New Preconditioners Based on the GPHSS Iterative Method for Solving Saddle Point Problems

Shiheng Wang and Jinmei Wang

ABSTRACT

A new preconditioner based on a generalized preconditioned Hermitian and skew-Hermitian splitting (GPHSS) method is provided for solving saddle point problems, together with a more efficient alternative. Numerical example is given to show this compared with other preconditioners.

INTRODUCTION

The augmented system

\[
\begin{pmatrix}
A & B \\
B^T & 0
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix} =
\begin{pmatrix}
p \\
q
\end{pmatrix},
\]

where \( A \in \mathbb{R}^{m \times m} \) is symmetric and positive definite, and \( B \in \mathbb{R}^{m \times n} \), is also known as a Karush-Kuhn-Tucker (KKT) system, or an equilibrium system, or a saddle point problem. By introducing two parameters instead of one into Bai, Golub and Pan's preconditioned Hermitian and skew-Hermitian splitting (PHSS) method [1], Pan and Wang [2] developed a generalized preconditioned Hermitian and skew-Hermitian splitting (GPHSS) method for saddle problem (1).

A new preconditioner is designed for the GPHSS method for solving such saddle point problems where the (1, 1) block \( A \) is a symmetric and positive definite M-matrix, which was presented to a conference ICAEMT’2015 by S. Wang and K.
Wang [3]. This paper further discusses an alternative preconditioner that is more efficient, and numerical example is followed to show the superiority.

**THE GPHSS METHOD**

The equation (1) can be rewritten as

\[
\begin{pmatrix}
A & B \\
-B^T & 0
\end{pmatrix}
\begin{pmatrix}
x \\ y
\end{pmatrix} =
\begin{pmatrix}
p \\ -q
\end{pmatrix}.
\]

With two relaxation factors \(\omega > 0\) and \(\tau > 0\), compared with one parameter \(\alpha > 0\) in PHSS method [1], Pan and Wang [2] proposed:

**THE GPHSS ALGORITHM**

\[
x^{(k+\frac{1}{2})} = \frac{\omega}{1 + \omega} x^{(k)} + \frac{1}{1 + \omega} A^{-1} (p - By^{(k)}),
\]

\[
y^{(k+\frac{1}{2})} = y^{(k)} + \frac{1}{\tau} Q^{-1} (B^T x^{(k)} - q),
\]

\[
y^{(k+1)} = \tau D^{-1} Q y^{(k+\frac{1}{2})} + D^{-1} ((1 - \frac{1}{\omega})B^T x^{(k+\frac{1}{2})} + \frac{1}{\omega} B^T A^{-1} p - q),
\]

\[
x^{(k+1)} = \frac{\omega - 1}{\omega} x^{(k+\frac{1}{2})} + \frac{1}{\omega} A^{-1} (p - By^{(k+1)}),
\]

where \(Q \in \mathbb{R}^{n \times n}\) is nonsingular and symmetric, the preconditioning parameter matrix, and \(D = \omega^{-1} B^T A^{-1} B + \tau Q\). Taking \(\omega = \tau\) yields the PHSS algorithm.

**NEW PRECONDITIONERS**

In [3], the authors presented a new preconditioner \(Q = B^T S B\) for the GPHSS algorithm with

\[
S = (s_{ij}) = \begin{cases}
1, \\
\frac{1}{a_{ii}}, \\
\frac{1}{a_{jj}}, \\
\frac{1}{a_{kk}}, \\
\frac{1}{a_{mn}}
\end{cases} + \text{diag}(1/a_{11}, 1/a_{22}, \ldots, 1/a_{nn}),
\]

478
\[
\widetilde{a} = \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij}
\]

where \( \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} \) and \( A = (a_{ij}) \) is a positive definite M-matrix such as

\[
\begin{align*}
    a_{ii} > 0, & \quad i = 1, \cdots, m, \\
    a_{ij} = a_{ji} < 0, & \quad i \neq j, \\
    \sum_{k=1}^{m} a_{ik} > 0, & \quad i = 1, \cdots, m.
\end{align*}
\] (2)

Notice that \( S \) has positive diagonal entries, then taking

\[
\hat{S} = \begin{pmatrix}
    s_{11} & & \\
    & \ddots & \\
    & & s_{mm}
\end{pmatrix},
\] (3)

Where the off-diagonal entries are zeros, yields an alternative preconditioner \( B^\top \hat{S}B \).

