J. Appl. Math. & Informatics Vol. **34**(2016), No. 1 - 2, pp. 85 - 94
http://dx.doi.org/10.14317/jami.2016.085

FAST ONE-PARAMETER RELAXATION METHOD WITH A SCALED PRECONDITIONER FOR SADDLE POINT PROBLEMS†

SEYOUNG OH AND JAE HEON YUN*

Abstract. In this paper, we first propose a fast one-parameter relaxation (FOPR) method with a scaled preconditioner for solving the saddle point problems, and then we present a formula for finding its optimal parameter. To evaluate the effectiveness of the proposed FOPR method with a scaled preconditioner, numerical experiments are provided by comparing its performance with the existing one or two parameter relaxation methods with optimal parameters such as the SOR-like, the GSOR and the GSSOR methods.

AMS Mathematics Subject Classification : 65F10, 65F50.
Key word and phrases : Relaxation iterative method, Saddle point problem, Preconditioner, Spectral radius.

1. Introduction

We consider a fast one-parameter relaxation method with a scaled preconditioner for solving the saddle point problem

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

where $A \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix, and $B \in \mathbb{R}^{m\times n}$ is a matrix of full column rank. This problem (1) appears in many different scientific applications, such as constrained optimization [9], the finite element approximation for solving the Navier-Stokes equation [5, 6], and the constrained least squares problems and generalized least squares problems [1, 3, 12]. So many authors have proposed one or two parameter relaxation iterative methods for solving the saddle point problem (1). Golub et al. [7] proposed the one-parameter

Received November 2, 2015. Accepted December 10, 2015. *Corresponding author.
†This work was supported by the intramural research grant of Chungbuk National University in 2015.

SOR-like method and presented an incomplete formula for finding one optimal parameter, Bai et al. [3] proposed the two-parameter GSOR (Generalized SOR) method and presented a formula for finding two optimal parameters for the GSOR and a complete formula for finding one optimal parameter for SOR-like method, Zhang and Lu [14] studied the two-parameter GSSOR (Generalized symmetric SOR) method and Chao et al [4] presented a formula for finding two optimal parameters for the GSSOR, and so on [10, 13].

This paper is organized as follows. In Section 2, we propose a fast one-parameter relaxation (FOPR) method with a scaled preconditioner, and then we present a formula for finding its optimal parameter. In Section 3, numerical experiments are provided to examine the effectiveness of the proposed FOPR method with a scaled preconditioner by comparing its performance with the existing one or two parameter relaxation methods with optimal parameters such as the SOR-like, the GSOR and the GSSOR methods. Lastly, some conclusions are drawn.

2. Convergence of a fast one-parameter relaxation (FOPR) method

For the coefficient matrix of the augmented linear system (1), we consider the following splitting

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} = D - L - U, \tag{2}$$

where

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

and $Q \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix which approximates $B^T A^{-1} B$. Let

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad c = \begin{pmatrix} b \\ -q \end{pmatrix}, \quad \Omega = \begin{pmatrix} \omega I_m & 0 \\ 0 & \frac{1}{\omega} I_n \end{pmatrix},$$

where $\omega > 0$ is a relaxation parameter, $I_m \in \mathbb{R}^{m\times m}$ and $I_n \in \mathbb{R}^{n\times n}$ denote the identity matrices of order m and n, respectively. Then a fast one-parameter relaxation (FOPR) method for solving the saddle point problem (1) is defined by

$$z_{k+1} = T_\omega\, z_k + g_\omega, \;\; k = 0, 1, 2, \ldots, \tag{3}$$

where $T_\omega = (D - \Omega L)^{-1}((I - \Omega)D + \Omega U)$ is an iteration matrix for the FOPR method, $g_\omega = \Omega\, c$, and I is an identity matrix of order $m + n$. That is, the FOPR method is defined by

$$\begin{aligned} x_{k+1} =& (1-\omega)x_k + \omega A^{-1}(b - B y_k) \\ y_{k+1} =& y_k + (\omega\, Q)^{-1}(B^T x_{k+1} - q). \end{aligned} \tag{4}$$

Note that the FOPR method can be viewed as a special case of the GSOR method [3] with $\tau = \frac{1}{\omega}$.

