Journal of Computational Mathematics
Vol.32, No.3, 2014, 297–305.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1401-CR7

ON AUGMENTED LAGRANGIAN METHODS FOR SADDLE-POINT LINEAR SYSTEMS WITH SINGULAR OR SEMIDEFINITE $(1,1)$ BLOCKS*

Tatiana S. Martynova
Computing Center, Southern Federal University, Rostov-on-Don, Russia
Email: martynova@sfedu.ru

Abstract

An effective algorithm for solving large saddle-point linear systems, presented by Krukier et al., is applied to the constrained optimization problems. This method is a modification of skew-Hermitian triangular splitting iteration methods. We consider the saddle-point linear systems with singular or semidefinite $(1,1)$ blocks. Moreover, this method is applied to precondition the GMRES. Numerical results have confirmed the effectiveness of the method and showed that the new method can produce high-quality preconditioners for the Krylov subspace methods for solving large sparse saddle-point linear systems.

Mathematics subject classification: 65F10, 65F50.
Key words: Hermitian and skew-Hermitian splitting, Saddle-point linear system, Constrained optimization, Krylov subspace method.

1. Introduction

Large sparse linear systems of saddle-point type arise in different applications. In many cases the matrix of such linear system has zero $(2,2)$ block. Consider iterative solution of the large sparse indefinite system of linear equations in block-structured form

$$\begin{pmatrix} M & E^T \\ E & 0 \end{pmatrix} \begin{pmatrix} u \\ \mu \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}.$$

The matrix $M \in \mathbb{R}^{p\times p}$ is assumed to be symmetric positive semidefinite, the matrix $E \in \mathbb{R}^{q\times p}$ has full row rank, $q \le p$, $f \in \mathbb{R}^p$ and $g \in \mathbb{R}^q$ are two given vectors. Here E^T denotes the transpose of the matrix E. We assume that matrices M and E have no nontrivial null vectors. It is the condition that guarantees the existence and uniqueness of the solution; see [2].

This linear system corresponds to minimizing the quadratic objective functional $J(u) \equiv \frac{1}{2}u^T Mu - u^T f$, subject to q linear constraints $Eu = g$. The Lagrangian functional $\mathcal{L}(u,\mu) = J(u)+\mu^T(Eu-g)$ is associated with this constrained minimization problem, μ denotes the vector of Lagrange multipliers. Here M is the Hessian of the quadratic function to be minimized, and E is the Jacobian of the linear constraints [10, 11, 14].

A number of solvers have been developed for saddle-point linear systems in recent years, for example, projection methods [8], null space methods [1], HSS-like methods [3, 4, 6, 9], generalized successive overrelaxation methods [7], SSOR-like methods [18], and so on.

* Received August 30, 2013 / Revised version received December 16, 2013 / Accepted January 15, 2014 / Published online May 22, 2014 /

We consider cases when the $(1,1)$ block in the linear system is semidefinite or singular. Frequently the singularity appears in the form of semidefiniteness. Our approach is based on augmentation of the $(1,1)$ block. We replace the $(1,1)$ block by a matrix that is much easier to invert. We employ the augmented Lagrangian method, the matrix M will be replaced by a positive definite matrix, and iterative methods can be applied to solve the augmented linear system. We replace the quadratic objective functional $J(u)$ by a regularized functional $J_\gamma(u) \equiv J(u)+\gamma\|Eu-g\|_W^2$, where $\gamma > 0$ is a parameter, $W = W^T > 0$ is a weighting matrix of size q, and $\|Eu-g\|_W^2 = (Eu-g)^T W(Eu-g)$. The augmented Lagrangian $\mathcal{L}_\gamma(u,\mu)$ associated with the minimization of $J_\gamma(u)$ is defined as [13]:

$$\mathcal{L}_\gamma(u,\mu) = \mathcal{L}(u,\mu) + \frac{\gamma}{2}\|Eu-g\|_W^2. \tag{1.1}$$

In the augmented saddle-point linear system, matrix M will be replaced by the matrix $\widetilde{M} \equiv M + \gamma E^T W E$ that will be positive definite for $\gamma > 0$. Applying the first derivative test to determine the saddle point of $\mathcal{L}_\gamma(u,\mu)$ yields the following linear system [13,14]:

$$\begin{pmatrix} \widetilde{M} & E^T \\ E & 0 \end{pmatrix} \begin{pmatrix} u \\ \mu \end{pmatrix} = \begin{pmatrix} f + \gamma E^T W g \\ g \end{pmatrix}. \tag{1.2}$$

Clearly, this linear system has precisely the same solution as the original one.

