The Preconditioner of GPHSS Method for Saddle Point Systems

Shiheng Wang[1] and Ke Wang[2,a *]

[1]Department of Automotive Engineering, Nanyang Vocational College of Agriculture, Nanyang 473000, P.R. China

[2]Department of Mathematics, College of Sciences, Shanghai 200444, P.R. China

[a]kwang@shu.edu.cn

*corresponding author

Keywords: Saddle point problem, GHPSS method, Preconditioner

Abstract. Pan and Wang presented a generalized preconditioned Hermitian and skew-Hermitian splitting (GPHSS) method [J. Numer. Methods Comput. Appl., **32.**, 174-182, 2011] for saddle point problems. The method is improved to solve saddle point systems whose (1,1) block is a symmetric positive definite *M*-matrix with a new choice of the preconditioner and compared with other preconditioners. The results show that the new preconditioner outperforms the previous ones.

Introduction

For the saddle point problem (augmented system)

$$\begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix}, \tag{1}$$

where $A \in R^{m\times m}$ is a symmetric positive definite matrix, $B \in R^{m\times n}$ $(m > n)$ is of full column rank, $p \in R^m$ and $q \in R^n$, there are many kinds of iterative methods such as Uzawa [3], SOR-like [4], GSI [6], GAOR [8], HSS [1] and PHSS [2]. Based on the PHSS method, Pan and Wang [7] suggested a GPHSS method with two parameters. In this paper, we further study the GPHSS method to improve it for solving such saddle point systems whose (1,1) block is a symmetric positive definite *M*-matrix with a new choice of the preconditioner and compare it with other preconditioners.

The rest of the paper is organized as follows. In Section 2, the GPHSS method is briefly introduced. In Section 3, we discuss the new choice of the preconditioner. In Section 4, we give numerical examples to show the improvement is efficient. The conclusions are presented in Section 5.

GPHSS Method for Saddle Point Problem

In this section, we briefly introduce the GPHSS method for saddle point problems.

System (1) can be rewritten as [4]

$$\mathcal{A}\, z = b \tag{2}$$

where

$$\mathcal{A} = \begin{pmatrix} A & B \\ -B^{\mathrm{T}} & 0 \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, b = \begin{pmatrix} p \\ -q \end{pmatrix}.$$

Define matrix $P = \begin{pmatrix} A & 0 \\ 0 & Q \end{pmatrix}$, where $Q \in R^{n\times n}$ is nonsingular and symmetric, referred to the preconditioning parameter matrix.

Given the initial vectors $x^{(0)} \in R^m$, $y^{(0)} \in R^m$, as well as the relaxation factors $\omega > 0$ and $\tau > 0$, for $k = 0, 1, 2, \mathbf{L}$, till the iteration sequence $(x^{(k)^{\mathrm{T}}}, y^{(k)^{\mathrm{T}}})^{\mathrm{T}}$ converges, Pan and Wang [7] proposed the GPHSS algorithm as followed:

$$x^{(k+\frac{1}{2})} = \tfrac{w}{1+w} x^{(k)} + \tfrac{1}{1+w} A^{-1}(p - By^{(k)}),$$
$$y^{(k+\frac{1}{2})} = y^{(k)} + \tfrac{1}{t} Q^{-1}(B^{\mathrm{T}} x^{(k)} - q),$$
$$y^{(k+1)} = tD^{-1}Qy^{(k+\frac{1}{2})} + D^{-1}((1-\tfrac{1}{w})B^{\mathrm{T}} x^{(k+\frac{1}{2})} + \tfrac{1}{w} B^{\mathrm{T}} A^{-1} p - q),$$
$$x^{(k+1)} = \tfrac{w-1}{w} x^{(k+\frac{1}{2})} + \tfrac{1}{w} A^{-1}(p - By^{(k+1)}),$$

where $D = w^{-1} B^{\mathrm{T}} A^{-1} B + tQ \in R^{n\times n}$. When $\omega = \tau$, GPHSS algorithm is the PHSS algorithm.

