



6-2011

New Computational Algorithms for Analyzing the Stability of the Differential Equations System

H. S. Najafi
University of Guilan

A. H. Refahi Sheikhan
University of Guilan

Follow this and additional works at: <https://digitalcommons.pvamu.edu/aam>



Part of the [Numerical Analysis and Computation Commons](#)

Recommended Citation

Najafi, H. S. and Refahi Sheikhan, A. H. (2011). New Computational Algorithms for Analyzing the Stability of the Differential Equations System, *Applications and Applied Mathematics: An International Journal (AAM)*, Vol. 6, Iss. 1, Article 10.

Available at: <https://digitalcommons.pvamu.edu/aam/vol6/iss1/10>

This Article is brought to you for free and open access by Digital Commons @PVAMU. It has been accepted for inclusion in *Applications and Applied Mathematics: An International Journal (AAM)* by an authorized editor of Digital Commons @PVAMU. For more information, please contact hvkoshy@pvamu.edu.



Available at
<http://pvamu.edu/aam>
Appl. Appl. Math.
ISSN: 1932-9466

Applications and Applied
Mathematics:
An International Journal
(AAM)

Vol. 6, Issue 1 (June 2011) pp. 130 – 142
(Previously, Vol. 6, Issue 11, pp. 1870 – 1882)

New Computational Algorithms for Analyzing the Stability of the Differential Equations System

H. Saberi, Najafi and A.H. Refahi Sheikhani

Department of Applied Mathematics
Faculty of Mathematical Sciences
University of Guilan
Rasht, Iran

hnajafi@guilan.ac.ir, ah_refahi@yahoo.com

Received: October 15, 2010; Accepted: May 9, 2011

Abstract

In this paper we show how to improve the approximate solution of the large Lyapunov equation obtained by an arbitrary method. Moreover, we propose a new method based on refinement process and Weighted Arnoldi algorithm for solving large Lyapunov matrix equation. Finally, some numerical results will be reported to illustrate the efficiency of the proposed method.

Keywords: Krylov Subspace, Lyapunov Equation, Inertia , Stability, Refinement

MSC(2010) No.: 65F30; 65F50

1. Introduction

An important class of linear matrix equations are the Lyapunov equation

$$XA + A^T X = C. \tag{1.1}$$

Since the fundamental work of Lyapunov on the stability of the motion, these matrix equations have been widely used in stability theory of differential equations and play an important role in

control and communications theory [see Datta et al. (2004), Jbilou (2006)]. Direct methods for solving the matrix equation (1.1) are attractive if the matrices are of small size and these methods are based on Schur decomposition.

Iterative methods for solving large Lyapunov matrix equations have been developed during the past years [see Benner et al. (2008), Eppler et al. (2010), Jbilou (2010), Silva et al. (2007), Stykel (2008)]. These methods use Galerkin and GMRES procedures to produce low-dimensional Lyapunov matrix equations that are solved by the direct methods. Both procedures make use of the Arnoldi process to compute an orthonormal basis of certain Krylov subspace. In this paper, we extend the idea to propose a new projection method for solving (1.1) based on weighted block Krylov subspace method. Moreover, the refinement method presented in section 4 has the capability of improving the results obtained by any other methods. The paper is organized as follows. In Section 2, we introduce some important theorems and applications of the matrix equation (1.1). Then we describe a new direct method for solving Lyapunov equation in in Section 3. Section 4 contains the main minimal residual algorithm for iteratively solving the Lyapunov matrix equation (1.1). Several numerical examples presented in Section 5. Finally, the conclusions are given in the last section.

2. Some Fundamental Results

In this section we present some existence and uniqueness results related to (1.1).

Theorem 2.1:

Let $\lambda_1, \lambda_2, \dots, \lambda_n$ be the eigenvalues of A . Then the Lyapunov equation $XA + A^T X = C$ has a unique solution X , if and only if $\lambda_i + \lambda_j \neq 0$, $i, j = 1, \dots, n$.

Proof:

See Datta et al. (2004).

Theorem 2.2:

The system of differential equation

$$\dot{x}(t) = Ax(t). \tag{2.1}$$

is asymptotically stable (that is $x(t) \rightarrow 0$ as $t \rightarrow \infty$) if and only if all the eigenvalues of A have negative real parts.

Proof:

See Datta et al. (2000).

