

# Solution of Algebraic Riccati Equations using Schur Method and Newton-Kleinman Method

by

Xuanrui Li

A research paper  
presented to the University of Waterloo  
in fulfillment of the requirement for the degree of  
Master of Mathematics  
in  
Computational Mathematics

Supervisor: Professor Kirsten Morris

Waterloo, Ontario, Canada, 2019

© Xuanrui Li 2019

I hereby declare that I am the sole author of this research paper. This is a true copy of the research paper, including any required final revisions, as accepted by my examiners.

I understand that my research paper may be made electronically available to the public.

## Abstract

Algebraic Riccati equations determine the solution of infinite-horizon optimal control problems: Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG). These problems are fundamental problems in control theory. In this research paper, two main methods to solve the algebraic Riccati equations, the Schur method, and Newton-Kleinman method are discussed. The Schur method is for small order models as a direct solution method. In the Schur method, the computation of the basis that spans the stable invariant subspace of the Hamiltonian matrix is needed for the solution of the algebraic Riccati equations. However, this method fails for large dimension problems. An example is given to illustrate that the Schur method fails to give accurate solutions for problem order over 200. The Newton-Kleinman method uses an iterative scheme to get the solutions for the algebraic Riccati equations. The Alternating Direction Implicit (ADI) method and Cholesky-factor ADI method in solving Lyapunov equations are discussed. In these two methods, proper choice of the parameters can improve the convergence of the iterations [16]. In the simply supported beam example, we use MATLAB built-in function *care()* for the Schur method, and we use *lyap()* in solving Lyapunov equations in Newton-Kleinman method. An analysis of some numerical results is provided in the end.

# Table of Contents

List of Tables	v
List of Figures	vi
<b>1 Algebraic Riccati Equations</b>	<b>2</b>
<b>2 Schur Method</b>	<b>4</b>
2.1 Solution to ARE . . . . .	5
2.2 Implementation of the Schur method . . . . .	9
2.2.1 Balancing and Scaling . . . . .	12
2.2.2 Numerical Issues . . . . .	12
<b>3 Newton-Kleinman Method</b>	<b>13</b>
3.1 Solution to ARE . . . . .	14
3.2 Solution of Lyapunov Equation . . . . .	15
3.2.1 Numerical Issues . . . . .	21
<b>4 Example: Simply Supported Beam</b>	<b>23</b>
<b>5 Comparisons and Observations</b>	<b>30</b>
5.1 Observations . . . . .	30
<b>References</b>	<b>32</b>

# List of Tables

3.1	ADI and Cholesky-factor ADI method complexity comparison in solving Lyapunov equations . . . . .	22
4.1	Time per order n for Schur method with $Q = I, R = 1$ . . . . .	25
4.2	The Frobenius norm relative residual calculated using $\frac{\ Ric(P)\ _F}{\ P\ _F}$ for Schur method. $P$ is the solution to ARE obtained from <i>care()</i> . . . . .	27
4.3	The iteration numbers, timing and the Frobenius norm relative residual calculated using $\frac{\ Ric(P)\ _F}{\ P\ _F}$ for Newton-Kleinman method at different problem order with $Q = I, R = 1, K_0 = 0$ . $P$ is the solution to ARE approximated using Newton-Kleinman method . . . . .	28
4.4	Feedback Gain using 2-norm at each Newton-Kleinman iteration on problem order 24 with $Q = I, R = 1, K_0 = 0$ . . . . .	29

# List of Figures

4.1	The relative error calculated using $\frac{\ P-P_j\ _2}{\ P\ _2}$ in each Newton-Kleinman iteration at problem order 24. $P_j$ is the solution at $j^{th}$ Newton-Kleinman iteration. $P$ is the solution to ARE approximated by Newton-Kleinman method . . . . .	26
4.2	The Frobenius norm relative residual calculated using $\frac{\ Ric(P)\ _F}{\ P\ _F}$ for Schur method per order $n$ . $P$ is the solution to ARE obtained from <i>care()</i> . . . . .	28



# Chapter 1

## Algebraic Riccati Equations

Algebraic Riccati equations (AREs) determine the solutions of Linear-Quadratic Regulators (LQR) and Linear-Quadratic Gaussian (LQG) control problems. Assume  $R$  to be a positive definite matrix. Let  $(\cdot)^\top$  denote the transpose of a matrix. The objective is to find a feedback control  $u(t) = Kp(t)$  which solves the problem

$$\text{Min } J(u(t)) = \int_0^{+\infty} \frac{1}{2} [p^\top(t)Qp(t) + 2p^\top(t)Su(t) + u^\top(t)Ru(t)] dt \quad (1.1)$$

$$\text{subject to: } E\dot{p}(t) = Ap(t) + Bu(t), \quad p(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \quad (1.2)$$

$$y(t) = Cp(t), \quad y(t) \in \mathbb{R}^p \quad (1.3)$$

where the function  $p(t)$  is the state,  $y(t)$  is the controlled output, and  $u(t)$  is the input. Matrix  $A$  describes the internal dynamics in  $\mathbb{R}^{n \times n}$ ,  $B$  is the effect of the controlled input on the state in  $\mathbb{R}^{n \times m}$ , and  $E, C$  describe the sensors.  $E$  is assumed to be non-singular and  $C$  is in  $\mathbb{R}^{p \times n}$ . The number of state variables is  $n$ , namely the *order* of the system. The number of inputs of the system is  $m$ , and the number of outputs is  $p$ . Often  $Q = C^\top C$ . The feedback gain is  $-K$ . The matrix  $Q$  in  $\mathbb{R}^{n \times n}$  penalizes the state variable  $p(t)$  and  $Q$  is positive semi-definite. Matrix  $R$  is a symmetric matrix, which penalizes the control signal [13].

Equation (1.1) is defined to be the cost of the controlled system. The non-negative solution  $P$  to the ARE (1.4) yields the optimal cost. The optimal control is determined by this  $P$  [13].



The basic ARE arising in continuous-time problems takes the form [1]

$$A^\top P E + E^\top P A - (E^\top P B + S)R^{-1}(B^\top P E + S^\top) + C^\top C = 0 \quad (1.4)$$

or

$$\hat{A}^\top P E + E^\top P \hat{A} - E^\top P B R^{-1} B^\top P E + C^\top C - S R^{-1} S^\top = 0 \quad (1.5)$$

where

$$\hat{A} := A - B R^{-1} S^\top$$

The more familiar form of the generalized version occurs when  $S = 0$  and  $E = I$ . Also with  $Q = C^\top C$ , we have

$$A^\top P + P A - P B R^{-1} B^\top P + Q = 0 \quad (1.6)$$

for continuous-time case.

In this research paper, we will focus on talking about the continuous-time ARE. Two methods are introduced regarding how to solve the AREs. We will discuss these two methods with details, including the theories behind and how to implement in practice. Numerical issues are considered. An example of simply supported beam of Euler-Bernoulli Beam is given to illustrate the applications.

# Chapter 2

## Schur Method

Let us start by stating some definitions and theorems before the detailed discussion about the Schur method.

**Definition 2.0.1** A *skew-symmetric* matrix  $A$  is a square matrix whose transpose equals its negative. That is,  $A^\top = -A$ , where  $A$  is of size  $n \times n$  in  $\mathbb{R}$ .

