Bull. Korean Math. Soc. **52** (2015), No. 5, pp. 1669–1681
http://dx.doi.org/10.4134/BKMS.2015.52.5.1669

SEMI-CONVERGENCE OF THE PARAMETERIZED INEXACT UZAWA METHOD FOR SINGULAR SADDLE POINT PROBLEMS

Jae Heon Yun

Abstract. In this paper, we provide semi-convergence results of the parameterized inexact Uzawa method with singular preconditioners for solving singular saddle point problems. We also provide numerical experiments to examine the effectiveness of the parameterized inexact Uzawa method with singular preconditioners.

1. Introduction

We consider the singular saddle point problem of the form

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

where $A \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m\times n}$ is a rank-deficient matrix of $\operatorname{rank}(B) = r < n$ with $m \geq n$, $f \in \mathbb{R}^m$ and $g \in \mathbb{R}^n$. The singular saddle point problem (1) is important and arises in many different applications of scientific computing and engineering, such as the mixed finite element methods for Navier-Stokes equations, computational fluid dynamics, constrained optimization, the weighted least squares problems, electronic networks, linear elasticity, and so forth [1, 10, 14, 15].

When B is of full column rank, the linear system (1) is nonsingular. For this case, many relaxation iterative methods based on matrix splittings and their convergence properties have been proposed and analyzed, e.g., SOR-like method [11], GSOR (Generalized SOR) method [2], PIU (Parameterized Inexact Uzawa) method [3], SSOR-like method [8, 20], GSSOR (Generalized SSOR) method [7, 18], several variants of Uzawa method [16, 17], and so on.

Received November 11, 2014; Revised March 20, 2015.

2010 *Mathematics Subject Classification.* 65F10, 65F50.

Key words and phrases. singular saddle point problem, iterative method, Uzawa method, semi-convergence, Moore-Penrose inverse.

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013 R1A1A2005722).

Recently, several authors have presented semi-convergence analysis of relaxation iterative methods with *nonsingular preconditioners* for solving the singular saddle point problem (1). Zheng et al. [21] studied semi-convergence of the PU (Parameterized Uzawa) method, Li and Huang [13] examined semi-convergence of the GSSOR method, Zhang and Wang [19] studied semi-convergence of the GPIU method, Chao and Chen [6] provided semi-convergence analysis of the Uzawa-SOR method, and Zhou and Zhang [22] studied semi-convergence of the GMSSOR (Generalized Modified SSOR) method.

So far, most iterative methods for solving singular saddle point problems have been studied using nonsingular preconditioners. Also notice that the Schur complement $S = B^T A^{-1} B$ of the coefficient matrix of the singular saddle problem (1) is singular. Hence it may be better to choose a singular preconditioner as an approximation of the singular Schur complement S. For this reason, this paper studies semi-convergence of the PIU (Parameterized Inexact Uzawa) method with *singular preconditioners* for solving the singular saddle point problem (1). This paper is organized as follows. In Section 2, we provide preliminary results for semi-convergence of the basic iterative methods. In Section 3, we provide semi-convergence analysis and quasi-optimal parameters for the PIU method with singular preconditioners. In Section 4, numerical experiments are carried out to examine the effectiveness of the PIU method with singular preconditioners. Lastly, some conclusions are drawn.

2. Preliminaries for semi-convergence analysis

For simplicity of exposition, some notation and definitions are presented. For a vector x, x^* denotes the complex conjugate transpose of the vector x. For a square matrix G, $R(G)$ denotes the range space of G, $N(G)$ denotes the null space of G, $\sigma(G)$ denotes the set of all eigenvalues of G, and $\rho(G)$ denotes the spectral radius of G.

Let us recall some useful results on iterative methods for solving singular linear systems based on matrix splitting. For a matrix $E \in \mathbb{R}^{n\times n}$, the smallest nonnegative integer k such that $\operatorname{rank}(E^k) = \operatorname{rank}(E^{k+1})$ is called the *index* of E, and denoted by $k = \operatorname{index}(E)$. In other words, $\operatorname{index}(E)$ is the size of the largest Jordan block corresponding to the zero eigenvalue of E. For a square matrix T, the *pseudo-spectral radius* $\nu(T)$ is defined by

$$\nu(T) = \max\{|\lambda| \mid \lambda \in \sigma(T) - \{1\}\},$$

where $\sigma(T)$ is the set of eigenvalues of T.

