10.1007/s11075-013-9694-y

ORIGINAL PAPER

On approximated ILU and UGS preconditioning methods for linearized discretized steady incompressible Navier-Stokes equations

Zhong-Zhi Bai · Yu-Hong Ran · Li Yuan

Received: 12 April 2012 / Accepted: 18 January 2013 / Published online: 2 March 2013

Abstract When the artificial compressibility method in conjunction with high-order upwind compact finite difference schemes is employed to discretize the steady-state incompressible Navier-Stokes equations, in each pseudo-time step we need to solve a structured system of linear equations approximately by, for example, a Krylov subspace method such as the preconditioned GMRES. In this paper, based on the special structure and concrete property of the linear system we construct a structured preconditioner for its coefficient matrix and estimate eigenvalue bounds of the correspondingly preconditioned matrix. Numerical examples are given to illustrate the effectiveness of the proposed preconditioning methods.

Keywords Incompressible Navier-Stokes equations · Artificial compressibility method · Upwind compact finite difference scheme · Preconditioning · Krylov subspace method.

Mathematics Subject Classfication (2010) 65F10 · 65F15 · 76D05

Supported by The Hundred Talent Project of Chinese Academy of Sciences, The National Natural Science Foundation for Creative Research Groups (No. 11021101), The National Natural Science Foundation (No. 91118001, No. 10972230 and No. 11021101), and The National Basic Research Program (No. 2011CB309703 and No. 2010CB731505), P.R. China.

Z.-Z. Bai (✉) · Y.-H. Ran · L. Yuan
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, People's Republic of China
e-mail: bzz@lsec.cc.ac.cn

Y.-H. Ran
e-mail: ranyh@lsec.cc.ac.cn

L. Yuan
e-mail: lyuan@lsec.cc.ac.cn

1 Introduction

A wide spectrum of fluid flow problems can be described mathematically through the incompressible Navier-Stokes equations, which can be solved numerically by the directly coupled method, the pressure-correction method, and the projection method. These methods may be categorized into the methodology of self-consistent iterations, which computes approximate solutions by alternatively solving the pressure (or the pressure-correction) Poisson equation and the momentum equation, and in general they can be employed to solve both steady-state and time-dependent incompressible Navier-Stokes equations; see [18, 19, 23, 25]. However, for steady-state incompressible Navier-Stokes equations, the artificial compressibility method [18] is cost-effective and, hence, is frequently used in engineering computing. The main reason is that for using the artificial compressibility method we change the governing equations to hyperbolic ones by introducing an extra pseudo-time derivative with respect to the pressure into the continuity equation so that accurate discretization schemes and efficient solution methods developed for computing compressible flows can be straightforwardly utilized.

Compared with numerical solutions for the compressible flows, a main difficulty in numerically solving the incompressible Navier-Stokes equations is the lack of a time-derivative term in the continuity equation, which limits straightforward applications of the time-marching methods [19]. The artificial compressibility method overcomes this difficulty by introducing an extra pseudo-time derivative into the steady-state incompressible Navier-Stokes equations, so that the resulting equations are coupled in a time-marching manner. Alternatively, the mixed finite element method may be used to solve the steady-state incompressible Navier-Stokes equations, inducing a discretized linear system of the saddle-point form. As is well-known, the saddle-point matrix is indefinite and ill-conditioned, so computing an approximate solution to the saddle-point linear system is practically a very difficult problem and numerically a challenging task; see [4, 5, 7, 10, 12, 20] and the references therein.

Recently, based on the artificial compressibility method, in [30–32] the authors proposed and studied several simple and accurate discretization methods by making use of the third- and the fifth-order upwind compact finite difference schemes. After discretization, at each pseudo-time level an approximate solution of the discretized steady-state incompressible Navier-Stokes equations can be computed by the *approximate factorization and alternating direction implicit* (**AF-ADI**) iteration, the *lower and upper symmetric Gauss-Seidel* (**LU-SGS**) iteration, and the line relaxation method; see [15, 24, 27, 33]. These approaches are essentially based on approximate factorizations of the discretized incompressible Navier-Stokes equations. They may, however, cause errors related to the pseudo-time stepsizes and the artificial compressibility factor, which could affect the overall computational accuracy of the discretized solution and the global convergent rate of the iteration process.

Through a different approach, recently Ran and Yuan [28] fully discretized and linearized the two-dimensional steady-state incompressible Navier-Stokes equations by making use of the artificial compressibility method at each pseudo-time level on quadrilateral meshes, resulting in a class of large, sparse, and structured systems of linear equations. The discretized solution of the steady-state incompressible viscous flow problem can then be obtained via solving these linear systems by employing the *modified block symmetric successive overrelaxation* (**MBSSOR**) and the *modified alternating direction implicit* (**MADI**) iteration methods, respectively; see [1, 3, 26]. Both MBSSOR and MADI iteration methods require low computational complexity and small computer memory in actual implementations. However, finding the optimal iteration parameters for these two iteration methods is practically a challenging task.

Note that for any real ℓ-by-ℓ banded matrix $\mathbf{A} \in \mathbb{R}^{\ell\times\ell}$ (with a fixed small bandwidth) and any real ℓ-dimensional vector $\mathbf{b} \in \mathbb{R}^{\ell}$ the matrix-vector product $\mathbf{Ab}$ may be computed in $\mathcal{O}(\ell)$ operations. Hence, we can employ Krylov subspace iteration methods such as GMRES [29] to solve the system of linear equations $\mathbf{Ax} = \mathbf{b}$ in an economical cost. To further accelerate the convergence rates of the Krylov subspace methods, we need to construct economical and high-quality preconditioners for the matrix $\mathbf{A}$.

Along this approach, in this paper we construct and analyze approximated *incomplete LU* (**ILU**) factorization and approximated *unsymmetric Gauss-Seidel* (**UGS**) splitting preconditioning matrices for the coefficient matrix $\mathbf{A}^n$ of the discretized linear system $\mathbf{A}^n\,\Delta\mathbf{q}^n = \widetilde{\mathbf{g}}^n$. These preconditioning matrices essentially consist of two approximation steps: First, selectively dropping the off-diagonal elements of $\mathbf{A}^n$ to obtain an approximate matrix $\mathbf{P}^n$; Second, constructing ILU factorization and UGS splitting for the matrix $\mathbf{P}^n$ to obtain the ILU and the UGS preconditioning matrices for the matrix $\mathbf{A}^n$. The matrix $\mathbf{P}^n$ is much sparser than the matrix $\mathbf{A}^n$, so that the ILU and the UGS preconditioning matrices are also sparser than the ILU factorization and the UGS splitting preconditioning matrices straightforwardly computed from $\mathbf{A}^n$.

Theoretically, we prove that both $\mathbf{A}^n$ and $\mathbf{P}^n$ are strictly diagonally dominant matrices by columns under certain restrictions on the spatial and the temporal stepsizes, and all eigenvalues of the preconditioned matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ are located within a complex disk centered at $(1, 0)$ with radius being less than 1, which conditionally guarantees the convergence of the GMRES method preconditioned by $\mathbf{P}^n$ when it is used to solve the discretized linear system $\mathbf{A}^n\,\Delta\mathbf{q}^n = \widetilde{\mathbf{g}}^n$; see [2, 11, 22, 29]. Numerically, we show that $\mathbf{P}^n$ possesses more desirable sparsity pattern and eigenvalue distribution than $\mathbf{A}^n$, and the eigenvalues of the preconditioned matrices with respect to the ILU and the UGS preconditioners are tightly clustered. Therefore, when the eigenvector matrices of the preconditioned matrices are not very ill-conditioned, the corresponding preconditioned GMRES methods are convergent fast, accurately and robustly to the exact solution of the discretized linear system $\mathbf{A}^n\,\Delta\mathbf{q}^n = \widetilde{\mathbf{g}}^n$, resulting in a reliable and effective numerical process, including the discretization, the linearization and the GMRES solve, for solving the two-dimensional steady-state incompressible Navier-Stokes equations; see [2, 22].

The paper is organized as follows. In Section 2, we briefly describe the governing equations, their finite difference discretizations, and the correspondingly induced system of linear equations. The strictly diagonal dominance of the linear system is also demonstrated in this section. In Section 3, we construct the ILU and the UGS preconditioning matrices for the matrix $\mathbf{A}^n$ and demonstrate the strictly diagonal dominance of the approximate matrix $\mathbf{P}^n$. The numerical results about the plane Poiseuille flow, the plane Couette-Poiseuille flow and the modified cavity flow are reported in Section 4. Finally, we give some concluding remarks in Section 5.

2 The governing equations and discretization

In this section, we will describe the governing equations, i.e., the two-dimensional steady-state incompressible Navier-Stokes equations, and derive the corresponding discretized linear system resulting from a technical combination of the artificial compressibility method and the high-order upwind compact finite difference scheme. For more details, we refer to [28].

2.1 The governing equations

The governing two-dimensional steady-state incompressible Navier-Stokes equations, in Cartesian coordinates (x, y) and without body force, are of the following dimensionless form:

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} = 0, \\ \dfrac{\partial u^2}{\partial x} + \dfrac{\partial uv}{\partial y} = -\dfrac{\partial p}{\partial x} + \dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), \\ \dfrac{\partial uv}{\partial x} + \dfrac{\partial v^2}{\partial y} = -\dfrac{\partial p}{\partial y} + \dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right), \end{cases} \tag{2.1}$$

where u and v are the velocity components, p is the pressure, and Re is the Reynolds number. By introducing a pseudo-time derivative into the continuity and the momentum equations, we further modify (2.1) to obtain the following target equations:

$$\begin{cases} \dfrac{\partial p}{\partial \tau} + \beta\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) = 0, \\ \dfrac{\partial u}{\partial \tau} + \dfrac{\partial u^2}{\partial x} + \dfrac{\partial uv}{\partial y} = -\dfrac{\partial p}{\partial x} + \dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), \\ \dfrac{\partial v}{\partial \tau} + \dfrac{\partial uv}{\partial x} + \dfrac{\partial v^2}{\partial y} = -\dfrac{\partial p}{\partial y} + \dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right), \end{cases} \tag{2.2}$$

where τ is the pseudo time and β is the artificial compressibility factor. Note that when $\tau \to \infty$, the equations in (2.2) approach to those in (2.1) or, in other words, the non-steady state solution of the equations in (2.2) asymptotically tends to the steady state solution of the equations in (2.1). We remark that the value of β is important for the numerical performance of the artificial compressibility method.

