East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.010916.261116a

Vol. 7, No. 1, pp. 101-115
February 2017

SOR-like Methods with Optimization Model for Augmented Linear Systems

Rui-Ping Wen[1,*], Su-Dan Li[1] and Guo-Yan Meng[2]

[1] *Department of Mathematics, Taiyuan Normal University, Taiyuan, 030012, China.*
[2] *Department of Mathematics, Xinzhou Teachers College, Xinzhou, 034000, China.*

Received 1 September 2016; Accepted (in revised version) 26 November 2016.

Abstract. There has been a lot of study on the SOR-like methods for solving the augmented system of linear equations since the outstanding work of Golub, Wu and Yuan (BIT 41(2001)71-85) was presented fifteen years ago. Based on the SOR-like methods, we establish a class of accelerated SOR-like methods for large sparse augmented linear systems by making use of optimization technique, which will find the optimal relaxation parameter ω by optimization models. We demonstrate the convergence theory of the new methods under suitable restrictions. The numerical examples show these methods are effective.

AMS subject classifications: 65F10, 65F50, 15A06

Key words: SOR-like method, optimization, augmented linear systems, convergence.

1. Introduction

Consider the augmented linear systems of the form

$$\begin{pmatrix} A & B \\ B^T & O \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}=\begin{pmatrix} b \\ q \end{pmatrix}, \tag{1.1}$$

where $A\in\mathbb{R}^{n\times n}$ is symmetric positive definite, $O\in\mathbb{R}^{m\times m}$ is zero, $B\in\mathbb{R}^{n\times m}$ has full column rank, $x,b\in\mathbb{R}^n, y,q\in\mathbb{R}^m$, $n\gg m$, and B^T is the transpose of the matrix B. These assumptions guarantee the existence and uniqueness of the solution of the system of linear equations (1.1). For the sake of simplicity, also we can consider the following equivalent form of (1.1)

$$\mathscr{A}u=\begin{pmatrix} A & B \\ -B^T & O \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}=\begin{pmatrix} b \\ -q \end{pmatrix}=f, \tag{1.2}$$

*Corresponding author. *Email addresses:* wenrp@163.com (R.-P. Wen), lisudan1990@163.com (S.-D. Li), mgy1226@126.com (G.-Y. Meng)

A large amount of study has been devoted to the augmented linear systems of the form (1.1) (or (1.2)). The reason for this interest is the fact that such problems appear in many different applications of scientific computing, such as the Karush-Kuhn-Tuker (KKT) conditions for linearly constrained quadratic programming problems, or saddle point problems, or an equilibrium system ([11, 26, 29]), the finite element method for solving the Navier-Stokes equation, or elasticity problems, or second-order elliptic problems ([17, 18]), generalized least squares problems ([20, 30]) and from Lagrange multiplier methods ([19]). See [3] and [10] for a comprehensive summary.

Because of the ubiquitous nature of augmented linear systems, numerical methods and results on the problems have published in a wide variety of books, journals and conference proceedings. Very effective solvers have been developed for the important classes of problems (1.1) (or (1.2)). When the matrix blocks A and B are large and sparse, iterative methods become more attractive than direct methods for solving the augmented linear system (1.1) (or (1.2)). For many years deriving an efficient iterative method based on a splitting of the coefficient matrix $\mathscr{A}$ for solving the system of linear equations (1.1) (or (1.2)) has been an important and active topic. Many iterative methods were proposed for solving the system (1.1) (or (1.2)). There are two sub-approaches to the iterative methods: one is "matrix splitting methods", another is "Krylov subspace methods". See [2, 5, 8–11, 20, 34, 35] for more details. Among these methods, the best known and the oldest iterations are the Uzawa and preconditioned Uzawa methods ([1, 12, 16]), but they are special cases of the SOR-like methods presented in [20]. The SOR-like methods, together with the inexact Uzawa algorithm studied in [16], are usually the methods of choice for solving the augmented linear systems (1.1) (or (1.2)), as they are simple, efficient and require small computer memory. Later, many researchers generalized or modified the SOR-like methods and studied their convergence properties for solving the augmented systems from different view in recent years, we refer to [6, 7, 13–15, 21–23, 25, 31–35] and the references therein.

The idea of minimizing the norm of either the error or the residual so that the numerically optimal value of the iterative parameter is determined, first introduced in [4], used to compute a numerically optimal relaxation parameter for the successive overrelaxation (SOR) iterative methods for solving the system of linear equations. Based on the standard quadratic programming technique, the authors of this paper and their collaborators seem to be the first to introduce the auto-optimal weighting matrices for parallel multisplitting iterative method (see [27]) and be the first to come up with the quasi-Chebyshev accelerated (QCA) method to a convergent splitting iteration (see [28]). The optimal weighting matrices of this multisplitting method and the optimal parameter of the QCA method are generated by optimization models for solving the linear systems. These motivated us to accelerate the SOR-like iterative methods, resulting in a class of new SOR-like methods with optimization model for the augmented systems (1.1) (or (1.2)).

The rest of this paper is organized as follows. In Section 2, we state and briefly summary the existing schemes resulting from the SOR-like methods. In Section 3, we present a class of new SOR-like methods with optimization models and provide its convergence results. Two numerical examples further show the proposed methods are effective than the SOR-like and the SSOR-like methods in Section 4. Finally, we end the paper with a conclusion

in Section 5.

2. Existing Methods Resulted from SOR-like Scheme

By the introduction we had in Section 1, we know that there exist many variants of the SOR-like methods. Let us briefly survey the several existing related methods that have been taken to iteratively solve the augmented linear systems (1.1) (or (1.2)), which result from the SOR-like scheme and were designed to do the job ([8,15,20,31,33]) from the view of the parameter.

It is known that the SOR method is a simple stationary iterative method which is popular in engineering applications. Its alternative variant is very useful in a parallel environment. Golub, Wu and Yuan [20] proposed several variants of the SOR method for solving the systems (1.1) (or (1.2)), which is the following SOR-like method.