**Definition 2.0.2** A *Hamiltonian* matrix  $A$  is  $2n \times 2n$  such that  $JA$  is symmetric, where  $J$  is the skew-symmetric matrix

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

and  $I_n$  is the  $n \times n$  identity matrix. In other words,  $A$  is Hamiltonian if and only if  $(JA)^\top = JA$  or equivalently,  $J^{-1}A^\top J = -A$

**Definition 2.0.3** A *symplectic* matrix  $A$  of size  $2n \times 2n$  if  $J^{-1}A^\top J = A^{-1}$  or equivalently,  $A^\top J A = J$  where  $J$  is skew-symmetric.

**Definition 2.0.4** A matrix  $A$  of size  $n \times n$  in  $\mathbb{R}$  is *unitary* if  $A^\top = A^{-1}$ .

**Definition 2.0.5** A matrix  $A$  of size  $n \times n$  in  $\mathbb{R}$  is *orthogonal* if  $A^\top A = I$  or equivalently,  $A^\top = A^{-1}$ .

**Theorem 2.0.1 (Lemma B.4 [13])** *The spectrum  $\Lambda$  of the Hamiltonian matrix*

$$Z = \begin{bmatrix} A & -BR^{-1}B^\top \\ -Q & -A^\top \end{bmatrix}$$

*is symmetric with respect to the imaginary axis.*

**Theorem 2.0.2 (Theorem B.5 [13])** *Let  $A$  and  $B$  be  $n \times n$  real matrices. The equation*

$$BX + XA = 0 \tag{2.1}$$

*has only the solution  $X = 0$  if  $\lambda_i(A) + \lambda_j(B) \neq 0$  for all  $i, j = 1 \dots n$ .*

**Theorem 2.0.3 (Theorem 2.5 [13])** *The pair  $(A, B)$  is **controllable** if and only if the  $n \times np$  controllability matrix such that*

$$[B \ AB \ A^2B \ \dots \ A^{n-1}B]$$

*has full row rank  $n$ .*

**Theorem 2.0.4 (Theorem 2.11 [13])** *The pair  $(C, A)$  is **observable** if and only if the  $p \times pn$  observability matrix such that*

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix}$$

*has full column rank  $n$ .*

**Theorem 2.0.5 (Theorem 5.16 [13])** *Assume the pair  $(A, B)$  is controllable and the pair  $(C, A)$  is observable in system (1.2)-(1.3), then the ARE has unique non-negative definite solution.*

## 2.1 Solution to ARE

Assumptions:

Assume  $(A, B)$  is controllable and  $(C, A)$  is observable.

By Theorem (2.0.5), the ARE has a unique, non-negative definite solution to the system (1.2)-(1.3).

The connection of the Hamiltonian matrix and ARE is established if we rewrite the ARE in the following equivalent form

$$\begin{bmatrix} A & -BR^{-1}B^\top \\ -Q & -A^\top \end{bmatrix} \begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ P \end{bmatrix} (A - BR^{-1}B^\top P) \quad (2.2)$$

We claim  $P$  is the solution to the ARE.

**Definition 2.1.1** *If  $\max_{1 \leq i \leq n} \operatorname{Re}(\lambda_i(A)) < 0$ ,  $A$  is **Hurwitz**.*

**Theorem 2.1.1 (Theorem 8.4 [13])** *Let  $Q$  and  $R$  be real symmetric matrices and let  $A$  and  $B$  be real matrices. Suppose that the Hamiltonian matrix*

$$Z = \begin{bmatrix} A & -BR^{-1}B^\top \\ -Q & -A^\top \end{bmatrix}$$

*has no eigenvalues on the imaginary axis. For matrices  $X, Y$  define the matrix  $\begin{bmatrix} X \\ Y \end{bmatrix}$  whose columns are the eigenvectors associated with the negative real eigenvalues of  $Z$ .*

*If*

$$\operatorname{Range} \begin{bmatrix} X \\ Y \end{bmatrix} \cap \operatorname{Range} \begin{bmatrix} 0 \\ I \end{bmatrix} = \{0\} \quad (2.3)$$

*then the ARE*

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0 \quad (2.4)$$

*has the solution  $P = YX^{-1}$  with the property that  $A - BR^{-1}B^\top P$  is Hurwitz.*

*Conversely, if the ARE has a unique, non-negative definite solution  $P$  so that  $A - BR^{-1}B^\top P$  is Hurwitz, then (2.3) holds and the Hamiltonian matrix has no imaginary axis eigenvalues.*

*If a solution  $P$  does exist,  $P$  is real and symmetric and it is the only solution with the property that  $A - BR^{-1}B^\top P$  is Hurwitz.*

Proof:

By equation (2.2), we say  $P$  is a solution of the ARE if and only if

$$\text{Range} \begin{bmatrix} I \\ P \end{bmatrix}$$

is an  $n$ -dimensional invariant subspace of  $Z$ . Also,  $A - BR^{-1}B^{\top}P$  characterizes the restriction of  $Z$  to this subspace.

First assume that  $Z$  has no imaginary axis eigenvalues and that (2.3) holds. Since the eigenvalues of  $Z$  are symmetric about the imaginary axis by Theorem 2.0.1, this means that  $Z$  has an  $n$ -dimensional invariant subspace  $\nu$  where  $\nu := \text{Range} \begin{bmatrix} X \\ Y \end{bmatrix}$  on which the restriction of  $Z$  has all eigenvalues with negative real parts. The statement (2.3) implies that  $X$  is non-singular. So

$$\nu = \text{Range} \begin{bmatrix} X \\ Y \end{bmatrix} = \text{Range} \begin{bmatrix} XX^{-1} \\ YX^{-1} \end{bmatrix} = \text{Range} \begin{bmatrix} I \\ YX^{-1} \end{bmatrix} \quad (2.5)$$

Define  $P = YX^{-1}$ . Then equation (2.2) implies that  $P$  solves the ARE.

Since  $\nu$  is the invariant subspace of  $Z$  associated with its negative eigenvalues, equation (2.2) implies that  $A - BR^{-1}B^{\top}P$  is Hurwitz.

Now, we prove if the ARE has a unique, non-negative definite solution  $P$  such that  $A - BR^{-1}B^{\top}P$  is Hurwitz, then (2.3) holds and  $Z$  has no imaginary axis eigenvalues:

Assume that a solution  $P$  to the ARE exists such that  $A - BR^{-1}B^{\top}P$  is Hurwitz. Then, from equation (2.2),

$$\text{Range} \begin{bmatrix} I \\ P \end{bmatrix}$$

is an  $n$ -dimensional invariant subspace of  $Z$ . Also,  $A - BR^{-1}B^{\top}P$  characterizes the restriction of  $Z$  to this subspace. Thus,  $Z$  has an  $n$ -dimensional eigenspace associated with eigenvalues with negative real values. Since the spectrum of  $Z$  is symmetric about the imaginary axis (Theorem 2.0.1), this implies that  $Z$  has  $n$  eigenvalues with positive real parts,  $n$  with negative real parts, and thus no imaginary eigenvalues. Clearly,

$$\text{Range} \begin{bmatrix} I \\ P \end{bmatrix} \cap \text{Range} \begin{bmatrix} 0 \\ I \end{bmatrix} = \{0\} \quad (2.6)$$

Then it only remains to show that the solution  $P$  is real, symmetric, and unique.

