Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(5): 7625–7641.
DOI: 10.3934/math.2022428
Received: 03 August 2021
Revised: 29 January 2022
Accepted: 08 February 2022
Published: 15 February 2022

Research article

Generalized accelerated AOR splitting iterative method for generalized saddle point problems

Jin-Song Xiong*

Computer Information Engineering College, Guizhou University of Commerce, Guiyang 550014, China

* **Correspondence:** Email: xiongjins2015@163.com; Tel: +8618848509286.

Abstract: Generalized accelerated AOR (GAAOR) splitting iterative method for the generalized saddle point problems is proposed in this paper. The iterative scheme and the convergence of the GAAOR splitting method are researched. The eigenvalues distributions of its preconditioned matrix is discussed under two different choices of the parameter matrix Q. The resulting GAAOR preconditioner is used to precondition Krylov subspace method such as the restarted generalized minimal residual (GMRES) method for solving the equivalent formulation of the generalized saddle point problems. The theoretical results and effectiveness of the GAAOR splitting iterative method are supported by some numerical examples.

Keywords: generalized saddle point problem; GAAOR splitting iterative method; convergence
Mathematics Subject Classification: 65F10, 65F50

1. Introduction

Consider the generalized saddle point problems

$$\mathscr{A}X = \begin{pmatrix} A & B \\ -B^* & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} p \\ -q \end{pmatrix} = b, \tag{1.1}$$

where $A \in \mathbb{C}^{m\times m}$ is symmetric positive definite matrix and B^* is the conjugate transpose of a full column rank matrix $B \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{n\times n}$ is symmetric semi-positive definite matrix, $p \in \mathbb{C}^n$ and $q \in \mathbb{C}^n$.

Problems associated with linear system like the one of Eq (1.1) arise in a wide fields of scientific computing and engineering applications, such as computational fluid dynamics [1–9], least-squares problems [10–12], electronic networks [13]. In general, iterative method is more attractive than direct methods for large and sparse problem. In particular, a lot of iterative methods have been developed to solve the problems (1.1). For example, the HSS method [14–19], Block triangular preconditioner [20–23], SOR method [3, 24–27], PIU method [29–31] and PPS method [32–34].

In recent decades, many scholars have studied the AOR method to solve saddle point problem and achieved some results [35–40]. However, there is still further research on the solution speed and the number of iterations, and there is less research on the generalized saddle point problem with the generalized method. In this paper, the GAAOR splitting iterative method is proposed to solve the problems (1.1). We analyzed the iterative scheme and the convergence of the GAAOR splitting iterative method. We discussed the eigenvalues distributions of its preconditioned matrix under two different choices of the parameter matrix Q. In addition, the resulting GAAOR preconditioner is used to accelerate Krylov subspace method.

The rest of the paper is organized as follows. In Section 2, we propose the GAAOR splitting iterative method for solving the problems (1.1). In Section 3, the convergence of the GAAOR splitting iterative method is discussed. In Section 4, we discuss the eigenvalues distributions of its preconditioned matrix under two different choices of the parameter matrix Q, the resulting GAAOR preconditioner is used to accelerate Krylov subspace method and two numerical examples are given to demonstrate the theoretical results and the effectiveness of the GAAOR splitting iterative method.

2. GAAOR splitting iterative method

In this section, we shall introduce GAAOR splitting iterative method and its preconditioner for solving the problems (1.1).

For given symmetric positive definite matrix $Q \in \mathbb{C}^{n\times n}$, $\alpha > 0, \beta > 0$. Let the coefficient matrix $\mathscr{A}$ of the problems (1.1) be split as follows:

$$\mathscr{A} = D - L - U, \tag{2.1}$$

where

$$D = \begin{pmatrix} \alpha A & O \\ O & \beta Q \end{pmatrix}, L = \begin{pmatrix} -A & O \\ B^* & \frac{\beta}{2}Q \end{pmatrix}, U = \begin{pmatrix} \alpha A & -B \\ O & \frac{\beta}{2}Q - C \end{pmatrix}.$$

For given $0 < \omega \leq \gamma < 2$. Through the splitting of (2.1), we obtained the GAAOR splitting iterative scheme for the problems (1.1).

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = T_{(\alpha,\omega,\gamma)} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \omega(D - \gamma L)^{-1} \begin{pmatrix} p \\ -q \end{pmatrix}, \tag{2.2}$$

where

$$\begin{aligned} T_{(\alpha,\omega,\gamma)} &= (D - \gamma L)^{-1}[(1 - \omega)D + (\omega - \gamma)L + \omega U] \\ &= \begin{pmatrix} (\alpha+\gamma)A & O \\ -\gamma B^* & \frac{\beta(2-\gamma)}{2}Q \end{pmatrix}^{-1} \begin{pmatrix} (\alpha+\gamma-\omega)A & -\omega B \\ (\omega-\gamma)B^* & \frac{\beta(2-\gamma)}{2}Q - \omega C \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{(\alpha+\gamma)}A^{-1} & O \\ \frac{2\gamma}{\beta(\alpha+\gamma)(2-\gamma)}Q^{-1}B^*A^{-1} & \frac{2}{\beta(2-\gamma)}Q^{-1} \end{pmatrix} \begin{pmatrix} (\alpha+\gamma-\omega)A & -\omega B \\ (\omega-\gamma)B^* & \frac{\beta(2-\gamma)}{2}Q - \omega C \end{pmatrix} \\ &= \begin{pmatrix} \frac{\alpha+\gamma-\omega}{(\alpha+\gamma)}I_m & -\frac{\omega}{\alpha+\gamma}B \\ \frac{2\alpha\omega}{\beta(\alpha+\gamma)(2-\gamma)}Q^{-1}B^* & I_n - \frac{2\omega}{\beta(2-\gamma)}Q^{-1}C - \frac{2\omega\gamma}{\beta(\alpha+\gamma)(2-\gamma)}Q^{-1}B^*A^{-1}B \end{pmatrix}. \end{aligned}$$

Thus, the GAAOR splitting iterative method can be defined as follows.

