Turk. J. Math. Comput. Sci.
11(1)(2019) 1–7
© MatDer
http://dergipark.gov.tr/tjmcs
http://tjmcs.matder.org.tr

MATDER

Solving the Singular Semi-Sylvester Equation Using Drazin-Inverse and DGMRES Algorithm

Majid Adib

Department of Mathematics, Faculty of Sciences, University of Zanjan, 415195-313, Iran

Received: 18-02-2019 • Accepted: 06-06-2019

Abstract. In this paper, we want to solve the singular semi-Sylvester equation using the Drazin-inverse and the Drazin-inverse generalized minimum residual method ($DGMRES(m)$ algorithm). First, we transform the semi-Sylvester equation into a multiple linear systems. Then, we present the conditions and assumptions needed to apply the $DGMRES(m)$ algorithm. We compare our proposed method with the Galerkin projection method in point of view CPU-time, accuracy and iteration number. Finally, by some numerical experiments, we show the efficiency of the proposed method.

2010 AMS Classification: Primary 47A55; Secondary 39B52, 34K20, 39B82.

Keywords: Semi-Sylvester equations, multiple linear systems, DGMRES algorithm, Galerkin projection method.

1. Introduction

The semi-Sylvester equation

$$AX - EXB = C, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{s\times s}$ and $C \in \mathbb{R}^{n\times s}$ are given and $X \in \mathbb{R}^{n\times s}$ is to be determined, is one of the most important matrix equations in theory and applications and appear frequently in many areas. We refer the reader to the elegant survey by Bhatia and Rosenthal [6] and references therein for a history of the equation and many interesting and important theoretical result. These types of equations are important in a number of applications such as matrix eigen-decompositions [13, 23], control theory [10, 17], model reduction [1, 3, 24], numerical solution of matrix differential Riccati equations, and many more.

In [16], Karimi and Attarzadeh showed the semi-Sylvester equation (1.1) has a unique solution if and only if (A, C) and (B^T, I) are regular matrix pairs with disjoint spectra. Several direct and iterative methods are proposed for solving semi-Sylvester equation (1.1). When the size of the cofficient matrices A and B are small, the popular and widely used numerical method is the Hessenberg-Schur algorithm [12]. For large and sparse matrices A and B, iterative schemes to solve the semi-Sylvester equations such as those based on the matrix sign function or Newton method are widely used [5, 14, 16]. During last years, sevral projection methods based on Krylov subspace methods have also been proposed, see, e.g., [11, 15, 18].

Email address: madib@znu.ac.ir (M. Adib)

In [16] Karimi and Attarzadeh showed that in a particular case, the semi-Sylvester equation (1.1) can be converted into the following multiple linear systems

$$A^{(i)}x^{(i)} = b^{(i)}, \quad i = 1, 2, \ldots, s. \tag{1.2}$$

In [8], Chan and Michael presented the Galerkin projection method for solving multiple linear systems (1.2). In [16], Karimi and Attarzadeh have considered a special case of the semi-Sylvester equation (1.1), in which the matrix B is normal. Then, by using the Schur decomposition of B, they transformed the semi-Sylvester equation (1.1) into the multiple linear systems (1.2). Finally, by presenting the following propositions 1.1 and 1.2, they studied the nonsingular case of multiple linear systems (1.2) and, in this case, they applied Galerkin projection method to solve the semi-Sylvester equation (1.1).

Proposition 1.1. *Let A and B are symmetric matrices and E is symmetric positive definite matrix and*

$$\lambda_j < \frac{\langle Ax, x\rangle}{\langle Ex, x\rangle}, \quad j = 1, 2, \ldots, s, \tag{1.3}$$

where λ_j be the eigenvalues of B. Then $\hat{A}^{(i)}$ is symmetic positive definite.

