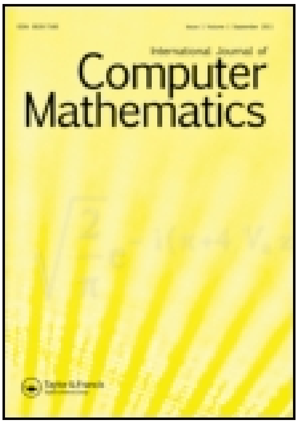

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

Modified alternating direction-implicit iteration method for linear systems from the incompressible Navier-Stokes equations

Yu-Hong Ran [a] & Li Yuan [a]

[a] 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, China

Available online: 19 Oct 2011

International Journal of Computer Mathematics
Vol. 88, No. 17, November 2011, 3762–3779

Taylor & Francis
Taylor & Francis Group

Modified alternating direction-implicit iteration method for linear systems from the incompressible Navier–Stokes equations

Yu-Hong Ran* and 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, China

(*Received 26 November 2010; revised version received 2 June 2011; accepted 4 September 2011*)

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, Ran and Yuan [*On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems*, Appl. Math. Comput. 217 (2010), pp. 3050–3068] presented the block symmetric successive over-relaxation (BSSOR) and the modified BSSOR iteration methods based on the special structures of the coefficient matrices. In this study, we present the modified alternating direction-implicit (MADI) iteration method for solving the linear systems. Under suitable conditions, we establish convergence theorems for the MADI iteration method. In addition, the optimal parameter involved in the MADI iteration method is estimated in detail. Numerical experiments show that the MADI iteration method is a feasible and effective iterative solver.

Keywords: incompressible Navier–Stokes equations; modified ADI iteration method; diagonally dominant; optimal parameter; convergence

2010 AMS Subject Classifications: 65F10; 65F50

1. Introduction

The incompressible Navier–Stokes equations are the mathematical basis for a wide spectrum of fluid flow problems. A difficulty in solving the incompressible Navier–Stokes equations numerically is the lack of a time-derivative term in continuity equation, which limits the straightforward use of time-marching numerical methods. Most numerical methods for these equations require solving pressure or pressure-correction Poisson equation, which serves to satisfy the continuity equation. A representative of these methods is the projection method [13], which is very efficient for solving unsteady problems. However, for steady problems, the artificial compressibility (AC) method [12] is a cost-effective method, which avoids solution of the pressure Poisson equation. Many numerical methods for solving incompressible flows have been developed [14,16]. A simple and accurate discretization method for the numerical solution of two-dimensional incompressible

*Corresponding author. Email: ranyh@lsec.cc.ac.cn

ISSN 0020-7160 print/ISSN 1029-0265 online

http://dx.doi.org/10.1080/00207160.2011.622375
http://www.tandfonline.com

viscous flow problems, which uses familiar third-order and fifth-order upwind compact finite-difference schemes in conjunction with the well-known AC method, has been developed [21–23]. After discretization of the incompressible Navier–Stokes equations, we need to solve a large sparse system of linear equations. Traditionally, the approximate factorization and alternating direction-implicit (AF–ADI) method [11,19], or the lower-upper symmetric-Gauss–Seidel method [24], or the line relaxation method [15] has been employed to solve the system of linear equations, but seldom direct methods have been used due to heavy costs. These traditional methods are very simple, but they have some drawbacks:

(a) The AF–ADI method and the line relaxation method can only be applied to a structured grid.
(b) They all bring some type of errors, usually factorization error, which is related to the time step size and AC factor, and this may affect the global convergent rate.

In [21–23], the AF–ADI method has been used. In order to get rid of the factorization error and make use of the special structure of the coefficient matrix of the linear systems, the block symmetric successive over-relaxation (BSSOR) and the modified block symmetric successive over-relaxation (MBSSOR) iteration methods [20] have been presented to solve the system of linear equations, which are based on [3,4], where a class of MBSSOR preconditioners for solving symmetric positive definite systems of linear equations was proposed and discussed [6]. Numerical experiments show that these two methods are better than the AF–ADI method. In this study, based on the classic ADI iteration method [17], we present another alternative iteration method, that is, the modified ADI (MADI) iteration method in which the choice of acceleration parameter is more flexible than in the BSSOR and the MBSSOR methods.

The classic ADI iteration method was introduced by Peaceman and Rachford for solving linear systems obtained by the finite-difference discretization of elliptic and parabolic problems. Let A be a matrix, and suppose that A is split as

$$A = A_1 + A_2,$$

where A_1 and A_2 are the discretization operators of the differential operators working in the x-direction and in the y-direction, respectively. Then, the classic ADI iteration method for solving the linear system $Ax = b$ is as follows:

$$(\alpha I + A_1)x^{(k+1/2)} = (\alpha I - A_2)x^{(k)} + b,$$
$$(\alpha I + A_2)x^{(k+1)} = (\alpha I - A_1)x^{(k+1/2)} + b.$$

It has been proved that the ADI iteration method is convergent if A_1 and A_2 are positive definite and the parameter $\alpha > 0$. Moreover, it is easy to determine the optimal parameter α when A_1 and A_2 are both Hermitian positive definite and commutative [1]. Motivated by the classic ADI iteration method, we present the MADI iteration method to solve the large sparse systems of linear equations arising from the numerical solutions of the incompressible Navier–Stokes equations.

This paper is organized as follows. In Section 2, we brief the numerical method for incompressible viscous flow problems presented in [21,22]. Furthermore, we give the specific form of the coefficient matrix of the linear system arising from the discretization of upwind finite-difference scheme. In Section 3, the MADI iteration method is presented. In Section 4, properties of the splitting matrices are presented. In Section 5, the convergence analysis of the MADI iteration method is presented. In Section 6, we estimate the optimal acceleration parameter of the MADI iteration method for a positive definite case. In Section 7, numerical experiments that show that the MADI method is a feasible and effective iterative solver are presented. In Section 8, we give some concluding remarks about the MADI iteration method.