Denote by

$$\mathbf{q} = \begin{pmatrix} p \\ u \\ v \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} \beta u \\ u^2 + p \\ uv \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \beta v \\ uv \\ v^2 + p \end{pmatrix}$$

and

$$\mathbf{e}_\nu = \frac{1}{\mathrm{Re}}\begin{pmatrix} 0 \\ u_x \\ v_x \end{pmatrix}, \quad \mathbf{f}_\nu = \frac{1}{\mathrm{Re}}\begin{pmatrix} 0 \\ u_y \\ v_y \end{pmatrix},$$

where $\mathbf{q}$ represents the solution vector, $\mathbf{e}$, $\mathbf{f}$ and $\mathbf{e}_\nu$, $\mathbf{f}_\nu$ stand for the inviscid and the viscous flux vectors, and u_x, v_x and u_y, v_y denote the first-order derivatives of u, v with respect to x and y, respectively. Then we can rewrite the target equations in (2.2) into a unified vector form as follows:

$$\frac{\partial \mathbf{q}}{\partial \tau}+\frac{\partial(\mathbf{e}-\mathbf{e}_\nu)}{\partial x}+\frac{\partial(\mathbf{f}-\mathbf{f}_\nu)}{\partial y}=0. \tag{2.3}$$

By direct computations we know that the Jacobian matrices $\mathbf{J}_\mathbf{e}$ and $\mathbf{J}_\mathbf{f}$ of the inviscid flux vectors $\mathbf{e}$ and $\mathbf{f}$ are given by

$$\mathbf{J}_\mathbf{e}=\frac{\partial \mathbf{e}}{\partial \mathbf{q}}=\begin{pmatrix}0 & \beta & 0\\ 1 & 2u & 0\\ 0 & v & u\end{pmatrix} \quad \text{and} \quad \mathbf{J}_\mathbf{f}=\frac{\partial \mathbf{f}}{\partial \mathbf{q}}=\begin{pmatrix}0 & 0 & \beta\\ 0 & v & u\\ 1 & 0 & 2v\end{pmatrix},$$

and the Jacobian matrices $\mathbf{J}_{\mathbf{e}\nu}$ and $\mathbf{J}_{\mathbf{f}\nu}$ of the viscous flux vectors $\mathbf{e}_\nu$ and $\mathbf{f}_\nu$ are given by

$$\mathbf{J}_{\mathbf{e}\nu}=\frac{\partial \mathbf{e}_\nu}{\partial \mathbf{q}}=\frac{1}{\mathrm{Re}}\mathbf{I}_m\frac{\partial}{\partial x} \quad \text{and} \quad \mathbf{J}_{\mathbf{f}\nu}=\frac{\partial \mathbf{f}_\nu}{\partial \mathbf{q}}=\frac{1}{\mathrm{Re}}\mathbf{I}_m\frac{\partial}{\partial y},$$

with

$$\mathbf{I}_m=\begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}.$$

The Jacobian matrices $\mathbf{J}_\mathbf{e}$ and $\mathbf{J}_\mathbf{f}$ admit the spectral decompositions

$$\mathbf{J}_\mathbf{e}=\Phi_\mathbf{e}\Lambda_\mathbf{e}\Phi_\mathbf{e}^{-1} \quad \text{and} \quad \mathbf{J}_\mathbf{f}=\Phi_\mathbf{f}\Lambda_\mathbf{f}\Phi_\mathbf{f}^{-1},$$

where

$$\Lambda_\mathbf{e}=\mathrm{diag}(u-c_1,u,u+c_1) \quad \text{and} \quad \Lambda_\mathbf{f}=\mathrm{diag}(v-c_2,v,v+c_2)$$

are diagonal matrices containing the eigenvalues, with

$$c_1=\sqrt{u^2+\beta} \quad \text{and} \quad c_2=\sqrt{v^2+\beta},$$

and $\Phi_\mathbf{e}$ and $\Phi_\mathbf{f}$ are the matrices corresponding to the right eigenvectors, of the matrices $\mathbf{J}_\mathbf{e}$ and $\mathbf{J}_\mathbf{f}$, respectively.

2.2 The temporal and spatial discretizations

Approximating the pseudo-time derivative $\frac{\partial \mathbf{q}}{\partial \tau}$ by the first-order backward difference, from (2.3) we can obtain an implicit difference scheme

$$\frac{\Delta \mathbf{q}^n}{\Delta \tau}=-\left[\frac{\partial(\mathbf{e}-\mathbf{e}_\nu)}{\partial x}+\frac{\partial(\mathbf{f}-\mathbf{f}_\nu)}{\partial y}\right]^{n+1}, \tag{2.4}$$

where n represents the pseudo-time level (the number of iterations), $\Delta\tau$ is the pseudo-time stepsize determined according to the CFL number, and $\Delta\mathbf{q}^n=\mathbf{q}^{n+1}-\mathbf{q}^n$. Because the first-order Taylor expansions of $\mathbf{e}$, $\mathbf{f}$ and $\mathbf{e}_\nu$, $\mathbf{f}_\nu$ are given by

$$\mathbf{e}^{n+1}\approx\mathbf{e}^n+\mathbf{J}_\mathbf{e}^n\Delta\mathbf{q}^n, \quad \mathbf{f}^{n+1}\approx\mathbf{f}^n+\mathbf{J}_\mathbf{f}^n\Delta\mathbf{q}^n$$

and

$$\mathbf{e}_\nu^{n+1} \approx \mathbf{e}_\nu^n + \mathbf{J}_{\mathbf{e}\nu}^n \Delta \mathbf{q}^n, \quad \mathbf{f}_\nu^{n+1} \approx \mathbf{f}_\nu^n + \mathbf{J}_{\mathbf{f}\nu}^n \Delta \mathbf{q}^n,$$

after substitution of these approximations into (2.4) we have

$$\left[\mathbf{I} + \Delta\tau\left(\frac{\partial(\mathbf{J_e} - \mathbf{J_{e\nu}})}{\partial x} + \frac{\partial(\mathbf{J_f} - \mathbf{J_{f\nu}})}{\partial y}\right)\right]^n \Delta\mathbf{q}^n = -\Delta\tau\left[\frac{\partial(\mathbf{e} - \mathbf{e}_\nu)}{\partial x} + \frac{\partial(\mathbf{f} - \mathbf{f}_\nu)}{\partial y}\right]^n,$$

where $\mathbf{I}$ denotes the identity matrix. If we further denote by

$$\mathbf{g}^n = -\Delta\tau\left[\frac{\partial(\mathbf{e} - \mathbf{e}_\nu)}{\partial x} + \frac{\partial(\mathbf{f} - \mathbf{f}_\nu)}{\partial y}\right]^n, \tag{2.5}$$

then the above equation can be briefly rewritten as

$$\left[\mathbf{I} + \Delta\tau\left(\frac{\partial(\mathbf{J_e} - \mathbf{J_{e\nu}})}{\partial x} + \frac{\partial(\mathbf{J_f} - \mathbf{J_{f\nu}})}{\partial y}\right)\right]^n \Delta\mathbf{q}^n = \mathbf{g}^n. \tag{2.6}$$

Owing to the hyperbolic nature of the Eq. (2.3), we can split the convective flux derivative $\mathbf{e}_x = \frac{\partial \mathbf{e}}{\partial x}$ in (2.5) into two parts $\mathbf{e}_x^+$ and $\mathbf{e}_x^-$ along the positive and the negative x-directions with respect to the positive and the negative eigenvalues of $\mathbf{J_e}$, respectively. That is to say,

$$\mathbf{e}_x = \mathbf{e}_x^+ + \mathbf{e}_x^-,$$

with $\mathbf{e}_x^+$ and $\mathbf{e}_x^-$ being the split flux derivatives propagating information from left to right and from right to left, respectively. The two derivatives $\mathbf{e}_x^+$ and $\mathbf{e}_x^-$ can be computed by the third- and the fifth-order upwind compact difference schemes defined as

$$\begin{cases} \frac{2}{3}\left(\mathbf{e}_x^+\right)_i + \frac{1}{3}\left(\mathbf{e}_x^+\right)_{i-1} = \frac{1}{6\,\Delta x}\left(5\delta^-\mathbf{e}_i^+ + \delta^-\mathbf{e}_{i+1}^+\right), \\ \frac{2}{3}\left(\mathbf{e}_x^-\right)_i + \frac{1}{3}(\mathbf{e}_x^-)_{i+1} = \frac{1}{6\,\Delta x}\left(5\delta^+\mathbf{e}_i^- + \delta^+\mathbf{e}_{i-1}^-\right) \end{cases} \tag{2.7}$$

and

$$\begin{cases} \frac{3}{5}\left(\mathbf{e}_x^+\right)_i + \frac{2}{5}\left(\mathbf{e}_x^+\right)_{i-1} = \frac{1}{60\,\Delta x}\left(-\delta^-\mathbf{e}_{i+2}^+ + 11\delta^-\mathbf{e}_{i+1}^+ + 47\delta^-\mathbf{e}_i^+ + 3\delta^-\mathbf{e}_{i-1}^+\right), \\ \frac{3}{5}\left(\mathbf{e}_x^-\right)_i + \frac{2}{5}\left(\mathbf{e}_x^-\right)_{i+1} = \frac{1}{60\,\Delta x}\left(-\delta^+\mathbf{e}_{i-2}^- + 11\delta^+\mathbf{e}_{i-1}^- + 47\delta^+\mathbf{e}_i^- + 3\delta^+\mathbf{e}_{i+1}^-\right), \end{cases} \tag{2.8}$$

respectively. Here Δx denotes the grid spacing, and we have used the notations

$$\delta^+ f_i = f_{i+1} - f_i \quad \text{and} \quad \delta^- f_i = f_i - f_{i-1}$$

for any given sequence $\{f_i\}$ of real numbers. Noticing that each term in the right-hand sides of the Eqs. (2.7)–(2.8) represents the difference of the split fluxes between two neighboring points, we may compute them by using the flux difference splitting

$$\mathbf{e}_{i+1}^\pm - \mathbf{e}_i^\pm \equiv \Delta\mathbf{e}_{i+\frac{1}{2}}^\pm = \mathbf{J}_\mathbf{e}^\pm(\overline{\mathbf{q}})(\mathbf{q}_{i+1} - \mathbf{q}_i),$$

where $\Delta\mathbf{e}_{i+\frac{1}{2}}^\pm$ are the flux differences across the positive and the negative traveling waves, and the split Jacobian matrices $\mathbf{J}_\mathbf{e}^\pm(\overline{\mathbf{q}})$ are defined by

$$\mathbf{J}_\mathbf{e}^\pm = \Phi_\mathbf{e}\Lambda_\mathbf{e}^\pm\Phi_\mathbf{e}^{-1},$$

with

$$\Lambda_{\mathbf{e}}^{\pm} = \frac{1}{2}(\Lambda_{\mathbf{e}} \pm |\Lambda_{\mathbf{e}}|)$$

being evaluated using the Roe average value $\overline{\mathbf{q}} = \frac{1}{2}(\mathbf{q}_{i+1} + \mathbf{q}_i)$ for incompressible flow.

In addition, we approximate the derivative $\frac{\partial \mathbf{e}_v}{\partial x}$ in (2.5) by the fourth- and the sixth-order symmetric compact difference schemes at interior points. For example, for $s_i \approx \left(\frac{\partial^2 u}{\partial x^2}\right)_i$, it holds that

$$\frac{1}{12}(s_{i-1} + 10 s_i + s_{i+1}) = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}$$

and

$$2s_{i-1} + 11 s_i + 2 s_{i+1} = 12\frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + \frac{3}{4}\frac{u_{i-2} - 2u_i + u_{i+2}}{\Delta x^2}.$$

In an analogous fashion, we can obtain approximations to the convective flux $\mathbf{f}_y = \frac{\partial \mathbf{f}}{\partial y}$, the second-order derivative $\frac{\partial^2 v}{\partial x^2}$ and the derivative for the viscous term $\frac{\partial \mathbf{f}_v}{\partial y}$ in (2.5).

Moreover, we discretize the convective terms and the viscous terms in the left-hand side of (2.6) by the first-order upwind difference and the central difference, respectively, which are defined as

$$\delta_x^+ f_i = \frac{f_{i+1} - f_i}{\Delta x}, \quad \delta_x^- f_i = \frac{f_i - f_{i-1}}{\Delta x} \quad \text{and} \quad \delta_x^2 f_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2}$$

for any given sequence $\{f_i\}$ of real numbers and any given stepsize Δx.

Hence, through the above-mentioned finite difference discretization we can obtain the following approximation to the Eq. (2.3):

$$\begin{aligned}\Bigg[\mathbf{I} + \Delta\tau\left(\delta_x^- \widetilde{\mathbf{J}}_{\mathbf{e}}^+ + \delta_x^+ \widetilde{\mathbf{J}}_{\mathbf{e}}^- - \frac{1}{\mathrm{Re}}\mathbf{I}_m \delta_x^2\right) \\ + \Delta\tau\left(\delta_y^- \widetilde{\mathbf{J}}_{\mathbf{f}}^+ + \delta_y^+ \widetilde{\mathbf{J}}_{\mathbf{f}}^- - \frac{1}{\mathrm{Re}}\mathbf{I}_m \delta_y^2\right)\Bigg]^n \Delta\mathbf{q}^n = \widetilde{\mathbf{g}}^n, \end{aligned} \tag{2.9}$$

where

$$\widetilde{\mathbf{J}}_{\mathbf{e}}^{\pm} = \frac{1}{2}(\mathbf{J}_{\mathbf{e}} \pm \varrho(\mathbf{J}_{\mathbf{e}})\mathbf{I}), \qquad \widetilde{\mathbf{J}}_{\mathbf{f}}^{\pm} = \frac{1}{2}(\mathbf{J}_{\mathbf{f}} \pm \varrho(\mathbf{J}_{\mathbf{f}})\mathbf{I})$$

and

$$\varrho(\mathbf{J}_{\mathbf{e}}) = \varkappa \cdot \max|\lambda(\mathbf{J}_{\mathbf{e}})|, \qquad \varrho(\mathbf{J}_{\mathbf{f}}) = \varkappa \cdot \max|\lambda(\mathbf{J}_{\mathbf{f}})|,$$

with $\varkappa$ being a given positive constant, $\lambda(\cdot)$ representing the eigenvalues of the corresponding matrix, and $|\cdot|$ denoting the absolute value of the corresponding number. Now, by solving the Eq. (2.9) we can get the increment $\Delta\mathbf{q}^n$ and, hence, $\mathbf{q}^{n+1}$ at the $(n+1)$-th pseudo-time level.