Theorem 2.3:

The system (2.1) is asymptotically stable if and only if for any Hermitian positive definite matrix C , there exists a unique Hermitian positive definite matrix X satisfying the Lyapunov equation $XA + A^T X = C$.

Proof:

See Datta et al. (2000).

Definition 2.1.

The inertia of a matrix A of dimension n , denoted by $In(A)$, is a triplet $(\pi(A), \nu(A), \delta(A))$, where $\pi(A)$, $\nu(A)$ and $\delta(A)$ are the number of eigenvalues of A with positive, negative and zero real parts, respectively.

Note that $\pi(A) + \nu(A) + \delta(A) = n$, and A is a stable matrix if and only if $In(A) = (0, n, 0)$.

Theorem 2.4 (The main inertia theorem):

(i) A necessary and sufficient condition for a Hermitian matrix X to satisfies

$$XA + A^T X = C > 0 \tag{2.2}$$

is $\delta(A) = 0$.

(ii) If X is Hermitian and satisfies (2.2), then $In(A) = In(X)$.

Proof:

See Ostrowski et al. (1962).

3. New Direct Method

As we already have mentioned, many numerical methods have been developed by different authors. For example, Stykel (2008) generalize an alternating direction implicit method and the Smith method for large-scale generalized Lyapunov equations. Moreover, the Schur method proposed by Bartels and Stewart is now widely used as an effective computational method for the Lyapunov equation. But, it's the numerical stability has not been investigated. Before giving a new algorithm in this section, let us present a summary description of the Arnoldi process.

The Arnoldi process constructs an orthonormal basis $V_m = [v_1, \dots, v_m]$ of the Krylov subspace

$$K_m(A, v) = \text{span}(v, Av, \dots, A_{m-1}v),$$

using the modified Gram-Schmidt process and starts with the vector $v_1 = \frac{v}{\|v\|_2}$.

Algorithm 1 (Arnoldi Process):

```

for
     $j = 1, \dots, m$ 
         $w = A\tilde{v}_j$ 
    for
         $i = 1, \dots, j$ 
             $h_{i,j} = (w, v_i)_2$ 
             $w = w - h_{i,j}v_i$ 
    end
         $h_{j+1,j} = \sqrt{w^T w}$ , if  $h_{j+1,j} = 0$  stop
         $v_{j+1} = w/h_{j+1,j}$ 
    end
    
```

Let the matrix $H_m \in R^{m \times m}$ be the Hessenberg matrix whose nonzero entries are the scalars $h_{i,j}$ constructed by the Arnoldi process. Then, we have

$$V_m^T V_m = I_m \tag{3.1}$$

and

$$H_m = V_m^T A V_m. \tag{3.2}$$

Now we present the new direct method in this section. The following method is based on reduction of A^T to the Hessenberg matrix using Krylov’s subspace methods.

Algorithm 2:

(1) Reduction of the problem.

Let the matrix $H_n \in R^{n \times n}$ be the Hessenberg matrix whose nonzero entries constructed by the Arnoldi process. Then, we have

$$V_n^T V_n = I_n \quad H_n = V_n^T A^T V_n.$$

By this transformation, the Lyapunov matrix equation $XA + A^T X = C$ is reduced to

$$YH^T + HY = \hat{C}, \tag{3.3}$$

where

$$H = V^T A^T V, \hat{C} = V^T C V, Y = V^T X V.$$

(2) Solving the Reduced Problem

The reduced equation to be solved is $YH^T + HY = \hat{C}$. Let

$$Y = (y_{ij}), \quad \hat{C} = (\hat{c}_{ij}) \quad \text{and} \quad H = (h_{ij}).$$

Thus, we can rewrite (3.3) as

$$\begin{pmatrix} \sum_{i=1}^n (h_{1i} y_{i1} + h_{1i} y_{1i}) & \sum_{i=1}^n (h_{2i} y_{i1} + h_{1i} y_{2i}) & \dots & \sum_{i=1}^n (h_{ni} y_{i1} + h_{1i} y_{ni}) \\ \sum_{i=1}^n (h_{1i} y_{i2} + h_{2i} y_{1i}) & \sum_{i=1}^n (h_{2i} y_{i2} + h_{2i} y_{2i}) & \dots & \sum_{i=1}^n (h_{ni} y_{i2} + h_{2i} y_{ni}) \\ \vdots & \dots & \ddots & \vdots \\ \sum_{i=1}^n (h_{1i} y_{in} + h_{ni} y_{1i}) & \sum_{i=1}^n (h_{2i} y_{in} + h_{ni} y_{2i}) & \dots & \sum_{i=1}^n (h_{ni} y_{in} + h_{ni} y_{ni}) \end{pmatrix} = \begin{pmatrix} \hat{c}_{11} & \hat{c}_{12} & \dots & \hat{c}_{1n} \\ \hat{c}_{21} & \hat{c}_{22} & \dots & \hat{c}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{c}_{n1} & \hat{c}_{n2} & \dots & \hat{c}_{nn} \end{pmatrix}. \tag{3.4}$$

