Chiang Mai J. Sci. 2012; 39(2) : 213-221
http://it.science.cmu.ac.th/ejournal/
Contributed Paper

A Preconditioner for the SOR-like Method for the Augmented Systems

Davod Khojasteh Salkuyeh* and Somayyeh Shamsi**

*Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran.

**Young Researchers Club, Ardabil branch, Islamic Azad University, Ardabil, Iran.

*Author for correspondence; e-mail: *khojasteh@guilan.ac.ir or salkuyeh@gmail.com
**shams.math81@yahoo.com

Received: 10 November 2009
Accepted: 2 September 2011

ABSTRACT

Golub et al. [Golub G.H., Wu X. and Yuan J.Y., SOR-like methods for augmented systems, *BIT*, 2001; 41: 71-85] have presented the SOR-like algorithm to solve augmented systems. In this note, we propose a simple but effective preconditioner for the SOR-like method. Some numerical results are given to show the efficiency of the proposed preconditioned SOR-like method.

Keywords: augmented system, symmetric positive definite, SOR-like, preconditioner.

1. INTRODUCTION

Consider the augmented system

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ -q \end{pmatrix}, \qquad (1.1)$$

where $A \in \Re^{m\times m}$ is symmetric positive definite, $B \in \Re^{m\times m}$, $m \geq n$ is of column full rank. These assumptions guarantee that the system (1.1) has a unique solution. This system is important as it arises in many applications such as the finite element discretization of the Stokes equations for steady flow of a viscous fluid [1], constrained optimization [2] and weighted least-squares problems [3].

Since the matrices A and B are usually large and sparse, iterative methods become more attractive than direct methods for solving problem (1.1). Several iterative methods to solve Eq. (1.1) have been presented in the literature. The successive overrelaxation (SOR) [4] is a stationary iterative method which is popular in science and engineering applications. In [5], Golub et al. have presented several SOR-like algorithms to solve augmented systems (1.1). Then, several papers have been written by the authors to improve, modify or generalize the SOR-like methods [6, 7, 8]. In this paper, we propose a simple but very effective preconditioner for the augmented system (1.1).

This paper is organized as follows. In section 2, the SOR-like method is reviewed. Section 3 is devoted to the proposed preconditioner. Some numerical results are given to show the efficiency of the proposed preconditioner. Concluding remarks are given in section 5.

2. A BRIEF DESCRIPTION OF THE SOR-LIKE METHOD

Let
$$\mathbf{A} = \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} = \mathbf{D} - \mathbf{L} - \mathbf{U}, \tag{2.1}$$

where
$$\mathbf{D} = \begin{pmatrix} A & 0 \\ 0 & Q \end{pmatrix}, \mathbf{L} = \begin{pmatrix} 0 & 0 \\ B^T & 0 \end{pmatrix}, \mathbf{U} = \begin{pmatrix} 0 & -B \\ 0 & Q \end{pmatrix},$$

in which $Q \in \Re^{n\times n}$ is a nonsingular symmetric matrix. Now, the SOR-like method [5] is written as following
following
$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \mathbf{M}_\omega \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \omega(\mathbf{D} - \omega \mathbf{L})^{-1} \begin{pmatrix} b \\ q \end{pmatrix}, \tag{2.2}$$

where
$$\mathbf{M}_\omega = (\mathbf{D} - \omega \mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega \mathbf{U}]$$
$$= \begin{pmatrix} A & 0 \\ -\omega B^T & Q \end{pmatrix}^{-1} \begin{pmatrix} (1-\omega)A & -\omega B \\ 0 & Q \end{pmatrix}.$$

By some simple manipulation one can see that the SOR-like iteration takes the following form.

Algorithm 2.1: ***The SOR-like method***

1. Choose the nonsingular symmetric matrix Q and initial guesses $x^{(0)}$ and $y^{(0)}$, and a relaxation parameter $\omega > 0$.
2. For $k = 0,1,2,\ldots$, until convergence, Do:
   $$x^{(k+1)} := (1-\omega)x^{(k)} + \omega A^{-1}(b - By^{(k)})$$
   $$y^{(k+1)} := y^{(k)} + \omega Q^{-1}(B^T x^{(k+1)} - q).$$
3. EndDo

The next theorem gives sufficient conditions for the convergence of the SOR-like method.