Proposition 1.2. *Let A, B and E be symmetric positive definite matrices and symmetric positive semi-definite matrix, respectively. Then $(A - \lambda_j E)$, $j = 1, 2, \ldots, s$ are symmetric positive definite, where λ_j be the eigenvalues of B.*

In this paper, we intend to consider a general case that the above propositions 1.1 and 1.2, dose not exist, that is, the multiple linear systems (1.2) be singular, so in this regard, we provide the following definition.

Definition 1.3. We say that the multiple linear systems (1.2) is singular, if at least one of the coefficients matrices is singular. Also we say that the semi-Sylvester equation (1.1) is singular if the corresponding multiple linear systems (1.2) is singular.

Now assume that the semi-sylvester equation is singular. In this case, we apply the Drazin-inverse and $DEGMRES(m)$ method for solving the multiple linear systems (1.2) and hence the semi-Sylvester equation (1.1). The results of this method will be compared with the results of Galerkin projection method [16], in point of view CPU-time, accurancy and iteration number. Note that the semi-Sylvester equation (1.1) is the generalization of the standard Sylvester equation (this means that, if E be identity matrix I or an arbitary nonsingular matrix then the semi-Sylvester equation (1.1) becomes the standard Sylvester equation).
The remainder of the paper is organized as follows. In Section 2, we will review the $DGMRES$ method. In Section 3, we explain how to numerical solve the semi-Sylvester equation (1.1) with the $DGMRES(m)$ method, and in Section 4, we will give some numerical expriments and compare them with the Galerkin projection method. Finally, we'll make some concluding remarks in Section 5.

2. *DGMRES* Method

Consider the following linear system

$$Ax = b, \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$ is a singular matrix, $b \in \mathbb{R}^n$ and $ind(A)$ is α. Here ind(A) is the smallest nonnegative number that satisfy in $rank(A^{\alpha+1}) = rank(A^{\alpha})$.

Definition 2.1. Let $A \in \mathbb{R}^{n\times n}$ and $ind(A) = \alpha$. The matrix $X \in \mathbb{R}^{n\times n}$ satisfying the conditions

(1) $AX = XA$,
(2) $A^{\alpha}XA = A^{\alpha}$,
(3) $XAX = X$,

is called the Drazin-inverse of the matrix A. The Drazin-inverse of A denoted by A^D.

We recall that the Drazin-inverse solution of the linear system (2.1) is the vector $A^D b$ [4, 7]. The Drazin-inverse solution $A^D b$ is the unique solution of the equation $A^{\alpha+1}x = A^{\alpha}b$ that belongs to $\mathcal{R}(A^{\alpha})$ ($\mathcal{R}(A^{\alpha})$ means range space of A) [25]. In [21], Sidi developed the $DGMRES$ method for singular system that is analogous to $GMRES$ method for nonsingular system. In addition, in [21], the author proposed an effective mode of usage for $DGMRES$, denoted $DGMRES(m)$, which is analogous to the $GMRES(m)$ and requires a fixed amount of storage for its implementation.

In restarted $DGMRES$ ($DGMRES(m)$) the method is restarted once Krylov subspace reachs dimension m, and the current approximate solution becomes the new initial guess for the next m iterations. The restart parameter m is generally chosen small relative to n to keep storage and computation requirments reasonable. In the sequel, we review the $DGMRES(m)$ method.

Krylov space methods are considered as one of the ten most important classes of numerical methods [9]. $DGMRES(m)$ method is a Krylov subspace method for computing the Drazin-inverse solution of consistent or inconsistent linear system (2.1) [21, 22]. In this method, there are not any restriction on the matrix A. Thus, in general, A is non-Hermitian, $\alpha = ind(A)$ is arbitrary, and the spectrum of A can be any shape. Thus, it is unnecessary for us to put any restriction on the linear system $Ax = b$. So the system may be consistent or inconsistent. We only assume that $ind(A)$ is known.

