Journal of Computational and Applied Mathematics 367 (2020) 112436

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Preconditioners for nonsymmetric indefinite linear systems☆

Zhen Chao [a], Dexuan Xie [a,*], Ahmed H. Sameh [b]

[a] Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA
[b] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 1 January 2019
Received in revised form 23 August 2019

MSC:
65F08
65F10

Keywords:
Indefinite linear systems
GMRES
Least squares problem
Preconditioning
Diffusion–convection equation

ABSTRACT

In this paper, we develop algorithms for solving nonsymmetric indefinite linear systems by considering the augmented linear systems resulting from a weighted linear least squares problem. Even though the augmented system is more ill-conditioned than the original linear system, one can construct preconditioned GMRES methods for solving these augmented systems capable of obtaining reasonable approximation of the solution in fewer iterations than the classical ILU preconditioned GMRES method for solving the original linear system. More specifically, we present two different preconditioners for these augmented systems, examine the spectral properties of these preconditioned augmented systems, and report numerical results to illustrate the effectiveness of these preconditioners.

1. Introduction

In this paper, we consider solving large sparse linear systems

$$Ax = b, \tag{1.1}$$

where A is a nonsymmetric indefinite nonsingular sparse matrix of order n. This kind of linear systems often arises in many important scientific and engineering applications such as those modeled by diffusion–convection equations [1–3]. Classical stationary iterative methods (such as Jacobi, Gauss–Seidel, and successive over-relaxation (SOR) methods [4]) may fail in solving such linear systems when the coefficient matrix A is highly indefinite and nonsymmetric [5,6]. When A is highly indefinite, a black-box preconditioners such as various forms of incomplete LU-factorizations (ILU) are often not effective in addition to not being suitable for implementation on parallel computing platforms [7–10] due to the dependencies in the forward and backward sweeps.

Instead of modifying "black-box" preconditioners such as ILU preconditioning schemes, we develop preconditioners through considering the weighted linear least squares problem:

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_W, \tag{1.2}$$

where $\|u\|_W = [u^T W u]^{1/2}$ with the weight W being a symmetric positive definite matrix. In this study, we simply choose

$$W = \gamma^{-2} I,$$

☆ For Special Issue "Numerical Methods for Complex Scientific and Engineering Problems". Editor: Zahari Zlatev.

* Corresponding author.
E-mail address: dxie@uwm.edu (D. Xie).

https://doi.org/10.1016/j.cam.2019.112436
0377-0427/

where I is the identity matrix, and γ is a positive scalar. When the residual $r = b - Ax$ is treated as an unknown vector, the least squares problem (1.2) yields a two-block linear system of the form

$$\mathcal{C}y = d, \tag{1.3}$$

where the matrix $\mathcal{C}$ and vectors y and d are given by

$$\mathcal{C} = \begin{pmatrix} I & B \\ -B^T & 0 \end{pmatrix}, \quad y = \begin{pmatrix} \tilde{r} \\ x \end{pmatrix}, \quad d = \begin{pmatrix} \tilde{b} \\ 0 \end{pmatrix}. \tag{1.4}$$

Here, $B = A/\gamma$, $\tilde{r} = r/\gamma$, and $\tilde{b} = b/\gamma$.

It is this special structure of our matrix $\mathcal{C}$ that allows us to choose a solver of (1.3) from the variety of numerical techniques used for saddle point problems [11,12].

Because of this special matrix structure, similar to Lemma 2.2 in [13], we find that the augmented matrix $\mathcal{C}$ has the following properties:

1. $\mathcal{C}$ is nonsingular.
2. $\mathcal{C}$ is positive real (i.e., the real parts of all the eigenvalues of $\mathcal{C}$ are positive).
3. There exists a positive integer t such that the singular values, $\{\sigma_k\}_{k=1}^n$, of A satisfy

$$\sigma_1 \geqslant \cdots \geqslant \sigma_t > \frac{\gamma}{2} > \sigma_{t+1} \geqslant \cdots \geqslant \sigma_n,$$

and the eigenvalues of $\mathcal{C}$ are given by

$$\frac{1}{2} \pm \mathbf{i}\frac{1}{2}\sqrt{\left(\frac{2\sigma_j}{\gamma}\right)^2 - 1}, \quad j = 1, 2, \ldots, t,$$

where $\mathbf{i} = \sqrt{-1}$, and

$$\frac{1}{2} \pm \frac{1}{2}\sqrt{1 - \left(\frac{2\sigma_k}{\gamma}\right)^2}, \quad k = t+1, t+2, \ldots, n.$$

4. If we choose $\gamma = \sigma_1$, then the spectral condition number of $\mathcal{C}$ can be estimated approximately as $1.618\kappa^2$. Here κ denotes the spectral condition number of A, which is defined by

$$\kappa = \sigma_1/\sigma_n. \tag{1.5}$$

These properties of the augmented matrix $\mathcal{C}$ can be particularly valuable in the development of preconditioners for the GMRES method [14] for solving the augmented linear system (1.3). We introduce two preconditioners, $\mathcal{M}_1$ and $\mathcal{M}_2$, and propose iterative schemes for solving linear systems involving them effectively and efficiently.

To verify the effectiveness of preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$, we construct linear systems (1.1) from finite element approximations of partial differential equation boundary value problems and by selecting matrices from the SuiteSparse Matrix Collection [15]. Numerical results show that GMRES for solving the augmented system preconditioned by $\mathcal{M}_1$ and $\mathcal{M}_2$ can be much more effective than GMRES for solving the original linear system $Ax = b$ preconditioned via ILU and its variants.

The remaining part of this paper is organized as follows. In Section 2, we introduce these two preconditioners and the SOR convergence analysis for solving linear systems involving $\mathcal{M}_1$ and $\mathcal{M}_2$. In Section 3, we present the test models and the results of the numerical experiments. Finally, conclusions are made in Section 4.

2. Two preconditioners for augmented systems

We present two preconditioners for the augmented system (1.3), and explore the spectral properties of the resulting preconditioned matrices.

Let the block matrix $\mathcal{C}$ be defined in (1.4) with $B = \gamma^{-1}A$. The first preconditioner, $\mathcal{M}_1$, which is called the HSS preconditioner, introduced by Benzi and Golub [12] motivated by the Hermitian skew-Hermitian splitting technique by Z.-Z. Bai, Golub, and Ng [16], is given by

$$\mathcal{M}_1 = (\alpha\mathcal{I} + \mathcal{H})(\alpha\mathcal{I} + \mathcal{S}), \tag{2.1}$$

where $\mathcal{H} = \frac{1}{2}(\mathcal{C} + \mathcal{C}^T)$, $\mathcal{S} = \frac{1}{2}(\mathcal{C} - \mathcal{C}^T)$, which are the symmetric and skew-symmetric parts of $\mathcal{C}$, respectively, $\mathcal{I}$ denotes an identity matrix of order $2n$, and α is a positive scalar. Preconditioner $\mathcal{M}_1$ has favorable spectral properties. The schemes that we use to solve linear systems involving $\mathcal{M}_1$ take advantage of the fact that B is nonsingular, and the (1, 1)th block

entry of $\mathcal{C}$ is the identity matrix. For example, as shown in [17], our preconditioned matrices $\mathcal{M}_1^{-1}\mathcal{C}$ and $\mathcal{C}\mathcal{M}_1^{-1}$ are positive real for all $\alpha > 0$, which assures that the preconditioned GMRES method is convergent.

