Open Mathematics **Open Access**

Research Article

Jakub Kierzkowski*

New SOR-like methods for solving the Sylvester equation

Abstract: We present new iterative methods for solving the Sylvester equation belonging to the class of SOR-like methods, based on the SOR (Successive Over-Relaxation) method for solving linear systems. We discuss convergence characteristics of the methods. Numerical experimentation results are included, illustrating the theoretical results and some other noteworthy properties of the Methods.

Keywords: Sylvester equation, SOR-like iterative method, Iterative methods

MSC: 65F10, 65F50, 15A24

DOI 10.1515/math-2015-0017
Received December 7, 2013; accepted December 6, 2014.

1 Introduction

The Sylvester equation ($AX - XB = C$) has many applications, for example in control theory and when solving partial differential equations numerically (see e.g. [3, pp. 245-246], [8, 9]). Two direct methods, the Bartels-Stewart algorithm ([1]) and the Hessenberg-Schur algorithm ([4]), are widely known and used (the former is implemented within Matlab and Octave).

Solving a large-scale Sylvester equation begets an iterative approach. The simplest, though not necessarily the best, approach is to convert the Sylvester equation to a linear system (see Theorem 1.5) and then solve it using an iterative method for linear systems. Another way is to use Krylov-subspace methods, which are available in the literature (see [5]). In [9] Starke and Niethammer proposed another routine: the Successive Over-Relaxation method for solving linear systems applied to the Sylvester equation. This however results in a block-iterative method not iterative one, as it requires solving a number of linear systems per iteration.

The purpose of this paper is to present the SOR-like methods and highlights some of their known properties. In Section 2 we give the SOR-like method as proposed by Woźnicki in [10], and propose two similar methods based upon it. All three are stationary iterative methods for solving $AX - XB = C$. In Section 3 we form two sufficient conditions under which one of those methods will converge. Section 4 contains some computed examples. In Section 5 we summarize the results.

Definition 1.1. *Let $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times n}$ be given matrices, and $X \in \mathbb{R}^{m\times n}$ be an unknown matrix. The Sylvester equation then is the equation of the form*

$$AX - XB = C. \tag{1}$$

For the problem (1) to have a unique solution matrices A and B must have no common eigenvalues (see [3, p. 247]).

The Sylvester equation appears in the block diagonalization of a block triangular matrix. The following theorem by Roth (see [7]) shows that connection.

***Corresponding Author: Jakub Kierzkowski:** Faculty of Mathematics and Computer Science, Warsaw University of Technology, Koszykowa 75, 00-662, Warsaw, Poland, E-mail: J.Kierzkowski@mini.pw.edu.pl

Theorem 1.2 (William E. Roth, 1952, [7]). *Let $A, B, C \in K^{r\times r}$. The necessary and sufficient condition that the equation $AX - XB = C$ has a solution $X \in K^{r\times r}$, is that the matrices*

$$\begin{bmatrix} A & C \\ 0 & B \end{bmatrix} \text{ and } \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$$

be similar.

Then it is

$$\begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} A & C \\ 0 & B \end{bmatrix} \cdot \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & -(AX - XB - C) \\ 0 & B \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}.$$

Definition 1.3. *Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$. The block matrix*

$$C = \left[\begin{array}{c|c|c} a_{11}B & a_{12}B & \dots\ a_{1n}B \\ \hline a_{21}B & a_{22}B & \dots\ a_{2n}B \\ \hline \vdots & \vdots & \ddots\ \vdots \\ \hline a_{m1}B & a_{m2}B & \dots\ a_{mn}B \end{array}\right] \in \mathbb{R}^{mp\times nq}$$

is called the Kronecker product of matrices A and B, and is denoted as $A \otimes B$.

Definition 1.4. *Let $X = \left[x^1, x^2, \dots, x^n\right]$ be a n-times-n matrix. Then we denote the vector $\left[(x^1)^T, (x^2)^T, \dots, (x^n)^T\right]^T \in \mathbb{R}^{n^2}$ as $vec\ (X)$ and call it the vectorization of X.*

The Kronecker product is important in research on the Sylvester equation; it can be used to transform equation (1) into a linear system of equations (theorem 1.5).

Theorem 1.5 one can find in [6].

Theorem 1.5 (equivalence between the Sylvester equation and a particular linear system). *Let A, B, C, X be as in Definition 1.1. The problem of finding the solution of the equation $AX - XB = C$ is equivalent to finding the solution of the linear system*

$$R \cdot vec\ X = vec\ C, \tag{2}$$

where $R = \left(I_n \otimes A - B^T \otimes I_m\right)$ and $R \in \mathbb{R}^{m\cdot n\times m\cdot n}$.