GAAOR splitting iterative method: Let Q be symmetric and positive definite. Given an initial guess $x^{(0)^*}$ and $y^{(0)^*}$, parameters $\alpha > 0, \beta > 0$, $0 < \omega \leq \gamma < 2$. For $k = 0, 1, 2, \ldots$ until iteration sequence $x^{(k)^*}$ and $y^{(k)^*}$ are convergent, compute

$$\begin{cases} x^{(k+1)} = x^{(k)} + \frac{\omega}{\alpha+\gamma}(p - Ax^{(k)} - By^{(k)}), \\ y^{(k+1)} = y^{(k)} + \frac{2\omega}{\beta(2-\gamma)}(B^* x^{(k+1)} - Cy^{(k)} - q). \end{cases} \tag{2.3}$$

In addition, the iteration matrix $\mathscr{A}$ can be induced from the splitting

$$\mathscr{A} = M_{(\alpha,\beta,\omega,\gamma)} - N_{(\alpha,\beta,\omega,\gamma)},$$

where

$$M_{(\alpha,\beta,\omega,\gamma)} = \frac{1}{\omega}(D - \gamma L) = \begin{pmatrix} \frac{\alpha+\gamma}{\omega}A & O \\ -\frac{\gamma}{\omega}B^* & \frac{\beta(2-\gamma)}{2\omega}Q \end{pmatrix}, \tag{2.4}$$

$$N_{(\alpha,\beta,\omega,\gamma)} = \begin{pmatrix} \frac{\alpha+\gamma-\omega}{\omega}A & -B \\ \frac{\omega-\gamma}{\omega}B^* & \frac{\beta(2-\gamma)}{2\omega}Q - C \end{pmatrix}. \tag{2.5}$$

It is readily seen that

$$T_{(\alpha,\beta,\omega,\gamma)} = M^{-1}_{(\alpha,\beta,\omega,\gamma)} N_{(\alpha,\beta,\omega,\gamma)}, \tag{2.6}$$

where $T_{(\alpha,\beta,\omega,\gamma)}$ and $M^{-1}_{(\alpha,\beta,\omega,\gamma)}\mathscr{A}$ are iteration matrix and preconditioning matrix of the GAAOR splitting iterative method, respectively. The splitting matrix $M_{(\alpha,\beta,\omega,\gamma)}$ can be served as a preconditioner, called the GAAOR preconditioner. Furthermore, the action of GAAOR splitting matrix $M_{(\alpha,\beta,\omega,\gamma)}$ as a preconditioner can be realized through solving sequences of generalized residual equations of the form

$$\begin{pmatrix} \frac{\alpha+\gamma}{\omega}A & O \\ -\frac{\gamma}{\omega}B^* & \frac{\beta(2-\gamma)}{2\omega}Q \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{2.7}$$

where $(z_1^*, z_2^*)^*$ and $(r_1^*, r_2^*)^*$ represent the generalized residual vector and the current residual vector, respectively. From the Eq (2.7), we can compute the vector $(z_1^*, z_2^*)^*$ through the following two steps:

Step 1. Solve $(\alpha + \gamma)Az_1 = \omega r_1$ for z_1.

Step 2. Solve $\beta(2 - \gamma)Qz_2 = 2(\omega r_2 + \gamma B^* z_1)$ for z_2.

Clearly, at each step of the GAAOR splitting iterative method or applying the GAAOR preconditioner $M_{(\alpha,\beta,\omega,\gamma)}$ with a Krylov subspace method, we only need to solve two sub-system with coefficient matrix $(\alpha + \gamma)A$ and $\beta(2 - \gamma)Q$.

3. The convergence analysis of the GAAOR splitting iterative method

In this section, we will analyze the convergence of the GAAOR splitting iterative method for solving the problems (1.1). We first recall in Lemmas 3.1–3.3 some useful result of [41].

Lemma 3.1. *Let* $(x^{(0)^*}, y^{(0)^*})^*$ *is given initial vector. If* $(x^{(k)^*}, y^{(k)^*})^*$ *is defined by the split iteration* (2.1) *of coefficient matrix* $\mathscr{A}$*, then Eq* (2.2) *holds. If the spectral radius* $\rho(T_{(\alpha,\beta,\omega,\gamma)}) < 1$*, then for any initial vector* $(x^{(0)^*}, y^{(0)^*})^*$*, the iterative sequence* $(x^{(k)^*}, y^{(k)^*})^*$ *will converge to the unique solution* $(x^*, y^*)^*$ *of the problems* (1.1).

Lemma 3.2. *The GAAOR splitting iterative method* (2.3) *converges if and only if the spectral radius* $\rho(T_{(\alpha,\beta,\omega,\gamma)})$ *of the iteration matrix* $T_{(\alpha,\beta,\omega,\gamma)}$ *satisfies the inequality* $\rho(T_{(\alpha,\beta,\omega,\gamma)}) < 1$.

Lemma 3.3. *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* $|b| < 1 + c$.

Secondly, we are in the position to state and prove our main results.

Theorem 3.1. *Suppose that* $A \in \mathbb{C}^{m\times m}$ *is symmetric positive definite matrix and* B^* *is the transpose of a full column rank matrix* $B \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{n\times n}$ *is symmetric semi-positive definite matrix. If* λ *is an eigenvalue of the iteration matrix* $T_{(\alpha,\beta,\omega,\gamma)}$ *and* $(u^*, v^*)^* \in \mathbb{C}^{m+n}$ *is the corresponding eigenvector, then*

$$\lambda^2 - b\lambda + c = 0, \tag{3.1}$$

where

$$b = \frac{\beta(2\alpha + 2\gamma - \omega)(2-\gamma) - 2\omega\eta(\alpha+\gamma) - 2\omega\gamma\delta}{\beta(\alpha+\gamma)(2-\gamma)}, \tag{3.2}$$

$$c = \frac{(\alpha+\gamma-\omega)[\beta(2-\gamma) - 2\omega\eta] + 2\omega\delta(\omega-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)}, \tag{3.3}$$

where we defined the new constant

$$\eta: = \frac{v^*Cv}{v^*Qv},\ \delta: = \frac{v^*B^*A^{-1}Bv}{v^*Qv}. \tag{3.4}$$

Proof. Let $w = (u^*, v^*)^*$, λ is an eigenvalue of the iteration matrix $T_{(\alpha,\beta,\omega,\gamma)}$ and $(u^*, v^*)^* \in \mathbb{C}^{m+n}$ is the corresponding eigenvector, then by $T_{(\alpha,\beta,\omega,\gamma)}w = \lambda w$ and Eq (2.6), we obtained $N_{(\alpha,\beta,\omega,\gamma)}w = \lambda M_{(\alpha,\beta,\omega,\gamma)}w$. Thus, we have

$$\begin{pmatrix} \frac{\alpha+\gamma-\omega}{\omega}A & -B \\ \frac{\omega-\gamma}{\omega}B^* & \frac{\beta(2-\gamma)}{2\omega}Q - C \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \lambda\begin{pmatrix} \frac{\alpha+\gamma}{\omega}A & O \\ -\frac{\gamma}{\omega}B^* & \frac{\beta(2-\gamma)}{2\omega}Q \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \tag{3.5}$$

by Eqs (2.4) and (2.5). That is,

$$\begin{cases} [\frac{(\alpha+\gamma)(1-\lambda)}{\omega} - 1]Au = Bv, \\ (\frac{\beta(2-\gamma)(1-\lambda)}{2\omega}Q - C)v = [\frac{\gamma(1-\lambda)}{\omega} - 1]B^*u. \end{cases} \tag{3.6}$$

