Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER
PUBLISHER

Research Article

A Jordan-Schur Algorithm for Solving Sylvester and Lyapunov Matrix Equations

P. H. Petkov

Department of Engineering Sciences, Bulgarian Academy of Sciences, Sofia 1040, Bulgaria
E-mail: php@tusofia.bg

Received: 23 August 2022; **Revised:** 17 October 2022; **Accepted:** 27 October 2022

Abstract: This paper presents a version of the Bartels-Stewart algorithm for solving the Sylvester and Lyapunov equations that utilizes the Jordan-Schur form of the equation matrices. The Jordan-Schur form is a type of Schur form which contains additional information about the Jordan structure of the corresponding matrix. This information can be used to solve more efficiently the Sylvester and Lyapunov equations in some cases. A two-level algorithm is implemented which allows us to find directly non-scalar blocks of the solution matrix. These blocks have sizes that are determined by the Weyr characteristics associated with the eigenvalues of the equation matrices. In the case of large elements of the Weyr characteristics associated with multiple eigenvalues, the determination of the solution blocks can be done more efficiency. Also, the blocks equations can be more appropriate in solving the Sylvester and Lyapunov equations in the case of parallel computations. Results obtained from numerical experiments confirm that the accuracy of the new algorithm is comparable with the accuracy of the Bartels-Stewart algorithm.

***Keywords*:** Sylvester equation, Lyapunov equation, Jordan-Schur form, Bartels-Stewart algorithm

MSC: 65F45, 65F15

1. Introduction

The Sylvester

$$AX + XB = C,\ A \in \mathbb{C}^{n\times n},\ B \in \mathbb{C}^{m\times m} \tag{1}$$

and Lyapunov

$$A^H X + XA = C,\ A \in \mathbb{C}^{n\times n} \tag{2}$$

matrix equations arise frequently in the solution of many scientific problems. Several methods are proposed for the computational solution of these equations, see for an extensive bibliography of the surwey paper [1]. One of the best available algorithms for solving these equations is the Bartels-Stewart algorithm [2] which exploits the Schur form of

DOI: https://doi.org/10.37256/cm.3420221851

the matrices A and B. Using this algorithm the solution of the corresponding equation is reduced to the solution of scalar (in the complex case) equations which are easily solved recursively. The Schur form is obtained in a numerically stable way which ensures a robust numerical behavior of the Bartels-Stewart algorithm. Further development of the Bartels-Stewart algorithm is the highly efficient recursive blocked algorithms for solving the Sylvester and Lyapunov equation proposed in [3].

This paper presents a new version of the Bartels-Stewart algorithm for solving the Sylvester and Lyapunov equations which instead of the Schur form utilizes the Jordan-Schur form of the matrices A and B. The Jordan-Schur form is a type of Schur form which contains additional information about the Jordan structure of the corresponding matrix. This information can be used to solve more efficiently (1) and (2) in some cases. A two-level algorithm is implemented which allows us to find directly non-scalar blocks of the solution matrix. These blocks have sizes that are determined by the Weyr characteristics associated with the eigenvalues of the matrices A and B. In the case of multiple eigenvalues with large elements of the Weyr characteristics this may improve the efficiency. Also, the blocks equations can be more appropriate for solving the Sylvester and Lyapunov equations in the case of parallel computations. Results from numerical experiments are presented which confirm that the accuracy of the new algorithm is comparable with the accuracy of the Bartels-Stewart algorithm.

The paper is organized as follows. The Jordan-Schur form of a matrix is introduced briefly in section 2. A new algorithm for solving the Sylvester equation is presented in section 3 and the modification of this algorithm for solving the Lyapunov equation is described in section 4. The numerical properties of the algorithm are discussed in section 5. Several numerical experiments illustrating the behavior of the proposed algorithm are presented in section 6 and some conclusions are drawn in section 7.

Notation will be the following. $\mathbb{C}$ is the set of complex numbers; A^T is the transpose of A; $A^H = \bar{A}^T$ is the Hermitian transpose (the complex conjugate transpose) of A; $0_{n\times m}$ is the zero $n \times m$ matrix; In is the identity matrix of size $n \times n$; $\lambda_i(A)$ is the ith eigenvalue of A; $\|A\|_2$ and $\|A\|_F$ are the spectral norm and the Frobenius norm of A, respectively; null(A) is the null space of A; dim($\mathcal{X}$) is the dimension of the subspace $\mathcal{X}$; ΔA is a perturbation of A; cond(A) is the condition number of A in respect to the inversion; cond_{sylv} and cond_{lyap} are the condition numbers of the Sylvester and Lyapunov equations, respectively; $fl(.)$ is an expression evaluated in floating point arithmetic; vec(A) is an $n \cdot m$ vector obtained by stacking the columns of the $n \times m$ matrix $A = [a_1, a_2, ..., a_m]$, $\mathrm{vec}(A) = [a_1^T, a_2^T, ..., a_m^T]^T$, $A \otimes B$ is the Kronecker product of A and B and =: denotes equal by definition. The lower case italics d, e, f, g, i, j, k, ℓ, p, q are used to represent integer variables.

2. The Jordan-Schur form

Assume that the complex $n \times n$ matrix A has p distinct eigenvalues λ_i, $i = 1, 2, ..., p$ each with a multiplicity κ_i, $i = 1, 2, ..., p$. Then the Jordan-Schur form J_S of A [4, 5], [6, Sect. 2.5], is a block-triangular matrix that is unitarily similar to A,

$$J_S = U^H A U = \begin{bmatrix} S(w_1, \lambda_1) & T_{12} & \cdots & T_{1p} \\ & S(w_2, \lambda_2) & \cdots & T_{2p} \\ & & \ddots & \vdots \\ & & & S(w_p, \lambda_p) \end{bmatrix}, \tag{3}$$

where the diagonal blocks $S(w_i, \lambda_i)$, $i = 1, 2, ..., p$ have the form

$$S(w_i, \lambda_i) = \begin{bmatrix} \lambda_i I_{w_{i1}} & N_{12}^{(i)} & \cdots & N_{1,h_i-1}^{(i)} & N_{1,h_i}^{(i)} \\ & \lambda_i I_{w_{i2}} & \cdots & N_{2,h_i-1}^{(i)} & N_{2,h_i}^{(i)} \\ & & \ddots & \vdots & \vdots \\ & & & \lambda_i I_{w_{i,h_i-1}} & N_{h_i-1,h_i}^{(i)} \\ & & & & \lambda_i I_{w_{i,h_i}} \end{bmatrix} \in \mathbb{C}^{\kappa_i \times \kappa_i}. \tag{4}$$

for some $h_i \geq 1$. The sequence of positive numbers $w_{i1}, w_{i2}, ..., w_{i,h_i}$ is called the Weyr characteristic of A associated with λ_i which satisfy

$$w_{i1} \geq w_{i2} \geq \ldots \geq w_{i,h_i}, \; \sum_{j=1}^{h_i} w_{ij} = \kappa_i, \tag{5}$$

for $i = 1, 2, ..., p$, for more details see [7, p. 170], [8]. Furthermore, we write $w_i = w_i(A, \lambda_i) = (w_{i1}, w_{i2}, ..., w_{i,h_i})$.

The Jordan-Schur form is actually a Schur form with an additional structure determined by the Weyr characteristics. The Weyr characteristics are closely connected with the Jordan structure of the matrix. In fact, the Weyr characteristic $w_i(A, \lambda_i)$ determines in a unique way the Segre characteristic

$$s_i =: s_i(A, \lambda_i) = (s_{1i}, s_{2i}, \ldots, s_{q_i,i}), \, s_{1i} \geq s_{2i} \geq \ldots \geq s_{q_i,i}, \; \sum_{j=1}^{q_i} s_{ji} = \kappa_i \tag{6}$$

associated with λ_i whose elements are equal to the sizes of the blocks in the Jordan form of A having λ_i as an eigenvalue. The Weyr characteristic $w_i(A, \lambda_i)$ and the Segre characteristic $s_i(A, \lambda_i)$ associated with λ_i are conjugate partitions of κ_i [7, p. 170] and can be found from each other by the following rule:

The element s_{ji} of $s_i(A, \lambda_i)$ is equal to the number of the elements of $w_i(A, \lambda_i)$ that are greater than j and vice versa.

Hence the largest element s_{1i} (the size of the largest Jordan block) of the Segre characteristic is equal to the number h_i of the elements of the Weyr characteristic. In turn, the first element w_{i1} of the Weyr characteristic (the geometric multiplicity of the eigenvalue λ_i) is equal to the number q_i of the elements of Segre characteristic (the number of Jordan blocks) associated with λ_i. Note that the number h_i is the smallest positive integer for which

$$\dim(\text{null}((S(w_i, \lambda_i) - \lambda_i I_{\nu_i})^{h_i})) = \kappa_i$$

and thus is the index of nilpotence of $S(w_i, \lambda_i) - \lambda_i I_{\kappa_i}$.

The Jordan-Schur form is an intermediate step in the reduction of the matrix A to the numerical Jordan form. In particular, this form can be found by a slight modification of the algorithm of Kågström and Ruhe [9, 10] for reducing A to Jordan form using the following steps [11]:

1. Reduction of A to upper triangular (Schur) form $T = Q^H A U$ using unitary similarity transformations. This is done by the well known QR algorithm [12, Ch. 7], [13, Ch. 2].

2. Grouping the close eigenvalues. At this step, the algorithm obtains an upper triangular form with eigenvalues, sorted such that close eigenvalues appear in adjacent positions. This is done by a sequence of unitary transformations (complex plane rotations), each of them exchanging two adjacent diagonal elements [14].

3. Clustering of multiple eigenvalues in blocks. At this step, the close eigenvalues are grouped in p clusters corresponding to numerically multiple eigenvalues. This is done by using Gershgorin heuristic to construct a diagonal similarity transformation which isolates groups of intersecting discs containing eigenvalues of the matrix $T + \Delta T$, $\|\Delta T\| \leq ein$ where ein is an appropriate tolerance reflecting the uncertainty in A. A sophisticated algorithm for this aim is described in [4]. As a result, the upper triangular form of A takes the block-triangular form

$$\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1p} \\ & T_{22} & \cdots & T_{2p} \\ & & \ddots & \vdots \\ & & & T_{pp} \end{bmatrix},$$

where each diagonal block $T_{ii} \in \mathbb{C}^{\kappa_i \times \kappa_i}$ corresponds to a distinct numerically multiple eigenvalue λ_i with multiplicity κ_i.