Now let $Y = [y_1, y_2, \dots, y_n]$, where y_i is the i^{th} column of Y and $\hat{C} = [c_1, c_2, \dots, c_n]$, where c_i is the i^{th} column of \hat{C} . We define a linear operator $vec: R^{n \times n} \rightarrow R^{n^2}$,

$$vec(Y) = [y_1^T, y_2^T, \dots, y_n^T]^T.$$

Since H is upper Hessenberg ($h_{ij} = 0, i > j + 1$), the Lyapunov equation (3.4) can be formulated as $n^2 \times n^2$ linear systems

$$\hat{H} \text{vec}(Y) = \text{vec}(C), \tag{3.5}$$

where \hat{H} is a block Hessenberg matrix as

$$\hat{H} = \begin{pmatrix} \hat{H}_{11} & \hat{H}_{12} & \dots & \dots & \hat{H}_{1n} \\ \hat{H}_{21} & \hat{H}_{22} & \dots & \dots & \hat{H}_{2n} \\ 0 & \hat{H}_{32} & \hat{H}_{33} & \dots & \hat{H}_{3n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \hat{H}_{nn-1} & \hat{H}_{nn} \end{pmatrix},$$

where

$$\hat{H}_{ii} = H_n + H_{ii} * I_n \quad (i = 1, 2, \dots, n),$$

$$\hat{H}_{ij} = \text{diag}(H_{ij} \quad (i = 2, 3, \dots, n \quad j = i - 1, \dots, n)).$$

Remark 3.1:

Krylov subspace method such as the GMRES algorithm could be used to solve the linear system (3.5). This formulation is expensive for larg problem. In next Section we give a new iterative method based on refinement process and Weighted Krylov subspace technique for solving large sparse Lyapunov equations.

4. New Iterative Method

Due to the efficiency of iterative methods, we aimed to propose an iterative algorithm for solving (1.1). There are numerous iterative methods for solving Lyapunov equation (see Benner et al. (2008), Eppler et al. (2010)). The reduced model described by Eppler is based on the positive semi-definite of the solution. In this section we first, briefly describe the Weighted Arnoldi algorithm. Then, we introduce a refinement process to improve the approximate solution of the Lyapunov equation obtained by an arbitrary method. Finally, we propose the main algorithm based on Weighted Arnoldi process for analyzing the stability of the differential equations system.

4.1. Weighted Arnoldi Process

If u and v are two vectors of R^n , their D-scalar product is

$$(u, v)_D = v^T D u = \sum_{i=1}^n d_i u_i v_i,$$

where $D = \text{diag}(d_1, d_2, \dots, d_n)$ is a diagonal matrix.

This inner product is well defined if and only if the matrix D is positive definite, that is, $d_i > 0, \forall i \in \{1, \dots, n\}$.

In this case, we can define the D-norm $P \cdot P_D$ associated with this inner product by

$$PuP_D = \sqrt{(u, u)_D} = \sqrt{u^T D u} = \sqrt{\sum_{i=1}^n d_i u_i^2}, \quad \forall u \in R^n.$$

The following algorithm describes the weighted Arnoldi process which uses the D-inner product $(\dots)_D$ to construct a D-orthonormal basis of $K_m(A, v)$ starting with the vector $\tilde{v}_1 = v/PvP_D$.