For the following splitting of the coefficient matrix $\mathscr{A}$ of the augmented linear systems (1.2),

$$\mathscr{A} \equiv \begin{pmatrix} A & B \\ -B^T & O \end{pmatrix} = \mathscr{D} - \mathscr{L} - \mathscr{U},$$

where

$$\mathscr{D} = \begin{pmatrix} A & O \\ O & Q \end{pmatrix}, \qquad \mathscr{L} = \begin{pmatrix} O & O \\ B^T & O \end{pmatrix}, \qquad \mathscr{U} = \begin{pmatrix} O & -B \\ O & Q \end{pmatrix},$$

and $Q \in \mathbb{R}^{m\times m}$ is a prescribed nonsingular and symmetric matrix, the following iterative method was proposed in 2001.

SOR-Like Method ([20]). Let $Q \in \mathbb{R}^{m\times m}$ be a nonsingular and symmetric matrix. Given two initial guesses $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^m$, and a relaxation factor $\omega > 0$. For $k = 0,1,2,\cdots$ until the iteration sequence $\{({x^{(k)}}^T, {y^{(k)}}^T)^T\}$ is convergent, compute

$$\begin{cases} x^{(k+1)} = (1-\omega)x^{(k)} + \omega A^{-1}(b - By^{(k)}), \\ y^{(k+1)} = y^{(k)} + \omega Q^{-1}(B^T x^{(k+1)} - q). \end{cases}$$

Here, Q is an approximate (preconditioning) matrix of the Schur complement matrix $\mathscr{S} = B^T A^{-1} B$.

This method can be regarded as the iteration based on the following splitting:

$$\mathscr{A} \equiv \begin{pmatrix} A & B \\ -B^T & O \end{pmatrix} = \mathscr{M}_\omega - \mathscr{N}_\omega,$$

where

$$\mathscr{M}_\omega = \frac{1}{\omega}(\mathscr{D} - \omega\mathscr{L}) = \begin{pmatrix} \frac{1}{\omega}A & O \\ -B^T & \frac{1}{\omega}Q \end{pmatrix},$$

and

$$\mathscr{N}_\omega = \mathscr{M}_\omega - \mathscr{A} = \frac{1}{\omega}\big[(1-\omega)\mathscr{D} + \omega\mathscr{U}\big] = \begin{pmatrix} (\frac{1}{\omega}-1)A & -B \\ O & \frac{1}{\omega}Q \end{pmatrix}.$$

The iteration matrix of the SOR-like iteration can be defined as

$$\mathscr{T}_\omega = \mathscr{M}_\omega^{-1}\mathscr{N}_\omega.$$

Remark 2.1. The SOR-like algorithm is usually the method of choice for solving the augmented linear systems (1.1) (or (1.2)), as it is simple, efficient, requires small computer memory and less arithmetic work per iteration step relatively, but has the burden of choosing a good, or an optimal relaxation parameter in order to achieve a comparable rate of convergence.

In [20], the authors proved the convergence of the SOR-like method, and determined its optimal relaxation parameter as well as the corresponding optimal convergence factor. These results are summarized as the following theorem.

Let $\mathscr{H} = Q^{-1}B^TA^{-1}B$, and μ be a nonzero eigenvalue of the matrix $\mathscr{H}$ as well as ρ be its spectral radius, i.e., $\rho = \rho(\mathscr{H})$.

Theorem 2.1 ([20, Theorems 3.1-3.2]). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $B \in \mathbb{R}^{n\times m}$ be of full column rank. Assume that all eigenvalues μ of the matrix $\mathscr{H}$ are real, and denote the smallest and the largest nonzero eigenvalues of the matrix $\mathscr{H}$ by $\mu_{\min}$ and $\mu_{\max}$, respectively. Then,*
(i) if $\mu > 0$, the SOR-like method is convergent for all ω such that

$$0 < \omega < \frac{4}{\sqrt{4\rho+1}+1};$$

(ii) if $\mu_{\min} > 1/4$, it holds that

$$\rho(\mathscr{T}_\omega) = \begin{cases} \sqrt{1-\omega}, & \text{if } 0 < \omega \le \frac{2\sqrt{\rho}-1}{\rho}, \\ \frac{1}{2}\left[|2-\omega-\omega^2\rho| + \omega\sqrt{(\omega\rho+1)^2-4\rho}\right], & \text{if } \frac{2\sqrt{\rho}-1}{\rho} \le \omega < \frac{4}{\sqrt{4\rho+1}+1}. \end{cases}$$

Moreover, the optimal parameter $\omega_\star$ and the optimal convergence factor $\rho(\mathscr{T}_{\omega_\star})$ are given respectively by

$$\omega_\star = \frac{2\sqrt{\rho}-1}{\rho} \le 1 \ \text{ and } \ \rho(\mathscr{T}_{\omega_\star}) = \frac{|\sqrt{\rho}-1|}{\sqrt{\rho}};$$

(iii) if $\mu_{\min} \le 1/4$, it holds that

$$\rho(\mathscr{T}_\omega) = \begin{cases} \frac{1}{2}\left[|2-\omega-\omega^2\mu_{\min}| + \omega\sqrt{(\omega\mu_{\min}+1)^2-4\mu_{\min}}\right], & \text{if } 0 < \omega \le \omega_\star, \\ \frac{1}{2}\left[|2-\omega-\omega^2\rho| + \omega\sqrt{(\omega\rho+1)^2-4\rho}\right], & \text{if } \omega_\star < \omega < \frac{4}{\sqrt{4\rho+1}+1}. \end{cases}$$

Moreover, the optimal relaxation parameter $\omega_\star$ satisfies $\omega_\star < 2$ and it is the positive root of the equation

$$|2-\omega-\omega^2\mu_{\min}| + \omega\sqrt{(\omega\mu_{\min}+1)^2-4\mu_{\min}} = |2-\omega-\omega^2\rho| + \omega\sqrt{(\omega\rho+1)^2-4\rho}.$$

By introducing two parameters ω, τ (or say, a parameter matrix Ω), Bai, Parlett, Wang [8] proposed the following generalized successive overrelaxation (GSOR) scheme for the systems (1.1) (or (1.2)).