Lemma 2.1 ([11]). *Consider the quadratic equation $x^2 - bx + c = 0$, where b and c are real numbers. Both roots of the equation are less than one in modulus if and only if $|c| < 1$ and $|b| < 1 + c$.*

Let λ be an eigenvalue of T_ω and $\begin{pmatrix} u \\ v \end{pmatrix}$ be the corresponding eigenvector. Then we have

$$\begin{pmatrix} (1-\omega)A & -\omega B \\ 0 & Q \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} A & 0 \\ -\frac{1}{\omega} B^T & Q \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix},$$

or equivalently,

$$\begin{aligned} &(1 - \lambda - \omega)Au = \omega Bv, \\ &\frac{\lambda}{\omega} B^T u = (\lambda - 1)Qv. \end{aligned} \tag{5}$$

From now on, let $\rho(H)$ denote the spectral radius of a square matrix H. The following theorem provides the convergence result for the FOPR method.

Theorem 2.2. *Let μ_{max} be the spectral radius of $Q^{-1}B^T A^{-1}B$. If $\mu_{max} < 4$, then the FOPR method converges for all $0 < \omega < 2 - \frac{\mu_{max}}{2}$.*

Proof. Let μ be an eigenvalue of $Q^{-1}B^T A^{-1}B$ and λ be an eigenvalue of T_ω. Then $\mu > 0$. From equation (5), one can obtain the following quadratic equation for λ

$$\lambda^2 + (\omega + \mu - 2)\lambda + 1 - \omega = 0. \tag{6}$$

Applying Lemma 2.1 to (6), one easily obtains $0 < \omega < 2 - \frac{\mu}{2}$. If $0 < \omega < 2 - \frac{\mu_{max}}{2}$, then $\rho(T_\omega) < 1$, which completes the proof. □

Notice that if $\mu_{max} \geq 4$ in Theorem 2.2, then the convergence region for which the FOPR method converges may be an empty set. Next theorem provides an optimal parameter ω for which the FOPR method performs best.

Theorem 2.3. *Let μ_{min} and μ_{max} be the minimum and maximun eigenvalues of $Q^{-1}B^T A^{-1}B$, respectively. Assume that $\mu_{max} < 4$. Then the optimal parameter ω for the FOPR method is given by $\omega = \omega_o$, where*

$$\omega_o = \min\{2\sqrt{\mu_{min}} - \mu_{min},\ 2\sqrt{\mu_{max}} - \mu_{max}\}.$$

Moreover $\rho(T_{\omega_o}) = \sqrt{1 - \omega_o}$. That is,

$$\rho(T_{\omega_o}) = \begin{cases} |1 - \sqrt{\mu_{min}}| & \text{if } \omega_o = 2\sqrt{\mu_{min}} - \mu_{min} \\ |1 - \sqrt{\mu_{max}}| & \text{if } \omega_o = 2\sqrt{\mu_{max}} - \mu_{max} \end{cases}. \tag{7}$$

Proof. Let μ be an eigenvalue of $Q^{-1}B^T A^{-1}B$ and λ be an eigenvalue of T_ω. From the quadratic equation (6) for λ, one obtains two roots

$$\lambda = \frac{1}{2}\left((2 - \omega - \mu) \pm \sqrt{(\omega + \mu)^2 - 4\mu}\right).$$

Let $f(\omega) = 2 - \omega - \mu$ and $g(\omega) = (\omega + \mu)^2 - 4\mu$. The necessary and sufficient condition for the roots λ to be real is $g(\omega) \geq 0$, which is equivalent to $\omega \geq 2\sqrt{\mu} - \mu$. Since $\mu_{max} < 4$, $2\sqrt{\mu} - \mu < 2 - \frac{\mu}{2}$. Hence one obtains

$$|\lambda| = \begin{cases} \frac{1}{2}\left(\,|f(\omega)| + \sqrt{g(\omega)}\,\right) & \text{if } 2\sqrt{\mu} - \mu \leq \omega < 2 - \frac{\mu}{2} \\ \sqrt{1-\omega} & \text{if } 0 < \omega \leq 2\sqrt{\mu} - \mu \end{cases}. \tag{8}$$