We remark that $\widetilde{\mathbf{J}}_{\mathbf{e}}^{\pm}$ and $\widetilde{\mathbf{J}}_{\mathbf{f}}^{\pm}$ constructed in such a way can guarantee that the eigenvalues of matrices with "+" are nonnegative and those with "−" are nonpositive. In actual applications, we may take $\varkappa = 1.0$ for the third-order upwind compact difference scheme and $\varkappa \geq 1.3$ for the fifth-order upwind compact difference scheme, so that $\widetilde{\mathbf{J}}_{\mathbf{e}}^{\pm}$ and $\widetilde{\mathbf{J}}_{\mathbf{f}}^{\pm}$ are diagonally dominant and their eigenvalues possess the required sign patterns; see [31].

2.3 The algebraic system of linear equations

We now rewrite the discretized system of linear equations (2.9) into an explicit matrix-vector form. To this end, we suppose that the computational grid has $(n_x+2)\times(n_y+2)$ grid points, and Δx and Δy are, respectively, the stepsizes in the x- and the y-directions. For simplicity, without loss of generality we impose the Dirichlet-type boundary condition on the definition domain, which leads to the determinant conditions

$$\Delta\mathbf{q}_{i,j}^{n}=0, \qquad \text{for} \quad i=0, \quad i=n_x+1, \quad j=0 \quad \text{and} \quad j=n_y+1,$$

for the system of linear equations (2.9). Thereby, at the grid point (i,j), after dividing the temporal stepsize $\Delta\tau_{i,j}$ through both sides, we can reformulate the (i,j)-th sub-system of linear equations of the discretized linear system (2.9) as

$$\begin{aligned}&\Bigg[\Delta\mathbf{q}_{i,j}+\Delta\tau_{i,j}\Bigg(\frac{\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{+}\big)_{i,j}\Delta\mathbf{q}_{i,j}-\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{+}\big)_{i-1,j}\Delta\mathbf{q}_{i-1,j}}{\Delta x}\\&+\frac{\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{-}\big)_{i+1,j}\Delta\mathbf{q}_{i+1,j}-\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{-}\big)_{i,j}\Delta\mathbf{q}_{i,j}}{\Delta x}-\mathbf{I}_m\frac{\Delta\mathbf{q}_{i+1,j}-2\Delta\mathbf{q}_{i,j}+\Delta\mathbf{q}_{i-1,j}}{\operatorname{Re}\Delta x^2}\Bigg)\\&+\Delta\tau_{i,j}\Bigg(\frac{\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{+}\big)_{i,j}\Delta\mathbf{q}_{i,j}-\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{+}\big)_{i,j-1}\Delta\mathbf{q}_{i,j-1}}{\Delta y}+\frac{\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{-}\big)_{i,j+1}\Delta\mathbf{q}_{i,j+1}-\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{-}\big)_{i,j}\Delta\mathbf{q}_{i,j}}{\Delta y}\\&-\mathbf{I}_m\frac{\Delta\mathbf{q}_{i,j+1}-2\Delta\mathbf{q}_{i,j}+\Delta\mathbf{q}_{i,j-1}}{\operatorname{Re}\Delta y^2}\Bigg)\Bigg]^{n}=\widetilde{\mathbf{g}}_{i,j}^{n},\end{aligned}\tag{2.10}$$

where $\Delta\tau_{i,j}$ stands for the pseudo-time stepsize at the grid point (i,j). By introducing the constants

$$\theta_{i,j}^{x}=\frac{\Delta\tau_{i,j}}{2\Delta x}, \quad \theta_{i,j}^{y}=\frac{\Delta\tau_{i,j}}{2\Delta y}, \quad \mu_{i,j}^{x}=\frac{\Delta\tau_{i,j}}{\operatorname{Re}\Delta x^2} \quad \text{and} \quad \mu_{i,j}^{y}=\frac{\Delta\tau_{i,j}}{\operatorname{Re}\Delta y^2},$$

and the matrices

$$\mathbf{C}_{i+1,j}=-\mu_{i,j}^{x}\mathbf{I}_m+2\theta_{i,j}^{x}\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{-}\big)_{i+1,j}, \quad \mathbf{H}_{i,j+1}=-\mu_{i,j}^{y}\mathbf{I}_m+2\theta_{i,j}^{y}\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{-}\big)_{i,j+1},$$

$$\mathbf{E}_{i,j}^{x}=2\mu_{i,j}^{x}\mathbf{I}_m+2\theta_{i,j}^{x}\Big(\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{+}\big)_{i,j}-\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{-}\big)_{i,j}\Big), \mathbf{E}_{i,j}^{y}=2\mu_{i,j}^{y}\mathbf{I}_m+2\theta_{i,j}^{y}\Big(\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{+}\big)_{i,j}-\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{-}\big)_{i,j}\Big),$$

$$\mathbf{F}_{i-1,j}=-\mu_{i,j}^{x}\mathbf{I}_m-2\theta_{i,j}^{x}\big(\widetilde{\mathbf{J}}_{\mathbf{e}}^{+}\big)_{i-1,j}, \quad \mathbf{G}_{i,j-1}=-\mu_{i,j}^{y}\mathbf{I}_m-2\theta_{i,j}^{y}\big(\widetilde{\mathbf{J}}_{\mathbf{f}}^{+}\big)_{i,j-1}$$

as well as

$$\mathbf{E}_{i,j} = \mathbf{I} + \mathbf{E}^{x}_{i,j} + \mathbf{E}^{y}_{i,j},$$

we can simply express the sub-system of linear equations (2.10) as

$$\begin{aligned}\big[\mathbf{F}_{i-1,j}\Delta\mathbf{q}_{i-1,j} + \mathbf{E}_{i,j}\Delta\mathbf{q}_{i,j} + \mathbf{C}_{i+1,j}\Delta\mathbf{q}_{i+1,j}\\ +\mathbf{G}_{i,j-1}\Delta\mathbf{q}_{i,j-1} + \mathbf{H}_{i,j+1}\Delta\mathbf{q}_{i,j+1}\big]^{n} = \widetilde{\mathbf{g}}^{n}_{i,j}.\end{aligned}$$

Therefore, the discretized linear system (2.9) can be equivalently rewritten into the matrix-vector form

$$\mathbf{A}^{n}\,\Delta\mathbf{q}^{n} = \widetilde{\mathbf{g}}^{n}, \tag{2.11}$$

where the coefficient matrix $\mathbf{A}^{n} \in \mathbb{R}^{\ell\times\ell}$, with $\ell = 3\times n_x\times n_y$, is a block-tridiagonal matrix given by

$$\mathbf{A}^{n} = \begin{pmatrix} \mathbf{D}^{n}_{1} & \mathbf{U}^{n}_{2} & 0 & \cdots & 0 \\ \mathbf{L}^{n}_{1} & \mathbf{D}^{n}_{2} & \mathbf{U}^{n}_{3} & \cdots & \vdots \\ 0 & \mathbf{L}^{n}_{2} & \mathbf{D}^{n}_{3} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{U}^{n}_{n_y} \\ 0 & \cdots & \cdots & \mathbf{L}^{n}_{n_y-1} & \mathbf{D}^{n}_{n_y} \end{pmatrix} \equiv \text{Tridiag}\left(\mathbf{L}^{n}_{i-1}, \mathbf{D}^{n}_{i}, \mathbf{U}^{n}_{i+1}\right), \tag{2.12}$$

the unknown vector $\Delta\mathbf{q}^{n} \in \mathbb{R}^{\ell}$ is given by

$$\Delta\mathbf{q}^{n} = \left(\Delta\mathbf{q}^{n}_{1,1}, \ldots, \Delta\mathbf{q}^{n}_{n_x,1}, \Delta\mathbf{q}^{n}_{1,2}, \ldots, \Delta\mathbf{q}^{n}_{n_x,2}, \ldots, \Delta\mathbf{q}^{n}_{1,n_y}, \ldots, \Delta\mathbf{q}^{n}_{n_x,n_y}\right)^{T},$$

and the right-hand side vector $\widetilde{\mathbf{g}}^{n} \in \mathbb{R}^{\ell}$ is given by

$$\widetilde{\mathbf{g}}^{n} = \left(\widetilde{\mathbf{g}}^{n}_{1,1}, \ldots, \widetilde{\mathbf{g}}^{n}_{n_x,1}, \widetilde{\mathbf{g}}^{n}_{1,2}, \ldots, \widetilde{\mathbf{g}}^{n}_{n_x,2}, \ldots, \widetilde{\mathbf{g}}^{n}_{1,n_y}, \ldots, \widetilde{\mathbf{g}}^{n}_{n_x,n_y}\right)^{T},$$

with $(\cdot)^{T}$ representing the transpose of the corresponding vector, $\mathbf{L}^{n}_{j}$ $(j = 1, 2, \ldots, n_y - 1)$ and $\mathbf{U}^{n}_{j}$ $(j = 2, 3, \ldots, n_y)$ being block-diagonal matrices defined as

$$\mathbf{L}^{n}_{j} = \begin{pmatrix} \mathbf{G}^{n}_{1,j} & 0 & \cdots & 0 \\ 0 & \mathbf{G}^{n}_{2,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{G}^{n}_{n_x,j} \end{pmatrix} \equiv \text{Diag}\left(\mathbf{G}^{n}_{1,j}, \mathbf{G}^{n}_{2,j}, \cdots, \mathbf{G}^{n}_{n_x,j}\right) \tag{2.13}$$

and

$$\mathbf{U}^{n}_{j} = \begin{pmatrix} \mathbf{H}^{n}_{1,j} & 0 & \cdots & 0 \\ 0 & \mathbf{H}^{n}_{2,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{H}^{n}_{n_x,j} \end{pmatrix} \equiv \text{Diag}\left(\mathbf{H}^{n}_{1,j}, \mathbf{H}^{n}_{2,j}, \cdots, \mathbf{H}^{n}_{n_x,j}\right), \tag{2.14}$$

respectively, and $\mathbf{D}^n_j$, $j = 1, 2, \ldots, n_y$, being block-tridiagonal matrices defined as

$$\mathbf{D}^n_j = \begin{pmatrix} \mathbf{E}^n_{1,j} & \mathbf{C}^n_{2,j} & \cdots & 0 \\ \mathbf{F}^n_{1,j} & \mathbf{E}^n_{2,j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}^n_{n_x,j} \\ 0 & \cdots & \mathbf{F}^n_{n_x-1,j} & \mathbf{E}^n_{n_x,j} \end{pmatrix} \equiv \text{Tridiag}\left(\mathbf{F}^n_{i-1,j}, \mathbf{E}^n_{i,j}, \mathbf{C}^n_{i+1,j}\right). \quad (2.15)$$