Theorem 2.1. [5, Theorem 2.3] *Suppose that B has full rank and A is symmetric positive definite. Assume that all eigenvalues* μ *of* $Q^{-1}B^TA^{-1}B$ *are real. If* $\mu > 0$, *then the SOR-like method is convergent for all* ω *such that*
$$0 < \omega < \frac{4}{1+\sqrt{4\rho+1}},$$
where ρ *is the spectral radius of* $Q^{-1}B^TA^{-1}B$.

For simplicity of notation, denote $\rho = \rho(Q^{-1}B^TA^{-1}B)$ and $0 < \mu_0 = \min_{\mu \neq 0} \mu$, where μ is a nonzero eigenvalue of $Q^{-1}B^TA^{-1}B$. Now, the next theorem gives the optimal relaxation parameter for the SOR-like method.

Theorem 2.2. [5, Theorems 3.1] *If* $\mu_0 > 1/4$, *then*

$$\rho(\mathbf{M}_\omega) = \begin{cases} \sqrt{1-\omega}, & \text{if } 0 < \omega \le \dfrac{2\sqrt{\rho}-1}{\rho}, \\ 0.5\left[\left|2-\omega-\omega^2\rho\right| + \omega\sqrt{(\omega\rho+1)^2 - 4\rho}\right], & \text{if } \dfrac{2\sqrt{\rho}-1}{\rho} \le \omega < \dfrac{4}{1+\sqrt{4\rho+1}}. \end{cases}$$

Moreover, the optimal parameter ω_{0pt} *and* $\rho(\mathrm{M}_{\omega_{0pt}})$ *are given respectively by*

$$\omega_{opt} = \frac{2\sqrt{\rho-1}}{\rho} \le 1,$$

and

$$\rho(\mathbf{M}_{\omega_{opt}}) = \frac{\left|\sqrt{\rho}-1\right|}{\sqrt{\rho}}.$$

Based on the theoretical results presented in [5], $Q = B^T A^{-1} B$ is the best choice for the matrix Q and the SOR-like method converges in one step. In practice Q is chosen an approximation of $B^T A^{-1} B$.

3. A PRECONDITIONER FOR THE SOR-LIKE METHOD

Since A is symmetric positive definite, its Cholesky factorization exists, i.e., there exists a lower triangular matrix L with positive diagonal entries such that $A=LL^T$ [9]. Let $P=\overline{L}\,\overline{L}^T$ be an incomplete cholesky factorization of A where $\overline{L}$ is a sparse approximation of L [10].

Now, we defined a preconditioner for (1.1) as following

$$\mathbf{P} = \begin{pmatrix} \overline{L}^{-1} & 0 \\ 0 & 1 \end{pmatrix},$$

where I is the identity matrix of order n. Now, system (1.1) is transformed into

$$\mathbf{P}\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\mathbf{P}^T\mathbf{P}^{-T}\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{P}\begin{pmatrix} b \\ -q \end{pmatrix}. \tag{3.1}$$

Obviously, the solution of this system is the same as that of the system (1.1). It is easy to see that system (3.1) is equivalent to

$$\begin{pmatrix} \overline{L}^{-1}A\overline{L}^{-T} & \overline{L}^{-1}B \\ -B^T\overline{L}^{-T} & 0 \end{pmatrix}\begin{pmatrix} \overline{L}^T x \\ y \end{pmatrix} = \begin{pmatrix} \overline{L}^{-1}b \\ -q \end{pmatrix}. \tag{3.2}$$

For simplicity of notation, let

$$\overline{A} = \overline{L}^{-1}A\overline{L}^{-T}, \quad \overline{B} = \overline{L}^{-1}B, \quad \overline{x} = \overline{L}^T x \ \text{ and } \ \overline{b} = \overline{L}^{-1}b. \tag{3.3}$$

Hence, system (3.2) is equivalent to

$$\begin{pmatrix} \overline{A} & \overline{B} \\ -\overline{B}^T & 0 \end{pmatrix}\begin{pmatrix} \overline{x} \\ y \end{pmatrix} = \begin{pmatrix} \overline{b} \\ -q \end{pmatrix}. \tag{3.4}$$

