Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

Some generalizations of the new SOR-like method for solving symmetric saddle-point problems

Ruiping Wen[1*], Ruihuan Wu[2] and Jinrui Guan[2]

*Correspondence: wenrp@163.com
[1]Key Laboratory of Engineering & Computing Science, Shanxi Provincial Department of Education/Department of Mathematics, Taiyuan Normal University, Jinzhong, P.R. China
Full list of author information is available at the end of the article

Abstract

Saddle-point problems arise in many areas of scientific computing and engineering applications. Research on the efficient numerical methods of these problems has become a hot topic in recent years. In this paper, we propose some generalizations of the new SOR-like method based on the original method. Convergence of these methods is discussed under suitable restrictions on iteration parameters. Numerical experiments are given to show that these methods are effective and efficient.

Keywords: Saddle-point problems; SOR-like method; Matrix splitting iteration method; Convergence

1 Introduction

Consider the solution of the following symmetric saddle-point linear system with block 2-by-2 structure:

$$\mathcal{A}x \equiv \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix} \equiv b, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m\times n}$ is a matrix of full column rank ($m \gg n$), $B^T \in \mathbb{R}^{n\times m}$ is the transpose of the matrix B, and $p \in \mathbb{R}^m$ and $q \in \mathbb{R}^n$ are given vectors. Under such conditions, system (1.1) has a unique solution [8]. Problems of this type arise in many areas of scientific computing and engineering applications, such as computational fluid dynamics, constrained and weighted least squares estimation, constrained optimization, image reconstruction, computer graphics, and so on. For background and a comprehensive survey, we refer to [3, 5, 8, 12].

Since the matrices A and B are usually very large and sparse in applications, the direct methods are not suitable to be applied to solve system (1.1). So, more attention has been paid on the iteration methods for solving problem (1.1). Many iteration methods were found for system (1.1): the Uzawa-type methods [1], matrix splitting methods, Krylov subspace methods, and so on. See [2, 3, 5, 8, 10, 11, 14–17, 19–23, 25–28] for more details. One of the best iteration methods is SOR-like method, introduced by Golub et al. [13], which includes the Uzawa-type methods as special cases. Later, many researchers generalized or

modified the SOR-like method and studied their convergence properties for solving problem (1.1) from different points of view in recent years. For instance, Bai et al. [5] proposed a generalized SOR-like method, which has two parameters and is more effective than the SOR-like method; Shao et al. [19] extended this method and proposed a modified SOR-like method, and Guo et al. [15] presented another modified SOR-like method; Zheng et al. [27] also discussed a new SOR-like method based on a different splitting of the coefficient matrix; Darvishi and Hessari [10], Wu et al. [23], and Najafi et al. [18] considered the SSOR method. We refer to [2–28] and the references therein.

Recently, Guan et al. [14] came up with a new SOR-like method (NSOR-like) for solving the following equivalent system (1.2) of problem (1.1):

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} p \\ -q \end{pmatrix}. \tag{1.2}$$

The coefficient matrix can be split as follows:

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

where

$$\mathcal{D} = \begin{pmatrix} \frac{1}{\alpha}A & 0 \\ 0 & Q \end{pmatrix}, \qquad \mathcal{L} = \begin{pmatrix} 0 & 0 \\ B^T & \beta Q \end{pmatrix}, \qquad \mathcal{U} = \begin{pmatrix} (\frac{1}{\alpha} - 1)A & -B \\ 0 & (1-\beta)Q \end{pmatrix} \tag{1.3}$$

with a symmetric positive definite matrix $Q \in \mathbb{R}^{n\times n}$ and parameters $\alpha > 0$, $\beta \geq 0$, and $0 < \omega < 2$. The following iteration scheme was introduced for solving (1.2):

$$(\mathcal{D} - \omega\mathcal{L}) \begin{pmatrix} y_{k+1} \\ z_{k+1} \end{pmatrix} = \big[(1-\omega)\mathcal{D} + \omega\mathcal{U}\big] \begin{pmatrix} y_k \\ z_k \end{pmatrix} + \omega \begin{pmatrix} p \\ -q \end{pmatrix},$$

or, equivalently,

$$\begin{pmatrix} y_{k+1} \\ z_{k+1} \end{pmatrix} = \mathcal{M} \begin{pmatrix} y_k \\ z_k \end{pmatrix} + \mathcal{N} \begin{pmatrix} p \\ -q \end{pmatrix},$$

where

$$\begin{aligned} \mathcal{M} &= (\mathcal{D} - \omega\mathcal{L})^{-1}\big[(1-\omega)\mathcal{D} + \omega\mathcal{U}\big] \\ &= \begin{pmatrix} \frac{1}{\alpha}A & 0 \\ -\omega B^T & (1-\omega\beta)Q \end{pmatrix}^{-1} \begin{pmatrix} (\frac{1}{\alpha} - \omega)A & -\omega B \\ 0 & (1-\omega\beta)Q \end{pmatrix}, \end{aligned} \tag{1.4}$$

and $\mathcal{N} = \omega(\mathcal{D} - \omega\mathcal{L})^{-1}$.