Denote $H = \frac{1}{2}(A + A^{\mathrm{T}})$, $S = \frac{1}{2}(A - A^{\mathrm{T}})$, then the matrix form of the GPHSS is as follows:

$$\begin{cases} (\Omega P + H)\begin{pmatrix} x^{(k+\frac{1}{2})} \\ y^{(k+\frac{1}{2})} \end{pmatrix} = (\Omega P - S)\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + b, \\ (\Omega P + S)\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = (\Omega P - H)\begin{pmatrix} x^{(k+\frac{1}{2})} \\ y^{(k+\frac{1}{2})} \end{pmatrix} + b, \end{cases} \tag{3}$$

where

$$\Omega = \begin{pmatrix} wI_m & 0 \\ 0 & tI_n \end{pmatrix},$$

and I_m and I_n are $m\times m$ and $n\times n$ identity matrices. The iterative matrix is

$$M_{(w,t)} = (\Omega P + S)^{-1}(\Omega P - H)(\Omega P + H)^{-1}(\Omega P - S).$$

New Choice of the Preconditioner

Consider $m \times m$ symmetric matrices $T = (t_{ij})$ with negative off-diagonal elements and positive row (and column) sums, i.e.,

$$\begin{cases} t_{ii} > 0, & i = 1, \ldots, m, \\ t_{ij} = t_{ji} < 0, & i \neq j, \\ \sum_{k=1}^{m} t_{ik} > 0, & i = 1, \ldots, m. \end{cases} \tag{4}$$

Such matrices are positive definite and thus are *M*-matrices.

Let $S = (s_{ij})$ be the $m \times m$ matrix defined in [5] as $s_{ij} = \frac{d_{ij}}{t_{ij}} + \frac{1}{\bar{t}}$, where δ_{ij} is the Kronecker delta function $d_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$ and $\bar{t} = \sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij}$ with t_{ij} as in (4). From [5], we see that the matrix S is symmetric positive definite and S has the form

$$S = \frac{1}{\bar{t}}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \cdot (1, 1, \ldots, 1) + \mathrm{diag}(1/t_{11}, 1/t_{22}, \ldots, 1/t_{mm}). \tag{5}$$

For augmented system (1) with matrix A satisfying (4), we can take $T = A$ and the preconditioning parameter matrix $Q = B^{\mathrm{T}}SB$ in the GPHSS method. We have the following results.

Theorem *If $A = (a_{ij})_{m\times m}$ in* (1) *satisfies* (4)*, taking $T = A$, S as* (5) *and $Q = B^{\mathrm{T}}SB$, then Q is symmetric and positive definite, thus the GPHSS method* (3) *is convergent with $\omega > 0$ and $\tau > 0$.*

Proof The results can directly obtained from Theorems 1 and 2 in [7]. □

Numerical Experiments

In this section, we give three examples to illustrate the GPHSS method with the new choice of the preconditioner. All performances are taken in MATLAB 7.12 with 2.70GHz CPU, 4.00GB RAM and Windows 7 Professional. In our experiments, the initial guess is 0 and the stopping criterion is

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} < 10^{-6},$$

where $r^{(k)}$ is the residual vector after k iterations, $Q = B^{\mathrm{T}}B$ and the parameters $\omega = \omega^*$, $\tau = \tau^*$. The results are listed in Tables 1-3.

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

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

$B = \mathrm{eye}(m, n)$, $b = (1, 1, \ldots, 1)^{\mathrm{T}}$ and $q = (0, 0, \ldots, 0)^{\mathrm{T}}$.

Table 1

Iterations (IT), CPU time (t) and relative error (ERR) for Example 1

m	n	$Q = B^{\mathrm{T}}B$			$Q = \frac{1}{2}I$			$Q = B^{\mathrm{T}}SB$		
		IT	t	ERR	IT	t	ERR	IT	t	ERR
50	40	21	0.04	6.3e-7	21	0.03	6.3e-7	8	0.00	1.5e-7
200	150	30	0.44	7.8e-7	30	0.42	7.8e-7	8	0.14	1.4e-7
400	300	36	3.20	9.5e-7	36	3.17	9.5e-7	8	0.72	1.9e-7
500	400	39	6.19	9.5e-7	39	5.97	9.5e-7	8	1.29	4.4e-7
700	500	42	14.91	7.8e-7	42	14.13	7.8e-7	8	2.91	1.4e-7
1000	700	46	42.45	8.2e-7	46	41.67	8.2e-7	8	7.50	1.3e-7

Example 2 Consider $(m + n) \times (m + n)$ augmented system (1) with

$$A = (a_{ij})_{m\times m} = \begin{cases} a_{ij} = -\frac{1}{2} - \frac{i}{2m}, & i < j, \\ a_{ij} = a_{ji}, & i > j, \\ a_{ij} = -\sum a_{ik} + 1 + \frac{i}{m}, & i = j, \end{cases} \quad 1 \le i, j \le m,$$

$B = (1/2)*\mathrm{eye}(m, n)$, $b = (1, 0, \ldots, 0)^{\mathrm{T}}$ and $q = (1, 0, \ldots, 0)^{\mathrm{T}}$.