It is clear that $\overline{A}$ is symmetric positive definite and $\overline{B}$ is of full rank. Moreover, the (2, 2) block of the coefficient matrix is zero. Therefore, each iteration of the SOR-like method for solving (3.4) takes the following form

$$\overline{x}^{(k+1)} := (1-\omega)\overline{x}^{(k)} + \omega\overline{A}^{-1}(\overline{b} - \overline{B}y^{(k)}),$$
$$y^{(k+1)} := y^{(k)} + \omega\overline{Q}^{-1}(\overline{B}^{T}\overline{x}^{(k+1)} - q),$$

where $\overline{Q}$ is an approximation of $\overline{B}^{T}\overline{A}^{-1}\overline{B}$. Substituting $\overline{A}, \overline{B}, \overline{b}$ and $\overline{x}^{(k+1)} = \overline{L}^{T}x^{(k+1)}$ from (3.2) into (3.3), we obtain

$$x^{(k+1)} := (1-\omega)x^{(k)} + \omega A^{-1}(b - By^{(k)}),$$
$$y^{(k+1)} := y^{(k)} + \omega\overline{Q}^{-1}(B^{T}x^{(k+1)} - q).$$

As we observe the only difference between this method and the SOR-like method is the way of choosing preconditioning matrices Q and $\overline{Q}$. Therefore, the SOR-like for system (3.4), can be summarized in the following form which we call it PSOR-like (for preconditioned SOR-like) method.

Algorithm 3.1: ***The PSOR-like method***

1. Compute a sparse approximate Cholesky factor $\overline{L}$ for A.
2. Choose the nonsingular symmetric matrix $\overline{Q}$ which approximates $\overline{B}^{T}\overline{A}^{-1}\overline{B}$ and initial guesses $x^{(0)}$ and $y^{(0)}$, and a relaxation parameter $\omega > 0$.
3. For $k=0,1,2,\ldots,$ until convergence, Do:

$$x^{(k+1)} := (1-\omega)x^{(k)} + \omega A^{-1}(b - By^{(k)}),$$
$$y^{(k+1)} := y^{(k)} + \omega\overline{Q}^{-1}(B^{T}x^{(k+1)} - q).$$

4. EndDo

Similar to the SOR-like method we can state the following theorem for the PSOR-like method.

Theorem 3.1 *Suppose that B has full rank and A is symmetric positive definite. Assume that all eigenvalues* μ *of* $\overline{Q}^{-1}B^{T}A^{-1}B$ *are real. If* $\mu > 0$*, then the PSOR-like method is convergent for all* ω *such that*

$$0 < \omega < \frac{4}{1+\sqrt{4\rho+1}},$$

where ρ is the spectral radius of $\overline{Q}^{-1}B^{T}A^{-1}B$.

Proof. It is easy to see that

$$\overline{B}^{T}\overline{A}^{-1}\overline{B} = B^{T}AB.$$

Therefore, the theorem immediately follows from Theorem 2.1. □

In the PSOR-like method, let $\overline{\rho} = \rho(\overline{Q}^{-1} B^T A^{-1} B)$ and $0 < \overline{\mu}_0 = \min_{\mu \neq 0} \mu$ where μ is a nonzero eigenvalue of $\overline{Q}^{-1} B^T A^{-1} B$. Now, by replacing ρ by $\overline{\rho}$ in Theorem 2.2, it is also valid for the PSOR-like method.

4. NUMERICAL EXPERIMENTS

In this section, some numerical experiments are given to compare the SOR-like method with the PSOR-like method. All the numerical experiments presented in this section were computed in double precision using a MATLAB code on a Pentium 4 PC, with a 3.06 GHz CPU and 1.00GB of RAM. The initial guess was always a zero vector and the right hand side $(b^T, q^T)^T$ was selected such that the exact solution of the augmented system (1.1) is $(x^{(*)^T}, y^{(*)^T})^T = (1,1,...,1)^T$. The stopping criterion

criterion

$$\frac{\| r_k \|_2}{\| r_0 \|_2} < 10^{-12} ,$$

was used in the computations, where

$$r_k = \begin{pmatrix} b \\ -q \end{pmatrix} - \mathbf{A} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}$$

The preconditioning matrix Q and $\overline{Q}$, which are approximations of $B^T A^{-1} B$ and $\overline{B}^T \overline{A}^{-1} \overline{B}$, respectively, are chosen according to the cases listed in Table 4.1.