First we show  $P$  is real, i.e. we need to prove  $P = P_c$ :

The column of  $\begin{bmatrix} X \\ Y \end{bmatrix}$  can be chosen complex conjugate in pair so that if  $X_c, Y_c$  are the complex conjugates of  $X$  and  $Y$ , respectively, then

$$\begin{bmatrix} X_c \\ Y_c \end{bmatrix} M = \begin{bmatrix} X \\ Y \end{bmatrix}$$

where  $M$  is a permutation matrix. Then

$$P = YX^{-1} = Y_c M M^{-1} X_c^{-1} = Y_c X_c^{-1} = P_c$$

and so  $P = P_c$ . The matrix  $P$  is real.

Then we show  $P$  is symmetric.

We multiply equation (2.2) on the left by

$$\begin{bmatrix} I & P^\top \end{bmatrix} J$$

where  $J$  is the skew-symmetric matrix such that  $J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$  where  $I$  is a identity matrix of size  $n \times n$ . We obtain

$$\begin{bmatrix} I & P^\top \end{bmatrix} J Z \begin{bmatrix} I \\ P \end{bmatrix} = (P^\top - P)(A - BR^{-1}B^\top P) \quad (2.7)$$

and we take transpose of the left hand side of equation (2.7), we get

$\begin{bmatrix} I & P^\top \end{bmatrix} Z^\top J^\top \begin{bmatrix} I \\ P \end{bmatrix}$ . Since  $Z$  is Hamiltonian, i.e.  $JZ = Z^\top J^\top$ , we get

$$(P^\top - P)(A - BR^{-1}B^\top P) = (A - BR^{-1}B^\top P)^\top (P^\top - P)^\top$$

so that

$$(P^\top - P)(A - BR^{-1}B^\top P) + (A - BR^{-1}B^\top P)^\top (P^\top - P) = 0.$$

By Theorem 2.0.2 and the fact that  $A - BR^{-1}B^\top P$  is Hurwitz, the only solution is  $P^\top - P = 0$ . Thus,  $P$  is symmetric.

Finally, we prove uniqueness.

Suppose that an ARE has non-negative definite solutions,  $P_1$  and  $P_2$ , then

$$A^\top P_1 + P_1 A - P_1 B R^{-1} B^\top P_1 + Q = 0 \quad (2.8)$$

$$A^\top P_2 + P_2 A - P_2 B R^{-1} B^\top P_2 + Q = 0. \quad (2.9)$$

Subtracting the second equation from the first and re-arranging,

$$(P_1 - P_2)(A - B R^{-1} B^\top P) + (A - B R^{-1} B^\top P)^\top (P_1 - P_2) = 0 \quad (2.10)$$

then we use Theorem (2.0.2) again and get  $P_1 - P_2 = 0$ , proving that the non-negative definite solution is unique[13]. ■

Equation (1.6) can be solved by finding an orthogonal transformation matrix  $U$  of size  $2n \times 2n$  in  $\mathbb{R}$  such that

$$U^\top Z U = S \quad (2.11)$$

where

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \quad U_{ij} \in \mathbb{R}^{n \times n}$$

$$Z = \begin{bmatrix} A & -B R^{-1} B^\top \\ -Q & -A^\top \end{bmatrix} \text{ is a Hamiltonian matrix}$$

and  $S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$  is a quasi-upper-triangular (see Definition 2.2.1) matrix with all eigenvalues of  $S_{11}$  in the strict left-half-plane. The  $n$  column vectors comprising  $\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix}$  span the stable invariant subspace and the solution of equation (1.6) is given by  $P = U_{21} U_{11}^{-1}$ . This is so-called **Schur method** for solving ARE [1].

## 2.2 Implementation of the Schur method

There are two main steps for the Schur method:

1. The reduction of  $S$  in  $U^\top Z U = S$  to an ordered real Schur form, which is the most difficult step in this algorithm.

2. The solution of an  $n^{\text{th}}$  order linear matrix equation, i.e.  $PU_{11} = U_{21}[9]$ .

1. **Step one**

Our objective in this step is to find an orthogonal matrix  $U$  of size  $2n \times 2n$  with matrix  $\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix}$  whose columns are the eigenvectors correspond to the negative real eigenvalues of Hamiltonian matrix  $Z$ . The order of eigenvalues of  $S$  matters since  $S_{11}$  has to be in the strict left-half-plane.

One approach is presented here to achieve step one.

Let's discuss these two theorems from classical similarity theory first:

**Theorem 2.2.1 (Schur canonical form)** [Theorem 3 [9]]: Let  $A \in \mathbb{R}^{n \times n}$  has  $n$  eigenvalues  $\lambda_1, \dots, \lambda_n$  of this order. Then there exists a unitary similarity transformations  $U$  that is uniquely determined such that  $U^H A U$  is upper triangular with diagonal elements  $\lambda_1, \dots, \lambda_n$  of that order.

**Definition 2.2.1** A *quasi upper triangular matrix* is a block upper triangular matrix where the blocks on the diagonal are  $1 \times 1$  or  $2 \times 2$ .

It is possible to work over  $\mathbb{R}$  since we can reduce Schur canonical form to a matrix of form quasi-upper-triangular with  $2 \times 2$  and  $1 \times 1$  blocks. The  $2 \times 2$  blocks correspond to complex conjugate eigenvalues of Hamiltonian matrix  $Z$  and  $1 \times 1$  blocks correspond to the real eigenvalues of Hamiltonian matrix  $Z$ . We refer to this canonical form as the real Schur form (RSF) [9].

**Theorem 2.2.2 (real Schur form)** [Theorem 4 [9]] Let  $A \in \mathbb{R}^{n \times n}$ , then there exists an orthogonal similarity transformation  $U$  such that  $U^T A U$  is quasi-upper-triangular. Moreover,  $U$  can be chosen so that the  $1 \times 1$  and  $2 \times 2$  diagonal blocks appear in any desired order.

**Definition 2.2.2** If in the above theorem, we partition  $S := U^T A U$  into  $\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$  where  $S_{11} \in \mathbb{R}^{n \times k}$ ,  $0 < k \leq n$ , we shall refer to the first  $k$  columns of  $U$  as the **Schur vectors** corresponding to  $\Lambda(S_{11}) \subseteq \Lambda(A)$



In MATLAB, the Schur decomposition has syntax

$$[U, S] = \text{schur}(Z, \dots)$$

which returns a unitary matrix  $U$  so that  $Z = USU^H$ . The matrix  $S$  is the Schur canonical form and

$$U^H U = \text{eye}(\text{size}(Z))$$

To re-order eigenvalues in Schur factorization, another syntax is used:

$$[U_S, Z_S] = \text{ordschur}(U, Z, \text{keyword})$$

This syntax sets the selected cluster of eigenvalues appears in the region specified by the eigenvalue region keyword.

The eigenvalue region keywords can be:

Keyword	Selected Region
'lhp'	Left-half plane ( $\text{real}(e) < 0$ )
'rhp'	Right-half plane ( $\text{real}(e) \geq 0$ )
'udi'	Interior of unit disk ( $\text{abs}(e) < 1$ )
'udo'	Exterior of unit disk ( $\text{abs}(e) \geq 1$ )

where  $e$  represents the eigenvalues of  $S$ . In our case, 'lhp' is what we want to use.

## 2. Step two

Consider  $n$ -th order linear matrix equation

$$PU_{11} = U_{21}$$

we solve for  $P = U_{21}U_{11}^{-1}$ . There are some linear equation solvers that we can use in MATLAB such as *decomposition()* and *solve()* or *linsolve()* in one step.

We will briefly summarize the algorithm as follows:

Consider the system  $PU_{11} = U_{21}$ , we take transpose so it is in the form of  $U_{11}^T P^T = U_{21}^T$  with  $U_{11}^T = LU$ ,

### Two-step Solution Procedures

- (a) Let  $UP^T = y$ , solve the lower triangular system  $Ly = U_{21}^T$  for  $y$  by forward substitution.

