A Parameterized Splitting Preconditioner for Generalized Saddle Point Problems

Wei-Hua Luo¹,² and Ting-Zhu Huang¹

¹ School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
² Key Laboratory of Numerical Simulation of Sichuan Province University, Neijiang Normal University, Neijiang, Sichuan 641112, China

Correspondence should be addressed to Ting-Zhu Huang; tingzhuhuang@126.com

Received 4 January 2013; Revised 31 March 2013; Accepted 1 April 2013

Academic Editor: P. N. Shivakumar

Copyright © 2013 W.-H. Luo and T.-Z. Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By using Sherman-Morrison-Woodbury formula, we introduce a preconditioner based on parameterized splitting idea for generalized saddle point problems which may be singular and nonsymmetric. By analyzing the eigenvalues of the preconditioned matrix, we find that when \( \alpha \) is big enough, it has an eigenvalue at 1 with multiplicity at least \( n \), and the remaining eigenvalues are all located in a unit circle centered at 1. Particularly, when the preconditioner is used in general saddle point problems, it guarantees eigenvalue at 1 with the same multiplicity, and the remaining eigenvalues will tend to 1 as the parameter \( \alpha \to 0 \). Consequently, this can lead to a good convergence when some GMRES iterative methods are used in Krylov subspace. Numerical results of Stokes problems and Oseen problems are presented to illustrate the behavior of the preconditioner.

1. Introduction

In some scientific and engineering applications, such as finite element methods for solving partial differential equations [1, 2], and computational fluid dynamics [3, 4], we often consider solutions of the generalized saddle point problems of the form

\[
\begin{pmatrix}
A & B^T \\
-B & C
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
=
\begin{pmatrix}
f \\
g
\end{pmatrix},
\]

where \( A \in \mathbb{R}^{m \times n} \), \( B \in \mathbb{R}^{m \times n} \) (\( m \leq n \)), and \( C \in \mathbb{R}^{m \times m} \) are positive semidefinite, \( x, f \in \mathbb{R}^n \), and \( y, g \in \mathbb{R}^m \). When \( C = 0 \), (1) is a general saddle point problem which is also a researching object for many authors.

It is well known that when the matrices \( A, B, \) and \( C \) are large and sparse, the iterative methods are more efficient and attractive than direct methods assuming that (1) has a good preconditioner. In recent years, a lot of preconditioning techniques have arisen for solving linear system; for example, Saad [5] and Chen [6] have comprehensively surveyed some classical preconditioning techniques, including ILU preconditioner, triangular preconditioner, SPAI preconditioner, multilevel recursive Schur complements preconditioner, and sparse wavelet preconditioner. Particularly, many preconditioning methods for saddle problems have been presented recently, such as dimensional splitting (DS) [7], relaxed dimensional factorization (RDF) [8], splitting preconditioner [9], and Hermitian and skew-Hermitian splitting preconditioner [10].

Among these results, Cao et al. [9] have used splitting idea to give a preconditioner for saddle point problems where the matrix \( A \) is symmetric and positive definite and \( B \) is of full row rank. According to his preconditioner, the eigenvalues of the preconditioned matrix would tend to 1 when the parameter \( t \to \infty \). Consequently, just as we have seen from those examples of [9], preconditioner has guaranteed a good convergence when some iterative methods were used.

In this paper, being motivated by [9], we use the splitting idea to present a preconditioner for the system (1), where \( A \) may be nonsymmetric and singular (when \( \text{rank}(B) < m \)). We find that, when the parameter is big enough, the preconditioned matrix has the eigenvalue at 1 with multiplicity at least \( n \), and the remaining eigenvalues are all located in a unit circle centered at 1. Particularly, when the preconditioner is used in some general saddle point problems (namely, \( C = 0 \), we...
see that the multiplicity of the eigenvalue at 1 is also at least \( n \), but the remaining eigenvalues will tend to 1 as the parameter \( \alpha \to 0 \).

The remainder of the paper is organized as follows. In Section 2, we present our preconditioner based on the splitting idea and analyze the bound of eigenvalues of the preconditioned matrix. In Section 3, we use some numerical examples to show the behavior of the new preconditioner. Finally, we draw some conclusions and outline our future work in Section 4.