4. Transforming the diagonal blocks to staircase form. At this step each block $T_{ii} - \lambda_i I_{\kappa_i}$ is reduced by using unitary similarity transformations into the so called staircase form

$$S(w_i, \lambda_i) - \lambda_i I_{\kappa_i} = \begin{bmatrix} 0 & N_{12}^{(i)} & N_{13}^{(i)} & \dots & N_{1,s_{1i}-1}^{(i)} & N_{1,s_{1i}}^{(i)} \\ 0 & 0 & N_{23}^{(i)} & \dots & N_{2,s_{1i}-1}^{(i)} & N_{2,s_{1i}}^{(i)} \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & N_{s_{1i}-2,s_{1i}-1}^{(i)} & N_{s_{1i}-2,s_{1i}}^{(i)} \\ 0 & 0 & 0 & \dots & 0 & N_{s_{1i}-1,s_{1i}}^{(i)} \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

where $N_{k\ell}^{(i)} \in \mathbb{C}^{w_{ik} \times w_{i\ell}}$, $k = 1, 2, \dots s_{1i} - 1$, $\ell = 2, 3, \dots s_{1i}$. The reduction of each upper triangular block $T_{ii} - \lambda_i I_{\kappa_i}$ to staircase form is done in $h_i = s_{1i}$ steps, where the index of nilpotency h_i of the block is not known a priori and is found during the reduction process. The reduction to staircase form represents a consecutive deflation of null spaces done with the aid of the singular value decomposition. Determining the size of a null space requires a decision about the numerical rank of the corresponding matrix which is done by using a tolerance *tol*. A detailed description of the reduction can be found in [4, 8].

As a result of this step, one obtains the matrix $S(w_i, \lambda_i)$ (4), so that the Jordan structure of the block T_{ii} is determined.

In this paper we use only the complex Jordan-Schur forms of A and B.

Example 1 Consider an 8th order matrix A with a Jordan form

$$J_A = \operatorname{diag}\left(\begin{bmatrix} \lambda_1 & 1 & \\ & \lambda_1 & \\ & & \lambda_1 \end{bmatrix}, \begin{bmatrix} \lambda_2 & 1 & & & \\ & \lambda_2 & 1 & & \\ & & \lambda_2 & & \\ & & & \lambda_2 & 1 \\ & & & & \lambda_2 \end{bmatrix} \right)$$

and eigenvalue multiplicities $\kappa_1 = 3$ and $\kappa_2 = 5$, respectively. The Weyr characteristic of A associated with λ_1 is $w_1(A, \lambda_1) = (2, 1)$ and the Weyr characteristic associated with λ_2 is $w_2(A, \lambda_2) = (2, 2, 1)$. Hence the Jordan-Schur form of A is

$$J_{SA} = U^H A U = \left[\begin{array}{c|c} S(w_1, \lambda_1) & T_{12} \\ \hline & S(w_2, \lambda_2) \end{array} \right].$$

The diagonal blocks of the Jordan-Schur form look as follows

$$S(w_1, \lambda_1) = \left[\begin{array}{cc|c} \lambda_1 & & \mathrm{x} \\ & \lambda_1 & \mathrm{x} \\ \hline & & \lambda_1 \end{array} \right] \in \mathbb{C}^{\kappa_1 \times \kappa_1}, \; S(w_2, \lambda_2) = \left[\begin{array}{cc|cc|c} \lambda_2 & & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ & \lambda_2 & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \hline & & \lambda_2 & & \mathrm{x} \\ & & & \lambda_2 & \mathrm{x} \\ \hline & & & & \lambda_2 \end{array} \right] \in \mathbb{C}^{\kappa_2 \times \kappa_2},$$

where (x) denotes a nonzero element. The 3×5 off-diagonal block T_{12} has non-zero elements.

The reduction to Jordan-Schur form by the modified algorithm of Kågström and Ruhe is more reliable than the reduction to Jordan form. This follows from the fact that the computation of the Weyr and Jordan forms at the final step of the full algorithm requires using of additional non-unitary transformations which can be ill-conditioned in some cases. The numerical reduction to Jordan-Schur form is backward stable in the sense that the computed form is exact for a slightly perturbed matrix $A + \delta A$. The weak point of this reduction is the determination of the Weyr (and consequently the Segre) characteristics which requires accurate rank determination in presence of rounding errors. This may be a difficult task for matrices with ill-conditioned eigensystems and may lead to errors in the determination of the characteristics. The choice of the parameters *ein* and *tol* is discussed latter on in sect. 5.

In cases of distinct eigenvalues the Jordan-Schur form coincides with the Schur form and the Jordan-Schur algorithm do not have some advantages over the Schur algorithm for solving Sylvester and Lyapunov equations.

3 Solution of the Sylvester equation

3.1 *Preliminary transformations*

Let the matrix A have p distinct eigenvalues $\lambda_1, \lambda_2, ..., \lambda_p$ with the respective multiplicities $\kappa_1, \kappa_2, ..., \kappa_p$ and B has q distinct eigenvalues $\mu_1, \mu_2, ..., \mu_q$ with multiplicities $\nu_1, \nu_2, ..., \nu_q$. Denote the Weyr characteristic of A associated with λ_i by

$$wa(A,\lambda_i) = (wa_{i1}, wa_{i2}, \ldots, wa_{i,ha_i}),\ wa_{i1} \geq wa_{i2} \geq \ldots \geq wa_{i,ha_i},\ \sum_{k=1}^{ha_i} wa_{ik} = \kappa_i$$

and the Weyr characteristic of B associated with μ_j by

$$wb(B,\mu_j) = (wb_{j1}, wb_{j2}, \ldots, wb_{j,hb_j}),\ wb_{j1} \geq wb_{j2} \geq \ldots \geq wb_{j,hb_j},\ \sum_{\ell=1}^{hb_j} wb_{j\ell} = \nu_j.$$

Reduce the matrices A and B to the Jordan-Schur form

$$J_S A = U^H AU = \begin{bmatrix} S(wa_1,\lambda_1) & R_{12} & \cdots & R_{1p} \\ & S(wa_2,\lambda_2) & \cdots & R_{2p} \\ & & \ddots & \vdots \\ & & & S(wa_p,\lambda_p) \end{bmatrix} \in \mathbb{C}^{n\times n}, \tag{7}$$

$$J_{SB} = V^H BV = \begin{bmatrix} S(wb_1,\mu_1) & T_{12} & \cdots & T_{1q} \\ & S(wb_2,\mu_2) & \cdots & T_{2q} \\ & & \ddots & \vdots \\ & & & S(wb_q,\mu_q) \end{bmatrix} \in \mathbb{C}^{m\times m}, \tag{8}$$

where the diagonal blocks of J_{SA} and J_{SB} are given by

$$S(wa_i,\lambda_i) = \begin{bmatrix} \lambda_i I_{wa_{i1}} & M_{12}^{(i)} & \cdots & M_{1,ha_i}^{(i)} \\ & \lambda_i I_{wa_{i2}} & \cdots & M_{2,ha_i}^{(i)} \\ & & \ddots & \vdots \\ & & & \lambda_i I_{wa_{i,ha_i}} \end{bmatrix} \in \mathbb{C}^{\kappa_i\times\kappa_i},\ i = 1,2,\ldots,p \tag{9}$$

and

$$S(wb_j,\mu_j)=\begin{bmatrix}\mu_j I_{wb_{j1}} & N_{12}^{(j)} & \dots & N_{1,hb_j}^{(j)}\\ & \mu_j I_{wb_{j2}} & \dots & N_{2,hb_j}^{(j)}\\ & & \ddots & \vdots\\ & & & \mu_j I_{wb_{j,hb_j}}\end{bmatrix}\in\mathbb{C}^{\nu_j\times\nu_j},\ \ j=1,2,\dots,q. \tag{10}$$

Note that the matrices $S(wa_i, \lambda_i) - \lambda_i I_{\kappa_i}$ and $S(wb_j, \mu_j) - \mu_j I_{\nu_j}$ are in staircase form with indices of nilpotence ha_i and hb_j, respectively.

As it is well known [15, Ch. VIII], the Sylvester equation (1) has a unique solution for X if and only if

$$\lambda_i+\mu_j\neq 0,\, i=1,2,\dots,p,\ j=1,2,\dots,q.$$

Setting

$$\tilde{C}=U^H CV,\ \ \tilde{X}=U^H XV$$

one obtains the transformed Sylvester equation

$$J_{SA}\cdot\tilde{X}+\tilde{X}\cdot J_{SB}=\tilde{C}. \tag{11}$$

Let the partition of the matrices $\tilde{C}$ and $\tilde{X}$ be conformal to the partition of J_{SA} and J_{SB},

$$\tilde{C}=\begin{bmatrix}C_{11} & C_{12} & \dots & C_{1q}\\ C_{21} & C_{22} & \dots & C_{2q}\\ \vdots & \vdots & \ddots & \vdots\\ C_{p1} & C_{p2} & \dots & C_{pq}\end{bmatrix},\ \tilde{X}=\begin{bmatrix}X_{11} & X_{12} & \dots & X_{1q}\\ X_{21} & X_{22} & \dots & X_{2q}\\ \vdots & \vdots & \ddots & \vdots\\ X_{p1} & X_{p2} & \dots & X_{pq}\end{bmatrix}. \tag{12}$$

Then (11) can be written as a set of block Sylvester equations

$$S(wa_i,\lambda_i)X_{ij}+X_{ij}S(wb_j,\mu_j)=C_{ij}-\sum_{d=i+1}^{p}R_{id}X_{dj}-\sum_{e=1}^{j-1}X_{ie}T_{ej}, \tag{13}$$

$$i=1,2,\dots,p,$$

$$j=1,2,\dots,q.$$

Due to the specific structure of the matrices $S(wa_i, \lambda_i)$ and $S(wb_j, \mu_j)$, equation (13) will be called *staircase Sylvester equation*. Note that the dimensions of $S(wa_i, \lambda_i)$ and $S(wb_j, \mu_j)$ can be arbitrary depending on the multiplicities of λ_i and μ_j.