$DGMRES(m)$ method starts with an initial vector x_0 and generates a sequence of vectors $x_1, x_2, \dots$ as follows

$$x_m = x_0 + q_{m-1}(A)r_0, \quad r_0 = b - Ax_0, \tag{2.2}$$

where $q_{m-1}(\lambda)$ is a polynomial in λ of degree at most $m-1$ defined as follows

$$q_{m-1}(\lambda) = \sum_{i=1}^{m-\alpha} c_i \lambda^{\alpha+i-1}, \quad \alpha = ind(A). \tag{2.3}$$

Let we define

$$p_m(\lambda) = 1 - \lambda q_{m-1}(\lambda) = 1 - \sum_{i=1}^{m-\alpha} c_i \lambda^{\alpha+i}, \quad r_m = p_m(A)r_0. \tag{2.4}$$

Thus we have

$$x_m = x_0 + \sum_{i=1}^{m-\alpha} c_i A^{\alpha+i-1}, \quad r_m = b - Ax_m = r_0 - \sum_{i=1}^{m-\alpha} c_i A^{\alpha+i} r_0. \tag{2.5}$$

The Krylov subspace used is as follows

$$\mathcal{K}_{m-\alpha}(A, A^\alpha r_0) = span\{A^\alpha r_0, A^{\alpha+1} r_0, \dots, A^{m-1} r_0\}. \tag{2.6}$$

We orthogonize the Krylov vectors $\{A^\alpha r_0, A^{\alpha+1} r_0, \dots, A^{m-1} r_0\}$ by the Arnoldi-Gram-Schmidt process [2, 20], carried out like the modified Gram-Schmidt process:

(1) Let $\beta = \|A^\alpha r_0\|$ and set $v_1 = \beta^{-1}(A^\alpha r_0)$.
(2) For $i = 1, 2, \dots, m$ do
   (a) Compute $h_{ji} = \langle v_j, Av_i \rangle, \quad j = 1, 2, \dots, i.$
   (b) Compute $\hat{v}_i = Av_i - \sum_{j=1}^{i} v_j h_{ji}$.
(3) Let $h_{i+1,i} = \|\hat{v}_i\|$ and set $v_{i+1} = \dfrac{\hat{v}_i}{h_{i+1,i}}$.

Let we set resulting orthonormal vectors as the columns of the matrix $\hat{V}_k$ as follows

$$\hat{V}_k = [v_1 | v_2 | \dots | v_k], \quad k = 1, 2, \dots, m. \tag{2.7}$$

Thus we can write

$$x_m = x_0 + \hat{V}_{m-\alpha}\xi_m, \quad \xi \in \mathbb{R}^{m-\alpha}, \tag{2.8}$$

which we need to determine ξ_m. First, note that $r_m = r_0 - A\hat{V}_{m-\alpha}\xi_m$, so we have

$$A^\alpha r_m = A^\alpha r_0 - A^{\alpha+1}\hat{V}_{m-\alpha}\xi_m = \beta v_1 - A^{\alpha+1}\hat{V}_{m-\alpha}\xi_m. \tag{2.9}$$

Next, we write

$$A\hat{V}_k = \hat{V}_{k+1}\bar{H}_k; \quad \bar{H}_k = \begin{bmatrix} h_{11} & h_{12} & \dots & \dots & h_{1k} \\ h_{21} & h_{22} & \dots & \dots & h_{2k} \\ 0 & h_{32} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & h_{kk} \\ 0 & \dots & \dots & 0 & h_{k+1,k} \end{bmatrix}. \tag{2.10}$$