- (b) Solve the upper triangular system  $UP^\top = y$  for  $P^\top$  by backward substitution as desired solution since  $P$  is symmetric.

### 2.2.1 Balancing and Scaling

In Schur method, proper balancing and scaling for the Hamiltonian matrix  $Z$  can increase the computational efficiency. There are several ways to do balancing and scaling in our case. Two methods are presented here:

1. Use balancing matrix  $PD$  to reduce  $Z$  to  $Z_b$  where  $P$  is a permutation matrix,  $D$  is a diagonal matrix such that

$$D^{-1}PZPD = Z_b$$

Then find orthogonal matrix  $U$  which reduces  $Z_b$  to real Schur form. Combine all of the above, we get

$$U^\top Z_b U = S$$

where  $PDU$  is non-singular with the first  $n$  columns to be the eigenvectors correspond to the negative real eigenvalues of the Hamiltonian matrix  $Z$  [9].

2. The other way is to use a transformation matrix. Suppose  $T_0$  is a non-singular transformation such as  $\begin{bmatrix} T & 0 \\ 0 & T^{-\top} \end{bmatrix}$ . Since  $Z_w$  is symplectically similar to  $Z$  such that

$$Z_w = \begin{bmatrix} T & 0 \\ 0 & T^{-\top} \end{bmatrix}^{-1} Z \begin{bmatrix} T & 0 \\ 0 & T^{-\top} \end{bmatrix}$$

Here  $Z_w$  is Hamiltonian (or symplectic in the discrete time case). We will use  $Z_w$  instead of  $Z$  during our calculations [9].

### 2.2.2 Numerical Issues

#### Operation Counts:

The approximated operation counts are given for the solution of  $n^{\text{th}}$  order ARE of form (1.6). Assume the size of the Hamiltonian matrix is  $2n \times 2n$ . However, the actual operation counts depend on the specific cases. Sometimes, the reduction might need more operation counts when ordering of real Schur form is required. So the approximated operation counts for Schur method is order  $n^3$  [9].

# Chapter 3

## Newton-Kleinman Method

In the last chapter, we introduced the Schur method to solve ARE directly based on calculating the basis that spans the stable invariant subspace of the Hamiltonian matrix. However, for example, if the structure of the Hamiltonian matrix cannot be preserved during computation, the Schur method will be inaccurate in computing the solutions. In this chapter, we introduced an iterative scheme - Newton-Kleinman method to solve ARE.

Let us discuss some more definitions and theorems before we talk about the Newton-Kleinman method.

**Definition 3.0.1** A *Hermitian matrix* (or *self-adjoint matrix*) is a complex square matrix that is equal to its own conjugate transpose.

**Definition 3.0.2** The equation of form

$$AX + XA^H + M = 0 \tag{3.1}$$

where  $M$  is a Hermitian matrix and  $A^H$  is the conjugate transpose of  $A$  is **continuous Lyapunov equation**.

We also state a theorem about stability for continuous-time case:

**Theorem 3.0.1** Given any positive definite matrix  $M$ , there exists a unique  $P$  that is positive definite satisfying  $AX + XA^H + M = 0$  if and only if the linear system  $\dot{x} = Ax$  is globally asymptotically stable.

**Definition 3.0.3 (Definition 3.18 [13])** *The pair  $(A, B)$  is **stabilizable** if there exists  $K$  such that  $A + BK$  is Hurwitz.*

**Definition 3.0.4 (Definition 3.20 [13])** *The pair  $(A, C)$  is **detectable** if there exists  $F$  such that  $A + FC$  is Hurwitz.*

## 3.1 Solution to ARE

Assumptions:

Assume  $(A, B)$  is stabilizable and  $(C, A)$  is detectable.

For convenience, we restate the ARE

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0$$

then rewrite the above equation as

$$(A + BK)^\top P + P(A + BK) = -Q - K^\top RK \quad (3.2)$$

with  $K = -R^{-1}B^\top P$ . Recall that a matrix  $A_0$  is Hurwitz if  $\sigma(A_0) \subset \mathbb{C}_-$  where  $\sigma(\cdot)$  denotes the singular value of a real matrix. If  $A + BK$  is Hurwitz, then the above equation is a Lyapunov equation [11].

**Theorem 3.1.1 (Theorem 1.1 [15])** *Consider a stabilizable pair  $(A, B)$  with a feedback  $K_0$  so that  $A + BK_0$  is Hurwitz. Define  $S_i = A + BK_i$ , and solve the Lyapunov equation*

$$S_i^\top P_i + P_i S_i = -Q - K_i^\top R K_i$$

for  $P_i$  and then update the feedback as  $K_{i+1} = -R^{-1}B^\top P_i$ . Then

$$\lim_{i \rightarrow \infty} P_i = P$$

with quadratic convergence. The solution  $P$  is the stabilizing solution to the ARE.

To iteratively solve the Lyapunov equations, first choose an initial feedback  $K_0$  that makes  $A + BK_0$  to be Hurwitz. Denote  $S_i = A + BK_i$ . By Theorem (3.1.1), we solve for  $P_i$  in each iteration, then the sequence of solutions converge to  $P$  quadratically.

The algorithm can be summarized as follows:

---

**Algorithm 1.** Newton-Kleinman Algorithm

---

Given  $A, B, K_0, Q, R, tol_1, tol_2$   
Initialization  $S_0 = A + BK_0, \quad i = 0, \quad P = 0_{n \times n}$   
Solve  $S_0^T P_0 + P_0 S_0 = -Q - K_0^T R K_0$  for  $P_0$   
**while**  $\|P_i - P\|_F > tol_1$  or  $\|P_i - P\|_2 > tol_2$  **do**  
     $P \leftarrow P_i$   
     $i \leftarrow i + 1$   
    Update  $K_i = -R^{-1} B^T P$   
    Update  $S_i = A + BK_i$   
    Solve  $S_i^T P_i + P_i S_i = -Q - K_i^T R K_i$  for  $P_i$   
**end while**  
**return**  $P_i$

---

In this paper, three methods are discussed to solve Lyapunov equations, and two of them are used for implementation. We will introduce Smith's method, the Alternating Direction Implicit (ADI) method, and Cholesky-factor ADI method to solve large scale Lyapunov equations.

## 3.2 Solution of Lyapunov Equation

### 1. Method 1: Smith's method

Consider the Lyapunov equation

$$X A_0 + A_0^T X = -D D^T \tag{3.3}$$

where  $A_0 = A + BK_i$  is Hurwitz of  $n \times n$  in  $\mathbb{R}$  and  $D$  is a  $n \times r$  matrix in  $\mathbb{R}$  with  $r$  satisfying  $r = m(\text{observations}) + p(\text{controls})$ .  $X$  is the solution to this Lyapunov equation. Since  $A_0$  is stable, there exist a unique, symmetric positive semi-definite solution to the above equation.

The idea of Smith's method is that we rewrite the equation (3.3) as

$$(A_0^\top + qI)X(A_0 + qI) - (A_0^\top - qI)X(A_0 - qI) = -2qDD^\top$$

pre-multiply by  $(A_0^\top + qI)$  and post-multiply by  $(A_0 + qI)$ . Rearrange the term to get the update formula for  $X_i$  with  $X_0 = 0$  such that for any  $q$  in  $\mathbb{C}_-$ , we have

$$X_i = U^\top X_{i-1} U + V \quad (3.4)$$

where  $U = (A_0 - qI)(A_0 + qI)^{-1}$  and  $V = -2q(A_0^\top + qI)^{-1}DD^\top(A_0 + qI)^{-1}$  with  $q < 0$  [16].