We note that $\lambda \neq 1$ and $u \neq 0$. Indeed, assuming $\lambda = 1$, we obtain

$$\begin{cases} -Au = Bv, \\ Cv = B^*u. \end{cases} \tag{3.7}$$

By the first equation in Eq (3.7), we get $u = -A^{-1}Bv$. Substituting it into the second equation in Eq (3.7), we have

$$(C + B^*A^{-1}B)v = 0.$$

Since A is symmetric positive definite matrix, B is a full column rank matrix, C is symmetric semi-positive definite matrix, $C + B^*A^{-1}B$ is a symmetric positive definite matrix. Hence, we have $v = 0$ and $u = -A^{-1}Bv = 0$. This contradicts the assumption that v is an eigenvector of $T_{(\alpha,\beta,\omega,\gamma)}$. Hence $\lambda \neq 1$.

The result $u \neq 0$ can be proven similarly.

Suppose that $\frac{(\alpha+\gamma)(1-\lambda)}{\omega} - 1 \neq 0$. By the first equation in Eq (3.6), we obtain $u = \frac{\omega}{(\alpha+\gamma)(\lambda-1)+\omega} A^{-1}Bv$. Substituting it into the second equation in Eq (3.6), we have

$$(\frac{\beta(2-\gamma)(1-\lambda)}{2\omega} Q - C)v = \frac{\gamma(\lambda-1)+\omega}{(\alpha+\gamma)(\lambda-1)+\omega} B^*A^{-1}Bv. \tag{3.8}$$

It is straightforward to obtain that $v \neq 0$ and $v^*Qv \neq 0$ (since Q is a positive definite matrix). Hence, on multiplying the Eq (3.8) by $\frac{v^*}{v^*Qv}$ from the left, we obtain the Eq (3.1) through Eqs (3.2)–(3.4). □

Theorem 3.2. *Suppose that $A \in \mathbb{C}^{m\times m}$ is symmetric positive definite matrix and B^* is the transpose of a full column rank matrix $B \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{n\times n}$ is symmetric semi-positive definite matrix, parameters $\alpha > 0, \beta > 0$, $0 < \omega \le \gamma < 2$. If α, β, ω, γ, η and δ satisfy one of the conditions*

$$(1) \begin{cases} \eta = 0, \\ \delta < \dfrac{\beta(2-\gamma)(2\alpha+2\gamma-\omega)}{\omega(2\gamma-\omega)}; \end{cases} \tag{3.9}$$

or

$$(2) \begin{cases} 0 < \eta < \dfrac{\beta(2-\gamma)}{\omega}, \\ \delta < \dfrac{[\beta(2-\gamma)-\omega\eta](2\alpha+2\gamma-\omega)}{\omega(2\gamma-\omega)}; \end{cases} \tag{3.10}$$

then the splitting iteration method (2.3) converges.

Proof. Without loss of generality, let $\gamma \neq \omega$. Since $C \in \mathbb{C}^{n\times n}$ is a symmetric semi-positive definite matrix and Q is symmetric and positive definite, we have $\eta \ge 0$ by Eq (3.4).

(1) Let $\eta = 0$. From Eqs (3.2) and (3.3), we obtain

$$b = \frac{\beta(2\alpha+2\gamma-\omega)(2-\gamma) - 2\omega\gamma\delta}{\beta(\alpha+\gamma)(2-\gamma)},$$

$$c = 1 + \frac{2\omega\delta(\omega-\gamma) - \omega\beta(2-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)},$$

$$1 + c = \frac{\beta(2\alpha+2\gamma-\omega)(2-\gamma) + 2\omega\delta(\omega-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)}. \tag{3.11}$$

From $|c| < 1$, we have

$$\left|1 + \frac{2\omega\delta(\omega-\gamma) - \omega\beta(2-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)}\right| < 1$$

$$\Longleftrightarrow -1 < 1 + \frac{2\omega\delta(\omega-\gamma) - \omega\beta(2-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)} < 1,$$

that is,

$$2\omega\delta(\gamma-\omega) + \omega\beta(2-\gamma) > 0, \tag{3.12}$$

or

$$\frac{2\beta(\alpha+\gamma)(2-\gamma) + 2\omega\delta(\omega-\gamma) - \omega\beta(2-\gamma)}{\beta(\alpha+\gamma)(2-\gamma)} > 0. \tag{3.13}$$

Since A and Q are symmetric positive definite matrices, $\alpha > 0, \beta > 0$ and $0 < \omega < \gamma < 2$, we have $\delta > 0$, $\omega\delta(\gamma - \omega) > 0$, $\omega\beta(2 - \gamma) > 0$. Hence inequality (3.12) holds. Inequality (3.13) is transformed into

$$\beta(2\alpha + 2\gamma - \omega)(2 - \gamma) > 2\omega\delta(\gamma - \omega). \tag{3.14}$$

Since $\alpha > 0, \beta > 0$ and $0 < \omega < \gamma < 2$, we have $\gamma - \omega > 0$, $(2 - \gamma) > 0$, $2\alpha + 2\gamma - \omega = 2\alpha + \gamma + (\gamma - \omega) > 0$. From inequality (3.14), we obtain

$$\delta < \frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma)}{2\omega(\gamma - \omega)}. \tag{3.15}$$

From Eqs (3.2), (3.11) and $|b| < 1 + c$, we have

$$\left|\frac{\varepsilon - 2\omega\gamma\delta}{\beta(\alpha + \gamma)(2 - \gamma)}\right| < \frac{\varepsilon + 2\omega\delta(\omega - \gamma)}{\beta(\alpha + \gamma)(2 - \gamma)}, \tag{3.16}$$

where $\varepsilon = \beta(2\alpha + 2\gamma - \omega)(2 - \gamma)$.