Note that $\bar{H}_k \in \mathbb{R}^{(k+1)\times k}$ and $rank(\bar{H}_k) = k$. If we apply (2.10) to $A^{\alpha+1}\hat{V}_{m-\alpha}$, we have

$$\begin{aligned} A^{\alpha+1}\hat{V}_{m-\alpha} &= A^{\alpha}\hat{V}_{m-\alpha+1}\bar{H}_{m-\alpha} \\ &= A^{\alpha-1}\hat{V}_{m-\alpha+2}\bar{H}_{m-\alpha+1}\bar{H}_{m-\alpha} = \cdots = \hat{V}_{m+1}\hat{H}_m; \\ \hat{H}_m &\equiv \bar{H}_m\bar{H}_{m-1}\ldots\bar{H}_{m-\alpha}. \end{aligned}$$

Thus

$$A^{\alpha} r_m = \beta v_1 - \hat{V}_{m+1}\hat{H}_m\xi_m, \tag{2.11}$$

we also have $\hat{V}_{m+1}^T\hat{V}_{m+1} = I_{(m+1)\times(m+1)}$ and $rank(\hat{H}_m) = m-\alpha$. We finally have the $(m+1)\times(m-\alpha)$ least squares problem

$$\|A^{\alpha} r_m\| = \|\beta e_1 - \hat{H}_m\xi_m\| = \min_{\xi\in\mathbb{R}^{m-\alpha}} \|\beta e_1 - \hat{H}_m\xi\|. \tag{2.12}$$

Note that n is normally very large and $m \ll n$, which implies that the problem in (2.12) is very small. Also, note that since $\hat{H}_m$ is a full rank, we can determine ξ_m by applying the QR decomposition on $\hat{H}_m$. Thus $\hat{H}_m = Q_m R_m$, where $Q_m \in \mathbb{R}^{(m+1)\times(m-\alpha)}$ is a unitary matrix, that is, $Q_m^T Q_m = I_{(m-\alpha)\times(m-\alpha)}$ and $R_m \in \mathbb{R}^{(m-\alpha)\times(m-\alpha)}$ is a upper triangular matrix. Since $\hat{H}_m$ is full rank, so R_m is nonsingular, therefore we can compute ξ_m by solution the upper triangular system as follows

$$R_m\xi_m = \beta(Q_m^T e_1), \quad e_1 = [1, 0, \ldots, 0]^T. \tag{2.13}$$

Consequently, the algorithm of the $DGMRES(m)$ method is as follows

Algorithm 2.1 (DGMRES(m) algorithm).

(1) *Choose an initial guess x_0 and compute $r_0 = b - Ax_0$ and $A^{\alpha} r_0$.*
(2) *Compute $\beta = \|A^{\alpha} r_0\|$ and set $v_1 = \beta^{-1}(A^{\alpha} r_0)$.*
(3) *Orthogonalize the Krylov vectors $A^{\alpha} r_0, A^{\alpha+1} r_0, \ldots, A^{m+\alpha-1} r_0$ via the Arnoldi-Gram-Schmidt process carried out like the modified Gram-Schmidt process:*
For $j = 1, \ldots, m$ do
$u = Av_j$
For $i = 1, \ldots, j$ do
$h_{i,j} = \langle u, v_j\rangle$
$u = u - h_{i,j}v_i$
end
$h_{j+1,j} = \|u\|$,
$v_{j+1} = u/h_{j+1,j}$
end (The vectors $v_1, v_2, \ldots, v_{m+1}$ obtained by this way form an orthonormal set.)
(4) *For $k = 1 : m$ form the matrices $\hat{V}_k \in \mathbb{R}^{n\times k}$ and $\bar{H}_k \in \mathbb{R}^{(k+1)\times k}$*

$$\hat{V}_k = [v_1|v_2|\ldots|v_k], \quad \bar{H}_k = \begin{bmatrix} h_{11} & h_{12} & \ldots & \ldots & h_{1k} \\ h_{21} & h_{22} & \ldots & \ldots & h_{2k} \\ 0 & h_{32} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & h_{kk} \\ 0 & \ldots & \ldots & 0 & h_{k+1,k} \end{bmatrix}.$$

