NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THE ITERATIVE METHODS FOR COMPUTING INVARIANT SUBSPACES AND THEIR APPLICATIONS

by

Young Mee Lee

B.Sc., Yeon Sei University, 1985

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Mathematics and Statistics

SIMON FRASER UNIVERSITY

August 1987

ISBN 0-315-42615-2

APPROVAL

Name: Young Mee Lee

Degree: MASTER OF SCIENCE

Title of thesis: THE ITERATIVE METHODS FOR COMPUTING INVARIANT SUBSPACES AND THEIR APPLICATIONS

Examining Committee:

Chairman: Dr. J. J. Sember

Dr. R. D. Russell
Senior Supervisor

Dr. E. Pechlaner

Dr. L. Dieci

Dr. B. K. Bhattacharya
External Examiner
Department of Computing Science
Simon Fraser University

Date Approved: Aug., 7, 1987

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

THE ITERATIVE METHODS FOR COMPUTING INVARIANT SUBSPACES AND THEIR APPLICATIONS

Author:

(signature)

YOUNG MEE LEE

(name)

Aug 11 1987

(date)

ABSTRACT

Iterative methods for computing invariant subspaces and their various applications, which appear in many fields, are considered. Three iterative schemes which have been suggested are those by Dongarra-Moler-Wilkinson, Stewart and Chatelin. Recently, Demmel has shown that they all eventually reduce to solving the same equation, the algebraic Riccati equation, which has been deeply studied in mathematics and engineering. This leads us to directly consider numerical methods for solving Riccati equations as methods for the invariant subspace problem.

A few numerical methods for solving Riccati equations have been proposed: Kokotovic's iterative method, the Schur method, the linear convergence method, the generalized secant method and Newton's method. In this thesis, we analyze and implement these methods. We give numerical details of the various algorithms and rigorous operation counts for each of them. Our comparison shows that the generalized secant and Newton's methods are competitive with both Kokotovic's and the Schur methods, even in situations where the latter methods are known to be efficient and reliable, as in solving Riccati equations arising from singular perturbation and control problems, respectively. Our analysis also shows that the generalized secant method is generally more efficient than the other two iterative methods.

The potential drawback of iterative methods is the need of a good initial guess. While showing that, in practice, the convergence regions predicted by the theory are rather restrictive, we have also discussed and implemented matrix algorithms of steepest descent type, which when used in conjunction with the generalized secant method and Newton's method produce very appealing results.

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor, Dr. R. D. Russell for his suggestions, guidance and constant encouragement.

I would like to thank Dr. L. Dieci who has given generously of his time and knowledge in discussions of various aspects of the problem.

The support received from Dr. Russell's research grant is also appreciated.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{1.1}$$

where $A_{11} \in \mathbf{R}^{n\times n}$ and $A_{22} \in \mathbf{R}^{m\times m}$. A subspace $X \subset \mathbf{R}^{n+m}$ with the property that

$$x \in X \;\rightarrow\; Ax \in X$$

is called an *invariant subspace* of A. We now have two basic questions: How can we compute invariant subspaces? Why do we need to compute invariant subspaces?

The main purposes of this thesis are to analyze and implement methods for computing invariant subspaces, especially iterative methods which are motivated from the *Riccati equation*:

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R, \tag{1.2}$$

and to show their various applications.

In this chapter, we briefly state two important theorems which are often used through this thesis. In chapter 2, we describe three iterative methods for computing invariant subspaces which are suggested by Dongarra-Moler-Wilkinson [1], Stewart [2] and Chatelin [3], and then we briefly discuss Demmel's recent paper [4] which shows that all three methods eventually amount to solving the same equation, the algebraic Riccati equation (1.2).

It is well-known that one cannot expect to compute the accurate eigenvectors when their corresponding eigenvalues belong to a cluster of poorly separated eigen-

values. Also when A is defective, that is, the Jordan canonical form is not strictly diagonal, the computed eigenvectors corresponding to the relevant eigenvalues will be almost linearly dependent. However, the invariant subspace associated with an ill-conditioned eigenproblem which has close eigenvalues and/or almost parallel eigenvectors is well determined provided the cluster is well separated from the remaining eigenvectors. Thus it is advisable to group some eigenvalues and to compute a basis of the corresponding invariant subspace instead of computing eigenvectors. This is the well-known and important role of an invariant subspace. Naturally, we have a general question: Even if eigenproblem is well-conditioned, is it sufficient to only compute an invariant subspace? In chapter 3, we discuss the various applications of invariant subspaces which appear in singular perturbation and control problems and boundary value problems for ordinary differential equations.