Recently in [15] the authors presented generalized skew-Hermitian triangular splitting iteration method (GSTS) for solving large non-Hermitian linear systems. The GSTS iteration method reduces to the skew-Hermitian triangular splitting (STS) iteration method studied in [17] and the product-type skew-Hermitian triangular splitting (PSTS) iteration method established in [16]. Then authors applied the GSTS iteration method to solve non-Hermitian saddle-point linear systems, proved its convergence under suitable restrictions on the iteration parameters, and implemented the method by solving the Stokes problem [15].

For iteratively solving linear systems arising in the constrained optimization problems, we use the GSOR [7] and the GSTS iteration methods. Numerical results show that the GSTS iteration method is effective for solving saddle-point linear systems arising in the constrained optimization problems with singular or semidefinite $(1,1)$ blocks by the augmented Lagrangian method.

2. Iteration Methods

We can rewrite the saddle-point linear system into an equivalent non-symmetric form [10,11]

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \tag{2.1a}$$

where

$$\mathcal{A} = \begin{pmatrix} M & E^T \\ -E & 0 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} u \\ \mu \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} f \\ -g \end{pmatrix}. \tag{2.1b}$$

The matrix $\mathcal{A} \in \mathbb{R}^{(p+q)\times(p+q)}$ is positive stable now, that is, the eigenvalues of $\mathcal{A}$ have positive real parts [9,11]. Analogous to [5] the matrix $\mathcal{A}$ can be split into its symmetric and skew-symmetric parts as

$$\mathcal{A} = \mathcal{A}_{\mathcal{H}} + \mathcal{A}_{\mathcal{S}}, \tag{2.2}$$

where

$$\mathcal{A}_{\mathcal{H}} = \frac{1}{2}(\mathcal{A} + \mathcal{A}^T), \quad \mathcal{A}_{\mathcal{S}} = \frac{1}{2}(\mathcal{A} - \mathcal{A}^T). \tag{2.3}$$

The symmetric and the skew-symmetric parts of the matrix $\mathcal{A}$ are given by

$$\mathcal{A}_{\mathcal{H}} = \begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{A}_{\mathcal{S}} = \begin{pmatrix} 0 & E^T \\ -E & 0 \end{pmatrix}, \tag{2.4}$$

and the skew-symmetric part $\mathcal{A}_{\mathcal{S}}$ can be split into

$$\mathcal{A}_{\mathcal{S}} = \mathcal{K}_L + \mathcal{K}_U = \begin{pmatrix} 0 & 0 \\ -E & 0 \end{pmatrix} + \begin{pmatrix} 0 & E^T \\ 0 & 0 \end{pmatrix}, \tag{2.5}$$

where 0 is a zero matrix with suitable dimension, $\mathcal{K}_L$ and $\mathcal{K}_U$ are, respectively, the strictly lower- and the strictly upper- triangular parts of $\mathcal{A}_{\mathcal{S}}$. Note that $\mathcal{K}_L = -\mathcal{K}_U^T$. Based on these splittings, in [15] the authors established the generalized skew-Hermitian triangular splitting (GSTS) iteration method for solving the saddle-point linear systems. Note that the GSTS iteration method can be applied when $(2,2)$ block of the linear system is not equal to zero.

Let the matrix $\mathcal{B}_C$ be defined as

$$\mathcal{B}_C = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}, \tag{2.6}$$

where B_1 and B_2 are symmetric and nonsingular matrices.

We have the following iteration sequence for the approximate solution $\mathbf{x}^{(k)}$ of the saddle-point linear systems:

$$\mathbf{x}^{(k+1)} = \mathcal{G}(\omega_1, \omega_2, \tau)\mathbf{x}^{(k)} + \tau\mathcal{B}(\omega_1, \omega_2)^{-1}\mathbf{b}, \tag{2.7a}$$

where

$$\mathcal{G}(\omega_1, \omega_2, \tau) = \mathcal{B}(\omega_1, \omega_2)^{-1}(\mathcal{B}(\omega_1, \omega_2) - \tau\mathcal{A}) = I - \tau\mathcal{B}(\omega_1, \omega_2)^{-1}\mathcal{A}, \tag{2.7b}$$

and τ is a positive parameter. Here, the matrix $\mathcal{B}(\omega_1, \omega_2)$ is defined as

$$\mathcal{B}(\omega_1, \omega_2) = (\mathcal{B}_C + \omega_1\mathcal{K}_L)\mathcal{B}_C^{-1}(\mathcal{B}_C + \omega_2\mathcal{K}_U), \tag{2.7c}$$

where ω_1 and ω_2 are nonnegative acceleration parameters and, at least, one of them is nonzero.