2. Discretization of the governing equations

The governing two-dimensional steady incompressible Navier–Stokes equations in Cartesian coordinates (x, y) in a dimensionless form are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{1}$$

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

$$\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \tag{3}$$

where u and 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

$$\frac{\partial p}{\partial \tau} + \beta\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0, \tag{4}$$

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

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

where τ is the pseudo-time and β is the AC parameter. Equations (4)–(6) can also be written as

$$\frac{\partial Q}{\partial \tau} + \frac{\partial(\bar{E} - \bar{E}_v)}{\partial x} + \frac{\partial(\bar{F} - \bar{F}_v)}{\partial y} = 0, \tag{7}$$

where $Q = [p, u, v]^{\mathrm{T}}$ is the solution variable vector, and $\bar{E}, \bar{F}$ and $\bar{E}_v, \bar{F}_v$ are the inviscid and viscous flux vectors, respectively, that is,

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

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

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

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

$$A_v = \frac{\partial \bar{E}_v}{\partial Q} = \frac{1}{\mathrm{Re}} I_m \frac{\partial}{\partial x}, \quad B_v = \frac{\partial \bar{F}_v}{\partial Q} = \frac{1}{\mathrm{Re}} I_m \frac{\partial}{\partial y}, \quad \text{with} \quad I_m = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

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

$$\frac{\Delta Q^n}{\Delta \tau} = -\left[\frac{\partial(\bar{E} - \bar{E}_v)}{\partial x} + \frac{\partial(\bar{F} - \bar{F}_v)}{\partial y}\right]^{n+1}, \tag{8}$$

where $\Delta Q^n = Q^{n+1} - Q^n$, n is the pseudo-time level and $\Delta\tau$ is the pseudo-time step size. We linearize the implicit part of Equation (8) by using Taylor's expansion and obtain

$$\left[\Delta Q + \Delta\tau\left(\frac{\partial(A - A_\nu)\Delta Q}{\partial x} + \frac{\partial(B - B_\nu)\Delta Q}{\partial y}\right)\right]^n = -\Delta\tau\left[\frac{\partial(\bar{E} - \bar{E}_\nu)}{\partial x} + \frac{\partial(\bar{F} - \bar{F}_\nu)}{\partial y}\right]^n \equiv S^n. \tag{9}$$

On the right-hand side of Equation (9), the convective terms are discretized by the third-order or the fifth-order upwind compact scheme and the viscous terms by the fourth-order or the sixth-order central compact scheme. On the left-hand side of Equation (9), the convective terms are discretized by the first-order upwind difference and the viscous terms by the traditional second-order central difference. Thus, one obtains the following form:

$$\left[I + \Delta\tau\left(\delta_x^- A^+ + \delta_x^+ A^- - \frac{1}{\mathrm{Re}} I_m \delta_x^2\right) + \Delta\tau\left(\delta_y^- B^+ + \delta_y^+ B^- - \frac{1}{\mathrm{Re}} I_m \delta_y^2\right)\right]^n \Delta Q^n = S^n. \tag{10}$$

On the left-hand side of Equation (10), A^+ and A^- are constructed such that eigenvalues of '+' matrices are non-negative and those of '−' matrices are non-positive:

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

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 $\kappa \geq 1.3$ for the fifth-order upwind compact scheme.

We discretize Equation (10) at point (i, j) and suppose that the computational grid has $(N_x + 2) \times (N_y + 2)$ grid points, and Δx and Δy are the step sizes in the x-direction and in the y-direction, respectively. Obviously, the boundary condition is of Dirichlet type, that is,

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

Then,

$$\begin{aligned}\Bigg[\Delta Q_{ij} + \Delta\tau_{ij}\Bigg(&\frac{A_{ij}^+\Delta Q_{ij} - A_{i-1,j}^+\Delta Q_{i-1,j}}{\Delta x} + \frac{A_{i+1,j}^-\Delta Q_{i+1,j} - A_{ij}^-\Delta Q_{ij}}{\Delta x} \\ &- I_m\frac{\Delta Q_{i+1,j} - 2\Delta Q_{ij} + \Delta Q_{i-1,j}}{\mathrm{Re}\Delta x^2}\Bigg) + \Delta\tau_{ij}\Bigg(\frac{B_{ij}^+\Delta Q_{ij} - B_{i,j-1}^+\Delta Q_{i,j-1}}{\Delta y} \\ &+ \frac{B_{i,j+1}^-\Delta Q_{i,j+1} - B_{ij}^-\Delta Q_{ij}}{\Delta y} - I_m\frac{\Delta Q_{i,j+1} - 2\Delta Q_{ij} + \Delta Q_{i,j-1}}{\mathrm{Re}\Delta y^2}\Bigg)\Bigg]^n = S_{ij}^n,\end{aligned} \tag{11}$$

where

$$\Delta\tau_{ij} = \frac{c \cdot \Delta x \Delta y}{\Delta y(|u_{ij}| + \sqrt{u_{ij}^2 + \beta}) + \Delta x(|v_{ij}| + \sqrt{v_{ij}^2 + \beta})},$$

c is the Courant number. Let

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

and omit the subscript (space grid points) 'i,j' and superscript (the pseudo-time level) 'n' for simplicity. If we let

$$\theta_1 = \frac{\Delta\tau}{2\Delta x}, \quad r_1 = \frac{\Delta\tau}{\mathrm{Re}\Delta x^2}, \quad \theta_2 = \frac{\Delta\tau}{2\Delta y}, \quad r_2 = \frac{\Delta\tau}{\mathrm{Re}\Delta y^2},$$

then it follows that

$$E' = \begin{pmatrix} 2\theta_1\rho(A) & 0 & 0 \\ 0 & 2\theta_1\rho(A) + 2r_1 & 0 \\ 0 & 0 & 2\theta_1\rho(A) + 2r_1 \end{pmatrix},$$

$$E'' = \begin{pmatrix} 2\theta_2\rho(B) & 0 & 0 \\ 0 & 2\theta_2\rho(B) + 2r_2 & 0 \\ 0 & 0 & 2\theta_2\rho(B) + 2r_2 \end{pmatrix},$$

$$C = \begin{pmatrix} -\theta_1\rho(A) & \theta_1\beta & 0 \\ \theta_1 & \theta_1(2u - \rho(A)) - r_1 & 0 \\ 0 & \theta_1 v & \theta_1(u - \rho(A)) - r_1 \end{pmatrix},$$

$$H = \begin{pmatrix} -\theta_2\rho(B) & 0 & \theta_2\beta \\ 0 & \theta_2(v - \rho(B)) - r_2 & \theta_2 u \\ \theta_2 & 0 & \theta_2(2v - \rho(B)) - r_2 \end{pmatrix},$$

$$F = -\begin{pmatrix} \theta_1\rho(A) & \theta_1\beta & 0 \\ \theta_1 & \theta_1(2u + \rho(A)) + r_1 & 0 \\ 0 & \theta_1 v & \theta_1(u + \rho(A)) + r_1 \end{pmatrix},$$

$$G = -\begin{pmatrix} \theta_2\rho(B) & 0 & \theta_2\beta \\ 0 & \theta_2(v + \rho(B)) + r_2 & \theta_2 u \\ \theta_2 & 0 & \theta_2(2v + \rho(B)) + r_2 \end{pmatrix}.$$

From Equation (11) with the Dirichlet boundary condition, we get the block system of linear equations as follows:

$$\mathcal{A}\Delta Q = S, \tag{12}$$

where

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

$$D_j = \begin{pmatrix} E_{1j} & C_{2j} & \cdots & 0 \\ F_{1j} & E_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{N_xj} \\ 0 & \cdots & F_{N_x-1,j} & E_{N_xj} \end{pmatrix}, \quad j = 1, \ldots, N_y, \tag{13}$$

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

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

Our main task is to solve the linear equations (12) to get ΔQ. If $\mathcal{A}$ is positive definite, we can use the Hermitian and skew-Hermitian splitting (HSS) method [8] and the block triangular and skew-Hermitian splitting (BTSS) method [9] to solve the system of linear equations. In Section 3, we will establish the MADI iteration method for solving the large-scale systems of linear equations.