Thus the solution of (1) is reduced to the solution of $p \cdot q$ staircase Sylvester equations (13). Since the matrices $S(wa_i, \lambda_i)$ and $S(wb_j, \lambda_j)$ are in upper triangular form, these equations can be solved by the Bartels-Stewart algorithm. However, as shown in the next section, the solution of (13) can be done more efficiently if one uses the staircase structure of $S(wa_i, \lambda_i)$ and $S(wb_j, \lambda_j)$.

3.2 *Solution of staircase Sylvester equations*

Consider the staircase Sylvester equation

$$S_A \cdot Y + Y \cdot S_B = F \tag{14}$$

where

$$S_A = \begin{bmatrix} \lambda I_{wa_1} & M_{12} & \dots & M_{1,ha} \\ & \lambda I_{wa_2} & \dots & M_{2,ha} \\ & & \ddots & \vdots \\ & & & \lambda I_{wa_{ha}} \end{bmatrix} \in \mathbb{C}^{\kappa \times \kappa}, \ S_B = \begin{bmatrix} \mu I_{wb_1} & N_{12} & \dots & N_{1,hb} \\ & \mu I_{wb_2} & \dots & N_{2,hb} \\ & & \ddots & \vdots \\ & & & \mu I_{wb_{hb}} \end{bmatrix} \in \mathbb{C}^{\nu \times \nu}. \tag{15}$$

The solution of (14) can be done exploiting the structure of the matrices S_A and S_B revealed by (15). Partition the $\kappa \times \nu$ matrices F and Y as

$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1,hb} \\ F_{21} & F_{22} & \dots & F_{2,hb} \\ \vdots & \vdots & \ddots & \vdots \\ F_{ha,1} & F_{ha,2} & \dots & F_{ha,hb} \end{bmatrix}, \ Y = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1,hb} \\ Y_{21} & Y_{22} & \dots & Y_{2,hb} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{ha,1} & Y_{ha,2} & \dots & Y_{ha,hb} \end{bmatrix}. \tag{16}$$

one obtains from (14) the equation

$$\lambda Y_{k\ell} + \mu Y_{k\ell} = F_{k\ell} - \sum_{f=i+1}^{ha} M_{kf} Y_{f\ell} - \sum_{g=1}^{\ell-1} Y_{kg} N_{g\ell}, \tag{17}$$

$$k = 1, 2, \dots, ha,$$

$$\ell = 1, 2, \dots, hb.$$

Note the similarity between equations (17) and (13).

It follows from (17) that

$$Y_{k\ell} = \frac{1}{\lambda + \mu}\left(F_{k\ell} - \sum_{f=k+1}^{ha} M_{kf} Y_{f\ell} - \sum_{g=1}^{\ell-1} Y_{kg} N_{g\ell} \right), \tag{18}$$

$$k = 1, 2, \dots, ha,$$

$$\ell = 1, 2, \dots, hb.$$

Equation (18) allows to find directly the $k\ell$th block of the solution matrix Y.

Equations (18) are evaluated recursively for

$$Y_{ha,1}, Y_{ha-1,1}, \dots, Y_{11}, Y_{ha,2}, \dots, Y_{12}, \dots, Y_{ha,hb}, \dots, Y_{1,hb}.$$

The staircase Sylvester equation (18) is solved by the Algorithm 1.

Algorithm 1 Solution of a staircase Sylvester equation

Solution of the Sylvester equation $S_A \cdot Y + Y \cdot S_B = F$ where
$S_A = \lambda I + M$, $S_B = \mu I + N$ and the matrices M and N are
in staircase form with Weyr characteristics
$wa(S_A, \lambda) = (wa_1, wa_2, ..., wa_{ha})$ and
$wb(S_B, \mu) = (wb_1, wb_2, ..., wb_{hb})$, respectively.

function $Y = \text{sylv\_stairs}(S_A, S_B, F)$

Input: Matrices $S_A \in \mathbb{C}^{\kappa\times\kappa}$ and $S_B \in \mathbb{C}^{\nu\times\nu}$ with strictly
upper triangular parts in staircase form
Matrix $F \in \mathbb{C}^{\kappa\times\nu}$

Output: $Y \in \mathbb{C}^{\kappa\times\nu}$

for $\ell = 1 : hb$
  for $k = ha : -1 : 1$
    Set $W = 0_{wa_k, wb_\ell}$
    if $k + 1 \leq ha$
      for $f = k + 1 : ha$
        $W = W + M_{kf}Y_{f\ell}$
      end
    end
    Set $Z = 0_{wa_k, wb_\ell}$
    if $\ell > 1$
      for $g = 1 : \ell - 1$
        $Z = Z + Y_{kg}N_{g\ell}$
      end
    end
    if $\lambda + \mu \neq 0$
      $Y_{k\ell} = (F_{k\ell} - W - Z)/(\lambda + \mu)$
    else
      Error: The equation has no solution
    end
  end
end

Algorithm 2 Solution of the Sylvester equation by the Jordan-Schur method

Solution of the Sylvester equation $AX + XB = \mathrm{C}$

function $X = \text{sylv\_jsf}(A, B, C)$

Input: Matrices $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{m\times m}$ and $C \in \mathbb{C}^{n\times m}$

Output: $X \in \mathbb{C}^{n\times m}$

Reduce A to the Jordan-Schur form $J_{SA} = U^HAU$ with p blocks
on the diagonal each having eigenvalue λ_i with multiplicity κ_i,
$i = 1, 2, ..., p$
Reduce B to the Jordan-Schur form $J_{SB} = V^HAV$ with q blocks
on the diagonal each having eigenvalue μ_j with multiplicity ν_j,
$j = 1, 2, ..., q$
Transform $C \rightarrow U^HCV$
for $j = 1 : q$
  for $i = p : -1 : 1$
    Set $W = 0_{\kappa_i, \nu_j}$
    if $i < p$
      for $d = i + 1 : p$

$W = W + R_{id} X_{dj}$

end

end

Set $Z = 0_{\kappa_i, \nu_j}$

if $j > 1$

for $e = 1 : j - 1$

$Z = Z + X_{ie} T_{ej}$

end

end

Solve the staircase Sylvester equation

$S_{Aii} X_{ij} + X_{ij} S_{Bjj} = C_{ij} - W - Z$

using Algorithm 1

end

end

Transform back $X \rightarrow UXV^H$

Algorithm 1 resembles the back substitution in solving systems of linear systems of equations and can be considered as a version of the Bartels-Stewart algorithm [2] for solving the Sylvester equation. There is, however, one significant difference - the block $Y_{k\ell}$ of the solution Y is of dimension $wa_k \times wb_\ell$. In case of multiple eigenvalues with large values of the elements wa_k and wb_ℓ of the corresponding Weyr characteristics, the solution of the Sylvester equation by using the Jordan-Schur algorithm can be more efficient. If the elements of the Weyr characteristics are equal to one (this is the case of non-derogatory matrices), then the Jordan-Schur algorithm coincides with the Bartels-Stewart algorithm.

Equations (13) and (18) together constitute a two level algorithm for solving the Sylvester equation.

Using Algorithm 1, the equations (13) are solved recursively for

$$X_{p1}, X_{p-1,1}, \ldots, X_{11}, X_{p2}, \ldots, X_{12}, \ldots, X_{pq}, \ldots, X_{1q}.$$

The corresponding computations are carried out by Algorithm 2.

Example 2 Consider the solution of a Sylvester equation with the matrix A given in Example 1 and a 5th order matrix B with diagonal Jordan form

$$J_B = \mathrm{diag}\left(\mu_1, \mu_1, \mu_2, \mu_2, \mu_2\right), \ \lambda_i + \mu_j \neq 0$$

whose eigenvalue multiplicities are $\nu_1 = 2$ and $\nu_2 = 3$, respectively. Since the Weyr characteristics of B associated with μ_1 and μ_2 are $wb_1(B, \mu_1) = (2)$ and $wb_2(B, \mu_2) = (3)$, respectively, the blocks of the solution matrix are obtained in the following order

$$U^H XV = \begin{array}{r|c||c|} & wb_{11} = 2 & wb_{21} = 3 \\ wa_{11} = 2 & \mathbf{5} & \mathbf{10} \\ \hline wa_{12} = 1 & \mathbf{4} & \mathbf{9} \\ \hline\hline wa_{21} = 2 & \mathbf{3} & \mathbf{8} \\ \hline wa_{22} = 2 & \mathbf{2} & \mathbf{7} \\ \hline wa_{23} = 1 & \mathbf{1} & \mathbf{6} \end{array},$$

where the double lines correspond to the partition of blocks with sizes equal to the eigenvalue multiplicities, while the single lines correspond to the partition of blocks with sizes determined by the elements of the corresponding Weyr characteristics. Note that there are two blocks with dimension 1×2, two blocks with dimension 1×3 and three blocks with dimension 2×3. Thus, instead of solving $8 \cdot 5 = 40$ scalar equations as in the Bartels-Stewart algorithm, it is necessary to solve 7 block matrix equations.

It should be pointed out that the staircase Sylvester equations have independent application in solving problems involving the Jordan-Schur form, for instance in computing the matrix exponential [16].

