Applied Mathematics and Computation 217 (2010) 3050–3068

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems

Yu-Hong Ran *, Li 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, Beijing 100190, PR China

ARTICLE INFO

Keywords:
Steady incompressible viscous flow
Block SSOR iteration method
Block LU factorization
Block diagonally dominant
Preconditioning

ABSTRACT

In order to solve the large sparse systems of linear equations arising from numerical solutions of two-dimensional steady incompressible viscous flow problems in primitive variable formulation, we present block SSOR and modified block SSOR iteration methods based on the special structures of the coefficient matrices. In each step of the block SSOR iteration, we employ the block LU factorization to solve the sub-systems of linear equations. We show that the block LU factorization is existent and stable when the coefficient matrices are block diagonally dominant of type-II by columns. Under suitable conditions, we establish convergence theorems for both block SSOR and modified block SSOR iteration methods. In addition, the block SSOR iteration and AF-ADI method are considered as preconditioners for the nonsymmetric systems of linear equations. Numerical experiments show that both block SSOR and modified block SSOR iterations are feasible iterative solvers and they are also effective for preconditioning Krylov subspace methods such as GMRES and BiCGSTAB when used to solve this class of systems of linear equations.

1. Introduction

The incompressible Navier–Stokes equations are one of the fundamental governing equations of fluid motion. The numerical solution of these equations is an indispensable tool for understanding the behavior of complicated fluid flow problems. Many computational methods for solving these equations have been developed, see Refs. [14,16] for a review. Refs. [18,19] developed a simple and accurate discretization method for numerical solution of two-dimensional incompressible viscous flow problems, which uses familiar third-order and fifth-order upwind compact finite difference schemes in conjunction with the well-known artificial compressibility approach [10]. In Refs. [18–20], ADI method was used to solve the discretized equations. In this paper, we start from implicit discretized equations obtained by using the schemes presented in Refs. [18,19] to discrete the governing equations in 2D composed of $(N_x + 2) \times (N_y + 2)$ grid points. Then we usually have to solve the large sparse system of linear equations

$$\mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{m\times m} \ \text{ nonsingular}, \quad x, b \in \mathbb{R}^m \tag{1}$$

* Corresponding author.
E-mail addresses: ranyh@lsec.cc.ac.cn (Y.-H. Ran), lyuan@lsec.cc.ac.cn (L. Yuan).

0096-3003/$ - see front matter Crown Copyright © 2010 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.08.039

with

$$\mathcal{A} = \begin{pmatrix} D_1 & U_2 & \cdots & 0 \\ L_1 & D_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & U_{N_y} \\ 0 & \cdots & L_{N_y-1} & D_{N_y} \end{pmatrix}, \tag{2}$$

$$D_j = \begin{pmatrix} E_{1j} & C_{2j} & \cdots & 0 \\ F_{1j} & E_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{N_x j} \\ 0 & \cdots & F_{N_x-1\,j} & E_{N_x j} \end{pmatrix}, \quad j = 1, \ldots, N_y, \tag{3}$$

$$L_j = \begin{pmatrix} G_{1j} & 0 & \cdots & 0 \\ 0 & G_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{N_x j} \end{pmatrix}, \quad j = 1, \ldots, N_y - 1,$$

$$U_j = \begin{pmatrix} H_{1j} & 0 & \cdots & 0 \\ 0 & H_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_{N_x j} \end{pmatrix}, \quad j = 2, \ldots, N_y,$$

where D_j is block tridiagonal matrix, L_j and U_j are block diagonal matrices, E_{ij} is 3×3 diagonal matrix, C_{ij}, F_{ij}, G_{ij} and H_{ij} are 3×3 dense matrices. Then $3 \times N_x \times N_y = m$. This class of linear systems also arises in many other applications.

In CFD, people usually employ approximate factorization and alternating direction implicit (AF-ADI) method [9,17], or LU-SGS method [21], or line relaxation method [15], but seldom direct methods to solve the system of linear equations (1) due to heavy cost. In Refs. [18,19], the authors used AF-ADI method. Although AF-ADI method is very simple, it has some drawbacks:

(a) It applies only to structured grid;
(b) It brings a fixed error which is related to the time step size and artificial compressibility factor, and this may affect global convergent rate.

In order to avoid these deficiencies, we present some iteration methods to solve the system of linear equations. In Refs. [1,2], a class of modified block SSOR preconditioners is presented for solving symmetric positive definite systems of linear equations, and these methods are very robust. Based on the special structure of the coefficient matrix $\mathcal{A}$, we present block SSOR and modified block SSOR iteration methods. Theoretical analysis shows that, under certain conditions, block SSOR and modified block SSOR iteration methods are convergent for nonsymmetric system of linear equations (1). In addition, the block SSOR iteration and AF-ADI method are considered as preconditioners for Krylov subspace methods such as GMRES and BiCGSTAB when they are used to solve this class of linear systems. The computational cost of these iterations is less than that of AF-ADI method, and the iteration error can be controlled to speed up global convergence rate. Numerical experiments show that both block SSOR and modified block SSOR iterations are superior to AF-ADI method and the two preconditioners for Krylov subspace methods such as GMRES and BiCGSTAB are very effective when they are used to solve this class of linear systems.

The paper is organized as follows: In Section 2, we brief the numerical method for incompressible viscous flow problems in primitive variable formulation presented in [18,19]. In addition, we show that block LU factorization is existent and stable when the coefficient matrix is block diagonally dominant of type-II by columns. In Section 3, because of the special structure of the coefficient matrix $\mathcal{A}$ of the linear systems, block SSOR iteration and its convergence analysis are presented. In each step of block SSOR iteration, we can implement block LU factorization to solve sub-systems of linear equations whose coefficient matrices are block tridiagonal. In Section 4, modified block SSOR iteration and its convergence analysis are presented. In Section 5, the block SSOR iteration and AF-ADI method can be considered as preconditioners for this nonsymmetric system of linear equations. In Section 6, Numerical experiments show that both block SSOR and modified block SSOR iterations are feasible iterative solvers and they are also effective for preconditioning Krylov subspace methods such as GMRES and BiCGSTAB when used to solve this class of systems of linear equations.

2. Discretization of governing equations and block diagonally dominant of type-II

This section presents an artificial compressibility-based numerical method using high-order accurate upwind compact finite difference schemes for the numerical solution of two-dimensional incompressible viscous flow problems in primitive variable formulation [18,19]. These methods rely on the artificial compressibility approach. The convective terms in the governing equations can be discretized by third-order or fifth-order upwind compact scheme based on the flux-difference splitting. To show the order of accuracy for the two upwind compact schemes, fourth-order or sixth-order central compact scheme is employed for the viscous terms. After discretization, we get the large sparse system of linear equations whose coefficient matrix is of blocked structure. Finally, we show that the block LU factorization is existent and stable when the coefficient matrix is block diagonally dominant of type-II by columns.

2.1. Governing equations

The governing two-dimensional steady incompressible Navier–Stokes equations in Cartesian coordinates (x,y) in dimensionless form and in the absence of body force are

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{4}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{5}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \tag{6}$$

here u, v are the velocity components, p is the pressure and Re is the Reynolds number. By introducing pseudo-time derivatives into the continuity and momentum equations, we have

$$\frac{\partial p}{\partial \tau}+\beta\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0, \tag{7}$$

$$\frac{\partial u}{\partial \tau}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{8}$$

$$\frac{\partial v}{\partial \tau}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \tag{9}$$

where τ is pseudo-time, β is the artificial compressibility parameter whose value is important to the performance of the artificial compressibility method. Eqs. (7)–(9) can also be written as:

$$\frac{\partial Q}{\partial \tau}+\frac{\partial (E-E_v)}{\partial x}+\frac{\partial (F-F_v)}{\partial y}=0, \tag{10}$$

where $Q=[p,u,v]^T$ is the solution variable vector, E, F and E_v, F_v are the inviscid and viscous flux vectors, respectively, i.e.,

$$E=\begin{pmatrix}\beta u\\ u^2+p\\ uv\end{pmatrix},\quad F=\begin{pmatrix}\beta v\\ uv\\ v^2+p\end{pmatrix},\quad E_v=\frac{1}{Re}\begin{pmatrix}0\\ u_x\\ v_x\end{pmatrix},\quad F_v=\frac{1}{Re}\begin{pmatrix}0\\ u_y\\ v_y\end{pmatrix}.$$

The Jacobian matrices A and B of the inviscid flux vectors are

$$A=\frac{\partial E}{\partial Q}=\begin{pmatrix}0 & \beta & 0\\ 1 & 2u & 0\\ 0 & v & u\end{pmatrix},\quad B=\frac{\partial F}{\partial Q}=\begin{pmatrix}0 & 0 & \beta\\ 0 & v & u\\ 1 & 0 & 2v\end{pmatrix},$$

and the Jacobian matrices A_v and B_v of the viscous flux vectors are

$$A_v=\frac{\partial E_v}{\partial Q}=\frac{1}{Re}I_m\frac{\partial}{\partial x},\quad B_v=\frac{\partial F_v}{\partial Q}=\frac{1}{Re}I_m\frac{\partial}{\partial y},\quad \text{with } I_m=\begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}.$$

It is possible to diagonalize A and B by using similarity transform as:

$$A=X\Lambda_A X^{-1},\quad B=Y\Lambda_B Y^{-1},$$

where diagonal matrices Λ_A and Λ_B contain the eigenvalues of matrices A and B:

$$diag(\Lambda_A)=\{u-c_1,u,u+c_1\},\quad diag(\Lambda_B)=\{v-c_2,v,v+c_2\},$$

with $c_1 = \sqrt{u^2+\beta}$ and $c_2 = \sqrt{v^2+\beta}$. X and Y are the matrices of the right eigenvectors, while X^{-1} and Y^{-1} are their inverses respectively. The eigenvalues of the Jacobian matrix play an important role in determining the mathematical characteristics of the governing equations. More importantly, it gives the speed and direction of propagation providing basis for the development of an upwind scheme.