In this paper, we generalize this method to several variants for solving problem (1.1) or (1.2). These methods include some of the above-mentioned methods as particular cases. We discuss the convergence of these methods under suitable restrictions on iteration parameters, which are very easy to use in computations. Numerical experiments are given to show that these methods are effective and efficient.

The rest of the paper is organized as follows. In Sect. 2, we propose several generalizations based on the new SOR-like method for solving problem (1.1) or (1.2). In Sect. 3, we give the convergence analysis of these methods. We use some numerical examples to show the effectiveness of them in Sect. 4. A concluding remark is drawn in Sect. 5.

2 Methods

In the section, we derive some generalizations for solving system (1.1) or (1.2) based on the new SOR-like method introduced by Guan et al. [14].

2.1 The new SSOR-like method (NSSOR-like)

By combining the NSOR-like method and its backward version the NSSOR-like method can be easily obtained for solving problem (1.2). The backward iteration scheme of the NSOR-like method is as follows:

$$(\mathcal{D}-\omega\mathcal{U})\begin{pmatrix} y_{k+1}\\ z_{k+1}\end{pmatrix}=\big[(1-\omega)\mathcal{D}+\omega\mathcal{L}\big]\begin{pmatrix} y_k\\ z_k\end{pmatrix}+\omega\begin{pmatrix} p\\ -q\end{pmatrix},$$

or, equivalently,

$$\begin{pmatrix} y_{k+1}\\ z_{k+1}\end{pmatrix}=\mathcal{M}_1\begin{pmatrix} y_k\\ z_k\end{pmatrix}+\mathcal{N}_1\begin{pmatrix} p\\ -q\end{pmatrix},$$

where

$$\begin{aligned}\mathcal{M}_1&=(\mathcal{D}-\omega\mathcal{U})^{-1}\big[(1-\omega)\mathcal{D}+\omega\mathcal{L}\big]\\&=\begin{pmatrix}\frac{\alpha\omega-\omega+1}{\alpha}A & \omega B\\ 0 & (\omega\beta-\omega+1)Q\end{pmatrix}^{-1}\begin{pmatrix}\frac{1-\omega}{\alpha}A & 0\\ \omega B^T & (\omega\beta-\omega+1)Q\end{pmatrix},\end{aligned}\tag{2.1}$$

and $\mathcal{N}_1=\omega(\mathcal{D}-\omega\mathcal{U})^{-1}$. Then the NSSOR-like method can be written as

$$\begin{pmatrix} y_{k+1}\\ z_{k+1}\end{pmatrix}=\mathcal{T}_1\begin{pmatrix} y_k\\ z_k\end{pmatrix}+\mathcal{C}_1\begin{pmatrix} p\\ -q\end{pmatrix},$$

where $\mathcal{T}_1=\mathcal{M}_1\mathcal{M}$ and $\mathcal{C}_1=\mathcal{M}_1\mathcal{N}+\mathcal{N}_1$.

The NSSOR-like method

Given two initial guesses $y_0\in\mathbb{R}^m$, $z_0\in\mathbb{R}^n$. For $k=0,1,\ldots$, until y_k and z_k converge, compute y_{k+1} and z_{k+1} from y_k and z_k by the following procedure:

$$\begin{cases} y_{k+1}=y_k-\frac{\alpha\omega(2-\omega)}{\alpha\omega-\omega+1}(A^{-1}Bz_k+y_k-A^{-1}p)\\ \qquad\quad+\frac{\alpha\omega^2(2-\omega)}{(1-\beta\omega)(\alpha\omega-\omega+1)(\beta\omega-\omega+1)}A^{-1}BQ^{-1}(q-B^Ty_k)\\ \qquad\quad+\frac{\alpha^2\omega^3(2-\omega)}{(1-\beta\omega)(\alpha\omega-\omega+1)(\beta\omega-\omega+1)}A^{-1}BQ^{-1}B^T(A^{-1}Bz_k+y_k-A^{-1}p),\\ z_{k+1}=z_k+\frac{\omega(2-\omega)}{(1-\beta\omega)(\beta\omega-\omega+1)}Q^{-1}(B^Ty_k-q)\\ \qquad\quad-\frac{\alpha\omega^2(2-\omega)}{(1-\beta\omega)(\beta\omega-\omega+1)}Q^{-1}B^T(A^{-1}Bz_k+y_k-A^{-1}p),\end{cases}$$

where $\alpha>0$, $\beta\geq 0$, $\omega>0$ are given parameters.

2.2 The generalized NSOR-like method (GNSOR-like)

By introducing a diagonal matrix $\Omega = \operatorname{diag}(\omega I_m, \tau I_n)$, where I_m, I_n, and further I are all identity matrices, we can obtain a generalization of the NSOR-like method:

$$\begin{pmatrix} y_{k+1} \\ z_{k+1} \end{pmatrix} = (\mathcal{D} - \Omega\mathcal{L})^{-1}\big((I - \Omega)\mathcal{D} + \Omega\mathcal{U}\big) \begin{pmatrix} y_k \\ z_k \end{pmatrix} + (\mathcal{D} - \Omega\mathcal{L})^{-1}\Omega \begin{pmatrix} p \\ -q \end{pmatrix}.$$

More precisely, we have the following algorithmic description of the GNSOR-like method.