4. Solution of the Lyapunov equation

The solution of the Lyapunov equation (2) is slightly more complicated than the solution of the Sylvester equation. In the given case

$$J_{SA} = U^H A U = \begin{bmatrix} S_{A11} & R_{12} & \cdots & R_{1p} \\ & S_{A22} & \cdots & R_{2p} \\ & & \ddots & \vdots \\ & & & S_{App} \end{bmatrix} \in \mathbb{C}^{n \times n},$$

$B = A$, $J_{SB} = J_{SA}$, $V = U$ and (13) obtains the form

$$S_{Aii}^H X_{ij} + X_{ij} S_{Ajj} = C_{ij} - \sum_{d=i+1}^{p} R_{di}^H X_{dj} - \sum_{e=1}^{j-1} X_{ie} R_{ej},$$

$$X_{ji} = X_{ij}^H, \tag{19}$$

$$j = 1, 2, \ldots, p,$$

$$i = j, j+1, \ldots, p.$$

In (19) one exploits the Hermitian property of the solution X. If $i = j$ one obtains the staircase Lyapunov equation

$$S_{Aii}^H X_{ii} + X_{ii} S_{Aii} = C_{ii} - \sum_{d=i+1}^{p} R_{di}^H X_{di} - \sum_{e=1}^{i-1} X_{ie} R_{ei}, \tag{20}$$

$$i = 1, 2, \ldots, p.$$

Equation (20) has a Hermitian solution for X_{ii}, i.e., $X_{ii}^H = X_{ii}$. In fact, since $X^H = X$ it follows that $X_{di} = X_{id}^H$, $X_{ei} = X_{ie}^H$ and since $C_{ii}^H = C_{ii}$, it is possible to show that the right hand side of (20) is Hermitian. This circumstance can be used to solve (20) more efficiently than the non-Hermitian ($i \neq j$) equation (19).

Thus the solution of (19) should be done by two different algorithms for $i = j$ and for $i \neq j$.

4.1 *Solution of Hermitian staircase Lyapunov equations*

Consider the staircase Lyapunov equation

$$S^H Y + YS = F \tag{21}$$

where

$$S = \begin{bmatrix} \lambda I_{w_1} & M_{12} & \dots & M_{1p} \\ & \lambda I_{w_2} & \dots & M_{2p} \\ & & \ddots & \vdots \\ & & & \lambda I_{w_h} \end{bmatrix} \in \mathbb{C}^{\kappa \times \kappa} \tag{22}$$

and $F^H = F$, $Y^H = Y$.

Equation (21) has a unique solution for Y if and only if $\lambda \neq 0$.

Representing the lower block-triangular parts of F and Y as

$$\begin{bmatrix} F_{11} & & & \\ F_{21} & F_{22} & & \\ \vdots & \vdots & \ddots & \\ F_{h1} & F_{h2} & \dots & F_{hh} \end{bmatrix}, \begin{bmatrix} Y_{11} & & & \\ Y_{21} & Y_{22} & & \\ \vdots & \vdots & \ddots & \\ Y_{h1} & Y_{h2} & \dots & Y_{hh} \end{bmatrix}$$

one obtains from (21) that

$$Y_{k\ell} = \frac{1}{2\lambda}\left(F_{k\ell} - \sum_{f=1}^{k-1} M_{fk}^H Y_{f\ell} - \sum_{g=1}^{\ell-1} Y_{kg} M_{g\ell} \right),$$

$$Y_{\ell k} = Y_{k\ell}^H, \tag{23}$$

$$\ell = 1, 2, \dots, h,$$

$$k = \ell, \ell+1, \dots, h.$$

For $k = \ell$ it is fulfilled that $Y_{kk}^H = Y_{kk}$.

Algorithm 3 Solution of a staircase Lyapunov equation

Solution of the Lyapunov equation $S^H Y + YS = F$ where
$S = \lambda I + M$ and the matrix M is in staircase form
with a Weyr characteristic $w(S, \lambda) = (w_1, w_2, ..., w_h)$.

function Y = lyap_stairs(S, F)

Input: Matrix $S \in \mathbb{C}^{\kappa \times \kappa}$ with strictly
upper triangular part in staircase form
Matrix $F \in \mathbb{C}^{\kappa \times \kappa}$, $F^H = F$

Output: $Y \in \mathbb{C}^{\kappa \times \kappa}$, $Y^H = Y$

for $\ell = 1 : h$
  for $k = \ell : h$
    Set $W = 0_{w_k, w_\ell}$
    if $k > 1$
      for $f = 1 : k - 1$
        $W = W + M_{fk}^H Y_{f\ell}$
      end
    end
    Set $Z = 0_{w_k, w_\ell}$

if $\ell > 1$
for $g = 1 : \ell - 1$
$Z = Z + Y_{kg}M_{g\ell}$
end
end
if $\lambda \neq 0$
$Y_{k\ell} = (F_{k\ell} - W - Z)/(2\lambda)$
if $k \neq \ell$
$Y_{\ell k} = Y_{k\ell}^H$
end
else
Error: The equation has no solution
end
end
end

Algorithm 4 Solution of a staircase Sylvester equation with a transposed matrix
Solution of the Sylvester equation $S_A^H \cdot Y + Y \cdot S_B = F$ where
$S_A = \lambda I + M$, $S_B = \mu I + N$ and the matrices M and N are
in staircase form with Weyr characteristics
$wa(S_A, \lambda) = (wa_1, wa_2, ..., wa_{ha})$ and
$wb(S_B, \mu) = (wb_1, wb_2, ..., wb_{hb})$, respectively.
function Y = sylv_transp(S_A, S_B, F)
Input: Matrices $S_A \in \mathbb{C}^{\kappa\times\kappa}$ and $SB \in \mathbb{C}^{\nu\times\nu}$ with strictly
upper triangular parts in staircase form
Matrix $F \in \mathbb{C}^{\kappa\times\nu}$
Output: $Y \in \mathbb{C}^{\kappa\times\nu}$
for $\ell = 1 : hb$
for $k = 1 : ha$
Set $W = 0_{wa_k, wb_\ell}$
if $k > 1$
for $f = 1 : k - 1$
$W = W + M_{fk}^H Y_{f\ell}$
end
end
Set $Z = 0_{wa_k, wb_\ell}$
if $\ell > 1$
for $g = 1 : \ell - 1$
$Z = Z + Y_{kg}N_{g\ell}$
end
end
if $\lambda + \mu \neq 0$
$Y_{k\ell} = (F_{k\ell} - W - Z)/(\lambda + \mu)$
else
Error: The equation has no solution
end
end
end

Algorithm 5 Solution of the Lyapunov equation by the Jordan-Schur method
Solution of the Lyapunov equation $A^H X + XA = C$
function X = lyap_jsf(A, C)

Input: Matrices $A \in \mathbb{C}^{n\times n}$ and $C \in \mathbb{C}^{n\times n}$, $C^H = C$
Output: $X \in \mathbb{C}^{n\times n}$, $X^H = X$
Reduce A to the Jordan-Schur form $J_{SA} = U^H AU$ with p blocks
on the diagonal each having eigenvalue λ_i with multiplicity κ_i,
$i = 1, 2, ..., p$
Transform $C \rightarrow U^H CU$
for $j = 1 : p$
  for $i = j : p$
    Set $W = 0_{\kappa_i,\kappa_j}$
    if $i < p$
      for $d = i + 1 : p$
        $W = W + R_{di}^H X_{dj}$
      end
    end
    Set $Z = 0_{\kappa_i,\kappa_j}$
    if $j > 1$
      for $e = 1 : j - 1$
        $Z = Z + X_{ie} R_{ej}$
      end
    end
    if $i = j$
      Solve the staircase Lyapunov equation
        $S_{ii}^H X_{ii} + X_{ii} S_{ii} = C_{ii} - W - Z$
      using Algorithm 3
    else
      Solve the staircase Sylvester equation
        $S_{ii}^H X_{ij} + X_{ij} S_{jj} = C_{ij} - W - Z$
      using Algorithm 4
      $X_{ji} = X_{ij}^H$
    end
  end
end
Transform back $X \rightarrow UXU^H$

Equations (23) are evaluated consecutively for the subdiagonal blocks

$$Y_{11}, Y_{21}, \ldots, Y_{h1}, Y_{22}, \ldots, Y_{h2}, \ldots, Y_{hh}$$

and the superdiagonal blocks are found by Hermitian transposition of the corresponding subdiagonal blocks.

The staircase Lyapunov equation (23) is solved by the Algorithm 3.

4.2 *Solution of the equations for off-diagonal blocks*

The off-diagonal ($i \neq j$) equations (20) have the form of a Sylvester equation with a transposed matrix A,

$$S_A^H \cdot Y + Y \cdot S_B = F \tag{24}$$

where S_A and S_B are the same as in (15). Partitioning F and Y as in (16), the equation (24) is represented as

$$\lambda Y_{k\ell} + \mu Y_{k\ell} = F_{k\ell} - \sum_{f=1}^{k-1} M_{fk}^H Y_{f\ell} - \sum_{g=1}^{\ell-1} Y_{kg} M_{g\ell}, \tag{25}$$

$$k = 1, 2, \ldots, ha,$$

$$\ell = 1, 2, \ldots, hb$$

Note that in the given case both λ and μ are eigenvalues of A.
As a result from (25) one obtains

$$Y_{kl} = \frac{1}{\lambda + \mu}\left(F_{k\ell} - \sum_{f=k+1}^{ha} M_{fk}^H Y_{f\ell} - \sum_{g=1}^{\ell-1} Y_{kg} N_{g\ell} \right), \tag{26}$$

$$k = 1, 2, \ldots, ha$$

$$\ell = 1, 2, \ldots, hb,$$

Equations (26) are similar to equations (17) but are solved in the order

$$Y_{11}, Y_{21}, \ldots, Y_{ha,1}, Y_{12}, Y_{22}, \ldots, Y_{ha,2}, \ldots, Y_{1,hb}, Y_{2,hb}, \ldots, Y_{ha,hb}.$$

Equations (26) are solved by Algorithm 4.
Using Algorithms 3 and 4, the Lyapunov equation can be solved by the Algorithm 5.

5. Numerical considerations

The basic steps of the algorithm proposed are the reduction to Jordan-Schur form and the solution of the transformed Sylvester or Lyapunov equation.