2.2. Time and spatial discretization

By applying first-order backward difference to the pseudo-time derivative, one obtains implicit scheme

$$\frac{\Delta Q^n}{\Delta\tau} = -\left[\frac{\partial(E-E_v)}{\partial x} + \frac{\partial(F-F_v)}{\partial y}\right]^{n+1}, \tag{11}$$

where $\Delta Q^n = Q^{n+1} - Q^n$, n is the pseudo-time level (the number of iterations), $\Delta\tau$ is the pseudo-time step size which is determined based on the CFL number.

We linearize the implicit part of (11) by using Taylor's expansion

$$E^{n+1} \approx E^n + \left(\frac{\partial E}{\partial Q}\right)^n (Q^{n+1}-Q^n) = E^n + A^n\Delta Q^n,$$
$$F^{n+1} \approx F^n + \left(\frac{\partial F}{\partial Q}\right)^n (Q^{n+1}-Q^n) = F^n + B^n\Delta Q^n,$$
$$E_v^{n+1} \approx E_v^n + \left(\frac{\partial E_v}{\partial Q}\right)^n (Q^{n+1}-Q^n) = E_v^n + A_v^n\Delta Q^n,$$
$$F_v^{n+1} \approx F_v^n + \left(\frac{\partial F_v}{\partial Q}\right)^n (Q^{n+1}-Q^n) = F_v^n + B_v^n\Delta Q^n.$$

Thus we can obtain

$$\left[I + \Delta\tau\left(\frac{\partial(A-A_v)}{\partial x} + \frac{\partial(B-B_v)}{\partial y}\right)\right]^n \Delta Q^n = -\Delta\tau\left[\frac{\partial(E-E_v)}{\partial x} + \frac{\partial(F-F_v)}{\partial y}\right]^n \equiv S^n. \tag{12}$$

Convective terms in the left hand side of (12) are discretized by first-order upwind difference and viscous terms by traditional central difference, e.g.,

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

Owing to the hyperbolic nature of the system (10), the convective flux can be split in two parts, i.e., along the x-direction

$$E_x = E_x^+ + E_x^-, \tag{13}$$

where E^+ corresponds to the split flux in positive x-direction with information being propagated from left to right by positive eigenvalues and E^- corresponds to the split flux in negative x-direction with information being propagated from right to left by negative eigenvalues. To compute the two split derivatives in Eq. (13), we use the third-order and fifth-order upwind compact schemes, which are

$$\frac{2}{3}(E_x^+)_i + \frac{1}{3}(E_x^+)_{i-1} = \frac{5\delta^- E_i^+ + \delta^- E_{i+1}^+}{6\Delta x}, \tag{14}$$
$$\frac{2}{3}(E_x^-)_i + \frac{1}{3}(E_x^-)_{i+1} = \frac{5\delta^+ E_i^- + \delta^+ E_{i-1}^-}{6\Delta x}, \tag{15}$$

and

$$\frac{3}{5}(E_x^+)_i + \frac{2}{5}(E_x^+)_{i-1} = \frac{-\delta^- E_{i+2}^+ + 11\delta^- E_{i+1}^+ + 47\delta^- E_i^+ + 3\delta^- E_{i-1}^+}{60\Delta x}, \tag{16}$$
$$\frac{3}{5}(E_x^-)_i + \frac{2}{5}(E_x^-)_{i+1} = \frac{-\delta^+ E_{i-2}^- + 11\delta^+ E_{i-1}^- + 47\delta^+ E_i^- + 3\delta^+ E_{i+1}^-}{60\Delta x}, \tag{17}$$

respectively, where $\delta^+ f_i = f_{i+1} - f_i$, $\delta^- f_i = f_i - f_{i-1}$, and Δx is the grid spacing. Since each term in the right hand side of Eqs. (14) and (15) and (16) and (17) represents the difference of split fluxes between neighboring points, one can compute them by using FDS:

$$E_{i+1}^\pm - E_i^\pm \equiv \Delta E_{i+\frac{1}{2}}^\pm = A^\pm(\overline{Q})(Q_{i+1} - Q_i),$$

where $\Delta E_{i+\frac{1}{2}}^\pm$ is the flux-difference across the positive or negative traveling waves. The split Jacobian matrix is calculated by

$$A^\pm = X\Lambda_A^\pm X^{-1}$$

with

$$\Lambda_A^{\pm} = \frac{1}{2}(\Lambda_A \pm |\Lambda_A|),$$

which is evaluated using Roe average value $\overline{Q}$

$$\overline{Q} = \frac{1}{2}(Q_{i+1} + Q_i).$$

The second derivative for the viscous term in the right hand side of Eq. (12) is approximated by a fourth-order and sixth-order symmetric compact schemes at interior points:

$$\frac{1}{12}(S_{i-1} + 10S_i + S_{i+1}) = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}, \tag{18}$$

and

$$2S_{i-1} + 11S_i + 2S_{i+1} = 12\frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + \frac{3}{4}\frac{u_{i-2} - 2u_i + u_{i+2}}{\Delta x^2}, \tag{19}$$

where S_i denotes $(\partial^2 u/\partial x^2)_i$.

The upwind compact scheme (14), (15) or (16) and (17) and high-order central compact scheme (18) or (19) are used for the convective terms and the viscous terms in right hand side of (12). Convective terms in left hand side of (12) are discretized by first-order upwind difference and viscous terms by traditional central difference. Thus one obtains the following form

$$\left[I + \Delta\tau\left(\delta_x^- A^+ + \delta_x^+ A^- - I_m\frac{\delta_x^2}{Re}\right) + \Delta\tau\left(\delta_y^- B^+ + \delta_y^+ B^- - I_m\frac{\delta_y^2}{Re}\right)\right]^n \Delta Q^n = S^n. \tag{20}$$

In the left hand side of (20), A^+ and A^- are constructed so that eigenvalues of "+" matrices are non-negative and those of "−" matrices are non-positive:

$$A^{\pm} = \frac{1}{2}[A \pm \rho(A)I]$$

with the spectral radius of Jacobian

$$\rho(A) = \kappa \cdot \max[|\lambda(A)|],$$

$\lambda(A)$ represents the eigenvalues of the Jacobian matrix A, $\kappa = 1$ for the third-order upwind compact scheme, and $\kappa \geqslant 1.3$ for the fifth-order upwind compact scheme. We let

$$\left[\Delta\tau\left(\delta_x^- A^+ + \delta_x^+ A^- - I_m\frac{\delta_x^2}{Re}\right)\right]^n = \mathcal{L}^n,$$

$$\left[\Delta\tau\left(\delta_y^- B^+ + \delta_y^+ B^- - I_m\frac{\delta_y^2}{Re}\right)\right]^n = \mathcal{U}^n,$$

then (20) can be written as:

$$(I + \mathcal{L}^n + \mathcal{U}^n)\Delta Q^n = S^n. \tag{21}$$

At last, we must solve Eq. (20) to obtain Q^{n+1} in each pseudo-time level (iteration), but the cost of direct solution is very large. Thus Refs. [18,19] implement AF-ADI method, i.e., Eq. (20) is approximated by

$$\left[I + \Delta\tau\left(\delta_x^- A^+ + \delta_x^+ A^- - I_m\frac{\delta_x^2}{Re}\right)\right]^n \left[I + \Delta\tau\left(\delta_y^- B^+ + \delta_y^+ B^- - I_m\frac{\delta_y^2}{Re}\right)\right]^n \cdot \Delta Q^n = S^n,$$

then we only need to solve two one-dimensional equations directly

$$\left[I + \Delta\tau\left(\delta_x^- A^+ + \delta_x^+ A^- - I_m\frac{\delta_x^2}{Re}\right)\right]^n \Delta Q^* = S^n,$$

$$\left[I + \Delta\tau\left(\delta_y^- B^+ + \delta_y^+ B^- - I_m\frac{\delta_y^2}{Re}\right)\right]^n \Delta Q^n = \Delta Q^*.$$

As a whole, AF-ADI method is an efficient direct method which reduce the dimension of equations, it still has some drawbacks as mentioned in the introduction. In order to avoid these deficiencies, we give some iteration methods to solve (20). We discrete Eq. (20) at point (i,j) and suppose that the computational grid used has $(N_x+2)\times(N_y+2)$ grid points, Δx is step size in the x-direction, Δy is step size in the y-direction, and for simplicity yet without loss of generality, the boundary condition is assumed to be Dirichlet type

$$\Delta Q_{i,j}=0 \quad \text{if} \quad i,j=0 \quad \text{or} \quad i=N_x+1, \quad j=N_y+1.$$

Then

$$\left[\Delta Q_{i,j}+\Delta\tau\left(\frac{A^+_{i,j}\Delta Q_{i,j}-A^+_{i-1,j}\Delta Q_{i-1,j}}{\Delta x}+\frac{A^-_{i+1,j}\Delta Q_{i+1,j}-A^-_{i,j}\Delta Q_{i,j}}{\Delta x}-I_m\frac{\Delta Q_{i+1,j}-2\Delta Q_{i,j}+\Delta Q_{i-1,j}}{Re\Delta x^2}\right)\right.$$
$$\left.+\Delta\tau\left(\frac{B^+_{i,j}\Delta Q_{i,j}-B^+_{i,j-1}\Delta Q_{i,j-1}}{\Delta y}+\frac{B^-_{i,j+1}\Delta Q_{i,j+1}-B^-_{i,j}\Delta Q_{i,j}}{\Delta y}-I_m\frac{\Delta Q_{i,j+1}-2\Delta Q_{i,j}+\Delta Q_{i,j-1}}{Re\Delta y^2}\right)\right]^n=S^n_{ij}. \tag{22}$$