The *Moore-Penrose inverse* [4] of a singular matrix $E \in \mathbb{R}^{n\times n}$ is defined by the unique matrix $E^\dagger$ which satisfies the following equations

$$E = EE^\dagger E, \quad E^\dagger = E^\dagger E E^\dagger, \quad (EE^\dagger)^T = EE^\dagger, \quad (E^\dagger E)^T = E^\dagger E.$$

Let $A = M - N$ be a splitting of a singular matrix A, where M is singular. Then an iterative method corresponding to this singular splitting for solving a singular linear system $Ax = b$ is given by

$$x_{i+1} = (I - M^{\dagger}A)x_i + M^{\dagger}b \quad \text{for } i = 0, 1, \ldots. \tag{2}$$

Definition 2.1. The iterative method (2) is *semi-convergent* if for any initial guess x_0, the iteration sequence $\{x_i\}$ produced by (2) converges to a solution x_* of the singular linear system $Ax = b$.

Notice that a matrix T is called *semi-convergent* if $\lim_{k\to\infty} T^k$ exists, or equivalently $\mathrm{index}(I - T) = 1$ and $\nu(T) < 1$ [4].

Theorem 2.2 ([5])**.** *The iterative method* (2) *is semi-convergent if and only if* $\mathrm{index}(M^{\dagger}A) = 1$*,* $\nu(I - M^{\dagger}A) < 1$*, and* $N(M^{\dagger}A) = N(A)$*, i.e.,* $I - M^{\dagger}A$ *is semi-convergent and* $N(M^{\dagger}A) = N(A)$*.*

3. Semi-convergence analysis of the PIU method

In this section, we study semi-convergence of the PIU method with singular preconditioners for solving the singular saddle point problem (1). For the coefficient matrix of the singular saddle point problem (1), we consider the following splitting

$$\mathcal{A} = \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} = \mathcal{D} - \mathcal{L} - \mathcal{U}, \tag{3}$$

where

$$\mathcal{D} = \begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix}, \quad \mathcal{L} = \begin{pmatrix} 0 & 0 \\ B^T & 0 \end{pmatrix}, \quad \mathcal{U} = \begin{pmatrix} P - A & -B \\ 0 & Q \end{pmatrix}, \tag{4}$$

where $P \in \mathbb{R}^{m\times m}$ is a symmetric positive definite (SPD) matrix which approximates A, and $Q \in \mathbb{R}^{n\times n}$ is a singular symmetric positive semi-definite matrix which approximates the approximated Schur complement matrix $B^T P^{-1} B$.

Let

$$\Omega = \begin{pmatrix} \omega I_m & 0 \\ 0 & \tau I_n \end{pmatrix},$$

where $\omega > 0$ and $\tau > 0$ are relaxation parameters, $I_m \in \mathbb{R}^{m\times m}$ and $I_n \in \mathbb{R}^{n\times n}$ denote the identity matrices of order m and n, respectively. Let us assume that Q is chosen as $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates P. It is clear that such a Q is singular symmetric positive semi-definite. Then the PIU (Parameterized Inexact Uzawa) method with the singular preconditioner Q for solving the singular saddle point problem (1) is defined by

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = H(\omega,\tau) \begin{pmatrix} x_k \\ y_k \end{pmatrix} + M(\omega,\tau) \begin{pmatrix} f \\ -g \end{pmatrix}, \; k = 0, 1, 2, \ldots, \tag{5}$$

where

$$H(\omega,\tau) = I - (\mathcal{D} - \Omega\mathcal{L})^{\dagger}\Omega\mathcal{A},$$
$$M(\omega,\tau) = (\mathcal{D} - \Omega\mathcal{L})^{\dagger}\Omega.$$

Since $QQ^\dagger$ is an orthogonal projection onto $R(Q)$, $QQ^\dagger B^T = B^T$. Thus, simple calculation yields

$$(\mathcal{D} - \Omega\mathcal{L})^\dagger = \begin{pmatrix} P^{-1} & 0 \\ \tau Q^\dagger B^T P^{-1} & Q^\dagger \end{pmatrix} \tag{6}$$

and

$$H(\omega, \tau) = \begin{pmatrix} I_m - \omega P^{-1}A & -\omega P^{-1}B \\ \tau Q^\dagger B^T(I_m - \omega P^{-1}A) & I_n - \omega\tau Q^\dagger B^T P^{-1}B \end{pmatrix}. \tag{7}$$

From (5), (6) and (7), the PIU method with the singular preconditioner Q can be rewritten as

```
**Algorithm 1: PIU Method with singular $Q$**
Choose $\omega$, $\tau$ and initial vectors $x_0$, $y_0$
For $k = 0, 1, \ldots,$ until convergence
    $x_{k+1} = x_k + \omega P^{-1}(f - Ax_k - By_k)$
    $y_{k+1} = y_k + \tau Q^\dagger (B^T x_{k+1} - g)$
End For
```

Here, $Q^\dagger$ (Moore-Penrose inverse of the matrix Q) is computed only once to reduce computational amount, and then it is stored for later use. If $P = A$ in Algorithm 1, then the PIU method reduces to the PU method.