Method 2.1. *(The GSTS Iteration Method)* ([15]) *Given an initial guess $\boldsymbol{x}^{(0)} = (u^{(0)}, \mu^{(0)}) \in \mathbb{R}^{(p+q)}$, and a positive iteration parameter τ, for $k = 0, 1, 2, ...$ until $\{\boldsymbol{x}^{(k)}\}$ convergence, compute*

$$\begin{cases} B_2\mu^{(k+1)} = B_2\mu^{(k)} + \tau[\omega_1 E B_1^{-1}(f - Mu^{(k)} - E^T\mu^{(k)}) + Eu^{(k)} - g], \\ B_1 u^{(k+1)} = B_1 u^{(k)} - \tau M u^{(k)} + E^T[(\omega_2 - \tau)\mu^{(k)} - \omega_2\mu^{(k+1)}] + \tau f, \end{cases} \tag{2.8}$$

where ω_1 and ω_2 are nonnegative acceleration parameters and, at least, one of them is nonzero.

We consider two cases for the GSTS iteration method according to the different choices of the matrix B_2, and obtain effective solvers for the saddle-point linear systems. In actual implementations we choose $B_1 = \widetilde{M}$.

Method 2.2. *(The GSOR Iteration Method)* ([7])

Let $S \in \mathbb{R}^{q\times q}$ be a nonsingular and symmetric matrix. Given an initial guess $\boldsymbol{x}^{(0)} = (u^{(0)}, \mu^{(0)}) \in \mathbb{R}^{(p+q)}$, for $k = 0, 1, 2, ...$ until $\{\boldsymbol{x}^{(k)}\}$ convergence, compute

$$\begin{cases} u^{(k+1)} = (1-\omega)u^{(k)} + \omega M^{-1}(f - E^T \mu^{(k)}), \\ \mu^{(k+1)} = \mu^{(k)} + \tau S^{-1}(Eu^{(k+1)} - g). \end{cases}$$

Here, S is an approximate (preconditioning) matrix of the Schur complement matrix $EM^{-1}E^T$, and $\omega, \tau \neq 0$ are two relaxation factors.

3. Practical Choice of the Parameter γ and the Weighting Matrix W

We need to choose the positive parameter γ and the weighting matrix W so that augmented linear system is easier to solve than the original one. There is goal like minimizing the condition number of the augmented matrix, maintaining sparsity, obtaining positive definiteness of the $(1, 1)$ block, and other considerations, not all of which can be satisfied simultaneously. One of the difficulties in forming $\widetilde{M}$ is the loss of sparsity, which could occur if M and E^T have different sparsity patterns. Moreover, the matrix E^TE is usually not as sparse as the matrix E^T. When $W = I$ is used, the choice $\gamma = \|M\|_2/\|E\|_2^2$ has been found to perform well in practice, in the sense that the condition number of both $(1, 1)$ block and the whole coefficient matrix are approximately minimized [11]. The point made in [14] was that for the positive semidefine M and the choice $W = I$ there exists a range of (typically moderate) values of γ for which the spectral properties and the conditioning of the associated matrix $\mathcal{A}(\gamma)$ are possibly better than those of the original matrix $\mathcal{A}$. If the $(1, 1)$ block is singular or ill-conditioned, we should seek a value of γ that is large enough so as to eliminate the effect of the ill-conditioning or the singularity of the matrix M, but not too large, so as to avoid the effect of the singular matrix E^TE [14]. The condition number of the matrix $M + \gamma E^TWE$ grows like a (possibly large) multiple of γ for the equality-constrained least squares problems. In practice, for the large values of γ the coefficient matrix $M + \gamma E^TWE$ is dominated by the term γE^TWE, and approximate solutions of the linear system are difficult to obtain.