We let

$$C_{i+1,j}=\frac{\Delta\tau}{\Delta x}A^-_{i+1,j}-\frac{\Delta\tau}{Re\Delta x^2}I_m, \quad H_{i,j+1}=\frac{\Delta\tau}{\Delta y}B^-_{i,j+1}-\frac{\Delta\tau}{Re\Delta y^2}I_m,$$
$$E'_{ij}=\frac{\Delta\tau}{\Delta x}(A^+_{i,j}-A^-_{i,j})+\frac{2\Delta\tau}{Re\Delta x^2}I_m, \quad E''_{ij}=\frac{\Delta\tau}{\Delta y}(B^+_{i,j}-B^-_{i,j})+\frac{2\Delta\tau}{Re\Delta y^2}I_m,$$
$$F_{i-1,j}=-\frac{\Delta\tau}{\Delta x}A^+_{i-1,j}-\frac{\Delta\tau}{Re\Delta x^2}I_m, \quad G_{i,j-1}=-\frac{\Delta\tau}{\Delta y}B^+_{i,j-1}-\frac{\Delta\tau}{Re\Delta y^2}I_m,$$
$$E_{ij}=I+E'_{ij}+E''_{ij},$$

here E_{ij}, E'_{ij}, E''_{ij} are 3×3 diagonal matrices, and C_{ij}, F_{ij}, G_{ij}, H_{ij} are 3×3 dense matrices. Form (22) with Dirichlet boundary condition, we get the block system of linear equations (1) and the time level n is omitted for brevity, where

$$x\equiv\Delta Q=[\Delta Q_{11}\cdots\Delta Q_{N_x1}\Delta Q_{12}\cdots\Delta Q_{N_x2}\cdots\Delta Q_{1N_y}\cdots\Delta Q_{N_xN_y}]^T,$$
$$b\equiv S=[S_{11}S_{21}\cdots S_{N_x1}S_{12}\cdots S_{N_x2}\cdots S_{1N_y}\cdots S_{N_xN_y}]^T.$$

According to (3), we have

$$D_j=\begin{pmatrix} I+E'_{1j}+E''_{1j} & C_{2j} & \cdots & 0\\ F_{1j} & I+E'_{2j}+E''_{2j} & \ddots & \vdots\\ \vdots & \ddots & \ddots & C_{N_xj}\\ 0 & \cdots & F_{N_x-1,j} & I+E'_{N_xj}+E''_{N_xj}\end{pmatrix}.$$

We let

$$D'_j=\begin{pmatrix} E'_{1j} & C_{2j} & \cdots & 0\\ F_{1j} & E'_{2j} & \ddots & \vdots\\ \vdots & \ddots & \ddots & C_{N_xj}\\ 0 & \cdots & F_{N_x-1,j} & E'_{N_xj}\end{pmatrix}, \quad D''_j=\begin{pmatrix} E''_{1j} & 0 & \cdots & 0\\ 0 & E''_{2j} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & E''_{N_xj}\end{pmatrix},$$

thus

$$D_j=I+D'_j+D''_j, \quad j=1,\cdots,N_y.$$

Because of (21), we have

$$\mathcal{A}=I+\mathcal{L}+\mathcal{U},$$

where

$$\mathcal{L}=\begin{pmatrix} D'_1 & 0 & \cdots & 0\\ 0 & D'_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & D'_{N_y}\end{pmatrix}, \quad \mathcal{U}=\begin{pmatrix} D''_1 & U_2 & \cdots & 0\\ L_1 & D''_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & U_{N_y}\\ 0 & \cdots & L_{N_y-1} & D''_{N_y}\end{pmatrix}.$$

2.3. Block diagonally dominant linear systems

Consider an $n \times n$ complex matrix A, which is partitioned as the following form

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1s} \\ A_{21} & A_{22} & \cdots & A_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ A_{s1} & A_{s2} & \cdots & A_{ss} \end{pmatrix},$$

where A_{ll} is an $n_l \times n_l$ nonsingular principle sub-matrix of A, $l = 1, \ldots, s$, $\sum n_l = n$, $\|\cdot\|$ is some consistent matrix norm. Let $S = 1, 2, \ldots, s$.

Definition 1 [13]. Let $A = (A_{lm})_{s\times s} \in c_s^{n\times n}$ and A_{ll}, $l = 1, \ldots, s$, be nonsingular. If

$$\|A_{ll}^{-1}\|^{-1} \geqslant \sum_{m=1, m\neq l}^{s} \|A_{lm}\|, \quad \forall l \in S, \tag{23}$$

or

$$\|A_{ll}^{-1}\|^{-1} \geqslant \sum_{m=1, m\neq l}^{s} \|A_{ml}\|, \quad \forall l \in S, \tag{24}$$

then A is block diagonally dominant of type-I (abbreviated as BDD-I) by rows or by columns; if the strict inequality in (23) or (24) is valid for all $l \in S$, then A is block strictly diagonally dominant of type-I (abbreviated as BSDD-I).

Corollary 1 [11]. *Suppose that $A \in R^{n\times n}$ is nonsingular and* BDD-I *by rows or columns with respect to a subordinate matrix norm in (23) or (24). Then A has a block* LU *factorization, and the block* LU *factorization is stable when A is* BDD-I *by columns. Unfortunately, the block* LU *factorization could be unstable when A is* BDD-I *by rows.*

Definition 2 [22]. Let $A = (A_{lm})_{s\times s} \in c_s^{n\times n}$ and A_{ll}, $l = 1, \ldots, s$, be nonsingular. If

$$1 \geqslant \sum_{m=1, m\neq l}^{s} \|A_{ll}^{-1} A_{lm}\|, \forall l \in S, \tag{25}$$

or

$$1 \geqslant \sum_{m=1, m\neq l}^{s} \|A_{ml} A_{ll}^{-1}\|, \forall l \in S, \tag{26}$$

then A is block diagonally dominant of type-II (abbreviated as BDD-II) by rows or by columns; if the strict inequality in (25) or (26) is valid for all $l \in S$, then A is block strictly diagonally dominant of type-II (abbreviated as BSDD-II).

Because $\|A_{ll}^{-1} A_{lm}\| \leqslant \|A_{ll}^{-1}\| \cdot \|A_{lm}\|$, we easily conclude that if A is BDD-I (BSDD-I), then A is BDD-II (BSDD-II).

Theorem 1. *Suppose that $A \in R^{n\times n}$ is nonsingular and* BDD-II *by rows or columns with respect to subordinate matrix norm in (25) or (26). If $A^{(k)}$ denotes the matrix obtained after $k-1$ steps of the block* LU *factorization, then*

$$\max_{k\leqslant i,j\leqslant s} \|A_{ij}^{(k)}\| \leqslant 2 \max_{1\leqslant i,j\leqslant s} \|A_{ij}\|.$$

Proof. Let A be BDD-II by rows (the proof for BDD-II by columns is similar), i.e.,

$$\sum_{j\neq i} \|A_{ii}^{-1} A_{ij}\| \leqslant 1.$$

Then

$$\sum_{j=2}^{s} \|A_{ij}^{(k)}\| = \sum_{j=2}^{s} \|A_{ij} - A_{i1} A_{11}^{-1} A_{1j}\| \leqslant \sum_{j=2}^{s} \|A_{ij}\| + \|A_{i1}\| \sum_{j=2}^{s} \|A_{11}^{-1} A_{1j}\| \leqslant \sum_{j=1}^{s} \|A_{ij}\|.$$

Using (25), it follows from induction that

$$\sum_{j=k}^{s} \|A_{ij}^{(k)}\| \leqslant \sum_{j=1}^{s} \|A_{ij}\|,$$

which yields

$$\max_{k\leqslant i,j\leqslant s}\|A_{ij}^{(k)}\|\leqslant\max_{k\leqslant i\leqslant s}\sum_{j=k}^{s}\|A_{ij}^{(k)}\|\leqslant\max_{k\leqslant i\leqslant s}\sum_{j=1}^{s}\|A_{ij}\|.$$

From

$$\sum_{j\neq i}\|A_{ii}^{-1}A_{ij}\|\leqslant 1,$$

we have

$$\|A_{ii}\|\geqslant\sum_{j\neq i}\|A_{ii}\|\|A_{ii}^{-1}A_{ij}\|\geqslant\sum_{j\neq i}\|A_{ii}A_{ii}^{-1}A_{ij}\|=\sum_{j\neq i}\|A_{ij}\|.$$

Then

$$\sum_{j=1}^{s}\|A_{ij}\|\leqslant 2\|A_{ii}\|$$

and

$$\max_{k\leqslant i\leqslant s}\sum_{j=1}^{s}\|A_{ij}\|\leqslant 2\max_{k\leqslant i\leqslant s}\|A_{ii}\|.$$

So

$$\max_{k\leqslant i,j\leqslant s}\|A_{ij}^{(k)}\|\leqslant\max_{k\leqslant i\leqslant s}\sum_{j=1}^{s}\|A_{ij}\|\leqslant 2\max_{k\leqslant i\leqslant s}\|A_{ii}\|\leqslant 2\max_{1\leqslant i,j\leqslant s}\|A_{ij}\|.$$

Theorem 2. *Suppose that $A\in R^{n\times n}$ is nonsingular and* BDD-II *by columns with respect to a subordinate matrix norm in (26), and the norm has the property*

$$\max_{i,j}\|A_{ij}\|\leqslant\|A\|\leqslant\sum_{i,j}\|A_{ij}\|,$$

which holds for any p norm. Then A has a stable block LU *factorization, and all the Schur complements arising in the block* LU *factorization are also* BDD-II *by columns.*

Proof. The proof for the case that all the Schur complements arising in the block LU factorization are also BDD-II by columns can be found in [22]. Because of this conclusion, we can easily prove that A has a block LU factorization. The stability of the block LU factorization will be proved as follows.

