Performance Analysis of a Special GPIU Method for Singular Saddle Point Problems

Jae Heon Yun, Member, IAENG

Abstract—In this paper, we first provide semi-convergence analysis for a special GPIU (Generalized Parameterized Inexact Uzawa) method with singular preconditioners for solving singular saddle point problems. We next provide a methodology of how to choose nearly quasi-optimal parameters of the special GPIU method. Lastly, numerical experiments are carried out to examine the effectiveness of the special GPIU method with singular preconditioners by comparing its performance with that of other existing iterative methods for solving singular saddle point problems.

Index Terms—singular saddle point problem, GPIU method, semi-convergence, singular splitting, Moore-Penrose inverse.

I. INTRODUCTION

We consider the following large sparse augmented linear system

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

(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 rank-deficient matrix with $m \geq n$. In this case, the coefficient matrix of (1) is singular and so the problem (1) is called a singular saddle point problem. This type of problem appears in many different scientific applications, such as constrained optimization problems [13], [18], [30], the finite element approximation for solving the Navier-Stokes equation [10], the constrained least squares problems and generalized least squares problems [11], [24], and so on.

In case of $B$ being of full rank, many relaxation iterative methods have been proposed for solving the augmented linear system (1), e.g., SOR-like method [11], GSOR (Generalized SOR) method [2], PIU (Parameterized Inexact Uzawa) method [3], GPIU (Generalized Parameterized Inexact Uzawa) method [7], the modified SOR-like method [16], SSOR-like method [8], the modified SSOR-like method [17], Uzawa-SAOR method [21], GSSOR (Generalized SSOR) method [25], and MIAOR (Modified inexact AOR) method [22].


The purpose of this paper is to provide performance analysis of a special case of the GPIU 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 analysis of the basic iterative methods. In Section 3, we provide semi-convergence results for a special case of the GPIU method with singular preconditioners. In Section 4, we first provide a methodology of how to choose nearly quasi-optimal parameters of the special GPIU method, and then we provide numerical experiments in order to examine the effectiveness of the special GPIU method with singular preconditioners. Lastly, some conclusions are drawn.

II. 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 $x$. $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the minimum and maximum eigenvalues of the Hermitian matrix $H$, respectively. 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 $\rank(E^k) = \rank(E^{k+1})$ is called the index of $E$, and denoted by $k = \text{index}(E)$. In other words, $\text{index}(E)$ 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 $E^+$ which satisfies the
following equations
\[ E = EE' E, \quad E' = E' EE', \quad (EE')^T = EE', \quad (E'E)^T = E'E. \]

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
\[ x_{i+1} = (I - M^T A)x_i + M^T b \quad \text{for} \quad i = 0, 1, \ldots \] (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 \( \lim_{k \to \infty} \nu(T) < 1 \) [4].

Theorem 2.2 (5)): The iterative method (2) is semi-convergent if and only if \( \text{index}(M^T A) = 1, \nu(I - M^T A) < 1 \), and \( N(M^T A) = N(A) \), i.e., \( I - M^T A \) is semi-convergent and \( N(M^T A) = N(A) \).

III. SEMI-CONVERGENCE ANALYSIS OF A SPECIAL GPU Method

In this section, we provide semi-convergence analysis for a special case of the GPU method with singular preconditioners for solving the singular saddle point problem (1). Notice that Chen and Jiang [7] presented convergence analysis of the GPU method for nonsingular saddle point problems, and Zhang and Wang [27] provided semi-convergence analysis of the GPU method with nonsingular preconditioners for the singular saddle point problem (1).