The choice of the weighting matrix W is highly problem-dependent. For example, if we consider solving the steady-state Oseen equations, a good choice of W is a pressure mass matrix, but in practice we should use the main diagonal of this matrix instead, in order to maintain the sparsity in the matrix $\widetilde{M}$ [12]. If the weighting matrix W is diagonal or block diagonal with small blocks, then the matrix $M + \gamma E^TWE$ is also going to be sparse.

4. Numerical Results

In this section, we use examples to examine the effectiveness of the GSTS iteration method for solving the constrained optimization problems with different iteration parameters from aspects of number of iteration steps (denoted by "IT"), elapsed CPU time in seconds (denoted by "CPU") and norm of absolute residual vectors (denoted by "RES") or the norm of absolute error vectors (denoted by "ERR"). Here, the "RES" and the "ERR" are defined to be

$$RES := \sqrt{||f - Mu^{(k)} - E^T\mu^{(k)}||_2^2 + ||g - Eu^{(k)}||_2^2},$$

and

$$ERR := \frac{\sqrt{||u^{(k)} - u^*||_2^2 + ||\mu^{(k)} - \mu^*||_2^2}}{\sqrt{||u^{(0)} - u^*||_2^2 + ||\mu^{(0)} - \mu^*||_2^2}},$$

respectively, where $(u^{(k)^T}, \mu^{(k)^T})^T$ is the final approximate solution. In actual computations we choose the right-hand-side vector $(f^T, g^T)^T \in \mathbb{R}^{p+q}$ such that the exact solutions of the linear systems are $((u^*)^T, (\mu^*)^T)^T = (1, 1, ..., 1)^T \in \mathbb{R}^{p+q}$. All runs are started from the zero vector and terminated if the current iterations satisfy $RES \leq 10^{-7}$ or if the number of iteration steps exceed 1000 (this case is denoted by the symbol "-" in the tables). Numerical experiments are performed in MATLAB with a machine precision 10^{-16}.

We test two cases for the GSTS iteration method: GSTS(1) and GSTS(2) with different choices of the matrix B_2 (Table 4.1 for Example 4.1 and Table 4.2 for Example 4.2). In all tests we choose $B_1 = \widetilde{M}$.

Table 4.1: Choices of the matrix B_2 for Example 4.1.

Methods	Matrix B_2	Description
GSTS(1)	$E\widehat{M}^{-1}E^T$	$\widehat{M} = tridiag(\widetilde{M})$
GSTS(2)	$tridiag(E\widetilde{M}^{-1}E^T)$	

Table 4.2: Choices of the matrix B_2 for Example 4.2.

Methods	Matrix B_2	Description
GSTS(1)	$E\widehat{M}^{-1}E^T$	$\widehat{M} = tridiag(\widetilde{M})$
GSTS(2)	$E\widehat{M}^{-1}E^T$	$\widehat{M} = tridiag(M) + \gamma diag(E^T E)$

Optimal values of the relaxation parameters ω, τ for GSOR are the theoretical optimal values in [7]. The optimal values of the parameters for GSTS(1) and GSTS(2) are numerical optimal values, and the parameters $\omega_1 = \omega_2 = \omega_{exp}, \tau = \tau_{exp}$. The approximate matrix S of the Schur complement for GSOR is $S = E\widehat{M}^{-1}E^T$, $\widehat{M} = tridiag(\widetilde{M})$. All testing methods are employed both as solvers and as preconditioners to full GMRES.

In the numerical experiments we choose $W \equiv I$, where $I \in \mathbb{R}^{q\times q}$ is the identity matrix. For the augmented linear system we take $\gamma = \|M\|_2/\|E\|_2^2$.

Table 4.3: IT, CPU and RES for Example 4.1 ($m = 1/8$).

p		1024	1600	2304	3136	4096	5184
q		32	40	48	56	64	72
$p+q$		1056	1640	2352	3192	4160	5256
GSTS(1)	IT	24	26	28	29	32	36
	CPU	0.0089	0.0088	0.0091	0.054	0.061	0.076
	RES	9.71e-7	3.92e-7	7.72e-7	6.54e-7	5.33e-7	4.33e-7
GSTS(2)	IT	19	20	21	23	28	30
	CPU	0.0043	0.0055	0.0061	0.018	0.034	0.043
	RES	6.73e-6	5.66e-7	8.78e-7	7.71e-7	9.52e-6	2.19e-7
GSOR	IT	22	24	25	27	30	33
	CPU	0.0062	0.0069	0.0078	0.034	0.052	0.059
	RES	8.07e-7	9.73e-7	8.42e-7	8.01e-7	6.96e-7	7.68e-6