Moreover, $\mathbf{E}^x_{i,j}$, $\mathbf{E}^y_{i,j}$, $\mathbf{C}_{i+1,j}$, $\mathbf{F}_{i-1,j}$, $\mathbf{G}_{i,j-1}$ and $\mathbf{H}_{i,j+1}$ are 3-by-3 matrices of the following forms:

$$\begin{aligned}
\mathbf{E}^x_{i,j} &= \begin{pmatrix} 2\theta^x_{i,j}\varrho((\mathbf{J_e})_{i,j}) & 0 & 0 \\ 0 & 2\theta^x_{i,j}\varrho((\mathbf{J_e})_{i,j}) + 2\mu^x_{i,j} & 0 \\ 0 & 0 & 2\theta^x_{i,j}\varrho((\mathbf{J_e})_{i,j}) + 2\mu^x_{i,j} \end{pmatrix}, \\
\mathbf{E}^y_{i,j} &= \begin{pmatrix} 2\theta^y_{i,j}\varrho((\mathbf{J_f})_{i,j}) & 0 & 0 \\ 0 & 2\theta^y_{i,j}\varrho((\mathbf{J_f})_{i,j}) + 2\mu^y_{i,j} & 0 \\ 0 & 0 & 2\theta^y_{i,j}\varrho((\mathbf{J_f})_{i,j}) + 2\mu^y_{i,j} \end{pmatrix}, \\
\mathbf{C}_{i+1,j} &= \begin{pmatrix} -\theta^x_{i,j}\varrho((\mathbf{J_e})_{i+1,j}) & \theta^x_{i,j}\beta & 0 \\ \theta^x_{i,j} & \theta^x_{i,j}(2u_{i+1,j} - \varrho((\mathbf{J_e})_{i+1,j})) - \mu^x_{i,j} & 0 \\ 0 & \theta^x_{i,j}v_{i+1,j} & \theta^x_{i,j}(u_{i+1,j} - \varrho((\mathbf{J_e})_{i+1,j})) - \mu^x_{i,j} \end{pmatrix}, \\
\mathbf{F}_{i-1,j} &= -\begin{pmatrix} \theta^x_{i,j}\varrho((\mathbf{J_e})_{i-1,j}) & \theta^x_{i,j}\beta & 0 \\ \theta^x_{i,j} & \theta^x_{i,j}(2u_{i-1,j} + \varrho((\mathbf{J_e})_{i-1,j})) + \mu^x_{i,j} & 0 \\ 0 & \theta^x_{i,j}v_{i-1,j} & \theta^x_{i,j}(u_{i-1,j} + \varrho((\mathbf{J_e})_{i-1,j})) + \mu^x_{i,j} \end{pmatrix}, \\
\mathbf{G}_{i,j-1} &= -\begin{pmatrix} \theta^y_{i,j}\varrho((\mathbf{J_f})_{i,j-1}) & 0 & \theta^y_{i,j}\beta \\ 0 & \theta^y_{i,j}(v_{i,j-1} + \varrho((\mathbf{J_f})_{i,j-1})) + \mu^y_{i,j} & \theta^y_{i,j}u_{i,j-1} \\ \theta^y_{i,j} & 0 & \theta^y_{i,j}(2v_{i,j-1} + \varrho((\mathbf{J_f})_{i,j-1})) + \mu^y_{i,j} \end{pmatrix}
\end{aligned}$$

and

$$\mathbf{H}_{i,j+1} = \begin{pmatrix} -\theta^y_{i,j}\varrho((\mathbf{J_f})_{i,j+1}) & 0 & \theta^y_{i,j}\beta \\ 0 & \theta^y_{i,j}(v_{i,j+1} - \varrho((\mathbf{J_f})_{i,j+1})) - \mu^y_{i,j} & \theta^y_{i,j}u_{i,j+1} \\ \theta^y_{i,j} & 0 & \theta^y_{i,j}(2v_{i,j+1} - \varrho((\mathbf{J_f})_{i,j+1})) - \mu^y_{i,j} \end{pmatrix},$$

where $u_{i,j}$ and $v_{i,j}$ are the values of the functions u and v at the (i, j)-th grid point. Here for notational convenience we have omitted the superscript n used to label the pseudo-time level. Obviously, $\mathbf{E}^x_{i,j}$ and $\mathbf{E}^y_{i,j}$ are diagonal matrices, $\mathbf{C}_{i+1,j}$ and $\mathbf{F}_{i-1,j}$ are tridiagonal matrices, and $\mathbf{G}_{i,j-1}$ and $\mathbf{H}_{i,j+1}$ are dense matrices.

We remark that when $n_x = n_y = 3$, for example, the matrix $\mathbf{A}^n$ possesses the sparse pattern

$$\left(\begin{array}{ccc|ccc|ccc}
\textcircled{d} & \textcircled{f} & & \textcircled{f} & & & & & \\
\textcircled{f} & \textcircled{d} & \textcircled{f} & & \textcircled{f} & & & & \\
& \textcircled{f} & \textcircled{d} & & & \textcircled{f} & & & \\
\hline
\textcircled{f} & & & \textcircled{d} & \textcircled{f} & & \textcircled{f} & & \\
& \textcircled{f} & & \textcircled{f} & \textcircled{d} & \textcircled{f} & & \textcircled{f} & \\
& & \textcircled{f} & & \textcircled{f} & \textcircled{d} & & & \textcircled{f} \\
\hline
& & & \textcircled{f} & & & \textcircled{d} & \textcircled{f} & \\
& & & & \textcircled{f} & & \textcircled{f} & \textcircled{d} & \textcircled{f} \\
& & & & & \textcircled{f} & & \textcircled{f} & \textcircled{d}
\end{array}\right),$$

where

$$ⓓ = \begin{pmatrix} \star & 0 & 0 \\ 0 & \star & 0 \\ 0 & 0 & \star \end{pmatrix} \quad \text{and} \quad ⓕ = \begin{pmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{pmatrix}$$

are 3-by-3 diagonal and dense matrices, respectively.

2.4 Diagonal dominance property

We are going to explore diagonal dominance for the matrix $\mathbf{A}^n \in \mathbb{R}^{\ell\times\ell}$. Note that the i-th block column in the j-th block column of the matrix $\mathbf{A}^n$ consists of the matrix blocks $\mathbf{C}^n_{i,j}$, $\mathbf{E}^n_{i,j}$, $\mathbf{F}^n_{i,j}$, $\mathbf{G}^n_{i,j}$ and $\mathbf{H}^n_{i,j}$, with $\mathbf{E}^n_{i,j} = \mathbf{I} + [\mathbf{E}^x_{i,j}]^n + [\mathbf{E}^y_{i,j}]^n$ being the diagonal block. By further expanding all elements of these matrix blocks to the first-order accuracy at the spatial grid point (i, j), and then omitting the subscripts i, j and the superscript n used to label the spatial and the pseudo-time levels, respectively, we know that the matrix $\mathbf{A} \in \mathbb{R}^{\ell\times\ell}$ is strictly diagonally dominant by columns provided the following inequalities hold true:

(a) $\theta^x + \theta^y < \frac{1}{2}$;
(b) $\theta^x(\beta + |v|) < \frac{1}{2}$; and
(c) $\theta^y(\beta + |u|) < \frac{1}{2}$.

Here we have applied the facts that

$$\theta^x(2u - \varrho(\mathbf{J_e})) - \mu^x < 0, \qquad \theta^x(u - \varrho(\mathbf{J_e})) - \mu^x < 0$$

and

$$\theta^y(2v - \varrho(\mathbf{J_f})) - \mu^y < 0, \qquad \theta^y(v - \varrho(\mathbf{J_f})) - \mu^y < 0.$$

The above inequalities (a)–(c) straightforwardly lead to a sufficient condition for guaranteeing the strictly diagonal dominance of the matrix $\mathbf{A} \in \mathbb{R}^{\ell\times\ell}$.

Theorem 2.1 *The matrix $\mathbf{A}^n \in \mathbb{R}^{\ell\times\ell}$ defined by Eqs.* (2.12)–(2.15) *is strictly diagonally dominant, if Δx and Δy are reasonably small and $\Delta\tau$ satisfies*

$$\Delta\tau < \min\left\{\frac{\Delta x\Delta y}{\Delta x + \Delta y}, \frac{\Delta x}{\beta + \max\limits_{\substack{1\le i\le n_x\\ 1\le j\le n_y}} |v^n_{i,j}|}, \frac{\Delta y}{\beta + \max\limits_{\substack{1\le i\le n_x\\ 1\le j\le n_y}} |u^n_{i,j}|}\right\}. \quad (2.16)$$

When the spatial and the temporal stepsizes satisfy the condition (2.16), we know from Theorem 2.1 that the matrix $\mathbf{A}^n$ is nonsingular, so that the discretized linear system (2.11) has a unique solution. Moreover, $\mathbf{A}^n$ is positive real. That the matrix $\mathbf{A}^n$ is diagonally dominant guarantees existence of its LU factorization and convergence of its Gauss-Seidel splitting and GMRES iteration; see [11, 17, 21, 22].

3 Construction of preconditioning matrices

We can obtain an approximate matrix, say, $\mathbf{P}^n$, to the matrix $\mathbf{A}^n \in \mathbb{R}^{\ell\times\ell}$ defined in (2.12) by selectively dropping its off-diagonal elements. More precisely, for the 3-by-3 matrices $\mathbf{C}_{i+1,j}$, $\mathbf{F}_{i-1,j}$, $\mathbf{G}_{i,j-1}$ and $\mathbf{H}_{i,j+1}$, we only keep the diagonal elements, but drop all of the off-diagonal elements, obtaining their approximated matrices denoted as $\widetilde{\mathbf{C}}_{i+1,j}$, $\widetilde{\mathbf{F}}_{i-1,j}$, $\widetilde{\mathbf{G}}_{i,j-1}$ and $\widetilde{\mathbf{H}}_{i,j+1}$, respectively. Then we define matrices $\widetilde{\mathbf{L}}_j^n$ $(j = 1, 2, \ldots, n_y - 1)$, $\widetilde{\mathbf{U}}_j^n$ $(j = 2, 3, \ldots, n_y)$ and $\widetilde{\mathbf{D}}_j^n$ $(j = 1, 2, \ldots, n_y)$ as

$$\left\{\begin{aligned} \widetilde{\mathbf{L}}_j^n &= \mathrm{Diag}\left(\widetilde{\mathbf{G}}_{1,j}^n, \widetilde{\mathbf{G}}_{2,j}^n, \cdots, \widetilde{\mathbf{G}}_{n_x,j}^n\right), \\ \widetilde{\mathbf{U}}_j^n &= \mathrm{Diag}\left(\widetilde{\mathbf{H}}_{1,j}^n, \widetilde{\mathbf{H}}_{2,j}^n, \cdots, \widetilde{\mathbf{H}}_{n_x,j}^n\right), \\ \widetilde{\mathbf{D}}_j^n &= \mathrm{Tridiag}\left(\widetilde{\mathbf{F}}_{i-1,j}^n, \mathbf{E}_{i,j}^n, \widetilde{\mathbf{C}}_{i+1,j}^n\right). \end{aligned}\right. \tag{3.1}$$

Finally, the matrix $\mathbf{P}^n \in \mathbb{R}^{\ell\times\ell}$ is given by

$$\mathbf{P}^n = \mathrm{Tridiag}\left(\widetilde{\mathbf{L}}_{i-1}^n, \widetilde{\mathbf{D}}_i^n, \widetilde{\mathbf{U}}_{i+1}^n\right). \tag{3.2}$$

Note that the matrix $\mathbf{P}^n$ is a block-tridiagonal matrix. Its diagonal blocks $\widetilde{\mathbf{D}}_j^n$ $(j = 1, 2, \ldots, n_y)$ are block-tridiagonal matrices, with the block elements being 3-by-3 diagonal matrices. And its off-diagonal blocks $\widetilde{\mathbf{L}}_j^n$ $(j = 1, 2, \ldots, n_y - 1)$ and $\widetilde{\mathbf{U}}_j^n$ $(j = 2, 3, \ldots, n_y)$ are block-diagonal matrices, with the block elements being 3-by-3 diagonal matrices, too.

We remark that when $n_x = n_y = 3$, for example, the matrix $\mathbf{P}^n$ possesses the sparse pattern

$$\left(\begin{array}{ccc|ccc|ccc} \textcircled{d} & \textcircled{d} & & \textcircled{d} & & & & & \\ \textcircled{d} & \textcircled{d} & \textcircled{d} & & \textcircled{d} & & & & \\ & \textcircled{d} & \textcircled{d} & & & \textcircled{d} & & & \\ \hline \textcircled{d} & & & \textcircled{d} & \textcircled{d} & & \textcircled{d} & & \\ & \textcircled{d} & & \textcircled{d} & \textcircled{d} & \textcircled{d} & & \textcircled{d} & \\ & & \textcircled{d} & & \textcircled{d} & \textcircled{d} & & & \textcircled{d} \\ \hline & & & \textcircled{d} & & & \textcircled{d} & \textcircled{d} & \\ & & & & \textcircled{d} & & \textcircled{d} & \textcircled{d} & \textcircled{d} \\ & & & & & \textcircled{d} & & \textcircled{d} & \textcircled{d} \end{array}\right),$$

where

$$\textcircled{d} = \begin{pmatrix} \star & 0 & 0 \\ 0 & \star & 0 \\ 0 & 0 & \star \end{pmatrix}$$

is a 3-by-3 diagonal matrix.