The GNSOR-like method

Given two initial guesses $y_0 \in \mathbb{R}^m$, $z_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$, until y_k and z_k converge, compute y_{k+1} and z_{k+1} from y_k and z_k by the following procedure:

$$\begin{cases} y_{k+1} = (1 - \alpha\omega)y_k + \alpha\omega A^{-1}(p - Bz_k), \\ z_{k+1} = z_k + \frac{\tau}{1-\beta\tau} Q^{-1}(B^T y_{k+1} - q), \end{cases}$$

where $\alpha > 0$, $\beta \geq 0$, $\omega > 0$, $\tau > 0$ are given parameters.

Remark This method, in spirit, is analogous to the GSOR method [5]. It uses a relaxation matrix Ω for the NSOR-like method instead of a single relaxation parameter. Obviously, when $\omega = \tau$, this method reduces to the NSOR-like method mentioned in the previous section.

2.3 The generalized NSSOR-like method (GNSSOR-like)

Ordinarily, the GNSSOR-like method can also be derived by combining the symmetric technique, as introduced in [25].

The GNSSOR-like method

Given initial guesses $y_0 \in \mathbb{R}^m$, $z_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$, until y_k and z_k converge, compute y_{k+1} and z_{k+1} from y_k and z_k by the following procedure:

$$\begin{cases} y_{k+1} = y_k - \frac{\alpha\omega(2-\omega)}{\alpha\omega-\omega+1}(A^{-1}Bz_k + y_k - A^{-1}p) \\ \qquad + \frac{\alpha\omega\tau(2-\tau)}{(1-\beta\tau)(\alpha\omega-\omega+1)(\beta\tau-\tau+1)} A^{-1}BQ^{-1}(q - B^T y_k) \\ \qquad + \frac{\alpha^2\omega^2\tau(2-\tau)}{(1-\beta\tau)(\alpha\omega-\omega+1)(\beta\tau-\tau+1)} A^{-1}BQ^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \\ z_{k+1} = z_k + \frac{\tau(2-\tau)}{(1-\beta\tau)(\beta\tau-\tau+1)} Q^{-1}(B^T y_k - q) \\ \qquad - \frac{\alpha\omega\tau(2-\tau)}{(1-\beta\tau)(\beta\tau-\tau+1)} Q^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \end{cases}$$

where $\alpha > 0$, $\beta \geq 0$, $\omega > 0$, $\tau > 0$ are given parameters.

Remark It can be seen easily that, when $\omega = \tau$, this method reduces to the NSSOR-like method mentioned in the previous subsection.

With different choices of parameters, the GNSSOR-like method covers several SSOR methods as follows:

(i) When $\alpha = 1$ and $\beta = 0$, we get the SSOR method in [10],

The SSOR method 1

Given initial guesses $y_0 \in \mathbb{R}^m$, $z_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$, until y_k and z_k converge, compute y_{k+1} and z_{k+1} from y_k and z_k by the following procedure:

$$\begin{cases} y_{k+1} = y_k - \omega(2-\omega)(A^{-1}Bz_k + y_k - A^{-1}p) \\ \qquad + \frac{\omega\tau(2-\tau)}{1-\tau}A^{-1}BQ^{-1}(q - B^T y_k) \\ \qquad + \frac{\omega^2\tau(2-\tau)}{1-\tau}A^{-1}BQ^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \\ z_{k+1} = z_k + \frac{\tau(2-\tau)}{1-\tau}Q^{-1}(B^T y_k - q) \\ \qquad - \frac{\omega\tau(2-\tau)}{1-\tau}Q^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \end{cases}$$

where $\omega > 0$ and $\tau > 0$ are given parameters.

(ii) When $\alpha = 1$ and $\beta = 1/2$, we get the SSOR method in [23],

The SSOR method 2

Given initial guesses $y_0 \in \mathbb{R}^m$, $z_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$, until y_k and z_k converge, compute y_{k+1} and z_{k+1} from y_k and z_k by the following procedure:

$$\begin{cases} y_{k+1} = y_k - \omega(2-\omega)(A^{-1}Bz_k + y_k - A^{-1}p) \\ \qquad + \frac{\omega\tau(2-\tau)}{(1-\frac{\tau}{2})^2}A^{-1}BQ^{-1}(q - B^T y_k) \\ \qquad + \frac{\omega^2\tau(2-\tau)}{(1-\frac{\tau}{2})^2}A^{-1}BQ^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \\ z_{k+1} = z_k + \frac{\tau(2-\tau)}{(1-\frac{\tau}{2})^2}Q^{-1}(B^T y_k - q) \\ \qquad - \frac{\omega\tau(2-\tau)}{(1-\frac{\tau}{2})^2}Q^{-1}B^T(A^{-1}Bz_k + y_k - A^{-1}p), \end{cases}$$

where $\omega > 0$ and $\tau > 0$ are given parameters.

3 Convergence analysis

In the section, we give a convergence analysis of these methods. We need the following lemmas.