Table 4.4: IT, CPU and RES for Example 4.1 ($m = 3$).

p		1024	1600	2304	3136	4096	5184
q		32	40	48	56	64	72
$p+q$		1056	1640	2352	3192	4160	5256
GSTS(1)	IT	53	56	67	76	92	108
	CPU	0.019	0.027	0.047	0.068	0.097	0.232
	RES	6.61e-7	3.12e-6	1.22e-7	1.44e-6	4.93e-7	8.31e-7
GSTS(2)	IT	48	52	61	68	87	96
	CPU	0.013	0.019	0.033	0.049	0.077	0.173
	RES	6.73e-6	5.66e-7	8.78e-7	7.71e-7	9.52e-6	2.19e-7
GSOR	IT	50	54	65	72	89	103
	CPU	0.016	0.022	0.042	0.054	0.083	0.211
	RES	8.07e-7	9.73e-7	8.42e-7	8.01e-7	6.96e-7	7.68e-6

Table 4.5: IT, CPU and RES of GMRES for Example 4.1 ($m = 1/8$).

p			1024	1600	2304	3136	4096	5184
q			32	40	48	56	64	72
$p+q$			1056	1640	2352	3192	4160	5256
GMRES		IT	253	323	379	488	695	-
		CPU	37.97	43.44	49.52	58.12	85.15	-
		RES	2.21e-7	1.12e-6	6.02e-7	1.24e-6	3.73e-7	-
PGMRES	GSTS(1)	IT	13	13	15	16	21	27
		CPU	0.033	0.051	0.082	0.096	0.216	0.357
		RES	6.61e-7	3.12e-6	1.22e-7	1.44e-6	4.93e-7	8.31e-7
	GSTS(2)	IT	10	11	12	13	16	22
		CPU	0.028	0.041	0.069	0.088	0.174	0.321
		RES	6.73e-6	5.66e-7	8.78e-7	7.71e-7	9.52e-6	2.19e-7
	GSOR	IT	12	13	14	15	19	25
		CPU	0.031	0.048	0.078	0.092	0.191	0.342
		RES	8.07e-7	9.73e-7	8.42e-7	8.01e-7	6.96e-7	7.68e-6

Example 4.1. Consider the saddle-point linear system, in which the matrix block $M \in \mathbb{R}^{p\times p}$ is the three-point difference matrix of the one-dimensional Laplace operator with periodic boundary conditions. So the matrix M is singular. The matrix $E^T = U\Sigma V$ with $U \in \mathbb{R}^{p\times q}$ being a column orthogonal matrix, $V \in \mathbb{R}^{q\times q}$ being an orthogonal matrix, $\Sigma = diag(1, 2^m, ..., q^m) \in \mathbb{R}^{q\times q}$ and m a given positive real number. Here, U and V are generated randomly with normal distribution by the MATLAB code *randn*.

In Tables 4.3 and 4.4 we list numerical results for Example 4.1 with respect to varying p and q when $m = 1/8$ and $m = 3$, correspondingly. From these tables we see that all the testing methods with the experimental or the theoretical optimal parameters quickly produce approximate solution of high-quality. GSTS(2) always outperforms the other testing methods in iteration steps and CPU time. The performance of GSOR is a little bit better than GSTS(1) in iteration steps and CPU time.

In Tables 4.5 and 4.6 we list numerical results for Example 4.1 when the full GMRES method and the preconditioned GMRES method, with the testing methods as preconditioners, are used to solving saddle-point linear systems. We can see that if no preconditioner is used,

Table 4.6: IT, CPU and RES of GMRES for Example 4.1 ($m = 3$).