Assume that the coefficient matrix \( A \) of (1) is split as
\[ A = \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} = D - \mathcal{L} - \mathcal{U}, \] (3)
where
\[ D = \begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix}, \quad \mathcal{L} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{U} = \begin{pmatrix} P - A & -B \\ 0 & Q \end{pmatrix}, \] (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 symmetric positive semi-definite matrix which approximates the approximated Schur complement matrix \( B^T P^{-1} B \). Let \( s \) be a real parameter and \( Q \) be chosen as \( Q = B^T M^{-1} B \), where \( M \) is a SPD matrix which approximates \( P \). Then a special case of the GPU method with the singular preconditioning matrix \( Q \), which is called the SGPIU method, is defined by
\[ \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = H_4(s) \begin{pmatrix} x_k \\ y_k \end{pmatrix} + M_4(s) \begin{pmatrix} f \\ -g \end{pmatrix}, \quad k = 0, 1, 2, \ldots \] (5)
where
\[ H_4(s) = I - (D + (s - 1)\mathcal{L})^\dagger A, \quad M_4(s) = (D + (s - 1)\mathcal{L})^\dagger. \]

By some manipulation, one obtains
\[ M_4(s) = \begin{pmatrix} P & 0 \\ (1-s)Q^1 B^T P^{-1} & 0 \end{pmatrix} \] (6)
and
\[ H_4(s) = \begin{pmatrix} Q^1 B^T (I_m - (s - 1)P^{-1} A) \quad I_n + (s - 1)Q^1 B^T P^{-1} B \end{pmatrix}. \] (7)

From (5), (6) and (7), the SGPIU method with the singular preconditioner \( Q \) for solving the singular saddle point problem (1) can be rewritten as

**Algorithm 1: SGPIU method with singular \( Q \)**

Choose \( s \) and initial vectors \( x_0, y_0 \)

For \( k = 0, 1, \ldots \), until convergence
\[ \begin{array}{l}
x_{k+1} = x_k + P^{-1}(f - Az_k - By_k) \\
y_{k+1} = y_k + Q^1 ((B^T x_{k+1} - g - sB^T (x_{k+1} - x_k)) \\
\end{array} \]

End For

If \( s = 0, P \) is replaced by \( \frac{1}{\mu} P \) with \( \mu \in (0, 2) \) and \( Q \) is replaced by \( \frac{1}{\mu} Q \), then the SGPIU method reduces to the PIU method. In particular, if \( s = 0, P = \frac{1}{\mu} A \) and \( Q \) is replaced by \( \frac{1}{\mu} Q \), then the SGPIU method reduces to the PU method.

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

Let
\[ B = W \Sigma V^*, \quad \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n} \] (8)
be the singular value decomposition of \( B \), where \( W \) and \( V \) are unitary matrices, \( \Sigma_r = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \) and \( \sigma_i \)'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 \( W_1 \in \mathbb{C}^{m \times r}, W_2 \in \mathbb{C}^{m \times (n-r)}, V_1 \in \mathbb{C}^{r \times n}, V_2 \in \mathbb{C}^{(n-r) \times n} \), respectively. Let us define an \((m+n) \times (m+n)\) unitary matrix \( \mathcal{P} \) as
\[ \mathcal{P} = \begin{pmatrix} W & 0 \\ 0 & V \end{pmatrix}. \] (9)

Let \( \hat{H}_4(s) = \mathcal{P}^* H_4(s) \mathcal{P} \). If we define \( \hat{P} = W^* P W, \hat{A} = W^* A W, \) and \( \hat{Q} = V^* Q V \). Since \( Q = B^T M^{-1} B \) and \( B = W \Sigma V^* \), one can obtain
\[ \hat{Q} = V^* Q V = \begin{pmatrix} \hat{Q}_1 & 0 \\ 0 & 0 \end{pmatrix}, \] (10)
where \( \hat{Q}_1 = \Sigma_r W_1^T M^{-1} W_1 \Sigma_r \) is an \( r \times r \) SPD matrix. Thus
\[ \hat{Q}^1 = V^* Q^1 V = \begin{pmatrix} Q_1^1 & 0 \\ 0 & 0 \end{pmatrix} \] (11)
and
\[ \hat{H}_4(s) = \begin{pmatrix} I_m - \hat{P}^{-1} A & -\hat{P}^{-1} Q^1 \Sigma_r \hat{P}^{-1} \Sigma_r \end{pmatrix}. \] (12)