From inequality (3.15), we obtain

$$\beta(2\alpha + 2\gamma - \omega)(2 - \gamma) + 2\omega\delta(\omega - \gamma) > 0.$$

Because $\beta(\alpha + \gamma)(2 - \gamma) > 0$, we have inequalities (3.16) is equivalent to

$$\begin{cases} \beta(2\alpha + 2\gamma - \omega)(2 - \gamma) - 2\omega\gamma\delta < \beta(2\alpha + 2\gamma - \omega)(2 - \gamma) + 2\omega\delta(\omega - \gamma), \\ \beta(2\alpha + 2\gamma - \omega)(2 - \gamma) - 2\omega\gamma\delta > -[\beta(2\alpha + 2\gamma - \omega)(2 - \gamma) + 2\omega\delta(\omega - \gamma)], \end{cases}$$

that is,

$$\begin{cases} 2\delta\omega^2 > 0, \\ \beta(2\alpha + 2\gamma - \omega)(2 - \gamma) > \omega\delta(2\gamma - \omega). \end{cases} \tag{3.17}$$

Obviously, the first inequality of inequalities (3.17) holds. Because $\beta > 0$ $2\alpha + 2\gamma - \omega > 0$ and $\omega(2\gamma - \omega) = \omega[\gamma + (\gamma - \omega)] > 0$, we obtain

$$\delta < \frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma)}{\omega(2\gamma - \omega)} \tag{3.18}$$

by the second inequality of inequalities (3.17).

Since $2\omega(\gamma - \omega) - \omega(2\gamma - \omega) = -\omega^2 < 0$, we have

$$0 < 2\omega(\gamma - \omega) < \omega(2\gamma - \omega). \tag{3.19}$$

And because $\beta(2\alpha + 2\gamma - \omega)(2 - \gamma) > 0$, we obtain

$$\frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma)}{2\omega(\gamma - \omega)} > \frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma)}{\omega(2\gamma - \omega)}. \tag{3.20}$$

From inequalities (3.15), (3.18) and (3.20), we have

$$\delta < \frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma)}{\omega(2\gamma - \omega)}.$$

Hence, we have Eq (3.9) holds.

(2) Let $\eta > 0$. From Eqs (3.2) and (3.3), we get

$$\begin{aligned} 1 + c &= \frac{\beta(2\alpha + 2\gamma - \omega)(2 - \gamma) - 2\omega\eta(\alpha + \gamma) - 2\omega\gamma\delta + 2\omega^2(\eta + \delta)}{\beta(\alpha + \gamma)(2 - \gamma)} \\ &= b + \frac{2\omega^2(\eta + \delta)}{\beta(\alpha + \gamma)(2 - \gamma)}. \end{aligned} \tag{3.21}$$

Since $2\omega^2(\eta + \delta) > 0$ and $\beta(\alpha + \gamma)(2 - \gamma) > 0$, we have $\frac{2\omega^2(\eta+\delta)}{\beta(\alpha+\gamma)(2-\gamma)} > 0$. Hence, we have $b < 1 + c$ by Eq (3.21). To prove that inequality $|b| < 1+c$ holds. Next, we only need to prove inequality $-(1+c) < b$ holds.

From Eqs (3.2), (3.21) and $-(1 + c) < b$, we obtain

$$-\frac{\psi + 2\omega^2(\eta + \delta)}{\beta(\alpha + \gamma)(2 - \gamma)} < \frac{\psi}{\beta(\alpha + \gamma)(2 - \gamma)}, \tag{3.22}$$

where $\psi = \beta(2\alpha + 2\gamma - \omega)(2 - \gamma) - 2\omega\eta(\alpha + \gamma) - 2\omega\gamma\delta$. Simplifying by inequality (3.22), we get

$$\delta\omega(2\gamma - \omega) < (2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta]. \tag{3.23}$$

Since $\delta\omega(2\gamma - \omega) > 0$ and $2\alpha + 2\gamma - \omega > 0$, the inequality (3.23) has a solution if and only if

$$\begin{cases} \beta(2 - \gamma) - \omega\eta > 0, \\ \delta < \frac{(2\alpha+2\gamma-\omega)[\beta(2-\gamma)-\omega\eta]}{\omega(2\gamma-\omega)}, \end{cases}$$

that is,

$$\begin{cases} \eta < \frac{\beta(2-\gamma)}{\omega}, \\ \delta < \frac{(2\alpha+2\gamma-\omega)[\beta(2-\gamma)-\omega\eta]}{\omega(2\gamma-\omega)}. \end{cases} \tag{3.24}$$

From Eq (3.3) and $|c| < 1$, we have

$$\left|\frac{(\alpha + \gamma - \omega)[\beta(2 - \gamma) - 2\omega\eta] + 2\omega\delta(\omega - \gamma)}{\beta(\alpha + \gamma)(2 - \gamma)}\right| < 1. \tag{3.25}$$

Since $\alpha > 0, \beta > 0$ and $0 < \omega < \gamma < 2$, we have $\beta(\alpha + \gamma)(2 - \gamma) > 0$. Hence, inequality (3.25) is equivalent to

$$\begin{cases} (\alpha + \gamma - \omega)[\beta(2 - \gamma) - 2\omega\eta] + 2\omega\delta(\omega - \gamma) < \beta(\alpha + \gamma)(2 - \gamma), \\ (\alpha + \gamma - \omega)[\beta(2 - \gamma) - 2\omega\eta] + 2\omega\delta(\omega - \gamma) > -\beta(\alpha + \gamma)(2 - \gamma), \end{cases}$$

that is,

$$\begin{cases} 2\delta(\omega - \gamma) < 2\eta(\alpha + \gamma - \omega) + \beta(2 - \gamma), \\ 2\delta\omega(\gamma - \omega) < (2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta] + \omega^2\eta. \end{cases} \tag{3.26}$$

Because $\delta > 0$ and $\omega - \gamma < 0$, we have

$$2\delta(\omega - \gamma) < 0,$$

$$2\eta[\alpha + (\gamma - \omega)] + \beta(2 - \gamma) > 0.$$

Hence, we have the first inequality of inequalities (3.26) holds.