(5) *Form the matrix $\hat{H}_m = \bar{H}_m\bar{H}_{m-1}\ldots\bar{H}_{m-\alpha}$.*
(6) *Compute the QR decomposition of $\hat{H}_m$: $\hat{H}_m = Q_m R_m$; $Q_m \in \mathbb{R}^{(m+1)\times(m-\alpha)}$ and $R_m \in \mathbb{R}^{(m-\alpha)\times(m-\alpha)}$.($R_m$ is upper triangular.)*
(7) *Solve the (upper triangular) system $R_m\xi_m = \beta(Q_m^T e_1)$, where $e_1 = [1, 0, \ldots, 0]^T$.*
(8) *Compute $x_m = x_0 + \hat{V}_{m-\alpha}\xi_m$(then $\|A^{\alpha} r_m\| = \beta\sqrt{1 - \|Q_m^T e_1\|^2}$). If satisfied then stop.*
(9) *Set $x_0 = x_m$, compute $r_0 = b - Ax_0$, and go to (2).*

3. Numerically Solving the Semi-Sylvester Equation

In this section, we want to numerically solve the semi-Sylvester equation (1.1), by using the following theorem.

Theorem 3.1. *Let $A \in \mathbb{R}^{n\times n}$. Then A is a normal matrix if and only if it is unitarily similar to a diagonal matrix [19].*

Now let in the semi-Sylvester equation (1.1), B is a normal matrix. So, according to Theorem 3.1 there are a unitary matrix Q_B and a diagonal matrix Λ_B such that

$$B = Q_B \Lambda_B Q_B^T, \tag{3.1}$$

where the diagonal components of Λ_B are eigenvalues of B and the columns of the unitary matrix Q_B are normalized eigenvectores of B. By substitution of (3.1) in (1.1), we have

$$AXQ_B - EXQ_B\Lambda_B = CQ_B.$$

By taking $\hat{X} = XQ_B$ and $\hat{C} = CQ_B$, we obtain the following multiple linear systems

$$\hat{A}^{(i)}\hat{x}^{(i)} = \hat{c}^{(i)}, \quad i = 1, 2, \ldots, s, \tag{3.2}$$

where $\hat{A}^{(i)} = (A - \lambda_i E)$, $\hat{x}^{(i)}$ is the i-th column of $\hat{X}$ and $\hat{c}^{(i)}$ is the i-th column of $\hat{C}$.
Therefore, the semi-Sylvester equation (1.1) is converted to s linear systems. Notice, in this paper we considered the general case; that is, we did not impose any conditions and constraints on coefficients matrices of the resulting system. Therefore, it is possible to solve the semi-Sylvester equation by using s-time of the $DGMRES(m)$ method. In the next section we present some examples and numerical results.

4. Numerical Experiments and Conclusion

In this paper, we used the corresponding multiple linear systems form (form (3.2)) to solve the semi-Sylvester equation (1.1) and we considered the singular case. In this case, we used the $DGMRES(m)$ method to solve these systems. In this section, we present some experiments and numerical results. The described method is written with MATLAB.. In the following, we give three examples, in the first example 4.1, the equation is standard Sylvester equation and coefficients matrices are nonsingular and well-conditioned. In examples 4.2 and 4.3, we consider the singular case of the semi-Sylvester equation. In this tow examples the coefficients matrices are singular and ill-conditioned. In all examples, the initial matrix X_0, is the zero matrix and the stop condition is $\|A^\alpha r_i\|_2 \leq 1e-04$. The results obtained from these examples are presented in tables 1 and 2, which are compared with Galerkin projection method in point of view CPU-time, iteration numbers and residuals norm. In both tables 1 and 2, the symbols **Total itr**, **time** and **Cond** are total iteration numbers, total CPU-time and the maximun condition number of coefficients matrices, respectively.