Lemma 3.1 (see [24]) *For the real quadratic equation $x^2 - bx + c = 0$, both roots are less than one in modulus if and only if $|c| < 1$ and $|b| < 1 + c$.*

Lemma 3.2 *Let $\mathcal{R}$ be the iteration matrix of the GNSOR-like method, and let λ be an eigenvalue of the matrix $\mathcal{R}$. Then $\lambda \neq 1$.*

Proof Suppose on the contrary that $\lambda = 1$ is an eigenvalue of the iteration matrix $\mathcal{R}$ and that the corresponding eigenvector is $(y^T, z^T)^T$. Then we have

$$\mathcal{R}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} y \\ z \end{pmatrix},$$

or

$$\begin{pmatrix} \frac{1}{\alpha}A & 0 \\ -\tau B^T & (1-\tau\beta)Q \end{pmatrix}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} (\frac{1}{\alpha} - \omega)A & -\omega B \\ 0 & (1-\tau\beta)Q \end{pmatrix}\begin{pmatrix} y \\ z \end{pmatrix}.$$

From this equation we can deduce that

$$B^T y = 0, \qquad Ay + Bz = 0.$$

Thus $y = -A^{-1}Bz$ and $B^T A^{-1}Bz = 0$. Since the matrix $B^T A^{-1}B$ is symmetric positive definite, we have $z = 0$. Hence we get $y = 0$, a contradiction! □

Theorem 3.3 *Let the matrix A be symmetric positive definite for the saddle-point problem (1.2), and let the matrix B be of full column rank. Let $\mathcal{R}$ be the iteration matrix of the GNSOR-like method. Then the GNSOR-like method is convergent if the parameters α, β, ω, and τ satisfy*

$$0 < \omega, \tau < 2, \qquad 0 < \alpha < \frac{2}{\omega}, \qquad 0 < \frac{\alpha\rho\omega\tau}{1 - \beta\tau} < 2(2 - \alpha\omega),$$

where ρ denotes the spectral radius of the matrix $Q^{-1}B^T A^{-1}B$.

Proof Evidently, we can see that the eigenvalues of the matrix $Q^{-1}B^T A^{-1}B$ are all real and positive. Let λ be a nonzero eigenvalue of the iteration matrix $\mathcal{R}$, and let $\binom{y}{z}$ be the corresponding eigenvector. Then we have

$$\mathcal{R}\begin{pmatrix} y \\ z \end{pmatrix} = \lambda \begin{pmatrix} y \\ z \end{pmatrix}$$

or, equivalently,

$$\lambda \begin{pmatrix} \frac{1}{\alpha}A & 0 \\ -\tau B^T & (1 - \tau\beta)Q \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} (\frac{1}{\alpha} - \omega)A & -\omega B \\ 0 & (1 - \tau\beta)Q \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix}.$$

Computations show that

$$\begin{cases} (1 - \lambda - \alpha\omega)Ay = \alpha\omega Bz, \\ (\lambda - 1)(1 - \tau\beta)Qz = \lambda\tau B^T y. \end{cases} \tag{3.1}$$

We get $(1 - \lambda - \alpha\omega)y = \alpha\omega A^{-1}Bz$ from the first equality in (3.1), and hence $\lambda\tau(1 - \lambda - \alpha\omega)B^T y = \lambda\tau\alpha\omega B^T A^{-1}Bz$ when $\lambda \neq 1 - \alpha\omega$. From this equality and the second equality in (3.1) it follows that

$$\lambda\tau\alpha\omega Q^{-1}B^T A^{-1}Bz = (\lambda - 1)(1 - \tau\beta)(1 - \lambda - \alpha\omega)z.$$

If $\lambda = 1 - \alpha\omega \neq 0$, then $Bz = 0$ and $-\alpha\omega(1 - \tau\beta)Qz = \lambda\tau B^T y$. It then follows that $z = 0$ and $y \in \text{null}(B^T)$, where $\text{null}(B^T)$ is the null space of the matrix B^T. Hence $\lambda = 1 - \alpha\omega$ is an eigenvalue of the matrix $\mathcal{R}$ with the corresponding eigenvector $(y^T, 0)^T$ with $y \in \text{null}(B^T)$.

Therefore, except for $\lambda = 1 - \alpha\omega$, the rest of the eigenvalues λ of the matrix $\mathcal{R}$ and all the eigenvalues μ of the matrix $Q^{-1}B^T A^{-1}B$ are of the functional relationship

$$\lambda\alpha\omega\tau\mu = (\lambda - 1)(1 - \tau\beta)(1 - \lambda - \alpha\omega),$$

that is, λ satisfies the quadratic equation

$$\lambda^2 - \left(2 - \alpha\omega - \frac{\alpha\mu\omega\tau}{1-\tau\beta}\right)\lambda + 1 - \alpha\omega = 0. \tag{3.2}$$

By Lemma 3.1 we know that $\lambda = 1 - \alpha\omega$ and both roots λ of Eq. (3.2) satisfy $|\lambda| < 1$ if and only if

$$|1 - \alpha\omega| < 1, \qquad \left|2 - \alpha\omega - \frac{\alpha\mu\omega\tau}{1-\beta\tau}\right| < 2 - \alpha\omega.$$

Thus we can deduce that

$$0 < \alpha < \frac{2}{\omega}, \qquad 0 < \frac{\alpha\rho\omega\tau}{1-\beta\tau} < 2(2 - \alpha\omega),$$

where ρ denotes the spectral radius of the matrix $Q^{-1}B^TA^{-1}B$.