p			1024	1600	2304	3136	4096	5184
q			32	40	48	56	64	72
$p+q$			1056	1640	2352	3192	4160	5256
GMRES		IT	328	542	648	868	-	-
		CPU	51.44	68.12	71.23	89.11	-	-
		RES	6.61e-7	3.12e-6	1.22e-7	1.44e-6	4.93e-7	8.31e-7
PGMRES	GSTS(1)	IT	30	32	32	35	37	39
		CPU	0.388	0.395	0.416	0.436	0.452	1.397
		RES	6.61e-7	3.12e-6	1.22e-7	1.44e-6	4.93e-7	8.31e-7
	GSTS(2)	IT	25	26	26	28	29	30
		CPU	0.351	0.364	0.387	0.413	0.436	1.181
		RES	6.73e-6	5.66e-7	8.78e-7	7.71e-7	9.52e-6	2.19e-7
	GSOR	IT	29	31	31	34	36	37
		CPU	0.379	0.386	0.397	0.429	0.441	1.261
		RES	8.07e-7	9.73e-7	8.42e-7	8.01e-7	6.96e-7	7.68e-6

Table 4.7: IT, CPU and RES for Example 4.2.

p		500	1000	1500	2000	2500	3000
q		500	500	500	500	500	500
$p+q$		1000	1500	2000	2500	3000	3500
γ		1.98e-1	2.77e-1	1.09e-1	3.75e-1	1.75e-1	7.98e-2
GSTS(1)	IT	79	91	132	141	156	173
	CPU	0.091	0.166	0.176	0.193	0.453	1.954
	RES	1.22e-7	3.82e-6	6.12e-7	9.44e-6	3.88e-7	6.31e-7
GSTS(2)	IT	70	83	108	129	138	147
	CPU	0.041	0.085	0.093	0.129	0.252	1.564
	RES	1.23e-6	9.06e-7	7.58e-7	4.71e-7	1.32e-6	1.19e-7
GSOR	IT	75	88	119	137	149	159
	CPU	0.072	0.152	0.163	0.175	0.386	1.873
	RES	3.57e-7	7.33e-6	2.42e-7	3.55e-7	3.96e-7	9.48e-6

GMRES converges very slowly to the approximate solution. However, when GSTS(i) ($i = 1, 2$) and GSOR are used, the preconditioned GMRES method successfully converges to the exact solution of the saddle-point linear systems, and GSTS(2) show the best preconditioning effect than the other testing methods.

Example 4.2. ([14]) Consider the saddle-point linear system constructed in the following manner: the matrix $M \in \mathbb{R}^{p\times p}$ is the block-diagonal matrix of n pentadiagonal blocks ($n = 10l, l = 1, ..., 6$) with dimension 50×50, consisting of normally distributed random numbers. Each pentadiagonal block M_i has nullity equal to 1, which is generated by setting $M_i \leftarrow M_i - \lambda_{min}(M_i)I$ after the construction. So, the matrix M is semidefinite and its rank is $49n$. The random matrix $E \in \mathbb{R}^{q\times p}$ is comprised of l tridiagonal blocks E_i ($i = 1, ..., l$) with dimension 500×500 assembled together ($l = 1, ..., 6$).

The numerical results for Example 4.2 are listed in Table 4.7 and Table 4.8 when the testing methods are used both as solvers and as preconditioners to GMRES. Numerical results show that GSTS is effective for solving the augmented saddle-point linear systems and, in some

Table 4.8: IT, CPU and RES of GMRES for Example 4.2.

p			500	1000	1500	2000	2500	3000
q			500	500	500	500	500	500
$p+q$			1000	1500	2000	2500	3000	3500
γ			1.8e-01	1.6e-01	1.1e-01	3.5e-01	2.9e-02	2.5e-02
GMRES		IT	256	348	417	653	869	-
		CPU	49.01	58.98	77.76	104.07	137.54	-
		RES	2.86e-7	2.12e-6	8.51e-7	9.24e-6	6.23e-7	-
PGMRES	GSTS(1)	IT	38	53	56	60	74	79
		CPU	17.21	28.81	35.52	40.01	46.54	54.81
		RES	1.41e-7	8.12e-6	4.33e-7	9.44e-6	6.65e-7	4.31e-7
	GSTS(2)	IT	29	44	46	48	57	61
		CPU	12.53	18.64	28.05	34.41	38.64	43.96
		RES	4.23e-6	2.76e-7	9.78e-7	3.71e-6	4.52e-6	8.49e-6
	GSOR	IT	35	51	54	55	59	68
		CPU	17.01	22.55	31.48	37.98	44.03	48.73
		RES	3.77e-7	2.43e-7	4.12e-7	9.41e-6	5.96e-6	3.68e-6

cases, GSTS(2) converges better than GSOR. As one can see, GSTS converges significantly faster than (full) GMRES without preconditioning. We also observe that as a preconditioner GSTS(2) shows the best preconditioning effect; the performance of the GMRES preconditioned by GSOR is better than those preconditioned by GSTS(1).