Remark 1.6. *It is easy to see that, if X is a solution of $AX - XB = C$, it is also a solution of $(A-\alpha I_m)X - X(B-\alpha I_n) = C$. In addition, the matrix R in (2) is the same for all α, as $\left(I_n \otimes (A - \alpha I_m) - (B - \alpha I_n)^T \otimes I_m\right) = \left(I_n \otimes A - B^T \otimes I_m\right) - I_m \otimes (\alpha I_m) + (\alpha I_n) \otimes I_m = \left(I_n \otimes A - B^T \otimes I_m\right)$. This property will be used in Section 4.*

Derivation of formulas of the methods for solving the Sylvester equation that are described in Section 2, is based on the derivation of the formula of the SOR (Successive Over-Relaxation) method for linear systems (for $\omega = 1$ that is the Gauss-Seidel method). Matrix A is cut into parts $A = D - L - U$, where L and U are lower and upper triangular, with zeros on the main diagonals, respectively, and D is nonsingular diagonal:

$$x^{(t)} = (D - \omega L)^{-1} \left((1 - \omega)\, D + \omega U\right) x^{(t-1)} + (D - \omega L)^{-1} \omega c, \tag{3}$$

where $\{x^{(t)}\}$ is a sequence of approximations of the solution.

2 SOR-like iterative methods for solving the equation

The original SOR-like method was devised by Prof. Zbigniew Woźnicki (see [10]), and is based upon the SOR method for solving linear systems, with the matrix A being split in the same way. The derivation of the formula is

similar, too, and is performed in the following way:

$$AX - XB = C$$

$$D^{-1} \cdot \backslash\backslash\, DX = LX + UX + XB + C$$

$$\omega \cdot \backslash\backslash\, X = D^{-1}(LX + UX + XB + C)$$

$$\omega X = \omega D^{-1} LX + \omega D^{-1} UX + \omega D^{-1} XB + \omega D^{-1} C$$

$$\omega X - \omega D^{-1} LX = \omega D^{-1} UX + \omega D^{-1} XB + \omega D^{-1} C$$

$$X - \omega D^{-1} LX = (1-\omega)X + \omega D^{-1} UX + \omega D^{-1} XB + \omega D^{-1} C$$

$$(I - \omega D^{-1} L)X = \left((1-\omega)I + \omega D^{-1} U\right) X + \omega D^{-1}(XB + C)$$

$$X = (I - \omega D^{-1} L)^{-1} \left(((1-\omega)I + \omega D^{-1} U)X + \omega D^{-1}(XB + C)\right).$$

We obtain the iterative formula:

Algorithm 2.1.

$$X^{(t)} = \left(I - \omega D^{-1} L\right)^{-1} \left(\left((1-\omega)\, I + \omega D^{-1} U\right) X^{(t-1)} + \omega D^{-1} \left(X^{(t-1,t)} B + C\right)\right) \tag{4}$$

The most characteristic idea in this method is the usage of the double index $(t-1, t)$. This form makes the method impossible to implement using the matrix product: the entries of the matrix X are computed column-wise, one after another. $X^{(t-1,t)}$ means that we take the entries of the latest approximation $X^{(t)}$ if they are already computed in the t-th iteration, and entries of the previous approximation $X^{(t-1)}$ for all the rest of the entries. This is the approach which led to the development of the Gauss-Seidel (and SOR) method for linear systems. The absence of the matrix-formed formula of the SOR-like method makes this method difficult to analyze. To avoid such difficulties, we can modify the method's formula by replacing the double index $(t-1, t)$ with the simple $(t-1)$:

Algorithm 2.2.

$$X^{(t)} = \left(I - \omega D^{-1} L\right)^{-1} \left(\left((1-\omega)\, I + \omega D^{-1} U\right) X^{(t-1)} + \omega D^{-1} \left(X^{(t-1)} B + C\right)\right) \tag{5}$$

We obtain a method that is almost the same as the SOR method derived for the Sylvester equation in the way it is derived for linear systems. On the other hand, if there already is the double index in the formula, it can be used in all occurrences of the matrix X, hoping that it will improve the convergence:

Algorithm 2.3.

$$X^{(t)} = \left(I - \omega D^{-1} L\right)^{-1} \left(\left((1-\omega)\, I + \omega D^{-1} U\right) X^{(t-1,t)} + \omega D^{-1} \left(X^{(t-1,t)} B + C\right)\right) \tag{6}$$

We name the method without the double index "lSOR-like" ("l" for "little"), and the last one "bSOR-like" ("b" for "big").

3 Convergence of the methods

For the lSOR-like method we give two sufficient conditions under which it converges. Spectral matrix norm is denoted as $\|\cdot\|$.