If we let \( B_1 = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times r} \), then using (10) to (12)
\[ \hat{H}_4(s) \hat{P}_1 = \begin{pmatrix} I_m - \hat{P}^{-1} A & -\hat{P}^{-1} Q^1 \Sigma_r \hat{P}^{-1} \Sigma_r \end{pmatrix} \] (13)
and \( \hat{H}_4(s) \hat{P}_1 = \begin{pmatrix} I_m - \hat{P}^{-1} A & -\hat{P}^{-1} Q^1 \Sigma_r \hat{P}^{-1} \Sigma_r \end{pmatrix} \)
\( \hat{Q} \) is a matrix of the form
\[ \hat{Q} = \begin{pmatrix} \hat{Q}_1^1 & 0 \\ 0 & 0 \end{pmatrix} \] (14)
then \( \hat{H}(s) \) is the iteration matrix of the SGPIU method applied to the following nonsingular saddle point problem
\[ \begin{pmatrix} \hat{A} & B_1^T \\ -B_1^T & 0 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} \hat{f} \\ -\hat{g} \end{pmatrix} \] (15)
with the preconditioning matrix \( \hat{Q}_1 \) and \( \hat{P} \) as an approximation of \( \hat{A} \).

(Advance online publication: 23 August 2017)
Before proceeding to semi-convergence result of the SGPIU method for solving the singular saddle point problem (1), we first consider convergence of the SGPIU method for solving the nonsingular saddle point problem (15) whose iteration matrix is $\hat{H}(s)$. Note that $B_1$ has full column rank $r$. From the convergence analysis described in [7, 27, one can easily obtain the following three lemmas.

Lemma 3.1: Let $\lambda$ be an eigenvalue of $\hat{H}(s)$ and $\left( \frac{x}{y} \right)$ be the corresponding eigenvector. Then $\lambda \neq 1$ and $\hat{x} \neq 0$.

Lemma 3.2: Let $\lambda$ be an eigenvalue of $\hat{H}(s)$ and $\left( \frac{x}{z} \right)$ be the corresponding eigenvector. Then $\lambda$ satisfies the following quadratic equation

$$\lambda^2 + \frac{\beta - 2\alpha + (1 - s)\gamma}{\alpha} \lambda + \frac{\alpha - \beta + s\gamma}{\alpha} = 0,$$

where $\hat{\alpha} = \frac{x^T P x}{x^T x}$, $\hat{\beta} = \frac{x^T A z}{x^T x}$ and $\hat{\gamma} = \frac{x^T B_1 Q_1^{-1} B_1^T x}{x^T x}$.

Lemma 3.3: Let $\lambda$ be an eigenvalue of $\hat{H}(s)$ and $\left( \frac{x}{z} \right)$ be the corresponding eigenvector. Then $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta$$

and $\gamma > 0$.

Proof: Since $\hat{A}_s(s)$ and $\hat{A}_s(s)$ are similar, $\lambda$ is an eigenvalue of $\hat{A}_s(s)$. Since $\lambda \neq 1$, from (13) and (14) $\lambda$ is also an eigenvalue of $\hat{H}(s)$. Let $\left( \frac{x}{y} \right)$ be an eigenvector of $\hat{H}_s(s)$ corresponding to the eigenvalue $\lambda$. Then $\hat{z} = \frac{z}{x}$ is the eigenvector of $\hat{H}(s)$ corresponding to the eigenvalue $\lambda$. From Lemma 3.1, $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta$$

and $\gamma > 0$.

Theorem 3.4: Let $\lambda \neq 1$ be an eigenvalue of $\hat{A}_s(s)$ and $\left( \frac{x}{y} \right)$ be the corresponding eigenvector. Then $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta$$

and $\gamma > 0$.