Therefore, GSTS is effective for solving the saddle-point linear systems both as a solver and as a preconditioner to GMRES.

5. Concluding Remarks

We have discussed indefinite saddle-point linear systems, possibly having singular $(1,1)$ blocks. Our focus is on how to modify the linear systems such that they are easily solvable. We have considered and examined some aspects of the augmented Lagrangian technique and solved the augmented saddle-point linear system by GSOR and GSTS iteration methods. Note that in [7] and [15] these methods employed in the case when the $(1,1)$ blocks of the original matrix is positive definite were only. Numerical results have shown that GSTS and GSOR iteration methods are effective for solving the saddle-point linear systems with semidefinite and singular $(1,1)$ blocks. The study of the convergence factor of GSTS for saddle-point linear systems and the relationships between the parameter γ of the augmented saddle-point linear system and the convergence factor of GSTS will be a research topic in future that is of both theoretical importance and practical value.

Acknowledgments. The author would like to thank Z.-Z. Bai and the reviewers for the suggestions towards improving this paper.

References

[1] M. Arioli and G. Manzini, A null space algorithm for mixed finite-element approximations of Darcy's equation, *Communications in Numerical Methods in Engineering*, **18** (2002), 645-657.

[2] Z.-Z. Bai, Eigenvalue estimates for saddle-point matrices of Hermitian and indefinite leading blocks, *Journal of Computational and Applied Mathematics*, **237** (2013), 295-306.

[3] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, *Numerical Linear Algebra with Applications*, **26** (2009), 447-479.

[4] Z.-Z. Bai and G.H. Golub, Accelerated Hermitian and skew-Hermitian splitting methods for saddle-point problems, *IMA Journal of Numerical Analysis*, **27** (2007), 1-23.

[5] Z.-Z. Bai, G.H. Golub and M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM Journal on Matrix Analysis and Applications*, **24** (2003), 603-626.

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

[7] Z.-Z. Bai, B.N. Parlett and Z.-Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, *Numerische Mathematik*, **102** (2005), 1-38.

[8] M. Benzi, Solution of equality-constrained quadratic programming problems by a projection iterative method, *Rendiconti di Matematica e delle sue Applicazioni*, **13** (1993), 275-296.

[9] M. Benzi and G.H. Golub, A preconditioner for generalized saddle point problems, *SIAM Journal on Matrix Analysis and Applications*, **26** (2004), 20-41.

[10] M. Benzi, M.J. Gander and G.H. Golub, Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems, *BIT Numerical Mathematics*, **43** (2003), 881-900.

[11] M. Benzi, G.H. Golub and J. Liesen, Numerical solution of saddle point problems, *Acta Numerica*, **14** (2005), 1-137.

[12] M. Benzi and M.A. Olshanskii, An augmented Lagrangian approach to the Oseen problem, *SIAM Journal on Scientific Computing*, **28** (2006), 2095-2113.

[13] R. Glowinski and P. Le Tallec, Augmented Lagrangian and operator splitting methods in nonlinear mechanics, SIAM, Philadelphia, 1989.

[14] G.H. Golub and C. Greif, On solving block-structured indefinite linear systems, *SIAM Journal on Scientific Computing*, **24** (2003), 2076-2092.

[15] L.A. Krukier, B.L. Krukier and Z.-R. Ren, Generalized skew-Hermitian triangular splitting iteration methods for saddle-point linear systems, Article first published online: DOI:10.1002/nla.1870.

[16] L.A. Krukier, T.S. Martynova and Z.-Z. Bai, Product-type skew-Hermitian triangular splitting iteration methods for strongly non-Hermitian positive definite linear systems, *Journal of Computational and Applied Mathematics*, **232** (2009), 3-16.

[17] L. Wang and Z.-Z. Bai, Skew-Hermitian triangular splitting iteration methods for non-Hermitian positive definite linear systems of strong skew-Hermitian parts, *BIT Numerical Mathematics*, **44** (2004), 363-386.

[18] B. Zheng, K. Wang and Y.-J. Wu, SSOR-like methods for saddle-point problems, *International Journal of Computer Mathematics*, **86** (2009), 1405-1423.