Theorem 3.1 (First sufficient condition for lSOR-like convergence)**.** *Let* $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$, $A \neq 0$, $B \neq 0$, $\det(D) \neq 0$ *and*

$$G = \left\| D^{-1} L \right\| \left(1 + \left\| D^{-1} U \right\| + \left\| D^{-1} \right\| \cdot \|B\|\right), \tag{7}$$

$$H = 1 + \left\|D^{-1}U\right\| + \left\|D^{-1}\right\| \cdot \|B\| - \left\|D^{-1}L\right\|, \tag{8}$$

$$M = \left\|D^{-1}L\right\| (-1 + \left\|D^{-1}U\right\| + \left\|D^{-1}\right\| \cdot \|B\|), \tag{9}$$

$$N = -1 + \left\|D^{-1}U\right\| + \left\|D^{-1}\right\| \cdot \|B\| + \left\|D^{-1}L\right\|. \tag{10}$$

Then

1. *If*

$$\begin{cases} \omega \geq 1 \\ \left\|I - \omega D^{-1}L\right\| \geq \left\|(I - \omega D^{-1}L)^{-1}\right\|, \\ G \cdot \omega^2 + H \cdot \omega - 2 < 0 \end{cases} \tag{11}$$

   then lSOR-like converges.

2. *If*

$$\begin{cases} 0 < \omega \leq 1 \\ \left\|I - \omega D^{-1}L\right\| \geq \left\|(I - \omega D^{-1}L)^{-1}\right\|, \\ M \cdot \omega^2 + N \cdot \omega < 0 \end{cases} \tag{12}$$

   then lSOR-like converges.

3. *For $\omega = 1$ the conditions above are equivalent.*

Proof. **1.** By transforming the third assumption in (11): $(\left\|D^{-1}L\right\| + \left\|D^{-1}L\right\| \cdot \left\|D^{-1}U\right\| + \left\|D^{-1}L\right\| \cdot \left\|D^{-1}\right\| \cdot \|B\|) \cdot \omega^2 + \left(1 + \left\|D^{-1}U\right\| + \left\|D^{-1}\right\| \cdot \|B\| - \left\|D^{-1}L\right\|\right) \cdot \omega - 2 < 0$ we obtain $\omega - 1 + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| + \omega^2 \left\|D^{-1}L\right\| - \omega \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \cdot \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| < 1$. If so, then there exists real $\gamma \in (0, 1)$, such that $\omega - 1 + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| + \omega^2 \left\|D^{-1}L\right\| - \omega \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma < 1$. Thus $\omega - 1 + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| + \omega(\omega - 1) \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma$.
From the first assumption in (11) ($\omega \geq 1$), we have: $\omega - 1 = |1 - \omega|$, so we can write $|1 - \omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| + \omega |1 - \omega| \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma$, which leads to:

$$\left(1 + \omega \left\|D^{-1}L\right\|\right) \left(|1 - \omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \cdot \|B\|\right) \leq \gamma. \tag{13}$$

From the properties of a norm, and from the second assumption in (11), we obtain $1 + \omega \left\|D^{-1}L\right\| \geq \left\|\mathbb{I} - \omega D^{-1}L\right\| \geq \left\|\left(\mathbb{I} - \omega D^{-1}L\right)^{-1}\right\|$ and $(|1 - \omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \cdot \|B\|) \geq \left\|(1 - \omega)\mathbb{I} + \omega D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|$, which gives:

$$\left\|\left(\mathbb{I} - \omega D^{-1}L\right)^{-1}\right\| \left(\left\|(1 - \omega)\mathbb{I} + \omega D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|\right) \leq \gamma. \tag{14}$$

Multiplying both sides of (14) by $\left\|E^{(t-1)}\right\|$, where $E^{(t-1)} = X - X^{(t-1)}$, gives: $\left\|(\mathbb{I} - \omega D^{-1}L)^{-1}\right\| \cdot (\left\|(1 - \omega)\mathbb{I} + \omega D^{-1}U\right\| \left\|E^{(t-1)}\right\| + \omega \left\|D^{-1}\right\| \left\|E^{(t-1)}\right\| \|B\|) \leq \gamma \left\|E^{(t-1)}\right\|$. From the properties of a norm we obtain: $\left\|(\mathbb{I} - \omega D^{-1}L)^{-1}\right\| \cdot (\left\|(1 - \omega)\mathbb{I} + \omega D^{-1}U\right\| \left\|E^{(t-1)}\right\| + \left\|D^{-1}\right\| \left\|E^{(t-1)}\right\| \|B\|) \geq \left\|(\mathbb{I} - \omega D^{-1}L)^{-1}(((1 - \omega)\,\mathbb{I} + \omega D^{-1}U)E^{(t-1)} + \omega D^{-1}\left(E^{(t-1)}B\right))\right\| = \left\|E^{(t)}\right\|$, which gives: $\gamma \left\|E^{(t-1)}\right\| \geq \left\|E^{(t)}\right\|$. This means that the sequence of the error norms is monotonically decreasing, and the transformation (5) is a contraction. The solution X of the initial problem $AX - XB = C$ is a fixed point of that transformation, so the lSOR-like method is convergent to that solution.