Notice that $(2\sqrt{\mu} - \mu) \in (0, 1]$ for $\mu \in (0, 4)$ and it has the maximum value 1 at $\mu = 1$. Since $\frac{\partial}{\partial \omega}(|f| + \sqrt{g}) = \text{sign}(f) + \frac{\omega+\mu}{\sqrt{g}} > 0$ for $\omega \geq 2\sqrt{\mu} - \mu$, $\frac{1}{2}(|f| + \sqrt{g})$ is an increasing function for $\omega \geq 2\sqrt{\mu} - \mu$. Clearly $\sqrt{1-\omega}$ is a decreasing function for $0 < \omega \leq 2\sqrt{\mu} - \mu$. Thus, (8) implies that given μ, $|\lambda|$ takes the minimum $\sqrt{1-\omega} = |1 - \sqrt{\mu}|$ when $\omega = 2\sqrt{\mu} - \mu$. If S is a set containing all eigenvalues of $Q^{-1}B^TA^{-1}B$, then

$$\min_{\omega} \rho(T_\omega) = \max_{\mu} \min_{\omega} |\lambda| = \max_{\omega = 2\sqrt{\mu} - \mu,\ \mu \in S} \sqrt{1-\omega} = \sqrt{1-\omega_o}, \tag{9}$$

where $\omega_o = \min\{2\sqrt{\mu_{min}} - \mu_{min},\ 2\sqrt{\mu_{max}} - \mu_{max}\}$. Hence the theorem follows. $\square$

As can be shown in Theorems 2.2 and 2.3, a big disadvantage of the FOPR method is that it requires a rather strong condition $\mu_{max} < 4$ which may not be true for some types of preconditioners Q. To remedy this problem, we need to scale the preconditioner Q so that $0 < \mu_{min},\ \mu_{max} < 4$. From Theorem 2.3, it can be also seen that in order to minimize $\rho(T_{\omega_o})$, Q needs to be scaled so that $2\sqrt{\mu_{min}} - \mu_{min} = 2\sqrt{\mu_{max}} - \mu_{max}$. Next theorem shows how to scale the preconditioner Q such that $0 < \mu_{min},\ \mu_{max} < 4$ and $\rho(T_{\omega_o})$ can be minimized. Next theorem also shows an optimal convergence factor of the FOPR method with a scaled preconditioner.

Theorem 2.4. *Let μ_{min} and μ_{max} be the minimum and maximun eigenvalues of $Q^{-1}B^TA^{-1}B$, respectively. Let $Q_s = s\,Q$ be a scaled preconditioner, where $s > 0$ is a scaling factor, and let ν_{min} and ν_{max} be the minimum and maximun eigenvalues of $Q_s^{-1}B^TA^{-1}B$, respectively. If $s = \left(\frac{\sqrt{\mu_{min}}+\sqrt{\mu_{max}}}{2}\right)^2$, then $0 < \nu_{min},\ \nu_{max} < 4$ and $2\sqrt{\nu_{min}} - \nu_{min} = 2\sqrt{\nu_{max}} - \nu_{max}$. Moreover $\tilde{\omega}_o = 2\sqrt{\nu_{min}} - \nu_{min} = 2\sqrt{\nu_{max}} - \nu_{max}$ and*

$$\rho(\tilde{T}_{\tilde{\omega}_o}) = |1 - \sqrt{\nu_{min}}\,| = |1 - \sqrt{\nu_{max}}\,| = \frac{\sqrt{\mu_{max}} - \sqrt{\mu_{min}}}{\sqrt{\mu_{max}} + \sqrt{\mu_{min}}},$$

where $\tilde{\omega}_o$ and $\tilde{T}_\omega$ refer to the optimal parameter and the iteration matrix for the FOPR method with the scaled preconditioner Q_s, respectively.