Consider first the reduction to Jordan-Schur form by the algorithm of Kågström and Ruhe. As mentioned earlier, this reduction is numerically stable due to the implementation of unitary transformations. However, it should be emphasized that the numerical stability doesn't guarantee that the clustering of the numerically multiple eigenvalues and the determination of the sizes of the Jordan blocks is always done in the best way. The extensive experiments with the algorithm of Kågström and Ruhe show that it works reliably except in case of very ill-conditioned system of eigenvectors and principal vectors when any clustering method will have serious difficulties in grouping the eigenvalues. Such difficulties manifest themselves by the large backward error $||UJ_SU^H - A||_F/||A||_F$ and can be removed by adjusting properly the tolerances *ein* and *tol*. The tolerances *ein* and *tol* play the role of regularization parameters which allow for achieving a Jordan structure by the use of well-conditioned similarity transformation. The tolerance *ein* affects the clustering of the eigenvalues in groups of numerically multiple eigenvalues while the tolerance *tol* in its turn affects the determination of the dimensions of the null spaces and consequently the Weyr characteristics of the associated eigenvalues. The choice of appropriate tolerances can be done so that to minimize the distance between $\hat{A}_k = U \cdot J_S \cdot U^H$ and A or to choose a specific Jordan structure.

The following example illustrates the influence of *ein* and *tol* on the numerical determination of the structure of Jordan-Schur form.

Example 3 Consider the matrix

$$A = \begin{bmatrix} -111 & -240 & -572 & 1752 & -4272 \\ -149 & -335 & -778 & 2340 & -5688 \\ 132 & 288 & 657 & -1980 & 4744 \\ -38 & -84 & -201 & 613 & -1504 \\ -22 & -48 & -112 & 340 & -823 \end{bmatrix}$$

which is obtained as $A = QJQ^{-1}$, where

$$J = \left(\begin{bmatrix} 1 & 1 \\ & 1 \end{bmatrix}, 1, \begin{bmatrix} -1 & 1 \\ & -1 \end{bmatrix} \right)$$

and

$$Q = \begin{bmatrix} 56 & 187 & 64 & -8 & 6 \\ 109 & 261 & 90 & -16 & 6 \\ -100 & -234 & -80 & 14 & -5 \\ 20 & 64 & 22 & -3 & 2 \\ 14 & 38 & 13 & -2 & 1 \end{bmatrix}.$$

The matrix of the similarity transformation into Jordan form is relatively ill-conditioned with condition number $\text{cond}(Q) = 1.34 \cdot 10^5$. The Segre characteristics associated with the eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -1$ are $s_1 = (2, 1)$ and $s_2 = (2)$, respectively.

Table 1. Jordan-Schur forms of A computed for different tolerances

k	ein, tol	Computed Segre charactersitics	dist $(\hat{A}_k, A)$
1	10^{-9}	(2, 1), (2)	$7.37 \cdot 10^{-16}$
2	10^{-8}	(4, 1)	$5.33 \cdot 10^{-7}$
3	10^{-7}	(3, 2)	$5.29 \cdot 10^{-7}$
4	10^{-6}	(3, 1, 1)	$3.07 \cdot 10^{-5}$
5	10^{-5}	(2, 1, 1, 1)	$3.70 \cdot 10^{-16}$
6	1	(1, 1, 1, 1, 1)	$5.79 \cdot 10^{-16}$

In Table 1 we give the Segre characteristics of the six matrices $\hat{A}_k = U\hat{J}_S U^H$ corresponding to the Jordan-Schur forms $\hat{J}_{Sk}$, $k = 1, 2, ..., 6$ computed for different tolerances ein and tol. The last column contains the relative distances

$$\text{dist}(\hat{A}_k, A) = \frac{\| \hat{A}_k - A \|_F}{\| A \|_F}, \, k = 1, 2, \ldots, 6$$

between the matrices $\hat{A}_k$ and the original matrix A in the space $\mathbb{C}^{25}$. The right Segre characteristics are determined for tolerances $ein = tol = 10^{-9}$. In case of well-conditioned eigensystem one can use tolerances which are of order $n^2\mathbf{u}\|A\|_F$, where **u** is the roundoff unit of the used floating point arithmetic.

It should be emphasized that the using of the algorithm of Kågström and Ruhe for reducing the equation matrices into Jordan-Schur form is not indispensable - for this purpose it is also possible to use other algorithms. For instance, an alternative method for reduction to JordanSchur form is to implement the function guptri from MCS Toolbox [17] which determines automatically appropriate tolerances *ein* and *tol*. This function is based on the Generalized UPper TRIangular (GUPTRI) algorithm of Demmel and Kågström [18, 19] intended for determining the Kronecker structure of matrix pencils. (The Kronecker structure represents a generalization of the Jordan structure of a matrix to the case of two matrices).

The accuracy of the solution of the transformed Sylvester and Lyapunov equations depends on two factors: the conditioning of the corresponding equation and the numerical properties of the method used to obtain the solution. Consider in brief the influence of these factors on the solution of Sylvester and Lyapunov equations obtained by the Jordan-Schur method.

The sensitivity of the solution of the Sylvester equation to perturbations in the data is analyzed in [20], see also [21, Ch. 16]. Let the Sylvester equation be

$$AX + XB = C \tag{27}$$

where $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{m\times m}$, $C \in \mathbb{C}^{n\times m}$ and let $\Omega = I_m \otimes A + B^T \otimes I_n \mathbb{C}^{nm\times nm}$. Then the relative change $\|\Delta X\|_F / \|X\|_F$ in the solution of this equation due to small relative perturbations $\Delta A/\|A\|_F$, $\Delta B/\|B\|_F$ and $\Delta C/\|C\|_F$, is given by [21, Ch. 16]

$$\|\Delta X\|_F / \|X\|_F \leq \sqrt{3}\,\mathrm{cond}_{\mathrm{sylv}}\eta \tag{28}$$

where

$$\mathrm{cond}_{\mathrm{sylv}} = \|\Omega^{-1}\left[\|A\|_F (X^T \otimes I_n), \|B\|_F (I_m \otimes X), -\|C\|_F I_{nm}\right]\|_2 / \|X\|_F$$

is the relative condition number of the Sylvester equation and

$$\eta = \max\{\|\Delta A\|_F / \|A\|_F, \|\Delta B\|_F / \|B\|_F, \|\Delta C\|_F / \|C\|_F\}$$

is the maximum relative perturbation in the data. The size of the condition number is closely related to the separation of the matrices A and $-B$ [22],

$$\mathrm{sep}(A, -B) = \min_{X\neq 0} \frac{\|AX + XB\|_F}{\|X\|_F}.$$

If this separation is small, then the condition number of the Sylvester equation is large which follows from the fact that

$$\|\Omega^{-1}\|_2 = 1/\mathrm{sep}(A, -B).$$

In respect to the Lyapunov equation one has the similar bound

$$\|\Delta X\|_F / \|X\|_F \leq \sqrt{3}\,\mathrm{cond}_{\mathrm{lyap}}\eta, \tag{29}$$

where

$$\mathrm{cond}_{\mathrm{lyap}} = \| \Omega^{-1} \left[\| A\|_F ((I_n \otimes X) + (X \otimes I_n)\Pi), -\| C\|_F I_{n^2} \right] \|_2 / \| X\|_F$$

is the relative condition number of the Lyapunov equation,

$$\Omega = I_m \otimes A^T + A^T \otimes I_n,$$

Π is the vec-permutation matrix defined by the property $\mathrm{vec}(A^T) = \Pi\mathrm{vec}(A)$ [23, p. 21] and

$$\eta = \max\{\| \Delta A\|_F / \| A\|_F, \| \Delta C\|_F / \| C\|_F\}.$$

The second factor influencing the solution accuracy of Sylvester and Lyapunov equations is the numerical behavior of the Jordan-Schur algorithm. Since the Jordan-Schur form is a type of Schur form and the equations solved are similar to the equations solved in the Bartels-Stewart algorithm, both methods have similar numerical properties. According to the analysis done in [20], [21, Ch. 16], the relative residual corresponding to the computed solution,

$$\frac{\| C - (A\hat{X} + \hat{X}B)\|_F}{(\| A\|_F + \| B\|_F)\| \hat{X}\|_F}$$

is always small and is of order $\mathbf{u}$. Nevertheless, from the analysis follows that the Jordan-Schur algorithm, like the Bartels-Stewart algorithm in its standard implementation, is only conditionally backward stable. This means that the backward error corresponding to the computed solution $\hat{X}$ can be large in some cases determining a large forward error in the solution.

An estimate close to the forward error of the solution can be obtained by the practical error bound proposed in [21, Ch. 16]. Let the computed residual of a computed solution be represented as

$$\hat{R} = fl(C - (fl(A\hat{X} + fl(\hat{X}B))) = R + \Delta R,$$

$$| \Delta R | \leq \mathbf{u} | C | + \gamma_A | A || \hat{X} | + \gamma_B | \hat{X} || B | =: R_u,$$

where

$$\gamma_A = (n+2)\mathbf{u} / (1 - (n+2)\mathbf{u}), \;\; \gamma_B = (m+2)\mathbf{u} / (1 - (m+2)\mathbf{u})$$

and $\mathbf{u}$ is the roundoff unit.

Then the relative error in the computed solution satisfies [21, Eq. (16.29)]

$$\frac{\| \hat{X} - X\|}{\| \hat{X}\|} \leq \frac{\| | \Omega^{-1} | (| \mathrm{vec}(\hat{R}) | + \mathrm{vec}(R_u))\|}{\| \hat{X}\|}, \tag{30}$$

where $\|X\| := \max_{i,j}|x_{ij}|$.

To avoid the use of large matrices in computing the condition numbers $\mathrm{cond}_{\mathrm{sylv}}$, $\mathrm{cond}_{\mathrm{lyap}}$ and the bound (30), one can use the technique described in [21, Sect. 16.4].

The numerical experiments presented in the next section confirm that the Jordan-Schur algorithm has an accuracy which is close to the accuracy of the Bartels-Stewart algorithm.