2. A Parameterized Splitting Preconditioner

Now we consider using splitting idea with a variable parameter to present a preconditioner for the system (1).

Firstly, it is evident that when \( \alpha \neq 0 \), the system (1) is equivalent to

\[
\begin{pmatrix}
A & B^T \\
-B & C
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
= \begin{pmatrix}
f \\
g
\end{pmatrix}.
\]

(2)

Let

\[
M = \begin{pmatrix}
A & B^T \\
-B & C
\end{pmatrix}, \quad N = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix},
\]

(3)

\[
X = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F = \begin{pmatrix} f \\ g \end{pmatrix}.
\]

Then the coefficient matrix of (2) can be expressed by \( M - N \).

Multiplying both sides of system (2) from the left with matrix \( M^{-1} \), we have

\[
(I - M^{-1}N)X = M^{-1}F.
\]

(4)

Hence, we obtain a preconditioned linear system from (1) using the idea of splitting and the corresponding preconditioner is

\[
H = \begin{pmatrix}
A & B^T \\
-B & C
\end{pmatrix}^{-1} \begin{pmatrix}
I & 0 \\
0 & \frac{1}{\alpha}
\end{pmatrix}.
\]

(5)

Now we analyze the eigenvalues of the preconditioned system (4).

**Theorem 1.** The preconditioned matrix \( I - M^{-1}N \) has an eigenvalue at 1 with multiplicity at least \( n \). The remaining eigenvalues \( \lambda \) satisfy

\[
\lambda = \frac{s_1 + s_2}{s_1 + \alpha},
\]

(6)

where \( s_1 = \omega^T BA^{-1}B^T \omega \), \( s_2 = \omega^T C \omega \), and \( \omega \in \mathbb{C}^m \) satisfies

\[
\left( BA^{-1}B^T + \frac{C}{\alpha} \right) \omega = \lambda \left( I + \frac{BA^{-1}B^T}{\alpha} \right) \omega, \quad \| \omega \| = 1.
\]

(7)

**Proof.** Because

\[
M^{-1} = \begin{pmatrix} (A + \frac{B^T B}{\alpha})^{-1} & -A^{-1}B^T \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \\ \frac{B}{\alpha} \left( A + \frac{B^T B}{\alpha} \right)^{-1} & \frac{1}{\alpha} \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \end{pmatrix},
\]

we can easily get

\[
I - M^{-1}N = \begin{pmatrix}
I & A^{-1}B^T \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \left( I + \frac{C}{\alpha} \right) \\
0 & \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \left( \frac{BA^{-1}B^T}{\alpha} + \frac{C}{\alpha} \right)
\end{pmatrix},
\]

(9)

which implies the preconditioned matrix \( I - M^{-1}N \) has an eigenvalue at 1 with multiplicity at least \( n \).

For the remaining eigenvalues, let

\[
\left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \left( \frac{BA^{-1}B^T}{\alpha} + \frac{C}{\alpha} \right) \omega = \lambda \omega
\]

with \( \| \omega \| = 1 \); then we have

\[
\left( \frac{BA^{-1}B^T}{\alpha} + \frac{C}{\alpha} \right) \omega = \lambda \left( I + \frac{BA^{-1}B^T}{\alpha} \right) \omega.
\]

(10)

By multiplying both sides of this equality from the left with \( \omega^T \), we can get

\[
\lambda = \frac{s_1 + s_2}{s_1 + \alpha}.
\]

(12)

This completes the proof of Theorem 1. \( \square \)

**Remark 2.** From Theorem 1, we can get that when parameter \( \alpha \) is big enough, the modulus of non-nil eigenvalues \( \lambda \) will be located in interval \((0,1)\).

**Remark 3.** In Theorem 1, if the matrix \( C = 0 \), then for non-nil eigenvalues we have

\[
\lim_{\alpha \to 0} \lambda = 1.
\]

(13)

Figures 1, 2, and 3 are the eigenvalues plots of the preconditioned matrices obtained with our preconditioner. As we can see in the following numerical experiments, this good phenomenon is useful for accelerating convergence of iterative methods in Krylov subspace.