We solve linear systems involving the preconditioner $\mathcal{M}_1$ of the form

$$\mathcal{M}_1 z = g, \tag{2.2}$$

by first solving the diagonal system $(\alpha\mathcal{I} + \mathcal{H})v = g$ for v, followed by using the SOR method to solve the shifted skew-symmetric system

$$(\alpha\mathcal{I} + \mathcal{S})z = v. \tag{2.3}$$

Each SOR iteration for solving the above system is of the form

$$\left(\frac{1}{\omega}\mathcal{D} - \mathcal{L}\right) z_{k+1} = \left[\left(\frac{1}{\omega} - 1\right)\mathcal{D} + \mathcal{U}\right] z_k + v, \tag{2.4}$$

where $\mathcal{D}$, $\mathcal{L}$ and $\mathcal{U}$ are defined by

$$\mathcal{D} = \begin{pmatrix} \alpha I & 0 \\ 0 & \alpha I \end{pmatrix}, \quad \mathcal{L} = \begin{pmatrix} 0 & 0 \\ B^T & 0 \end{pmatrix}, \quad \text{and} \quad \mathcal{U} = \begin{pmatrix} 0 & -B \\ 0 & 0 \end{pmatrix}.$$

Since

$$\left(\frac{1}{\omega}\mathcal{D} - \mathcal{L}\right)^{-1} = \begin{pmatrix} \frac{\omega}{\alpha} I & 0 \\ \left(\frac{\omega}{\alpha}\right)^2 B^T & \frac{\omega}{\alpha} I \end{pmatrix},$$

then in obtaining z_{k+1}, all we need is an efficient scheme for multiplying B and B^T by a vector. It has been shown in [17], see also Theorem 4.1 in [18], that if β is the spectral radius of the Jacobi iteration matrix for solving (2.3) then the SOR iteration (2.4) converges for all acceleration parameters ω in the interval $0 < \omega < 2/(1+\beta)$. Furthermore, the optimal acceleration parameter ω_b is given by,

$$\omega_b = \frac{2}{1 + \sqrt{1+\beta^2}}$$

and the spectral radius of the SOR iteration matrix is given by,

$$\rho = \frac{\beta^2}{(1 + \sqrt{1+\beta^2})^2}.$$

If γ is chosen as σ_1, the maximum singular value of A, then $\beta = 1/\alpha$, and

$$\alpha = \frac{1-\rho}{2\sqrt{\rho}}, \quad \omega_b = 1 - \rho. \tag{2.5}$$

Thus, if we choose ρ as a value between 0 and 1, we can determine α and ω_b.

On the other hand, for a given α, ρ can be found by $\rho = (\sqrt{1+\alpha^2} - \alpha)^2$. Note that the optimal relaxation parameter ω_b is less than 1, reducing to a case of under relaxation. Clearly, the smaller the value of ρ, the faster the convergence of the SOR method used to solve the system involving the preconditioner, i.e., one GMRES iteration step gets cheaper. But, from (2.5) it can be seen that a smaller value of ρ leads to a larger value of α. Numerical experiments indicate that large values of α result in larger numbers of the preconditioned GMRES iterations. Thus, there exists a trade off between the cost of solving systems involving the preconditioner and the number of outer GMRES iterations.

In practice, γ is chosen approximately by computing σ_1 numerically. We did so in our numerical experiments by using the limited memory block Krylov subspace optimization method [19]. Since this results in $\gamma = \mu\sigma_1$ with $0 < \mu < 1$, α should be chosen slightly larger than either $(1-\rho)/(2\sqrt{\rho})$ or $\omega_b/(2\sqrt{1-\omega_b})$.

The second preconditioner, $\mathcal{M}_2$, is given by

$$\mathcal{M}_2 = \begin{pmatrix} I & B \\ -B^T & \theta I \end{pmatrix}, \tag{2.6}$$

where θ is a positive scalar. This preconditioner has been proposed by Axelsson [20] for saddle point problems, see also [21]. Solving linear systems involving this preconditioner,

$$\mathcal{M}_2 z = g, \tag{2.7}$$

can be achieved again using the SOR method [17]. From the special structure of $\mathcal{M}_2$ we find that all the eigenvalues of the iteration matrix of the Jacobi method for solving the linear system (2.7) are also pure imaginary. Hence, the SOR method for solving (2.7) is convergent, and the SOR spectral radius is given by the following theorem.

Theorem 2.1. *Let ρ be the spectral radius of the SOR method for solving the linear system (2.7). If the relaxation parameter ω is set optimally as $1-\rho$, then ρ is given by the expression*

$$\rho = \left(\sqrt{\left(\frac{\gamma}{\sigma_1}\right)^2 \theta + 1} - \frac{\gamma}{\sigma_1}\sqrt{\theta} \right)^2, \tag{2.8}$$

where σ_1 is the maximum singular value of A.

Proof. The iteration matrix, $\mathcal{J}$, of the Jacobi method for solving (2.7) can be found as follows:

$$\mathcal{J} = \mathcal{I} - (\mathrm{diag}(\mathcal{M}_2))^{-1}\mathcal{M}_2 = \begin{pmatrix} 0 & -\frac{1}{\gamma}A \\ \frac{1}{\theta\gamma}A^T & 0 \end{pmatrix},$$

where $\mathrm{diag}(\mathcal{M}_2)$ denotes the diagonal matrix of $\mathcal{M}_2$. Hence, the eigenvalues of $\mathcal{J}$ are pure imaginary, and the spectral radius of $\mathcal{J}$ is given by $\sigma_1/(\gamma\sqrt{\theta})$. Theorem 4.1 in [18] yields the spectral radius expression (2.8). □

From the above theorem and proof it can be seen that the spectral radius of the Jacobi iteration matrix is given by $1/\sqrt{\theta}$ and the spectral radius ρ of the SOR method is given by $\rho = (\sqrt{1+\theta} - \sqrt{\theta})^2$ when $\gamma = \sigma_1$, and the relaxation parameter ω is set optimally as $1-\rho$. Hence, for any $\rho \in (0, 1)$, from the above expression we can select θ as follows:

$$\theta = \frac{(1-\rho)^2}{4\rho}. \tag{2.9}$$

Theorem 2.2. *The preconditioned matrix $\mathcal{M}_2^{-1}\mathcal{C}$ has n eigenvalues equal to 1, and the remaining n eigenvalues are given by*

$$\frac{\sigma_i^2}{\gamma^2\theta + \sigma_i^2}, \quad i = 1, 2, \ldots, n,$$

where σ_i denotes the ith singular value of A.

Proof. Clearly, the inverse $\mathcal{M}_2^{-1}$ of preconditioner $\mathcal{M}_2$ can be expressed in the factored form

$$\mathcal{M}_2^{-1} = \begin{pmatrix} I & -B \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & (\theta I + B^T B)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ B^T & I \end{pmatrix}.$$

Thus, $\mathcal{M}_2^{-1}\mathcal{C}$ is of the form

$$\begin{aligned} \mathcal{M}_2^{-1}\mathcal{C} &= \begin{pmatrix} I & -B \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & (\theta I + B^T B)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ B^T & I \end{pmatrix} \begin{pmatrix} I & B \\ -B^T & 0 \end{pmatrix} \\ &= \begin{pmatrix} I & \theta B(\theta I + B^T B)^{-1} \\ 0 & (\theta I + B^T B)^{-1} B^T B \end{pmatrix}. \end{aligned}$$

Let the singular value decomposition of B be given by

$$B = U^T \Lambda V, \tag{2.10}$$

where U and V are orthogonal matrices of order n, and Λ denotes a diagonal matrix with the diagonal entries being $\gamma^{-1}\sigma_i$ for $i = 1, 2, \ldots, n$. Using (2.10), we can factorize $\mathcal{M}_2^{-1}\mathcal{C}$ in the expression

$$\mathcal{M}_2^{-1}\mathcal{C} = \begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix} \begin{pmatrix} I & \theta\Lambda(\theta I + \Lambda^2)^{-1} \\ 0 & \Lambda^2(\theta I + \Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix},$$

where $\Lambda^2(\theta I + \Lambda^2)^{-1}$ is a diagonal matrix, whose ith entry is given by

$$\frac{(\sigma_i/\gamma)^2}{\theta + (\sigma_i/\gamma)^2}, \quad i = 1, 2, \ldots, n.$$

As a result, we obtain the eigenvalues of $\mathcal{M}_2^{-1}\mathcal{C}$. This completes the proof. □

In the case that $\gamma = \sigma_1$, we can establish upper and lower bounds on those n eigenvalues of $\mathcal{M}_2^{-1}\mathcal{C}$ that are different from 1,

$$\frac{1}{1+\theta\kappa^2} \le \frac{\sigma_i^2}{\gamma^2\theta + \sigma_i^2} \le \frac{1}{1+\theta}.$$

Thus, from Theorem 2.2 and the above inequalities it can be seen that for a small value of θ, all the eigenvalues of the preconditioned matrix are either at 1 or clustered around 1. This indicates that solving (1.3) using GMRES with preconditioner $\mathcal{M}_2$ will require few iterations to obtain an approximation of the solution with sufficiently low relative residual.