Algorithm 3 (Weighted-Arnoldi Process):

```

for
    j = 1, ..., m
    w = A\tilde{v}_j
for
    i = 1, ..., j
    \tilde{h}_{i,j} = (w, \tilde{v}_i)_D
    w = w - \tilde{h}_{i,j}\tilde{v}_i
end
\tilde{h}_{j+1,j} = PwP_D, if \tilde{h}_{j+1,j} = 0 stop
\tilde{v}_{j+1} = w/\tilde{h}_{j+1,j}
end
    
```

The basis $\tilde{V}_m = [\tilde{v}_1, \dots, \tilde{v}_m]$ constructed by this algorithm is D-orthonormal, i.e.,

$$\tilde{V}_m^T D \tilde{V}_m = I_m. \tag{4.1}$$

The square Hessenberg matrix \tilde{H}_m whose nonzero entries are the scalars $\tilde{h}_{i,j}$, constructed by the weighted Arnoldi process, can be expressed in the form

$$\tilde{H}_m = \tilde{V}_m^T D A \tilde{V}_m. \tag{4.2}$$

4.2. Refinement Process

Theorem 4.1:

Let X_0 be the approximate solution obtained by an arbitrary method for the matrix equation (1.1). Let also $R(x_0) = C - (X_0A + A^T X_0)$ be the corresponding residual. Moreover, the matrices $H_m \in R^{m \times m}$ and $V_m \in R^{n \times m}$, obtained from the weighted Arnoldi process and Y_m is the solution of the low dimensional Lyapunov equation

$$X_m H_m + H_m^T X_m = V_m^T R_0 V_m. \quad (4.3)$$

Then,

$$V_m^T R(X_1) V_m = 0, \quad (4.4)$$

where

$$X_1 = X_0 + DV_m Y_m V_m^T D. \quad (4.5)$$

Proof:

Since $Y_m \in R^{m \times m}$ is the solution of the Lyapunov equation (4.3), by using (4.5) the corresponding residual $R(X_1)$, thus, satisfies:

$$\begin{aligned} R(X_1) &= C - (X_1 A + A^T X_1) \\ &= C - [(X_0 + DV_m Y_m V_m^T D)A + A^T (X_0 + DV_m Y_m V_m^T D)] \\ &= R(X_0) - DV_m Y_m V_m^T DA - A^T DV_m Y_m V_m^T D. \end{aligned}$$

Multiplying this relation from the left by V_m^T and from the right by V_m , we have

$$V_m^T R(X_1) V_m = V_m^T R(X_0) V_m - (V_m^T DV_m Y_m V_m^T DAV_m + V_m^T A^T DV_m Y_m V_m^T DV_m).$$

The square Hessenberg matrices H_m and the basis $V_m = [v_1, v_2, \dots, v_m]$, constructed by the weighted Arnoldi process, satisfy relations (4.1) and (4.2). Now by using these relations and (4.3), we have

$$V_m^T R(X_1) V_m = V_m^T R(X_0) V_m - (Y_m H_m + H_m^T Y_m) = 0.$$

Remark 4.1:

According to the results obtained in Theorem 4.1, we can develop an iterative method for the solution of the Lyapunov equation when the matrices A and C are large (dimensions of A and C is n) and sparse. If $m < n$, then instead of solving $XA + A^T X = C$, we can solve (4.3), which is a lower dimensional problem and obtain the solution of initial Lyapunov equation by using (4.5). The algorithm is as follows.

Algorithm 4 (Weighted Iterative Refinement (WIR) method):

- (1) Choose X_0 , m , tolerance ε , and compute $R_0 = C - (X_0 A + A^T X_0)$.
- (2) Construct the D - orthonormal basis V_m by the weighted Arnoldi process.
- (3) Compute $H_m = V_m^T D A V_m$.
- (4) Solve the reduced Lyapunov equation $Y_m H_m + H_m^T Y_m = V_m^T R_0 V_m$.
- (5) Set $X_1 = X_0 + D V_m Y_m V_m^T D$.
- (6) Set $R_1 = C - (X_1 A + A^T X_1)$.
- (7) If $\|R_1\| \leq \varepsilon$ stop, otherwise
 set $X_0 = X_1, R_0 = R_1$ and goto step (2).

5. Numerical Experiments

In this Section, we present some numerical examples to illustrate the effectiveness of Algorithms described in this paper for large and sparse Lyapunov equations. The numerical experiments were performed in Matlab on a PC with 2.20 GHz with main memory 2 GB.

Example 5.1.