The following theorem shows semi-convergence of the PIU method with singular preconditioner.

Theorem 3.1. *Let Q be chosen as $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates P. Then the PIU method for solving the singular saddle point problem* (1) *is semi-convergent if ω and τ satisfy*

$$0 < \omega < \frac{2}{\eta_{\max}} \quad \textit{and} \quad 0 < \tau < \frac{2(2 - \eta_{\max}\,\omega)}{\omega\,\mu_{\max}},$$

where $\mu_{\max}$ and $\eta_{\max}$ are the largest eigenvalues of the matrices $Q^\dagger B^T P^{-1} B$ and $P^{-1}A$, respectively.

Proof. Assume that the rank of B is r, i.e., $r = \text{rank}(B) < n$. Let

$$B = W\Sigma V^* \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m\times n} \tag{8}$$

be the singular value decomposition of B, where W and V are unitary matrices, $\Sigma_r = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$ and σ_i's are positive singular values of B. Let W and V be partitioned into $W = (W_1, W_2)$ and $V = (V_1, V_2)$ with $W_1 \in \mathbb{C}^{m\times r}$, $W_2 \in \mathbb{C}^{m\times(m-r)}$, $V_1 \in \mathbb{C}^{n\times r}$, $V_2 \in \mathbb{C}^{n\times(n-r)}$, respectively. Let us define an $(m+n)\times(m+n)$ unitary matrix $\mathcal{P}$ as

$$\mathcal{P} = \begin{pmatrix} W & 0 \\ 0 & V \end{pmatrix}. \tag{9}$$

Let $\hat{H}(\omega,\tau) = \mathcal{P}^*H(\omega,\tau)\mathcal{P}$. Then

$$\hat{H}(\omega,\tau) = \begin{pmatrix} I_m - \omega W^*P^{-1}AW & -\omega W^*P^{-1}BV \\ \tau V^*Q^\dagger B^T(I_m - \omega P^{-1}A)W & I_n - \omega\tau V^*Q^\dagger B^TP^{-1}BV \end{pmatrix}. \tag{10}$$

If we define $\hat{A} = W^*AW$, $\hat{P} = W^*PW$ and $\hat{Q} = V^*QV$, then

$$\hat{H}(\omega,\tau) = \begin{pmatrix} I_m - \omega\hat{P}^{-1}\hat{A} & -\omega\hat{P}^{-1}\Sigma \\ \tau\hat{Q}^\dagger\Sigma^T(I_m - \omega\hat{P}^{-1}\hat{A}) & I_n - \omega\tau\hat{Q}^\dagger\Sigma^T\hat{P}^{-1}\Sigma \end{pmatrix}. \tag{11}$$

Since $Q = B^TM^{-1}B$ and $B = W\Sigma V^*$, one can obtain

$$\hat{Q} = V^*QV = \begin{pmatrix} \hat{Q}_1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{12}$$

where $\hat{Q}_1 = \Sigma_r W_1^* M^{-1} W_1 \Sigma_r$ is an $r \times r$ SPD matrix. Thus

$$\hat{Q}^\dagger = V^*Q^\dagger V = \begin{pmatrix} \hat{Q}_1^{-1} & 0 \\ 0 & 0 \end{pmatrix}. \tag{13}$$

If we let $B_1 = \left(\begin{smallmatrix} \Sigma_r \\ 0 \end{smallmatrix}\right) \in \mathbb{R}^{m\times r}$, then $\Sigma = (B_1, 0)$ and $\hat{Q}_1 = B_1^T(W^*M^{-1}W)B_1$. Hence, from (11) and (13) one obtains

$$\hat{H}(\omega,\tau) = \begin{pmatrix} I_m - \omega\hat{P}^{-1}\hat{A} & -\omega\hat{P}^{-1}B_1 & 0 \\ \tau\hat{Q}_1^{-1}B_1^T(I_m - \omega\hat{P}^{-1}\hat{A}) & I_r - \omega\tau\hat{Q}_1^{-1}B_1^T\hat{P}^{-1}B_1 & 0 \\ 0 & 0 & I_{n-r} \end{pmatrix}. \tag{14}$$

Notice that $\hat{A}$ and $\hat{Q}_1$ are Hermitian positive definite matrices. Let

$$\hat{H}_1(\omega,\tau) = \begin{pmatrix} I_m - \omega\hat{P}^{-1}\hat{A} & -\omega\hat{P}^{-1}B_1 \\ \tau\hat{Q}_1^{-1}B_1^T(I_m - \omega\hat{P}^{-1}\hat{A}) & I_r - \omega\tau\hat{Q}_1^{-1}B_1^T\hat{P}^{-1}B_1 \end{pmatrix}. \tag{15}$$