Consider now the computational work associated with the Jordan-Schur algorithm. Since in both Bartels-Stewart and Jordan-Schur algorithms the first step is the reduction of the matrices A and B to Schur form, it is appropriate to consider only the volume of computational work done in addition to the reduction into Schur form. For the Bartels-Stewart algorithm, the solution of the Sylvester equation with matrices in Schur form requires

$$\frac{1}{2}\left[n(n-1)m+m(m-1)n\right]$$

floating point multiplications and additions or flam as called in [24, Ch. 2], where 1 flam is 1 floating point multiplication and 1 floating point addition. For the Jordan-Schur algorithm, under the assumptions of matrices with one eigenvalue ($p = 1$, $\kappa_1 = n$; $q = 1$, $\nu_1 = m$) and equal elements of the Weyr characteristics ($wa_i = n/ha$, $wb_j = m/hb$), the necessary computational work for solving the staircase Sylvester equation (14) is

$$\frac{1}{2}\left[(ha-1)\frac{n^2 m}{ha}+(hb-1)\frac{nm^2}{hb}\right]\text{flam.}$$

For large indices of nilpotency ($ha = n$, $hb = m$) where the elements of the Weyr characteristics are equal to one (this is the case of non-derogatory but defective matrices with one Jordan block), this volume of work is exactly the same as for Bartels-Stewart algorithm. However, with the decreasing of the indices of nilpotency the necessary computational work tends to zero while for the Bartels-Stewart algorithm it remains the same. For the reduction to Jordan-Schur form one should take also into account the sorting of the eigenvalues, the clustering of multiple eigenvalues and the reduction of the diagonal blocks to staircase form of the matrices A and B which require at most

$$6n^3+2\frac{1}{3}n^3 ha+6m^3+2\frac{1}{3}m^3 hb \text{ flam.}$$

For $ha = n$, $hb = m$ this volume of work is of order

$$2\frac{1}{3}n^4+6n^3+2\frac{1}{3}m^4+6m^3 \text{ flam.}$$

and for $ha = 1$, $hb = 1$ (derogatory but non-defective matrices with scalar Jordan blocks) it is approximately

$$8\frac{1}{3}n^3+8\frac{1}{3}m^3 \text{ flam.}$$

Thus the Jordan-Schur algorithm can be competitive with the Bartels-Stewart algorithm when the Jordan-Schur forms of A and B are already available from other computations and the indices of nilpotency (i.e., the sizes of the Jordan blocks) of these matrices are small. Also, the Jordan-Schur algorithm can be more efficient in the case when the Sylvester equation is solved for the same matrices A and B but for several right-hand sides.

The number of computational operations for the solution of the Hermitian staircase Lyapunov equation (21) under the same assumptions as above ($p = 1$, $\kappa_1 = n$, $w = n/h$), is

$$\frac{1}{2}(h^2-1)\frac{n^3}{h^2} \text{ flam.}$$

For large indices of nilpotency ($h = n$) this volume is approximately equal to $\frac{1}{2}n^3$ flam.

The disadvantage in the computational complexity of the proposed algorithm could be overcome in the future by using alternative algorithms for finding the Jordan-Schur form which are more efficient than the algorithm of Kågström and Ruhe.

To summarize, the Jordan-Schur algorithm for solving Sylvester and Lyapunov equations can be considered as a generalization of the Bartel-Stewart algorithm which in case of distinct eigenvalues performs exactly as the Bartels-Stewart algorithm since in such case the JordanSchur form coincides with the Schur form. The Jordan-Schur algorithm has some advantage over the Bartels-Stewart algorithm in cases of matrices with multiple eigenvalues participating in Weyr blocks of large size.

6. Numerical experiments

The numerical experiments presented in this paper are done with MATLAB® Version 9.9 (R2020b) [25] using IEEE double precision arithmetic with roundoff unit $\mathbf{u} \approx 1.11 \cdot 10^{-16}$ on a machine equipped with an Intel i7-2670QM CPU running at 2.20 GHz and with 8 GB of RAM. The first three experiments involve solution of the Sylvester equation, while the last three are devoted to the solution of Lyapunov equation. The algorithms proposed are implemented as M-files for MATLAB®. The reduction to Jordan-Schur form is done using a part of the algorithm of Kågström and Ruhe [9, 10] which is translated almost literally to the MATLAB® language. Note that the information about the Jordan structure of the matrices A and B which is known in advance of the experiments, is not use in the reduction into Jordan-Schur form. The Bartels-Stewart algorithm is used as implemented by the function sylvsol from the Matrix Function Toolbox [26] of N. J. Higham. In the experiments with the Sylvester equation the exact solution is chosen as a random matrix while in the experiments with Lyapunov equation it is chosen as a matrix with unit entries. (The solution of the Lyapunov equation should be a Hermitian matrix.) The right hand side matrix is obtained from $C = AX + XB$ in the case of Sylvester equation or $C = A^H X + XA$ in the case of Lyapunov equation.

6.1 *Solving the Sylvester equation*

Experiment 1 The first experiment demonstrates the algorithm performance for Sylvester equations whose matrices A and B are defective but non-derogatory. The matrices A and B are taken as

$$A = Q_A J_A Q_A^{-1}, B = Q_B J_B Q_B^{-1},$$

where

$$J_A = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & & 1 \\ & & & & -1 \end{bmatrix}, \ J_B = \begin{bmatrix} 2 & 1 & & & \\ & 2 & 1 & & \\ & & \ddots & \ddots & \\ & & & & 1 \\ & & & & 2 \end{bmatrix}.$$

The transformation matrices Q_A and Q_B are constructed as described in [27],

$$Q_A = H_2 Y_A H_1, \ Q_B = M_2 Y_B M_1,$$

$$H_1 = I_n - 2uu^T / n, \ M_1 = I_m - 2ww^T / m,$$

$$H_2 = I_n - 2vv^T / n, \ M_2 = I_m - 2zz^T / m,$$

$$u = [1, 1, 1, \ldots, 1]^T, \ w = [1, 1, 1, \ldots, 1]^T,$$

$$v = [1, -1, 1, \ldots, (-1)^{n-1}]^T, \; z = [1, -1, 1, \ldots, (-1)^{m-1}]^T,$$

$$Y_A = \operatorname{diag}(1, \sigma_A, \sigma_A^2, \ldots, \sigma_A^{n-1}), \; Y_B = \operatorname{diag}(1, \sigma_B, \sigma_B^2, \ldots, \sigma_B^{m-1})$$

where H_1, H_2, M_1, M_2 are orthogonal elementary reflections. The condition numbers of Q_A and Q_B with respect to inversion are controlled by the variables σ_A and σ_B, respectively. The increasing of these variables allows to increase the condition number of the Sylvester equation thus revealing the numerical properties of the solution method.

The experiment is performed for $n = 2, 4, 6, \ldots, 100$ and $m = 10$. The parameters σ_A and σ_B are chosen as 1.05 and 1.2, respectively, so that the relative error in the solution is less than 1. With the increasing of n the condition number of the Sylvester equation increases.

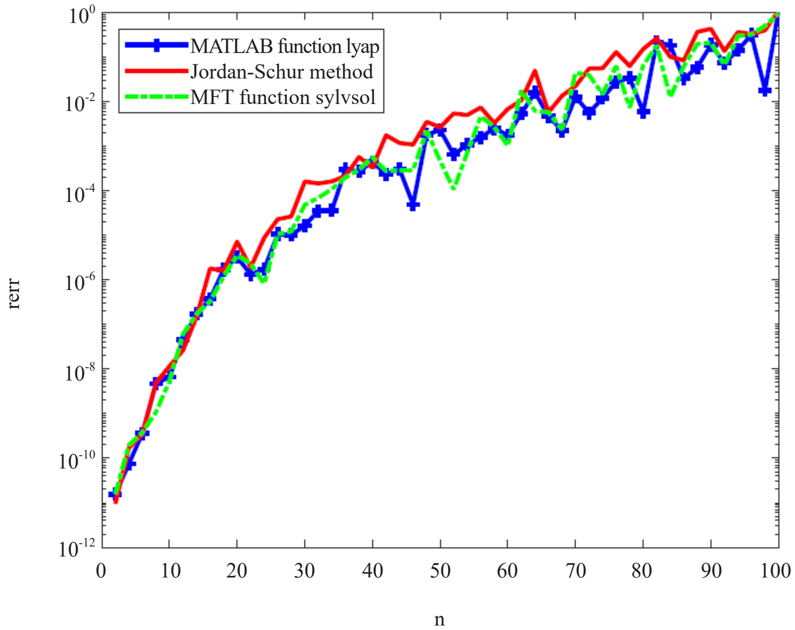

Figure 1. Relative errors for Experiment 1

In Figure 1 we show the relative error in the computed solution $\hat{X}$,

$$rerr = \frac{\| \hat{X} - X \|_F}{\| X \|_F}$$

as a function of n, obtained for three algorithms - the function lyap of MATLAB, the JordanSchur algorithm and the function sylvsol from the Matrix Function Toolbox. The three algorithms produce solutions of comparable accuracy.

Experiment 2 This experiment shows good numerical behavior of the function sylsol and the Jordan-Scur algorithm, and large errors in the solution of the Sylvester equation, produced by the function lyap. In the given case

$$A = Q_A J_A Q^{-1}, \; B = Q_B J_B Q_B^{-1},$$

where

$$J_A = \mathrm{diag}(\lambda, \lambda, 3, 3, 3),\ J_B = \begin{bmatrix} -2 & 1 & \\ & -2 & \\ & & -2 \end{bmatrix},$$

$$Q_A = Y_A \cdot (Y_A^T)^{-1},\ Y_A = \begin{bmatrix} 1 & -1 & 1 & -2 & 10 \\ & 1 & -1 & 1 & 2 \\ & & 1 & -1 & -1 \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix},$$

$$Q_B = H_2 Y_B H_1,$$

$$H_1 = I_3 - 2uu^T / 3,\ u = [1,1,1]^T,$$

$$H_2 = I_3 - 2vv^T / 3,\ v = [1,-1,1]^T,$$

$$Y_B = \mathrm{diag}(1, \sigma, \sigma^2),$$

and the eigenvalue λ is a variable parameter.