However, from (2.8) it can be seen that a very small value of θ results in a slow rate of convergence for the SOR method. Hence, there exists a trade off between the spectral radius of the SOR method used to solve linear systems involving the preconditioner and the number of outer GMRES iterations. The smaller the value of θ, the fewer the number of outer GMRES iterations, but the larger the number of inner SOR iterations for solving each linear system involving the preconditioner if one insists on realizing very low relative residuals for solving each inner system. Our extensive numerical experiments have shown that only few SOR iterations (around 10) are needed with a modest SOR spectral radius ($\rho \approx 0.8$).

3. Numerical experiments

In this section, we construct linear systems from three different sources: (i) a two dimensional diffusion–convection model, (ii) an ion channel protein model problem, and (iii) the SuiteSparse Matrix Collection [15]. These linear systems are nonsymmetric and highly indefinite. We use the GMRES algorithm as the outer iterative scheme for solving the augmented linear systems (1.3) with our preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$, and compare the effectiveness and performance of these two preconditioners with an ILU-preconditioned GMRES for solving the original linear system (1.1). We tried several options for ILU, which are specified in Section 3.5.

3.1. A diffusion–convection model for numerical tests

We consider a diffusion–convection model problem as follows:

$$\begin{cases} -\nabla \cdot (D(\mathbf{r})\nabla u(\mathbf{r})) + w(\mathbf{r}) \cdot \nabla u(\mathbf{r}) + c(\mathbf{r})u(\mathbf{r}) = f(\mathbf{r}), & \mathbf{r} \in \Omega, \\ u(\mathbf{r}) = u_D(\mathbf{r}), & \mathbf{r} \in \partial\Omega, \end{cases} \tag{3.1}$$

where Ω is an open bounded domain with the boundary $\partial\Omega$, D is a coefficient matrix with entries being differentiable, c and f are continuous functions, w is a continuous vector function, and u_D is a continuous boundary value function. This model problem is related to many applications in physics, chemistry, and biology [22,23]. For example, in the ion channel modeling study, a solution u of (3.1) gives a concentration of an ionic species crossing the membrane through the pore of an ion channel protein [23]. In such an application, D reflects the diffusion properties of an ionic solvent, and w is an electrostatic force field induced by charges from ions and the atoms of an ion channel molecule.

The finite element approximation of the model problem (3.1) can be derived as follows: Find $u \in \mathcal{V}$ satisfying $u = u_D$ on $\partial\Omega$ such that

$$a(u, v) = L(v) \quad \forall v \in \mathcal{V}_0, \tag{3.2}$$

where $a(u, v)$ is a bilinear functional as defined by

$$a(u, v) = \int_\Omega [D(\mathbf{r})\nabla u] \cdot \nabla v d\mathbf{r} + \int_\Omega [w(\mathbf{r}) \cdot \nabla u] v d\mathbf{r} + \int_\Omega c(\mathbf{r}) u v d\mathbf{r},$$

$L(v) = \int_\Omega f v d\mathbf{r}$, which is a linear functional, $\mathcal{V}$ denotes a linear Lagrange finite element function space defined on a quasi-uniform triangular mesh Ω_h of Ω with h being a mesh size, and $\mathcal{V}_0 = \{u \in \mathcal{V} | u = 0 \text{ on } \partial\Omega\}$. Here $\mathcal{V}$ is a subspace of the normal Sobolev space $H^1(\Omega)$ [24].

Clearly, for a set of basis functions of $\mathcal{V}$, ψ_j for $j = 1, 2, \ldots, n$, the finite element equation (3.2) can be expressed in a matrix form of (1.1) with the ith entries of x and b being u_i and $L(\psi_i)$, respectively, and A having the (i, j)th entries $a(\psi_i, \psi_j)$ for $i, j = 1, 2, \ldots, n$. Since $a(\psi_i, \psi_j) \neq a(\psi_j, \psi_i)$ for $i \neq j$, this matrix A is nonsymmetric. We can construct different indefinite matrices of A through selecting the coefficient functions D, w, and c of the model problem (3.1).

3.2. Linear systems for numerical experiments

We developed a Python program for generating the linear systems (1.1) from the test models (3.1) based on the state of the art finite element library from the FEniCS project [25]. Here, Ω is a 2 dimensional bounded domain with $\mathbf{r} = (x, y)$ as defined in Fig. 1, D is a 2 by 2 coefficient matrix with entries d_{ij} for $i, j = 1, 2$, and a linear finite element method is used to discretize the underlying partial differential equation defined on a quasi-uniform triangulation mesh Ω_h, as illustrated in Fig. 2. In the numerical experiments, we fixed the vector function $w = q/(2\pi)\nabla \ln(x^2 + y^2)$, which is singular at the origin $\mathbf{r} = (0, 0)$, the boundary value function $u_D = \sin(x)\cos(y)$, and the right hand side function f that is selected such that the model (3.1) has the analytical solution $u = \sin(x)\cos(y)$.

Using functions D and c and parameter q in Table 1, we generated the three linear systems of (1.1), which have the following properties, respectively: (i) The coefficient matrix A is dominated by its symmetric part (i.e., $\|H\|_2 > \|S\|_2$), (ii)

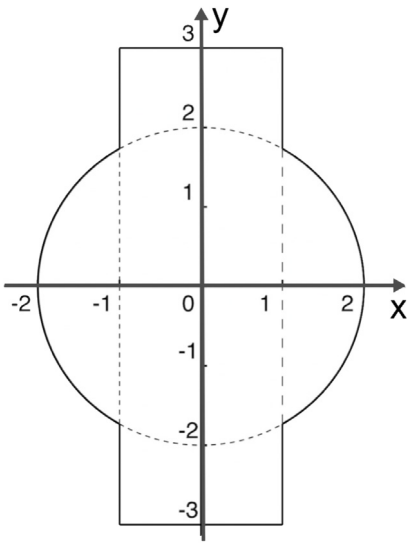


Fig. 1. A two dimensional (2D) domain Ω of model (3.1) for numerical tests.

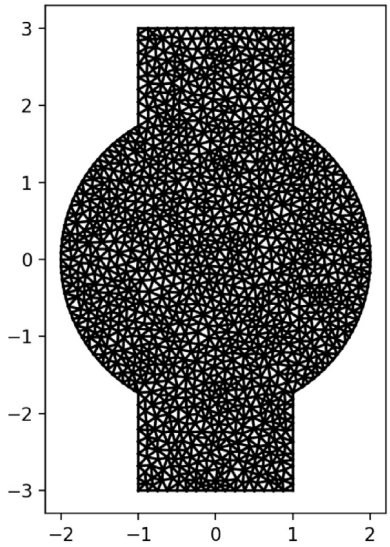


Fig. 2. A triangulation mesh Ω_h of the 2D domain Ω with mesh size $h = 1/32$.

Table 1
Coefficient functions c and D of the diffusion–convection problem (3.1) and the parameter q of vector function w used in Test models 1, 2, and 3.

	Test model 1 ($\|\|H\|\|_2 > \|\|S\|\|_2$)	Test model 2 ($\|\|H\|\|_2 \approx \|\|S\|\|_2$)	Test model 3 ($\|\|H\|\|_2 < \|\|S\|\|_2$)
c	1	1	1
q	1	10^3	10^3
D	$\begin{pmatrix} 10^{-2} & 10^4 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 10^{-2} & 10^4 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 10^{-2} & 10^3 \\ 1 & 10^{-2} \end{pmatrix}$

the symmetric and skew-symmetric parts of A satisfy $\|H\|_2 \approx \|S\|_2$, and (iii) the skew-symmetric part of A dominates its symmetric part (i.e., $\|H\|_2 < \|S\|_2$). Here, H and S are defined by

$$H = \frac{1}{2}(A + A^T), \quad S = \frac{1}{2}(A - A^T), \tag{3.3}$$

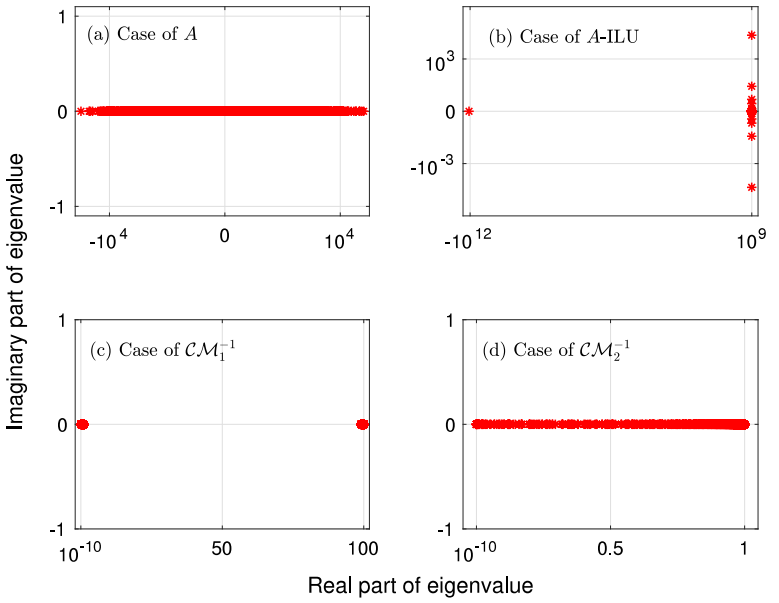


Fig. 3. A comparison of the eigenvalue distributions in Test model 1.