The proof of the theorem has been completed. □

Lemma 3.4 *Let $\mathcal{T}$ be the iteration matrix of the GNSSOR-like method. Then we have*:

(i) $\lambda = \frac{(1-\omega+\omega\alpha)-\omega\alpha(2-\omega)}{1-\omega+\omega\alpha}$ *is an eigenvalue of the matrix $\mathcal{T}$ with multiplicity $m - n$ at least.*

(ii) *A real number $\lambda \neq \frac{(1-\omega+\omega\alpha)-\omega\alpha(2-\omega)}{1-\omega+\omega\alpha}$ is an eigenvalue of the matrix $\mathcal{T}$ if and only if there exists a real eigenvalue μ of the matrix $Q^{-1}B^TA^{-1}B$ such that λ is a zero point of*

$$\begin{aligned}&(\lambda-1)(1-\tau+\beta\tau)(1-\beta\tau)\left[\left(\frac{1}{\alpha}-\omega\right)(1-\omega) - \lambda\left(\frac{1}{\alpha}+\omega-\frac{\omega}{\alpha}\right)\right]\\ &\quad - \lambda\omega\tau(2-\omega)(2-\tau)\mu.\end{aligned} \tag{3.3}$$

Proof Let $\lambda \neq 0$ be an eigenvalue of the matrix $\mathcal{T}$, and let $u = \binom{y}{z}$ be the corresponding eigenvector. Then we have

$$\mathcal{T}u = \lambda u$$

or, equivalently,

$$(1-\lambda)(\mathcal{D} - \Omega\mathcal{U})u = (2I - \Omega)\mathcal{D}(\mathcal{D} - \Omega\mathcal{L})^{-1}\Omega Au.$$

Computations show that

$$\big[(1-\lambda)(1-\omega+\omega\alpha) - \omega\alpha(2-\omega)\big]Ay = \omega\alpha(\lambda+1-\omega)Bz, \tag{3.4}$$

and we obtain that

$$(1-\lambda)(1-\tau+\beta\tau)(1-\beta\tau)z - \omega\alpha\tau(2-\tau)Q^{-1}B^TA^{-1}Bz = \tau(2-\tau)(\alpha\omega-1)Q^{-1}B^Ty. \tag{3.5}$$

Let $\lambda = \frac{(1-\omega+\omega\alpha)-\omega\alpha(2-\omega)}{1-\omega+\omega\alpha}$. From (3.4) we have $A^{-1}Bz = 0$.

We get $B^Ty = 0$ and $z = 0$ since the matrix A is symmetric positive definite and the matrix B is of full column rank. Note that $\operatorname{rank}(B^T) = n$, and then there exist $m - n$ independent

nonzero solutions of $B^T y = 0$, that is, there exist $m - n$ corresponding eigenvectors of $\lambda = \frac{(1-\omega+\omega\alpha)-\omega\alpha(2-\omega)}{1-\omega+\omega\alpha}$.

When $\lambda \neq \frac{(1-\omega+\omega\alpha)-\omega\alpha(2-\omega)}{1-\omega+\omega\alpha}$, from (3.4) we have

$$y = \frac{\alpha\omega(\lambda + 1 - \omega)}{(1-\lambda)(1-\omega+\alpha\omega) - \alpha\omega(2-\omega)} A^{-1}Bz.$$

Substituting y into (3.5), we get

$$\begin{aligned}&(1-\lambda)(1-\tau+\beta\tau)(1-\beta\tau)\frac{1}{\alpha}z\\&\quad= \left[\omega\tau(2-\tau) - \frac{\omega\tau(2-\tau)(1-\alpha\omega)(1-\omega+\lambda)}{(1-\lambda)(1-\omega+\alpha\omega)-\alpha\omega(2-\omega)}\right] Q^{-1}B^T A^{-1}Bz\end{aligned}$$

or, equivalently,

$$\begin{aligned}&(1-\lambda)(1-\tau+\beta\tau)(1-\beta\tau)\frac{1}{\alpha}\left[(1-\lambda)\left((1-\omega)\frac{1}{\alpha}+\omega\right) - \omega(2-\omega)\right]z\\&\quad= \omega\tau(2-\tau)\left\{\left[(1-\lambda)\left((1-\omega)\frac{1}{\alpha}+\omega\right) - \omega(2-\omega)\right] - \left(\omega - \frac{1}{\alpha}\right)(\omega-\lambda-1)\right\}\\&\qquad\times Q^{-1}B^T A^{-1}Bz.\end{aligned}$$