In general, we can demonstrate the following property for the matrix $\mathbf{P}^n$ by straightforward computations.

Theorem 3.1 *Let the spatial stepsizes Δx and Δy be reasonably small. Then*

(a) *the matrix $\mathbf{P}^n = (\mathbf{P}^n_{i,j}) \in \mathbb{R}^{\ell\times\ell}$, defined by (3.2) and (3.1), is strictly diagonally dominant by columns, and satisfies*

$$|\mathbf{P}^n_{j,j}| \geq 1 + \sum_{1\leq i\leq \ell, i\neq j} |\mathbf{P}^n_{i,j}|, \qquad j = 1, 2, \ldots, \ell;$$

(b) *the matrix $\mathbf{R}^n = (\mathbf{R}^n_{i,j}) \in \mathbb{R}^{\ell\times\ell}$, defined by $\mathbf{R}^n = \mathbf{P}^n - \mathbf{A}^n$, satisfies*

$$\sum_{1\leq i\leq \ell} |\mathbf{R}^n_{i,j}| \leq \max\left\{\frac{\Delta\tau}{\Delta x} + \frac{\Delta\tau}{\Delta y}, \quad \frac{\Delta\tau}{\Delta x}\left(\beta + \max_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |v^n_{i,j}|\right), \right.$$
$$\left. \frac{\Delta\tau}{\Delta y}\left(\beta + \max_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |u^n_{i,j}|\right)\right\}, \quad j = 1, 2, \ldots, \ell.$$

From Theorem 3.1 we know that the approximate matrix $\mathbf{P}^n$ is nonsingular and positive real. Moreover, when the velocity components u and v are uniformly bounded, $\|\mathbf{P}^n - \mathbf{A}^n\|_1$ is essentially bounded by a quantity of the order $\frac{\Delta\tau}{\Delta x} + \frac{\Delta\tau}{\Delta y}$. In addition, Theorem 3.1 immediately results in a bound for the eigenvalues of the matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ described in the following theorem.

Theorem 3.2 *Let the spatial stepsizes Δx and Δy be reasonably small. Then all eigenvalues of the matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ are located within the complex disk centered at $(1, 0)$ with radius r^n, where*

$$r^n = \max\left\{\frac{\Delta\tau}{\Delta x} + \frac{\Delta\tau}{\Delta y}, \quad \frac{\Delta\tau}{\Delta x}\left(\beta + \max_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |v^n_{i,j}|\right), \frac{\Delta\tau}{\Delta y}\left(\beta + \max_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |u^n_{i,j}|\right)\right\}.$$

As a result, if the spatial stepsizes Δx and Δy are reasonably small and the temporal stepsize $\Delta\tau$ satisfies

$$\Delta\tau < \min\left\{\frac{\Delta x\Delta y}{\Delta x + \Delta y}, \quad \frac{\Delta x}{\beta + \max\limits_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |v^n_{i,j}|}, \quad \frac{\Delta y}{\beta + \max\limits_{\substack{1\leq i\leq n_x\\ 1\leq j\leq n_y}} |u^n_{i,j}|}\right\},$$

then $r^n < 1$ and all eigenvalues of the matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ have positive real parts.

Proof Let λ^n be an eigenvalue of the matrix $(\mathbf{P}^n)^{-1}\mathbf{R}^n$. Then $\mu^n = 1 - \lambda^n$ is an eigenvalue of the matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$. In accordance with [13] and Theorem 3.1 we know that

$$|\lambda^n| \leq \max_{1\leq j\leq \ell} \frac{\sum\limits_{1\leq i\leq \ell} \left|\mathbf{R}^n_{i,j}\right|}{\left|\mathbf{P}^n_{j,j}\right| - \sum\limits_{1\leq i\leq \ell, i\neq j} \left|\mathbf{P}^n_{i,j}\right|} \leq \max_{1\leq j\leq \ell} \sum_{1\leq i\leq \ell} \left|\mathbf{R}^n_{i,j}\right| \leq r^n.$$

Hence, from [4] we see that all eigenvalues of the matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ are located within the complex disk centered at $(1, 0)$ with radius r^n. □

From [2, 11, 22, 29] we know that the convergence rate of the GMRES method is essentially determined by the condition number of the eigenvector matrix and the radius of the eigenvalue disk of the coefficient matrix of the referred linear system. The smaller of both quantities are, the faster the convergence rate of the GMRES will be. Therefore, it follows from Theorem 3.2 that the preconditioned GMRES method with the preconditioning matrix $\mathbf{P}^n$ is convergent to the exact solution of the discretized linear system (2.11) with a rate being at least r^n, provided the eigenvector matrix of $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ is not very ill-conditioned; see [2, 22].

By employing ILU factorization and, alternatively, UGS splitting, to the sparse and structured matrix $\mathbf{P}^n$, we can obtain the ILU and the UGS preconditioning matrices, say, $\mathbf{P}^n_{\mathrm{ILU}}$ and $\mathbf{P}^n_{\mathrm{UGS}}$, to the coefficient matrix $\mathbf{A}^n$ of the discretized linear system (2.11). In the ILU factorization process, we should adopt suitable dropping rules and certain sparsity patterns in the incomplete lower- and upper-triangular factors. And in the UGS splitting, we can define $\mathbf{P}^n_{\mathrm{UGS}}$ as

$$\mathbf{P}^n_{\mathrm{UGS}} = (\widetilde{\mathbf{D}} - \widetilde{\mathbf{L}})\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{D}} - \widetilde{\mathbf{U}}),$$

where $\widetilde{\mathbf{D}}$, $-\widetilde{\mathbf{L}}$ and $-\widetilde{\mathbf{U}}$ being the diagonal, the strictly lower-triangular and the strictly upper-triangular matrices of the matrix $\mathbf{P}^n$, respectively.

4 Numerical results

As in [30, 31], we use the *plane Poiseuille flow*, the *plane Couette-Poiseuille flow*, and the *modified cavity flow* as examples to examine the feasibility and effectiveness of the approximated ILU (i.e., $\mathbf{P}^n_{\mathrm{ILU}}$) and the approximated UGS (i.e., $\mathbf{P}^n_{\mathrm{UGS}}$) preconditioning matrices for the matrix $\mathbf{A}^n$, when they are employed to precondition the GMRES iteration method for solving the system of linear equations (2.11).

In the computations, we take the threshold ILU of the approximate matrix $\mathbf{P}^n$ as the ILU preconditioning matrix to the coefficient matrix $\mathbf{A}^n$ of the discretized linear system (2.11). In addition, when solving the discretized linear system (2.11) on certain pseudo-time level by the GMRES or the preconditioned GMRES methods, we set the initial vector to be 0 and adopt the stopping criterion that demands the ratio of the current and the initial residuals in the Euclidean norm be less than 10^{-6}.

4.1 Description of test problems

The plane Poiseuille flow and the plane Couette-Poiseuille flow are governed by the two-dimensional steady-state incompressible Navier-Stokes equations (2.1) imposed the constraint $u = u(y)$, satisfying

$$\Pi + \frac{d^2u}{dy^2} = 0, \qquad 0 \le x, y \le 1,$$

on the velocity component u, where $\Pi = -\mathrm{Re}\frac{\partial p}{\partial x}$ is a dimensionless constant pressure gradient. The boundary conditions for the velocity components u and v, and for the pressure p are

$$\begin{array}{llllll}
\frac{\partial p}{\partial y}(x,1) &= 0, & u(x,1) &= u_{\text{const}}, & v(x,1) &= 0,\\
p(0,y) &= p_{\text{inlet}}, & \frac{\partial u}{\partial x}(0,y) &= 0, & v(0,y) &= 0,\\
p(1,y) &= 0, & \frac{\partial u}{\partial x}(1,y) &= 0, & \frac{\partial v}{\partial x}(1,y) &= 0,\\
\frac{\partial p}{\partial y}(x,0) &= 0, & u(x,0) &= 0, & v(x,0) &= 0.
\end{array}$$

In particular, when $u_{\text{const}} = 0$ and $\Pi \neq 0$, we obtain the plane Poiseuille flow, which has the exact solution

$$u(y) = \frac{\Pi}{2}(y - y^2);$$

and when $u_{\text{const}} = 1$ and $\Pi \neq 0$, we obtain the plane Couette-Poiseuille flow, which has the exact solution

$$u(y) = \frac{\Pi}{2}(y - y^2) + y.$$

The modified cavity flow defined on the domain $0 \le x, y \le 1$ is governed by the two-dimensional steady-state incompressible Navier-Stokes equations (2.1). The boundary conditions for the velocity components u and v, and for the pressure p are of the Dirichlet type, i.e., the values of these three components are equal to 0 everywhere on the boundary of the unit square, except for

$$u(x,1) = 16(x^4 - 2x^3 + x^2), \qquad p(1,y) = \frac{6.4}{\mathrm{Re}}y,$$

and

$$\begin{aligned} p(x,1) = {} & \frac{8}{\mathrm{Re}}[24F(x) + 2f'(x)g''(1) + f'''(x)g(1)] \\ & -64[F_2(x)G(1) - g(1)g''(1)F_1(x)], \end{aligned}$$

where

$$\begin{array}{llllll}
f(x) &= x^4 - 2x^3 + x^2, & F(x) &= \int f(x)dx, & F_1(x) &= f(x)f''(x) - [f'(x)]^2,\\
F_2(x) &= \int f(x)f'(x)dx, & g(y) &= y^4 - y^2, & G(y) &= g(y)g'''(y) - g'(y)g''(y).
\end{array}$$

The exact solution of this flow is given by

$$u(x,y) = 8f(x)g'(y), \qquad v(x,y) = -8f'(x)g(y),$$

and

$$p(x, y) = \frac{8}{\mathrm{Re}}[F(x)g'''(y) + f'(x)g'(y)] + 64F_2(x)(g(y)g''(y) - [g'(y)]^2).$$

In our implementations, in the plane Poiseuille and the plane Couette-Poiseuille flows we set the constants p_{inlet} and Π to be both equal to 10, the Reynolds number Re to be equal to 1, and the artificial compressibility factor β to be 100; while in the modified cavity flow, we take the Reynolds number Re and the artificial compressibility factor β to be both equal to 100. Moreover, we adopt an equidistant spatial grid with the stepsize

$$h := \Delta x = \Delta y \equiv \frac{1}{N+1}$$

and a variable local temporal grid with the stepsize

$$\Delta\tau_{i,j} = \frac{ch}{|u_{i,j}| + |v_{i,j}| + \sqrt{u_{i,j}^2 + \beta} + \sqrt{v_{i,j}^2 + \beta}}, \quad i, j = 1, 2, \ldots, N,$$

where c is the Courant number, which is equal to 8.0 for the plane Poiseuille and the plane Couette-Poiseuille flows, and is equal to 30.0 for the modified cavity flow. Hence, on each pseudo-time level we obtain a system of linear equations of the form (2.11), with $\ell = 3 \times N \times N$.

4.2 Numerical property of preconditioning matrices

We show numerical advantages of the preconditioned GMRES over the AF-ADI in terms of absolute error, and also show computational effectiveness of the preconditioned GMRES methods over the GMRES method in terms of iteration count and CPU time.

We depict in Figs. 1 and 2 the curves of the absolute errors versus the number of pseudo-time levels and in Figs. 3 and 4 the curves of the absolute errors versus the CPU times with respect to the velocity components u, v, and/or the pressure p, for the plane Poiseuille flow, the plane Couette-Poiseuille flow and the modified cavity flow, respectively, when $N = 19$. Because the velocity component $v = 0$ and the pressure p is a known constant for the plane Poiseuille and the plane Couette-Poiseuille flows, we only depict the absolute error curves with respect to the velocity component u in Fig. 1 for these two flows.