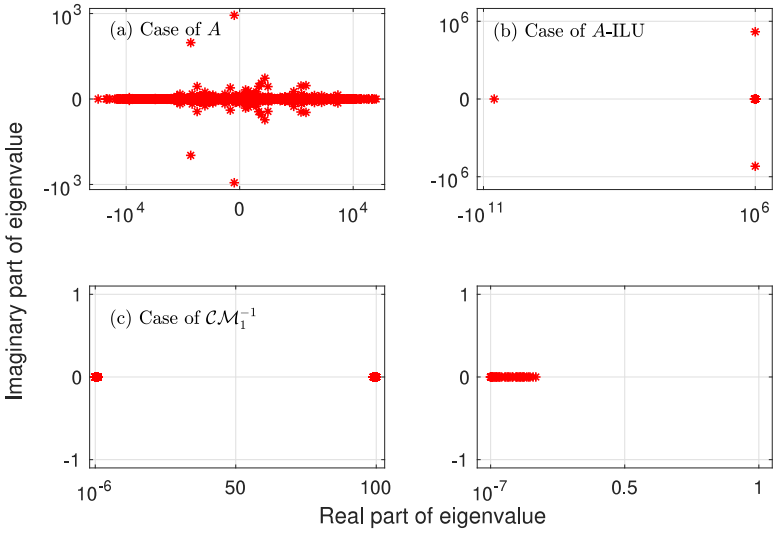


Fig. 4. A comparison of the eigenvalue distributions in Test model 2.

and $\|\cdot\|_2$ denotes the matrix norm induced by the 2-norm for vectors. These three linear systems will be referred to as Test models 1, 2, and 3, respectively.

In each of Figs. 3, 4, and 5 (one figure for each test model), we display the eigenvalue distributions of the original coefficient matrix A, the ILU-preconditioned matrix (A-ILU), and the preconditioned augmented matrices $\mathcal{CM}_1^{-1}$ and $\mathcal{CM}_2^{-1}$. Here we used $\alpha = 0.01$ and $\theta = 0.001$. Because each A-ILU has a few of eigenvalues with huge real parts, several clusters are produced in Figs. 3(b), 4(b), and 5(b). To truly reflect the distributions of these eigenvalues, we plotted the eigenvalues with small real parts in Fig. 6. For example, in the case of Fig. 3(b), there are 1274 eigenvalues that have real parts in the interval $[-70, 30]$ while the remaining 16 eigenvalues have real parts across a wide range from -10^{12} to 10^9.

From Figs. 3(a, b), 4(a, b), 5(a, b), and 6 it can be seen that both A and A-ILU have many eigenvalues across wide ranges. This indicates serious difficulties for unpreconditioned and ILU-preconditioned GMRES methods for these three linear systems.

Figs. 3(c), 4(c), and 5(c) confirm what is claimed in [26,27] — the eigenvalues of $\mathcal{CM}_1^{-1}$ are tightly grouped into two clusters, where each cluster consists of the eigenvalues that are distributed across a small range. For example, in the case

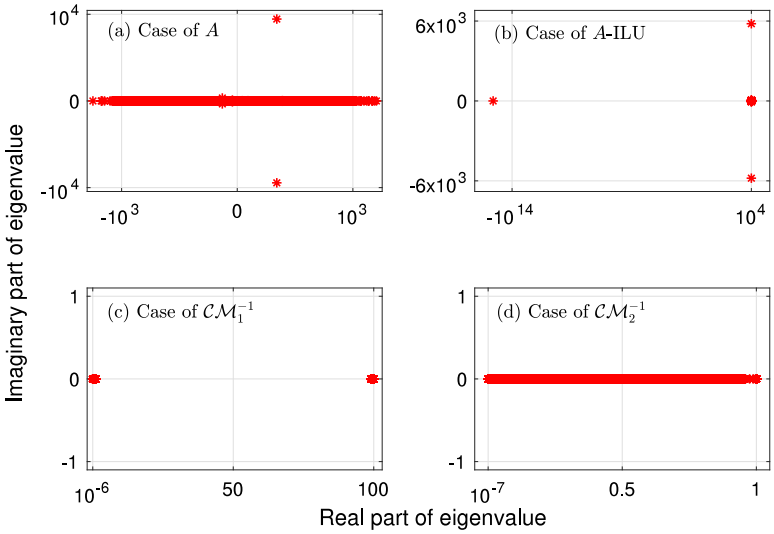


Fig. 5. A comparison of the eigenvalue distributions in Test model 3.

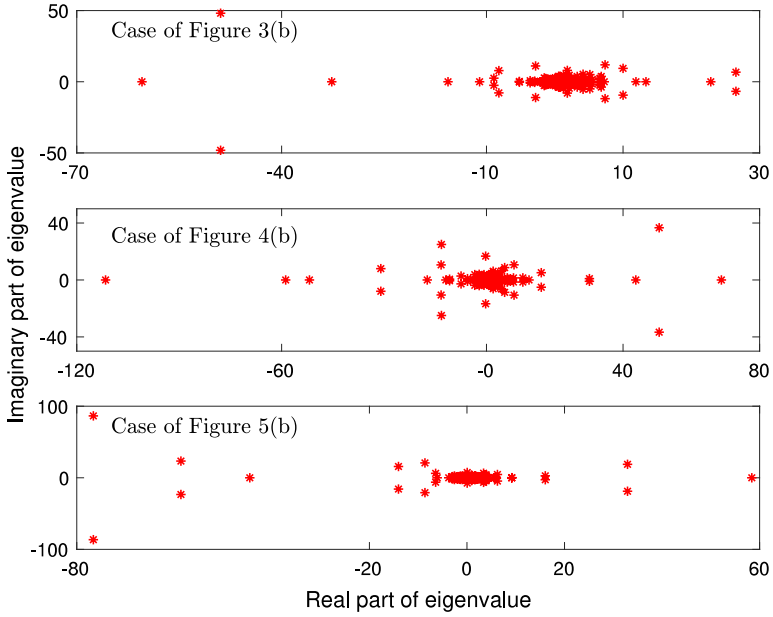


Fig. 6. The eigenvalues with small scales from Figs. 3(b), 4(b), and 5(b).

of Fig. 3(c), the left cluster consists of the 1290 eigenvalues with real parts between 0 and 1, and the right cluster consists of the 1290 eigenvalues with real parts varying from 99 to 100.

Figs. 3(d), 4(d), and 5(d) show that the eigenvalues of $\mathcal{CM}_2^{-1}$ are distributed in a small range with the real parts between 0 and 1. These results illustrate that GMRES preconditioned with $\mathcal{M}_1$ or $\mathcal{M}_2$ can be quite effective in solving these augmented linear system of (1.3).

Fig. 7 compares the numerical solution u_h of Test model 1 with the analytical solution u of the model problem (3.1) for a mesh domain Ω_h with $h = 1/32$. This demonstrates the close agreement between the exact and the finite element solutions. The relative error between the finite element solution u_h and u is 0.543 in the $L_2(\Omega)$ norm. We also validated such agreements for Test models 2 and 3.