The sequence of solutions converges to  $X$ . A formal solution [12] to (3.4) is

$$X = \sum_{k=0}^{\infty} U^k V U^{\top k} \quad (3.5)$$

Smith's method is unconditionally linear convergent for any  $q < 0$ . In each iteration, the value of  $q$  does not change. Proper choice of  $q$  can improve the convergence of the iterations [16]. The detailed parameter selection process is discussed in [12].

## 2. Method 2: ADI method

The ADI method is based on Smith's method. It is improved by selecting different  $q_i$ 's in each iteration.

Instead, we have two alternating linear systems with initial guess  $X_0 = 0_{n \times n}$  for any  $q_i$  in  $\mathbb{C}_-$  such that

$$(A_0^\top + q_i I)X_{i-\frac{1}{2}} = -DD^\top - X_{i-1}(A_0 - q_i I) \quad (3.6)$$

$$(A_0^\top + q_i I)X_i^\top = -DD^\top - X_{i-\frac{1}{2}}^\top(A_0 - q_i I) \quad (3.7)$$

for  $j = 1, 2, 3, \dots$ . The matrix  $A_0 = A + BK_i$  is Hurwitz of  $n \times n$  in  $\mathbb{R}$  and  $D$  is a  $n \times r$  matrix in  $\mathbb{R}$  with  $r$  satisfying  $r = m + p$  [16].

The updated  $X_j$  in  $j^{\text{th}}$  iteration is

$$\begin{aligned} X_j &= -2q_i(A_0^\top + q_i I)^{-1}DD^\top(A_0 + q_i I)^{-1} \\ &\quad + (A_0^\top + q_i I)^{-1}(A_0^\top - q_i I)X_{j-1}(A_0 - q_i I)(A_0 + q_i I)^{-1} \end{aligned}$$

Here is the algorithm that describes the ADI iterations:

---

**Algorithm 2.** Alternating Direction Implicit Algorithm [11]

---

Given  $A_0, D$   
Initialization  $j = 0, X_0 = 0_{n \times n}$   
Choose ADI parameters,  $\{q_1, q_2, \dots, q_J\}, \text{Re}\{q_i\} < 0$   
**for**  $j = 1, 2, \dots, J$ , **do**  
    Update  $(A_0^\top + q_i I)X_{i-\frac{1}{2}} = -DD^\top - X_{i-1}(A_0 - q_i I)$   
    Update  $(A_0^\top + q_i I)X_i^\top = -DD^\top - X_{i-\frac{1}{2}}^\top(A_0 - q_i I)$   
**end for**  
**return**  $X_J$

---

Proper choice of  $q_i$  can improve the convergence of the iterations, say  $J$  iterations where  $J \ll n$ . The parameters selection in ADI method is discussed as follows.

#### ADI Parameter Selection

We solve the min-max problem to find the optimal ADI parameters

$$\{q_1, q_2, \dots, q_J\} = \arg \min_{q_i} \max_{\lambda_j \in \sigma(A_0)} \left| \prod_j^J \frac{q_i - \lambda}{q_i + \lambda} \right|$$

where  $J$  represents the max iteration number and the min-max problem is a function of  $J$ .

The solution is completely determined by the eigenvalues of  $A_0$ . If  $A_0$  has strictly real eigenvalues, the solution is easily calculated [12]. If the eigenvalues of  $A_0$  are in the open left half plane, the optimal parameters can be approximated by several ways. We will discuss one way here [12].

Define the spectral bounds  $a, b$  and a sector angle  $\alpha$  for the matrix  $A_0$  [2]

$$\begin{aligned} a &= \min_i (\text{Re}(\lambda_i)) \\ b &= \max_i (\text{Re}(\lambda_i)) \\ \alpha &= \tan^{-1} \max_i \left| \frac{\text{Im}(\lambda_i)}{\text{Re}(\lambda_i)} \right| \end{aligned}$$

where  $\lambda_i$  are the the eigenvalues of  $A_0$ . The parameter  $\alpha$  is the maximum angle from the real axis to the eigenvalue. Define

$$\cos^2\beta = \frac{2}{1 + \frac{1}{2}(\frac{a}{b} + \frac{b}{a})}, \quad m = \frac{2\cos^2\alpha}{\cos^2\beta} - 1 \quad (3.8)$$

If  $\alpha < \beta$ , then  $m \geq 1$ , it is equivalent to the parameters  $p_j$  are real. Let

$$k' = \frac{1}{m + \sqrt{m^2 - 1}}, \quad k = \sqrt{1 - k'^2}$$

and  $k' = \frac{a}{b}$  if the eigenvalues of  $A_0$  are real.

Define the elliptic integrals  $K$  and  $v$  as

$$F[\psi, k] = \int_0^\psi \frac{1}{\sqrt{1 - k^2 \sin^2 x}} dx$$

$$K = K(k) = F[\frac{\pi}{2}, k], \quad v = F[\sin^{-1} \sqrt{\frac{a}{bk'}}, k']$$

where  $K$  is the incomplete elliptic integral of the first kind,  $k$  is its modulus, and  $\psi$  is its amplitude.

**Definition 3.2.1** ([19]) *The elliptic function region corresponding to  $A_0$  is defined as*

$$D(r) = \{p = dn(zK, k) | z = x + iy, 0 \leq x \leq 1 \text{ and } |y| \leq r\} \subset \mathbb{C}$$

where  $r = \alpha/K$ ,  $k$ ,  $K = K(k)$  are as above, and  $dn$  is the Jacobi elliptic function.

The maximum number of iterations is

$$J = \lceil \frac{K}{2v\pi} \log \frac{4}{\epsilon_1} \rceil \quad (3.9)$$

with ADI parameters

$$p_j = -\sqrt{\frac{ab}{k'}} dn[\frac{(2j-1)K}{2J}, k], \quad j = 1, 2, \dots, J \quad (3.10)$$

where  $dn(u, k)$  is the Jacobi elliptic function[12].



If  $m < 1$ , the parameters  $p_j$  are complex. We define the dual elliptic spectrum,  $a'$ ,  $b'$  and  $m'$  such that

$$\begin{aligned} a' &= \tan(\pi/4 - \alpha/2) \\ b' &= 1/a' \\ m' &= \frac{2\cos^2\beta}{\cos^2\alpha} - 1 \end{aligned}$$

by construction,  $m'$  must be greater than 1 now so that we can get the optimum real parameters  $p'_j$  for the dual problem. Substituting  $a'$  in (3.8), we get

$$\beta' = \alpha, \quad m' = \frac{2\cos^2\beta}{\cos^2\alpha} - 1$$

The complex parameters  $p_j$  can be obtained from

$$\cos\alpha_j = \frac{2}{p'_j + \frac{1}{p'_j}}$$

so that

$$p_{2j-1} = \sqrt{abe^{i\alpha_j}}, \quad p_{2j} = \sqrt{abe^{-i\alpha_j}}, \quad j = 1, 2, \dots, \lceil \frac{1+J}{2} \rceil [2].$$

### 3. Method 3: Cholesky-factor ADI method

**Definition 3.2.2** A matrix  $Z$  of size  $n \times p$  in  $\mathbb{R}$  is a **Cholesky factor** of the self-adjoint matrix  $X$  if it satisfies

$$X = ZZ^\top.$$

where  $X$  is a real, symmetric matrix of size  $n \times n$ .