Table 4.1 Choices of matrices Q and $\overline{Q}$.

Method	Case no.	Matrix Q	Description
SOR-like	I	$B^T \hat{A}^{-1} B$	$\hat{A} = \text{tridiag}(A)$
	II	$B^T \hat{A}^{-1} B$	$\hat{A} = \text{diag}(A)$
PSOR-like	I	$\overline{B}^T \hat{A}^{-1} \overline{B}$	$\hat{A} = \text{tridiag}(A)$
	II	$\overline{B}^T \hat{A}^{-1} \overline{B}$	$\hat{A} = \text{diag}(A)$

We consider following two examples for our numerical experiments as follows.

Example 4.1 Consider the augmented linear system (1.1) with [11].

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \Re^{2p^2 \times 2p^2},$$

$$B = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \Re^{2p^2 \times p^2},$$

and

$$T = \frac{1}{h^2}\, tridiag\,(-1,2,-1) \in \Re^{p \times p}, \qquad F = \frac{1}{h}\, tridiag\,(-1,1,0) \in \Re^{p \times p},$$

where ⊗ is the Kronecker product symbol and $h=1/(p+1)$ is the discretization mesh-size. Let $m=2p^2$ and $n=p^2$. Hence, the total number of variables is $m+n=3p^2$. In Table 4.2, we list the minimum positive eigenvalues μ_0 and $\bar{\mu}_0$ of $Q^{-1}B^TA^{-1}B$ and $\bar{Q}^{-1}B^TA^{-1}B$, respectively, for different values of m and n. This table shows that μ_0 and $\bar{\mu}_0$ are greater than 0.25. Therefore, we can use Theorem 2.2 to compute the optimal relaxation parameter. In Table 4.3, we list ω_{opt} and spectral radius of the matrix iteration of the SOR-like and the PSOR-like methods denoted by $\rho(M_{\omega_{opt}})$ and $\rho(G_{\omega_{opt}})$, respectively. Also, in this table ITS and Time stand for the number of iterations for convergence and the CPU time, respectively. Here we mention that the CPU time for the PSOR-like method includes the time for computing the preconditioner. We also mention that the MATLAB function cholinc with drop tolerance $\tau = 0.01$ has been used to compute the incomplete Cholesky factorization of A.

Numerical results presented in Table 4.3 shows that the PSOR-like method with optimal parameter always converge faster than the SOR-like method with optimal parameter. Moreover, the CPU time for computing the approximate solution by the PSOR-like method is always less than that of the SOR-like method.

Table 4.2 The values of μ_0 and $\bar{\mu}_0$ for Example 4.1.

m		**128**	**512**	**1152**
n		**64**	**256**	**576**
m + n		**192**	**768**	**1728**
μ_0	Case I	0.5319	0.5088	0.5040
	Case II	0.5162	0.5044	0.5020
$\bar{\mu}_0$	Case I	0.7193	0.6762	0.6670
	Case II	0.6515	0.6070	0.5976

Table 4.3 Numerical results for Example 4.1.

m			**128**	**512**	**1152**
n			**64**	**256**	**576**
$m+n$			**192**	**768**	**1728**
Case I	SOR-like	ω_{opt}	0.5958	0.3657	0.2620
		$\rho(M_{\omega_{opt}})$	0.6358	0.7964	0.8591
		ITS	72	144	218
		Time	0.08	1.55	10.69
	PSOR-like	ω_{opt}	0.9762	0.8871	0.7662
		$\rho(M_{\omega_{opt}})$	0.1542	0.3361	0.4835
		ITS	19	28	42
		Time	0.03	0.91	7.17
Case II	SOR-like	ω_{opt}	0.4664	0.2720	0.1915
		$\rho(M_{\omega_{opt}})$	0.7305	0.8533	0.8992
		ITS	105	211	318
		Time	0.03	1.89	12.52
	PSOR-like	ω_{opt}	0.9695	0.8807	0.7617
		$\rho(M_{\omega_{opt}})$	0.1745	0.3454	0.4882
		ITS	23	29	42
		Time	0.02	0.84	7.09