From the first inequality of inequalities (3.24), we have $\beta(2 - \gamma) - \omega\eta > 0$. Since $2\omega(\gamma - \omega) > 0$ and $2\alpha + 2\gamma - \omega > 0$, we obtained

$$\delta < \frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta] + \omega^2\eta}{2\omega(\gamma - \omega)}, \tag{3.27}$$

by the second inequality of inequalities (3.26). Since $(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta] > 0$, we have

$$\frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta]}{2\omega(\gamma - \omega)} > \frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta]}{\omega(2\gamma - \omega)} \tag{3.28}$$

by inequality (3.19). And because $\frac{\omega\eta}{2(\gamma-\omega)} > 0$, we have

$$\begin{aligned} &\frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta] + \omega^2\eta}{2\omega(\gamma - \omega)} \\ &= \frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta]}{2\omega(\gamma - \omega)} + \frac{\omega\eta}{2(\gamma - \omega)} \\ &> \frac{(2\alpha + 2\gamma - \omega)[\beta(2 - \gamma) - \omega\eta]}{\omega(2\gamma - \omega)} \end{aligned} \tag{3.29}$$

by inequality (3.28).

Hence, we have Eq (3.10) holds by the inequalities (3.24), (3.27) and (3.29). □

Therefore, by combining the above analysis and Lemma 2, we finally obtained the convergence result of the GAAOR splitting iterative method.

Theorem 3.3. *Suppose the conditions of Theorem 2 are satisfied. Then, the GAAOR splitting iterative method used for solving the generalized saddle point problems (1.1) is convergent.*

4. Numerical examples

In this section, two numerical examples are provided to illustrate both the theoretical results achieved in Section 3 and the effectiveness of the GAAOR splitting iterative method for solving the problems (1.1). The computation is conducted in **Matlab** on a personal computer with Intel(R) Core(TM)2 Quad CPU Q9500 2.83GHz, 4.0G memory and Windows 7 operating system. In the following tables, we list the iterative numbers (denoted by IT), the elapsed CPU times (denoted by CPU) in seconds that the stopping criterion, either $RES \leq 10^{-7}$ or the number of the prescribed iteration $k_{max} = 2000$, is met, and the residual error (denoted by RES) defined by

$$RES = \left(\frac{\| p - Ax^{(k)} - By^{(k)} \|_2^2 + \| q - B^* x^{(k)} + Cy^{(k)} \|_2^2}{\| p - Wx^{(0)} - By^{(0)} \|_2^2 + \| q - B^* x^{(0)} + Cy^{(0)} \|_2^2} \right)^{1/2}.$$

Example 4.1. *Consider the problems (1.1) with the coefficient matrix form*

$$A = \left(H + K + \frac{3-\sqrt{3}}{t}I\right) + i\left(H + K + \frac{3+\sqrt{3}}{t}I\right) \in \mathbb{C}^{l^2\times l^2},$$

$$H = I\otimes V_l + V_l\otimes I,\ V_l = \frac{1}{h^2}\cdot \text{tridiag}(-1,2,-1) \in \mathbb{C}^{l\times l},$$

$$K = I\otimes U_l + U_l\otimes I,\ U_l = \frac{1}{2h}\cdot \text{tridiag}(-1,0,1) \in \mathbb{C}^{l\times l},$$

$$B = (B_1\ B_2) = (B_1\ b_1\ b_2) \in \mathbb{C}^{l^2\times(l^2+2)},\quad C = I \in \mathbb{C}^{l^2\times l^2},$$

$$B_1 = I\otimes F + F\otimes I \in \mathbb{C}^{l^2\times l^2}, b_1 = B_1\begin{pmatrix}\mathbf{e}\\0\end{pmatrix}, b_2 = B_1\begin{pmatrix}0\\\mathbf{e}\end{pmatrix},$$

$\mathbf{e} = (1,1,\cdots,1)^* \in \mathbb{C}^{\frac{l^2}{2}}$, $F = \text{tridiag}(-1,1,0)/h \in \mathbb{C}^{l\times l}$, *t is the time step-size,* $\otimes$ *denotes the Kronecker product and* $h = \frac{1}{l+1}$ *is the discretization meshsize. Then* $m = 2l^2$ *and* $n = l^2+2$*. The Example 4.1 is a new technical modification in [3].*

All computations, we set the right-hand side vector of the problems (1.1) such that f^* to be a complex vector with its jth entry f_j being given by

$$p_j = \frac{(1-i)j}{t(j+1)^2}, t = h, j = 1,2,\cdots,m, q^* = (-1,-1,\cdots,-1)^* \in \mathbb{C}^n.$$

Let $C = I \in \mathbb{C}^{n\times n}$, the problem is normalized by multiplying both sides with h^2. Furthermore, we consider two different choices of the parameter matrix Q:

Case I: $Q = Diag(B^*A^{-1}B)$,

Case II: $Q = Diag(B^*A_1^{-1}B)$ with $A_1 = tridiag(A)$.

The numerical results of the GAAOR splitting iterative method in Tables 1–3, the PIU method [29], the SOR-like method [26] with respect to IT, CPU, RES for Example 1. In this table, the number outside the bracket denotes the outer iteration and inside number the inner iteration for preconditioned restarted GMRES(10) method.

From Tables 1–3, we can see that the GAAOR splitting iterative method and GAAOR-GMRES(10) costs less CPU time than the PIU method and the SOR-like method, more importantly requires much less iteration number than the others. The reason is that the GAAOR splitting iterative method does only compute the inverse of lower triangular matrix $D - \alpha L$, but not the inverse of matrix $\mathscr{A}$. As for the two cases of the GAAOR splitting iterative method, Case II is a much better choice, which has the least number of IT, costs less CPU time and RES comparing with all the other methods. Due to limited space, a large number of numerical experiments have not been listed in Tables 1–3. Furthermore, numerical experiments show that the GAAOR splitting iterative method depends on the reasonable choices of α, β, ω and γ. All the results show that we proposed the new method which is feasible and effective for the problems (1.1) in this paper.

Example 4.2. *Consider the problem (1.1) with the coefficient matrix form*

$$A = \begin{pmatrix} I\otimes T + T\otimes I & O \\ O & I\otimes T + T\otimes I \end{pmatrix} \in \mathbb{C}^{2l^2\times 2l^2}.$$

$$B = (B_1\ B_2) = (B_1\ b_1\ b_2) \in \mathbb{C}^{2l^2\times(l^2+2)},$$

$$B_1 = \begin{pmatrix} I\otimes F \\ F\otimes I \end{pmatrix} \in \mathbb{C}^{2l^2\times l^2}.$$

$$T = \frac{1}{h^2}\cdot \text{tridiag}(-1,2,-1) \in \mathbb{C}^{l\times l},$$

$$F = \text{tridiag}(-1,1,0)/h \in \mathbb{C}^{l\times l}.$$

$$b_1 = B_1\begin{pmatrix} \mathbf{e} \\ 0 \end{pmatrix}, b_2 = B_1\begin{pmatrix} 0 \\ \mathbf{e} \end{pmatrix}, \mathbf{e} = (1,1,\cdots,1)^* \in \mathbb{C}^{\frac{l^2}{2}}.$$

In this experiment, we consider the following matrix $C = diag(B^*B)$, $\otimes$ denotes the Kronecker product and $h = \frac{1}{l+1}$ is the discretization meshsize. Example 4.2 is a new technical modification in [2].