Based on ADI method, only the Cholesky factor  $Z$  is calculated during the iterations in the Cholesky-factor ADI method. Therefore, if the matrices are sparse or structured, we can take advantage of this properties and be more computationally efficient [11].

Instead of iteratively solving  $X_i$ , we solve  $Z_i$  such that  $X_i := Z_i Z_i^\top$  where  $Z_i$  is the Cholesky factor of  $X_i$ .

The iterations for  $Z_i$  are as follows:

$$Z_1 = \sqrt{-2q_1}(A_0^\top + q_1I)^{-1}D, \quad Z_1 \in \mathbb{R}^{n \times r} \quad (3.11)$$

$$Z_i = [\sqrt{-2q_i}(A_0^\top + q_iI)^{-1}D, (A_0^\top + q_iI)^{-1}(A_0^\top - q_iI)Z_{i-1}] \quad (3.12)$$

with  $Z_i$  of size  $n \times ir$  in  $\mathbb{R}$  for all  $i$ . The matrix  $A_0 = A + BK_i$  is Hurwitz of  $n \times n$  in  $\mathbb{R}$  and  $D$  is a  $n \times r$  matrix in  $\mathbb{R}$ .

Observe from the above formulation, at  $j^{\text{th}}$  iteration, multiply the previous Cholesky factor  $Z_{j-1}$  on the left by  $(A_0^\top + q_jI)^{-1}(A_0^\top - q_jI)$ , the column size of Cholesky factor is increased by  $r$  at a time[11].

Explicitly write the Cholesky factor out at  $J$ 's iteration as

$$Z_J = [z_J, \quad Q_{J-1}z_J, \quad Q_{J-2}Q_{J-1}z_J, \dots, \quad Q_1Q_2\dots Q_{J-1}z_J] \quad (3.13)$$

where

$$z_J := \sqrt{-2q_J}(A_0^\top + q_JI)^{-1}D$$

$$Q_m := \left( \frac{\sqrt{-2q_m}}{\sqrt{-2q_{m+1}}} \right) (A_0^\top + q_mI)^{-1} (A_0^\top - q_{m+1}I)$$

then the solution is  $X = Z_i Z_i^\top$  for any  $i \leq J$ .

Here is the algorithm that describes the Cholesky-factor ADI iterations:

---

**Algorithm 3.** Cholesky-factor Alternating Direction Implicit Algorithm [11]

---

Given  $A_0, D, tol$   
 Initialization  $j = 0, X_0 = 0_{n \times n}$   
 Choose ADI parameters,  $\{q_1, q_2, \dots, q_J\}, \text{Re}\{q_i\} < 0$   
 Define  $z_1 = \sqrt{-2q_1}(A_0 + q_1I)^{-1}$   
 Define  $Z_1 = [z_1]$   
**for**  $j = 2, 3, \dots, J$ , **do**  
     Update  $P_{j-1} = \left(\frac{\sqrt{-2q_j}}{\sqrt{-2q_{j-1}}}\right)[I - (q_j + q_{j-1})(A^\top + q_jI)^{-1}]$   
     Update  $z_j = P_{j-1}z_{j-1}$   
     **if**  $\|z_j\|_2 > tol$  &  $j \leq J$  **then**  
          $Z_j = [Z_{j-1}, z_j]$   
     **else**  
         Stop  
     **end if**  
**end for**  
**return**  $X \approx X_J = Z_J Z_J^\top$

---

This algorithm requires proper selection of the parameters  $q_j$  when  $A_0$  has complex spectra [12] [16] [3].

### 3.2.1 Numerical Issues

#### Operation Counts and Timing in Solving Lyapunov equations:

1. **ADI method:** The tridiagonalization of matrix  $A_0$  that is not sparse or structured, and the transformation to get back to ADI approximation takes  $\mathcal{O}(n^3)$ . Combining the system of equations (3.7) for each iteration takes  $\mathcal{O}(n^2 + n) \approx \mathcal{O}(n^2)$ , and  $J$  iterations take  $\mathcal{O}(Jn^2)$ . So the work for ADI algorithm is typically  $\mathcal{O}(n^3) + \mathcal{O}(Jn^2)$ . If  $A_0$  is sparse, the total work is  $\mathcal{O}(Jn^2)$  [11].
2. **Cholesky-factor ADI method:** By equation (3.13), the work to obtain  $Z_J$  of  $r$  columns is about  $\mathcal{O}((2n - 1)nr) \approx \mathcal{O}(n^2r)$ . And the total work for  $J$ 's iteration is about  $\mathcal{O}(n^2 Jr)$ .

If  $A_0$  is sparse or structured, the Cholesky-factor ADI always results in substantial savings, the total work is reduced from  $\mathcal{O}(n^2 Jr)$  to  $\mathcal{O}(n Jr)$  [11].

Table 3.1 summarizes the complexity of these two methods in solving Lyapunov equations where  $J$  is the maximum number of iterations,  $r$  is the increased column size of the Cholesky factor in each iteration [16].

	Cholesky-factor ADI	ADI
Sparse $A_0$	$\mathcal{O}(n Jr)$	$\mathcal{O}(Jn^2)$
Full $A_0$	$\mathcal{O}(n^2 Jr)$	$\mathcal{O}(n^3) + \mathcal{O}(Jn^2)$

Table 3.1: ADI and Cholesky-factor ADI method complexity comparison in solving Lyapunov equations

# Chapter 4

## Example: Simply Supported Beam

We will use MATLAB to generate the computational results with 1.4 GHz Quad-Core Intel Core i5 processor. We will compare how Schur method and Newton-Kleinman method perform in controller design for approximated simply supported beam.

Consider a simply supported Euler-Bernoulli beam. The deflection of the beam from its rigid body motion is denoted by  $w(x, t)$ , where  $t$  is time and  $x$  is the position. We apply a control (force) at position  $r = 0.55$  with width  $\epsilon_1 = 0.01$ . We normalize the variables, also include a viscous damping  $\xi = 1$  and the stiffness  $\epsilon = 40$ . The governing partial differential equation is

$$\frac{\partial^2}{\partial t^2}w(x, t) + \xi \frac{\partial}{\partial t}w(x, t) + \epsilon \frac{\partial^4}{\partial x^4}w(x, t) = b_{.55}(x)u(t), 0 < x < 1$$
$$b_{.55} = \begin{cases} \frac{1}{\epsilon_1} & |.55 - x| < \frac{\epsilon_1}{2} \\ 0 & |.55 - x| \geq \frac{\epsilon_1}{2} \end{cases}$$

with boundary conditions

$$w(0, t) = 0, \frac{\partial^2}{\partial t^2}w(0, t) = 0, w(1, t) = 0, \frac{\partial^2}{\partial t^2}w(1, t) = 0$$

Consider the linear Quadrature Regulator (LQR) problem. The objective is to minimize

the cost functional

$$J(u(r)) = \frac{1}{2} \int_0^{+\infty} \langle Q^{1/2}w(r), Q^{1/2}w(r) \rangle + u^\top(r)Ru(r)dr$$

subject to this partial differential equation with viscous damping and the boundary conditions. We will choose  $R = 1$  and  $Q = I$  in the computer simulations.

Since the closed form solution of this partial differential equation is not available, an approximation scheme in a finite-dimensional space is used for the controller design [6]. We will approximate the problem with a system of ordinary differential equations (ODEs).