3. MADI iteration method

Owing to Equation (9), we give the MADI iteration method in differential form

$$\begin{aligned} \left(\alpha I + I + \Delta\tau \frac{\partial (A - A_v)}{\partial x}\right) \Delta Q^{(k+1/2)} &= \left(\alpha I - \Delta\tau \frac{\partial (B - B_v)}{\partial y}\right) \Delta Q^{(k)} + S, \\ \left(\alpha I + I + \Delta\tau \frac{\partial (B - B_v)}{\partial y}\right) \Delta Q^{(k+1)} &= \left(\alpha I - \Delta\tau \frac{\partial (A - A_v)}{\partial x}\right) \Delta Q^{(k+1/2)} + S. \end{aligned} \tag{14}$$

We consider the splitting of $\mathcal{A}$,

$$\mathcal{A} = I + T_x + T_y, \tag{15}$$

where

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

$$D'_j = \begin{pmatrix} E'_{1j} & C_{2j} & \cdots & 0 \\ F_{1j} & E'_{2j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{N_xj} \\ 0 & \cdots & F_{N_x-1,j} & E'_{N_xj} \end{pmatrix}, \quad D''_j = \begin{pmatrix} E''_{1j} & 0 & \cdots & 0 \\ 0 & E''_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E''_{N_xj} \end{pmatrix}, \quad j = 1, \ldots, N_y, \tag{16}$$

that is, T_x and T_y are strictly the x-differential operator and the y-differential operator, respectively. In particular, T_x and T_y are sparse and possess a special structure, and with suitable reordering, T_x and T_y are block tridiagonal. In matrix–vector form, the MADI iteration method (14) can be described as follows.

The MADI iteration method. Given an initial guess $\Delta Q^{(0)} \in \mathbb{C}^n$, compute $\Delta Q^{(k)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{\Delta Q^{(k)}\}$ satisfies the stopping criterion:

$$(\alpha I + I + T_x)\Delta Q^{(k+1/2)} = (\alpha I - T_y)\Delta Q^{(k)} + S,$$
$$(\alpha I + I + T_y)\Delta Q^{(k+1)} = (\alpha I - T_x)\Delta Q^{(k+1/2)} + S,$$

where α is a given constant that satisfies $\alpha > -\frac{1}{2}$ and I denotes the identity matrix.

We can observe that if $\mathcal{A} = T_x + T_y$, then the MADI iteration method is the classic ADI iteration method. The MADI iteration method can be written as

$$\begin{aligned} (\tilde{\alpha} I + T_1)\Delta Q^{(k+1/2)} &= (\tilde{\alpha} I - T_2)\Delta Q^{(k)} + S, \\ (\tilde{\alpha} I + T_2)\Delta Q^{(k+1)} &= (\tilde{\alpha} I - T_1)\Delta Q^{(k+1/2)} + S, \end{aligned} \tag{17}$$

where $\tilde{\alpha} = \alpha + \frac{1}{2},\ T_1 = \frac{1}{2}I + T_x,\ T_2 = \frac{1}{2}I + T_y$. We can observe that all the diagonal entries of both T_1 and T_2 are positive. We can consider T_1 as the x-direction operator and T_2 as the y-direction operator and observe that

$$\mathcal{A} = (\tfrac{1}{2}I + T_x) + (\tfrac{1}{2}I + T_y) \equiv T_1 + T_2. \tag{18}$$

Clearly, Equation (17) is the classic ADI iteration method where the factor α is replaced by $\alpha + \frac{1}{2}$, and we know that $\tilde{\alpha} > 0$, that is, $\alpha > -\frac{1}{2}$. The iteration (17) can be equivalently written as

$$\Delta Q^{(k+1)} = \Delta Q^{(k)} + P^{-1}(S - \mathcal{A}\Delta Q^{(k)}),$$

where

$$P = \frac{1}{2\tilde{\alpha}}(\tilde{\alpha} I + T_1)(\tilde{\alpha} I + T_2).$$

If the MADI iteration method is convergent, then we may regard the matrix P^{-1} as an approximate inverse of the matrix $\mathcal{A}$, that is, P can be considered as a preconditioner for linear systems (12) and therefore be named the MADI iteration preconditioner.

4. Properties of the splitting matrices

In this section, we give some properties of the splitting matrices T_1 and T_2.

THEOREM 4.1 *Let* $\theta_1 = \Delta\tau/(2\Delta x)$ *and* $\theta_2 = \Delta\tau/(2\Delta y)$, *where* $\Delta\tau$ *is the pseudo-time step size, and* Δx *and* Δy *are the space step sizes in the x-direction and in the y-direction, respectively. If* $\theta_1 < \min\{1, 1/\beta, 1/|v|\}/4$ *and* $\theta_2 < \min\{1, 1/\beta, 1/|u|\}/4$, *then* T_1 *and* T_2 *are both strictly diagonally dominant by rows. If* $\theta_1 < \min\{1, 1/(\beta + |v|)\}/4$ *and* $\theta_2 < \min\{1, 1/(\beta + |u|)\}/4$, *then* T_1 *and* T_2 *are both strictly diagonally dominant by columns and by rows.*

Proof We consider any block row whose elements are third-order matrix of the matrix T_1. Note that the block row that we have chosen has the most non-zero block elements. Suppose that the diagonal block of the block row is $\frac{1}{2}I + E'_{ij}$, then the non-diagonal blocks of the block row are $C_{i+1,j}$ and $F_{i-1,j}$,

$$C_{i+1,j} = \frac{\Delta\tau_{ij}}{2\Delta x}\begin{pmatrix} -\rho(A_{i+1,j}) & \beta & 0 \\ 1 & 2u_{i+1,j} - \rho(A_{i+1,j}) & 0 \\ 0 & v_{i+1,j} & u_{i+1,j} - \rho(A_{i+1,j}) \end{pmatrix} - \frac{\Delta\tau_{ij}}{\mathrm{Re}\Delta x^2} I_m,$$

$$F_{i-1,j} = -\frac{\Delta\tau_{ij}}{2\Delta x}\begin{pmatrix} \rho(A_{i-1,j}) & \beta & 0 \\ 1 & 2u_{i-1,j} + \rho(A_{i-1,j}) & 0 \\ 0 & v_{i-1,j} & u_{i-1,j} + \rho(A_{i-1,j}) \end{pmatrix} - \frac{\Delta\tau_{ij}}{\mathrm{Re}\Delta x^2} I_m,$$

$$\frac{1}{2}I + E'_{ij} = \frac{1}{2}I + \frac{\Delta\tau_{ij}\rho(A_{ij})}{\Delta x} I + \frac{2\Delta\tau_{ij}}{\mathrm{Re}\Delta x^2} I_m.$$

We expand each element of $C_{i+1,j}$ and $F_{i-1,j}$ at point (i, j) by using the Taylor expansion. We only compare the main parts of the elements of each row of the block row. By straightforward computations, we obtain that if

$$\frac{\Delta\tau_{ij}}{\Delta x} < \frac{1}{2\beta},$$

then the first row of the block row is strictly diagonally dominant. If

$$\frac{\Delta\tau_{ij}}{\Delta x} < \frac{1}{2},$$

then the second row of the block row is strictly diagonally dominant. If

$$\frac{\Delta\tau_{ij}}{\Delta x} < \frac{1}{2|v_{ij}|},$$

then the last row of the block row is strictly diagonally dominant. Thus, if

$$\theta_1 \equiv \frac{\Delta\tau}{2\Delta x} < \frac{\min\{1, 1/\beta, 1/|v|\}}{4},$$

then matrix T_1 is strictly diagonally dominant by rows. The proof for T_2 is similar.