Then $\hat{H}_1(\omega,\tau)$ is the iteration matrix of the PIU method applied to the following nonsingular saddle point problem

$$\begin{pmatrix} \hat{A} & B_1 \\ -B_1^T & 0 \end{pmatrix}\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \hat{f} \\ -\hat{g} \end{pmatrix} \tag{16}$$

with the preconditioning matrix $\hat{Q}_1$ and $\hat{P}$ as an approximation of $\hat{A}$. From Theorem 2.2 in [3], $\rho(\hat{H}_1(\omega,\tau)) < 1$ is obtained if $0 < \omega < \frac{2}{\eta_{\max}}$ and $0 < \tau < \frac{2(2-\eta_{\max}\omega)}{\omega\,\mu_{\max}}$, where $\mu_{\max}$ and $\eta_{\max}$ are the largest eigenvalues of the nonsingular matrices $\hat{Q}_1^{-1}B_1^T\hat{P}^{-1}B_1$ and $\hat{P}^{-1}\hat{A}$, respectively. On the other hand, $W^*(P^{-1}A)W = \hat{P}^{-1}\hat{A}$ and

$$\begin{aligned} V^*(Q^\dagger B^TP^{-1}B)V &= \hat{Q}^\dagger\Sigma^T\hat{P}^{-1}\Sigma \\ &= \begin{pmatrix} \hat{Q}_1^{-1}B_1^T\hat{P}^{-1}B_1 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Hence, $\mu_{\max}$ and $\eta_{\max}$ are also the largest eigenvalues of the matrices

$$Q^\dagger B^TP^{-1}B \quad \text{and} \quad P^{-1}A, \quad \text{respectively.}$$

Since $\rho(\hat{H}_1(\omega,\tau)) < 1$, (14) and (15) imply that the matrix $\hat{H}(\omega,\tau)$ is semi-convergent. Since $\hat{H}(\omega,\tau) = \mathcal{P}^* H(\omega,\tau)\mathcal{P}$, $H(\omega,\tau)$ is also semi-convergent.

Notice that $\Omega\mathcal{A} = (\mathcal{D} - \Omega\mathcal{L}) - ((I-\Omega)\mathcal{D} + \Omega\mathcal{U})$. Hence, from Theorem 2.2 it remains to show that $N(\Omega\mathcal{A}) = N((\mathcal{D}-\Omega\mathcal{L})^\dagger \Omega\mathcal{A})$. Since $N(\Omega\mathcal{A}) = N(\mathcal{A})$, it is sufficient to show that $N((\mathcal{D}-\Omega\mathcal{L})^\dagger\Omega\mathcal{A}) \subset N(\mathcal{A})$. Suppose that $\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) \in N((\mathcal{D}-\Omega\mathcal{L})^\dagger\Omega\mathcal{A})$. Then

$$\omega P^{-1}(Ax+By) = 0 \quad \text{and} \quad \omega\tau Q^\dagger B^T P^{-1}(Ax+By) - \tau Q^\dagger B^T x = 0.$$

From these equations, $Ax + By = 0$ and $-Q^\dagger B^T x = 0$. Since $QQ^\dagger B^T = B^T$, $-B^T x = 0$, which implies $\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) \in N(\mathcal{A})$. Therefore, the proof is complete. □

Since the PIU method reduces to the PU method when $P = A$, the following corollary for semi-convergence of the PU method is obtained.

Corollary 3.2. *Let Q be chosen as $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates A. Then the PU method for solving the singular saddle point problem* (1) *is semi-convergent if ω and τ satisfy*

$$0 < \omega < 2 \quad \textit{and} \quad 0 < \tau < \frac{2(2-\omega)}{\omega\,\mu_{\max}},$$

where $\mu_{\max}$ is the largest eigenvalues of the matrices $Q^\dagger B^T A^{-1} B$.

From Theorem 3.1 in [3], we can obtain the following theorem about the quasi-optimal parameters and the corresponding quasi-optimal semi-convergence factor for the PIU method.

Theorem 3.3. *Consider the PIU method for solving the singular saddle point problem* (1). *Assume that Q is chosen as $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates P. Let $\mu_{\min}$ and $\mu_{\max}$ be the smallest and largest nonzero eigenvalues of the matrix $Q^\dagger B^T P^{-1} B$ respectively, and let $\eta_{\min}$ and $\eta_{\max}$ be the smallest and largest eigenvalues of the matrix $P^{-1}A$ respectively. Then the quasi-optimal parameters ω_{opt} and τ_{opt} are given by*

$$\omega_{opt} = \frac{4}{(\mu_{\min}+\mu_{\max})\,\tau_0 + 2\,\eta_{\max}} \quad \textit{and} \quad \tau_{opt} = \tau_0$$

and the corresponding quasi-optimal semi-convergence factor $\nu(H(\omega,\tau))$ is

$$\nu(H(\omega_{opt},\tau_{opt})) = \sqrt{1 - \frac{4\,\eta_{\min}}{(\mu_{\min}+\mu_{\max})\,\tau_0 + 2\,\eta_{\max}}}.$$