Proof: Since $\hat{A}_s(s)$ and $\hat{A}_s(s)$ are similar, $\lambda$ is an eigenvalue of $\hat{A}_s(s)$. Since $\lambda \neq 1$, from (13) and (14) $\lambda$ is also an eigenvalue of $\hat{H}(s)$. Let $\left( \frac{x}{y} \right)$ be an eigenvector of $\hat{H}_s(s)$ corresponding to the eigenvalue $\lambda$. Then $\hat{z} = \frac{z}{x}$ is the eigenvector of $\hat{H}(s)$ corresponding to the eigenvalue $\lambda$. By the relation $\hat{H}_s(s) = P^T \hat{A}_s(s) P$, it is easy to show that $\hat{x} = \hat{W}^T x$ and $\hat{y} = \hat{V}^T y$. From Lemma 3.3, $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta$$

and $\gamma > 0$. Since $\hat{x} = \hat{W}^T x$, $\hat{\alpha} = \alpha$ and $\hat{\beta} = \beta$ are immediately obtained. On the other hand,

$$x^T B_1^T B_1 x = x^T B_1^T \hat{Q}_1 \hat{Q}_1^{-1} B_1^T B_1 x = x^T B_1^T \hat{Q}_1^{-1} B_1^T x = x^T B_1 \hat{Q}_1^{-1} B_1^T \hat{z}.$$

From (16), $\hat{\gamma} = \gamma$ is also obtained. Therefore, the proof is complete.

Corollary 3.5: Let $\lambda$ be an eigenvalue of $\hat{H}(s)$ and $\left( \frac{x}{y} \right)$ be an eigenvector of $\hat{H}_s(s)$ corresponding to the eigenvalue $\lambda$. Then $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta$$

and $\gamma > 0$, where $\alpha = \frac{x^T P x}{x^T x}$, $\beta = \frac{x^T A z}{x^T x}$ and $\gamma = \frac{x^T B_1^T B_1 x}{x^T x}$.

Proof: Since $\lambda \neq 1$ from Lemma 3.1, this corollary follows from Theorem 3.4.

The following theorem shows that the condition $\gamma > 0$ in (17) can be omitted.

Theorem 3.6: Let $\lambda$ be an eigenvalue of $\hat{H}(s)$ and $\left( \frac{x}{y} \right)$ be an eigenvector of $\hat{H}_s(s)$ corresponding to the eigenvalue $\lambda$. Then $|\lambda| < 1$ if and only if

$$\gamma - 4\alpha + 2\beta < 2s\gamma < 2\beta,$$

where $\alpha = \frac{x^T P x}{x^T x}$, $\beta = \frac{x^T A z}{x^T x}$ and $\gamma = \frac{x^T B_1^T B_1 x}{x^T x}$.

Proof: From Theorem 3.4, it was shown that $\alpha = \beta > 0$ and $\gamma = \gamma > 0$. If $\gamma = 0$ (i.e., $x \in N(B_1^T)$), then Lemma 3.2 implies that $\lambda$ satisfies the following quadratic equation

$$\lambda^2 - (\gamma - 2\beta)\lambda + 1 - \frac{\beta}{\alpha} = 0.$$
Q for solving the singular saddle point problem (1) is semi-convergent if the following inequality holds
\[
\frac{1}{2} - \frac{\lambda_{\min}(\frac{2}{\tau} \hat{P} - A)}{\tau \rho(\hat{B}Q^{-1}B^T)} < s < \frac{\lambda_{\min}(A)}{\tau \rho(\hat{B}Q^{-1}B^T)}.
\]

**Corollary 3.10:** Let \( P = \frac{1}{\omega} A, \hat{Q} = B^T M^{-1} B \) and \( Q = \frac{1}{\omega} Q, \) where \( 0 < \omega < 2, \tau > 0 \) and \( M \) is a SPD matrix which approximates \( P. \) Then the SGPIU method with the singular \( Q \) for solving the singular saddle point problem (1) is semi-convergent if the following inequality holds
\[
\frac{1}{2} - \left( \frac{2}{\omega} - 1 \right) \frac{\lambda_{\min}(A)}{\tau \rho(\hat{B}Q^{-1}B^T)} < s < \frac{\lambda_{\min}(A)}{\tau \rho(\hat{B}Q^{-1}B^T)}.
\]

**Proof:** Since \( 0 < \omega < 2, 2 P - A \) is symmetric positive definite. Hence this corollary follows from Corollary 3.9.