We consider any block column whose elements are third-order matrix of the matrix T_1. Note that the block column that we have chosen has the most non-zero block elements. Suppose that the diagonal block of the block column is $\frac{1}{2}I + E'_{ij}$, then the non-diagonal blocks of the block column are C_{ij} and F_{ij},

$$C_{ij} = \frac{\Delta\tau_{i-1,j}}{2\Delta x}\begin{pmatrix} -\rho(A_{ij}) & \beta & 0 \\ 1 & 2u_{ij} - \rho(A_{ij}) & 0 \\ 0 & v_{ij} & u_{ij} - \rho(A_{ij}) \end{pmatrix} - \frac{\Delta\tau_{i-1,j}}{\mathrm{Re}\Delta x^2} I_m,$$

$$F_{ij} = -\frac{\Delta\tau_{i+1,j}}{2\Delta x}\begin{pmatrix} \rho(A_{ij}) & \beta & 0 \\ 1 & 2u_{ij} + \rho(A_{ij}) & 0 \\ 0 & v_{ij} & u_{ij} + \rho(A_{ij}) \end{pmatrix} - \frac{\Delta\tau_{i+1,j}}{\mathrm{Re}\Delta x^2} I_m,$$

$$\frac{1}{2}I + E'_{ij} = \frac{1}{2}I + \frac{\Delta\tau_{ij}\rho(A_{ij})}{\Delta x} I + \frac{2\Delta\tau_{ij}}{\mathrm{Re}\Delta x^2} I_m.$$

We expand $\Delta\tau_{i-1,j},\ \Delta\tau_{i+1,j},\ \Delta\tau_{i,j-1}$ and $\Delta\tau_{i,j+1}$ at point $(i,\ j)$ by using the Taylor expansion:

$$\Delta\tau_{i-1,j} = \Delta\tau_{ij} + O(\epsilon),$$
$$\Delta\tau_{i+1,j} = \Delta\tau_{ij} + O(\epsilon),$$
$$\Delta\tau_{i,j-1} = \Delta\tau_{ij} + O(\epsilon),$$
$$\Delta\tau_{i,j+1} = \Delta\tau_{ij} + O(\epsilon).$$

We only compare the main parts of the elements of each column of the block column. By straightforward computations, we obtain that if

$$\frac{\Delta\tau_{ij}}{\Delta x} < \frac{1}{2},$$

then the first column of the block column is strictly diagonally dominant. If

$$\frac{\Delta\tau_{ij}}{\Delta x}(\beta + |v_{ij}|) < \frac{1}{2},$$

then the second column of the block column is strictly diagonally dominant. The last column of the block column is strictly diagonally dominant unconditionally. Thus, if

$$\theta_1 \equiv \frac{\Delta\tau}{2\Delta x} < \frac{\min\{1, 1/(\beta + |v|)\}}{4},$$

then matrix T_1 is strictly diagonally dominant by columns. The proof for T_2 is similar. Because

$$\frac{\min\{1, 1/(\beta + |v|)\}}{4} < \frac{\min\{1, 1/\beta, 1/|v|\}}{4},$$
$$\frac{\min\{1, 1/(\beta + |u|)\}}{4} < \frac{\min\{1, 1/\beta, 1/|u|\}}{4}.$$

Thus, if $\theta_1 < \min\{1, 1/(\beta + |v|)\}/4$ and $\theta_2 < \min\{1, 1/(\beta + |u|)\}/4$, then T_1 and T_2 are both strictly diagonally dominant not only by rows but also by columns. ■

THEOREM 4.2 *If T_1 and T_2 are both strictly diagonally dominant not only by rows but also by columns, then T_1 and T_2 are both positive definite.*

Proof Because T_1 and T_2 are real matrices, we only need to prove that $T_1 + T'_1$ and $T_2 + T'_2$ are positive definite. Owing to Gerschgorin's theorem, we know that any eigenvalue λ of matrix

$T_1 + T_1'$ is located in one of the closed discs of the complex plane centred at and having the radius

$$\sum_{j\neq i} |(T_1)_{ij} + (T_1)_{ji}|.$$

In other words, $\forall\lambda \in \sigma(T_1 + T_1')$, $\exists i$ such that

$$|\lambda - 2(T_1)_{ii}| \leq \sum_{j\neq i} |(T_1)_{ij} + (T_1)_{ji}| \leq \sum_{j\neq i} |(T_1)_{ij}| + \sum_{j\neq i} |(T_1)_{ji}|.$$

Because T_1 is strictly diagonally dominant by rows and columns, and observe that all the diagonal entries of T_1 are positive,

$$(T_1)_{ii} > \sum_{j\neq i} |(T_1)_{ij}|$$

and

$$(T_1)_{ii} > \sum_{j\neq i} |(T_1)_{ji}|.$$

Thus,

$$\lambda > 2(T_1)_{ii} - \left(\sum_{j\neq i} |(T_1)_{ij}| + \sum_{j\neq i} |(T_1)_{ji}| \right) > 0.$$

Then, $T_1 + T_1'$ is positive definite. Therefore, T_1 is positive definite. The proof for T_2 is similar.

Suppose that T_1 and T_2 are not strictly diagonally dominant, but diagonally dominant by columns, that is, there exists at least some j, s.t.,

$$(T_1)_{jj} = \sum_{i\neq j} |(T_1)_{ij}|, \quad (T_2)_{jj} = \sum_{i\neq j} |(T_2)_{ij}|.$$

As the convective and the viscous terms of the Navier–Stokes equations are discretized by the first-order upwind difference and by the second-order traditional central difference, the discrete accuracy is $O(\Delta x + \Delta y)$. In order to ensure that T_1 and T_2 are strictly diagonally dominant by columns, we can add some small amounts of $c \cdot (\Delta x^2 + \Delta y^2)$ to the jth diagonal entry of matrix $\mathcal{A}$, where c is a suitable constant, and this does not affect the original accuracy. Furthermore, in order to ensure that T_1 and T_2 are both strictly diagonally dominant not only by columns but also by rows, therefore, T_1 and T_2 are positive definite, we can add some small amounts to the diagonal entries of $\mathcal{A}$ which do not affect the original accuracy. We will give the convergence theorems for the MADI iteration method in what follows. ■

5. The convergence of MADI iteration

In this section, we give the convergence theorem of the MADI iteration.