Suppose that the block LU factorization of A is

$$A=LU,$$

where L is a block lower triangular matrix and U is a block upper triangular matrix. Then we have

$$\left\|\left[L_{21}^{T},\cdots,L_{s,1}^{T}\right]^{T}\right\|\leqslant\|L_{21}\|+\|L_{31}\|+\cdots+\|L_{s1}\|=\left\|A_{21}A_{11}^{-1}\right\|+\left\|A_{31}A_{11}^{-1}\right\|+\cdots+\left\|A_{s1}A_{11}^{-1}\right\|=\sum_{i=2}^{s}\left\|A_{i1}A_{11}^{-1}\right\|\leqslant 1,$$

$$\left\|\left[L_{32}^{T},\cdots,L_{s,2}^{T}\right]^{T}\right\|\leqslant\|L_{32}\|+\|L_{42}\|+\cdots+\|L_{s2}\|=\left\|A_{32}^{(2)}A_{22}^{(2)-1}\right\|+\cdots+\left\|A_{s2}^{(2)}A_{22}^{(2)-1}\right\|=\sum_{i=3}^{s}\left\|A_{i2}^{(2)}A_{22}^{(2)-1}\right\|\leqslant 1,$$

$$\vdots$$

$$\left\|\left[L_{j+1\,j}^{T},\cdots,L_{s,j}^{T}\right]^{T}\right\|\leqslant\|L_{j+1\,j}\|+\|L_{j+2\,j}\|+\cdots+\|L_{s,j}\|\leqslant 1.$$

It follows that $\|L\|\leqslant 2s-1$. Since $U_{ij}=A_{ij}^{(i)}$ for $j\geqslant i$, Theorem 1 shows that $U_{ij}\leqslant 2\|A\|$ for each block of U, we know that $\|U\|\leqslant s(s+1)\|A\|$, hence, $\|L\|\|U\|\leqslant s(s+1)(2s-1)\|A\|$, we conclude that the block LU factorization is stable if A is BDD-II by columns. □

We remark that the block diagonally dominant matrices are closely related to the block H-matrices studied in [3]. For more details on the definitions and the associated iterative methods, we refer to [4,5,12].

3. The Block SSOR iteration

By decomposing the block matrix $\mathcal{A}$ into its block diagonal part D, strictly block lower triangular part L and strictly block upper triangular part U, i.e.,

$$\mathcal{A} = D + L + U$$

with

$$D = \begin{pmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_{N_y} \end{pmatrix}, \quad \text{where} \quad D_j = \begin{pmatrix} E_{1j} & C_{2j} & \cdots & 0 \\ F_{1j} & E_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{N_x j} \\ 0 & \cdots & F_{N_x-1\,j} & E_{N_x j} \end{pmatrix} \quad j = 1, \cdots, N_y,$$

$$L = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ L_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & L_{N_y-1} & 0 \end{pmatrix}, \quad \text{where} \quad L_j = \begin{pmatrix} G_{1j} & 0 & \cdots & 0 \\ 0 & G_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{N_x j} \end{pmatrix} \quad j = 1, \cdots, N_y - 1,$$

$$U = \begin{pmatrix} 0 & U_2 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & U_{N_y} \\ 0 & 0 & \cdots & 0 \end{pmatrix}, \quad \text{where} \quad U_j = \begin{pmatrix} H_{1j} & 0 & \cdots & 0 \\ 0 & H_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_{N_x j} \end{pmatrix} \quad j = 2, \cdots, N_y.$$

We consider the solution of the system of linear equations (1) by the block SSOR iteration (abbreviated as BSSOR):

$$\begin{cases} (D + \omega U)x_{k+\frac{1}{2}} = ((1-\omega)D - \omega L)x_k + \omega b, \\ (D + \omega L)x_{k+1} = ((1-\omega)D - \omega U)x_{k+\frac{1}{2}} + \omega b. \end{cases} \tag{27}$$

The residual form of the block SSOR iteration is

$$\begin{cases} x_{k+\frac{1}{2}} = x_k + \omega(D + \omega U)^{-1}(b - \mathcal{A}x_k), \\ x_{k+1} = x_{k+\frac{1}{2}} + \omega(D + \omega L)^{-1}(b - \mathcal{A}x_{k+\frac{1}{2}}). \end{cases} \tag{28}$$

When $\omega = 1$, the iteration (27) or (28) is the block symmetric Gauss–Seidel iteration (abbreviated as block SGS):

$$\begin{cases} (D + U)x_{k+\frac{1}{2}} = -Lx_k + b, \\ (D + L)x_{k+1} = -Ux_{k+\frac{1}{2}} + b. \end{cases} \tag{29}$$

In each step of (27) we must solve $2 \times N_y$ sub-systems of linear equations with the coefficient matrices $D_{N_y}, \ldots, D_1$ and $D_1, \ldots, D_{N_y}$. Under the following condition, we can employ block LU factorization to solve these $2 \times N_y$ sub-systems.

Theorem 3. *If D_j is* BDD-II *by columns with the partitioned form as (3), then D_j has stable block* LU *factorization.*

We consider sub-systems of linear equations with coefficient matrices D_j, $1 \leqslant j \leqslant N_y$, which are partitioned as follows

$$D_j = \begin{pmatrix} E_{1j} & C_{2j} & \cdots & 0 \\ F_{1j} & E_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{N_x j} \\ 0 & \cdots & F_{N_x-1\,j} & E_{N_x j} \end{pmatrix}, \quad j = 1, \cdots, N_y.$$

Since $C_{ij}E_{ij}^{-1}$, $F_{ij}E_{ij}^{-1}$ are 3×3 matrices, it is easy to get the norm of them. It holds that if $\|C_{ij}E_{ij}^{-1}\| + \|F_{ij}E_{ij}^{-1}\| \leqslant 1$, $i = 1, \cdots, N_x$, then D_j is BDD-II by columns with the partitioned form as (3).

We give the convergence analysis of the block SGS iteration (29) as follows.

Theorem 4 [22]. *If $\mathcal{A}$ is* BSDD-II *by rows or by columns with the partitioned form as (2), then $\mathcal{A}$ is nonsingular.*

Theorem 5. *If $\mathcal{A}$ is* BSDD-II *by rows with the partitioned form as (2), then block* SGS *iteration (29) is convergent.*

Proof. The iteration matrix of block SGS iteration (29) is $(D+L)^{-1}U(D+U)^{-1}L$. In order to prove block SGS iteration (29) is convergent, we only need to prove that $\rho((D+L)^{-1}U(D+U)^{-1}L)<1$. Let the following $\|\cdot\|$ is ∞-norm, $L_0=U_{N_y+1}=0$ and $y=(D+L)^{-1}Ux$, so it holds that

$$\|(D+L)^{-1}U\|=\max_{\|x\|=1}\|y\|,$$

by making use of $y=(D+L)^{-1}Ux$, we can obtain $(D+L)y=Ux$, i.e.,

$$y=-D^{-1}Ly+D^{-1}Ux,$$

here

$$D^{-1}L=\begin{pmatrix}0 & 0 & \cdots & 0\\ D_2^{-1}L_1 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & D_{N_y}^{-1}L_{N_y-1} & 0\end{pmatrix},$$

$$D^{-1}U=\begin{pmatrix}0 & D_1^{-1}U_2 & \cdots & 0\\ 0 & 0 & \ddots & \vdots\\ \vdots & \vdots & \ddots & D_{N_y-1}^{-1}U_{N_y}\\ 0 & 0 & \cdots & 0\end{pmatrix},$$

thus there exists some $1\leqslant i\leqslant N_y$, s.t.,

$$\|y\|=\|-D_i^{-1}L_{i-1}y_{i-1}+D_i^{-1}U_{i+1}x_{i+1}\|\leqslant\|D_i^{-1}L_{i-1}\|\|y\|+\|D_i^{-1}U_{i+1}\|\|x\|,$$

hence

$$\|y\|\leqslant\frac{\|D_i^{-1}U_{i+1}\|\|x\|}{1-\|D_i^{-1}L_{i-1}\|},$$

thus

$$\max_{\|x\|=1}\|y\|\leqslant\max_{\|x\|=1}\frac{\|D_i^{-1}U_{i+1}\|}{1-\|D_i^{-1}L_{i-1}\|},$$

because $\mathcal{A}$ is BSDD-II by rows, i.e.,

$$\|D_i^{-1}U_{i+1}\|+\|D_i^{-1}L_{i-1}\|<1,$$

thus

$$\|(D+L)^{-1}U\|=\max_{\|x\|=1}\|y\|<1.$$

The proof for $\|(D+U)^{-1}L\|<1$ is similar. Thus

$$\rho((D+L)^{-1}U(D+U)^{-1}L)<\|(D+L)^{-1}U\|\cdot\|(D+U)^{-1}L\|<1. \qquad \square$$

We point out that the conditions of Theorems 3–5 also hold true when $\mathcal{A}$ is a block H-matrix; see [6,7].

Remark. AF-ADI method needs to solve N_x+N_y systems of linear equations whose coefficient matrices are block tridiagonal with the same structure of D_j, and block SSOR needs to solve N_y systems of linear equations with coefficient matrices $D_1,\ldots,D_{N_y}$. We can say that, in some sense, block SSOR method is superior to the AF-ADI method.