Proof. Since $Q_s^{-1}B^TA^{-1}B = \frac{1}{s}Q^{-1}B^TA^{-1}B$, $\nu_{min} = \frac{\mu_{min}}{s}$ and $\nu_{max} = \frac{\mu_{max}}{s}$. Since $s = \left(\frac{\sqrt{\mu_{min}}+\sqrt{\mu_{max}}}{2}\right)^2$, one obtains

$$\begin{aligned} \nu_{min} =& \frac{4\mu_{min}}{(\sqrt{\mu_{max}}+\sqrt{\mu_{min}})^2} < 4, \\ \nu_{max} =& \frac{4\mu_{max}}{(\sqrt{\mu_{max}}+\sqrt{\mu_{min}})^2} < 4. \end{aligned} \tag{10}$$

Using (10), it can be easily shown that

$$2\sqrt{\nu_{min}} - \nu_{min} = 2\sqrt{\nu_{max}} - \nu_{max} = \frac{4\sqrt{\mu_{min}}\sqrt{\mu_{max}}}{(\sqrt{\mu_{max}}+\sqrt{\mu_{min}})^2} = \tilde{\omega}_o.$$

The remaining part of this theorem can be proved by simple calculation. □

From Theorem 2.4, it can be seen that optimal convergence factor of the FOPR method with the scaled preconditioner Q_s is the same as that of the GSOR method [3] with the preconditioner Q. Notice that the scaling factor s in Theorem 2.4 can be easily computed using MATLAB by computing only the largest and smallest eigenvalues of $Q^{-1}B^TA^{-1}B$.

We next summarize the formulas for finding optimal parameters of the SOR-like, the GSOR and the GSSOR methods which are used for numerical experiments in Section 3.

Remark 2.1 ([7, 3, 4])**.** Let μ_{min} and μ_{max} be the minimum and maximun eigenvalues of $Q^{-1}B^TA^{-1}B$, respectively. Then

(a) The optimal parameter ω_o for the SOR-like method takes one of the following 3 formulas depending upon the values of μ_{min} and μ_{max} (see [3] for more details):

$$\frac{4}{1+\sqrt{1+4(\mu_{min}+\mu_{max})}},\quad \frac{2\sqrt{\mu_{min}}-1}{\mu_{min}},\quad \frac{2\sqrt{\mu_{max}}-1}{\mu_{max}}.$$

(b) The optimal parameters ω_o and τ_o for the GSOR method are given by

$$\omega_o = \frac{4\sqrt{\mu_{min}\mu_{max}}}{(\sqrt{\mu_{min}}+\sqrt{\mu_{max}})^2} \quad \text{and} \quad \tau_o = \frac{1}{\sqrt{\mu_{min}\mu_{max}}}.$$

(c) The optimal parameters ω_o and τ_o for the GSSOR method are given by

$$\omega_o = 1 \pm \frac{\sqrt{\mu_{max}}-\sqrt{\mu_{max}}}{\sqrt{\mu_{max}}+\sqrt{\mu_{max}}} \quad \text{and} \quad \tau_o = 1 + \frac{1 \pm \sqrt{1+4\mu_{min}\mu_{max}}}{2\sqrt{\mu_{min}\mu_{max}}}.$$

3. Numerical results

To evaluate the effectiveness of the FOPR method with a scaled preconditioner, we provide numerical experiments by comparing its performance with the SOR-like, the GSOR and the GSSOR methods. For performance comparison, both the FOPR method with preconditioner Q and the FOPR method with scaled preconditioners $Q_s = s\,Q$ and $Q_{s+\epsilon} = (s+\epsilon)Q$ are provided, where s

is the scaling factor defined in Theorem 2.4 and ϵ is a positive number which is chosen appropriately small as compared with s. In Tables 2 to 5, *Iter* denotes the number of iteration steps and *CPU* denotes the elapsed CPU time in seconds. In all experiments, the right hand side vector $(b^T, -q^T)^T \in \mathbb{R}^{m+n}$ was chosen such that the exact solution of the saddle point problem (1) is $(x_*^T, y_*^T)^T = (1, 1, \ldots, 1)^T \in \mathbb{R}^{m+n}$, and the initial vector was set to the zero vector. From now on, let $\|\cdot\|$ denote the L_2-norm.