Additionally, for the purpose of practically executing our preconditioner

\[
H = \begin{pmatrix}
\left( A + \frac{B^T}{\alpha} \right)^{-1} & -A^{-1} \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1} \\ \frac{B}{\alpha} \left( A + \frac{B^T}{\alpha} \right)^{-1} & \frac{1}{\alpha} \left( I + \frac{BA^{-1}B^T}{\alpha} \right)^{-1}
\end{pmatrix},
\]

(14)
Figure 1: Spectrum of the preconditioned steady Oseen matrix with viscosity coefficient $\nu = 0.1$, 32 × 32 grid. (a) Q2-Q1 FEM, $C = 0$; (b) Q1-P0 FEM, $C \neq 0$.

Figure 2: Spectrum of the preconditioned steady Oseen matrix with viscosity coefficient $\nu = 0.01$, 32 × 32 grid. (a) Q2-Q1 FEM, $C = 0$; (b) Q1-P0 FEM, $C \neq 0$.

Figure 3: Spectrum of the preconditioned steady Oseen matrix with viscosity coefficient $\nu = 0.001$, 32 × 32 grid. (a) Q2-Q1 FEM, $C = 0$; (b) Q1-P0 FEM, $C \neq 0$.
Table 1: Preconditioned GMRES(20) on Stokes problems with different grid sizes (uniform grids).

<table>
<thead>
<tr>
<th>Grid</th>
<th>(\alpha)</th>
<th>its</th>
<th>LU time</th>
<th>its time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>0.0001</td>
<td>4</td>
<td>0.0457</td>
<td>0.0267</td>
<td>0.0724</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.0001</td>
<td>6</td>
<td>0.3912</td>
<td>0.0813</td>
<td>0.4725</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.0001</td>
<td>9</td>
<td>5.4472</td>
<td>0.5591</td>
<td>5.9991</td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.0001</td>
<td>14</td>
<td>91.4698</td>
<td>5.7732</td>
<td>97.2430</td>
</tr>
</tbody>
</table>

Table 2: Preconditioned GMRES(20) on steady Oseen problems with different grid sizes (uniform grids), viscosity \(\nu = 0.1\).

<table>
<thead>
<tr>
<th>Grid</th>
<th>(\alpha)</th>
<th>its</th>
<th>LU time</th>
<th>its time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>0.0001</td>
<td>3</td>
<td>0.0479</td>
<td>0.0244</td>
<td>0.0723</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.0001</td>
<td>3</td>
<td>0.6312</td>
<td>0.0696</td>
<td>0.7009</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.0001</td>
<td>4</td>
<td>13.2959</td>
<td>0.3811</td>
<td>13.6770</td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.0001</td>
<td>6</td>
<td>130.5463</td>
<td>3.7727</td>
<td>134.3190</td>
</tr>
</tbody>
</table>

Table 3: Preconditioned GMRES(20) on steady Oseen problems with different grid sizes (uniform grids), viscosity \(\nu = 0.01\).

<table>
<thead>
<tr>
<th>Grid</th>
<th>(\alpha)</th>
<th>its</th>
<th>LU time</th>
<th>its time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>0.0001</td>
<td>2</td>
<td>0.0442</td>
<td>0.0236</td>
<td>0.0678</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.0001</td>
<td>3</td>
<td>0.4160</td>
<td>0.0557</td>
<td>0.4717</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.0001</td>
<td>3</td>
<td>7.3645</td>
<td>0.2623</td>
<td>7.6268</td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.0001</td>
<td>4</td>
<td>169.4009</td>
<td>3.1709</td>
<td>172.5718</td>
</tr>
</tbody>
</table>

Table 4: Preconditioned GMRES(20) on steady Oseen problems with different grid sizes (uniform grids), viscosity \(\nu = 0.001\).

<table>
<thead>
<tr>
<th>Grid</th>
<th>(\alpha)</th>
<th>its</th>
<th>LU time</th>
<th>its time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>10000</td>
<td>5</td>
<td>0.0471</td>
<td>0.0272</td>
<td>0.0743</td>
</tr>
<tr>
<td>32 × 32</td>
<td>10000</td>
<td>7</td>
<td>0.3914</td>
<td>0.0906</td>
<td>0.4820</td>
</tr>
<tr>
<td>64 × 64</td>
<td>10000</td>
<td>9</td>
<td>5.6107</td>
<td>0.4145</td>
<td>6.0252</td>
</tr>
<tr>
<td>128 × 128</td>
<td>10000</td>
<td>14</td>
<td>92.7154</td>
<td>4.8908</td>
<td>97.6062</td>
</tr>
</tbody>
</table>

Table 7: Preconditioned GMRES(20) on steady Oseen problems with different grid sizes (uniform grids), viscosity \(\nu = 0.01\).