Since μ is an eigenvalue of the matrix $Q^{-1}B^T A^{-1}B$, we have

$$\begin{aligned}&(1-\lambda)(1-\tau+\beta\tau)(1-\beta\tau)\frac{1}{\alpha}\left[(1-\lambda)\left((1-\omega)\frac{1}{\alpha}+\omega\right) - \omega(2-\omega)\right]\\&\quad= \omega\tau(2-\tau)\left\{\left[(1-\lambda)\left((1-\omega)\frac{1}{\alpha}+\omega\right) - \omega(2-\omega)\right] - \left(\omega - \frac{1}{\alpha}\right)(\omega-\lambda-1)\right\}\mu;\end{aligned}$$

simply,

$$\begin{aligned}&(\lambda-1)(1-\tau+\beta\tau)(1-\beta\tau)\left[\left(\frac{1}{\alpha}-\omega\right)(1-\omega) - \lambda\left(\frac{1}{\alpha}+\omega-\frac{\omega}{\alpha}\right)\right]\\&\quad= \lambda\omega\tau(2-\omega)(2-\tau)\mu.\end{aligned}$$

Conversely, we can also trivially prove the following: □

Theorem 3.5 *Let the matrix A be symmetric positive definite, and let the matrix B be of full column rank in Eq. (1.2). Assume that α, β, and ω satisfy $(1-\beta\tau)(1-\tau+\beta\tau)(1-\omega+\omega\alpha) \neq 0$. We choose a nonsingular matrix Q such that all eigenvalues of the matrix $Q^{-1}B^T A^{-1}B$ are real. Let $\mu_{\max}$, $\mu_{\min}$ be the largest and the smallest eigenvalues of the matrix $Q^{-1}B^T A^{-1}B$, respectively. Then*:

(i) *If $\mu_{\min} > 0$, then the GNSSOR-like method is convergent if and only if*

$$0 < \omega < 2; \qquad 0 < \tau < 2;$$

$$\begin{cases}\frac{1}{\alpha} > \frac{\omega^2}{2(\omega-1)}, & \text{if } 0 < \omega < 1,\\ \frac{1}{\alpha} < \frac{\omega^2}{2(\omega-1)}, & \text{if } 1 < \omega < 2;\end{cases}$$

$$(1-\tau+\beta\tau)(1-\beta\tau)>0;$$
$$\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu_{\max}}{(1-\tau+\beta\tau)(1-\beta\tau)(1-\omega+\omega\alpha)}<2\left[1+\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}\right].$$

(ii) *If* $\mu_{\max}<0$, *then the GNSSOR-like method is convergent if and only if*

$$0<\omega<2;\qquad 0<\tau<2;$$
$$\begin{cases}\frac{1}{\alpha}>\frac{\omega^2}{2(\omega-1)}, & \text{if } 0<\omega<1,\\ \frac{1}{\alpha}<\frac{\omega^2}{2(\omega-1)}, & \text{if } 1<\omega<2;\end{cases}$$
$$(1-\tau+\beta\tau)(1-\beta\tau)<0;$$
$$\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu_{\min}}{(1-\tau+\beta\tau)(1-\beta\tau)(1-\omega+\omega\alpha)}<2\left[1+\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}\right].$$

Proof From (3.3) we get that

$$\lambda^2-\lambda\left[2-\frac{\omega\alpha(2-\omega)}{1+\alpha\omega-\omega}-\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1-\omega+\omega\alpha)}\right]+\left(1-\frac{\omega\alpha(2-\omega)}{1+\alpha\omega-\omega}\right)=0.$$

By Lemma 3.1, $|\lambda|<1$ if and only if

$$\left|\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}\right|<1 \tag{3.6}$$

and

$$\left|\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}+1-\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)}\right|<1+\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}.$$

Computations show that

$$\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)}>0,$$
$$\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)}<2+\frac{(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega};$$

if $\mu_{\min}>0$, then we have

$$(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)>0,$$
$$\begin{aligned}0&<\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)}\\&\le\frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu_{\max}}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)}\\&<2+\frac{2(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega};\end{aligned}$$

if $\mu_{\max}<0$, then we have

$$(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)<0,$$

$$
\begin{aligned}
0 &< \frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)} \\
&\leq \frac{\alpha\omega\tau(2-\omega)(2-\tau)\mu_{\min}}{(1-\tau+\beta\tau)(1-\beta\tau)(1+\alpha\omega-\omega)} \\
&< 2+\frac{2(1-\alpha\omega)(1-\omega)}{1+\alpha\omega-\omega}.
\end{aligned}
$$

From (3.6) we have

$$
\begin{aligned}
&\frac{1}{\alpha} > \frac{\omega^2}{2(\omega-1)}, \quad \text{if } 0<\omega<1, \\
&\frac{1}{\alpha} < \frac{\omega^2}{2(\omega-1)}, \quad \text{if } 1<\omega<2.
\end{aligned}
$$

Cosidering $1+\alpha\omega-\omega>0$, we obtain

$$
\begin{cases}
\frac{1}{\alpha} > \frac{\omega^2}{2(\omega-1)}, & \text{if } 0<\omega<1, \\
\frac{1}{\alpha} < \frac{\omega^2}{2(\omega-1)}, & \text{if } 1<\omega<2.
\end{cases}
$$