All computations, we set the right-hand side vector of the problems (1.1) such that

$$(p^*, q^*)^* = (1,1,\cdots,1)^* \in \mathbb{C}^{3l^2+2},$$

initial guess

$$(x_0^*, y_0^*)^* = (0,\cdots,0)^T \in \mathbb{C}^{3l^2+2}.$$

Furthermore, we consider two different choices of the parameter matrix Q:

Case I: $Q = Diag(B^*A^{-1}B)$,

Case II: $Q = Diag(B^*A_1^{-1}B)$ with $A_1 = tridiag(A)$.

Numerical results are compared with the PSS method [31] and the SPSS method [33], respectively. In this table, the number outside the bracket denotes the outer iteration and inside number the inner iteration for preconditioned restarted GMRES(10) method.

From Tables 4–6, we can see that the GAAOR splitting iterative method and GAAOR-GMRES(10) costs less CPU time than the PSS method and the SPSS method, more importantly requires much less iteration number than the others. As for the two cases of the GAAOR splitting iterative method, Case II is a much better choice, which has the least number of IT, costs less CPU time and RES comparing with all the other methods. Furthermore, numerical experiments show that the GAAOR splitting iterative method depends on the reasonable choices of α, β, ω and γ. All the results show that we proposed the new method which is feasible and effective for the problems (1.1) in this paper.

Table 1. Numerical results of PIU, SOR-like, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.1.

Method	Case I $\alpha = 200$ $\beta = 300$	l=8 m=64 n=66	l=16 m=256 n=258	l=24 m=576 n=578	l=32 m=1024 n=1026
	(ω, γ)	(1.92,1.64)	(0.98,0.85)	(0.45,0.26)	(0.04,0.02)
PIU	IT	44	55	78	94
	CPU	9.8639	19.9152	33.6273	94.1759
	RES	7.6538e-8	8.7126e-8	8.9537e-8	9.0724e-8
SOR-like	IT	31	38	44	49
	CPU	1.5248	4.7619	10.8813	42.0914
	RES	5.9327e-8	7.4132e-8	8.7218e-8	4.5312e-8
GAAOR	IT	19	22	27	34
	CPU	0.0319	0.3817	3.4931	17.0241
	RES	3.8653e-8	4.2159e-8	7.9123e-8	8.0361e-08
GAAOR-GMRES(10)	IT	9(2)	10(3)	10(3)	11(4)
	CPU	0.0281	0.2198	3.1105	12.2316
	RES	2.3275e-8	6.4417e-8	4.2398e-8	2.1137e-08

Table 2. Numerical results of PIU, SOR-like, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.1.

Method	Case II $\omega = 0.4$ $\gamma = 0.6$	l=8 m=64 n=66	l=16 m=256 n=258	l=24 m=576 n=578	l=32 m=1024 n=1026
	(α,β)	(0.2,0.3)	(2,3)	(20,30)	(200,300)
PIU	IT	41	53	75	92
	CPU	7.0105	15.9713	26.5328	80.2935
	RES	7.3276e-8	8.5617e-8	8.1095e-8	8.9268e-8
SOR-like	IT	29	38	45	52
	CPU	1.3126	5.0724	11.8714	42.7105
	RES	5.2591e-8	7.6319e-8	8.1295e-8	5.7821e-8
GAAOR	IT	15	18	25	28
	CPU	0.0306	0.3219	3.2173	16.1915
	RES	3.7516e-8	7.0517e-8	8.4254e-8	2.0917e-08
GAAOR-GMRES(10)	IT	6(2)	8(3)	9(3)	10(2)
	CPU	0.0203	0.1876	2.3917	10.9744
	RES	3.5286e-8	5.3891e-8	3.9152e-8	2.3729e-08

Table 3. Numerical results of PIU, SOR-like, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.1.

Method	Case II	l=8 m=64 n=66	l=16 m=256 n=258	l=24 m=576 n=578	l=32 m=1024 n=1026
	$(\alpha,\beta,$ $\omega,\gamma)$	(0.15,0.28, 0.42,0.67)	(13,31, 0.48,1.65)	(125,150, 0.85,1.27)	(240,320, 0.04,0.02)
PIU	IT	38	53	71	85
	CPU	6.1274	14.2153	23.2916	77.9831
	RES	7.3712e-8	8.4227e-8	8.5219e-8	8.5392e-8
SOR-like	IT	25	28	33	37
	CPU	0.3782	2.8619	8.6322	33.2947
	RES	4.1429e-8	7.9725e-8	8.3824e-8	4.1239e-8
GAAOR	IT	6	8	8	9
	CPU	0.0279	0.3012	2.9327	13.5121
	RES	3.1385e-8	7.6529e-8	4.7641e-8	2.1942e-08
GAAOR-GMRES(10)	IT	3(2)	4(3)	5(2)	5(3)
	CPU	0.0107	0.1196	1.8716	9.4137
	RES	2.9816e-8	6.8739e-8	4.2946e-8	2.7514e-08

Table 4. Numerical results of PSS, SPSS, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.2.

Method	Case I $\alpha = 200$ $\beta = 300$	l=8 m=128 n=66	l=16 m=512 n=258	l=24 m=1152 n=578	l=32 m=2048 n=1026
	(ω,γ)	(1.92,0.96)	(1.25,0.84)	(0.45,0.24)	(0.08,0.03)
PSS	IT	28	35	49	61
	CPU	5.2391	10.2917	21.5391	38.7412
	RES	5.8912e-8	6.2395e-8	7.1218e-8	6.4926e-8
SPSS	IT	20	27	35	47
	CPU	2.3429	5.7563	12.4671	28.9216
	RES	5.3236e-8	6.3492e-8	7.2914e-8	6.3329e-8
GAAOR	IT	8	9	11	12
	CPU	0.0284	0.2971	4.2164	18.2497
	RES	1.7149e-8	7.8763e-8	6.3529e-8	7.8137e-08
GAAOR-GMRES(10)	IT	4(2)	4(3)	5(2)	6(3)
	CPU	0.0121	0.1273	2.1376	10.2281
	RES	3.9825e-8	5.2738e-8	4.2391e-8	5.7219e-08

Table 5. Numerical results of PSS preconditioner, SPSS, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.2.