GSOR Method ([8, Method 2.1]). Let $Q \in \mathbb{R}^{m\times m}$ be a nonsingular and symmetric matrix. Given two initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^m$, and two relaxation factors $\omega, \tau \neq 0$. For $k = 0, 1, 2, \cdots$ until the iteration sequence $\{({x^{(k)}}^T, {y^{(k)}}^T)^T\}$ is convergent, compute

$$\begin{cases} x^{(k+1)} = (1-\omega)x^{(k)} + \omega A^{-1}(b - By^{(k)}), \\ y^{(k+1)} = y^{(k)} + \tau Q^{-1}(B^T x^{(k+1)} - q). \end{cases}$$

Here, Q is an approximate (preconditioning) matrix of the Schur complement matrix $\mathscr{S} = B^T A^{-1} B$.

Or equivalently, compute by the following procedure

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = (\mathscr{D} - \Omega\mathscr{L})^{-1}\big[(I-\Omega)\mathscr{D} + \Omega\mathscr{U}\big]\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + (\mathscr{D} - \Omega\mathscr{L})^{-1}\Omega\begin{pmatrix} b \\ -q \end{pmatrix}$$

with I is the identity matrix and

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

where ω and τ are two nonzero reals, $I_m \in \mathbb{R}^{m\times m}$ and $I_n \in \mathbb{R}^{n\times n}$ are the m-by-m and n-by-n identity matrices, respectively.

Clearly, this method can be regarded as iteration based on the following splitting:

$$\mathscr{A} \equiv \begin{pmatrix} A & B \\ -B^T & O \end{pmatrix} = \mathscr{M}_{\omega,\tau} - \mathscr{N}_{\omega,\tau},$$

where

$$\mathscr{M}_{\omega,\tau} = \Omega^{-1}(\mathscr{D} - \Omega\mathscr{L}) = \begin{pmatrix} \frac{1}{\omega}A & O \\ -B^T & \frac{1}{\tau}Q \end{pmatrix},$$

and

$$\mathscr{N}_{\omega,\tau} = \mathscr{M}_{\omega,\tau} - \mathscr{A} = (I-\Omega)D + \Omega\mathscr{U} = \begin{pmatrix} (\frac{1}{\omega}-1)A & -B \\ O & \frac{1}{\tau}Q \end{pmatrix}.$$

The iteration matrix of the GSOR iteration can be denoted as

$$\mathscr{T}_{\omega,\tau} = \mathscr{M}_{\omega,\tau}^{-1}\mathscr{N}_{\omega,\tau}.$$

Remark 2.2. The GSOR algorithm uses a relaxation parameter matrix Ω for the SOR-like method instead of a single relaxation parameter by introducing the following matrix

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

with ω and τ are two nonzero reals as well as $I_m \in \mathbb{R}^{m\times m}$ and $I_n \in \mathbb{R}^{n\times n}$ are the m-by-m and n-by-n identity matrices, respectively. Obviously, when $\omega = \tau$, the GSOR method reduces to SOR-like method. As we known, this GSOR has the more burden of choosing two good, or optimal relaxation parameters in order to achieve a comparable rate of convergence than the SOR-like method. Even though it has faster asymptotic convergence rate than the SOR-like method.

In [8], the authors proved the convergence of the GSOR method under suitable restrictions on the iteration parameters, and determined its optimal iteration parameters as well as the corresponding optimal convergence factor. These results are summarized as the following theorem.

Let

$$\omega_- = \frac{4\tau\mu_{\min}}{(1+\tau\mu_{\min})^2}, \qquad \omega_+ = \frac{4\tau\mu_{\max}}{(1+\tau\mu_{\max})^2}, \qquad \text{and} \qquad \omega_0 = \frac{4}{\tau(\mu_{\max}+\mu_{\min})+2}.$$

Theorem 2.2 ([8, Theorems 2.1 and 4.1]). *Let $A \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ be symmetric positive definite, and $B \in \mathbb{R}^{n\times m}$ be of full column rank. Denote the smallest and the largest eigenvalues of the matrix $\mathscr{H} = Q^{-1}B^TA^{-1}B$ by $\mu_{\min}$ and $\mu_{\max}$, respectively. Then,*
(i) the GSOR method is convergent, if ω satisfies $0 < \omega < 2$ and τ satisfies the following condition:

$$0 < \tau < \frac{2(2-\omega)}{\omega\mu_{\max}};$$

(ii) when $\tau \le \frac{1}{\sqrt{\mu_{\min}\mu_{\max}}}$, it holds that

$$\rho(\mathscr{T}_{\omega,\tau}) = \begin{cases} \sqrt{1-\omega}, \text{ for } 0 < \omega < \omega_-(\tau), \\ \frac{1}{2}\big(2-\omega-\tau\omega\mu_{\min}+\sqrt{(2-\omega-\tau\omega\mu_{\min})^2-4(1-\omega)}\big), \text{ for } \omega_-(\tau) \le \omega \le \omega_0(\tau), \\ \frac{1}{2}\big(\tau\omega\mu_{\max}+\omega-2+\sqrt{(\tau\omega\mu_{\max}+\omega-2)^2-4(1-\omega)}\big), \text{ for } \omega_0(\tau) < \omega < 2; \end{cases}$$

(iii) when $\frac{1}{\sqrt{\mu_{\min}\mu_{\max}}} < \tau < \frac{2(2-\omega)}{\omega\mu_{\max}}$, it holds that

$$\rho(\mathscr{T}_{\omega,\tau}) = \begin{cases} \sqrt{1-\omega}, \text{ for } 0 < \omega < \omega_+(\tau), \\ \frac{1}{2}\big(\tau\omega\mu_{\max}+\omega-2+\sqrt{(\tau\omega\mu_{\max}+\omega-2)^2-4(1-\omega)}\big), \text{ for } \omega_+(\tau) \le \omega < 2. \end{cases}$$