From these figures, we see that the GMRES method converges faster than the AF-ADI method in both iteration step and CPU time, which indicates that the overall numerical process, including the discretization, the linearization and the GMRES solve, is accurate, effective and robust for solving the two-dimensional steady-state incompressible Navier-Stokes equations (2.1).

In Figs. 5 and 6 we depict the spatial distributions of the absolute errors for the velocity components u and v, and/or the pressure p with respect to the plane Poiseuille flow, the plane Couette-Poiseuille flow and the modified cavity flow, respectively, when $N = 19$. Here the approximated solutions are computed by employing the $\mathbf{P}_{\mathrm{ILU}}^{n}$-preconditioned GMRES method. From these figures we see that

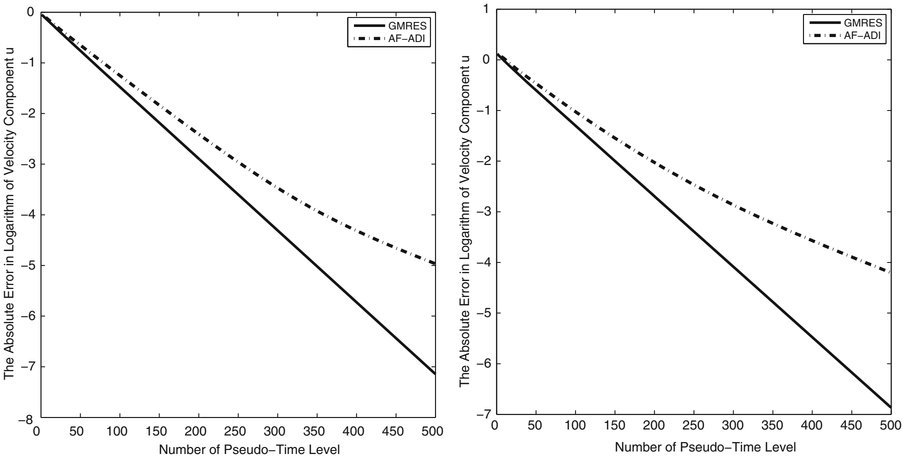

Fig. 1 The curves of the absolute errors in logarithm versus the number of pseudo-time levels with respect to the velocity component u when $N = 19$; the plane Poiseuille flow (*left*), and the plane Couette-Poiseuille flow (*right*)

all errors are of the order of 10^{-3}, which shows that the preconditioned GMRES method can be applied to produce an accurate solution of the two-dimensional steady-state incompressible Navier-Stokes equations (2.1).

In Tables 1, 2, 3, 4, 5 and 6 we list the number of iteration steps (denoted as **IT**) and the total CPU time in seconds (denoted as **CPU**) for the GMRES methods with or without a preconditioner for the plane Poiseuille flow, the plane Couette-Poiseuille flow and the modified cavity flow, at the pseudo-time levels $n = 0$ and 100, respectively. Besides, we also list the CPU time of the threshold ILU factorization (denoted as $\mathrm{CPU}_{\mathrm{ILU}}$) for the original matrix $\mathbf{A}^n$ or the approximate matrix $\mathbf{P}^n$. In these tables, the symbol **I** indicates that no preconditioner is used when solving the system of linear equations (2.11) with GMRES, while $\mathbf{A}^n_{\mathrm{ILU}}$ represents the ILU factorization preconditioner of the matrix $\mathbf{A}^n$, and $\mathbf{P}^n_{\mathrm{ILU}}$ and $\mathbf{P}^n_{\mathrm{UGS}}$ represent the ILU factorization

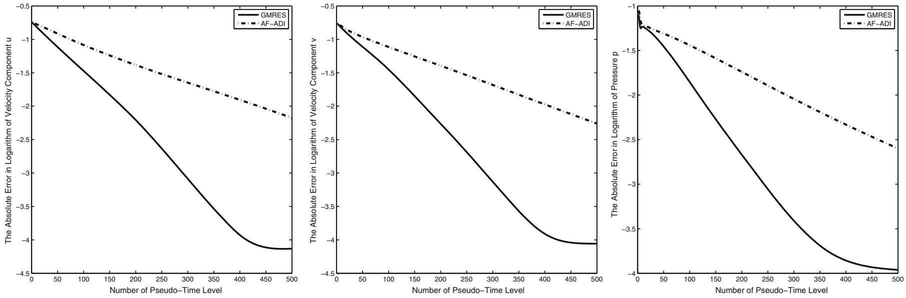

Fig. 2 The curves of the absolute errors in logarithm versus the number of pseudo-time levels for the modified cavity flow when $N = 19$; the velocity component u (*left*), the velocity component v (*middle*), and the pressure p (*right*)

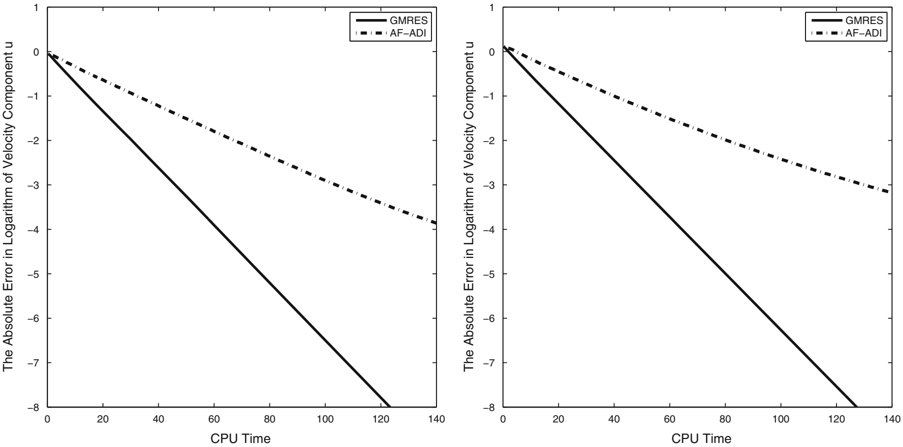

Fig. 3 The curves of the absolute errors in logarithm versus the CPU times with respect to the velocity component u when $N = 19$; the plane Poiseuille flow (*left*), and the plane Couette-Poiseuille flow (*right*)

and the UGS splitting preconditioners of the matrix $\mathbf{P}^n$, respectively. Here, we adopt 10^{-3} as the dropping tolerance in these ILU factorization processes.

For the plane Poiseuille flow, in Tables 1 and 2 we list the iteration steps and the CPU times of the vanilla and the preconditioned GMRES methods with respect to different spatial grids at the pseudo-time levels $n = 0$ and $n = 100$, respectively. From these tables we see that the iteration step and the CPU time of the vanilla GMRES method are very large and are monotonically increasing with respect to both N and n, but those of the preconditioned GMRES methods with both preconditioners $\mathbf{A}^n_{\mathrm{ILU}}$ and $\mathbf{P}^n$ are considerably small, especially when the spatial grid becomes fine. Moreover, when $n = 0$ or $n = 100$, with respect to N the number of iteration steps of the $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES method is slowly increasing, while that of the $\mathbf{P}^n$-preconditioned GMRES method is slowly decreasing, and it keeps almost a constant for each of these two preconditioned GMRES methods. To achieve the prescribed convergence criterion, the $\mathbf{P}^n$-preconditioned GMRES method requires the same or

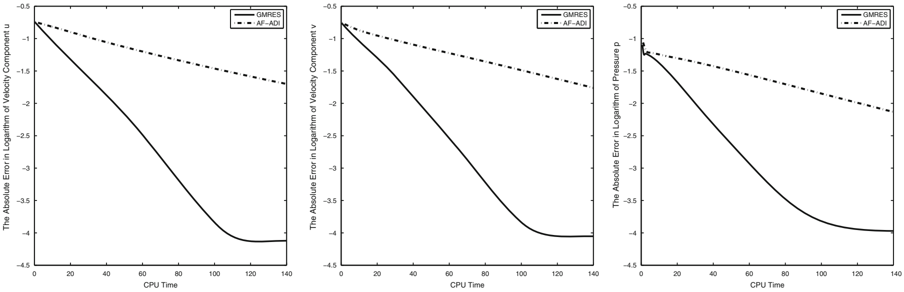

Fig. 4 The curves of the absolute errors in logarithm versus the CPU times for the modified cavity flow when $N = 19$; the velocity component u (*left*), the velocity component v (*middle*), and the pressure p (*right*)

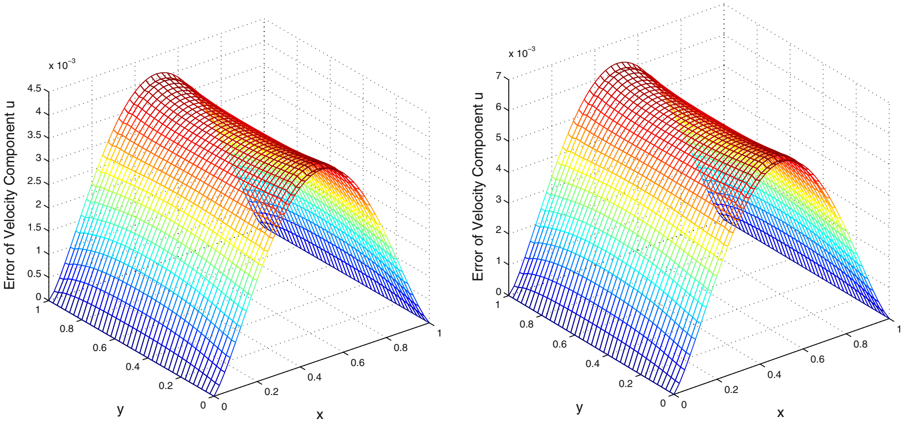

Fig. 5 The spatial distributions of the absolute errors for the velocity component u when $N = 19$ as computed by using $\mathbf{P}^n_{\mathrm{ILU}}$-preconditioned GMRES method; the plane Poiseuille flow (*left*), and the plane Couette-Poiseuille flow (*right*)

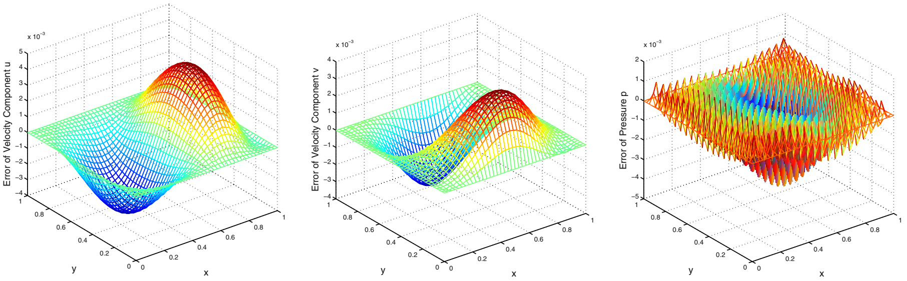

Fig. 6 The spatial distributions of the absolute errors for the modified cavity flow when $N = 19$ as computed by using $\mathbf{P}^n_{\mathrm{ILU}}$-preconditioned GMRES method; the velocity component u (*left*), the velocity component v (*middle*), and the pressure p (*right*)

Table 1 Numerical Results for the Plane Poiseuille Flow When $n = 0$

N		39	79	159	179	199
$\mathbf{I}$	IT	90	122	154	161	167
	CPU	0.44	3.31	34.17	47.29	61.64
$\mathbf{A}^n_{\mathrm{ILU}}$	IT	6	7	8	8	9
	CPU$_{\mathrm{ILU}}$	0.17	1.05	7.51	10.67	14.39
	CPU	0.20	1.14	7.96	11.32	15.21
$\mathbf{P}^n$	IT	12	10	9	8	8
	CPU	0.76	2.92	11.57	13.95	17.51
$\mathbf{P}^n_{\mathrm{ILU}}$	IT	12	11	11	11	12
	CPU$_{\mathrm{ILU}}$	0.03	0.22	1.57	2.20	3.00
	CPU	0.06	0.31	2.06	2.84	3.85
$\mathbf{P}^n_{\mathrm{UGS}}$	IT	33	43	53	54	56
	CPU	0.06	0.55	4.62	6.00	7.92

Table 2 Numerical Results for the Plane Poiseuille Flow When $n = 100$

N		39	79	159	179	199
$\mathbf{I}$	IT	98	129	165	172	188
	CPU	0.52	3.76	39.35	55.28	80.46
$\mathbf{A}^n_{\mathrm{ILU}}$	IT	6	7	9	9	10
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.16	1.12	7.64	10.60	14.40
	CPU	0.19	1.23	8.17	11.21	15.28
$\mathbf{P}^n$	IT	12	10	9	8	8
	CPU	0.70	2.94	11.82	14.12	17.68
$\mathbf{P}^n_{\mathrm{ILU}}$	IT	12	11	12	12	13
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.03	0.28	1.59	2.18	2.96
	CPU	0.08	0.35	2.17	2.85	3.89
$\mathbf{P}^n_{\mathrm{UGS}}$	IT	34	44	58	60	66
	CPU	0.08	0.55	5.31	7.21	10.54

even less number of iteration steps than the $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES method for almost all N except for the smallest two or three, but the former costs mildly more computing time than the latter, with the time difference becoming relatively small when N is growing. We note that in the total CPU times of the $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES method, the time of ILU factorization is strongly dominant over the time of iteration solve. Roughly speaking, when being used as preconditioners, the approximate matrix $\mathbf{P}^n$ outperforms the ILU factorization $\mathbf{A}^n_{\mathrm{ILU}}$ in terms of the iteration step of the GMRES method, but in terms of the CPU time the situation is just reversed.