Example 4.2 In this example the matrix $A \in \Re^{2p^2 \times 2p^2}$ is the same as the previous example. The matrix $B = (b_{ij}) \in \Re^{2p^2 \times p^2}$ is chosen such a way that

$$b_{ij} = \begin{cases} j, & i = j + m - n, \\ 0, & otherwise \end{cases}$$

in which $m = 2p^2$ and $n = p^2$. As the previous example the total number of unknowns is $m + n = 3p^2$. In Table 4.4, we list the minimum positive eigenvalues μ_0 and $\bar{\mu}_0$ of $Q^{-1}B^T A^{-1}B$ and $\bar{Q}^{-1}B^T A^{-1}B$, respectively, for different values of m and n. We observe that μ_0, $\hat{\mu}_0 > 0.25$, and hence Theorem 2.2 can be applied to compute the optimal relaxation parameter of the SOR like and PSOR like methods. Numerical results are given in Table 4.5. All of the assumptions and notations are as before. As we observe the PSOR like method is superior to the SOR like method.

Table 4.4 The values of μ_0 and $\bar{\mu}_0$ for Example 4.2.

m		**128**	**512**	**1152**
n		**64**	**256**	**576**
m + n		**192**	**768**	**1728**
μ_0	Case I	0.5302	0.5085	0.5039
	Case II	0.5155	0.5043	0.5020
$\bar{\mu}_0$	Case I	0.7166	0.6746	0.6661
	Case II	0.6490	0.6057	0.5968

Table 4.5 Numerical results for Example 4.2.

m			**128**	**512**	**1152**
n			**64**	**256**	**576**
m + n			**192**	**768**	**1728**
Case I	SOR-like	ω_{opt}	0.5608	0.3325	0.2345
		$\rho(M_{\omega_{opt}})$	0.6027	0.8170	0.8749
		ITS	73	157	248
		Time	0.09	1.83	13.52
	PSOR-like	ω_{opt}	0.9659	0.8640	0.7362
		$\rho(M_{\omega_{opt}})$	0.1848	0.3688	0.5136
		ITS	19	33	51
		Time	0.05	0.97	8.20
Case II	SOR-like	ω_{opt}	0.4308	0.2439	0.1697
		$\rho(M_{\omega_{opt}})$	0.7544	0.8695	0.9112
		ITS	113	207	351
		Time	0.05	2.16	16.83
	PSOR-like	ω_{opt}	0.9544	0.8553	0.7315
		$\rho(M_{\omega_{opt}})$	0.2135	0.3804	0.5181
		ITS	24	34	51
		Time	0.02	0.95	8.22

5. Conclusion

In this note, we have used a block diagonal preconditioner for the augmented linear system of equations. Then the SOR-like method has been applied to solve the preconditioned linear system of equations. Numerical results of the proposed method, which we have named PSOR-like method, is more effective than the SOR-like method.

References

[1] Brenner S.C. and Scott L.R., *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, 1994.

[2] Wright S., Stability of augmented system factorizations in interior-point methods, *SIAM J. Matr. Anal. Appl.*, 1997; **18**: 191-222.

[3] Bjorck A., *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[4] Young D.M., *Iterative Solution for Large Linear Systems*, Academic Press, New York, 1971.

[5] Golub G.H., Wu X. and Yuan J.Y., SOR-like methods for augmented systems, *BIT*, 2001; **41**: 71-85.

[6] Shao X., Li Z. and Li C., Modified SOR-like method for augmented syetem, *Int. J. Comp. Math.*, 2007; **84**: 1653-1662.

[7] Li Z., Li C., Evans D.J. and Zhang T., Two-parameter GSOR method for the augmented system, *Int. J. Comp. Math.*, 2005; **82**: 1033-1042.

[8] Li C., Li Z., Nie Y.Y. and Evans D.J., Genenralized AOR method for the augmented system, *Int. J. Comp. Math.*, 2004; **81**: 495- 504.

[9] Golub G.H. and van Loan C.F., *Matrix computations*, 3rd ed., Johns Hopkins Press, Baltimore, 1996.

[10] Saad Y., *Iterative Methods for Sparse linear Systems*, PWS press, New York, 1995.

[11] Wu S.L., Huang T.Z. and Zhao X.L., A modified SSOR iterative method for augmented systems, *J. Comp. Appl. Math.*, 2009; **228**: 424-433.