2. Similarly to the proof of (11), by transforming the third assumption in (12): $(-\left\|D^{-1}L\right\| + \left\|D^{-1}L\right\| \cdot \left\|D^{-1}U\right\| + \left\|D^{-1}L\right\| \cdot \left\|D^{-1}\right\| \cdot \|B\|) \cdot \omega^2 + (-1 + \left\|D^{-1}U\right\| + \left\|D^{-1}\right\| \cdot \|B\| + \left\|D^{-1}L\right\|) \cdot \omega < 0$ we obtain $-\omega + 1 + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| - \omega^2 \left\|D^{-1}L\right\| + \omega \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| < 1$. If so, then there exists real $\gamma \in (0, 1)$, such that $-\omega + 1 + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| - \omega^2 \left\|D^{-1}L\right\| + \omega \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma$. Thus $1 - \omega + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| + \omega(1 - \omega) \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma$.
From the first assumption in (12) ($0 < \omega \leq 1$), we have $1 - \omega = |1 - \omega|$, so we can write $|1 - \omega| + \omega \left\|D^{-1}U\right\| +$

$\omega \left\|D^{-1}\right\| \|B\| + \omega\,|1-\omega| \left\|D^{-1}L\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}U\right\| + \omega^2 \left\|D^{-1}L\right\| \left\|D^{-1}\right\| \|B\| \leq \gamma$. That leads to (13) and then to (14), so from this point, the proof of (12) is the same as the proof of (11).

3. To prove the equivalence of (11) and (12) for $\omega = 1$ it suffices to show that $G + H - 2 = M + N$ which is valid as can be readily checked. This ends the proof. □

Theorem 3.2 (Second sufficient condition for lSOR-like convergence)**.** *Let $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$, $A \neq 0$, $B \neq 0$ and* $\det(D) \neq 0$. *If $B \neq 0$, $\omega > 0$ and*

$$\frac{|1-\omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|}{1 - \omega \left\|D^{-1}L\right\|} \in (0;1) \tag{15}$$

then lSOR-like converges.

Proof. If (15) holds, then $1 - \omega \left\|D^{-1}L\right\| > |1-\omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\| > 0$, which leads to $\left\|\omega D^{-1}L\right\| = \omega \left\|D^{-1}L\right\| < 1$. Moreover, there exists real $\gamma \in (0,1)$, such that $\frac{|1-\omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|}{1 - \omega \left\|D^{-1}L\right\|} \leq \gamma$. Using the following inequality of norms:

$$\left\|\omega D^{-1}L\right\| < 1 \Rightarrow \left\|(\mathbb{I} - \omega D^{-1}L)^{-1}\right\| \leq \frac{1}{1 - \left\|\omega D^{-1}L\right\|} \tag{16}$$

and the properties of a norm, we obtain: $\left\|(\mathbb{I} - \omega D^{-1}L)^{-1}\right\| (|1-\omega| + \omega \left\|D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|) \leq \gamma$. Thus we have $\left\|\left(\mathbb{I} - \omega D^{-1}L\right)^{-1}\right\| \left(\left\|(1-\omega)\mathbb{I} + \omega D^{-1}U\right\| + \omega \left\|D^{-1}\right\| \|B\|\right) \leq \gamma$, which is the same as (14) in the proof of Theorem 3.1, so from this point the proof follows as the previous one. □

4 Numerical tests

We present two numerical tests. In both we compare the four methods: classic SOR for linear system (2) and the SOR-like methods, that is SOR-like, bSOR-like and lSOR-like. Convergence of all these methods depends on the value of a parameter ω. In solving the Sylvester equation with any of the SOR-like methods, we also use the property from Remark 1.6, i.e. we test a method for a number of problems of the form $(A-\alpha I_m)X - X(B-\alpha I_n) = C$ for multiple values of the parameter α (as mentioned in Remark 1.6, the parameter α has no influence on the corresponding linear system (2), so it is not used in testing the classic SOR).

The stopping condition is $\left\|X^{(t)} - X^{(t-1)}\right\|_2 \leq tol$ or $t \geq t_{MAX}$, where $tol = 2.2204 \cdot 10^{-13}$ and $t_{MAX} = 750$. Results for the SOR method are given in tables, which contain: t, the number of iterations and $l = \log_{10}(\left\|X - X^{(t)}\right\|_2)$, which shows how fast a method converged if it reached the t_{MAX} limit. X is the value computed using the Matlab function $lyap$.

The SOR-like methods were tested for multiple α and ω values. On Figures 1, 2, 3, 5, 6, 7 we plotted the pairs (α, ω) for which the methods converge. We call the set of such (α, ω) pairs the *region of convergence*. More precise data for a chosen value of α is shown in the tables. Empty cells in those tables mean that a method did not converge for the particular pair of parameters (α, ω). We present no results for $\omega < 0$ as for that values all the methods diverge.

Tests were run using Matlab 7.11.0.584 (R2010b) on a 64-bit Linux system.