Example 4.1. In this example we apply the $DGMRES(m)$ method on the standard Sylvester equation, that is, the matrix $E = I$. Also the matrices A, B and C are as follows

$$A = hilb(n,n), \quad E = eye(n,n),$$
$$B = -tridiag(-1 + \frac{1}{1+s}, 5, -1 + \frac{1}{1+s}), \quad C = ones(n,s),$$

where $n = 1000$ and $s = 4$. The numerical results are presented in table 1. We recall that resulting systems are nonsingular and well-conditioned (the maximum condition number is 1.66).

method(1000,4,m)	time(s)	Total itr	min $\|A^\alpha r_i\|_2$	max $\|A^\alpha r_i\|_2$	Cond
Galerkin	8.3464e-02	10	1.7554e-14	2.3386e-06	1.66
DGMRES(10)	1.5122e-01	4	3.5004e-29	1.5053e-13	1.66

Table 1. The results obtained from applying Galerkin projection and DGMRES(m) methods on example 4.1

Example 4.2. In this example we consider semi-Sylvester equation that coefficients matrices are singular, the maximum condition number is $3.36e + 22$ and $ind(A^{(i)})$ are all equal to 5. The matrices constituting the semi-Sylvester are as follows:

$$A = 5 * hilb(n, n), \quad E = hilb(n, n),$$
$$B = tridag(-1 + \frac{1}{s+1}, 5, -1 + \frac{1}{1+s}), \quad C = ones(n, s),$$

where $n = 1000$ and $s = 4$. The numerical results obtained in table 2 are presented.

Example 4.3. In this example we consider semi-Sylvester equation that coefficients matrices are singular, maximum condition number is $1.9e + 21$ and $ind(A^{(i)})$ are 6, 5, 5, 6, respectively. The matrices constituting the semi-Sylvester are as follows:

$$A = hilb(n, n), \quad E = hilb(n, n),$$
$$B = tridiag(-1 + \frac{1}{s+1}, 5, -1 + \frac{1}{1+s}), \quad C = eye(n, s),$$

where $n = 1000$ and $s = 4$. The numerical results are presented in table 2.

method	problem	Tol	time(s)	Total itr	min $\|A^{\alpha} r_i\|_2$	max $\|A^{\alpha} r_i\|_2$	Cond
Galerkin	example 4.2	1e-04	2.17	430	2.2590e-19	7.6387e-05	3.3e+22
Galerkin	example 4.3	1e-02	4.28e+01	8525	6.2908e-03	9.4382e-03	1.9e+21
Galerkin	example 4.3	1e-04	-	-	-	-	1.9e+21
DGMRES(10)	example 4.2	1e-04	1.35	4	6.5855e-23	2.0287e-05	3.3e+22
DGMRES(11)	example 4.3	1e-02	1.21	4	3.5937e-09	1.5710e-04	1.9e+21
DGMRES(11)	example 4.3	1e-04	1.52	5	3.5937e-09	5.1150e-06	1.9e+21

TABLE 2. The results obtained from applying the Galerkin projection and DGMRES(m) methods on examples 4.2 and 4.3

As the results of the examples in Tables 1 and 2 show, when the coefficients matrices are nonsingular and well-conditioned, the Galerkin projection method is better than the $DGMRES(m)$ method in point of view the CPU-time, although in terms of the number of itterations and the residuals norm $\|A^{\alpha} r_i\|_2$ the $DGMRES(m)$ method shows a better result. But the results of Table 2 show that when the coefficients matrices are singular and ill-conditioned, in point of view CPU-time, iteration numbers and residuals norm $\|A^{\alpha} r_i\|_2$, the $DGMRES(m)$ method has a more better performance than the Galerkin projection method.