4. The Modified Block SSOR iteration

In each step of (27) or (28) we must solve $2\times N_y$ sub-systems of linear equations with coefficient matrices $D_{N_y},\ldots,D_1$ and $D_1,\ldots,D_{N_y}$. In many applications, the cost of the direct solution of these sub-systems is prohibitively large. Therefore, we replace the block diagonal matrix D in (28) by another nonsingular, block diagonal matrix $\widetilde{D}$ obtaining the modified block SSOR iteration (abbreviated as MBSSOR):

$$\begin{cases}x_{k+\frac{1}{2}}=x_k+\omega(\widetilde{D}+\omega U)^{-1}(b-\mathcal{A}x_k),\\ x_{k+1}=x_{k+\frac{1}{2}}+\omega(\widetilde{D}+\omega L)^{-1}(b-\mathcal{A}x_{k+\frac{1}{2}}).\end{cases}\tag{30}$$

Another form of the modified block SSOR iteration is

$$\begin{cases} (\widetilde{D}+\omega U)x_{k+\frac{1}{2}} = (\widetilde{D}-\omega D-\omega L)x_k+\omega b, \\ (\widetilde{D}+\omega L)x_{k+1} = (\widetilde{D}-\omega D-\omega U)x_{k+\frac{1}{2}}+\omega b. \end{cases} \tag{31}$$

When $\omega = 1$, iteration (30) or (31) is modified block symmetric Gauss–Seidel iteration (abbreviated as modified block SGS):

$$\begin{cases} (\widetilde{D}+U)x_{k+\frac{1}{2}} = (\widetilde{D}-D-L)x_k+b, \\ (\widetilde{D}+L)x_{k+1} = (\widetilde{D}-D-U)x_{k+\frac{1}{2}}+b. \end{cases} \tag{32}$$

Refs. [1,2] presented a class of modified block SSOR iteration for symmetric positive definite systems of linear equations, and gave several choices of the matrix $\widetilde{D}$. We consider the following choice of the matrix $\widetilde{D}$, by decomposing the every diagonal block $D_j, j = 1,\ldots,N_y$ of the block diagonal matrix D into its block diagonal part d_j, block lower triangular part l_j and block upper triangular part u_j, i.e.,

$$D_j = d_j + l_j + u_j, \quad j = 1,\ldots,N_y, \tag{33}$$

where

$$d_j = \begin{pmatrix} E_{1j} & 0 & \cdots & 0 \\ 0 & E_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_{N_x j} \end{pmatrix}, \quad l_j = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ F_{1j} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & F_{N_x-1,j} & 0 \end{pmatrix},$$

$$u_j = \begin{pmatrix} 0 & C_{2j} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & C_{N_x j} \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

d_j is diagonal matrix, l_j is strictly lower triangular matrix and u_j is strictly upper triangular matrix. We consider the symmetric Gauss–Seidel iteration to linear systems with coefficient matrix D_j, and corresponding to the splitting $D_j = M_j - \Delta_j$, then we have

$$M_j = (d_j + u_j)d_j^{-1}(d_j + l_j), \quad j = 1,\ldots,N_y,$$

where d_j, l_j, u_j is the decomposition (33) of D_j, and

$$\Delta_j = M_j - D_j = u_j d_j^{-1} l_j, \quad j = 1,\ldots,N_y.$$

In fact, M_j is a approximation of the D_j, i.e., $D_j \approx M_j$, thus we can replace D_j by M_j for $j = 1,\ldots,N_y$. We let

$$\widetilde{D} = \begin{pmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_{N_y} \end{pmatrix} = \begin{pmatrix} (d_1+u_1)d_1^{-1}(d_1+l_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (d_{N_y}+u_{N_y})d_{N_y}^{-1}(d_{N_y}+l_{N_y}) \end{pmatrix},$$

after replacement, because of the special structures of d_j, l_j, u_j, in each step of (30), $2 \times N_y$ sub-systems of linear equations with the coefficient matrices $M_{N_y},\ldots,M_1$ and $M_1,\ldots,M_{N_y}$ can be solved directly and quickly.

We give the convergence analysis of the modified block SGS iteration (32) as follows.

Theorem 6. *We introduce matrix*

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \Delta_1 & M_1 & U_2 & \cdots & 0 \\ 0 & L_1+\Delta_2 & M_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & U_{N_y} \\ 0 & 0 & \cdots & L_{N_y-1}+\Delta_{N_y} & M_{N_y} \end{pmatrix}.$$

If P is BSDD-II *by rows (except for the first row) with respect to ∞-norm with the partitioned form as above, then modified block* SGS *iteration (32) is convergent.*

Proof. The iteration matrix of modified block SGS iteration (32) is $(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)(\widetilde{D}+U)^{-1}(\widetilde{D}-D-L)$. In order to prove modified block SGS iteration (32) is convergent, we only need to prove that $\rho((\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)(\widetilde{D}+U)^{-1}(\widetilde{D}-D-L))<1$. Let the following $\|\cdot\|$ is ∞-norm, $L_0=U_{N_y+1}=0$ and $y=(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)x$, so it holds that

$$\|(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)\|=\max_{\|x\|=1}\|y\|,$$

by making use of $y=(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)x$, we can obtain $(\widetilde{D}+L)y=(\widetilde{D}-D-U)x$, i.e.,

$$y=-\widetilde{D}^{-1}Ly+\widetilde{D}^{-1}(\widetilde{D}-D-U)x,$$

here

$$\widetilde{D}^{-1}L=\begin{pmatrix}0 & 0 & \cdots & 0\\ M_2^{-1}L_1 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & M_{N_y}^{-1}L_{N_y-1} & 0\end{pmatrix},\quad \widetilde{D}^{-1}(\widetilde{D}-D-U)=\begin{pmatrix}M_1^{-1}\Delta_1 & -M_1^{-1}U_2 & \cdots & 0\\ 0 & M_2^{-1}\Delta_2 & \ddots & \vdots\\ \vdots & \vdots & \ddots & -M_{N_y-1}^{-1}U_{N_y}\\ 0 & 0 & \cdots & M_{N_y}^{-1}\Delta_{N_y}\end{pmatrix},$$

thus there exists some $1\leqslant i\leqslant N_y$, s.t.,

$$\|y\|=\|-M_i^{-1}L_{i-1}y_{i-1}+M_i^{-1}\Delta_i x_i-M_i^{-1}U_{i+1}x_{i+1}\|\leqslant\|M_i^{-1}L_{i-1}\|\|y\|+\|M_i^{-1}\Delta_i\|\|x\|+\|M_i^{-1}U_{i+1}\|\|x\|,$$

hence

$$\|y\|\leqslant\frac{\|M_i^{-1}\Delta_i\|\|x\|+\|M_i^{-1}U_{i+1}\|\|x\|}{1-\|M_i^{-1}L_{i-1}\|},$$

thus

$$\max_{\|x\|=1}\|y\|\leqslant\max_{\|x\|=1}\frac{\|M_i^{-1}\Delta_i\|+\|M_i^{-1}U_{i+1}\|}{1-\|M_i^{-1}L_{i-1}\|},$$

because P is BSDD-II by rows, i.e.,

$$\|M_i^{-1}\Delta_i\|+\|M_i^{-1}U_{i+1}\|+\|M_i^{-1}L_{i-1}\|<1,$$

thus

$$\|(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)\|=\max_{\|x\|=1}\|y\|<1.$$

The proof for $\|(\widetilde{D}+U)^{-1}(\widetilde{D}-D-L)\|<1$ is similar. Thus

$$\rho((\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)(\widetilde{D}+U)^{-1}(\widetilde{D}-D-L))<\|(\widetilde{D}+L)^{-1}(\widetilde{D}-D-U)\|\cdot\|(\widetilde{D}+U)^{-1}(\widetilde{D}-D-L)\|<1.\qquad\square$$

5. Preconditioning the Krylov Subspace methods

We consider nonsymmetric system of linear equations as follows

$$Ax=b,\tag{34}$$

the basic requirements for M to be a good preconditioner of A are that: M needs to be nonsingular and is easy to invert. Furthermore, the condition number of $M^{-1}A$ should be definitely smaller than the one of A. After finding a good preconditioner M, we only to solve linear systems

$$M^{-1}Ax=M^{-1}b$$

in place of (34).

We use iteration method to solve (34), and consider the splitting of A

$$A=M-N,\tag{35}$$

where M is nonsingular, we construct the following iteration

$$Mx_k=Nx_{k-1}+b,\quad k=1,2,\ldots,\tag{36}$$

the iteration matrix is $M^{-1}N$. When the spectral radius of $M^{-1}N$, i.e., $\rho(M^{-1}N)<1$, the iteration (36) is convergent. We say that M can be supposed as a preconditioner of coefficient matrix A. In fact, $\rho(M^{-1}N)<1$ then $\lambda(M^{-1}N)<1$ and $M^{-1}N=I-M^{-1}A$, then we have $|\lambda(I-M^{-1}A)|=|1-\lambda(M^{-1}A)|<1$, this illustrates that the eigenvalues of $M^{-1}A$ are inside the unit circle whose

center is at (1,0). Here we only present two preconditioners for (1). The first preconditioner is the block SSOR iteration preconditioner.

The block SSOR iteration (27) can be written as:

$$(D+\omega U)D^{-1}(D+\omega L)x_{k+1} = (D+\omega U)D^{-1}((1-\omega)D-\omega U)\cdot(D+\omega U)^{-1}((1-\omega)D-\omega L)x_k + \omega(2-\omega)b,$$

corresponding to (36) and (35), we have

$$M = \frac{1}{\omega(2-\omega)}(D+\omega U)D^{-1}(D+\omega L),$$

then M can be viewed as a preconditioner for (1).

After preconditioning, we must solve the system of linear equations like "$Mz = r$" at each step of iteration, so we need to solve $2\times N_y$ sub-systems of linear equations with the coefficient matrices $D_{N_y},\ldots,D_1$ and $D_1,\ldots,D_{N_y}$, and we can implement block LU factorization.