Here, τ_0 is a positive root of the cubic equation $\tau^3 + a\tau^2 + b\tau + c = 0$ such that $\omega_o(\tau_0) = \omega_+(\tau_0)$, where ω_o and ω_+ are the functions defined as in [3],

$$a = \frac{2(\eta_{\max} - 2\eta_{\min})}{\mu_{\min}+\mu_{\max}},\ b = \frac{\eta_{\min}(\eta_{\min} - 2\eta_{\max})}{\mu_{\min}\mu_{\max}},\ c = \frac{2\,\eta_{\min}^2\,\eta_{\max}}{\mu_{\min}\,\mu_{\max}(\mu_{\min}+\mu_{\max})}.$$

From Corollary 3.2 and Theorem 4.1 in [2], we can obtain the following corollary about the optimal parameters and the corresponding optimal semi-convergence factor for the PU method.

Corollary 3.4. *Consider the PU method for solving the singular saddle point problem* (1)*. Let Q be chosen as $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates A, and let $\mu_{\min}$ and $\mu_{\max}$ be the smallest and largest nonzero eigenvalues of the matrix $Q^\dagger B^T A^{-1} B$, respectively. Then the optimal parameters ω_o and τ_o are given by*

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

and the corresponding optimal semi-convergence factor $\nu(H(\omega_o, \tau_o))$ is

$$\nu(H(\omega_o, \tau_o)) = \frac{\sqrt{\mu_{\max}} - \sqrt{\mu_{\min}}}{\sqrt{\mu_{\max}} + \sqrt{\mu_{\min}}}.$$

4. Numerical results

In this section, we provide numerical experiments to examine the effectiveness of the PIU method studied in Section 3 for solving the singular saddle point problem (1). In Tables 3 to 8, *Iter* denotes the number of iteration steps, CPU denotes the elapsed CPU time in seconds, and CPU_1 denotes the elapsed CPU time excluding the computational time of $Q^\dagger$ for the singular case of Q or the Cholesky factorization time of Q for the nonsingular case of Q.

Example 4.1 ([21])**.** We consider the saddle point problem (1), in which

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

$$B = \begin{pmatrix} \hat{B} & \tilde{B} \end{pmatrix} = \begin{pmatrix} \hat{B} & b_1 & b_2 \end{pmatrix} \in \mathbb{R}^{2p^2 \times (p^2+2)}, \quad \hat{B} = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times p^2},$$

$$b_1 = \hat{B} \begin{pmatrix} e_{p^2/2} \\ 0 \end{pmatrix}, \quad b_2 = \hat{B} \begin{pmatrix} 0 \\ e_{p^2/2} \end{pmatrix}, \quad e_{p^2/2} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{p^2/2},$$

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

with $\otimes$ denoting the Kronecker product and $h = \frac{1}{p+1}$ the discretization mesh size. For this example, $m = 2p^2$ and $n = p^2 + 2$. Thus the total number of variables is $3p^2 + 2$. We choose the right hand side vector $(f^T, -g^T)^T$ such that the exact solution of the saddle point problem (1) is $(x_*^T, y_*^T)^T = (1, 1, \ldots, 1)^T \in \mathbb{R}^{m+n}$. Numerical results for this example are listed in Tables 3 to 5.

Example 4.2. Consider the Stokes equations of the following form: find $\mathbf{u}$ and v such that

$$\begin{cases} -\triangle \mathbf{u} + \nabla w = \mathbf{f} & \text{in } \Omega \\ \quad -\nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \end{cases} \tag{17}$$

where $\Omega = (0,1) \times (0,1)$, $\mathbf{u}$ is a vector-valued function representing the velocity, and w is a scalar function representing the pressure. The boundary conditions are $\mathbf{u} = (0,0)^T$ on the three fixed walls ($x = 0, \; y = 0, \; x = 1$) and $\mathbf{u} =$

$(1,0)^T$ on the moving wall ($y = 1$). Dividing Ω into a uniform grid with mesh size $h = \frac{1}{p}$ and discretizing (17) by using MAC (marker and cell) finite difference scheme [9, 12], the singular saddle point problem (1) is obtained, where $A \in \mathbb{R}^{2p(p-1)\times 2p(p-1)}$ is a symmetric positive definite matrix and $B = \begin{pmatrix} \hat{B} & \tilde{B} \end{pmatrix} \in \mathbb{R}^{2p(p-1)\times p^2}$ is a rank-deficient matrix of $\text{rank}(B) = p^2 - 1$ with $\hat{B} \in \mathbb{R}^{2p(p-1)\times(p^2-1)}$ and $\tilde{B} \in \mathbb{R}^{2p(p-1)}$. For this example, $m = 2p(p-1)$ and $n = p^2$. Thus the total number of variables is $3p^2 - 2p$. We also choose the right hand side vector $(f^T, -g^T)^T$ such that the exact solution of the saddle point problem (1) is $(x_*^T, y_*^T)^T = (1, 1, \ldots, 1)^T \in \mathbb{R}^{m+n}$. Numerical results for this example are listed in Tables 6 to 8.