Define the operator  $L(\cdot) = \frac{\partial^4}{\partial x^4}(\cdot)$ . Let  $\lambda_n$  and  $\phi_n(x)$  be the eigenvalues and the eigenfunctions associated to the operator  $L: L\phi_n(x) = \lambda_n\phi_n(x)$  with boundary conditions. Also convert the problem to the following ODE:

$$\epsilon Lw + \ddot{w} + \xi\dot{w} = 0 \tag{4.1}$$

the solution can be written as  $w(x; t) = \sum_{n=1}^{\infty} \phi_n(x)w_n(t)$  where  $\phi_n(x)$  is the  $n^{th}$  order beam eigenfunction. Substitution of this into equation (4.1) yields

$$\epsilon L(\sum_{n=1}^{\infty} \phi_n(x)w_n) = -(\sum_{n=1}^{\infty} \phi_n(x)\ddot{w}_n) - \xi(\sum_{n=1}^{\infty} \phi_n(x)\dot{w}_n)$$

We have the following ODE for each  $n$  such that

$$\epsilon\lambda_n w_n = -\ddot{w}_n - \xi\dot{w}_n$$

Define

$$\frac{\partial^4}{\partial x^4} \phi_n = \lambda_n \phi_n := r_n^4 \phi_n$$

The solution to this differential equation with boundary conditions is

$$\phi_n(x) = \sqrt{2} \sin(n\pi x)$$

and  $r_n = n\pi$  so that  $\lambda_n = r_n^4 = (n\pi)^4$ .

In this simulations, the *tol* in the Newton-Kleinman method is set to  $10^{-3}$ . The built-in function *care()* is used to implement the Schur method. The built-in function *lyap()* is

used to solve the Lyapunov equations in the Newton-Kleinman method. Here are some results generated using MATLAB simulations:

$n$	Schur method Time in seconds
24	0.001821
48	0.005281
72	0.018509
80	0.020112
96	0.026565
120	0.033545
144	0.041117
192	0.072092
200	–
250	–

Table 4.1: Time per order  $n$  for Schur method with  $Q = I, R = 1$

Table 4.1 shows the computation time in seconds per order  $n$  for Schur method to get the solutions to ARE. The Schur method *care()* fails for problem order over 200. MATLAB reported that the solution may be inaccurate due to poor scaling or eigenvalues near the stability boundary when the problem order  $n$  is above 200.

In Table 4.2, we computed the Frobenius norm relative residual for Schur method. As the problem order gets larger, the relative residual gets larger. The Schur method fails for problem order over 200.

Table 4.3 shows the number of Newton-Kleinman iterations, the computation time and the Frobenius norm relative residual for Newton-Kleinman method per order  $n$ . There are more Newton-Kleinman iterations required for problem order  $n$  over 80. It is observed that it required more than 7 hours. Due to the hardware limitation, the exact iteration numbers, timing and relative residual are not observed. It is also observed that the Frobenius norm relative residual stays around  $e-8$ , which is much smaller than the relative residual generated using the Schur method for the same problem order  $n$ .

Figure 4.1 shows the relative 2-norm error calculated using  $\frac{\|P - P_j\|_2}{\|P\|_2}$  for each Lyapunov iteration at problem order 24. The relative error in the last Newton-Kleinman iteration is  $4.2603e-10$ , which is close to machine epsilon.

Table 4.4 shows the approximated optimal feedback gain at each Newton-Kleinman iteration for problem order 24. The approximated feedback gain stabilizes at a number in

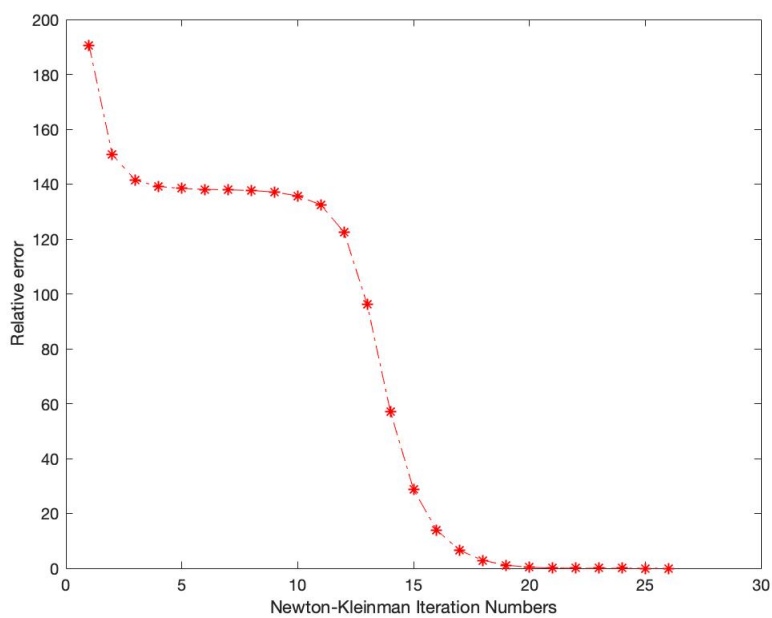


Figure 4.1: The relative error calculated using  $\frac{\|P - P_j\|_2}{\|P\|_2}$  in each Newton-Kleinman iteration at problem order 24.  $P_j$  is the solution at  $j^{th}$  Newton-Kleinman iteration.  $P$  is the solution to ARE approximated by Newton-Kleinman method



$n$	Schur method error
24	3.8176e-11
48	2.4194e-08
72	5.9503e-07
80	5.6398e-06
96	6.7965e-06
120	1.0576e-04
144	2.0665e-04
192	6.5597e-04
200	0.0029
250	0.0104

Table 4.2: The Frobenius norm relative residual calculated using  $\frac{\|Ric(P)\|_F}{\|P\|_F}$  for Schur method.  $P$  is the solution to ARE obtained from *care()*

the last three iterations.

Figure 4.2 shows the Frobenius norm relative residual  $\frac{\|Ric(P)\|_F}{\|P\|_F}$  per order  $n$  when substituting the solution  $P$  into the ARE. Here  $Ric(P) = A^\top P + PA - PBR^{-1}B^\top P + Q$ .  $P$  is the solution to ARE approximated by Schur method. It is observed that the relative residuals are increasing as problem order gets larger.

$n$	Newton-Kleinman Itn's	Time in seconds	N-K error
24	26	0.016150	4.0461e-8
48	37	0.049841	5.7367e-10
72	96	0.203994	2.4613e-8
76	140	0.520008	3.1225e-8
78	810	3.473147	6.7848e-8
80	57166	327.581468	1.2437e-8
82	-	> 7 hours	-

Table 4.3: The iteration numbers, timing and the Frobenius norm relative residual calculated using  $\frac{\|Ric(P)\|_F}{\|P\|_F}$  for Newton-Kleinman method at different problem order with  $Q = I, R = 1, K_0 = 0$ .  $P$  is the solution to ARE approximated using Newton-Kleinman method