4.1 Example 1

The first example comes from discretization of the Poisson equation $-\Delta u = 2\sin(y)(x\sin(x) - \cos(x))$, $u_{|\delta U} = 0$, $U = [0,\pi] \times [0, 2\pi]$. $A \in \mathbb{R}^{12\times 12}$ and $B \in \mathbb{R}^{25\times 25}$ are tridiagonal matrices of the form:

$$A = \begin{bmatrix} 2-\alpha & -1 & 0 & 0 \\ -1 & 2-\alpha & -1 & 0 \\ 0 & -1 & \ddots & \vdots \\ 0 & 0 & \dots & 2-\alpha \end{bmatrix}, B = \begin{bmatrix} -2-\alpha & 1 & 0 & 0 \\ 1 & -2-\alpha & 1 & 0 \\ 0 & 1 & \ddots & \vdots \\ 0 & 0 & \dots & -2-\alpha \end{bmatrix},$$

Those matrices can be also written as follows: $A = (2-\alpha)I_{12} - P_{12}$, $B = P_{25} - (2+\alpha)I_{25}$, where P_n is the adjacency matrix of a path graph with n vertices. The eigenvalues of P_n are $2\cos(\frac{j\pi}{n+1})$, $j = 1, \ldots, n$ (see [2, p. 32]), so eigenvalues of A are $\lambda_k(A) = 2 - \alpha - 2\cos(\frac{k\pi}{13})$, and those of B are $\lambda_l(B) = 2\cos(\frac{l\pi}{26}) - 2 - \alpha$ and $\lambda_k(A) \in (-\alpha; 4-\alpha)$, $\lambda_l(B) \in (-4-\alpha; -\alpha)$, where $k = 1, \ldots, 12$, $l = 1, \ldots, 25$.

The tested values of α were $[-11, -10.5, -10, \ldots, 1.5, 2, 2.5]$ and the tested values of ω were $[0.1, 0.2, 0.3, \ldots, 4.8, 4.9, 5]$. The matrix $C \in \mathbb{R}^{12\times 25}$ was obtained from the right hand side function of the Poisson equation. It was also the initial guess $X^{(0)}$.

Table 1. SOR: numbers of iterations for tested ω.

ω	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
t	750	750	750	750	750	750	750	750	750	681
l	1.8	0.9	0.2	−0.7	−1.7	−2.8	−4.1	−5.6	−7.5	−8.5
ω	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
t	561	460	372	295	225	159	89	133	272	750
l	−8.6	−8.7	−8.8	−8.9	−9.0	−9.2	−9.9	−9.6	−9.6	2.5

Table 2. SOR-like: numbers of iterations for tested ω, ($\alpha = -5$).

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$t_{SOR-like}$	750	750	750	750	750	750	750	750	750	750
$l_{SOR-like}$	2.2	1.7	1.2	0.7	0.2	−0.3	−0.8	−1.4	−2.1	−2.9
$t_{lSOR-like}$	750	750	750	750	750	750	750	750	750	750
$l_{lSOR-like}$	2.2	1.7	1.2	0.8	0.4	0.0	−0.4	−0.8	−1.3	−1.7
$t_{bSOR-like}$	750	750	750	750	750	750	750	750	750	750
$l_{bSOR-like}$	2.2	1.6	1.1	0.6	0.2	−0.4	−0.9	−1.6	−2.2	−2.9
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$t_{SOR-like}$	750	750	750	750	690	579	477	382	290	192
$l_{SOR-like}$	−3.8	−4.8	−6.1	−7.6	−8.5	−8.6	−8.7	−8.8	−8.9	−9.1
$t_{lSOR-like}$	750	750	750	750	750	750	750	750	750	750
$l_{lSOR-like}$	−2.2	−2.7	−3.2	−3.7	−4.3	−4.8	−5.4	−6.0	−6.6	−7.3
$t_{bSOR-like}$	750	750	750	750	750	750	692	633	581	536
$l_{bSOR-like}$	−3.7	−4.5	−5.4	−6.3	−7.3	−8.3	−8.5	−8.5	−8.6	−8.6
	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0
$t_{SOR-like}$	152	233	572							
$l_{SOR-like}$	−9.5	−9.2	−9.2							
$t_{lSOR-like}$	750	750	750							
$l_{lSOR-like}$	−8.0	−8.7	−9.5							
$t_{bSOR-like}$	497	462	432	405	381	360	341	325	311	
$l_{bSOR-like}$	−8.6	−8.7	−8.7	−8.7	−8.8	−8.8	−8.8	−8.8	−8.9	

As seen in Table 2, all three SOR-like methods converged slowly, but unlike the original SOR method, they converged for $\omega \geq 2$. As supposed, lSOR-like was slower than SOR-like, although bSOR-like was not faster, but it worked for a wider range of ω.

Figures 1, 2, 3 show the (α, ω) pairs for which the three SOR-like methods converge.

Fig. 1. Pairs (α, ω) for which SOR-like converged.