The symmetric positive definite matrices P used in the PIU method are chosen in three different ways. The first choice is $P = A$ which reduces PIU to PU. The second choice is $P = (E-F)E^{-1}(E-F)^T$, where $A = E - F - F^T$ is a splitting of the symmetric positive definite matrix A with E a diagonal matrix and F a strictly lower triangular matrix. The last choice is $P = L_0 L_0^T$, where $A = L_0 L_0^T - R_0$ is a splitting of A obtained by an incomplete Cholesky factorization of A with no fill-in.

For the singular case of Q, the preconditioning matrices Q are chosen as in Table 1. Notice that $Q^\dagger$ is computed only once using the Matlab function *pinv* with a drop tolerance, and then it is stored for later use. For the nonsingular case of Q, the preconditioning matrices Q are chosen as in Table 2, where $\hat{Q}$ denotes a block diagonal matrix consisting of two submatrices $\hat{B}^T \hat{A}^{-1} \hat{B}$ and $\tilde{B}^T \tilde{B}$. The PIU algorithm for the nonsingular case of Q is the same as that for the singular case of Q except that Q^{-1} is used instead of $Q^\dagger$.

In all experiments, the initial vector was set to the zero vector. From now on, let $\|\cdot\|$ denote the L_2-norm. The iterations for the PIU method are terminated if the current iteration satisfies RES $< 10^{-6}$, where *RES* is defined by

$$\text{RES} = \frac{\sqrt{\|f - Ax_k - By_k\|^2 + \|g - B^T x_k\|^2}}{\sqrt{\|f\|^2 + \|g\|^2}}.$$

All numerical tests are carried out on a PC equipped with Intel Core i5-4570 3.2GHz CPU and 8GB RAM using Matlab R2014b. In Tables 3 to 8, we report the numerical results for two different values of m and n and four cases of the singular and nonsingular matrices Q. For the elapsed CPU time, every experiment is repeated five times. The best and the worst ones out of 5 CPU times are discarded, and then the average of the remaining 3 CPU times is reported in Tables 3 to 8.

The PIU method proposed in this paper depends on the parameters to be used. For the case of $P = A$, the parameters ω and τ are chosen as the optimal parameters stated in Corollary 3.4. For other cases of P, the parameters are chosen as the quasi-optimal parameters stated in Theorem 3.3. More specifically, we first compute $\mu_{\min}$, $\mu_{\max}$, $\eta_{\min}$ and $\eta_{\max}$, which can be easily computed using Matlab by computing only the smallest and largest nonzero eigenvalues.

And then the optimal parameters are computed using the formula in Corollary 3.4, and the quasi-optimal parameters can be computed using Matlab by solving the cubic equation in Theorem 3.3.

Table 1. Choices of the singular matrix Q.

Case Number	Q	Description
I	$B^TM^{-1}B$	$M = \mathrm{diag}(A)$
II	$B^TM^{-1}B$	$M = \mathrm{tridiag}(A)$

Table 2. Choices of the nonsingular matrix Q with $\hat{Q} = \mathrm{Diag}(\hat{B}^T\hat{A}^{-1}\hat{B}, \tilde{B}^T\tilde{B})$.

Case Number	Q	Description
III	$\hat{Q}$	$\hat{A} = \mathrm{diag}(A)$
IV	$\mathrm{tridiag}(\hat{Q})$	$\hat{A} = \mathrm{tridiag}(A)$

Table 3. Performance of the PIU method with $P = A$ for Example 4.1. (*Iter*: the number of iterations, ω and τ: optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	0.2489	0.1423	131	0.214	0.129	0.3307	0.1985	90	0.175	0.090
2048	1026	0.1956	0.1084	174	0.767	0.362	0.2635	0.1519	120	0.657	0.252
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	0.2489	0.1423	131	0.175	0.132	0.5692	2.9447	44	0.048	0.042
2048	1026	0.1956	0.1084	174	0.415	0.311	0.5115	3.3270	52	0.099	0.075

The PIU methods with nonsingular preconditioning matrix Q performs better than those with singular Q (see CPU in Tables 3 to 8). The reason is as follows: For nonsingular Q, $Q^{-1}b$ is computed using Cholesky factorization of Q without constructing Q^{-1} explicitly, so computational cost is cheap. For singular Q, $Q^\dagger\, b$ is computed using matrix-times-vector operation after constructing $Q^\dagger$ explicitly, which is very time-consuming. Note that $Q^\dagger$ is constructed using the singular value decomposition of Q, which requires a lot of computational amount. If we exclude the construction time of $Q^\dagger$ for singular Q and the Cholesky factorization time for nonsingular Q, then the PIU method with singular Q is comparable to that with nonsingular Q (see CPU_1 in Tables 3 to 8).