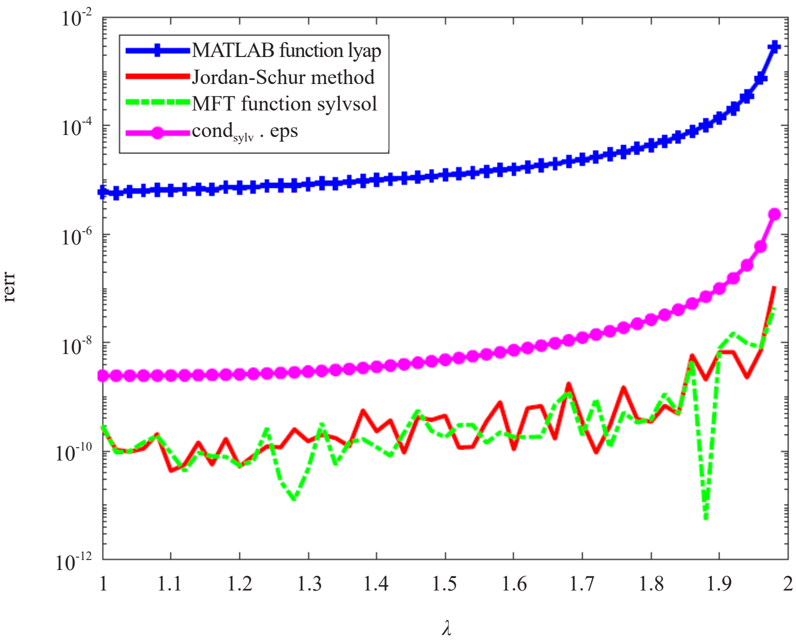

Figure 2. Relative errors for Experiment 2

In Figure 2 we show the relative errors in the solution of the Sylvester equation $AX + XB = C$ for 50 values of λ from 1.0 to 1.98 obtained for three methods (for $\tau = 2$ the equation has no solution). The Bartels-Stewart algorithm as implemented by the function solvsyl and the Jordan-Schur algorithm have close numerical behavior while the

MATLAB® function lyap produces a solution with large errors.

Experiment 3 This experiment illustrates the properties of the algorithms proposed in case of matrices with multiple eigenvalues and large elements of the Weyr characteristic. The matrix A and B are of even order and are taken as

$$A = Q_A J_A Q_A^{-1},\ B = Q_B J_B Q_B^{-1},$$

where the Jordan forms

$$J_A = \begin{bmatrix} \lambda & 1 & & & & & \\ & \lambda & & & & & \\ & & \lambda & 1 & & & \\ & & & \lambda & \ddots & & \\ & & & & \ddots & & \\ & & & & & \lambda & 1 \\ & & & & & & \lambda \end{bmatrix},\ J_B = \begin{bmatrix} \mu & 1 & & & & & \\ & \mu & & & & & \\ & & \mu & 1 & & & \\ & & & \mu & \ddots & & \\ & & & & \ddots & & \\ & & & & & \mu & 1 \\ & & & & & & \mu \end{bmatrix}$$

consist of 2 × 2 blocks with equal eigenvalues and the nonsingular transformation matrix Q_A and Q_B are chosen as in Experiment 1. In the given case we take $\lambda = 3.0$ and $\mu = -2$. The parameters controlling the conditioning of Q_A and Q_B are chosen $\sigma_A = 1.1$ and $\sigma_B = 1.5$, respectively. Note that the Segre characteristic of A and B associated with the corresponding eigenvalues are equal to (2, 2, 2, ..., 2) and the Weyr characteristics are equal to (n/2, n/2) and (m/2, m/2), respectively.

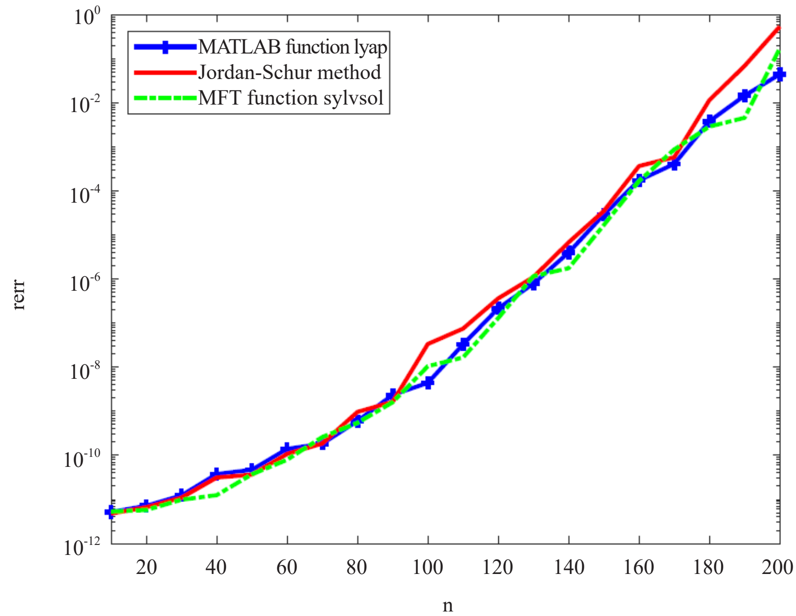

Figure 3. Relative errors for Experiment 3

In Figure 3 we show the relative errors in the solution of the Sylvester equation for n = 10, 20, ..., 200 and m = 20 obtained by the function lyap, the Jordan-Schur algorithm and the function sylvsol. The three methods produce close results with errors increasing with n.

Table 2. Execution time for Experiment 3 (transformed equations)

n	sylvsol	Jordan-Schur	n	sylvsol	Jordan-Schur
10	$4.3 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	110	$1.2 \cdot 10^{-2}$	$2.6 \cdot 10^{-3}$
20	$4.4 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	120	$1.5 \cdot 10^{-2}$	$3.8 \cdot 10^{-3}$
30	$4.1 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	130	$1.8 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$
40	$3.2 \cdot 10^{-3}$	$9.4 \cdot 10^{-4}$	140	$1.6 \cdot 10^{-2}$	$3.9 \cdot 10^{-3}$
50	$5.7 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	150	$1.9 \cdot 10^{-2}$	$4.8 \cdot 10^{-3}$
60	$4.9 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$	160	$2.3 \cdot 10^{-2}$	$5.7 \cdot 10^{-3}$
70	$6.1 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$	170	$1.9 \cdot 10^{-2}$	$5.4 \cdot 10^{-3}$
80	$1.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$	180	$2.0 \cdot 10^{-2}$	$4.2 \cdot 10^{-3}$
90	$1.0 \cdot 10^{-2}$	$2.4 \cdot 10^{-3}$	190	$2.2 \cdot 10^{-2}$	$4.4 \cdot 10^{-3}$
100	$1.5 \cdot 10^{-2}$	$2.8 \cdot 10^{-3}$	200	$3.0 \cdot 10^{-2}$	$6.0 \cdot 10^{-3}$

In Table 2 we present the computational time for solving the Sylvester equations with matrices A and B preliminary transformed to Schur form (in the case of using the function sylvsol) or to Jordan-Schur form (in the case of using the Jordan-Schur algorithm). The data presented in the table shows that the solution of the given Sylvester equations in which the matrices A and B are in Jordan-Schur form is done from 3 to 5 times faster than the solution of the same equations in which A and B are in the Schur form. These goods results for the Jordan-Schur algorithms are due to the simple structure of the Jordan-Schur forms

$$J_{SA} = \begin{bmatrix} \lambda I_{n/2} & M_{12} \\ & \lambda I_{n/2} \end{bmatrix}, \; J_{SB} = \begin{bmatrix} \lambda I_{m/2} & N_{12} \\ & \lambda I_{m/2} \end{bmatrix}$$

in the given case, which facilities very much the solution of the equations. Note that according to the estimates, given in sect. 5, the solution of the full (untransformed) Sylvester equations by the Jordan-Schur algorithms requires much more time than the Bartels-Stewart algorithm due to the reduction to Jordan-Schur form which is significantly slower than the reduction to Schur form.

6.2 *Solving the Lyapunov equation*

Experiment 4 In this experiment the matrix A is chosen in the same way as the one shown in Experiment 1. In the given case the parameter σ_A is chosen equal to 1.15.

In Figure 4 we show the relative errors in the solution of the Lyapunov equation $A^H + XA = C$ for n = 2, 4, 6, ..., 100 for the three methods used in the previous experiments. The three methods produce solutions of nearly the same accuracy.

Experiment 5 In this experiment

$$A = QJ_AQ^{-1}, \; Q = Y_A \cdot (Y_A^T)^{-1}$$

where

$$J_A = \left[\begin{array}{cc|ccc} -2 & 1 & & & \\ & -2 & & & \\ \hline & & -1 & 1 & \\ & & & -1 & 1 \\ & & & & -1 \end{array}\right], \; Y_A = \begin{bmatrix} 1 & -1 & 1 & -2 & \tau \\ & 1 & -1 & 1 & 2 \\ & & 1 & -1 & -1 \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix},$$

and τ is a parameter. The increasing of τ leads to an increasing of the condition number in respect to the inversion of Y_A and hence of Q.

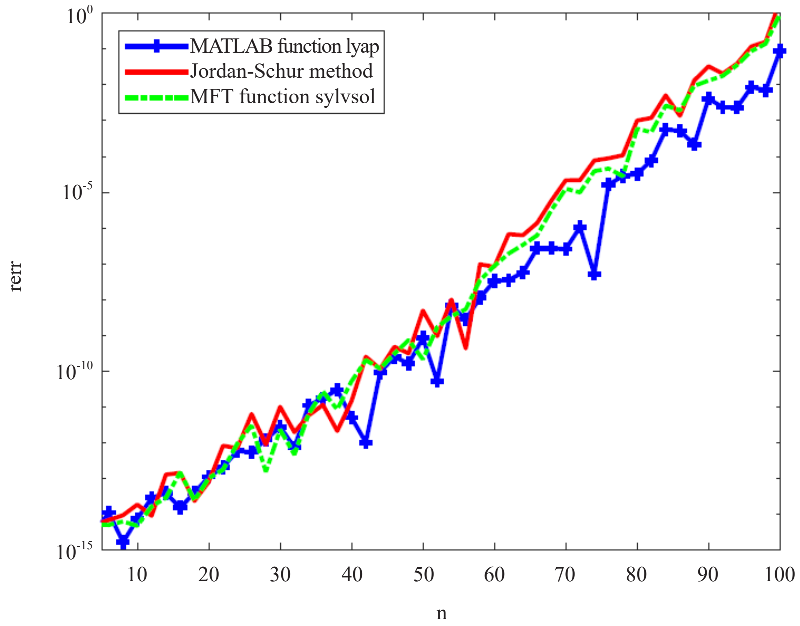

Figure 4. Relative errors for Experiment 4

In Figure 5 we show the relative errors for 51 values of τ from 0 to 5.25 for the three methods along with the forward accuracy estimate $\text{cond}_{lyap} \cdot eps$. All methods produce solutions whose error is less than the accuracy estimate.