Example 3.1 ([2])**.** We consider the saddle point problem (1), in which

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times 2p^2}, \quad B = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times p^2},$$

and

$$T = \frac{1}{h^2}\mathrm{Tridiag}(-1, 2, -1) \in \mathbb{R}^{p \times p}, \ F = \frac{1}{h}\mathrm{Tridiag}(-1, 1, 0) \in \mathbb{R}^{p \times p},$$

with $\otimes$ denoting the Kronecker product and $h = \frac{1}{p+1}$ the discretization mesh size. For this example, $m = 2p^2$ and $n = p^2$. Thus the total number of variables is $3p^2$. We choose the matrix Q as an approximation to the matrix $B^T A^{-1} B$, according to four cases listed in Table 1. The iterations for the relaxation iterative methods are terminated if the current iteration satisfies ERR $< 10^{-9}$, where *ERR* is defined by

$$\mathrm{ERR} = \frac{\sqrt{\|x_k - x_*\|^2 + \|y_k - y_*\|^2}}{\sqrt{\|x_0 - x_*\|^2 + \|y_0 - y_*\|^2}}.$$

Numerical results for this example are listed in Tables 2 to 5. In Tables 4 and 5, numerical results for the FOPR method are not listed since it does not work because of $\mu_{max} > 4$, and thus only those for the FOPR method with scaled preconditioners Q_s and $Q_{s+\epsilon}$ are listed.

Example 3.2 ([3])**.** We consider the same problem as Example 3.1 except that F is defined by

$$F = \frac{1}{h}K \ \text{ and } \ K = (k_{ij}) = \left(\frac{1}{2\sqrt{2\pi}} e^{-\frac{(i-j)^2}{8}} \right).$$

Note that the matrix F is a highly ill-conditioned Toeplitz matrix. So we choose only Cases III and IV of Q in Table 1 as an approximation to the matrix $B^T A^{-1} B$, and all iterations are terminated if the current iteration satisfies RES $< 10^{-9}$, where *RES* is defined by

$$\mathrm{RES} = \frac{\sqrt{\|b - Ax_k - By_k\|^2 + \|q - B^T x_k\|^2}}{\sqrt{\|b\|^2 + \|q\|^2}}.$$

Since $\mu_{max} > 4$ for these choices of Q, numerical results for the FOPR method with scaled preconditioners Q_s and $Q_{s+\epsilon}$ are listed in Tables 4 to 5.

All numerical tests are carried out on a PC equipped with Intel Core i5-4570 3.2GHz CPU and 8GB RAM using MATLAB R2014a. For numerical experiments of all relaxation iterative methods used in this paper, the optimal parameters described in Remark 2.1 are used. For test runs of the FOPR method with the scaled preconditioner $Q_{s+\epsilon}$, we have tried the values of $\epsilon \in [0.0001, 0.0005]$ in Tables 2 and 3, and the values of $\epsilon \in [0.01, 0.05]$ in Tables 4 and 5. For all of these values of ϵ, the FOPR method with $Q_{s+\epsilon}$ performs better than the GSOR method, and the values of ϵ for which it performs best are reported in Tables 2 to 5.

As can be expected from Theorem 2.4, the FOPR method with the scaled preconditioner Q_s performs as well as the GSOR method. The FOPR method with the scaled preconditioner $Q_{s+\epsilon}$ performs best of all methods considered in this paper, and specifically it performs much better than other methods for Cases III and IV of Q for which $\mu_{max} > 4$ (see Tables 2 to 5). On the other hand, the GSSOR method performs worse than the FOPR and the GSOR methods since its computational cost for each iteration is higher than others.

Table 1. Choices of the matrix Q.