Table 4. Performance of the PIU method with $P = (E - F)E^{-1}(E - F)^T$ for Example 4.1. (*Iter*: the number of iterations, ω and τ: quasi-optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	1.7657	0.0626	219	0.131	0.046	1.8654	0.0588	232	0.135	0.050
2048	1026	1.8494	0.0377	332	0.582	0.177	1.9177	0.0347	360	0.597	0.192
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	1.7657	0.0626	219	0.131	0.053	0.9617	1.8293	238	0.056	0.031
2048	1026	1.8494	0.0377	332	0.345	0.149	0.9580	1.8482	318	0.119	0.065

Table 5. Performance of the PIU method with $P = L_0 L_0^T$ for Example 4.1. (*Iter*: the number of iterations, ω and τ: quasi-optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	1.3236	0.0910	176	0.122	0.037	1.4733	0.0811	174	0.121	0.036
2048	1026	1.4259	0.0568	250	0.551	0.146	1.5388	0.0489	259	0.555	0.150
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1152	578	1.3236	0.0910	176	0.104	0.040	0.7849	1.8970	177	0.041	0.024
2048	1026	1.4259	0.0568	250	0.257	0.113	0.7844	1.9042	236	0.088	0.047

Table 6. Performance of the PIU method with $P = A$ for Example 4.2. (*Iter*: the number of iterations, ω and τ: optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	0.2442	0.1392	132	0.205	0.125	0.3246	0.1939	89	0.169	0.086
1984	1024	0.1895	0.1047	177	0.775	0.370	0.2555	0.1466	119	0.677	0.248
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	0.2442	0.1392	132	0.176	0.130	0.0949	22.49	452	0.484	0.402
1984	1024	0.1895	0.1047	177	0.423	0.318	0.0707	29.94	630	1.177	1.013

For the nonsingular case of Q, Case IV gives better performance than Case III for Example 4.1 since Case IV of Q is much sparser than Case III of Q,

TABLE 7. Performance of the PIU method with $P = (E - F)E^{-1}(E - F)^T$ for Example 4.2. (*Iter*: the number of iterations, ω and τ: quasi-optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	1.7489	0.0672	239	0.132	0.052	1.8550	0.0634	255	0.138	0.055
1984	1024	1.8410	0.0399	368	0.607	0.202	1.9127	0.0369	399	0.649	0.220
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	1.7489	0.0672	239	0.130	0.057	0.3201	7.7888	792	0.235	0.128
1984	1024	1.8410	0.0399	368	0.367	0.164	0.3091	8.1193	1262	0.630	0.313

TABLE 8. Performance of the PIU method with $P = L_0 L_0^T$ for Example 4.2. (*Iter*: the number of iterations, ω and τ: quasi-optimal parameters, CPU: total CPU time, CPU_1: CPU time excluding computational time of $Q^\dagger$ or Choelsky factorization time of Q.)

		Case I of Q					Case II of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	1.3034	0.0971	197	0.120	0.040	1.4599	0.0871	196	0.126	0.043
1984	1024	1.4146	0.0599	301	0.575	0.170	1.5321	0.0517	293	0.595	0.166
		Case III of Q					Case IV of Q				
m	n	ω	τ	*Iter*	CPU	CPU_1	ω	τ	*Iter*	CPU	CPU_1
1104	576	1.3034	0.0971	197	0.107	0.047	1.6239	0.0391	462	0.140	0.074
1984	1024	1.4146	0.0599	301	0.298	0.136	1.6380	0.0221	982	0.494	0.251

whereas Case IV gives worse performance than Case III for Example 4.2 since the convergence rate for Case IV is too slow as compared with Case III. For the singular case of Q, $Q^\dagger$ has almost the same computational complexity between Case I and Case II, so performance (i.e., CPU time) depends on convergence rate (i.e., *Iter*). As can be seen in Tables 3 to 8, $P = L_0 L_0^T$ performs best of three forms of P, and $P = A$ provides the fastest convergence rate of three forms. However $P = A$ performs worst since its computational cost for each iteration is much higher than others.

For singular preconditioning matrix Q, we only considered the form of $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates P, which restricts the choices of Q. Further research is needed to study semi-convergence analysis for other forms of singular matrix Q, so that we can try many different kinds of Q in order to achieve the best possible performance of the PIU method.