Now we turn to analyze and compare the numerical results of the ILU factorization and the UGS splitting preconditioners resulted from the approximate matrix $\mathbf{P}^n$. From

Table 3 Numerical Results for the Plane Couette-Poiseuille Flow When $n = 0$

N		39	79	159	179	199
$\mathbf{I}$	IT	91	118	150	157	163
	CPU	0.67	4.13	34.91	48.62	63.86
$\mathbf{A}^n_{\mathrm{ILU}}$	IT	6	6	7	8	8
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.22	1.41	10.08	14.17	19.33
	CPU	0.27	1.63	11.19	15.72	21.30
$\mathbf{P}^n$	IT	12	10	8	8	8
	CPU	0.55	2.53	13.61	21.64	24.30
$\mathbf{P}^n_{\mathrm{ILU}}$	IT	12	10	11	11	11
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.06	0.34	2.55	3.59	4.92
	CPU	0.14	0.67	4.19	5.67	7.45
$\mathbf{P}^n_{\mathrm{UGS}}$	IT	35	44	54	56	59
	CPU	0.25	1.53	9.56	12.91	17.16

Table 4 Numerical Results for the Plane Couette-Poiseuille Flow When $n = 100$

N		39	79	159	179	199
$\mathbf{I}$	IT	95	128	163	170	178
	CPU	0.70	5.69	40.83	56.31	75.67
$\mathbf{A}^n_{\mathrm{ILU}}$	IT	6	7	8	9	9
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.27	1.42	10.09	14.23	19.33
	CPU	0.31	1.67	11.31	15.97	21.48
$\mathbf{P}^n$	IT	12	10	8	8	8
	CPU	0.55	2.56	13.64	21.78	24.48
$\mathbf{P}^n_{\mathrm{ILU}}$	IT	12	11	11	11	11
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.05	0.36	2.58	3.59	4.95
	CPU	0.14	0.73	4.19	5.61	7.55
$\mathbf{P}^n_{\mathrm{UGS}}$	IT	33	43	56	56	59
	CPU	0.23	1.47	10.13	12.83	17.13

Tables 1–2 we observe that the $\mathbf{P}^n_{\mathrm{ILU}}$-preconditioned GMRES method costs greatly smaller computing time than both $\mathbf{P}^n$- and $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES method, though the former still shows more but comparable iteration steps to the latter. The ILU factorization time for generating $\mathbf{P}^n_{\mathrm{ILU}}$ is much smaller than that for generating $\mathbf{A}^n_{\mathrm{ILU}}$, which again confirms that the simple approximation $\mathbf{P}^n$ is very effective for being used to further build up an economical and high-quality preconditioner for the original matrix $\mathbf{A}^n$ in actual computations. For all N and n, the iteration steps and CPU times of the $\mathbf{P}^n_{\mathrm{UGS}}$-preconditioned GMRES method are greatly more than those of the $\mathbf{P}^n_{\mathrm{ILU}}$-preconditioned GMRES method, but are much less than those of both $\mathbf{P}^n$- and $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES methods.

Table 5 Numerical Results for the Modified Cavity Flow When $n = 0$

N		39	79	159	179	199
$\mathbf{I}$	IT	153	193	231	242	242
	CPU	1.83	10.38	80.69	112.03	137.03
$\mathbf{A}^n_{\mathrm{ILU}}$	IT	10	15	21	23	25
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.38	2.90	23.67	33.28	46.31
	CPU	0.50	3.69	29.13	40.88	56.84
$\mathbf{P}^n$	IT	34	32	30	30	30
	CPU	1.59	8.00	48.44	66.38	86.14
$\mathbf{P}^n_{\mathrm{ILU}}$	IT	36	36	35	35	35
	$\mathrm{CPU}_{\mathrm{ILU}}$	0.05	0.36	2.59	3.64	5.02
	CPU	0.34	1.66	8.55	11.22	14.53
$\mathbf{P}^n_{\mathrm{UGS}}$	IT	66	83	96	98	98
	CPU	0.66	3.50	22.31	29.44	36.58

Table 6 Numerical Results for the Modified Cavity Flow When $n = 100$

N		39	79	159	179	199
$\mathbf{I}$	IT	157	221	246	260	241
	CPU	1.94	14.28	90.98	129.63	137.86
$\mathbf{A}^n_{\text{ILU}}$	IT	10	16	23	25	26
	CPU$_{\text{ILU}}$	0.38	2.89	22.44	33.39	46.55
	CPU	0.47	3.69	28.05	41.86	57.78
$\mathbf{P}^n$	IT	38	38	34	34	34
	CPU	1.72	9.38	54.75	74.44	97.86
$\mathbf{P}^n_{\text{ILU}}$	IT	39	40	37	38	36
	CPU$_{\text{ILU}}$	0.05	0.36	2.61	3.64	4.95
	CPU	0.38	1.83	9.00	12.09	14.83
$\mathbf{P}^n_{\text{UGS}}$	IT	68	93	101	107	100
	CPU	0.59	4.55	24.05	33.83	37.61

For the plane Couette-Poiseuille and the modified cavity flows, from Tables 3–6 we can obtain analogous observations and conclusions. Therefore, among all the preconditioners $\mathbf{P}^n_{\text{ILU}}$ is the most effective and $\mathbf{P}^n_{\text{UGS}}$ is the second most effective in terms of the CPU times.

4.3 Discussion of eigen-properties

In Figs. 7, 8, 9, 10, 11 and 12 we depict the eigenvalue distributions of the matrices $\mathbf{A}^n$, $\mathbf{P}^n$, $(\mathbf{P}^n)^{-1}\mathbf{A}^n$, $(\mathbf{A}^n_{\text{ILU}})^{-1}\mathbf{A}^n$, $(\mathbf{P}^n_{\text{ILU}})^{-1}\mathbf{A}^n$ and $(\mathbf{P}^n_{\text{UGS}})^{-1}\mathbf{A}^n$ for the plane Poiseuille flow, the plane Couette-Poiseuille flow and the modified cavity flow, when $N = 19$ and $n = 100$, respectively.

From these figures we see that all eigenvalues of $\mathbf{A}^n$ and $\mathbf{P}^n$ ($n = 100$) are located on the right half of the complex plane, so these matrices could be positive definite and even strictly diagonally dominant, which coincides with the theoretical results established in Sections 2.4 and 3.

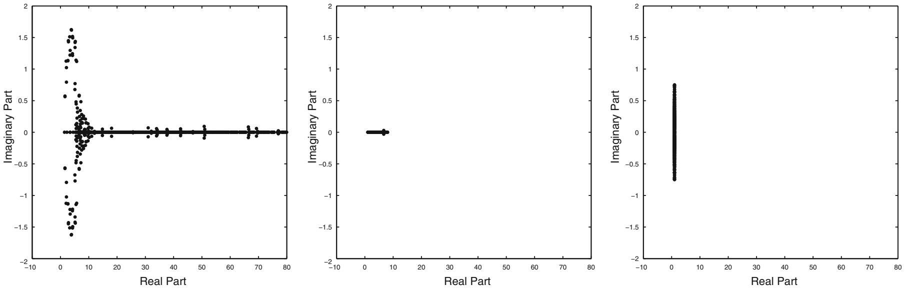

Fig. 7 Eigenvalue distributions for the original matrix $\mathbf{A}^n$ (*left*), the approximate matrix $\mathbf{P}^n$ (*middle*), and the preconditioned matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ (*right*) with respect to the plane Poiseuille flow when $N = 19$ and $n = 100$. The Euclidean condition numbers of their eigenvector matrices are $5.29e + 02$, $2.92e + 01$ and $1.24e + 04$, respectively

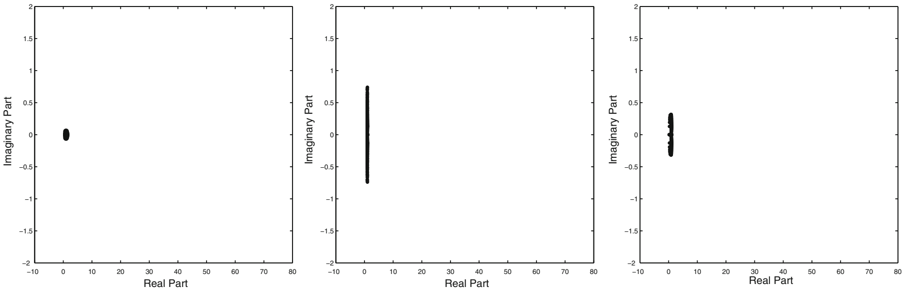

Fig. 8 Eigenvalue distributions of the preconditioned matrices with respect to the ILU factorization and the UGS splitting for the original matrix $\mathbf{A}^n$ and the approximate matrix $\mathbf{P}^n$ with respect to the plane Poiseuille flow when $N = 19$ and $n = 100$; $(\mathbf{A}^n_{\text{ILU}})^{-1}\mathbf{A}^n$ (*left*), $(\mathbf{P}^n_{\text{ILU}})^{-1}\mathbf{A}^n$ (*middle*), and $(\mathbf{P}^n_{\text{UGS}})^{-1}\mathbf{A}^n$ (*right*). The Euclidean condition numbers of their eigenvector matrices are $3.37e + 05$, $2.47e + 02$ and $9.08e + 04$, respectively

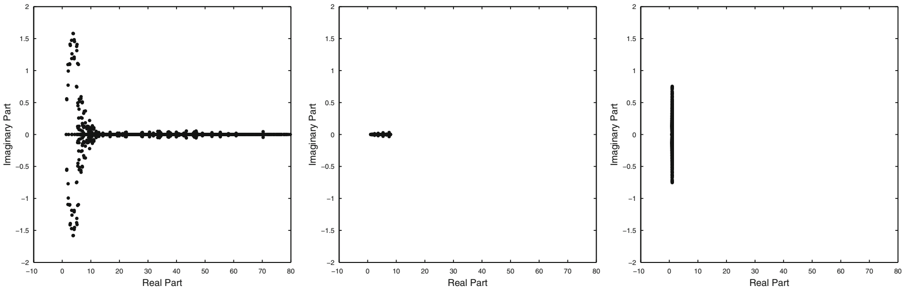

Fig. 9 Eigenvalue distributions for the original matrix $\mathbf{A}^n$ (*left*), the approximate matrix $\mathbf{P}^n$ (*middle*), and the preconditioned matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ (*right*) with respect to the plane Couette-Poiseuille flow when $N = 19$ and $n = 100$. The Euclidean condition numbers of their eigenvector matrices are $8.38e + 02$, $4.63e + 01$ and $5.90e + 03$, respectively