### IV. NUMERICAL RESULTS

In this section, we first provide a methodology of how to choose nearly quasi-optimal parameters of the special GPUI method, and then we provide numerical experiments in order to examine the effectiveness of the SGPIU method with singular preconditioners for solving the singular saddle point problem (1). Performance of the SGPIU method with singular preconditioners is compared with that of the SGPIU method with nonsingular preconditioners and the PU or PIU methods with singular or nonsingular preconditioners.

In Tables II to V, **Iter** denotes the number of iteration steps, and **CPU** denotes the elapsed CPU time in seconds excluding the computational time of \( Q^T \) for the singular case of \( Q \) or the Cholesky factorization time of \( Q \) for the nonsingular case of \( Q. \) In all experiments, the right hand side vector \((f^T, -g^T)^T \in \mathbb{R}^{m+n}\) 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. All iterations for the singular saddle point problem are terminated if the current iteration satisfies \( \text{RES} < 10^{-6}, \) where **RES** is defined by
\[
\text{RES} = \frac{\sqrt{\|f - Ax_k - \hat{B}y_k\|^2 + \|g - B^T x_k\|^2}}{\sqrt{|f|^2 + \|g\|^2}},
\]
where \( \|\cdot\| \) denotes the \( L_2 \)-norm.

All numerical tests are carried out on a PC equipped with Intel Core i5-4570 3.2Ghz CPU and 8GB RAM using MATLAB R2014b. 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 II to V.

**Example 4.1 (29b):** We consider the saddle point problem (1), in which
\[
A = \begin{pmatrix} I \otimes T & T \otimes I \\ I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times 2p^2},
\]
\[
B = (\hat{B} \quad \hat{B}) = (\hat{B} \quad b_1 \quad b_2) \in \mathbb{R}^{2p^2 \times (p^2+2)},
\]
\[
\hat{B} = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times p^2}, \quad b_1 = \hat{B} \begin{pmatrix} e_{p^2/2} \\ 0 \end{pmatrix},
\]
\[
b_2 = \begin{pmatrix} 0 \\ e_{p^2/2} \end{pmatrix}, \quad c_{p^2/2} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{p^2/2},
\]
\[
T = \frac{1}{h^2} \cdot \text{tridiag}(-1, 1, -1) \in \mathbb{R}^{p \times p},
\]
\[
F = \frac{1}{h^2} \cdot \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{p \times p},
\]

with \( \otimes \) denoting the Kronecker product and \( h = \frac{1}{20} \) 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. Numerical results for this example are listed in Tables II and III.

**Example 4.2:** Consider the Stokes equations of the following form: find \( u \) and \( v \) such that
\[
\begin{align*}
-\Delta u + \nabla w &= f \quad \text{in } \Omega, \\
-\nabla \cdot u &= 0 \quad \text{in } \Omega,
\end{align*}
\]
where \( \Omega = (0, 1) \times (0, 1), \) \( u \) is a vector-valued function representing the velocity, and \( w \) is a scalar function representing the pressure. The boundary conditions are \( u = (0, 0)^T \) on the three fixed walls \((x = 0, y = 0, x = 1)\) and \( u = (1,0)^T \) on the moving wall \((y = 1). \) Dividing \( \Omega \) into a uniform grid with mesh size \( h = \frac{1}{5} \) and discretizing (20) by using MAC (marker and cell) finite difference scheme \([9, 12], \) the singular saddle point problem (1) is obtained, where \( A \in \mathbb{R}^{3p^2(p-1) \times 3p^2(p-1)} \) is a symmetric positive definite matrix and \( B = (\hat{B} \quad \hat{B}) \in \mathbb{R}^{3p^2(p-1) \times p^2} \) is a rank-deficient matrix of \( \text{rank}(B) = p^2 - 1 \) with \( \hat{B} \in \mathbb{R}^{2p^2(p-1) \times (p^2-1)} \) and \( \hat{B} \in \mathbb{R}^{2p^2(p-1)}. \) For this example, \( m = 2p(p-1) \) and \( n = p^2. \) Thus the total number of variables is 3p^2 - 2p. Numerical results for this example are listed in Tables IV and V.