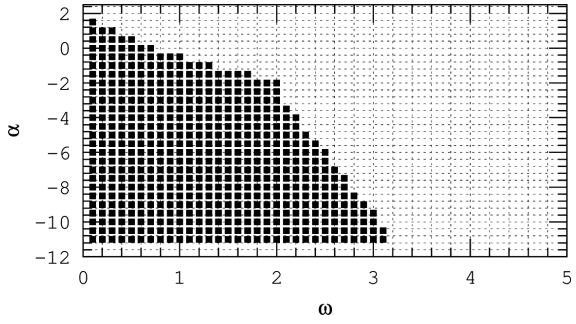


Fig. 2. Pairs (α, ω) for which lSOR-like converged.

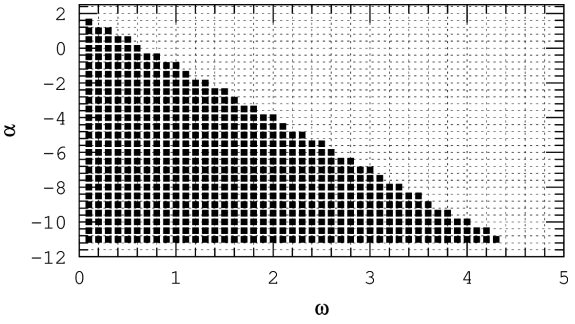


Fig. 3. Pairs (α, ω) for which bSOR-like converged.

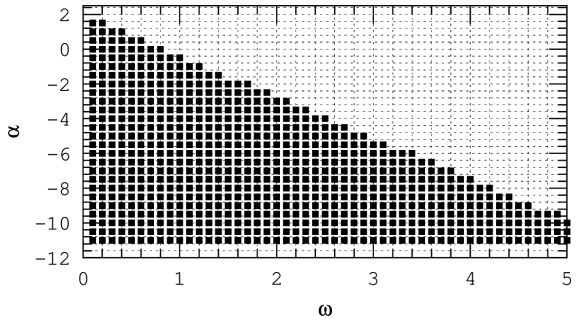
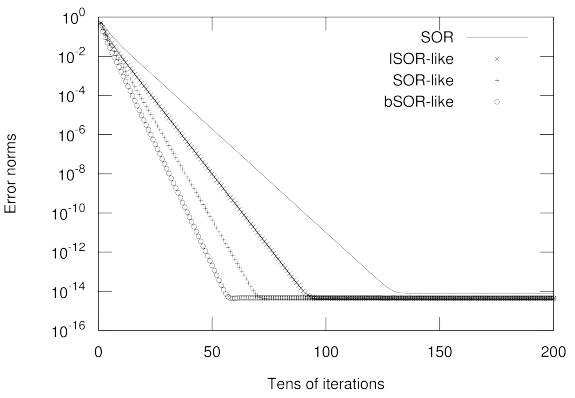


Fig. 4. Convergence comparison for $(\alpha, \omega) = (-0.5, 0.8)$.



The SOR-like method worked the best for $(\alpha, \omega) = (-2, 1.7)$. It needed 89 iterations, the same as the minimum for SOR, and the best result from amongst all the methods.

The lSOR-like method worked the best for $(\alpha, \omega) = (-6.5, 2.8)$. It needed 750 iterations in all the cases, but the error norm was the least then $(2.95 \cdot 10^{-10})$, which means the fastest convergence.

The bSOR-like method worked the best for $(\alpha, \omega) = (-5, 2.9)$. It needed 311 iterations.

The bSOR-like converged in the biggest number of cases, SOR-like in the smallest, but there were some values of (α, ω) (i.e., $(-2, 1.9)$) for which SOR-like converged but bSOR-like did not.

The last figure in this example (4) shows in a semi-logarithmic scale the convergence of all the tested methods. It shows that all the methods are stable. As the SOR-like methods did not converge for the 'natural', default values $(\alpha, \omega) = (0, 1)$, we give the test for $(\alpha, \omega) = (-0.5, 0.8)$, and without the error norm stopping condition (there were 2000 iterations computed for all the methods). Note that the marked values are for the iterations numbers $t = 10k, k = 1, \ldots, 200$.

The conditions of Theorem 3.1 were not satisfied for any of the tested values. Among the tested values, the conditions of Theorem 3.2 were satisfied for $\alpha = [-11, -10.5, \ldots, -2]$ and $\omega = [0.1, 0.2, \ldots, 1]$. All the SOR-like methods did converge for that values.

4.2 Example 2

$$A = \begin{bmatrix} 10-\alpha & 1 & 1 \\ 1 & 10-\alpha & 1 \\ 1 & 1 & 10-\alpha \end{bmatrix}, B = \begin{bmatrix} 1-\alpha & 1 & 1 \\ 1 & 2-\alpha & 3 \\ 1 & 3 & 6-\alpha \end{bmatrix}, C = \begin{bmatrix} 9 & 6 & 2 \\ 9 & 6 & 2 \\ 9 & 6 & 2 \end{bmatrix},$$

where the tested values of α were $[-20, -19, -18, \ldots, 8, 9, 10]$ and the tested values of ω were $[0.125, 0.25, 0.375, \ldots, 5.75, 5.875, 6]$. The matrix C was the initial guess $X^{(0)}$.