Experiment 6 In the last experiment we investigate the accuracy of the solution of Lyapunov equation when an eigenvalue of the matrix A is approaching 0. In this case one takes

$$A = QJ_AQ^{-1}, \; Q = Y_A \cdot (Y_A^T)^{-1},$$

where

$$J_A = \left[\begin{array}{cc|ccc} \lambda & 1 & & & \\ & -2 & & & \\ \hline & & -1 & 1 & \\ & & & -1 & 1 \\ & & & & -1 \end{array}\right], \; Y_A = \begin{bmatrix} 1 & -1 & 1 & -2 & 5 \\ & 1 & -1 & 1 & 2 \\ & & 1 & -1 & -1 \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix}$$

and $\lambda = -10^{-p}$ where p varies from 1.0 to 7.4. With the increasing of p the conditioning of Lyapunov equation worsens and the relative error of the solution increases. As one can see from Figure 6, all three methods produce solutions which are more accurate than it is predicted by the condition number based accuracy estimate.

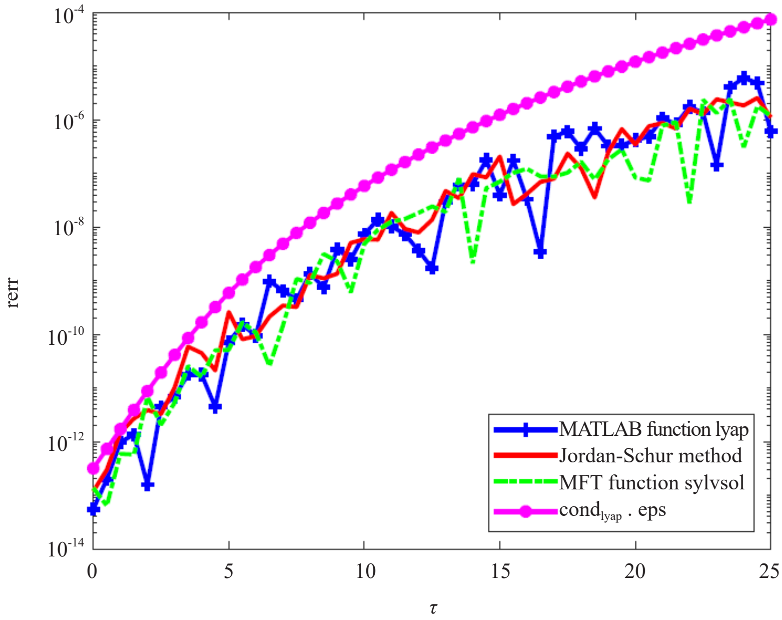

Figure 5. Relative errors for Experiment 5

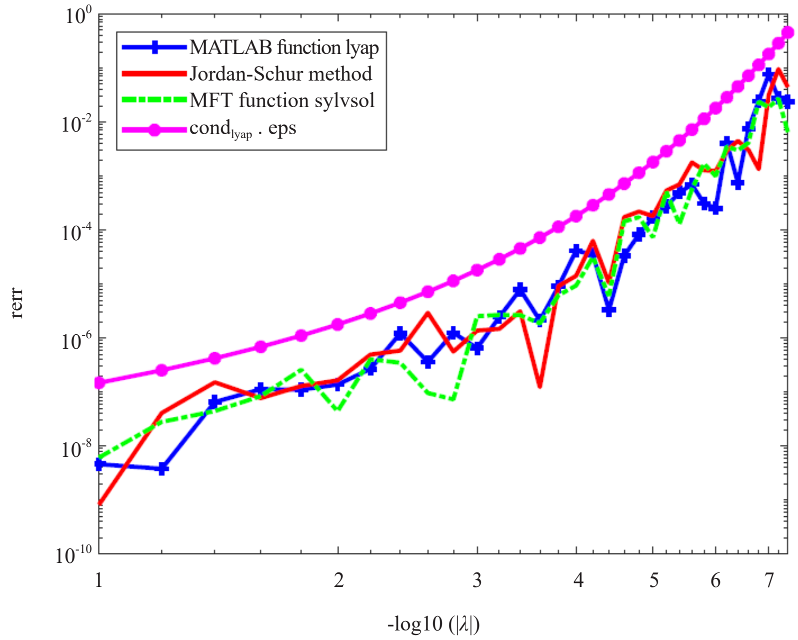

Figure 6. Relative errors for Experiment 6

7. Conclusions

The Jordan-Schur algorithm possesses the same accuracy and can be more efficient than the Bartels-Stwerat algorithm in respect of the computational work in some cases. With the coming of more reliable numerical methods for the reduction to Jordan-Schur form, this method may become an efficient alternative to the Bartels-Stewart algorithm. The proposed algorithm can be extended to the solution of the Stein equation $AXB - X = C$ and the discrete-time Lyapunov equation $A^H XA - X = C$.

Acknowledgments

The author expresses his gratitude to the peer reviewers whose valuable suggestions helped to improve the paper presentation.

Conflicts of interest

The author has no competing interests to declare that are relevant to the content of this article.

References

[1] Simoncini V. Computational methods for linear matrix equations. *SIAM Review*. 2016; 58: 377-441.
[2] Bartels RH, Stewart GW. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*. 1972; 15: 820-826. Available from: doi: 10.1145/361573.361582.
[3] Jonsson I, Kågström B. Recursive blocked algorithms for solving triangular systems-Part I: One-sided and coupled Sylvester-type matrix equations. *ACM Transactions on Mathematical Software*. 2002; 28: 392-415. Available from: doi: 10.1145/592843.592845.
[4] Kågström B, Wiberg P. Extracting partial canonical structure for large scale eigenvalue problems. *Numerical Algorithms*. 2000; 24: 195-237. Available from: doi: 10.1023/A:1019153528915.
[5] Elmroth E, Johansson P, Kågström B. Bounds for the distance between nearby Jordan and Kronecker structures in a closure hierarchy. *Journal of Mathematical Sciences*. 2003; 114: 1765-1779.
[6] Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst H. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, PA; 2000. Available from: doi: 10.1137/1.9780898719581.
[7] Horn RA, Johnson CR. *Matrix Analysis*. 2nd ed. Cambridge, UK: Cambridge University Press; 2013.
[8] Shapiro H. The Weyr characteristics. *The American Mathematical Monthly*. 1999; 106: 919-929. Available from: doi: 10.1080/00029890.1999.12005141.
[9] Kågström B, Ruhe A. An algorithm for numerical computation of the Jordan normal form of a complex matrix. *ACM Transactions on Mathematical Software*. 1980; 6: 398-419. Available from: doi: 10.1145/355900.355912.
[10] Kågström B, Ruhe A. Algorithm 560: JNF, an algorithm for numerical computation of the Jordan normal form of a complex matrix. *ACM Transactions on Mathematical Software.* 1980; 6: 437-443. Available from: doi: 10.1145/355900.355917.
[11] Petkov PH, Konstantinov MM. The numerical Jordan form. *Linear Algebra and its Applications*. 2022; 638: 1-45.
[12] Golub GH, Van Loan CF. *Matrix Computations*. 4th ed. Baltimore, MD: The Johns Hopkins University Press; 2013.
[13] Stewart GW. *Matrix Algorithms, volume 2: Eigensystems*. SIAM, Philadelphia, PA; 2001.
[14] Ruhe A. An algorithm for numerical determination of the structure of a general matrix. *BIT*. 1970; 10: 196-216.
[15] Gantmacher FR. *Theory of Matrices, volume 1*. New York: AMS Chelsea Publishing; 1959.
[16] Petkov PH. Jordan-Schur algorithms for computing the matrix exponential. *Submitted to Mathematics*. 2022.
[17] Matrix Canonical Structure (MCS) Toolbox. Available from: https://www.umu.se/en/research/projects/stratigraph-and-mcs-toolbox [Accessed 2nd October 2022].
[18] Demmel J, Kågström B. The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part I: Theory and algorithms. *ACM Transactions on Mathematical Software*. 1993;

19: 160-174. Available from: doi: 10.1145/152613.152615.
[19] Demmel J, Kågström B. The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part II: Software and applications. *ACM Transactions on Mathematical Software*. 1993; 19: 175-201. Available from: doi: 10.1145/152613.152616.
[20] Higham NJ. Perturbation theory and backward error for $AX - XB = C$. *BIT*. 1993; 33: 124-136. Available from: doi: 10.1007/BF01990348.
[21] Higham NJ. *Accuracy and Stability of Numerical Algorithms*. 2nd ed. SIAM, Philadelphia, PA; 2002.
[22] Stewart GW. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*. 1973; 15: 727-764.
[23] Horn RA, Johnson CR. *Topics in Matrix Analysis*. Cambridge, UK: Cambridge University Press; 1991.
[24] Stewart GW. *Matrix Algorithms, volume 1: Basic Decompositions*. SIAM, Philadelphia, PA; 1998.
[25] The MathWorks, Inc., Natick, Mass. *MATLAB Version 9.9.0.1538559 (R2020b)*. 2020.
[26] Higham NJ. *The Matrix Function Toolbox*. Available from: http://www.ma.man.ac.uk/~higham/mftoolbox/ [Accessed 2nd October 2022].
[27] Bavely CA, Stewart GW. An algorithm for computing reducing subspaces by block diagonalization. *SIAM Journal on Numerical Analysis*. 1979; 16: 359-367. Available from: doi: 10.1137/0716028.