THEOREM 5.1 *Assume that T_1 and T_2 are positive definite and the parameter $\alpha > -\frac{1}{2}$, then the MADI iteration method is convergent.*

For the proof of Theorem 5.1, see [18].

THEOREM 5.2 *Assume that T_1 and T_2 are strictly diagonally dominant by rows and the parameter $\tilde{\alpha} > \max\{(T_1)_{ii}, (T_2)_{ii}\}$ $\forall i$, then the MADI iteration method is convergent.*

In order to give its proof, we first give an important lemma [7].

LEMMA 5.1 *Assume that $M = [\omega_{ij}]$ is strictly diagonally dominant by rows, then for any matrix $N = [\eta_{ij}]$, we have*

$$\| M^{-1}N \|_\infty \leq \max_{1\leq i\leq n} \frac{\sum_{j=1}^{n} |\eta_{ij}|}{|\omega_{ii}| - \sum_{j\neq i} |\omega_{ij}|}.$$

Because $\| (M^{\mathrm{T}})^{-1}N^{\mathrm{T}} \|_\infty = \| NM^{-1} \|_1$, we can give its equivalent corollary easily.

COROLLARY 5.1 *Assume that $M = [\omega_{ij}]$ is strictly diagonally dominant by columns, then for any matrix $N = [\eta_{ij}]$, we have*

$$\| NM^{-1} \|_1 \leq \max_{1\leq j\leq n} \frac{\sum_{i=1}^{n} |\eta_{ij}|}{|\omega_{jj}| - \sum_{i\neq j} |\omega_{ij}|}.$$

With Lemma 5.1, we are now ready to prove Theorem 5.2 as follows.

Proof The iteration matrix of the MADI iteration method is M. Because T_1 and T_2 are strictly diagonally dominant by rows and the diagonal entries of both T_1 and T_2 are positive and $\tilde{\alpha} > \max\{(T_1)_{ii}, (T_2)_{ii}\}$ $\forall i$, $\tilde{\alpha}I + T_1$ and $\tilde{\alpha}I + T_2$ are strictly diagonally dominant by rows. From Lemma 5.1, we have

$$\| (\tilde{\alpha}I + T_2)^{-1}(\tilde{\alpha}I - T_1) \|_\infty \leq \max_i \frac{|\tilde{\alpha} - (T_1)_{ii}| + \sum_{j\neq i} |(T_1)_{ij}|}{|\tilde{\alpha} + (T_2)_{ii}| - \sum_{j\neq i} |(T_2)_{ij}|} < 1,$$

$$\| (\tilde{\alpha}I + T_1)^{-1}(\tilde{\alpha}I - T_2) \|_\infty \leq \max_i \frac{|\tilde{\alpha} - (T_2)_{ii}| + \sum_{j\neq i} |(T_2)_{ij}|}{|\tilde{\alpha} + (T_1)_{ii}| - \sum_{j\neq i} |(T_1)_{ij}|} < 1.$$

Finally, we have

$$\begin{aligned}\rho(M) &= \rho((\tilde{\alpha}I + T_2)^{-1}(\tilde{\alpha}I - T_1)(\tilde{\alpha}I + T_1)^{-1}(\tilde{\alpha}I - T_2)) \\ &\leq \| (\tilde{\alpha}I + T_2)^{-1}(\tilde{\alpha}I - T_1) \|_\infty \| (\tilde{\alpha}I + T_1)^{-1}(\tilde{\alpha}I - T_2) \|_\infty \\ &< 1.\end{aligned}$$

■

We remark that when T_1 and T_2 are both strictly diagonally dominant by columns and the parameter $\tilde{\alpha} > \max\{(T_1)_{ii}, (T_2)_{ii}\}$ $\forall i$, from Corollary 5.1, the MADI iteration method is also convergent.

6. The optimal acceleration parameter

In this section, we investigate the optimal acceleration parameter of the MADI iteration method when T_1 and T_2 are both positive definite.

When T_1 and T_2 are positive definite, we first define the following positive constants:

$$\gamma_1 = \min_{x\neq 0} \frac{(x, (T_1 + T_1^{\mathrm{H}})x)}{(x,x)}, \quad \delta_1^2 = \max_{x\neq 0} \frac{(x, T_1^{\mathrm{H}} T_1 x)}{(x,x)},$$
$$\gamma_2 = \min_{x\neq 0} \frac{(x, (T_2 + T_2^{\mathrm{H}})x)}{(x,x)}, \quad \delta_2^2 = \max_{x\neq 0} \frac{(x, T_2^{\mathrm{H}} T_2 x)}{(x,x)}.$$

Because T_1 and T_2 are positive definite, $T_1 + T_1^{\mathrm{H}}$ and $T_2 + T_2^{\mathrm{H}}$ are Hermitian positive definite. By using the Min–Max theorem, where $\lambda_{\min}(A)$ denotes the smallest eigenvalue of the matrix A, we can write these constants as follows:

$$\gamma_1 = \lambda_{\min}(T_1 + T_1^{\mathrm{H}}), \quad \delta_1 = \| T_1 \|_2,$$
$$\gamma_2 = \lambda_{\min}(T_2 + T_2^{\mathrm{H}}), \quad \delta_2 = \| T_2 \|_2 .$$

It then holds that

$$\begin{aligned}
\rho(M)^2 &\leq \| (\tilde{\alpha} I - T_1)(\tilde{\alpha} I + T_1)^{-1} \|_2^2 \cdot \| (\tilde{\alpha} I - T_2)(\tilde{\alpha} I + T_2)^{-1} \|_2^2 \\
&= \max_{x\neq 0} \frac{\tilde{\alpha}^2(x,x) - \tilde{\alpha}(x, (T_1^{\mathrm{H}} + T_1)x) + (T_1 x, T_1 x)}{\tilde{\alpha}^2(x,x) + \tilde{\alpha}(x, (T_1^{\mathrm{H}} + T_1)x) + (T_1 x, T_1 x)} \\
&\quad \cdot \max_{x\neq 0} \frac{\tilde{\alpha}^2(x,x) - \tilde{\alpha}(x, (T_2^{\mathrm{H}} + T_2)x) + (T_2 x, T_2 x)}{\tilde{\alpha}^2(x,x) + \tilde{\alpha}(x, (T_2^{\mathrm{H}} + T_2)x) + (T_2 x, T_2 x)} \\
&\leq \frac{\tilde{\alpha}^2 - \tilde{\alpha}\gamma_1 + \delta_1^2}{\tilde{\alpha}^2 + \tilde{\alpha}\gamma_1 + \delta_1^2} \cdot \frac{\tilde{\alpha}^2 - \tilde{\alpha}\gamma_2 + \delta_2^2}{\tilde{\alpha}^2 + \tilde{\alpha}\gamma_2 + \delta_2^2} \\
&\leq \frac{[(\tilde{\alpha} - \gamma_1/2)^2 + \delta_1^2 - \gamma_1^2/4] \cdot [(\tilde{\alpha} - \gamma_2/2)^2 + \delta_2^2 - \gamma_2^2/4]}{\delta_1^2 \delta_2^2} \\
&\leq \frac{[(\tilde{\alpha} - \gamma_1/2)^2 + \delta_1^2 - \gamma_1^2/4 + (\tilde{\alpha} - \gamma_2/2)^2 + \delta_2^2 - \gamma_2^2/4]^2}{4\delta_1^2 \delta_2^2}.
\end{aligned}$$