Table 2 lists the values of the spectral condition number κ of the coefficient matrix A, the numbers $n_+(H)$ and $n_-(H)$ of positive and negative eigenvalues of H, respectively, and the values of σ_1 for Test models 1, 2, and 3. Here σ_1 was calculated approximately by the limited memory block Krylov subspace optimization method [19] with the options $opts.tol = 10^{-4}$

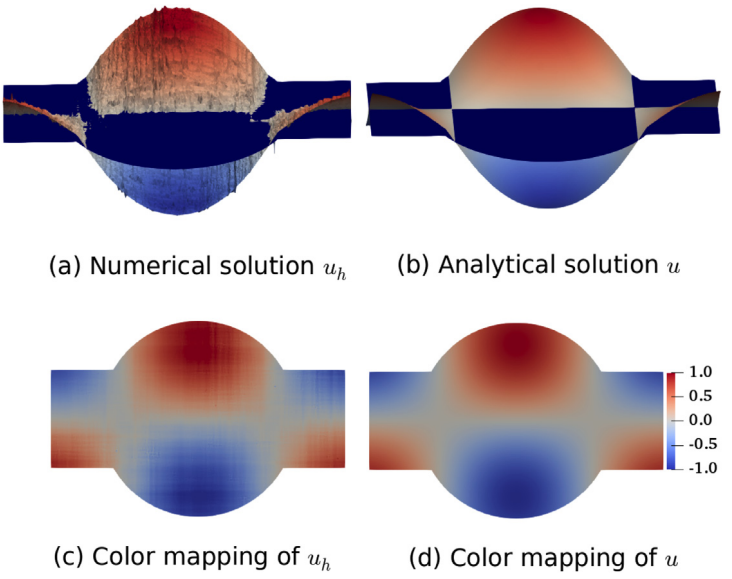


Fig. 7. (a, b) A view of solution in surface. (c, d) A color mapping of solution onto domain Ω. Here, the mesh size $h = 1/32$ for Test model 1.

Table 2
Matrix properties of Test models 1, 2 and 3.

	Test model 1 ($\|\|H\|\|_2 > \|\|S\|\|_2$)	Test model 2 ($\|\|H\|\|_2 \approx \|\|S\|\|_2$)	Test model 3 ($\|\|H\|\|_2 < \|\|S\|\|_2$)
Ω_h with $h = 1/32$ and 1290 interior mesh points			
σ_1	2.5×10^4	2.5×10^4	1.6×10^4
κ	8.1×10^6	1.5×10^5	2.6×10^5
$\|\|H\|\|_2$	2.5×10^4	2.5×10^4	7.2×10^3
$\|\|S\|\|_2$	1.1×10^1	1.1×10^4	1.1×10^4
$n_+(H)$	649	648	647
$n_-(H)$	641	642	643
Ω_h with $h = 1/64$ and 5324 interior mesh points			
σ_1	2.6×10^4	3.5×10^4	3.3×10^4
κ	1.9×10^7	8.8×10^6	1.7×10^7
$\|\|H\|\|_2$	2.6×10^4	2.6×10^4	1.3×10^4
$\|\|S\|\|_2$	2.4×10^1	2.4×10^4	2.4×10^4
$n_+(H)$	2659	2659	2658
$n_-(H)$	2665	2665	2666

and *opts.maxit* $= 200$. We set $\gamma = \sigma_1$ in the construction of augmented linear system (1.3). In all our experiments, the time consumed for estimating the largest singular value σ_1 of A is not more than 5% of the total CPU time.

3.3. Software packages for numerical experiments

We developed a Matlab code for the GMRES method for solving the augmented linear system (1.3) using our preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$ and sparse matrix techniques. Our Python code is called by our Matlab program to generate the coefficient matrix A and the right hand side vector b for Test model 1, 2, or 3. Linear systems involving preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$ are solved approximately by using the optimal relaxation parameter and a fixed number of SOR iterations (SOR-Ite). The outer GMRES iteration stopping criterion is defined by

$$\text{Relres}(\mathcal{C}) < \eta, \tag{3.4}$$

where $\eta = 10^{-7}$ by default, and $\text{Relres}(\mathcal{C}) = \|d - \mathcal{C}y^{(k)}\|_2/\|d\|_2$, which gives the relative residual norm of the augmented linear system (1.3). Here, $y^{(k)}$ denotes the kth outer GMRES iterate for $k = 0, 1, 2, \ldots$ with $y^{(0)}$ being an initial guess, and $\|\cdot\|_2$ is the 2-norm for vectors. This version of GMRES will be referred to as the GMRES-SOR method.

All the numerical experiments were performed on an iMac desktop with 4.3 GHz quad-core Intel Core i7 with 64 GB main memory. Here the initial guess $y^{(0)}$ was simply set as the zero vector.

Table 3
Validation of our software package for solving the diffusion–convection problem (3.1). Here, p_k is an estimation of finite element error order defined in (3.5), and N_{h_k} is the number of unknowns.

h_k	N_{h_k}	Relres$(A) < 10^{-3}$		Relres$(A) < 10^{-4}$		Relres$(A) < 10^{-5}$	
		E_α	p_k	E_α	p_k	E_α	p_k
1/4	1 457	4.736×10^{-2}		4.736×10^{-2}		4.736×10^{-2}	
1/8	6 481	1.234×10^{-2}	1.940	1.235×10^{-3}	1.938	1.236×10^{-2}	1.938
1/16	1 457	3.130×10^{-3}	1.979	3.120×10^{-3}	1.985	3.122×10^{-3}	1.984
1/32	6 481	9.772×10^{-4}	1.679	7.798×10^{-4}	2.001	7.827×10^{-4}	1.996
1/64	27 281	7.368×10^{-4}	0.407	1.989×10^{-4}	1.971	1.960×10^{-4}	1.998
1/128	111 889	7.866×10^{-4}	−0.094	7.933×10^{-5}	1.320	4.966×10^{-5}	1.980
1/256	453 137	8.312×10^{-4}	−0.080	7.242×10^{-5}	0.131	1.225×10^{-5}	2.019

3.4. Validation tests for our GMRES-SOR software package

We performed validation tests for our GMRES-SOR software package using a test model of (3.2) with $\Omega = (-1, 1) \times (-1, 1)$, $D = [1, 1; 1, 1]$, $c = 1$, and $w = 1/(2\pi)\nabla \ln(x^2 + y^2 + 1)$. We discretized this test model by a linear finite element method based on the uniform meshes of Ω with the mesh sizes $h_k = 1/2^k$ for $k = 2$ to 8, deriving 7 linear systems with coefficient matrices A. We solved these linear systems by our GMRES-SOR method to yield approximate solutions, $x^{(k)}$, of the original linear system (1.1) satisfying

$$\text{Relres}(A) < \eta,$$

where $\eta = 10^{-3}$, 10^{-4}, and 10^{-5}, and Relres(A) is defined by

$$\text{Relres}(A) = \frac{\|b - Ax^{(k)}\|_2}{\|b\|_2}.$$

We then calculated the absolute error, $E_a(h)$, between the finite element solution u_h and the analytical solution u in the $L_2(\Omega)$ norm via

$$E_a(h) = \sqrt{\int_\Omega |u_h(\mathbf{r}) - u(\mathbf{r})|^2 d\mathbf{r}},$$

and estimated the convergence order p of the finite element method by

$$p \approx p_k = \ln(|E_a(h_k)|/|E_a(h_{k-1})|)/\ln(h_k/h_{k-1}) \quad \text{for } k = 2 \text{ to } 8. \tag{3.5}$$

The numerical results are reported in Table 3.

For a linear finite element approximation, p is known to be 2 so that p_k should approach 2 as h_k approaches zero if the calculation is correct [24]. From Table 3 it can be seen that E_a approaches zero monotonically as the grid size h_k decreases, and p_k approaches 2 when the numerical solutions satisfy Relres$(A) < 10^{-5}$. Hence, these numerical results well validate our GMRES-SOR method and related software packages.

3.5. Test results

Using our Matlab and Python programs, we conducted the numerical tests using outer GMRES iterations with the preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$ for solving the augmented systems (1.3), and compared them with the ILU-preconditioned GMRES and unpreconditioned GMRES for solving the original linear system (1.1). In these tests, we constructed the linear systems resulting from Test models 1, 2, and 3 based on the mesh domain Ω_h with mesh size $h = 1/32$ and 1290 interior mesh points. To explore the effectiveness of preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$, we solved systems involving them exactly in each outer GMRES iteration, from which we also derived the lowest number of outer iterations realizable using these preconditioners. In what follows, we refer to this version of preconditioned GMRES by GMRES*. The numerical test results are reported in Table 4 and Figs. 8 and 9. Here, "Ite" denotes the number of outer iterations, and the iteration stopping criteria included both (3.4) and Ite $\leq$ 1000.