Moreover, the optimal iteration parameters $\omega_\star$ and $\tau_\star$ are given by

$$\omega_\star = \frac{4\sqrt{\mu_{\min}\mu_{\max}}}{(\sqrt{\mu_{\max}}+\sqrt{\mu_{\min}})^2} \qquad \text{and} \qquad \tau_\star = \frac{1}{\sqrt{\mu_{\min}\mu_{\max}}},$$

and the corresponding optimal convergence factor of the GSOR method is

$$\rho(\mathscr{T}_{\omega,\tau}) = \frac{\sqrt{\mu_{\max}}-\sqrt{\mu_{\min}}}{\sqrt{\mu_{\max}}+\sqrt{\mu_{\min}}}.$$

Remark 2.3. In fact, the formulas about the optimal parameters and the iteration parameters in Theorems 2.1-2.2 are of only theoretical meanings, and they are far away from practical applications. Because the spectral radius and the extremal eigenvalues of the matrix $\mathscr{H} = Q^{-1}B^TA^{-1}B$ are required and which may greatly decrease the computing efficiency of the SOR-like or GSOR methods. In addition, the computation of these is usually a difficult task to complete.

The following SSOR-like method is obtained by combining the SOR-like scheme and its backward one, introduced by Golub, Wu and Yuan [20], Darvishi and Hessari [15] as well as Zheng, Wang and Wu [33].

SSOR-like Method ([15, 20, 33]). Let $Q \in \mathbb{R}^{m\times m}$ be a nonsingular and symmetric matrix. Given two initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^m$, and a relaxation factor $\omega > 0$. For $k = 0, 1, 2, \cdots$ until the iteration sequence $\{({x^{(k)}}^T, {y^{(k)}}^T)^T\}$ converges, compute

$$\begin{cases} y^{(k+1)} = y^{(k)} + \omega(2-\omega)Q^{-1}B^T\left\{x^{(k)} - \frac{\omega}{1-\omega}A^{-1}By^{(k)} + \frac{\omega}{1-\omega}A^{-1}b\right\} - \frac{\omega(2-\omega)}{1-\omega}Q^{-1}q, \\ x^{(k+1)} = (1-\omega)^2x^{(k)} - \omega A^{-1}B\left\{y^{(k+1)} + (1-\omega)y^{(k)}\right\} + \omega(2-\omega)A^{-1}b. \end{cases}$$

Also, it involved a parameter ω. It is an alternative iterative method.

In addition, based on the SOR-like and GSOR methods, resulted in some trivial modifications or generalizations from the parameters or alternation views. For example, the GSSOR method (see [31]) is, it was proposed by involving two relaxation parameters ω and τ or say, a parameter matrix.

GSSOR-like Method ([31]) Let $Q \in \mathbb{R}^{m\times m}$ be a nonsingular and symmetric matrix. Given two initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^m$, and two relaxation factors $\omega, \tau > 0$. For $k = 0, 1, 2, \cdots$ until the iteration sequence $\{({x^{(k)}}^T, {y^{(k)}}^T)^T\}$ converges, compute

$$\begin{cases} y^{(k+1)} = y^{(k)} + \frac{\tau(2-\tau)}{1-\tau}Q^{-1}B^T\left\{(1-\omega)x^{(k)} - \omega A^{-1}By^{(k)} + \omega A^{-1}b\right\} - \frac{\tau(2-\tau)}{1-\tau}Q^{-1}q, \\ x^{(k+1)} = (1-\omega)^2x^{(k)} - \omega A^{-1}B\left\{y^{(k+1)} + (1-\omega)y^{(k)}\right\} + \omega(2-\omega)A^{-1}b. \end{cases}$$

However, we have noticed that the parameters ω, τ are fixed throughout the iteration process in all discussed above, meanwhile their decision in advance all involve in the determination of the spectral radius and the extremal eigenvalues of the matrix $\mathscr{H} = Q^{-1}B^TA^{-1}B$, which reduces the effectiveness of the methods from one point of view.

3. New Methods

In this section, we propose a class of new SOR-like methods. Moreover, theoretic convergence analysis to support the scheme will be presented.

Recall our discussion in the third paragraph of Section 1. Our intuitive idea is to update ω at each iterate step by an optimization model. To implement this idea, we can compute the relaxation parameter $\omega^{(k)}$ by the optimization technique.

Method 3.1. (SOR-Like Methods with Optimization Model) Let $Q \in \mathbb{R}^{m\times m}$ be a nonsingular and symmetric matrix. Given an initial point $((x^{(0)})^T,(y^{(0)})^T)^T \in \mathbb{R}^{n+m}$, and a precision $\epsilon > 0$. For $k = 0,1,2,\cdots$ until the iteration sequence $\{(x^{(k)^T},y^{(k)^T})^T\}$ converges, compute by the following scheme

$$\begin{cases} x^{(k+1)} = \left(1-\omega^{(k+1)}\right)x^{(k)} + \omega^{(k+1)}A^{-1}\left(b-By^{(k)}\right), \\ y^{(k+1)} = y^{(k)} + \omega^{(k+1)}Q^{-1}\left(B^T x^{(k+1)} - q\right), \end{cases}$$

where $\omega^{(k+1)}$ at each step can be obtained by the following optimal model

$$\min_{\omega} \|\mathscr{W}^{-1}r^{(k+1)}\|_2^2 \tag{3.1}$$

with

$$\mathscr{W} = \begin{pmatrix} A & O \\ -B^T & Q \end{pmatrix}, \qquad r^{(k)} = \begin{pmatrix} A & B \\ -B^T & O \end{pmatrix}\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} - \begin{pmatrix} b \\ -q \end{pmatrix}.$$