<table>
<thead>
<tr>
<th>Grid</th>
<th>(\alpha)</th>
<th>its</th>
<th>LU time</th>
<th>its time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>10000</td>
<td>3</td>
<td>0.0458</td>
<td>0.0223</td>
<td>0.0681</td>
</tr>
<tr>
<td>32 × 32</td>
<td>10000</td>
<td>5</td>
<td>0.5748</td>
<td>0.0670</td>
<td>0.6418</td>
</tr>
<tr>
<td>64 × 64</td>
<td>10000</td>
<td>5</td>
<td>12.2642</td>
<td>0.4179</td>
<td>12.6821</td>
</tr>
<tr>
<td>128 × 128</td>
<td>10000</td>
<td>7</td>
<td>128.2758</td>
<td>1.6275</td>
<td>129.9033</td>
</tr>
</tbody>
</table>
we should efficiently deal with the computation of \((I + (BA^{-1}B^T/\alpha))^{-1}\). This can been tackled by the well-known Sherman-Morrison-Woodbury formula:

\[
(A_1 + X_1A_2X_2^T)^{-1} = A_1^{-1} - A_1^{-1}X_1(A_2^{-1} + X_2^TA_1^{-1}X_1)^{-1}X_2^TA_1^{-1},
\] (15)

where \(A_1 \in \mathbb{R}^{n \times n}\), and \(A_2 \in \mathbb{R}^{r \times r}\) are invertible matrices, \(X_1 \in \mathbb{R}^{n \times r_1}\), and \(X_2 \in \mathbb{R}^{r \times r_1}\) are any matrices, and \(n, r_1\) are any positive integers.

From (15) we immediately get

\[
(I + \frac{BA^{-1}B^T}{\alpha})^{-1} = I - B(\alpha A + B^TB)^{-1}B^T.
\] (16)

In the following numerical examples we will always use (16) to compute \((I + (BA^{-1}B^T/\alpha))^{-1}\) in (14).

3. Numerical Examples

In this section, we give numerical experiments to illustrate the behavior of our preconditioner. The numerical experiments are done by using MATLAB 7.1. The linear systems are obtained by using finite element methods in the Stokes problems and steady Oseen problems, and they are respectively the cases of

(1) \(C = 0\), which is caused by using Q2-Q1 FEM;
(2) \(C \neq 0\), which is caused by using Q1-P0 FEM.

Furthermore, we compare our preconditioner with that of [9] in the case of general saddle point problems (namely, \(C = 0\)). For the general saddle point problem, [9] has presented the preconditioner

\[
\tilde{H} = \begin{pmatrix} A + tB^TB & 0 \\ -2B & \frac{I}{t} \end{pmatrix}^{-1}
\] (17)

with \(t\) as a parameter and has proved that when \(A\) is symmetric positive definite, the preconditioned matrix has an eigenvalue 1 with multiplicity at \(n\), and the remaining eigenvalues satisfy

\[
\lambda = \frac{t\sigma_i^2}{1 + t\sigma_i^2},
\] (18)

\[
\lim_{t \to \infty} \lambda = 1,
\] (19)

where \(\sigma_i, i = 1, 2, \ldots, m\), are \(m\) positive singular values of the matrix \(BA^{-1/2}\).

All these systems can be generalized by using IFISS software package [11] (this is a free package that can be downloaded from the site http://www.maths.manchester.ac.uk/~djs/ifiss/). We use restarted GMRES(20) as the Krylov subspace method, and we always take a zero initial guess. The stopping criterion is

\[
\frac{\|r_k\|_2}{\|r_0\|_2} \leq 10^{-6},
\] (20)

where \(r_k\) is the residual vector at the \(k\)th iteration.