Case Number	Q	Description
I	$\text{Tridiag}(B^T\tilde{A}^{-1}B)$	$\tilde{A} = \text{Tridiag}(A)$
II	$\text{Tridiag}(B^T A^{-1}B)$	
III	$B^T\tilde{A}^{-1}B$	$\tilde{A} = \text{Tridiag}(A)$
IV	$B^T\tilde{A}^{-1}B$	$\tilde{A} = \text{Diag}(A)$

4. conclusions

In this paper, we proposed a fast one-parameter relaxation (FOPR) method with a scaled preconditioner for solving the saddle point problems, and then we presented a formula for finding its optimal parameter. Both theoretical and computational results showed that the FOPR methods with the scaled preconditioner Q_s performs as well as the GSOR method. In addition, the FOPR method with the scaled preconditioner $Q_{s+\epsilon}$ performs better than other one or two parameter relaxation methods with optimal parameters, and specifically it performs much better than other methods for Cases III and IV of Q for which $\mu_{max} > 4$ (see Tables 2 to 5). Hence, it may be concluded that the FOPR method with the scaled preconditioner $Q_{s+\epsilon}$ is strongly recommended for solving the saddle point problems when $\mu_{max} = \rho(Q^{-1}B^TA^{-1}B) > 4$.

References

1. M. Arioli, I.S. Duff, P.P.M. de Rijk, On the augmented system approach to sparse least squares problems, Numer. Math. **55** (1989), 667-684.
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.

TABLE 2. Numerical results for Example 3.1 with Case I of Q.

	m	1152	2048
	n	576	1024
SOR-like	ω_o	1.0476	1.0451
	$Iter$	275	359
	CPU	0.240	0.575
GSOR	ω_o	0.5585	0.5087
	τ_o	2.9943	3.3529
	$Iter$	67	78
	CPU	0.059	0.126
FOPR	ω_o	0.4529	0.4056
	$Iter$	87	102
	CPU	0.076	0.166
FOPR with Q_s	ω_o	0.5585	0.5087
	s	0.6020	0.5863
	$Iter$	66	78
	CPU	0.059	0.126
FOPR with $Q_{s+\epsilon}$	ω_o	0.5584	0.5086
(s is the same as above)	ϵ	0.0002	0.0002
	$Iter$	64	74
	CPU	0.057	0.120
GSSOR	ω_o	0.3356	0.2991
	τ_o	0.6951	0.7244
	$Iter$	68	79
	CPU	0.118	0.246

TABLE 3. Numerical results for Example 3.1 with Case II of Q.

	m	1152	2048
	n	576	1024
SOR-like	ω_o	1.1413	1.1453
	$Iter$	248	324
	CPU	0.211	0.514
GSOR	ω_o	0.6161	0.5669
	τ_o	3.4069	3.8802
	$Iter$	56	65
	CPU	0.050	0.104
FOPR	ω_o	0.4562	0.4079
	$Iter$	86	101
	CPU	0.075	0.157
FOPR with Q_s	ω_o	0.6160	0.5667
	s	0.4764	0.4546
	$Iter$	56	65
	CPU	0.050	0.104
FOPR with $Q_{s+\epsilon}$	ω_o	0.6160	0.5668
(s is the same as above)	ϵ	0.0002	0.0002
	$Iter$	55	63
	CPU	0.049	0.100
GSSOR	ω_o	0.3804	0.3419
	τ_o	0.7282	0.7574
	$Iter$	57	66
	CPU	0.098	0.204

3. Z.Z. Bai, B.N. Parlett, Z.Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, Numer. Math. **102** (2005), 1-38.

TABLE 4. Numerical results for Case III of Q.

		Example 3.1		Example 3.2	
	m	1152	2048	1152	2048
	n	576	1024	576	1024
SOR-like	ω_o	0.2614	0.2035	0.2342	0.1807
	$Iter$	167	216	160	239
	CPU	0.285	0.806	1.120	5.813
GSOR	ω_o	0.3307	0.2635	0.2997	0.2363
	τ_o	0.1985	0.1519	0.1762	0.1340
	$Iter$	149	199	141	191
	CPU	0.255	0.744	0.990	4.553
FOPR with Q_s	ω_o	0.3307	0.2635	0.2997	0.2363
	s	15.24	24.98	18.94	31.58
	$Iter$	149	199	141	191
	CPU	0.255	0.744	0.990	4.553
FOPR with $Q_{s+\epsilon}$ (s is the same as above)	ω_o	0.3305	0.2634	0.2995	0.2362
	ϵ	0.02	0.02	0.02	0.03
	$Iter$	116	160	122	158
	CPU	0.199	0.601	0.864	3.675
GSSOR	ω_o	0.1819	0.1418	0.1631	0.1261
	τ_o	0.0943	0.0731	0.0842	0.0648
	$Iter$	150	200	143	190
	CPU	0.379	1.046	1.143	4.736

TABLE 5. Numerical results for Case IV of Q.