Remark 3.1. The way of computing

$$\begin{aligned} \mathscr{W}^{-1}r^{(k+1)} &= \mathscr{W}^{-1}\left[\begin{pmatrix} A & B \\ -B^T & O \end{pmatrix}\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} - \begin{pmatrix} b \\ -q \end{pmatrix}\right] \\ &= \begin{pmatrix} x^{(k+1)} + A^{-1}By^{(k+1)} \\ Q^{-1}B^TA^{-1}By^{(k+1)} \end{pmatrix} - \begin{pmatrix} A^{-1}b \\ Q^{-1}B^TA^{-1}b - Q^{-1}q \end{pmatrix}, \end{aligned}$$

thus, only two matrix-vector products $A^{-1}By^{(k+1)}$ and $Q^{-1}B^TA^{-1}By^{(k+1)}$ must to be compute at each step. Similarly to the SOR-Like method, Q is an approximation (preconditioner) of the Schur complement matrix $\mathscr{S} = B^TA^{-1}B$. For example, we may reasonably take $Q = \alpha I$ with $\alpha > 0$, νB^TB with $\nu > 0, B^TA^{-1}B, B^T\hat{A}^{-1}B$ with $\hat{A} = \text{tridiag}(A)$, $\text{tridiag}(B^T\hat{A}^{-1}B)$ with $\hat{A} = \text{tridiag}(A)$ and so on.

Remark 3.2. In fact, the formula of the exact solution of the optimization model (3.1) can be given by simple computation. The experimentally found optimal point ω are obtained according to the Nelder-Mead simplex method ([24]) in practical computing, and let $\omega_0 = 1$ be a starting value. The optimization models may be solved 2-3 steps approximately for saving cost.

Remark 3.3. The proceeding of solving the optimization model $\min_\omega \|\mathscr{W}^{-1}r^{(k+1)}\|_2^2$ is the process of obtaining ω which is the relaxation factor of the Method 3.1. In the SOR-like methods, the relaxation parameter ω is a fixed and positive number throughout the iteration process, and further it is clear that computing the optimal value of ω requests the expensive cost by Theorem 2.1. What's more, in the premise of minimizing the norm of the residual, $\omega^{(k)}$ is chosen dynamically using optimization model in the Method 3.1. And we need find the optimal parameter $\omega^{(k)}$ before each iteration step. There is no doubt that this process will increase the complexity of operations. Nevertheless, the optimal parameter $\omega^{(k)}$ is selected once at each five iteration steps, according to the results of several numerical experiments. The above approach not only ensure the optimality of parameters, but

also to ensure that the computations will not be too large. Two birds with one stone. The processing of selecting $\omega^{(k)}$ is introduced in Section 4 for more details.

Now, the convergence theory is presented for Method 3.1.

As the description in last Section, the Method 3.1 also can be regarded as the iteration based on the following splitting with a variable parameter:

$$\mathscr{A} \equiv \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} = \mathscr{M}_\omega - \mathscr{N}_\omega,$$

where

$$\mathscr{M}_\omega = \frac{1}{\omega}(\mathscr{D} - \omega\mathscr{L}) = \begin{pmatrix} \frac{1}{\omega}A & O \\ -B^T & \frac{1}{\omega}Q \end{pmatrix}, \qquad \mathscr{N}_\omega = \mathscr{M}_\omega - \mathscr{A} = \begin{pmatrix} (\frac{1}{\omega}-1)A & -B \\ O & \frac{1}{\omega}Q \end{pmatrix}.$$

When $\omega = 1$, the special case can be list as follows:

$$\mathscr{M}_1 = \begin{pmatrix} A & O \\ B^T & Q \end{pmatrix}, \qquad \mathscr{N}_1 = \begin{pmatrix} O & -B \\ O & Q \end{pmatrix}.$$

Theorem 3.1. *Consider the augmented linear systems* (1.2). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $B \in \mathbb{R}^{n\times m}$ have full column rank. Assume that u_* is the unique solution of the linear systems* (1.2). *Then the sequence $\{u^{(k)}\} = \{(x^{(k)^T}, y^{(k)^T})^T\}$ generated by Method* 3.1 *converges to u_* if the following condition is provided:*

$$\left\| B^T A^{-1} A^{-1} B + (I - B^T A^{-1} B Q^{-1})(I - Q^{-1} B^T A^{-1} B) \right\|_2 < 1. \tag{3.2}$$

Furthermore, if $Q = B^T A^{-1} B$, (3.2) *can be reduced*

$$\|A^{-1}B\|_2 < 1;$$

if $Q = \alpha I$, $\alpha > 0$, (3.2) *can be became*

$$\left\| B^T A^{-1} A^{-1} B + (I - \alpha B^T A^{-1} B)^2 \right\|_2 < 1.$$

Proof. We known that $r^{(k+1)} = \mathscr{N}_\omega \mathscr{M}_\omega^{-1} r^{(k)}$ by simple computation. And then,

$$\begin{aligned}
\|\mathscr{W}^{-1} r^{(k+1)}\|_2 =& \|\mathscr{W}^{-1} \mathscr{N}_\omega \mathscr{M}_\omega^{-1} r^{(k)}\|_2 \\
=& \|\mathscr{W}^{-1} \mathscr{N}_\omega \mathscr{M}_\omega^{-1} \mathscr{W} \mathscr{W}^{-1} r^{(k)}\|_2 \\
\le& \left\| \mathscr{W}^{-1} \mathscr{N}_1 \cdot \mathscr{W}^{-1} r^{(k)} \right\|_2 \\
\le& \|\mathscr{W}^{-1} \mathscr{N}_1\|_2 \cdot \|\mathscr{W}^{-1} r^{(k)}\|_2 \\
=& \left\| \begin{pmatrix} A^{-1} & O \\ Q^{-1} B^T A^{-1} & Q^{-1} \end{pmatrix} \begin{pmatrix} O & -B \\ O & Q \end{pmatrix} \right\|_2 \cdot \|\mathscr{W}^{-1} r^{(k)}\|_2 \\
=& \left\| \begin{pmatrix} O & -A^{-1}B \\ O & I - Q^{-1} B^T A^{-1} B \end{pmatrix} \right\|_2 \cdot \|\mathscr{W}^{-1} r^{(k)}\|_2 \\
=& \sqrt{\|B^T A^{-1} A^{-1} B + (I - B^T A^{-1} B Q^{-1})(I - Q^{-1} B^T A^{-1} B)\|_2} \cdot \|\mathscr{W}^{-1} r^{(k)}\|_2.
\end{aligned}$$

This theorem is proved from (3.2).