Consider the Lyapunov equations $XA + A^T X = C$. We apply Schur methods with and without the refinement process to solve the matrix equations and compare them. This algorithm has been tested when the dimension of matrices A and C increases. The results are shown in Table 1.

$$A = \begin{pmatrix} 10 & 1.2 & .42 & .8 & 2.3 & .8 & 0 & \dots & \dots & 0 \\ 1.8 & 10 & 1.2 & .42 & .8 & 2.3 & .8 & 0 & \ddots & 0 \\ 1.6 & 1.8 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ 1.64 & 1.6 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & .8 & 0 \\ 1.3 & 1.64 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 2.3 & .8 \\ 1.61 & 1.3 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & .8 & 2.3 \\ 0 & 1.61 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & .42 & .8 \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1.2 & .42 \\ \vdots & \vdots & 0 & 1.61 & 1.3 & 1.64 & 1.6 & 1.8 & 10 & 1.2 & \\ 0 & 0 & \dots & 0 & 1.61 & 1.3 & 1.64 & 1.6 & 1.8 & 10 & \end{pmatrix}_{n \times n}$$

$$C = \begin{pmatrix} .1 & 2.21 & 1.4 & 1.5 & .13 & 2.62 & 0 & \dots & \dots & 0 \\ 1.3 & .1 & 2.21 & 1.4 & 1.5 & .13 & 2.62 & 0 & \ddots & 0 \\ 2.6 & 1.3 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ 1.7 & 2.6 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 2.62 & 0 \\ 2.3 & 1.7 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & .13 & 2.62 \\ 2.6 & 2.3 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1.5 & .13 \\ 0 & 2.6 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1.4 & 1.5 \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 2.21 & 1.4 \\ \vdots & \vdots & 0 & 2.6 & 2.3 & 1.7 & 2.6 & 1.3 & .1 & 2.21 \\ 0 & 0 & \dots & 0 & 2.6 & 2.3 & 1.7 & 2.6 & 1.3 & .1 \end{pmatrix}_{n \times n}$$

Table 1. Implementation of Schur methods with and without refinement process

n	Schur method		Schur method with Refinement process	
	Error	CPU-time(s)	Error	CPU-time(s)
4	5.99E-013	0.0012	8.69E-016	0.0015
8	3.70E-012	0.0010	9.55E-016	0.0033
12	4.74E-012	0.0016	7.80E-016	0.0069
16	7.14E-012	0.0023	6.70E-016	0.0087
32	8.16E-012	0.0079	6.47E-016	0.0217
40	2.52E-011	0.0134	1.28E-015	0.0366
100	9.33E-011	0.25	8.38E-015	0.5763

As the results in Table 1 show the Refinement method has worked with less errors compared to the Schur method, but is more time consuming (see Error and Time columns). Therefore, we intend to develop a method which works with less errors and less time consumption, compared to the Schur method.

Example 5.2.

Consider the matrices A and C of Example 5.1 with $n = 100$. the estimated condition numbers of A and C are respectively $5.94E+03$ and $5.39E+04$. We apply the WIR method for solving the $XA + A^T X = C$ and take $\varepsilon = 10^{-6}$. In the Table 2, we report the results given for different value of m .

Table 2. Implementation of WIR method for solving the Lyapunov equation with different value of m

m	r	l	$\ V_m^T DAV_m - H\ $	Iterations	CPU – time(s)
4	2	2	8.21E–014	321	5.24
8	2	4	7.45E–014	99	3.05
10	2	5	6.18E–014	58	2.32
12	2	6	4.62E–014	31	2.08
14	2	7	3.12E–014	19	1.86
16	2	8	1.11E–014	11	1.13
18	2	9	7.29E–013	4	0.287
20	2	10	5.96E–013	1	0.024

In Table 2, the results show that by increasing the values of m and l , the number of iterations decreases. The last column of Table 2 also shows the decreasing of time consumption. Note that the fourth column of this table is the error of the orthogonalization method. The desired accuracy has been chosen as 10^{-6} , but the model works well with any choice of 10^{-t} .

Example 5.3.

In Example 5.1, we see that the Schur method with refinement process in comparison with Schur method works better. but when A and C are large is not very interesting. On the other hand in Example 5.2, we see that the WIR method when $n = 100$ works with good accuracy. Now consider A and C are the same matrix that used in Example 5.1. We apply the WIR method by 4 iterations and Schur method with refinement process for solving the Lyapunov equation when the dimension of the matrices A and C are large. Results are shown in Table 3.