Method	Case II $\omega = 1.4$ $\gamma = 0.5$	l=8 m=128 n=66	l=16 m=512 n=258	l=24 m=1152 n=578	l=32 m=2048 n=1026
	(α,β)	(0.6,0.4)	(15,30)	(150,180)	(300,400)
PSS	IT	24	31	44	57
	CPU	4.8421	8.8615	17.3247	34.5126
	RES	3.6538e-8	5.6521e-8	6.9342e-8	8.9126e-8
SPSS	IT	21	29	41	52
	CPU	3.1372	4.7528	15.7629	31.2471
	RES	5.1287e-8	6.3258e-8	7.1426e-8	8.3127e-8
GAAOR	IT	5	6	6	7
	CPU	0.0261	0.2272	3.2301	16.2325
	RES	8.5928e-8	9.6692e-8	8.8372e-8	4.0835e-08
GAAOR-GMRES(10)	IT	3(2)	3(3)	4(1)	5(2)
	CPU	0.0114	0.1862	2.1634	9.0528
	RES	3.4185e-8	7.2351e-8	8.4429e-8	5.1573e-08

Table 6. Numerical results of PSS preconditioner, SPSS, GAAOR splitting iterative methods and GAAOR-GMRES(10) for Example 4.2.

Method	Case II	l=8 m=128 n=66	l=16 m=512 n=258	l=24 m=1152 n=578	l=32 m=2048 n=1026
	$(\alpha,\beta,$ $\omega,\gamma)$	(0.6,0.4, 1.92,0.96)	(15,30, 1.25,0.84)	(150,180, 0.45,0.24)	(300,400, 0.08,0.03)
PSS	IT	22	30	41	51
	CPU	3.7826	7.9835	15.1827	29.3924
	RES	5.4327e-8	6.5318e-8	6.2735e-8	7.2769e-8
SPSS	IT	20	26	38	47
	CPU	3.0047	4.2185	13.9846	28.9326
	RES	5.2391e-8	6.5819e-8	8.3194e-8	5.1753e-8
GAAOR	IT	5	5	6	6
	CPU	0.0246	0.1931	2.8316	13.7329
	RES	7.5514e-8	1.2462e-8	4.2304e-8	7.2898e-08
GAAOR-GMRES(10)	IT	2(2)	3(1)	3(2)	4(3)
	CPU	0.0094	0.1923	3.1946	8.4327
	RES	3.5428e-8	6.3422e-8	6.4359e-8	5.2937e-08

5. Conclusions

In this paper, based on the AOR method for generalized saddle point problems (1.1), we have proposed the AAOR method for the problems (1.1). Compared with the the PIU, the SOR-like, the PSS preconditioner and the SPSS methods, the proposed method has the least number of IT, costs less CPU time and RES, and convergence conditions are easy to be satisfied. Numerical results verified and the efficiency of the AAOR method for the problems (1.1).

However, the AAOR method involved four parameters α, β, ω and γ. It is formidable to find the optimal parameters, therefore, we did not discuss the choice of the parameters α, β, ω and γ in this paper. Consider that the validity of the proposed method depends on the selection of the four parameters, how to find easy calculated the parameters should be a direction for future research.

Acknowledgments

The author would like to express our sincere thanks to the anonymous reviewers for their valuable suggestions which greatly improved the presentation of this paper. All data included in this study are available upon request by contact with the corresponding author. The author is supported by Natural science research project of Guizhou Provincial Department of Education (No. Qian Jiao He KY Zi[2021]269) and Basic Research Projects under the Department of Science and Technology of Guizhou with grant No. [2019]1175.

Conflict of interest

The author declares there is no conflict of interest.

References

1. K. J. Arrow, L. Hurwicz, H. Uzawa, *Studies in linear and non-linear programming*, Stanford: Stanford University Press, 1958.
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. https://doi.org/10.1007/s00211-004-0521-1
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. https://doi.org/10.1007/s00211-005-0643-0
4. M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, **14** (2005), 1–137. https://doi.org/10.1017/S0962492904000212
5. H. C. Elman, G. H. Golub, Inexact and preconditioned Uzawa algorithms for saddle point problems, *SIAM J. Numer. Anal.*, **31** (1994), 1645–1661. https://doi.org/10.1137/0731085
6. G. H. Golub, D. J. Silvester, A. J. Wathen, Performance and analysis of saddle point precon ditioners for the discrete steady-state Navier-stokes equations, *Numer. Math.*, **90** (2002), 665–688. https://doi.org/10.1007/s002110100300

7. J. S. Xiong, X. B. Gao, Semi-convergence analysis of Uzawa-AOR method for singular saddle point problems, *Comp. Appl. Math.*, **36** (2017), 383–395. https://doi.org/10.1007/s40314-015-0233-4

8. J. S. Xiong, X. B. Gao, GSTS-Uzawa method for a class of complex singular saddle point problems, *Comp. Appl. Math.*, **37** (2018), 3580–3592. https://doi.org/10.1007/s40314-017-0535-9

9. H. C. Elman, A. Ramage, D. J. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling imcompressible flow, *ACM Trans. Math. Softw.*, **33** (2007), 1–18. https://doi.org/10.1145/1236463.1236469

10. J. Y. Yuan, *Iterative methods for generalized least squares linear systems*, Ph D. Thesis, IMPA, Brazil, 1993.

11. J. Y. Yuan, A. N. Iusem, Preconditioned conjugate gradient method for generalized least squares linear systems, *J. Comput. Appl. Math.*, **71** (1996), 287–297. https://doi.org/10.1016/0377-0427(95)00239-1

12. J. Y. Yuan, R. J. B. Sampaio, W. Sun, Algebraic relationships between updating and downdating least squares linear systems, *Numer. Math. J. Chin. Univ.*, **3** (1996), 203–210.

13. I. Perugia, V. Simoncini, Block-diagonal and indefinite symmetric preconditioners for mixed finite clement formulations, *Numer. Linear Algebra Appl.*, **7** (2000), 585–616.