Letting

$$\phi(\tilde{\alpha})^2 = \frac{[(\tilde{\alpha} - \gamma_1/2)^2 + \delta_1^2 - \gamma_1^2/4 + (\tilde{\alpha} - \gamma_2/2)^2 + \delta_2^2 \gamma_2^2/4]^2}{4\delta_1^2 \delta_2^2},$$

we now compute the optimal value of $\tilde{\alpha}$ that minimizes $\phi(\tilde{\alpha})$. Because

$$2\delta_1 \geq \gamma_1, \quad 2\delta_2 \geq \gamma_2,$$
$$\phi(\tilde{\alpha}) = \frac{[(\tilde{\alpha} - \gamma_1/2)^2 + \delta_1^2 - \gamma_1^2/4 + (\tilde{\alpha} - \gamma_2/2)^2 + \delta_2^2 - \gamma_2^2/4]}{2\delta_1 \delta_2}.$$

Obviously, $\phi(\tilde{\alpha})$ achieves the minimum when

$$\tilde{\alpha} = \frac{\gamma_1 + \gamma_2}{4}.$$

Thus, the optimal acceleration parameter involved in the MADI iteration method is given by

$$\alpha = \frac{\gamma_1 + \gamma_2}{4} - \frac{1}{2}.$$

It is practically important to know how to compute an approximation of the optimal acceleration parameter as accurately as possible for improving the convergence speed of the MADI iteration method, and it is a hard task that needs further in-depth study from the viewpoint of both theory and computations.

7. Numerical results

Two numerical examples in this study are the two-dimensional steady plane Couette–Poiseuille flow and the modified cavity flow presented in [21,22]. The boundary conditions of the plane Couette–Poiseuille flow are as follows:

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

where 'P_{inlet}' is set to 10 for the present example. The boundary conditions of the modified cavity flow for the velocity components u and v and the pressure p are all of Dirichlet type, that is, zero everywhere except that

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

where

$$
\begin{aligned}
f(x) &= x^4-2x^3+x^2,\\
g(y) &= y^4-y^2,\\
F(x) &= \int f(x)\,\mathrm{d}x,\\
F_1(x) &= f(x)f''(x)-[f'(x)]^2,\\
F_2(x) &= \int f(x)f'(x)\,\mathrm{d}x,\\
G_1(y) &= g(y)g'''(y)-g'(y)g''(y).
\end{aligned}
$$

The computational domain is $\Omega=(0,1)\times(0,1)$. We consider the linear system at any pseudo-time level. 'IT' and 'CPU', respectively, denote the iteration number and computation time (seconds). The relaxation factor ω and the parameter α in the BSSOR iteration or the MBSSOR iteration and the MADI methods are all taken as the optimal value.

Let Re $=1.0$ and $\beta=100$ for the plane Couette–Poiseuille flow and Re $=100$ and $\beta=100$ for the modified cavity flow. By applying the AC method and the 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

$$
\begin{aligned}
\mathcal{A}(Q^n)\cdot\Delta Q^n &= S^n,\\
Q^{n+1} &= Q^n+\Delta Q^n, \quad n=0,1,\ldots,
\end{aligned}
\tag{19}
$$

where the dimension of $\mathcal{A}(Q^n)$ is $3\times(N-1)\times(N-1)$ and n is the pseudo-time level.

In each pseudo-time step of Equation (19), we must solve the system of linear equations. Here, we only consider the linear systems taking $n = 100$. We compare the MADI iteration method with the BSSOR and MBBSSOR methods reported in [20]. The initial guess is chosen to be $\Delta Q^{n,(0)} = 0$. In addition, the stopping criteria for the iterations of the MADI, BSSOR and MBSSOR methods are all set to be

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

where n is a certain pseudo-time level, for example, $n = 100$, k is the number of iterations for solving the system of linear equations (19) at the pseudo-time level n and η is a prescribed tolerance for controlling the accuracy of the iterations.

In Tables 1 and 2, we list the experimentally optimal parameter α_{exp} in the MADI iteration method and optimal relaxation factors ω_{exp} in the BSOR, BSSOR and MBSSOR methods for the modified cavity flow and the plane Couette–Poiseuille flow, respectively.

In Tables 3 and 4, we compare all the iteration methods for different N. We list the numerical results corresponding to the tolerance $\eta = 10^{-6}$ for the modified cavity flow and the plane Couette–Poiseuille flow. From these tables, we can see that to achieve the same relative residual error accuracy, the actual computing time (CPU) of the MADI iteration method is less than that of the BSSOR and MBSSOR methods but much less than that of the BSOR method for both the modified cavity flow and the plane Couette–Poiseuille flow. Moreover, the number of iteration steps (IT) of the MADI iteration method is also less than that of the BSOR, BSSOR and MBSSOR methods for the modified cavity flow.

Table 1. The optimal values α_{exp} and ω_{exp} for the modified cavity flow.

N		10	20	40	80	160
MADI	α_{exp}	8.30	5.09	4.05	2.91	1.75
BSOR	ω_{exp}	1.06	0.97	0.68	0.37	0.15
BSSOR	ω_{exp}	1.22	1.17	1.15	1.16	1.12
MBSSOR	ω_{exp}	1.26	1.24	1.24	1.24	1.22

Table 2. The optimal values α_{exp} and ω_{exp} for the plane Couette–Poiseuille flow.

N		10	20	40	80
MADI	α_{exp}	3.80	4.80	5.42	6.54
BSOR	ω_{exp}	1.15	1.22	1.30	1.38
BSSOR	ω_{exp}	1.38	1.47	1.57	1.64
MBSSOR	ω_{exp}	1.45	1.54	1.67	1.75

Table 3. IT and CPU with $\eta = 10^{-6}$ for the modified cavity flow.

N		10	20	40	80	160
MADI	IT	11	15	20	28	36
	CPU	0.016	0.094	0.563	3.734	23.688
BSOR	IT	20	58	189	574	1777
	CPU	0.031	0.218	4.875	113.453	2684.907
BSSOR	IT	10	16	28	41	53
	CPU	0.047	0.157	1.578	14.281	167.015
MBSSOR	IT	10	15	26	39	48
	CPU	0.062	0.265	3.031	37.234	335.715

Table 4. IT and CPU with $\eta = 10^{-6}$ for the plane Couette–Poiseuille flow.