Table 3. Implementation of WIR and Schur methods for solving the Lyapunov equation

n	Schur method with refinement process		WIR method		Cond(C)
	Error	CPU-time(s)	Error	CPU-time(s)	
200	9.84E–011	2.77	1.47E–016	1.73	6.00E+006
400	2.15E–010	39.19	2.13E–016	21.19	1.66E+011
600	7.45E–08	75.54	2.20E–016	54.37	4.15E+014
800	1.16E–05	111.97	2.28E–016	81.16	7.21E+017
1000	0.018	148.37	2.55E–015	108.61	3.91E+019

In Table 3 we see that when we increase the dimension the results are still much better than the Schur numerical results. Note that less time has been consumed compared to the Schur method.

Example 5.4.

In this example we use four matrices from MATLAB collection for the matrix A . The first matrix is a sparse, random finite element matrix. The second matrix is a symmetric, positive semi-definite (SPD) Toeplitz matrix composed of the sum of 800 rank 2 SPD Toeplitz matrices. The Third matrix is a row diagonally dominant, tridiagonal that is ill-conditioned. The last matrix is a sparse singular, row diagonally dominant matrix resulting from discrediting the Neumann problem with the usual five-point operator on a regular mesh. For all of these examples the matrix is $C = sprand(n,n,d)$, where C is a random, sparse matrix with approximately $d \times n^2$ uniformly distributed nonzero entries with $d = 0.5$. We apply the Schur method with refinement process and WIR method with 3 iterations for solving the $XA + A^T X = C$. The results were presented in Table 4.

It is also obvious from Table 4 that the performance of WIR method is much better than the Schur method with refinement process, specifically for the ill-conditioned matrices.

Table 4. Effectiveness of Schur method with refinement and WIR algorithm for randomly generated matrices

Matrix	Schur method with refinement		WIR method		Cond(A)
	Error	CPU-time	Error	CPU-time	
$A = gallery('wathen',736,736)$	7.33E-08	78.46	6.41E-015	59.95	1.84E+03
$A = gallery('toepd',800,800)$	1.92E-06	96.15	5.27E-014	78.81	6.18E+04
$A = gallery('dorr',1200,0.01)$	4.33E-02	140.83	7.33E-013	127.02	1.35E+010
$A = gallery('neumann',1156)$	7.41E-03	129.16	1.11E-015	110.65	5.56E+17

6. Comments and Conclusion

In this paper, we have proposed two new algorithms for solving the large Lyapunov matrix equations. The WIR method are based on the weighted block Arnoldi algorithm and a Refinement process presented in Section 4 also has the capability of improving the results obtained by an arbitrary method. For example in this paper we apply the refinement process with Schur method. The numerical tests presented in this paper show the effectiveness of the proposed methods.

REFERENCES

- Benner, P, Li J, Rebecca, Penzl, T. (2008). Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems, *Numer.Linear Algebra Appl*, Vol. 15, pp. 755-777.
- Datta, B. N. (2004). *Numerical Methods for Linear Control Systems: Design and Analysis*, Elsevier Academic Press, Amsterdam.
- Datta, B. N. (2000). Stability and Inertia, *Linear Algebra Appl*, Vol. 303, pp. 563-600.
- Eppler A.K. and Bollhofer, M. (2010). An alternative way of solving large Lyapunov equations, *Proceedings in Applied Mathematics and Mechanics*, pp. 1-3.
- Jbilou, K. (2010). ADI preconditioned Krylov methods for large Lyapunov matrix equations, *Linear Algebra Appl*, Vol. 432, pp. 2473-2485.
- Jbilou, K. and Riquet, A.J. (2006). Projection methods for large Lyapunov matrix equations, *Linear Algebra Appl*, Vol. 415, pp. 344-358.
- Ostrowski, A. and Schneider, H. (1962). Some theorems on the inertia of general matrices, *J. Math. Anal. Appl*, Vol. 4, pp. 72-84.
- Silva F. C. and Simoes, R. (2007). On the Lyapunov and Stein equations (I), *Linear Algebra Appl*, Vol. 420, pp. 329-338.
- Silva F. C. and Simoes, R. (2007). On the Lyapunov and Stein equations (II), *Linear Algebra App.*, Vol. 426, pp. 305-311.
- Stykel, T. (2008). Low-rank Iterative methods for projected generalized Lyapunov Equations, *Electronic Transactions on Numerical Analysis*, Vol. 30, pp. 187-202.