Table 4 shows that our preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$ for the augmented systems succeeded in satisfying our stopping criteria, while the unpreconditioned and the ILU-preconditioned GMRES for solving the original linear system failed for all the three test models. Furthermore, in Figs. 8 and 9, we display the behavior of the relative residual norms versus the number of outer iterations for Test models 1 and 3, respectively. From these figures we can see that the unpreconditioned and ILU-preconditioned GMRES reduced the relative residual norm very slowly, and the relative residual norm of the unpreconditioned GMRES reached only $O(10^{-3})$ before the last iteration Ite = 1290 (the size of the linear system). In contrast, our GMRES* reduced the relative residual norm quickly. These tests suggest that the augmented linear system approach is effective for handling highly indefinite linear systems.

Table 4
A comparison of GMRES* with ILU preconditioned GMRES for Test models 1, 2, and 3 based on the mesh domain Ω_h with mesh size $h = 1/32$.

	Test model 1		Test model 2		Test model 3	
	Ite	Relres(A)	Ite	Relres(A)	Ite	Relres(A)
No preconditioner	1000	2.0×10^{-3}	1000	7.7×10^{-3}	1000	1.4×10^{-3}
ILU	1000	3.4×10^{-2}	1000	2.2×10^{-2}	1000	1.6×10^{-1}
$\mathcal{M}_1$ ($\alpha = 0.01$)	89	3.5×10^{-4}	74	8.9×10^{-4}	180	7.8×10^{-4}
$\mathcal{M}_2$ ($\theta = 0.001$)	127	3.8×10^{-4}	112	8.9×10^{-4}	271	7.8×10^{-4}

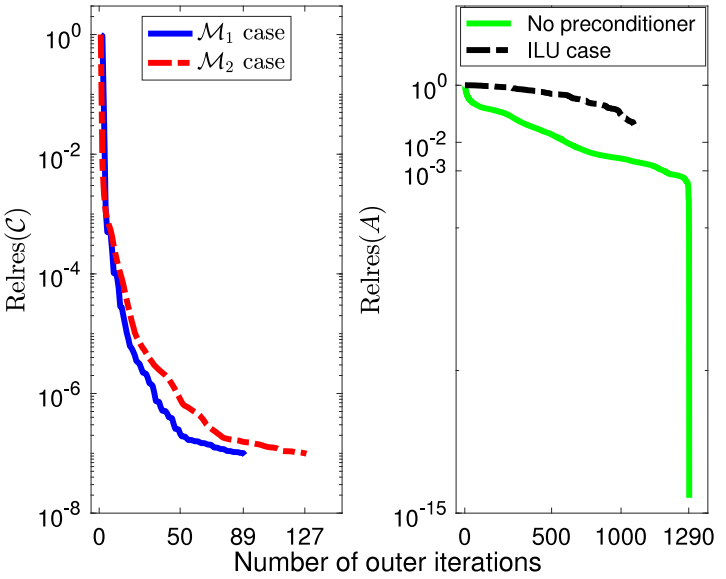


Fig. 8. A comparison of the relative residual reduction processes of GMRES* (left plot) with GMRES for solving the linear system $Ax = b$ without and with an ILU preconditioner (right plot) for Test model 1.

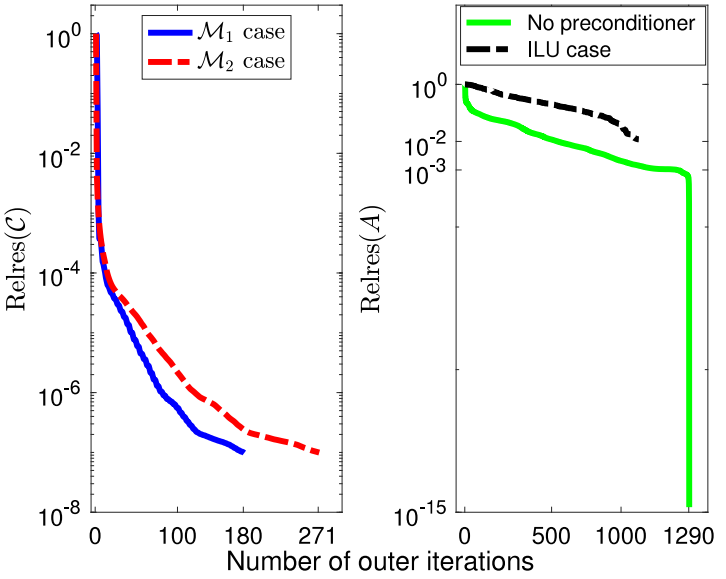


Fig. 9. A comparison of the relative residual reduction processes of the GMRES* (left plot) with GMRES for solving the linear system $Ax = b$ without and with an ILU preconditioner (right plot) for Test model 3.

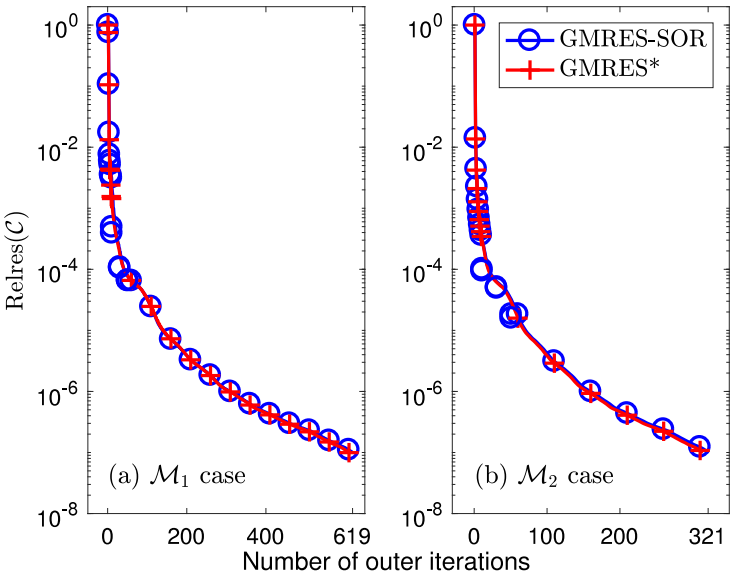


Fig. 10. A comparison of the relative residual reduction processes of GMRES-SOR (with SOR-Ite = 50) and GMRES* for Test model 1 using $h = 1/64$.

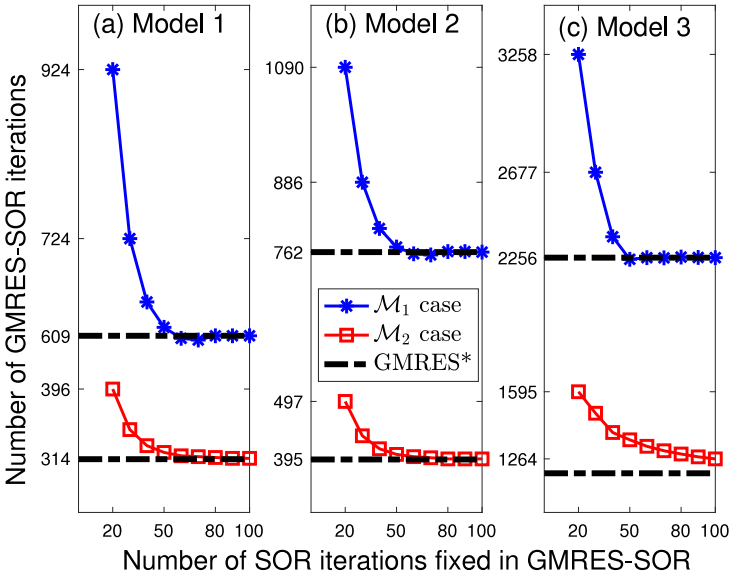


Fig. 11. GMRES-SOR vs. GMRES* (h = 1/64) for Test models 1, 2, 3.