Table 3. SOR: numbers of iterations for tested ω.

ω	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
t	750	399	256	181	135	103	80	61
l	−10	−11	−11.2	−11.4	−11.5	−11.7	−11.9	−12.1
ω	1.125	1.25	1.375	1.5	1.625	1.75	1.875	2
t	46	35	39	53	76	123	263	
l	−12.4	−12.4	−12.3	−12.6	−12.4	−12.3	−12.3	

Fig. 5. Pairs (α, ω) for which SOR-like converged.

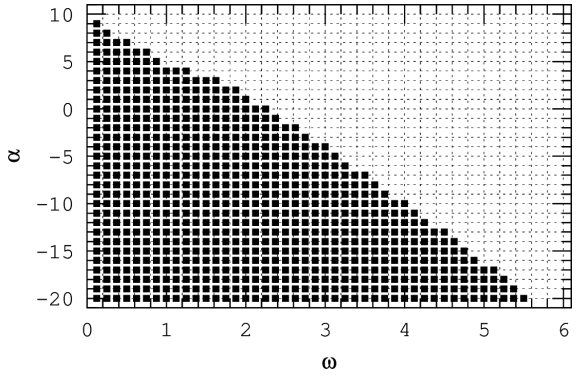
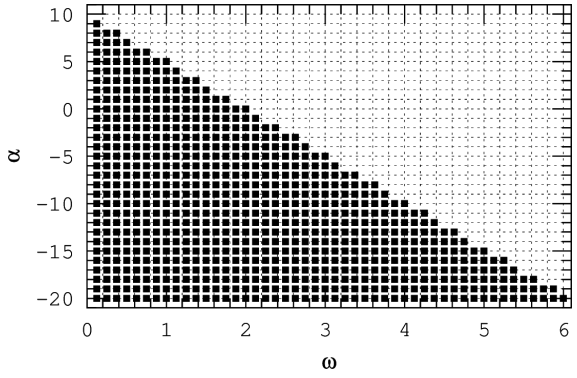


Fig. 6. Pairs (α, ω) for which lSOR-like converged.



The SOR-like method worked the best for $(\alpha, \omega) = (1, 1.625)$. It needed 62 iterations.

The lSOR-like method worked the best for $(\alpha, \omega) = (-16, 4.625)$. It needed 115 iterations.

The bSOR-like method worked the best for $(\alpha, \omega) = (-12, 3.75)$. It needed 76 iterations.

In this example, whenever bSOR-like converged, at least one of SOR-like and lSOR-like also converged, which can be seen in Figures 5, 6, 7.

The SOR method worked the best for $\omega = 1.25$. It needed 35 iterations, being the most efficient of all tested methods.

Table 4. SOR-like: numbers of iterations for tested ω, $(\alpha = 0)$.

	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{7}{8}$	1	$\frac{9}{8}$
$t_{SOR\text{-}like}$	750	750	514	381	298	242	201	169	144
$l_{SOR\text{-}like}$	−5.7	−10.4	−10.9	−11.0	−11.1	−11.3	−11.4	−11.4	-11.5
$t_{lSOR\text{-}like}$	750	750	569	431	346	289	247	215	190
$l_{lSOR\text{-}like}$	−5.6	−10.1	−11.3	−11.5	−11.6	−11.7	−11.7	−11.8	-11.8
$t_{bSOR\text{-}like}$	750	750	517	381	298	241	200	169	145
$l_{bSOR\text{-}like}$	−5.4	−10.3	−10.9	−11.0	−11.2	−11.2	−11.3	−11.4	-11.5
	$\frac{5}{4}$	$\frac{11}{8}$	$\frac{3}{2}$	$\frac{13}{8}$	$\frac{7}{4}$	$\frac{15}{8}$	2	$\frac{17}{8}$	$\frac{9}{4}$
$t_{SOR\text{-}like}$	123	105	90	77	65	70	103	194	750
$l_{SOR\text{-}like}$	−11.6	−11.7	−11.8	−11.9	−12.0	−12.7	−12.6	−12.5	-9.5
$t_{lSOR\text{-}like}$	170	154	140	128	118	194	750		
$l_{lSOR\text{-}like}$	−11.9	−12.0	−12.1	−12.1	−12.2	−13.0	−8.0		
$t_{bSOR\text{-}like}$	125	109	95	83	93	218			
$l_{bSOR\text{-}like}$	−11.6	−11.7	−11.7	−11.9	−12.9	−12.5			

Fig. 7. Pairs (α, ω) for which bSOR-like converged.