By simple substitution furthermore, when $Q = B^T A^{-1} B$, (3.2) can be briefly reduced the following form

$$\|A^{-1}B\|_2 < 1;$$

when $Q = \alpha I$, $\alpha > 0$, (3.2) can be became

$$\|B^T A^{-1} A^{-1} B + (I - \alpha B^T A^{-1} B)^2\|_2 < 1.$$

The theorem has be proved. □

4. Numerical Experiments

In this section, we provide numerical results to illustrate the effectiveness of the Method 3.1 in terms of the number of iterations (denoted by **IT**), the elapsed computing times in seconds (denoted by **CPU**) and the 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 as

$$\text{RES:} = \sqrt{\|b - Ax^{(k)} - By^{(k)}\|^2 + \|q - B^T x^{(k)}\|^2}$$

and

$$\text{ERR:} = \frac{\sqrt{\|x^{(k)} - x_*\|^2 + \|y^{(k)} - y_*\|^2}}{\sqrt{\|x^{(0)} - x_*\|^2 + \|y^{(0)} - y_*\|^2}},$$

respectively.

For the sake of convenience, the SOR-Like method with optimization model, say the Method 3.1, are termed briefly as SOR_{opt}. In our computations, all runs with respect to the SOR-like, the SSOR-like and the SOR_{opt} iteration methods are started from the zero vector $((x^{(0)})^T, (y^{(0)})^T)^T = 0$, and terminated if the current iteration satisfy ERR$< 10^{-9}$. In addition, the numerical experiments are performed in MATLAB (version R2013a) on PC in double precision, which is 2.40GHz central processing unit [Intel(R) Core(TM)i7-4500 CPU] with 8G memory and Microsoft Windows 8 operating system. Moreover, the ω_k in SOR_{opt} methods is updated for every five iteration steps.

The details of choosing the parameter $\omega^{(k)}$, $k = 0, 1, 2, \cdots$, in SOR_{opt} methods are described as follows.

First of all, an initial value $\omega^{(0)}$ should be chosen. In general, let $\omega^{(0)} = 1$. For the SOR-like Method, the optimal parameters $\omega_\star$ are between 0 and 2 based on the Theorem 2.1. And then in order to making the $\omega^{(k)}$ close to the optimal value quickly in the processing of iteration, the authors had better select 1 for the initial value of $\omega^{(k)}$. Then after iterating five steps, the relaxation parameter $\omega^{(5)}$ which is minimizing the norm of $\mathscr{W}^{-1} r^{(5)}$ is selected again. The new one is the parameter from the sixth step to the tenth step, and the next new parameter is $\omega^{(10)}$ which is minimizing the norm of $\mathscr{W}^{-1} r^{(10)}$, and so on, until the algorithm converges. In throughtout iterate process, every dynamic parameter is selected

Table 1: Choices of the matrix Q for Example 4.1.

Case No.	Matrix Q	Description
I	tridiag$(B^TA^{-1}B)$	
II	tridiag$(B^T\hat{A}^{-1}B)$	$\hat{A}$=tridiag(A)

at each five steps so that the overall iteration steps become more less than others. Therefore, the Method 3.1 is more efficient.

Example 4.1 ([7]). Consider the saddle-point problems (1.1), in which

$$A=\begin{pmatrix} I\bigotimes T+T\bigotimes I & 0 \\ 0 & I\bigotimes T+T\bigotimes I \end{pmatrix}_{2p^2\times 2p^2}, \qquad B=\begin{pmatrix} I\bigotimes F \\ F\bigotimes I \end{pmatrix}_{2p^2\times p^2}$$

and

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

with $\bigotimes$ being the Kronecker product symbol, $h=\frac{1}{p+1}$ the discretization meshsize and $S=\text{tridiag}(a,b,c)$ is a tridiagonal matrix with $S_{i,i}=b$, $S_{i-1,i}=a$ and $S_{i,i+1}=c$ for appropriate i.

For this experiment, we set $n=2p^2$ and $m=p^2$. Hence, the total number of variables is $n+m=3p^2$. We choose the matrix Q, as an approximation to the matrix $B^TA^{-1}B$, according to the cases listed in Table 1.

In Table 2, we list the iteration numbers, the computing times and the RES values (if the convergence criterion is not achieved within $k_{\max}$ iteration steps) of the SOR-like, the SSOR-like and the SOR$_{\text{opt}}$ iterative methods for the Example 4.1 with respect to different choices of the problem sizes.

From the Table 2, it can be seen that for different problem sizes the numbers of iteration steps of the SOR$_{\text{opt}}$ methods are less than those of the SOR-like and the SSOR-like methods at almost the same CPU times. Clearly, the SOR$_{\text{opt}}$ method outperforms both the SOR-like and the SSOR-like methods.

Example 4.2. Consider the saddle-point problems (1.1), in which

$$A=(a_{ij})_{n\times n}=\begin{cases} i+1, & i=j, \\ 1, & |i-j|=1, \\ 0, & \text{otherwise}, \end{cases} \qquad B=(b_{ij})_{n\times m}=\begin{cases} j, & i=j+n-m, \\ 0, & \text{otherwise}. \end{cases}$$

For this experiment, we set $m=2n$. Hence, the total number of variables is $m+n=3n$. We choose the matrix Q, as an approximation to the matrix $B^TA^{-1}B$, according to the cases listed in Table 3.