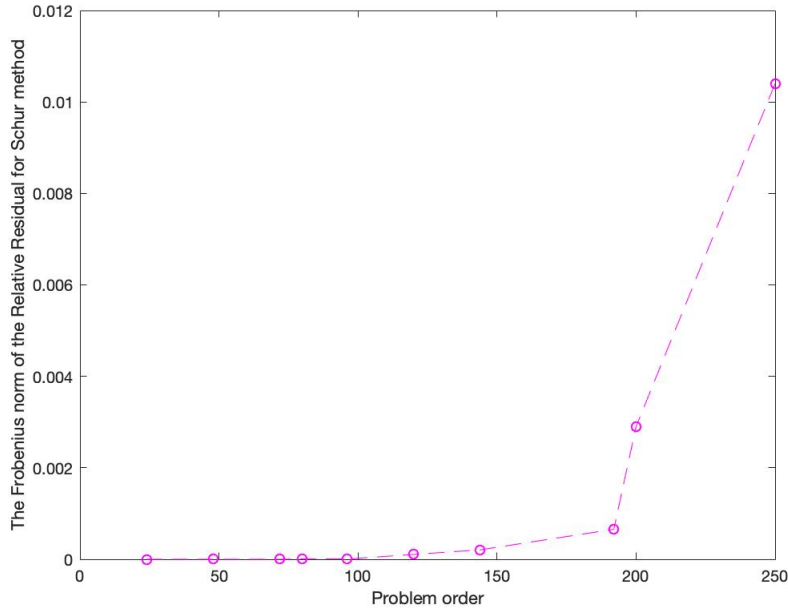


Figure 4.2: The Frobenius norm relative residual calculated using  $\frac{\|Ric(P)\|_F}{\|P\|_F}$  for Schur method per order  $n$ .  $P$  is the solution to ARE obtained from `care()`.

Newton-Kleinman Itn's	Optimal Feedback Gain
1	0
2	1.0970e+06
3	5.4850e+05
4	2.7425e+05
5	1.3713e+05
6	6.8563e+04
7	3.4282e+04
8	1.7147e+04
9	8.6170e+03
10	4.6397e+03
11	4.0199e+03
12	5.9684e+03
13	8.5923e+03
14	1.1034e+04
15	1.2989e+04
16	1.4190e+04
17	1.4822e+04
18	1.5134e+04
19	1.5234e+04
20	1.5055e+04
21	1.4247e+04
22	1.2525e+04
23	1.1034e+04
24	1.0593e+04
25	1.0566e+04
26	1.0566e+04

Table 4.4: Feedback Gain using 2-norm at each Newton-Kleinman iteration on problem order 24 with  $Q = I, R = 1, K_0 = 0$

# Chapter 5

## Comparisons and Observations

In Schur vector approach, explicitly computing the eigenvalues and the eigenvectors is not needed. The computation of eigenvalues and eigenvectors suffers from severe numerical issues especially when the problem orders become large. All we need to compute is just a basis that spans the stable invariant subspace of the Hamiltonian matrix  $Z$  [9].

However, Schur method can have problems if the Hamiltonian matrix has eigenvalues that are near imaginary axis. The Schur method even gives wrong results sometimes if it fails to accurately isolate the stable invariant subspace of the corresponding Hamiltonian matrix. There are some algorithms that can preserve the structure of the Hamiltonian matrix to guarantee the accuracy of the solutions, but these methods are very expensive in practice [6]. In simply supported beam example, the Schur method yields inaccurate results for problem order over 200 by MATLAB results.

Compared with the Schur method, the advantage of the Newton-Kleinman method is the reliability of the solution. Numerically speaking, the solution is more reliable in computation to reach the limiting accuracy. Once the initial condition is chosen and some necessary assumptions hold, a series of Lyapunov equations can be recursively constructed at each iteration. The feedback gain stabilizes to a number. The positive semi-definite solutions are preserved and converge to the stabilizing solution [8].

### 5.1 Observations

It is observed from the simply supported beam example that the Schur method fails for problem order over 200. For Newton-Kleinman method, it requires 57166 iterations with

the problem order 80.

It is also observed that with the increasing model orders, the relative residuals from the Schur method are increasing. Inaccurate solutions are obtained along with large relative residuals for problem order over 200.

The Frobenius norm relative residual for Newton-Kleinman method stays around  $e-8$  for problem order up to 80. Compared with the Schur method for the same problem order, the solution approximated by Newton-Kleinman method is more reliable in computation to reach the limiting accuracy.

The Schur method is very fast with accurate solutions when the problem orders are under 200 while Newton-Kleinman method is more accurate but requires more time.

# References

- [1] W. F. Arnold and A. J. Laub. “Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations”. *Proceedings of the IEEE*, 72(12):1746–1754, 1984.
- [2] P. Benner, H. Mena, and J. Saak. “On the Parameter Selection Problem in the Newton-ADI Iteration for Large-scale Riccati Equations”. *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, 29:136–149, 2007.
- [3] N. S. Ellner and E. L. Wachspress. “Alternating Direction Implicit Iteration for Systems with Complex Spectra”. *SIAM Journal on Numerical Analysis*, 28(3):859–870, 1991.
- [4] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computations*. Prentice Hall Professional Technical Reference, 1977.
- [5] G. H. Golub and J. H. Wilkinson. “Ill-conditioned Eigensystems and the Computation of the Jordan Canonical Form”. *SIAM Rev.*, 18(4):578–619, 1976.
- [6] D. Kasinathan, K. A. Morris, and S. Yang. “Solution of Large Descriptor  $\mathcal{H}_\infty$  Algebraic Riccati Equations”. *Journal of Computational Science*, 5:517–526, 2012.
- [7] D. Kleinman. “On an Iterative Technique for Riccati Equation Computations”. *IEEE Transactions on Automatic Control*, 13(1):114–115, 1968.
- [8] A. Lanzon, Y. Feng, B. Anderson, and M. Rotkowitz. “Computing the Positive Stabilizing Solution to Algebraic Riccati Equations with an Indefinite Quadratic Term via a Recursive Method”. *IEEE Transactions on Automatic Control*, 53(10):2280 – 2291, 2008.
- [9] A. Laub. “A Schur Method for Solving Algebraic Riccati Equations”. *IEEE Transactions on Automatic Control*, 24(6):913–921, 1979.

- [10] A. J. Laub. “Schur Techniques in Invariant Imbedding Methods for Solving Two-point Boundary Value Problems”. In *1982 21st IEEE Conference on Decision and Control*, pages 56–61. IEEE, 1982.
- [11] J. R. Li and J. White. “Low-Rank Solution of Lyapunov Equations”. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.
- [12] A. Lu and E. L. Wachspress. “Solution of Lyapunov Equations by Alternating Direction Implicit Iteration”. *Computers & Mathematics with Applications*, 21(9):43 – 58, 1991.
- [13] K. A. Morris. *Introduction to Feedback Control*. Harcourt-Brace Academic Press, 2000.
- [14] K. A. Morris. “Control of Systems Governed by Partial Differential Equation”. In W. S. Levine, editor, *The IEEE Control Theory Handbook*. CRC Press, 2010.
- [15] K. A. Morris and C. Navasca. “Iterative Solution of Algebraic Riccati Equations for Damped Systems”. *Proceedings of the IEEE Conference on Decision and Control*, pages 2436 – 2440, 2007.
- [16] K. A. Morris and C. Navasca. “Solution of Algebraic Riccati Equations Arising in Control of Partial Differential Equations”. *Control and Boundary Analysis*, 240, 2007.
- [17] R. A. Smith. “Matrix Equation  $XA + BX = C$ ”. *SIAM Journal of Applied Mathematics*, 16(1):198–201, 1968.
- [18] G. W. Stewart. “Algorithm 506: HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of A Real Upper Hessenberg Matrix [F2]”. *ACM Trans. Math. Softw.*, 2(3):275–280, 1976.
- [19] E. L. Wachspress. *The ADI model problem*. Springer, 2013.