		Example 3.1		Example 3.2	
	m	1152	2048	1152	2048
	n	576	1024	576	1024
SOR-like	ω_o	0.1912	0.1476	0.1696	0.1300
	$Iter$	230	298	249	328
	CPU	0.298	0.703	3.375	13.46
GSOR	ω_o	0.2489	0.1956	0.2236	0.1743
	τ_o	0.1423	0.1084	0.1248	0.0946
	$Iter$	213	286	205	273
	CPU	0.275	0.665	2.778	10.51
FOPR with Q_s	ω_o	0.2489	0.1956	0.2236	0.1743
	s	28.24	47.15	35.83	60.61
	$Iter$	213	286	201	272
	CPU	0.275	0.665	2.750	10.47
FOPR with $Q_{s+\epsilon}$ (s is the same as above)	ω_o	0.2488	0.1955	0.2235	0.1740
	ϵ	0.02	0.02	0.02	0.03
	$Iter$	171	223	170	239
	CPU	0.220	0.520	2.303	9.295
GSSOR	ω_o	0.1333	0.1031	0.1189	0.0913
	τ_o	0.0686	0.0528	0.0605	0.0462
	$Iter$	214	287	206	274
	CPU	0.453	1.092	2.970	11.23

4. Z. Chao, N. Zhang, Y.Z. Lu, Optimal parameters of the generalized symmetric SOR method for augmented systems, J. Comput. Appl. Math. **266** (2014), 52-60.
5. H. Elman, D.J. Silvester, Fast nonsymmetric iteration and preconditioning for Navier-Stokes equations, SIAM J. Sci. Comput. **17** (1996), 33-46.
6. B. Fischer, A. Ramage, D.J. Silvester, A.J. Wathen, Minimum residual methods for augmented systems, BIT **38** (1998), 527-543.

7. G.H. Golub, X. Wu, J.Y. Yuan, SOR-like methods for augmented systems, BIT **41** (2001), 71-85.
8. G.H. Santos, B.P.B. Silva, J.Y. Yuan, Block SOR methods for rank deficient least squares problems, J. Comput. Appl. Math. **100** (1998), 1-9.
9. S. Wright, Stability of augmented system factorization in interior point methods, SIAM J. Matrix Anal. Appl. **18** (1997), 191-222.
10. S.L. Wu, T.Z. Huang, X.L. Zhao, A modified SSOR iterative method for augmented systems, J. Comput. Appl. Math. **228** (2009), 424-433.
11. D.M. Young, Iterative solution of large linear systems, Academic Press, New York, 1971.
12. J.Y. Yuan, A.N. Iusem, Preconditioned conjugate gradient methods for generalized least squares problem, J. Comput. Appl. Math. **71** (1996), 287-297.
13. SeYoung Oh, J.H. Yun, K.S. Kim, ESOR method with diagonal preconditioners for SPD linear systems, J. Appl. Math. & Informatics 33 (2015), 111-118.
14. G.F. Zhang, Q.H. Lu, On generalized symmetric SOR method for augmented systems, J. Comput. Appl. Math. **219** (2008), 51-58.

SeYoung Oh received M.Sc. from Seoul National University and Ph.D at University of Minnesota. Since 1992 he has been at Chungnam National University. His research interests include numerical optimization and biological computation.

Department of Mathematics, Chungnam National University, Daejeon 34134, Korea
e-mail: soh@cnu.ac.kr

Jae Heon Yun received M.Sc. from Kyungpook National University, and Ph.D. from Iowa State University. He is currently a professor at Chungbuk National University since 1991. His research interests are computational mathematics, iterative method and parallel computation.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 28644, Korea
e-mail: gmjae@chungbuk.ac.kr