For the SGPIU method, the symmetric positive definite matrices \( P \) are chosen as \( P = \frac{1}{\omega} \hat{P} \) with a positive parameter \( \omega \in (0, 2) \) in three different ways. The first choice is \( P = A, \) the second choice is \( \hat{P} = (A - F)E^{-1}(A - F)^T, \) 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, and the third choice is \( \hat{P} = L_0 L_0^T, \) 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. The singular or nonsingular preconditioning matrices \( Q \) are chosen as \( Q = \frac{1}{\omega} \hat{Q} \) with a positive parameter \( \tau, \) where the matrices \( Q \) are chosen as in Table I. In Table I, \( \text{Diag}(B^T A^{-1} B, B^T B) \) denotes a block diagonal matrix consisting of two submatrices \( B^T A^{-1} B \) and \( B^T B. \) The SGPIU 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 \( Q^T. \)

For these choices of \( P \) and \( Q, \) the SGPIU method with \( s = 0 \) reduces to the PU method for \( P = \frac{1}{\omega} A \) or the PIU method for other two choices of \( P. \) For \( s = 0, \) the parameters \( \omega \) and \( \tau \) are chosen as the optimal or quasi-optimal parameters which are computed using the formulas given in \([28] \) or \([19], \) respectively (see data reported in the first line of Tables II - V for each case of \( P). \) For \( s \neq 0, \) the parameters are chosen in two different ways: One choice is that \( \omega \) and \( \tau \) are chosen first as the optimal or quasi-optimal parameters and then \( s \) is chosen as the best one by tries (see data reported in the second line of Tables II - V for each case of \( P). \) and the other choice is the experimentally chosen optimal parameters \( s, \omega \) and \( \tau \) (see data reported in the third line of Tables II - V for each case of \( P). \) For singular matrix \( Q, Q^T \) is computed only once using the Matlab function \text{pinv} \) with a drop tolerance \( 10^{-13}, \) and then it is stored for later use. For nonsingular matrix \( Q, \) the Cholesky factorization of \( Q \) is computed only.

(Advance online publication: 23 August 2017)
TABLE I

<table>
<thead>
<tr>
<th>Case Number</th>
<th>$Q$</th>
<th>Description</th>
<th>Property of $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$B^T M^{-1} B$</td>
<td>$M = \text{diag}(A)$</td>
<td>singular</td>
</tr>
<tr>
<td>II</td>
<td>$B^T M^{-1} B$</td>
<td>$M = \text{diag}(A)$</td>
<td>singular</td>
</tr>
<tr>
<td>III</td>
<td>Diag($B^T A^{-1} B$, $B^T B$)</td>
<td>$A = \text{diag}(A)$</td>
<td>nonsingular</td>
</tr>
<tr>
<td>IV</td>
<td>tridiag(Diag($B^T A^{-1} B$, $B^T B$))</td>
<td>$A = \text{diag}(A)$</td>
<td>nonsingular</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>$m$</th>
<th>$n$</th>
<th>$\hat{P}$</th>
<th>Case I of $Q$</th>
<th>Case II of $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$s$</td>
<td>$\omega$</td>
</tr>
<tr>
<td>1152</td>
<td>578</td>
<td>$0.02488$</td>
<td>0</td>
<td>0.1423</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.04$</td>
<td>-0.02</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.2$</td>
<td>-0.4</td>
<td>0.626</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.7657$</td>
<td>-0.65</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.1956$</td>
<td>0</td>
<td>0.1084</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.21$</td>
<td>-0.04</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.8494$</td>
<td>-0.35</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.4259$</td>
<td>-0.25</td>
<td>0.15</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>$m$</th>
<th>$n$</th>
<th>$\hat{P}$</th>
<th>Case III of $Q$</th>
<th>Case IV of $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$s$</td>
<td>$\omega$</td>
</tr>
<tr>
<td>1152</td>
<td>578</td>
<td>$0.02488$</td>
<td>0</td>
<td>0.1423</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.04$</td>
<td>-0.02</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.2$</td>
<td>-0.4</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.7657$</td>
<td>-0.65</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.1956$</td>
<td>0</td>
<td>0.1084</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.21$</td>
<td>-0.04</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.8494$</td>
<td>-0.35</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1.4259$</td>
<td>-0.25</td>
<td>0.15</td>
</tr>
</tbody>
</table>