Based on the AF-ADI scheme, we can construct the second preconditioner. In fact, AF-ADI method is to solve the system of linear equations

$$(I+\mathcal{L})(I+\mathcal{U})x = b$$

instead of the system of linear equations

$$(I+\mathcal{L}+\mathcal{U})x = b,$$

that is to say $(I+\mathcal{L})(I+\mathcal{U})$ is a good approximation to $(I+\mathcal{L}+\mathcal{U})\equiv\mathcal{A}$, then $(I+\mathcal{L})(I+\mathcal{U})$ can also be viewed as a preconditioner for (1) and the preconditioner is called AF-ADI preconditioner.

Since the coefficient matrix $\mathcal{A}$ of the linear systems (1) is nonsymmetric, we can implement GMRES and BiCGSTAB method. In order to accelerate the convergence rate, M based on the block SSOR iteration can be used as a preconditioner. Finally, we implement GMRES and BiCGSTAB method to solve the preconditioned linear systems $M^{-1}\mathcal{A}x = M^{-1}b$. These methods are usually called P-GMRES and P-BiCGSTAB methods (abbreviated for BSSOR-GMRES, BSSOR-BiCGSTAB respectively).

6. Numerical experiments

Two numerical examples in this paper are the two-dimensional steady plane Couette–Poisuile flow and modified cavity flow as presented in the Refs. [18,19]. They have the same governing equations as (4)–(6). The boundary conditions of plane Couette–Poisuile flow for the velocity components u, v and the pressure p are

$$\begin{aligned}
&\frac{\partial p}{\partial y}(x,1) = 0, \quad u(x,1) = 1.0, \quad v(x,1) = 0,\\
&p(0,y) = \text{Pinlet}, \quad \frac{\partial u}{\partial x}(0,y) = 0, \quad v(0,y) = 0,\\
&p(1,y) = 0, \quad \frac{\partial u}{\partial x}(1,y) = 0, \quad \frac{\partial v}{\partial x}(1,y) = 0,\\
&\frac{\partial p}{\partial y}(x,0) = 0, \quad u(x,0) = 0, \quad v(x,0) = 0,
\end{aligned}$$

where 'Pinlet' is set to 10 for the considered case. The boundary conditions of modified cavity flow for the velocity components u, v and the pressure p are of Dirichlet type, i.e., zero everywhere except that

$$\begin{aligned}
&u(x,1) = 16(x^4-2x^3+x^2), \quad p(1,y) = \frac{6.4y}{Re},\\
&p(x,1) = \frac{8}{Re}[24F(x)+2f'(x)g''(1)+f'''(x)g(1)] - 64[F_2(x)G_1(1)-g(1)g''(1)F_1(x)].
\end{aligned}$$

The computational domain is $\Omega = (0,1)\times(0,1)$. We consider the linear systems at any pseudo-time level. 'IT', 'CPU' and 'η_1' respectively denote iteration numbers, computation time (second) and the relative residual error of numerical solution obtained by using AF-ADI method. The relaxation factor ω in block SSOR iteration, modified block SSOR iteration and block SSOR iteration preconditioner is taken as arbitrarily 1.2.

We let $Re = 1.0$, $\beta = 100$ for plane Couette–Poisuile flow and $Re = 100$, $\beta = 100$ for modified cavity flow. By applying the artificial compressibility method and fifth-order upwind compact scheme on the equidistant grid with the step size $h = \Delta x = \Delta y = 1/N$, we obtain the system of linear equations of the form

$$\begin{cases} \mathcal{A}(Q^n)\cdot\Delta Q^n = S^n, \\ Q^{n+1} = Q^n + \Delta Q^n, n = 0,1,\ldots, \end{cases} \tag{37}$$

where the dimension of $\mathcal{A}(Q^n)$ is $3\times(N-1)\times(N-1)$, n is the pseudo-time level (the number of sub-iteration), see [18,19].

In each step of (37), we must solve the system of linear equations and [18,19] use AF-ADI method. Here we implement the iteration methods presented in this paper. The initial guess is chosen to be $\Delta Q^{n,(0)} = 0$. We consider the linear systems taking $n = 0$ and $n = 100$ as examples. When using AF-ADI method to solve the linear systems at $n = 0$ and $n = 100$, the relative residual error of solution $\Delta\widetilde{Q}^n$ is

$$\frac{\|S^n - \mathcal{A}(Q^n)\Delta\widetilde{Q}^n\|}{\|S^n\|} \equiv \eta_1.$$

In addition, the stopping criterions for the iterations of BSSOR, MBSSOR, GMRES, BSSOR-GMRES, BiCGSTAB, BSSOR-BiCGSTAB methods are all set to be

$$\frac{\|S^n - \mathcal{A}(Q^n)\Delta Q^{n,(k)}\|}{\|S^n\|} \leqslant \eta,$$

where n is a certain pseudo-time level, e.g., 0 or 100, k is the number of the iterations for solving the system of linear equations (37), η is a prescribed tolerance for controlling the accuracy of the iterations.

In Tables 1–3, all the iteration methods are compared with AF-ADI method for different N when $n = 0$ and $n = 100$. In Tables 4–6, we compare all the iteration methods for different N when $n = 0$ and $n = 100$. Tables 1, 2, 4 and 5 are for plane Couette–Poisuile flow, Tables 3 and 6 are for modified cavity flow.

In Tables 1–3, we list the numerical results corresponding to the tolerance $\eta = \eta_1$, i.e., the relative residual error of solution $\Delta\widetilde{Q}^n$ when using AF-ADI method to solve the linear systems with $n = 0$ and $n = 100$. From these three tables, we see that all the iteration methods are convergent, and to achieve the same relative residual error accuracy, all the iteration methods spend the same or less time than the AF-ADI method for different N.

In Tables 4–6, we list the numerical results corresponding to the tolerance $\eta = 10^{-4}$ when $n = 0$ and $n = 100$. From these three tables, we see that all the iteration methods are convergent, and to achieve the same relative residual error accuracy,

Table 1
Numerical results for plane Couette–Poisuile flow when $n = 0$, $\eta = \eta_1$.

N		20	40	80
AF-ADI	η_1	0.56	0.28	0.18
	CPU	0.47	3.58	69.91
BSSOR	IT	2	4	13
	CPU	0.5	3.25	63.16
MBSSOR	IT	2	6	17
	CPU	0.48	3.44	68.39
GMRES	IT	13	22	32
	CPU	0.47	3.02	60.99
BSSOR-GMRES	IT	2	3	4
	CPU	0.48	3.13	60.26
BiCGSTAB	IT	11	35	37
	CPU	0.47	2.97	57.80
BSSOR-BiCGSTAB	IT	1	2	3
	CPU	0.50	3.23	60.41

Table 2
Numerical results for plane Couette–Poisuile flow when $n = 100$, $\eta = \eta_1$.

N		20	40	80
AF-ADI	η_1	0.68	0.62	0.63
	CPU	0.63	3.61	65.24
BSSOR	IT	1	1	1
	CPU	0.48	3.16	56.27
MBSSOR	IT	1	1	1
	CPU	0.5	3.14	56.59
GMRES	IT	2	2	4
	CPU	0.42	3.03	54.50
BSSOR-GMRES	IT	2	2	2
	CPU	0.49	3.28	56.11
BiCGSTAB	IT	1	1	2
	CPU	0.41	3.02	54.84
BSSOR-BiCGSTAB	IT	1	1	1
	CPU	0.47	3.14	55.36

Table 3
Numerical results for modified cavity flow when $n = 100$, $\eta = \eta_1$.

N		20	40	80
AF-ADI	η_1	0.97	0.93	0.60
	CPU	0.56	3.66	64.92
BSSOR	IT	1	3	5
	CPU	0.47	3.25	64.37
MBSSOR	IT	1	1	2
	CPU	0.44	3.17	58.30
MBSSOR	IT	2	2	9
	CPU	0.39	2.94	56.88
BSSOR-GMRES	IT	2	2	5
	CPU	0.45	3.23	58.31
BiCGSTAB	IT	1	1	21
	CPU	0.39	2.88	54.45
BSSOR-BiCGSTAB	IT	1	1	4
	CPU	0.44	3.17	60.40

Table 4
Numerical results for plane Couette–Poisuile flow when $n = 0$, $\eta = 10^{-4}$.

N		20	40	80
BSSOR	IT	16	36	81
	CPU	0.58	4.80	80.88
MBSSOR	IT	21	49	108
	CPU	0.81	7.23	128.31
GMRES	IT	49	73	101
	CPU	0.59	3.84	64.14
BSSOR-GMRES	IT	7	11	16
	CPU	0.53	3.61	62.88
BiCGSTAB	IT	32	55	77
	CPU	0.47	3.13	58.89
BSSOR-BiCGSTAB	IT	4	7	10
	CPU	0.55	3.88	64.73

Table 5
Numerical results for plane Couette–Poisuile flow when $n = 100$, $\eta = 10^{-4}$.

N		20	40	80
BSSOR	IT	14	30	62
	CPU	0.56	4.39	74.77
MBSSOR	IT	17	38	80
	CPU	0.74	7.47	129.67
GMRES	IT	43	64	83
	CPU	0.58	3.69	62.00
BSSOR-GMRES	IT	6	9	13
	CPU	0.50	3.41	59.99
BiCGSTAB	IT	27	50	58
	CPU	0.44	3.13	58.28
BSSOR-BiCGSTAB	IT	4	5	8
	CPU	0.53	3.61	63.31

Table 6
Numerical results for modified cavity flow when $n = 100$, $\eta = 10^{-4}$.