The following convergence theorem is obtained.

**Theorem.** If \( A = (a_{ij})_{m \times m} \) in (1) satisfies (2) and \( B \) is of full column rank, taking \( \hat{S} \) as (3) and \( Q = B^\top \hat{S}B \), then \( Q \) is symmetric and positive definite, thus the GPHSS method is convergent with \( \omega > 0 \) and \( \tau > 0 \).

**Proof.** Obviously, \( \hat{S} \) is symmetric and positive definite, thus, \( Q \) is positive definite. By Theorems 1 and 2 in [2], the convergence follows.

**Remark.** If \( \widetilde{a} \) is greater than one, then the off-diagonal entries of \( S \) are less than one. Therefore, \( \hat{S} \) is an approximation of \( S \). As \( \hat{S} \) is much simpler than \( S \), it can be expected that the GPHSS method with \( Q = B^\top \hat{S}B \) would be better than with \( Q = B^\top SB \).

**NUMERICAL EXAMPLE**

In this section, an example is provided to illustrate the GPHSS method with the new preconditioners. The initial guess is 0 and the stopping criterion is
where \( r^{(k)} \) is the residual vector after \( k \) iterations, and \( \omega = \omega^* \), \( \tau = \tau^* \) (\( \omega^* \) and \( \tau^* \) are the optimal parameters). The results are listed in Table I compared with \( Q = B^\top B \).

Example. Consider \((m + n) \times (m + n)\) augmented system (1) with

\[
A = (a_{ij})_{m \times m} = \begin{cases} 
  a_{ij} = i + j, & i = j, \\
  a_{ij} = -\frac{1}{m}, & i \neq j, 
\end{cases} \quad 1 \leq i, j \leq m,
\]

\( B = \text{eye}(m, n), \ p = (1,1,\cdots,1)^\top \) and \( q = (0,0,\cdots,0)^\top \).

<table>
<thead>
<tr>
<th>( m )</th>
<th>( n )</th>
<th>( Q = B^\top B )</th>
<th>( Q = B^\top SB )</th>
<th>( Q = B^\top \hat{S}B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>( t )</td>
<td>ERR</td>
<td>IT</td>
<td>( t )</td>
</tr>
<tr>
<td>200</td>
<td>150</td>
<td>25</td>
<td>0.37</td>
<td>7.8e-7</td>
</tr>
<tr>
<td>400</td>
<td>300</td>
<td>30</td>
<td>2.37</td>
<td>7.3e-7</td>
</tr>
<tr>
<td>500</td>
<td>400</td>
<td>32</td>
<td>4.90</td>
<td>8.6e-7</td>
</tr>
<tr>
<td>700</td>
<td>500</td>
<td>34</td>
<td>11.36</td>
<td>7.7e-7</td>
</tr>
<tr>
<td>1000</td>
<td>700</td>
<td>37</td>
<td>32.40</td>
<td>7.8e-7</td>
</tr>
</tbody>
</table>

**CONCLUSIONS**

In this paper, a new preconditioner \( Q = B^\top SB \) and an alternative \( Q = B^\top \hat{S}B \) are presented to improve the GPHSS method proposed by Pan and Wang [2] for saddle point problem (1), where \( A \) is a symmetric positive definite M-matrix as (2). With the new preconditioners, computing \( A^{-1} \) can be avoided. Numerical results show that the two new preconditioners outperform the common one \( Q = B^\top B \) and the number of iterations is almost constant.
REFERENCES