For singular $Q$, $Q^T b$ is computed using matrix-times-vector operation after constructing $Q^T$ explicitly, which is very time-consuming. For nonsingular $Q$, $Q^{-1} b$ is computed using the forward and backward substitutions after constructing the Cholesky factorization of $Q$ explicitly. As can be seen in Tables II to V, $P = \frac{1}{2} L_0 L_0^T$ provides better performance and faster convergence rate than other two cases of $P$. From Tables II to V, it can also be seen that the SGPIU method with an appropriately chosen number $s$ and optimal or quasi-optimal parameters $\omega$ and $\tau$ performs better than the PU or PIU methods with optimal or quasi-optimal parameters $\omega$ and $\tau$ (i.e., the SGPIU methods with $s = 0$).

More specifically, when $P = \frac{1}{2} A$, SGPIU method with an appropriately chosen number $s$ corresponding to optimal parameters $\omega$ and $\tau$ of the PU method performs significantly better than PU method with the optimal parameters $\omega$ and $\tau$ for all types of preconditioners $Q$ used in this paper. When $P = \frac{1}{2} (E - F) E^{-1} (E - F)^T$ or $\frac{1}{2} L_0 L_0^T$, SGPIU method with an appropriately chosen number $s$ corresponding to quasi-optimal parameters $\omega$ and $\tau$ of the PIU method performs much better than PIU method with the quasi-optimal parameters $\omega$ and $\tau$ for the preconditioners $Q$ of types I and III. Clearly, the SGPIU method with experimentally chosen optimal parameters $s$, $\omega$ and $\tau$ performs best. However, we do not have a formula for finding optimal parameters of the SGPIU method, which should be done in the future work.

(Advance online publication: 23 August 2017)
V. Conclusion

In this paper, we provided semi-convergence analysis of the SGPIU method with singular preconditioners for solving singular saddle point problems. Numerical experiments show that the SGPIU method with an appropriately chosen number $s$ and optimal or quasi-optimal parameters $\omega$ and $\tau$ performs better than the PU or PIU methods with optimal or quasi-optimal parameters $\omega$ and $\tau$. More specifically, when $P = \frac{1}{3}A$, SGPIU method with an appropriately chosen number $s$ corresponding to optimal parameters $\omega$ and $\tau$ of the PU method performs significantly better than the PU method for all types of preconditioners $Q$ used in this paper. When $P = \frac{1}{3}(E - F)E^{-1}(E - F)^T$ or $\frac{1}{3}L_0L_0^T$, SGPIU method with an appropriately chosen number $s$ corresponding to quasi-optimal parameters $\omega$ and $\tau$ of the PIU method performs about twice faster than the PIU method for the preconditioners $Q$ of types I and III. It means that the methodology of choosing an appropriate value of $s$ corresponding to the optimal or quasi-optimal parameters $\omega$ and $\tau$ of the PU or PIU methods works quite well for the SGPIU method.

It is clear that the SGPIU method with experimentally chosen optimal parameters $s$, $\omega$ and $\tau$ performs best. So, further research for finding optimal parameters of the SGPIU method will be done in the future work. The SGPIU method with singular preconditioners performs rather well as compared with that with nonsingular preconditioners. However, the SGPIU method with singular preconditioners $Q$ requires the computation of $Q^1b$ for a given vector $b$, which is very time-consuming. Future work will also include how to compute $Q^1b$ efficiently for a given vector $b$.

Acknowledgment

The author is grateful to the anonymous reviewers for their valuable comments and suggestions which improved the quality and the clarity of the paper.

References


(Advance online publication: 23 August 2017)