□

4 Numerical experiments

In this section, we test several experiments to show the effectiveness of the GNSOR-like method and compare it with the SOR-like method in [13], MSOR-like method in [19], and MSOR-like method in [15]. We present computational results in terms of the numbers of iterations (denoted by IT) and computing time (denoted by CPU). We denote the choices of parameters α, β, ω, τ by $\alpha_{\exp}$, $\beta_{\exp}$, $\omega_{\exp}$, $\tau_{\exp}$ in our test, respectively. In our implementations, all iterations are run in MATLAB R2015a on a personal computer and are terminated when the current iterate satisfies $RES < 10^{-6}$ or the number of iterations is more than 1000. In our experiments, the residue is defined to be

$$
RES := \sqrt{\|Ay_k + Bz_k - p\|^2 + \left\|B^T y_k - q\right\|^2} < 10^{-6},
$$

the right-hand-side vector $(p^T, q^T)^T = \mathcal{A}e$ with $e = (1,1,\ldots,1)^T$, and the initial vectors are set to be $y_0 = (0,0,\ldots,0)^T$ and $z_0 = (0,0,\ldots,0)^T$.

Example 4.1 ([4]) Consider the saddle-point problem (1.1) in which

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

and

$$
T = \frac{1}{h^2}\,\text{tridiag}(-1,2,-1) \in \mathbb{R}^{l\times l}, \qquad F = \frac{1}{h}\,\text{tridiag}(-1,1,0) \in \mathbb{R}^{l\times l},
$$

where $h = \frac{1}{l+1}$ is the mesh size, and $\otimes$ be the Kronecker product.

In the test, we set $m = 2l^2$ and $n = l^2$. The choices of the matrix Q are listed in Table 1 for Example 4.1. We have the following computational results summarized in Table 2.

Table 1 Choices of matrix Q for Example 4.1

	Matrix Q	Description
Case I	$B^T\hat{A}^{-1}B$	$\hat{A} = \text{diag}(A)$
Case II	$B^T\hat{A}^{-1}B$	$\hat{A} = \text{tridiag}(A)$

Table 2 ITs and CPUs of SOR-like, MSOR-like [15], NSOR-like [14], GNSOR-like, and NSSOR-like for Example 4.1

m			128	512	1152	2048	8192
n			64	256	576	1024	4096
$m+n$			192	768	1728	3072	12,288
Case I	SOR-like	ω_{exp}	0.4644	0.2720	0.1886	0.1386	0.0741
		IT	73	202	221	315	657
		CPU	0.0284	0.1321	0.3670	0.9262	9.0859
	MSOR-like [15]	α_{exp}	1.7212	1.8256	1.8737	1.9136	1.9201
		ω_{exp}	0.3159	0.1873	0.1328	0.1020	0.0541
		IT	67	140	183	223	462
		CPU	0.0313	0.1171	0.2937	0.6137	6.3687
	NSOR-like [14]	α_{exp}	1.7212	1.8256	1.8689	1.9699	1.9219
		β_{exp}	0.3700	0.3655	0.3492	0.2399	0.2148
		ω_{exp}	0.3159	0.1873	0.1328	0.1001	0.0545
		IT	55	105	160	212	457
		CPU	0.0250	0.1031	0.2624	0.5687	6.2634
	GNSOR-like	α_{exp}	1.7212	1.8256	1.8689	1.9699	1.9219
		β_{exp}	0.3700	0.3655	0.3492	0.2399	0.2148
		ω_{exp}	0.3250	0.1923	0.1361	0.1029	0.0555
		τ_{exp}	0.3130	0.1830	0.1282	0.0985	0.0534
		IT	50	101	155	211	447
		CPU	0.0154	0.1019	0.2561	0.5663	6.1488
Case II	SOR-like	ω_{exp}	0.5958	0.3657	0.2215	0.1961	0.0945
		IT	56	103	183	216	509
		CPU	0.0310	0.1343	0.5855	1.4298	16.1417
	MSOR-like [15]	α_{exp}	1.6599	1.7732	1.8315	1.8753	1.9200
		ω_{exp}	0.3996	0.2498	0.1806	0.1257	0.0745
		IT	45	94	167	175	330
		CPU	0.0306	0.1194	0.6196	1.2068	10.4589
	NSOR-like [14]	α_{exp}	1.6469	1.7582	1.8318	1.8750	1.9100
		β_{exp}	0.3397	0.3438	0.3640	0.3541	0.4001
		ω_{exp}	0.3986	0.2513	0.1812	0.1420	0.0758
		IT	39	77	116	155	320
		CPU	0.0164	0.0916	0.4162	1.0173	9.4823
	GNSOR-like	α_{exp}	1.6469	1.7582	1.8318	1.8750	1.9100
		β_{exp}	0.3397	0.3438	0.3640	0.3541	0.4001
		ω_{exp}	0.4030	0.2513	0.1825	0.1402	0.0758
		τ_{exp}	0.4210	0.2581	0.1823	0.1443	0.0755
		IT	37	75	115	154	319
		CPU	0.0134	0.0871	0.3219	1.0129	9.4803

From Table 2 we can see that the GNSOR-like method requires much less iteration number than NSOR-like [14], SOR-like, MSOR-like [15], so that it costs less CPU time than the others. So, the GNSOR-like method is effective for solving the saddle-point problem (1.1).