14. Z. Z. Bai, M. Benzi, F. Chen, Modified HSS iteration methods for a class of complex symmetric linear systems, *Computing*, **87** (2010), 93–111. https://doi.org/10.1007/s00607-010-0077-0

15. Z. Z. Bai, M. Benzi, F. Chen, On preconditioned MHSS iteration methods for complex symmetric linear systems, *Numer. Algorithms*, **56** (2011), 297–317. https://doi.org/10.1007/s11075-010-9441-6

16. Z. Z. Bai, M. Benzi, F. Chen, Z. Q. Wang, Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, *IMA J. Numer. Anal.*, **33** (2013), 343–369. https://doi.org/10.1093/imanum/drs001

17. T. Z. Huang, S. L. Wu, C. X. Li, The spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for generalized saddle point problems, *J. Comput. Appl. Math.*, **229** (2009), 37–46. https://doi.org/10.1016/j.cam.2008.10.012

18. Z. H. Huang, T. Z. Huang, Spectral properties of the preconditioned AHSS iteration method for generalized saddle point problems, *Comput. Appl. Math.*, **29** (2010), 269–295.

19. K. Zhang, L. N. Wang, Efficient HSS-based preconditioners for generalized saddle point problems, *Comp. Appl. Math.*, **39** (2020), 154. https://doi.org/10.1007/s40314-020-01180-0

20. Z. Z. Bai, Y. M. Huang, M. K. Ng, Block-triangular preconditioners for system arising from edge-preserving image restoration, *J. Comput. Math.*, **28** (2010), 848–863.

21. M. Q. Jing, Y. Cao, Block triangular preconditioners for generalized saddle point problems, *Math. Numer. Sinica*, **32** (2010), 47–58. https://doi.org/10.12286/jssx.2010.1.47

22. Y. Cao, S. Li, Block triangular preconditioners based on symmetric-triangular decomposition for generalized saddle point problems, *Appl. Math. Comput.*, **358** (2019), 262–277. https://doi.org/10.1016/j.amc.2019.04.039

23. S. H. Cheng, L. T. Zhang, Generalized block triangular preconditioner for saddle point systems, *Adv. Mater. Res.*, **571** (2012), 614–617. https://doi.org/10.4028/www.scientific.net/AMR.571.614

24. G. F. Zhang, Q. H. Lu, On generalized symmetric SOR method for augmented systems, *J. Comput. Appl. Math.*, **219** (2008), 51–58. https://doi.org/10.1016/j.cam.2007.07.001

25. M. T. Darvishi, P. Hessari, Symmetric SOR method for augmented systems, *Appl. Math. Comput.*, **183** (2006), 409–415. https://doi.org/10.1016/j.amc.2006.05.094

26. M. T. Darvishi, P. Hessari. A modified symmetric successive overrelaxation method for augmented systems, *Comput. Appl. Math.*, **61** (2011), 3128–3135. https://doi.org/10.1016/j.camwa.2011.03.103

27. T. T. Feng, G. L. Chen, X. P. Guo, An accelerated SOR-LIKE method for generalised saddle point problems, *East Asian J. Appl. Math.*, **8** (2018), 70–81. https://doi.org/10.4208/eajam.261016.260817a

28. H. S. Najafi, S. A. Edalatpanah, On the modified symmetric successive over-relaxation method for augmented systems, *Comp. Appl. Math.*, **34** (2015), 607–617. https://doi.org/10.1007/s40314-014-0127-x

29. Z. Z. Bai, Z. Q. Wang, On parameterized inexact Uzawa methods for generalized saddle point problems, *Linear Algebra Appl.*, **428** (2008), 2900–2932. https://doi.org/10.1016/j.laa.2008.01.018

30. Y. Y. Zhou, G. Zhang, A generalization of parameterized inexact Uzawa method for generalized saddle point problems, *Appl. Math. Comput.*, **215** (2009), 599–607. https://doi.org/10.1016/j.amc.2009.05.036

31. L. T. Zhang, L. M. Shi, An improved generalized parameterized inexact Uzawa method for singular saddle point problems, *J. Comput. Anal. Appl.*, **23** (2017), 671–683.

32. Z. G. Huang, L. G. Wang, Z. Xu, J. J. Cui, An efficient preconditioned variant of the PSS preconditioner for generalized saddle point problems, *Appl. Math. Comput.*, **376** (2020), 125110. https://doi.org/10.1016/j.amc.2020.125110

33. H. L. Shen, H. Y. Wu, X. H. Shao, X. D. Song, The PPS method-based constraint preconditioners for generalized saddle point problems, *Comp. Appl. Math.*, **38** (2019), 21. https://doi.org/10.1007/s40314-019-0792-x

34. H. L. Shen, H. Y. Wu, X. H. Shao, A simplified PSS preconditioner for non-Hermitian generalized saddle point problems, *Appl. Math. Comput.*, **394** (2021), 125810. https://doi.org/10.1016/j.amc.2020.125810

35. C. J. Li, Z. Li, Y. Y. Nie, D. J. Evans, Generalized AOR method for the augmented system, *Int. J. Comput. Math.*, **81** (2004), 495–504. https://doi.org/10.1080/00207160410001661663

36. Y. T. Li, C. X. Li, S. L. Wu, Improving AOR method for consistent linear systems, *Appl. Math. Comput.*, **186** (2007), 379–388. https://doi.org/10.1016/j.amc.2006.07.097

37. Y. X. Zhang, H. F. Ding, W. S. He, S. F. Wang, A class of new generalized AOR method for augmented systems, In: D. Liu, H. Zhang, M. Polycarpou, C. Alippi, H. He, *Advances in neural networks-ISNN 2011*, Lecture Notes in Computer Science, Springer, **6675** (2011), 158–165. https://doi.org/10.1007/978-3-642-21105-8_20

38. M. M. Martins, W. Yousif, J. L. Sants, A variants of the AOR method for augmented systems, *Math. Comp.*, **81** (2012), 399–417. https://doi.org/10.1090/S0025-5718-2011-02483-X

39. F. Chen, N. Gao, Y. L. Jiang, On product-type generalized block AOR method for augmented linear systems, *Numer. Algebra Control Optim.*, **2** (2012), 797–809. http://dx.doi.org/10.3934/naco.2012.2.797
40. C. H. Zhang, X. Wang, X. B. Tang, Generalized AOR method for solving a class of generalized saddle point problems, *J. Comput. Appl. Math.*, **350** (2019), 67–79. https://doi.org/10.1016/j.cam.2018.10.001
41. D. M. Young, *Iterative solution of large linear systems*, Academic Press, 1971. https://doi.org/10.1016/C2013-0-11733-3