In Table 4, we give the iteration numbers, the computing times and the RES values (if the convergence criterion is not achieved within $k_{\max}$ iteration steps) of the SOR-like, the SSOR-like and the SOR$_{\text{opt}}$ iterative methods for the Example 4.2 with respect to different

Table 2: IT, CPU, and RES of the SOR-like, the SSOR-like and the SOR_{opt} methods for Example 4.1.

n		128	512	1152	2048	4608	8192	12800
m		64	256	576	1024	2304	4096	6400
$n+m$		192	768	1728	3072	6912	12288	19200
Case I								
SOR-like	$\omega_\star$	1.0585	1.0519	1.0476	1.0451	1.0460	1.0408	1.0506
	IT	113	209	301	391	568	743	916
	CPU	0.02	0.11	0.33	0.82	2.84	7.78	20.20
	RES	8.37e-9	1.36e-8	1.71e-8	2.03e-8	2.58e-8	3.04e-8	3.49e-8
SSOR-like	ω	0.4990	0.5444	0.5321	0.5300	0.5316	0.5200	0.5068
	IT	61	130	193	254	375	494	612
	CPU	0.02	0.15	0.41	1.03	3.95	11.11	26.22
	RES	1.54e-8	1.62e-8	2.02e-8	2.46e-8	2.97e-8	3.49e-8	3.94e-8
SOR_{opt}	IT	42	90	113	122	165	235	348
	CPU	0.02	0.09	0.29	0.61	1.22	5.92	9.41
	RES	7.31e-8	2.78e-7	9.28e-8	9.92e-8	4.06e-6	2.26e-7	2.85e-7
Case II								
SOR-like	IT	114	220	322	421	617	811	1002
	CPU	0.02	0.12	0.35	0.85	3.13	8.30	21.62
	RES	7.40e-9	1.13e-8	1.39e-8	1.69e-8	2.13e-8	2.47e-8	2.85e-8
SSOR-like	IT	144	141	143	177	264	349	433
	CPU	0.04	0.14	0.31	0.68	3.03	8.08	18.43
	RES	1.31e-7	5.14e-7	1.05e-6	2.86e-8	2.91e-8	3.44e-8	3.96e-8
SOR_{opt}	IT	54	77	132	161	188	278	312
	CPU	0.04	0.11	0.34	0.68	2.65	6.61	14.98
	RES	1.11e-7	2.23e-7	1.19e-6	1.51e-7	2.41e-6	1.39e-6	3.71e-7

Table 3: Choices of the matrix Q for Example 4.2.

Case No.	Matrix Q	Description
I	$\text{tridiag}(B^T\tilde{A}^{-1}B)$	$\tilde{A}=\text{diag}(A)$
II	$\text{tridiag}(B^T\hat{A}^{-1}B)$	$\hat{A}$=tridiag(A)

choices of the problem sizes. It is clear that both the SOR-like, the iteration steps of the SOR_{opt} methods are much less than those of the SOR-like and the SSOR-like methods at almost the same computing time for the Example 4.2 with respect to the different problem sizes.

Table 4: IT, CPU, and RES of the SOR-like, the SSOR-like and the SOR_{opt} methods for Example 4.2.

n		128	512	1152	2048	4608	8192
m		64	256	576	1024	2304	4096
$n+m$		192	768	1728	3072	6912	12288
Case I							
SOR-like	$\omega_\star$	0.5958	0.3657	0.2619	0.2037	0.1451	0.1000
	IT	64	131	205	268	339	429
	CPU	0.0109	0.0149	0.0324	0.0673	0.0931	0.3702
	RES	2.70e-6	7.83e-6	8.68e-6	3.84e-5	1.25e-5	6.21e-5
SSOR-like	ω	0.4990	0.3000	0.2321	0.1530	0.1316	0.1200
	IT	29	43	50	73	88	132
	CPU	0.0215	0.2880	1.2721	5.8606	30.1528	175.7147
	RES	4.47e-7	2.57e-6	4.83e-6	4.75e-5	7.11e-4	2.56e-4
SOR_{opt}	IT	12	25	37	63	71	110
	CPU	0.0074	0.0102	0.0199	0.0441	0.2209	0.1989
	RES	6.00e-6	5.79e-6	2.97e-5	4.43e-5	6.39e-5	2.91e-4
Case II							
SOR-like	$\omega_\star$	0.4664	0.2720	0.1915	0.1476	0.1251	0.1000
	IT	43	92	131	187	268	426
	CPU	0.0194	0.3749	1.4338	3.5292	10.4478	26.4989
	RES	4.46e-6	7.39e-6	6.06e-5	2.71e-5	1.99e-5	6.45e-5
SSOR-like	ω	0.5100	0.3900	0.3321	0.2530	0.2016	0.1500
	IT	30	58	81	122	142	189
	CPU	0.2668	1.6957	7.9386	28.2987	103.7132	496.3327
	RES	5.25e-7	8.30e-6	1.82e-5	1.76e-4	6.05e-4	1.30e-3
SOR_{opt}	IT	13	23	25	37	71	74
	CPU	0.0136	0.1998	0.5608	1.6763	5.2310	10.9156
	RES	7.09e-6	7.71e-6	2.20e-5	7.12e-5	4.76e-5	9.17e-5

5. Conclusion

In this study, we modified the SOR-like method from the view point of the relaxation parameter and propose a new SOR-like scheme with optimization model for solving large sparse augmented linear systems. It is noted that the proposed Methods 3.1 (SOR_{opt}) need to compute the updated optimal relaxation parameter, while the SOR-like and SSOR-like methods do not require this additional computation. Even so, the Methods 3.1 (SOR_{opt} methods) are very effective for the large sparse augmented linear systems.