Acknowledgement

The authors are grateful of the anonymous referees for their com- ments which substantially improved the quality of this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] Autoulas, A.C., Approximation of Large-Scale Dynamical Systems, Advances in Design and Control, Siam, Philadelphia, PA, USA, 2005. 1

[2] Arnoldi, W.E., *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of applied mathematics, **9**(2007), 17–290. 2

[3] Baur, U., Benner, P., *Cross-gramian based model reduction for data-sparse systems*, Electronic Transactions on Numerical Analysis, **31**(2008), 256–270. 1

[4] Ben-Israel, A., Greville, T.N., Generalized Inverses: Theory and Applications, volume 15. Springer Science & Business Media, 2003. 2

[5] Benner, P., Factorized Solution of Sylvester Equations with Applications in Control, Sign (H), 1:2, 2004. 1

[6] Bhatia, R., Rosenthal, P., *How and why to solve the operator equation axxb= y*, Bulletin of the London Mathematical Society, **29**(1997), 1–21. 1

[7] Campbell, S.L., Meyer, C.D., Generalized Inverses of Linear Transformations, Siam, 2009. 2

[8] Chan, T.F., Ng, M.K., *Galerkin projection methods for solving multiple linear systems*, SIAM Journal on Scientic Computing, **21**(1999), 836–850. 1

[9] Dangarra, J., Sullivan, F., Guest Editors Introduction to The Top 10 Algorithms, Comput. Scince. Eng, 2(1):2, 2000. 2

[10] Datta, B.N., Numerical Methods for Linear Control Systems: Design and Analysis, volume 1. Academic Press, 2004. 1

[11] Guennouni, A.E., Jbilou, K., Riquet, A., *Block krylov subspace methods for solving large sylvester equations*, Numerical Algorithms, **29**(2002), 1–3. 1

[12] Golub, G., Nash, S., Van Loan, C., *A hessenberg-schur method for the problem ax+ xb= c*, IEEE Transactions on Automatic Control, **24**(1979), 909–913. 1

[13] Golub, G., Van Loan, C., Matrix Computations, 2nd Missing. This means that the interpolation was to be ed, 1989. 1

[14] Hoskins, W., Meek, D., Walton, D., *The numerical solution of the matrix equationxa+ay= f*, BIT Numerical Mathematics, **17**(1977), 184–190. 1

[15] Jbilou, K., *Low rank approximate solutions to large sylvester matrix equations*, Applied mathematics and computation, **177**(2006), 365–376. 1

[16] Karimi, S., Attarzadeh, F., *A new iterative scheme for solving the semi sylvester equation*, Applied Mathematics, **4**(2013), 1–6. 1, 1, 1

[17] Lu, L., Wachspress, E.L., *Solution of lyapunov equations by alternating direction implicit iteration*, Computers & Mathematics with Applications, **21**(1991), 43–58. 1

[18] Robbe, M. anf Sadkane, M., *Use of near-breakdowns in the block arnoldi method for solving large sylvester equations*, Applied Numerical Mathematics, **58**(2008), 486–498. 1

[19] Saad, Y., Iterative Methods for Sparse Linear Systems. SIAM, 2003. 3.1

[20] Saad, Y., Schultz, M.H., *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on scientic and statistical computing, **7**(1986), 856–869, 2

[21] Sidi, A., *A unied approach to krylov subspace methods for the drazin-inverse solution of singular nonsymmetric linear systems*, Linear Algebra and its Applications, **298**(1999), 99–113. 2

[22] Sidi, A., *Dgmres: A gmres-type algorithm for drazin-inverse solution of singular non-symmetric linear systems*, Linear Algebra and its Applications, **335**(2001), 189–204. 2

[23] Sima, V., Algorithms for Linear-Quadratic Optimization, volume 200. CRC Press, 1996. 1

[24] Sorensen, D.C., Antoulas, A., *The sylvester equation and approximate balanced reduction*, Linear Algebra and its Applications, **351**(2002), 671–700. 1

[25] Wei, Y., Wu, H., *Additional results on index splittings for drazin inverse solutions of singular linear systems*, Electronic Journal of Linear Algebra, **27**(2001), 300–332. 2