To explore the performance of GMRES-SOR, we solved the linear systems of Test models 1, 2 and 3 based on the mesh domain Ω_h with $h = 1/32$ and $1/64$. Here, we used SOR-Ite = 50, and $\rho = 0.9$, from which we derived $\alpha \approx 0.0527$ and $\theta \approx 0.0028$ from (2.5) and (2.9), respectively. Numerical results are reported in Table 5. They demonstrate the performance of GMRES-SOR in terms of outer iteration numbers, CPU time, and the relative residual realized for the original systems. Furthermore, we compare a relative residual reduction process of the GMRES-SOR method with that of GMRES* in Fig. 10, showing that the GMRES-SOR method can have a behavior of Relres($\mathcal{C}$) that is almost identical to that of GMRES*. Further, Fig. 10 reveals that $\mathcal{M}_2$ uniformly requires fewer iterations than $\mathcal{M}_1$ for Test models 1, 2, 3.

In order to demonstrate the influence of using a fixed number of inner SOR iterations (i.e., SOR-Ite), we used all the parameters in Table 5 except varying SOR-Ite from 20 to 100 with increments of 10. Fig. 11 shows that for the linear systems derived from Test models 1, 2, and 3, setting SOR-Ite = 50 is sufficient for GMRES-SOR to realize the same number of outer iterations as GMRES* under the same stopping criterion Relres($\mathcal{C}$)$< 10^{-7}$.

In addition, we considered two more linear systems (Test models 4 and 5) to further explore the performance of our GMRES-SOR method. Test model 4 has the coefficient matrix A selected from the SuiteSparse Matrix Collection

Table 5
Performance of our GMRES-SOR method for solving Test models 1, 2, and 3. Here, each preconditioning linear system was solved by at most 50 SOR iterations, and CPU denotes the total time for solving an augmented linear system in seconds.

Preconditioner for GMRES	Test model 1			Test model 2			Test model 3		
	Ite	CPU	Relres(A)	Ite	CPU	Relres(A)	Ite	CPU	Relres(A)
Case with $h = 1/32$									
$\mathcal{M}_1$	411	0.7	3.8×10^{-4}	368	0.6	9.0×10^{-4}	832	2.0	7.8×10^{-4}
$\mathcal{M}_2$	214	0.3	3.8×10^{-4}	190	0.2	9.0×10^{-4}	458	0.8	7.9×10^{-4}
Case with $h = 1/64$									
$\mathcal{M}_1$	619	6.4	4.8×10^{-4}	771	8.9	5.3×10^{-4}	2248	51.2	1.3×10^{-3}
$\mathcal{M}_2$	321	2.5	4.8×10^{-4}	403	3.5	5.3×10^{-4}	1358	21.7	1.3×10^{-3}

Table 6
Properties of Test models 4 and 5. Here *nnz* denotes the number of nonzero entries.

	n	nnz	σ_1	κ	$\|\|H\|\|_2$	$\|\|S\|\|_2$
Test model 4	76 480	329 762	62.79	8.2×10^{277}	62.5	25.1
Test model 5	14 203	191 689	178.58	9.4×10^{6}	122.2	102.0

Table 7
Performance of our GMRES-SOR method for solving Test models 4 and 5.

Preconditioner for GMRES	Test model 4			Test model 5		
	Ite	CPU	Relres(A)	Ite	CPU	Relres(A)
$\mathcal{M}_1$	210	7.8	2.5×10^{-6}	993	78.6	9.8×10^{-4}
$\mathcal{M}_2$	84	3.1	2.3×10^{-6}	217	14.3	9.7×10^{-4}

(https://sparse.tamu.edu/Shyy/shyy161). This matrix is derived from viscous flow calculation via a direct, fully-coupled method for solving the Navier–Stokes equations. It is nonsymmetric and highly indefinite. The right hand side vector b is simply set as $b = A * ones(n, 1)$.

Test model 5 was generated from a linear finite element approximation to the Nernst–Planck boundary value problem:

$$\nabla \cdot D(\mathbf{r})\left[\nabla u(\mathbf{r}) + u(\mathbf{r})\nabla \Phi(\mathbf{r})\right] = 0, \quad \mathbf{r} \in \Omega, \tag{3.6}$$

where Ω is a three dimensional bounded domain with $\mathbf{r} = (x, y, z)$, as illustrated in Fig. 12, $D(\mathbf{r})$ is a diffusion coefficient function, Φ is an electrostatic potential function induced from the atomic charges of an ion channel protein and the ionic charges of a salt (NaCl) solution (in 0.1 moles per liter) contained in the domain Ω, and u is a concentration function of sodium ions (Na^+). The dimensions of Ω are marked on Fig. 12(b, c). In the numerical tests, we set $D(\mathbf{r}) = 0.0196$ in the channel pore portion (the middle part of Ω in the z-axis direction), and 0.196 in the other part of Ω. The ion channel protein was downloaded from the Protein Data Bank (PDB) website http://www.rcsb.org with the PDB ID: 1MAG.pdb. A mixed boundary value condition is applied to the Nernst–Planck equation (3.6): (i) $u(\mathbf{r}) = 0.1$ on the bottom and top surfaces defined by $z = -33.42, 26.58$ (i.e., Dirichlet boundary condition), (ii) u is periodic on the four side surfaces of Ω defined by $x = -20.32, 19.68$ and $y = -20, 20$, and (iii) u satisfies the Robin boundary condition

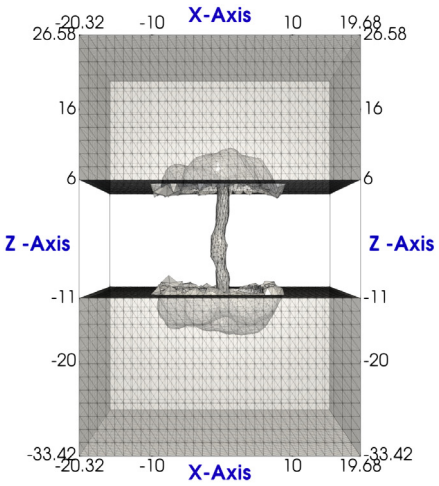
$$\frac{\partial u(\mathbf{r})}{\partial \mathbf{n}} + u(\mathbf{r})\frac{\partial \Phi(\mathbf{r})}{\partial \mathbf{n}} = 0$$

on the remainder of the boundary $\partial\Omega$. Here $\mathbf{n}$ denotes the unit outward normal direction of Ω. An irregular tetrahedral mesh, as illustrated in Fig. 12, is used to generate the coefficient matrix A and the right hand side vector b.

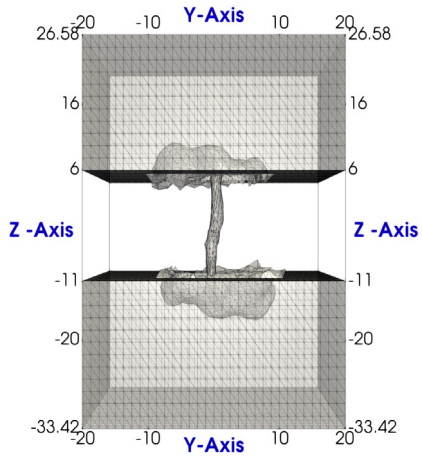
Relevant properties of the coefficient matrices of Test models 4 and 5, are listed in Table 6. We use the restarted GMRES(50)-SOR method (restart = 50) for both Test models 4 and 5. For Test model 4, we set the SOR spectral radius $\rho = 0.9$, SOR-Ite = 50, and $\eta = 10^{-10}$ for the iteration stopping criterion (3.4). For Test model 5, we set $\rho = 0.98$, SOR-Ite = 200, and $\eta = 10^{-7}$. From (2.5) and (2.9), we have $\alpha \approx 0.0527$ and $\theta \approx 0.0028$ for Test model 4 and $\alpha \approx 0.01$ and $\theta \approx 0.0001$ for Test model 5. The test results are reported in Table 7. In this table, the total number of GMRES iterations, Ite, is computed via,

$$\text{Ite} = (i - 1) \times 50 + j,$$

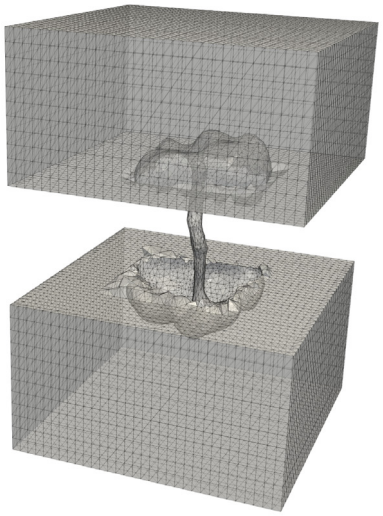
where i is the number of restarts, and j is the number of iterations taken in the last restart step. Similar to Test models 1, 2 and 3, we find that the restarted GMRES method failed for the linear system of Test models 4 and 5 with ILU preconditioning. Here, we used several ILU options from the MATLAB library such as *setup.type* = *"crout"*, *setup.milu* = *"row"*, and *setup.droptol* = 0.001. For example, for Test model 5, these options resulted in the lower and upper triangular



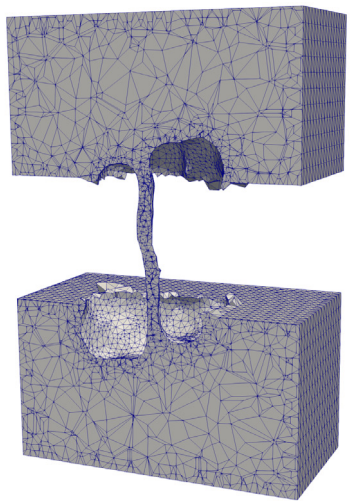
(a) Dimensions in x/z axes.