The numerical experiments further show that the Methods 3.1 (SOR_{opt} methods) are superior to the SOR-like and SSOR-like methods.

Acknowledgments

The authors are very much indebted to the anonymous referees for their helpful comments and suggestions which greatly improved the original manuscript of this paper. This work is supported by the National Nature Science Foundation of China (No. 11371275) and the National Nature Science Foundation of the Shanxi Province, China (No. 201601D0 11004).

References

[1] K. ARROW, L. HURWICZ, AND H. UZAWA, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, 1958.
[2] Z. Z. BAI, *Optimal parameters in the HSS-like methods for saddle-point problems*, Numer. Linear Algebra Appl., 16(2009), pp. 447–479.
[3] Z. Z. BAI, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comput., 75 (2006), pp. 791–815.
[4] Z. Z. BAI, AND X. B. CHI, *Asymptotically optimal successive overrelaxation methods for systems of linear equations*, J. Comput. Math., 21(5) (2003), pp. 603–612.
[5] Z. Z. BAI, AND G. H. GOLUB, *Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems*, IMA J. Numer. Anal., 27(1) (2007), pp. 1–23.
[6] Z. Z. BAI, G. H. GOLUB, AND M. K. NG, *On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations*, Numer. Linear Algebra Appl., 14 (2007), pp. 319–335.
[7] Z. Z. BAI, G. H. GOLUB, AND J. Y. PAN, *Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems*, Numer. Math., 98 (2004), pp. 1–32.
[8] Z. Z. BAI, B. N. PARLETT, AND Z. Q. WANG, *On generalized successive overrelaxatioon methods for augmented linear systems*, Numer. Math., 102 (2005), pp. 1–38.
[9] M. BENZI, AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
[10] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.
[11] A. BJÖRCK, *Numerical stablity of methods for solving augmented systems. in Proceedings from Recent Developments in Optimization Theory and Nonlinear Analysis*, Jerusalem, 1995, Y. Censor and S. Reich, eds., Contemp. Math., 204, Amer. Math. Soc., Providence, RI, (1997), pp. 51–60.
[12] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal. 34 (1997), pp. 1072–1092.
[13] Z. CHAO, AND G. L. CHEN, *Semi-convergence analysis of the Uzawa-SOR methods for singular saddle point problems*, Appl. Math. Letter, 35 (2014), pp. 52–57.
[14] Z. CHAO, N. M. ZHANG, AND Y. Z. LU, *Optimal parameters of the generalized symmetric SOR method for augmented system*, J. Comput. Appl. Math., 266 (2014), pp. 52–60.
[15] M. T. DARVISHI, AND P. HESSARI, *Symmetric SOR method for augumented systems*, Appl. Math. Comput., 173(1) (2006), pp. 404–420.
[16] H. C. ELMAN, AVD G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.
[17] H. ELMAN, AND D. SILVESTER, *Fast nonsymmetric iteration and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.

[18] FISCHER, A. RAMAGE, D. J. SILVESTER, AND A. J. WATHEN, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.
[19] M. FORTIN, AND R. GLOWINSKI, *Augmented Lagrangian Methods, Applications to the Numerical Solution of Boundary Value Problems*, Amsterdam: North-Holland, 1983.
[20] G. H. GOLUB, X. WU, AND J. Y. YUAN, *SOR-like methods for augumented systems*, BIT, 41 (2001), pp. 71–85.
[21] P. GUO, C. X. LI, AND S. L. WU. *A modified SOR-like method for the augmented systems*, J. Comput. Appl. Math., 274 (2015), pp. 58–69.
[22] Z. LI, C.J. LI, AND D. J. EVANS, *Chebyshev acceleration for SOR-like method*, Inter. J. Comput. Math., 82(5) (2005), pp. 583–593.
[23] Z. Z. LIANG, AND G. F. ZHANG, *Modified unsymmetric SOR method for augmented systems*, Appl. Math. Comput., 234 (2014), pp. 584–598.
[24] J. A. NELDER, AND R. MEAD, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.
[25] X. H. SHAO, Z. LI, AND C. J. LI, *Modified SOR-like methods for the augmented systems*, Inter. J. Comput. Math., 84 (2007), pp. 1653–1662.
[26] J. G. SUN, *Structured backward errors for KKT systems*, Linear Algebra Appl., 288 (1999), pp. 75–88.
[27] R. P. WEN, C. L. WANG, AND X. H. YAN, *Generalization of the nonstationary multisplitting iterative method for symmetric positive definite linear systems*, Appl. Math. Comput., 216 (2010), pp. 1707–1714.
[28] R. P. WEN, G. Y. MENG, AND C. L. WANG, *Quasi-Chebyshev accelerated iteration methods based on optimization for linear systems*, Comput. Math. Appl., 66 (2013), pp. 934–942.
[29] S. WRIGHT, *Stablity of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997) 191–222.
[30] J. Y. YUAN, *Numerical methods for generalized least squares problems*, J. Comput. Appl. Math., 66 (1996), pp. 571–584.
[31] G. F. ZHANG, AND Q. H. LU, *On generalized symmetric SOR method for augumented systems*, J. Comput. Appl. Math., 219 (2008), pp. 51–58.
[32] J. J. ZHANG, AND J. J. SHANG, *A class of Uzawa-SOR methods for saddle point problem*, Appl. Math. Comput., 216 (2010), pp. 2163–2168.
[33] B. ZHENG, K. WANG, AND Y. J. WU, *SSOR-like methods for saddle point problem*, Inter. J. Comput. Math., 86 (2009), pp. 1405–1423.
[34] Q. Q. ZHENG, AND C. F. MA, *A new SOR-Like method for the saddle point problems*, Appl. Math. Comput., 233 (2014), pp. 421–429.
[35] Q. Q. ZHENG, AND C. F. MA, *A class of triangular splitting methods for saddle point problems*, J. Comput. Appl. Math., 298 (2016), pp. 13–23.