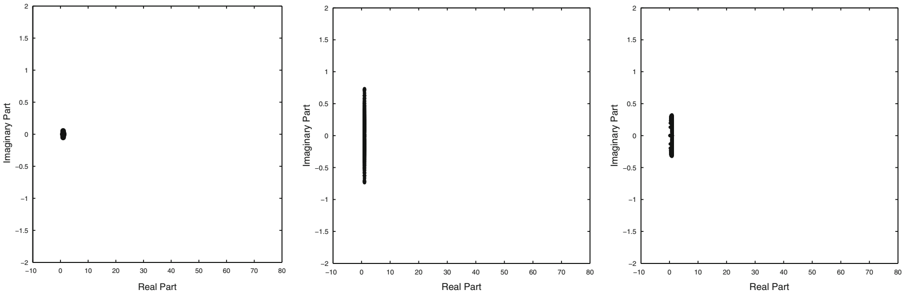

Fig. 10 Eigenvalue distributions of the preconditioned matrices with respect to the ILU factorization and the UGS splitting for the original matrix $\mathbf{A}^n$ and the approximate matrix $\mathbf{P}^n$ with respect to the plane Couette-Poiseuille flow when $N = 19$ and $n = 100$; $(\mathbf{A}^n_{\text{ILU}})^{-1}\mathbf{A}^n$ (*left*), $(\mathbf{P}^n_{\text{ILU}})^{-1}\mathbf{A}^n$ (*middle*), and $(\mathbf{P}^n_{\text{UGS}})^{-1}\mathbf{A}^n$ (*right*). The Euclidean condition numbers of their eigenvector matrices are $1.42e + 10$, $3.53e + 02$ and $1.39e + 05$, respectively

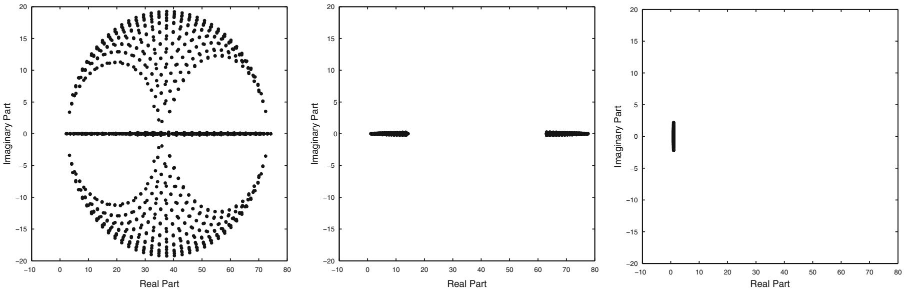

Fig. 11 Eigenvalue distributions for the original matrix $\mathbf{A}^n$ (*left*), the approximate matrix $\mathbf{P}^n$ (*middle*), and the preconditioned matrix $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ (*right*) with respect to the modified cavity flow when $N = 19$ and $n = 100$. The Euclidean condition numbers of their eigenvector matrices are $9.52e + 02$, $1.04e + 02$ and $3.23e + 03$, respectively

The eigenvalues of $\mathbf{P}^n$ are much more clustered than those of $\mathbf{A}^n$, as they are located closely to the real axis, with their real parts being bounded within relatively small intervals. We recall that $\mathbf{P}^n$ is a trivial approximation to $\mathbf{A}^n$ after selectively dropping the off-diagonal elements. Hence, via $\mathbf{P}^n$ we may construct economical and high-quality preconditioning matrices for the original matrix $\mathbf{A}^n$ in actual computations.

Many eigenvalues of $\mathbf{A}^n$ are located around the origin and some have very large real and imaginary parts. And the condition numbers of its eigenvector matrices are of the order 10^2. So this matrix is highly ill-conditioned and, as a result, the GMRES method may converge very slowly to the solution of the system of linear equations (2.11). For the preconditioning matrices $\mathbf{A}^n_{\mathrm{ILU}}$, $\mathbf{P}^n$, $\mathbf{P}^n_{\mathrm{ILU}}$ and $\mathbf{P}^n_{\mathrm{UGS}}$, the eigenvalues of $(\mathbf{A}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$ are the most clustered, and those of $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ and $(\mathbf{P}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$ are about equally clustered. Moreover, the condition numbers of their eigenvector matrices are constantly bounded. Hence, in iteration steps the $\mathbf{A}^n_{\mathrm{ILU}}$-preconditioned GMRES method is the fastest, and the $\mathbf{P}^n$- and the

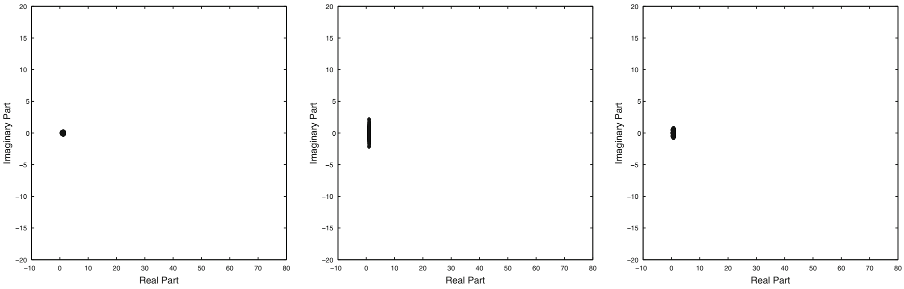

Fig. 12 Eigenvalue distributions of the preconditioned matrices with respect to the ILU factorization and the UGS splitting for the original matrix $\mathbf{A}^n$ and the approximate matrix $\mathbf{P}^n$ with respect to the modified cavity flow when $N = 19$ and $n = 100$; $(\mathbf{A}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$ (*left*), $(\mathbf{P}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$ (*middle*), and $(\mathbf{P}^n_{\mathrm{UGS}})^{-1}\mathbf{A}^n$ (*right*). The Euclidean condition numbers of their eigenvector matrices are $1.79e + 07$, $4.75e + 02$ and $4.93e + 03$, respectively

$\mathbf{P}^n_{\mathrm{ILU}}$-preconditioned GMRES methods are comparable. These observations and analyses conform with the numerical behaviors shown in Tables 1–6.

As an exception, we see from Figs. 7–12 that the eigenvalues of $(\mathbf{P}^n_{\mathrm{UGS}})^{-1}\mathbf{A}^n$ are less clustered than those of $(\mathbf{A}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$ and are more clustered than those of both $(\mathbf{P}^n)^{-1}\mathbf{A}^n$ and $(\mathbf{P}^n_{\mathrm{ILU}})^{-1}\mathbf{A}^n$, but from Tables 1–6 the iteration steps of the $\mathbf{P}^n_{\mathrm{UGS}}$-preconditioned GMRES method are the smallest among all the preconditioned GMRES methods. This phenomenon may occur due to the highly ill-conditioning of the eigenvectors of $(\mathbf{P}^n_{\mathrm{UGS}})^{-1}\mathbf{A}^n$.

5 Concluding remarks

We have constructed and analyzed the approximated ILU factorization and the approximated UGS splitting preconditioning matrices for the coefficient matrix $\mathbf{A}^n$ of the discretized linear system (2.11) from the two-dimensional steady-state incompressible Navier-Stokes equations (2.1). Both theoretical analyses and numerical experiments have shown that these structured preconditioners can efficiently improve the convergence property of the GMRES method when used to solve the discretized linear system (2.11), resulting in a reliable and effective numerical process, including the discretization, the linearization and the GMRES solve, for solving the two-dimensional steady-state incompressible Navier-Stokes equations (2.1). This new approach provides one feasible way for designing other economical and high-quality preconditioning matrices for the matrix $\mathbf{A}^n$, that is, by applying the well-known sparse factorizations and matrix splittings to the matrix $\mathbf{P}^n$, we may obtain effective preconditioners like incomplete orthogonal-triangular factorization [6, 14], sparse approximate inverse [16], Hermitian and skew-Hermitian splitting [9], and positive-definite and skew-Hermitian splitting [8].

References

1. Bai, Z.-Z.: A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations. Adv. Comput. Math. **10**, 169–186 (1999)
2. Bai, Z.-Z.: Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems. Appl. Math. Comput. **109**, 273–285 (2000)
3. Bai, Z.-Z.: Modified block SSOR preconditioners for symmetric positive definite linear systems. Ann. Oper. Res. **103**, 263–282 (2001)
4. Bai, Z.-Z.: Structured preconditioners for nonsingular matrices of block two-by-two structures. Math. Comput. **75**, 791–815 (2005)
5. Bai, Z.-Z.: Eigenvalue estimates for saddle point matrices of Hermitian and indefinite leading blocks. J. Comput. Appl. Math. **237**, 295–306 (2013)
6. Bai, Z.-Z., Duff, I.S., Wathen, A.J.: A class of incomplete orthogonal factorization methods. I: Methods and theories. BIT Numer. Math. **41**, 53–70 (2001)
7. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
8. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput. **26**, 844–863 (2005)
9. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)

10. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
11. Bai, Z.-Z., Ng, M.K.: Preconditioners for nonsymmetric block Toeplitz-like-plus-diagonal linear systems. Numer. Math. **96**, 197–220 (2003)
12. Bai, Z.-Z., Ng, M.K., Wang, Z.-Q.: Constraint preconditioners for symmetric indefinite matrices. SIAM J. Matrix Anal. Appl. **31**, 410–433 (2009)
13. Bai, Z.-Z., Wang, D.-R.: Generalized matrix multisplitting relaxation methods and their convergence. Numer. Math. J. Chinese Univ. (English Ser.) **2**, 87–100 (1993)
14. Bai, Z.-Z., Yin, J.-F.: Modified incomplete orthogonal factorization methods using Givens rotations. Computing **86**, 53–69 (2009)
15. Beam, R.M., Warming, R.F.: An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. J. Comput. Phys. **22**, 87–110 (1976)
16. Benzi, M., Tůma, M.: A comparative study of sparse approximate inverse preconditioners. Appl. Numer. Math. **30**, 305–340 (1999)
17. Berman, A., Plemmons, R.J.: Nonnegative Matrices in the Mathematical Sciences. Academic Press, New York (1979)
18. Chorin, A.J.: A numerical method for solving incompressible viscous flow problems. J. Comput. Phys. **2**, 12–26 (1967)
19. Chorin, A.J.: Numerical solution of the Navier-Stokes equations. Math. Comput. **22**, 745–762 (1968)
20. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics. Oxford University Press, New York (2005)
21. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
22. Greenbaum, A., Pták, V., Strakoš, Z.: Any nonincreasing convergence curve is possible for GMRES. SIAM J. Matrix Anal. Appl. **17**, 465–469 (1996)
23. Gresho, P.M.: Incompressible fluid dynamics: some fundamental formulation issues. Ann. Rev. Fluid Mech. **23**, 413–453 (1991)
24. MacCormack, R.W., Candler, G.V.: The solution of the Navier-Stokes equations using Gauss-Seidel line relaxation. Comput. Fluids **17**, 135–150 (1989)
25. Orlandi, P.: Fluid Flow Phenomena: A Numerical Toolkit. Kluwer Academic Publishers, Dordrecht (2000)
26. Peaceman, D.W., Rachford Jr., H.H.: The numerical solution of parabolic and elliptic differential equations. J. Soc. Indust. Appl. Math. **3**, 28–41 (1955)
27. Pulliam, T.H., Chaussee, D.S.: A diagonal form of an implicit approximate-factorization algorithm. J. Comput. Phys. **39**, 347–363 (1981)
28. Ran, Y.-H., Yuan, L.: On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems. Appl. Math. Comput. **217**, 3050–3068 (2010)
29. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
30. Shah, A., Guo, H., Yuan, L.: A third-order upwind compact scheme on curvilinear meshes for the incompressible Navier-Stokes equations. Commun. Comput. Phys. **5**, 712–729 (2009)
31. Shah, A., Yuan, L.: Flux-difference splitting-based upwind compact schemes for the incompressible Navier-Stokes equations. Intern. J. Numer. Methods Fluids **61**, 552–568 (2009)
32. Shah, A., Yuan, L., Khan, A.: Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier-Stokes equations. Appl. Math. Comput. **215**, 3201–3213 (2010)
33. Yoon, S., Jameson, A.: Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. AIAA J. **26**, 1025–1026 (1988)