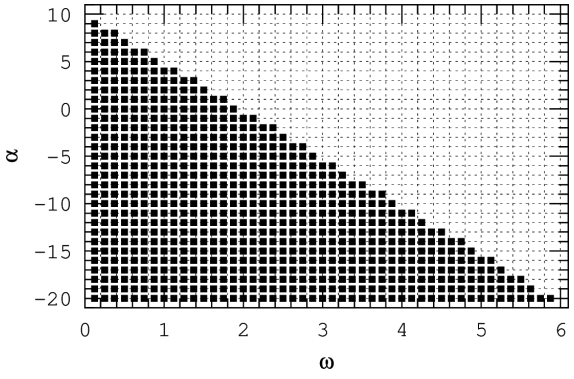


Fig. 8. Convergence comparison for $(\alpha, \omega) = (3, 1.25)$.

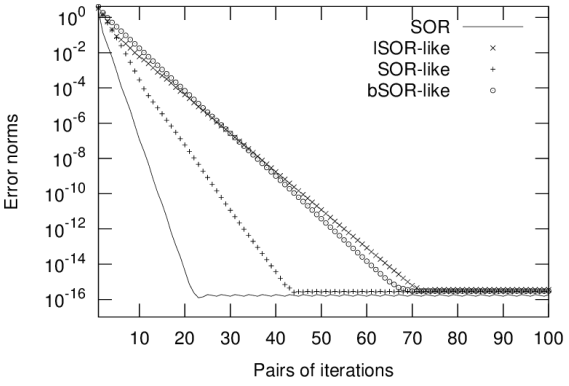


Figure 8 shows the convergence of all the tested methods. Again, it shows that all the methods are stable. Similarly to Example 4.1, the SOR-like methods did not converge for $(\alpha, \omega) = (0, 1)$ and we give the test for $(\alpha, \omega) = (3, 1.25)$, and without the error norm stopping condition (there were 200 iterations computed for all the methods). The marked values are for the even iteration numbers ($t = 2k, k = 1, \ldots, 100$).

The conditions of Theorem 3.1 and 3.2 were not satisfied for any of the tested values.

5 Conclusions and open problems

The examples given show that, in general, for SOR-like methods it need not to be $\omega \in (0; 2)$, as it is for the original SOR. Moreover, we see that the matrices A and B and parameters α and ω for those methods are connected in some way: the methods converged when $\alpha < \max_{i=1,\ldots,n} A_{i,i} (= A_{1,1}$ in those cases) and $\omega \in (0, \omega_{max}(\alpha))$. One can also observe that the best found ω_{opt} was relatively close to the boundary $\omega_{max}(\alpha)$, unlike for the SOR method. As the region of convergence of the SOR-like is generally bigger than for SOR, finding a way to obtain the values $\omega_{opt}(\alpha)$ – the optimal value – and $\omega_{max}(\alpha)$ – the greatest ω for which a method converges for a given α – seem the most important in studying the properties of the SOR-like methods. However, it can be supposed that in many cases it would be enough to take α close to $\max_{i=1,\ldots,n} A_{i,i}$ and ω close to 0, to make a method converge.

A further issue is to give other sufficient conditions and some necessary conditions on the convergence of the lSOR-like method. The conditions in Theorems 3.1 and 3.2 are not contradicting (there are matrices A, B, C and parameters ω for which they are satisfied). Among the values of parameters tested in subsection 4.1 the were some satisfying the conditions set of Theorem 3.2, but they were only a part of those for which the lSOR-like method did converge. This shows that the conditions in the theorems are far from being necessary. However, for the lSOR-like and bSOR-like methods there are no conditions on convergence at all. This is due to the use of the double index $(t-1, t)$ – finding a way to implement the methods without the double index could enable giving similar conditions and make programmes using those methods work faster (depending on the matrix multiplication implementation) and make it easier to study the methods.

References

[1] Bartels, R.H., Stewart G.W., Algorithm 432: the solution of the matrix equation $AX - BX = C$, Communications of the ACM, 1972, 15(9), 820-826

[2] Beineke, L.W., Wilson, R.J. (Eds.), Topics in Algebraic Graph Theory, Encyclopedia Math. Appl., Cambridge University Press, 102, Cambridge University Press, 2005

[3] Datta, B., Numerical Methods for Linear Control Systems, Elsevier Science, 2004

[4] Golub G.H., Nash S., Van Loan C., Hessenberg–Schur method for the problem $AX + XB = C$, IEEE Trans. Automat. Control, 1979, AC-24(6), 909-913
[5] Hu D.Y., Reichel L., Krylov-subspace methods for the Sylvester equation, Linear Algebra Appl., 1992, 172, 283-313
[6] Lancaster, P., Tismenetsky, M., The theory of matrices: with applications, 2nd ed., Academic Press, Orlando, 1985
[7] Roth W.E., The equations $AX - YB = C$ and $AX - XB = C$ in matrices, Proc. Amer. Math. Soc., 1952, 3(3), 392-396
[8] Simoncini, V., On the numerical solution of $AX - XB = C$, BIT, 1996, 36(4), 814-830
[9] Starke G., Niethammer W., SOR for $AX - XB = C$, Linear Algebra Appl., 1991, 154/156, 355-375
[10] Woźnicki, Z.I., Solving linear systems: an analysis of matrix prefactorization iterative methods, Matrix Editions, 2009