N		20	40	80
BSSOR	IT	12	307	906
	CPU	0.64	177.45	335.64
MBSSOR	IT	10	15	20
	CPU	0.56	3.97	70.72
GMRES	IT	70	118	155
	CPU	0.75	5.47	71.33
BSSOR-GMRES	IT	11	20	29
	CPU	0.53	3.9	67.13
BiCGSTAB	IT	44	76	116
	CPU	0.44	3.38	58.72
BSSOR-BiCGSTAB	IT	6	11	20
	CPU	0.52	3.90	69.44

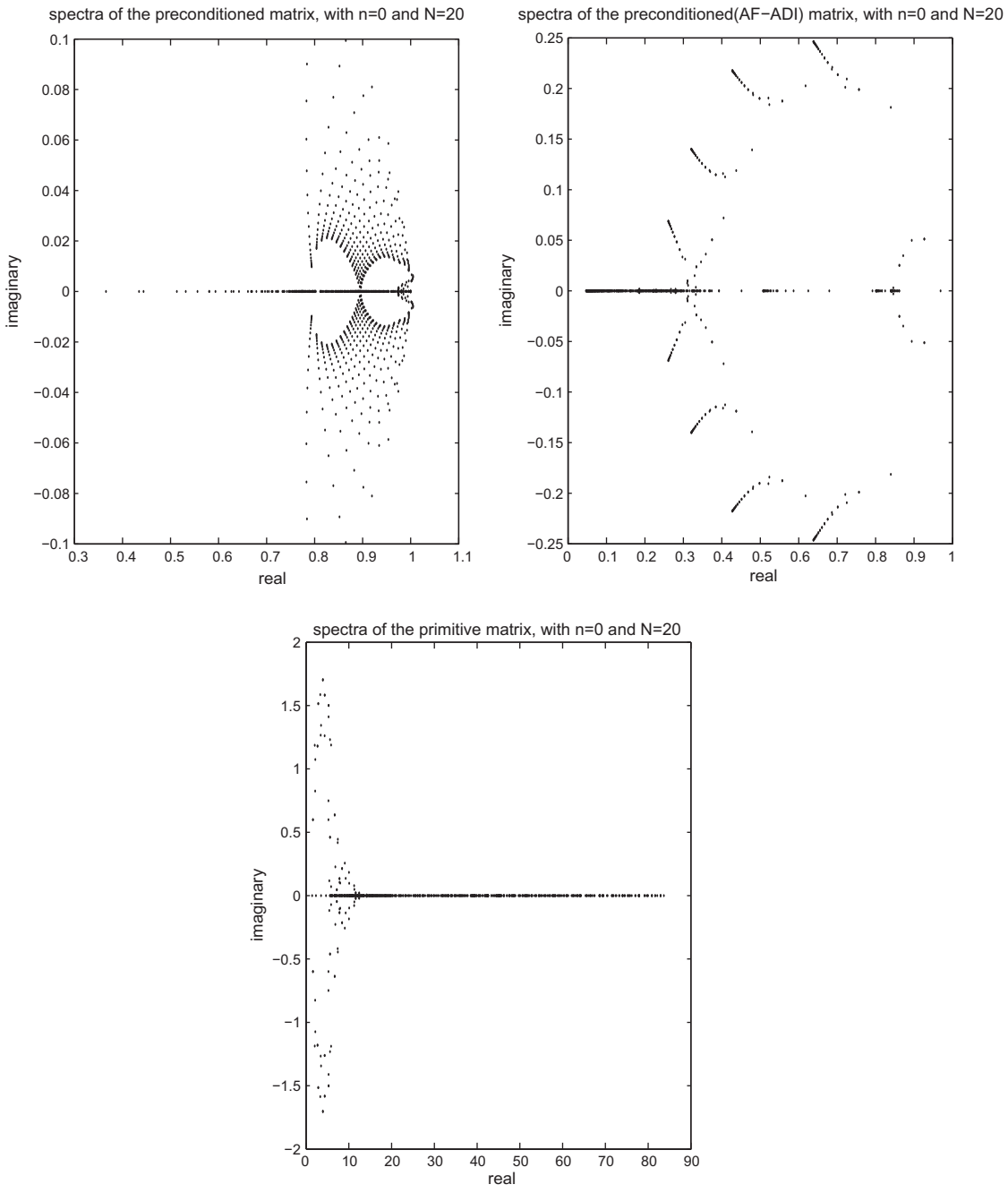

Fig. 1. Distribution of the eigenvalues of preconditioned matrix (BSSOR iteration preconditioner and AF-ADI preconditioner) and no-preconditioned matrix with $n = 0$ and $N = 20$.

the number of iterations of preconditioned methods are much less than the iteration methods. Moreover, with the grid refinement, the number of iterations of iteration methods increase very rapidly. On the contrary the number of iterations of preconditioned methods increase very slowly. Moreover, the modified block SSOR iteration method is superior to the block SSOR iteration method.

x-axis and y-axis in Figs. 1 and 2 denote the real and the imaginary of eigenvalues of matrix respectively. The lower subfigure of Fig. 1 is distribution of the eigenvalues of the no-preconditioned matrix $\mathcal{A}(Q^n)$ and the upper subfigures of Fig. 1 from left to right are distribution of the eigenvalues of the preconditioned matrix $M^{-1}\mathcal{A}(Q^n)$, where M are the block SSOR iteration preconditioner and the AF-ADI preconditioner when $n = 0$ with $N = 20$. Fig. 2 is the same as Fig. 1 except that

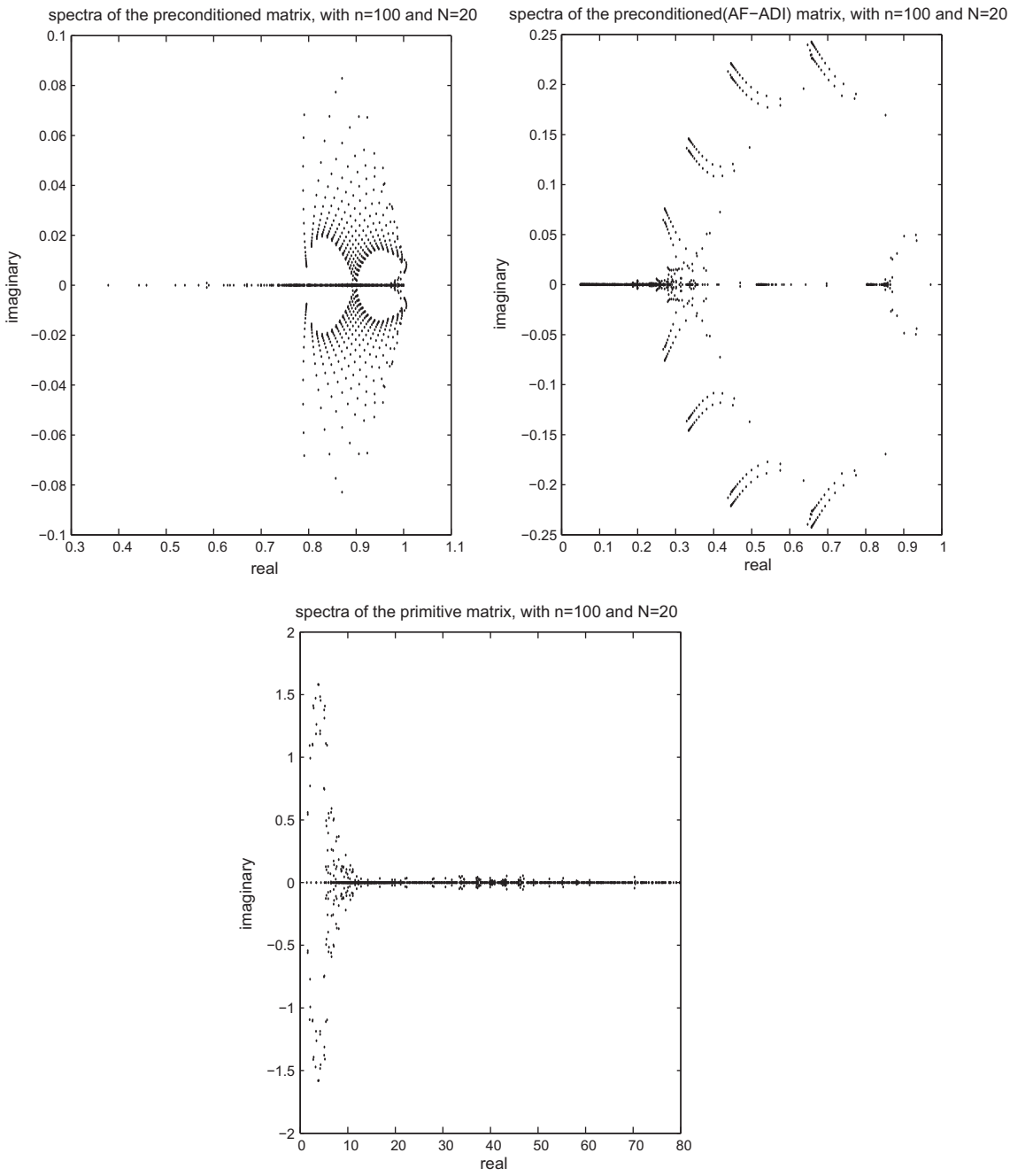

Fig. 2. Distribution of the eigenvalues of preconditioned matrix (BSSOR iteration preconditioner and AF-ADI preconditioner) and no-preconditioned matrix with $n = 100$ and $N = 20$.

$n = 100$. We conclude that the eigenvalues of the preconditioned matrix is more concentrated distribution than the ones of itself. When Krylov subspace methods are implemented to the preconditioned systems, they converge faster.

Fig. 3 shows curves of iteration number and relative residual error for block SSOR method, modified block SSOR method, GMRES method and BiCGSTAB method. We see that all the iteration methods are convergent and the number of iterations of modified block SSOR are less than that of other iteration methods.