Table 2

Iterations (IT), CPU time (t) and relative error (ERR) for Example 1

m	n	$Q = B^{\mathrm{T}}B$			$Q = \frac{1}{2}I$			$Q = B^{\mathrm{T}}SB$		
		IT	t	ERR	IT	t	ERR	IT	t	ERR
50	40	18	0.03	7.1e-7	18	0.03	6.3e-7	5	0.00	1.6e-8
200	150	29	0.44	4.7e-7	29	0.45	7.8e-7	4	0.08	7.6e-7
400	300	35	2.79	7.3e-7	35	2.99	9.5e-7	4	0.44	6.9e-7
500	400	38	5.85	5.8e-7	38	6.05	9.5e-7	5	0.87	6.6e-9
700	500	41	13.51	5.1e-7	41	14.06	7.8e-7	5	1.89	5.4e-9
1000	700	45	41.25	5.5e-7	45	42.49	8.2e-7	5	4.82	1.1e-8

Example 3 Consider $(m + n) \times (m + n)$ augmented system (1) with

$$A=(a_{ij})_{m\times m}=\begin{cases} a_{ij}=i+1, & i=j, \\ a_{ij}=1, & |i-j|=1, \\ a_{ij}=0, & \text{others}, \end{cases} \quad 1\le i,j\le m,$$

$$B=(b_{ij})_{m\times m}=\begin{cases} b_{ij}=j, & i=j+m-n, \\ b_{ij}=0, & \text{others}, \end{cases} \quad 1\le i\le m, 1\le j\le n,$$

$b=(1, 0, \ldots, 0)^{\mathrm{T}}$ and $q=(1, 0, \ldots, 0)^{\mathrm{T}}$.

Table 3
Iterations (IT), CPU time (t) and relative error (ERR) for Example 1

m	n	$Q=B^{\mathrm{T}}B$			$Q=\frac{1}{2}I$			$Q=B^{\mathrm{T}}SB$		
		IT	t	ERR	IT	t	ERR	IT	t	ERR
50	40	12	0.03	3.0e-7	41	0.06	7.6e-7	8	0.00	1.6e-8
200	150	12	0.22	3.2e-7	88	1.72	9.5e-7	8	0.12	7.6e-7
400	300	12	1.44	7.6e-7	129	14.99	8.8e-7	8	0.66	6.9e-7
500	400	13	2.94	1.0e-7	141	29.09	8.7e-7	8	1.31	6.6e-9
700	500	12	5.12	5.2e-7	177	72.29	9.6e-7	9	3.39	5.4e-9
1000	700	12	12.39	5.2e-7	216	221.60	9.6e-7	9	9.55	1.1e-8

Tables 1, 2 and 3 give the number of iterations required for convergence, CPU time and relative error, which show that the new choice of the preconditioner $B^{\mathrm{T}}SB$ is much better than $B^{\mathrm{T}}B$ and αI with the optimal parameters ω^* and τ^*.

Conclusion

In this paper, a new choice of the preconditioner $B^{\mathrm{T}}SB$ is presentedto accelerate the GPHSS method suggested by Pan and Wang [7] for solving saddle point problems with A is a positive definite M-matrix satisfying (4). With the new choice, we can avoid computing A^{-1} and obtain a good approximation of the choice $B^{\mathrm{T}}A^{-1}B$. Thus, the method with the new choice is very fast. Numerical experiments show that the new choice of the parameter matrix are better than $B^{\mathrm{T}}B$ and αI which are common choices for solving saddle point problems and the number of iterations is almost constant.

Acknowledgments

The second author was supported by National Natural Science Foundation of China (11301330) and the grant "The First-class Discipline of Universities in Shanghai".

References

[1] 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 (2003) 603-626.

[2] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math. 98 (2004) 1-32.

[3] Y. Cao, Y. Lin, Y. Wei, Nonlinear Uzawa methods for solving nonsymmetric saddle point problems, J. Appl. Math. Comput. 21 (2006) 1-21.

[4] G.H. Golub, X. Wu, J.-Y. Yuan. SOR-like methods for augmented systems, BIT 41 (2001) 71-85.

[5] X.-Q. Jin, *M*-preconditioner for *M*-matrices, Appl. Math. Comput. 172 (2006) 701-707.

[6] S.-X. Miao, K. Wang, On generalized stationary iterative method for solving the saddle point problems, J. Appl. Math. Comput. 35 (2011) 459-468.

[7] C.P. Pan, H.Y. Wang, On generalized preconditioned Hermitian and skew-Hermitian splitting methods for saddle point problems (in Chinese), J. Numer. Methods Comput. Appl. 32 (2011) 174-182.

[8] X. Shao, H. Shen, C. Li, T. Zhang, Generalized AOR method for augmented systems (in Chinese), J. Numer. Methods Comput. Appl. 27 (2006) 241-248.