N		10	20	40	80
MADI	IT	13	19	32	49
	CPU	0.016	0.093	0.656	4.078
BSOR	IT	21	42	81	144
	CPU	0.016	0.156	1.891	20.109
BSSOR	IT	10	15	23	37
	CPU	0.016	0.203	1.000	13.078
MBSSOR	IT	11	16	27	46
	CPU	0.063	0.313	3.250	43.593

Figures 1 and 2 show the curves of CPU and relative residual error for the MADI iteration method, the BSSOR method and the MBSSOR method for the modified cavity flow and the plane Couette–Poiseuille flow, respectively. We can see that the CPU of the MADI method is much less than that of the other iteration methods, which straightforwardly implies that the MADI method is more efficient than the BSSOR method and the MBSSOR method.

In Figure 3, the x-axis and the y-axis denote the real and the imaginary eigenvalues of the non-preconditioned matrix and the preconditioned matrix for the plane Couette–Poiseuille flow, respectively. Figure 3(a) shows the distribution of the eigenvalues of the non-preconditioned matrix $\mathcal{A}$ and Figure 3(b) shows the distribution of the eigenvalues of the preconditioned matrix $P^{-1}\mathcal{A}$, where P is the MADI iteration preconditioner. We conclude that the eigenvalues of the preconditioned matrix are more concentrated in distribution than those of matrix $\mathcal{A}$. When Krylov subspace methods are implemented to the preconditioned linear systems, they converge faster.

Figure 4 shows the curves of the parameter α or ω and the spectra radius of the iteration matrices of the MADI iteration method (Figure 4(a)), the BSSOR method (Figure 4(b)) and the MBSSOR method (Figure 4(c)). It is clear that when $1.75 \le \omega \le 2$, the BSSOR and the MBSSOR methods

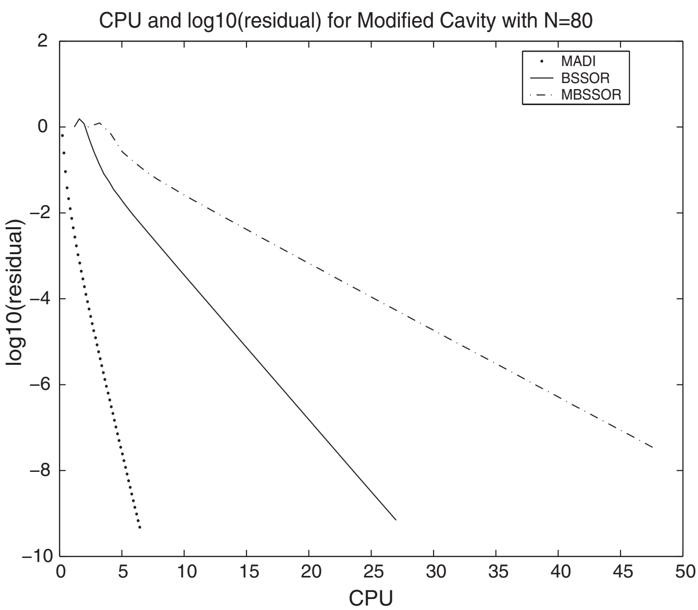

Figure 1. The curves of relative residual error versus CPU (seconds) for the modified cavity flow.

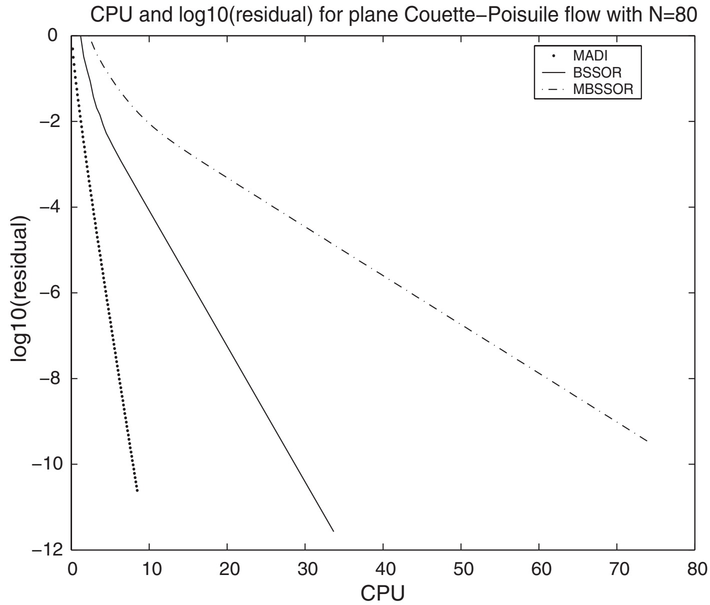

Figure 2. The curves of relative residual error versus CPU (seconds) for the plane Couette–Poiseuille flow.

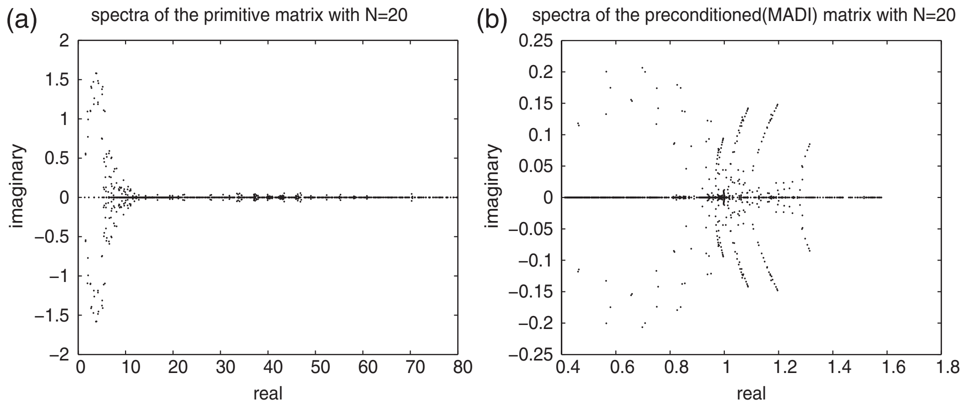

Figure 3. Distribution of eigenvalues of the non-preconditioned matrix and the preconditioned matrix for the plane Couette–Poiseuille flow.

are not convergent. This means that the parameter α in the MADI iteration method is more flexible than the relaxation factors ω in the BSSOR and the MBSSOR methods.

8. Concluding remarks

For the large-scale systems of linear equations which come from the implicit discretization of two-dimensional steady incompressible Navier–Stokes equations with the AC method and the upwind compact finite difference, we have established the MADI iteration method. Both theoretical analysis and numerical experiments have shown that this new method is a feasible, robust and efficient linear solver. Moreover, the acceleration parameter of the MADI iteration method is more flexible than the relaxation factors of the BSSOR and the MBSSOR methods.