5. Conclusions

In this paper, we provide semi-convergence analysis of the PIU method with singular preconditioner for solving singular saddle point problems. Numerical experiments show that the PIU method with nonsingular preconditioning matrix Q performs better than that with singular Q. The reason is that the PIU method with singular Q takes a lot of CPU time for constructing $Q^{\dagger}$. If we have an efficient algorithm for computing $Q^{\dagger}b$ for a given vector b, then the PIU method with singular Q is comparable to that with nonsingular Q.

For singular preconditioning matrix Q, we only considered the form of $Q = B^T M^{-1} B$, where M is a SPD matrix which approximates A or P, which restricts the choices of Q. Hence, future work will include semi-convergence analysis for other forms of singular matrix Q and development of an efficient algorithm for computing $Q^{\dagger}b$ for a given vector b.

Acknowledgements. The author would like to thank the anonymous referees for their useful comments and detailed suggestions for revising this manuscript, which greatly improved this paper.

References

[1] M. Arioli, I. S. Duff, and P. P. M. de Rijk, *On the augmented system approach to sparse least squares problems*, Numer. Math. **55** (1989), no. 6, 667–684.

[2] Z. Z. Bai, B. N. Parlett, and Z. Q. Wang, *On generalized successive overrelaxation methods for augmented linear systems*, Numer. Math. **102** (2005), no. 1, 1–38.

[3] Z. Z. Bai and Z. Q. Wang, *On parameterized inexact Uzawa methods for generalized saddle point problems*, Linear Algebra Appl. **428** (2008), no. 11-12, 2900–2932.

[4] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[5] Z. H. Cao, *On the convergence of general stationary linear iterative methods for singular linear systems*, SIAM J. Matrix Anal. Appl. **29** (2007), 1382–1388.

[6] Z. Chao and G. Chen, *Semi-convergence analysis of the Uzawa-SOR methods for singular saddle point problems*, Appl. Math. Lett. **35** (2014), 52–57.

[7] Z. Chao, N. Zhang, and Y. Lu, *Optimal parameters of the generalized symmetric SOR method for augmented systems*, J. Comput. Appl. Math. **266** (2014), 52–60.

[8] M. T. Darvishi and P. Hessari, *Symmetric SOR method for augmented systems*, Appl. Math. Comput. **183** (2006), no. 1, 409–415.

[9] H. C. Elman, *Preconditioning for the steady-state Navier-Stokes equations with low viscosity*, SIAM J. Sci. Comput. **20** (1999), no. 4, 1299–1316.

[10] H. C. Elman and D. J. Silvester, *Fast nonsymmetric iteration and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comput. **17** (1996), no. 1, 33–46.

[11] G. H. Golub, X. Wu, and J. Y. Yuan, *SOR-like methods for augmented systems*, BIT **41** (2001), no. 1, 71–85.

[12] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Phys. Fluids **8** (1965), 2182–2189.

[13] J. I. Li and T. Z. Huang, *The semi-convergence of generalized SSOR method for singular augmented systems*, High Performance Computing and Applications, Lecture Notes in Computer Science **5938** (2010), 230–235.

[14] S. Wright, *Stability of augmented system factorization in interior point methods*, SIAM J. Matrix Anal. Appl. **18** (1997), no. 1, 191–222.

[15] J. Y. Yuan and A. N. Iusem, *Preconditioned conjugate gradient methods for generalized least squares problem*, J. Comput. Appl. Math. **71** (1996), no. 2, 287–297.

[16] J. H. Yun, *Variants of the Uzawa method for saddle point problem*, Comput. Math. Appl. **65** (2013), no. 7, 1037–1046.

[17] ———, *Convergence of relaxation iterative methods for saddle point problem*, Appl. Math. Comput. **251** (2015), 65–80.

[18] G. F. Zhang and Q. H. Lu, *On generalized symmetric SOR method for augmented systems*, J. Comput. Appl. Math. **219** (2008), no. 1, 51–58.

[19] G. F. Zhang and S. S. Wang, *A generalization of parameterized inexact Uzawa method for singular saddle point problems*, Appl. Math. Comput. **219** (2013), no. 9, 4225–4231.

[20] N. Zhang and Y. Wei, *On the convergence of general stationary iterative methods for range-Hermitian singular linear systems*, Numer. Linear Algebra Appl. **17** (2010), no. 1, 139–154.

[21] B. Zheng, Z. Z. Bai, and X. Yang, *On semi-convergence of parameterized Uzawa methods for singular saddle point problems*, Linear Algebra Appl. **431** (2009), no. 5-7, 808–817.

[22] L. Zhou and N. Zhang, *Semi-convergence analysis of GMSSOR methods for singular saddle point problems*, Comput. Math. Appl. **68** (2014), no. 5, 596–605.

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