Fig. 4 shows curves of CPU and relative residual error for block SSOR method, modified block SSOR method, GMRES method and BiCGSTAB method. We see that modified block SSOR is much more effective than block SSOR and GMRES in actual computations.

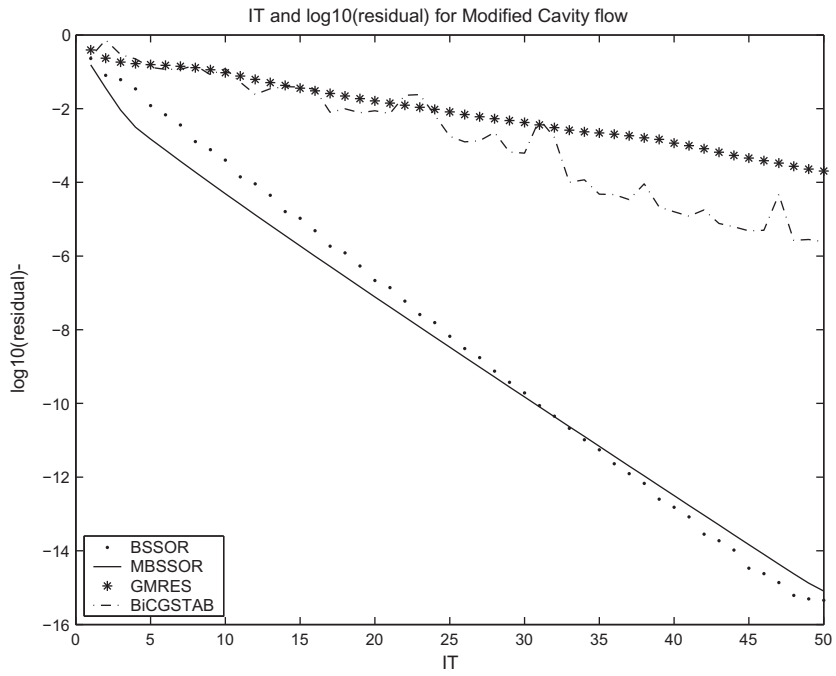

Fig. 3. The curves of relative residual error versus iteration number when $N = 20$ for different iteration methods.

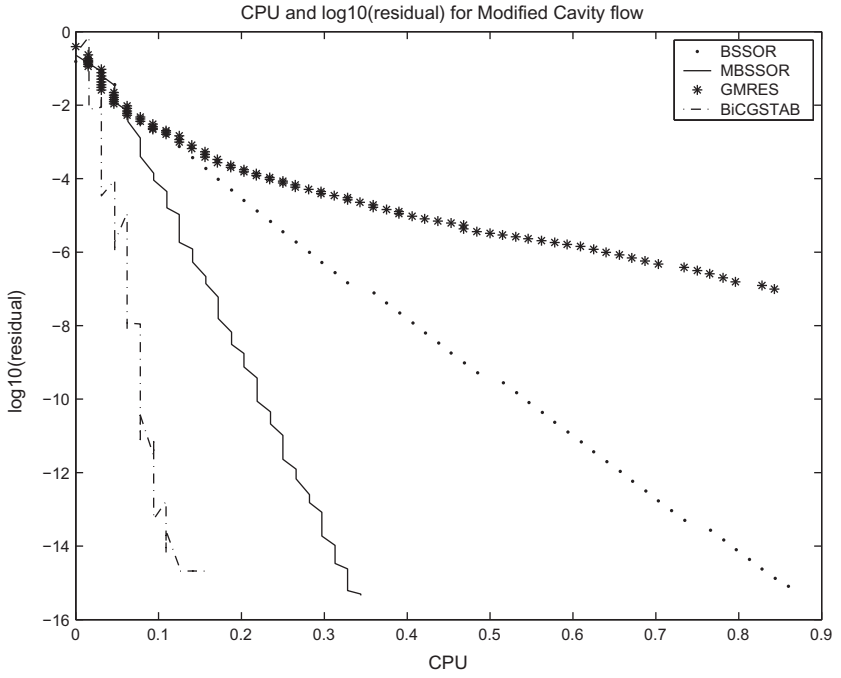

Fig. 4. The curves of relative residual error versus CPU (seconds) when $N = 20$ for different iteration methods.

7. Concluding remarks

For large sparse system of linear equations (1) which come from numerical solutions of two-dimensional steady incompressible viscous flow problems, based on the special structure of the coefficient matrix, we have established block SSOR and modified block SSOR iteration methods. Furthermore, we show that the modified block SSOR iteration in this paper are for

nonsymmetric block diagonally dominant linear systems, but the existing ones proposed by Bai [1,2] are only for symmetric positive definite linear systems. Finally, we have presented two preconditioners, i.e., the block SSOR iteration preconditioner and AF-ADI preconditioner. Both theoretical analysis and numerical experiments have shown that under certain conditions these new methods and two preconditioners are feasible, robust, and efficient solvers. Moreover, they are superior to the AF-ADI method.

One limitation of these iteration methods is lack of more in-depth theoretical convergence analysis. To design feasible and effective iteration methods for solving large scale systems from fluid flows with the special structure and study the theoretical and numerical properties of the iterative methods will be topics of theoretical importance and practical value. Another limitation is that it is very difficulty to determine whether the coefficient matrix $\mathcal{A}$ has the property of block diagonally dominant. But we can give the following theorem easily.

Theorem 7. *Let $2\Delta\tau/h \equiv \theta$, where $\Delta\tau$ is the pseudo-time step size, $h = \Delta x = \Delta y$ is the space step size. If $\theta < min\{1, 1/max\{\beta, |u|, |v|\}\}$, then the coefficient matrix $\mathcal{A}$ is strictly diagonally dominant by columns.*

Owing to the theorem, block Jacobi iteration, block Gauss–Seidel iteration, block SOR iteration and block SSOR iteration are all convergent. Noticed that the diagonal entries of $\mathcal{A}$ are positive, using Gerschgorin Circle theorem, we can conclude that $\mathcal{A}$ is positive definite. Thus block-triangular and skew-Hermitian splitting methods proposed by Bai [8] are convergent.

Acknowledgements

The authors are very much indebted to Professor Zhong-Zhi Bai for his useful discussions.

References

[1] Z.-Z. Bai, A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations, Adv. Comput. Math. 10 (1999) 169–186.
[2] Z.-Z. Bai, Modified block SSOR preconditioners for symmetric positive definite linear systems, Ann. Oper. Res. 103 (2001) 263–282.
[3] Z.-Z. Bai, Parallel matrix multisplitting block relaxation iteration methods, Math. Numer. Sinica 17 (1995) 238–252. In Chinese.
[4] Z.-Z. Bai, D.-R. Wang, Generalized asynchronous matrix multisplitting forward and backward relaxation methods, Math. Appl. 9 (1996) 121–126. In Chinese.
[5] Z.-Z. Bai, Asynchronous parallel matrix multisplitting block relaxation iteration methods, Numer. Math., A J. Chinese Univ. (Chinese Ser.) 19 (1997) 28–39. In Chinese.
[6] Z.-Z. Bai, V. Migallón, J. Penadés, D.B. Szyld, Block and asynchronous two-stage methods for mildly nonlinear systems, Numer. Math. 82 (1999) 1–20.
[7] Z.-Z. Bai, D.J. Evans, R.C. Calinescu, A class of asynchronous multisplitting two-stage iterations for large sparse block systems of weakly nonlinear equations, J. Comput. Appl. Math. 110 (1999) 271–286.
[8] Z.-Z. Bai, G.H. Golub, L.-Z. Lu, J.-F. Yin, Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, SIAM J. Sci. Comput. 26 (2005) 844–863.
[9] R.M. Beam, R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, J. Comput. Phys. 22 (1976) 87–110.
[10] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[11] J.W. Demmel, N.J. Higham, R.S. Schreiber, Stability of block LU factorization, Numer. Linear Algebr. 2 (1995) 173–190.
[12] D.J. Evans, Z.-Z. Bai, Blockwise matrix multi-splitting multi-parameter block relaxation methods, Int. J. Comput. Math. 64 (1997) 103–118.
[13] D.G. Feingold, R.S. Varga, Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem, Pac. J. Math. 12 (1962) 1241–1250.
[14] P.M. Gresho, Incompressible fluid dynamics: some fundamental formulation issues, Ann. Rev. Fluid Mech. 23 (1991) 413–453.
[15] R.W. MacCormack, G.V. Candler, The solution of the Navier–Stokes equations using Gauss–Seidel line relaxation, Comput. Fluids 17 (1989) 135–150.
[16] P. Orland, Fluid Flow Phenomena: A Numeirical Toolkit, Kluwer Academic Publishers., 1999.
[17] T.H. Pulliam, D.S. Chaussee, A diagonal form of an implicit approximate-factorization algorithm, J. Comput. Phys. 39 (1981) 347–363.
[18] A. Shah, H. Guo, L. Yuan, A third-order upwind compact scheme on curvilinear meshes for the incompressible Navier–Stokes equations, Commun. Comput. Phys. 5 (2009) 712–729.
[19] A. Shah, L. Yuan, Flux-difference splitting-based upwind compact schemes for the incompressible Navier–Stokes equations, Int. J. Numer. Meth. Fluids 61 (2009) 552–568.
[20] A. Shah, L. Yuan, A. Khan, Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier–Stokes equations, Appl. Math. Comput. 215 (2010) 3201–3213.
[21] S. Yoon, A. Jameson, Lower-upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations, AIAA J. 26 (1988) 1025–1026.
[22] C.-Y. Zhang, Y.-T. Li, F. Chen, On Schur complement of block diagonally dominant matrices, Linear Algebra Appl. 414 (2006) 533–546.