In the whole course of computation, we always replace \((I + (BA^{-1}B^T/\alpha))^{-1}\) in (14) with (16) and use the LU factorization of \(A + (B^TB/\alpha)\) to tackle \((A + (B^TB/\alpha))^{-1}v\), where \(v\) is a corresponding vector in the iteration. Concretely, let \(A + (B^TB/\alpha) = LU\); then we complete the matrix-vector product \((A + (B^TB/\alpha))^{-1}v\) by \(U \setminus L \setminus v\) in MATLAB term.

In the following tables, the denotation norm \((C, fro)\) means the Frobenius form of the matrix \(C\). The total time is the sum of LU time and iterative time, and the LU time is the time to compute LU factorization of \(A + (B^TB/\alpha)\).

Case 1 (for our preconditioner). \(C = 0\) (using Q2-Q1 FEM in Stokes problems and steady Oseen problems with different viscosity coefficients. The results are in Tables 1, 2, 3, 4).

Case 1’ (for preconditioner of [9]). \(C = 0\) (using Q2-Q1 FEM in Stokes problems and steady Oseen problems with different viscosity coefficients. The results are in Tables 5, 6, 7, 8).

Case 2. \(C \neq 0\) (using Q1-P0 FEM in Stokes problems and steady Oseen problems with different viscosity coefficients. The results are in Tables 9, 10, 11, 12).

From Tables 1, 2, 3, 4, 5, 6, 7, and 8 we can see that these results are in agreement with the theoretical analyses (13) and (19), respectively. Additionally, comparing with the results in Tables 9, 10, 11, and 12, we find that, although the iterations used in Case 1 (either for the preconditioner of [9] or our preconditioner) are less than those in Case 2, the time spent by Case 1 is much more than that of Case 2. This is because the density of the coefficient matrix generalized by Q2-Q1 FEM is much larger than that generalized by Q1-P0 FEM. This can be partly illustrated by Tables 13 and 14, and the others can be illustrated similarly.
Table 14: Size and number of non-nil elements of the coefficient matrix generalized by using Q1-P0 FEM in steady Stokes problems.

<table>
<thead>
<tr>
<th>Grid</th>
<th>dim(A)</th>
<th>nnz(A)</th>
<th>dim(B)</th>
<th>nnz(B)</th>
<th>nnz(A + (1/α)B^TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>578 × 578</td>
<td>3826</td>
<td>256 × 578</td>
<td>1800</td>
<td>7076</td>
</tr>
<tr>
<td>32 × 32</td>
<td>2178 × 2178</td>
<td>16818</td>
<td>1024 × 2178</td>
<td>7688</td>
<td>31874</td>
</tr>
<tr>
<td>64 × 64</td>
<td>8450 × 8450</td>
<td>70450</td>
<td>4096 × 8450</td>
<td>31752</td>
<td>136582</td>
</tr>
<tr>
<td>128 × 128</td>
<td>33282 × 33282</td>
<td>288306</td>
<td>16384 × 33282</td>
<td>129032</td>
<td>567192</td>
</tr>
</tbody>
</table>

4. Conclusions

In this paper, we have introduced a splitting preconditioner for solving generalized saddle point systems. Theoretical analysis showed the modulus of eigenvalues of the preconditioned matrix would be located in interval (0, 1) when the parameter is big enough. Particularly when the submatrix $C = 0$, the eigenvalues will tend to 1 as the parameter $\alpha \to 0$. These performances are tested by some examples, and the results are in agreement with the theoretical analysis.

There are still some future works to be done: how to properly choose a parameter $\alpha$ so that the preconditioned matrix has better properties? How to further precondition submatrix $(I + (BA^{-1}B^T/\alpha))^{-1}((BA^{-1}B^T/\alpha) + (C/\alpha))$ to improve our preconditioner?

Acknowledgments

The authors would express their great thankfulness to the referees and the editor Professor P. N. Shivakumar for their helpful suggestions for revising this paper. The authors would like to thank H. C. Elman, A. Ramage, and D. J. Silvester for their free IFISS software package. This research is supported by Chinese Universities Specialized Research Fund for the Doctoral Program (20111085110020), Sichuan Province Sci. & Tech. Research Project (2012GZX0080), and the Fundamental Research Funds for the Central Universities.

References