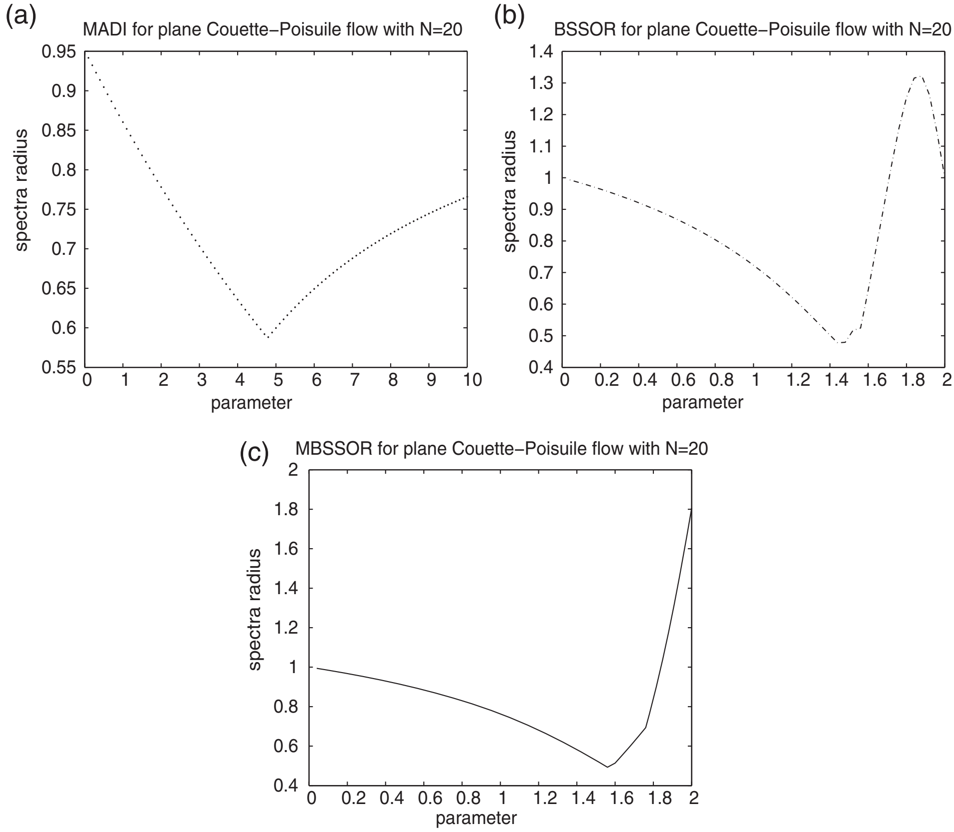

Figure 4. Curves of the spectra radius of iteration matrix $\rho(M)$ versus parameter α or ω.

One limitation of this iteration method is that it is difficult to compute an approximation of the optimal acceleration parameter which can improve the convergence speed of the MADI iteration method. Another limitation is that it is costly to compute the exact solutions to the linear sub-systems in its two-half iterates at each step. Hence, further study on practically computing the optimal parameter and inexactly solving the linear sub-systems in the MADI iteration method will be of practical value [2,5,6,10].

References

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.
[2] O. Axelsson, Z.-Z. Bai, and S.-X. Qiu, *A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part*, Numer. Algorithms 35 (2004), pp. 351–372.
[3] Z.-Z. Bai, *A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations*, Adv. Comput. Math. 10 (1999), pp. 169–186.
[4] Z.-Z. Bai, *Modified block SSOR preconditioners for symmetric positive definite linear systems*, Ann. Oper. Res. 103 (2001), pp. 263–282.
[5] Z.-Z. Bai and X.-B. Chi, *Asymptotically optimal successive overrelaxation methods for systems of linear equations*, J. Comput. Math. 21 (2003), pp. 603–612.
[6] Z.-Z. Bai and T.-Z. Huang, *On the convergence of the relaxation methods for positive definite linear systems*, J. Comput. Math. 16 (1998), pp. 527–538.
[7] Z.-Z. Bai and D.-R. Wang, *Generalized matrix multisplitting relaxation methods and their convergence*, Numer. Math. J. Chinese Univ. (English Ser.) 2 (1993), pp. 87–100.
[8] Z.-Z. Bai, G.H. Golub, and M.K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl. 24 (2003), pp. 603–626.

[9] Z.-Z. Bai, G.H. Golub, L.-Z. Lu, and J.-F. Yin, *Block triangular and skew-Hermitian splitting methods for positive-definite linear systems*, SIAM J. Sci. Comput. 26 (2005), pp. 844–863.

[10] Z.-Z. Bai, G.H. Golub, and M.K. Ng, *On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, Linear Algebra Appl. 428 (2008), pp. 413–440.

[11] R.M. Beam and R.F. Warming, *An implicit finite-difference algorithm for hyperbolic systems in conservation-law form*, J. Comput. Phys. 22 (1976), pp. 87–110.

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

[13] A.J. Chorin, *Numerical solution of the Navier–Stokes equations*, Math. Comput. 22 (1968), pp. 742–762.

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

[15] R.W. MacCormack and G.V. Candler, *The solution of the Navier–Stokes equations using Gauss–Seidel line relaxation*, Comput. Fluids 17 (1989), pp. 135–150.

[16] P. Orlandi, *Fluid Flow Phenomena: A Numerical Toolkit*, Kluwer Academic Publishers, Amsterdam, 1999.

[17] D.W. Peaceman and H.H. Rachford Jr, *The numerical solution of elliptic and parabolic differential equations*, SIAM J. Soc. Indust. Appl. Math. 3 (1955), pp. 28–41.

[18] C. Pearcy, *On convergence of alternating direction procedures*, Numer. Math. 4 (1962), pp. 172–176.

[19] T.H. Pulliam and D.S. Chaussee, *A diagonal form of an implicit approximate-factorization algorithm*, J. Comput. Phys. 39 (1981), pp. 347–363.

[20] Y.-H. Ran and L. Yuan, *On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems*, Appl. Math. Comput. 217 (2010), pp. 3050–3068.

[21] A. Shah and L. Yuan, *Flux-difference splitting-based upwind compact schemes for the incompressible Navier–Stokes equations*, Intern. J. Numer. Methods Fluids 61 (2009), pp. 552–568.

[22] A. Shah, H. Guo, and L. Yuan, *A third-order upwind compact scheme on curvilinear meshes for the incompressible Navier–Stokes equations*, Commun. Comput. Phys. 5 (2009), pp. 712–729.

[23] A. Shah, L. Yuan, and A. Khan, *Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier–Stokes equations*, Appl. Math. Comput. 215 (2010), pp. 3201–3213.

[24] S. Yoon and A. Jameson, *Lower-upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations*, AIAA J. 26 (1988), pp. 1025–1026.