(b) Dimensions in y/z axes.



(c) A view of the whole domain



(d) A view of the clip surface mesh

Fig. 12. Dimensions and mesh views of the domain Ω for Test model 5.

matrices L and U of ILU having 694 779 and 705 871 nonzero entries, respectively (i.e., too many fill-in produced). For Test model 4, these options failed. Further, we performed experiments using the options *setup.type* = *"ilutp"* and *setup.droptol* = 0.001, resulting in GMRES stopped after 5 iterations with relres(A) = 0.6171 only even though we used the stopping criterion relres(A) $< 10^{-6}$.

From Table 7 it can be seen that both $\mathcal{M}_1$ and $\mathcal{M}_2$ are successful with $\mathcal{M}_2$ being more effective than $\mathcal{M}_1$ in requiring a fewer number of outer GMRES iterations. It is interesting to note that Test model 4 has a much smaller value of σ_1 than Test model 5, which caused our GMRES-SOR method to produce numerical solutions with lower relative residuals for Test model 4 than those for Test model 5 even though Test model 4 has a much larger condition number κ. This indicates that σ_1 is a more important factor than κ in the determination of the effectiveness of our GMRES-SOR method. In other words, our GMRES-SOR method is especially effective for linear systems with σ_1 not very large even though the condition number of A may be very large.

Finally, we considered a much larger linear system selected from the SuiteSparse Matrix Collection (Hamrle3, https://sparse.tamu.edu/Hamrle). This linear system $Ax = b$ has size $n = 1,447,360$. The nonsymmetric coefficient matrix A has $5,514,242$ nonzero entries, and the right-hand side vector $b = A * ones(n, 1)$. For this system, we found that $\sigma_1 \approx 16.1$ and $\kappa \approx 1.6 \times 10^7$. Computing σ_1 consumed only 10 s. GMRES(50) with the ILU preconditioner (no-refills) for solving the system $Ax = b$ failed. Attempting to solve the corresponding augmented system by GMRES(50)-SOR using $\mathcal{M}_1$ as a

preconditioner failed as well. But, using $\mathcal{M}_2$ as a preconditioner proved to be successful in satisfying the stopping criterion (3.4) with $\eta = 10^{-6}$ in 186 iterations, which consumed 217 s (CPU time), and yielded Relres$(A) = 3.5 \times 10^{-3}$. In these GMRES-SOR tests, we used $\rho = 0.95$, and SOR-Ite $= 50$.

4. Conclusion

We have demonstrated that for strongly indefinite sparse linear systems $Ax = b$, casting the problem as a weighted linear least squares problem enabled the creation of two effective preconditioners $\mathcal{M}_1$ and $\mathcal{M}_2$ of the corresponding augmented linear systems. Using GMRES preconditioned by $\mathcal{M}_1$ or $\mathcal{M}_2$ proved to be successful in obtaining an approximation of the solution x while ILU-preconditioned GMRES failed in solving $Ax = b$. Further, our approach proved to be successful in solving those ill-conditioned augmented systems due to favorable eigenvalue distributions of the preconditioned augmented matrices. In addition, we have shown that the SOR method can be effectively used in solving linear system involving $\mathcal{M}_1$ and $\mathcal{M}_2$ which can be useful for implementation on parallel computing platforms since in each inner iteration one avoids global inner products.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

References

[1] M. Stynes, D. Stynes, Convection Diffusion Problems: An Introduction to their Analysis and Numerical Solution, in: Graduate Studies in Mathematics, vol. 196, American Mathematical Society, 2018.
[2] J. Douglas Jr., T.F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 19 (5) (1982) 871–885.
[3] H.-G. Roos, M. Stynes, L. Tobiska, Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection–Diffusion-Reaction and Flow Problems, Vol. 24, Springer Science & Business Media, 2008.
[4] R.S. Varga, Matrix Iterative Analysis, Vol. 27, Springer Science & Business Media, 2009.
[5] Z.-Z. Bai, Quasi-hss iteration methods for non-hermitian positive definite linear systems of strong skew-Hermitian parts, Numer. Linear Algebra Appl. 25 (4) (2018) e2116.
[6] O. Axelsson, Z.-Z. Bai, 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 (2–4) (2004) 351–372.
[7] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2) (2002) 418–477.
[8] M. Benzi, J.C. Haws, M. Tuma, Preconditioning highly indefinite and nonsymmetric matrices, SIAM J. Sci. Comput. 22 (4) (2000) 1333–1353.
[9] Y. Saad, Preconditioning techniques for nonsymmetric and indefinite linear systems, J. Comput. Appl. Math. 24 (1–2) (1988) 89–105.
[10] E. Chow, Y. Saad, Experimental study of ilu preconditioners for indefinite matrices, J. Comput. Appl. Math. 86 (2) (1997) 387–414.
[11] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[12] M. Benzi, G.H. Golub, A preconditioner for generalized saddle point problems, SIAM J. Matrix Anal. Appl. 26 (1) (2004) 20–41.
[13] B. Fischer, A. Ramage, D.J. Silvester, A.J. Wathen, Minimum residual methods for augmented systems, BIT Numer. Math. 38 (3) (1998) 527–543.
[14] Y. Saad, M.H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[15] T.A. Davis, Y. Hu, The university of Florida sparse matrix collection, ACM Trans. Math. Softw. (TOMS) 38 (1) (2011) 1.
[16] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl. 24 (3) (2003) 603–626.
[17] Z. Zhang, A.H. Sameh, A parallel sparse linear system solver based on hermitian/skew-Hermitian splitting, Comput. Math. Appl. 72 (8) (2016) 2000–2007.
[18] D.M. Young, Iterative Solution of Large Linear Systems, Academic Press, 1971.
[19] X. Liu, Z. Wen, Y. Zhang, Limited memory block Krylov subspace optimization for computing dominant singular value decompositions, SIAM J. Sci. Comput. 35 (3) (2013) A1641–A1668.
[20] O. Axelsson, Preconditioning of indefinite problems by regularization, SIAM J. Numer. Anal. 16 (1) (1979) 58–69.
[21] O. Axelsson, M. Neytcheva, Preconditioning methods for linear systems arising in constrained optimization problems, Numer. Linear Algebra Appl. 10 (1–2) (2003) 3–31.
[22] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Vol. 33, Springer Science & Business Media, 2013.
[23] C. Maffeo, S. Bhattacharya, J. Yoo, D. Wells, A. Aksimentiev, Modeling and simulation of ion channels, Chem. Rev. 112 (12) (2012) 6250–6284.
[24] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, third ed., Springer-Verlag, New York, 2008.
[25] A. Logg, K.-A. Mardal, G. Wells, Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, Vol. 84, Springer Science & Business Media, 2012.
[26] V. Simoncini, M. Benzi, Spectral properties of the hermitian and skew-hermitian splitting preconditioner for saddle point problems, SIAM J. Matrix Anal. Appl. 26 (2) (2004) 377–389.
[27] M. Benzi, M.J. Gander, G.H. Golub, Optimization of the hermitian and skew-hermitian splitting iteration for saddle-point problems, BIT Numer. Math. 43 (5) (2003) 881–900.