5 Concluding remark

In this paper, we first presented several iteration methods for solving the saddle-point problem (1.1) and then give their convergence results. The GNSOR-like method is faster and requires much less iteration steps than the other methods mentioned in the paper.

Acknowledgements
The authors are very much indebted to the anonymous referees for their helpful comments and suggestions, which greatly improved the original manuscript.

Funding
This work is supported by grants from the NSF of Shanxi Province (201601D011004).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Author details
[1]Key Laboratory of Engineering & Computing Science, Shanxi Provincial Department of Education/Department of Mathematics, Taiyuan Normal University, Jinzhong, P.R. China. [2]Department of Mathematics, Taiyuan Normal University, Taiyuan Normal University, Jinzhong, P.R. China.

Publisher's Note

Received: 11 February 2018 Accepted: 24 May 2018 Published online: 26 June 2018

References

1. Arrow, K., Hurwicz, L., Uzawa, H.: Studies in Nonlinear Programming. Stanford University Press, Stanford (1958)
2. Bai, Z.Z.: Optimal parameters in the HSS-like methods for saddle point problems. Numer. Linear Algebra Appl. **16**, 447–479 (2009)
3. Bai, Z.Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
4. Bai, Z.Z., Golub, G.H., Pan, J.Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
5. Bai, Z.Z., Parlett, B.N., Wang, Z.Q.: On generalized successive overrelaxation methods for augmented linear systems. Numer. Math. **102**, 1–38 (2005)
6. Bai, Z.Z., Wang, Z.Q.: On parameterized inexact Uzawa methods for generalized saddle point problems. Linear Algebra Appl. **428**, 2900–2932 (2008)
7. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 20–41 (2004)
8. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
9. Benzi, M., Simoncini, V.: On the eigenvalues of a class of saddle point matrices. Numer. Math. **103**, 173–196 (2006)
10. Darvishi, M.T., Hessari, P.: Symmetric SOR method for augmented systems. J. Comput. Appl. Math. **183**, 409–415 (2006)
11. Elman, H.C., Golub, G.H.: Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J. Numer. Anal. **31**, 1645–1661 (1994)
12. Elman, H.C., Silvester, D.: Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations. SIAM J. Sci. Comput. **17**, 33–46 (1996)
13. Golub, G.H., Wu, X., Yuan, J.Y.: SOR-like methods for augmented systems. BIT Numer. Math. **41**, 71–85 (2001)
14. Guan, J.R., Ren, F.J., Feng, Y.H.: New SOR-like iteration method for saddle point problems. Math. Appl. (accepted, in press)
15. Guo, P., Li, C.X., Wu, S.L.: A modified SOR-like method for the augmented systems. J. Comput. Appl. Math. **274**, 58–69 (2015)
16. Li, C.J., Li, Z., Nie, Y.Y., Evans, D.J.: Generalized AOR method for the augmented systems. Int. J. Comput. Math. **81**, 495–504 (2004)
17. Li, J., Zhang, N.M.: A triple-parameter modified SSOR method for solving singular saddle point problems. BIT Numer. Math. **56**, 501–521 (2016)
18. Saberi Najafi, H., Edalatpanah, S.A.: On the modified symmetric successive over-relaxation method for augmented systems. J. Comput. Appl. Math. **34**, 607–617 (2015)
19. Shao, X.H., Li, Z., Li, C.J.: Modified SOR-like methods for the augmented systems. Int. J. Comput. Math. **84**, 1653–1662 (2007)
20. Wang, H.D., Huang, Z.D.: On a new SSOR-like method with four parameters for the augmented systems. East Asian J. Appl. Math. **7**, 82–100 (2017)
21. Wen, R.P., Li, S.D., Meng, G.Y.: SOR-like methods with optimization model for augmented linear systems. East Asian J. Appl. Math. **7**, 101–115 (2017)
22. Wen, R.P., Wang, C.L., Yan, X.H.: Generalizations of nonstationary multisplitting iterative method for symmetric positive definite linear systems. Appl. Math. Comput. **216**, 1707–1714 (2010)
23. Wu, S.L., Huang, T.Z., Zhao, X.L.: A modified SSOR iterative method for augmented systems. J. Comput. Appl. Math. **228**, 424–433 (2009)
24. Young, D.M.: Iterative Solution of Large Linear Systems. Academic Press, New York (1971)
25. Zhang, G.F., Lu, Q.H.: On generalized symmetric SOR method for augmented systems. J. Comput. Appl. Math. **219**, 51–58 (2008)
26. Zheng, B., Wang, K., Wu, Y.J.: SSOR-like methods for saddle point problem. Int. J. Comput. Math. **86**, 1405–1423 (2009)
27. Zheng, Q.Q., Ma, C.F.: A new SOR-like method for the saddle point problems. Appl. Math. Comput. **233**, 421–429 (2014)
28. Zheng, Q.Q., Ma, C.F.: A class of triangular splitting methods for saddle point problems. J. Comput. Appl. Math. **298**, 13–23 (2016)