In chapter 4, we consider the Schur method [14]. This is the one of methods for solving algebraic Riccati equations arising from control problems.

In chapter 5, we consider three iterative methods, namely, the linear convergence method, Newton's method and the generalized secant method, which are called LCM, NEM and SEM, respectively. We give numerical details of the various algorithms and rigorous operation counts for each of them. We also state the rate of convergence, condition number, advantages and disadvantages for each method. We choose several examples to compare these three methods with Kokotovic's iterative method and the Schur method (we call KIM and SCM, respectively), which are known to be reliable and efficient methods for solving Riccati equations in singular perturbation and control problems, respectively, present some numerical results of interest, and make comparisons.

The main advantage of SEM and NEM for solving Riccati equations is their speed of convergence once a sufficiently accurate approximation is known. In chapter 6, we implement matrix algorithms of steepest descent type which give a good starting value.

Conclusions drawn from the research and suggestions for the further study are given in chapter 7.

Theorem: Real Schur Decomposition ([22])

If $A \in \mathbf{R}^{l\times l}$ then there exists an orthogonal $U \in \mathbf{R}^{l\times l}$ such that U^TAU is quasi-upper-triangular. Futhermore, U can be chosen so that 2×2 and 1×1 diagonal blocks appear in any desired order.

□

Theorem ([4])

Let $A \in \mathbf{R}^{l\times l}$ and $X = (X_1, X_2) \in \mathbf{R}^{l\times l}$ and define

$$X^{-1}AX = \begin{pmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{pmatrix}$$

where $X_1 \in \mathbf{R}^{l\times n}$, $A'_{11} \in \mathbf{R}^{n\times n}$, $A'_{22} \in \mathbf{R}^{m\times m}$ and $l = n+m$. The range of X_1 denoted by $R(X_1)$ is an invariant subspace if and only if $A'_{21} = 0$.

□

CHAPTER 2

THE ITERATIVE METHODS FOR COMPUTING INVARIANT SUBSPACES

In this chapter, we will describe methods for refining estimates of an invariant subspace which have been devised by Dongarra-Moler-Wilkinson [1], Stewart [2] and Chatelin [3]. These three methods (henceforth called DMW, S and C, respectively), all solve apparently different equations, since they represent the desired invariant subspace slightly differently. However, by a simple change of basis Demmel [4] shows that all three methods are attempting to solve the same equation, the *Riccati equation*, which is our main concern. Thus this chapter shows how computing invariant subspaces is related to solving algebraic Riccati equations and gives us the idea that numerical methods for solving algebraic Riccati equations may be considered as methods for computing invariant subspaces. The details of iterative methods for solving Riccati equations will be presented in chapter 5.

2.1 The method DMW

The method DMW which is devised by Dongarra, Moler and Wilkison is a computational method for improving the numerical accuracy of matrix eigenvalues and eigenvectors. They extend this method to determine invariant subspaces.

2.1.1 When A is *nondefective*

First consider two initial approximate eigenpairs λ_1, x_1 and λ_2, x_2 where $|\lambda_1-\lambda_2|/\|A\|$ is small and x_1 and x_2 are linearly independent. Although x_1 and x_2 may have substantial errors, they should belong to the appropriate two-space. Hence we have

$$A(x_1+\bar{y}_1) = (\lambda_1+\mu_{11})(x_1+\bar{y}_1) + \mu_{21}(x_2+\bar{y}_2), \qquad (2.1)$$

$$A(x_2+\bar{y}_2) = \mu_{12}(x_1+\bar{y}_1) + (\lambda_2+\mu_{22})(x_2+\bar{y}_2)$$

where $\bar{y}_1$, $\bar{y}_2$ and μ_{ij} are expected to be small. Because (2.1) implies that

$$A[x_1+\bar{y}_1, x_2+\bar{y}_2] = [x_1+\bar{y}_1, x_2+\bar{y}_2]\begin{pmatrix} \lambda_1+\mu_{11} & \mu_{12} \\ \mu_{21} & \lambda_2+\mu_{22} \end{pmatrix},$$

the vectors $x_1+\bar{y}_1$, $x_2+\bar{y}_2$ span the exact invariant subspace of A, the corresponding eigenvalues being those of the 2×2 matrix on the right.

The procedure for computing $\bar{y}_1$, $\bar{y}_2$ and μ_{ij} is as follows:

We assume that $\|x_1\|_\infty = \|x_2\|_\infty = 1$. To select specific vectors in the space, we must prescribe some form of normalization of $x_1+\bar{y}_1$ and $x_2+\bar{y}_2$. We shall require $\bar{y}_1$ and $\bar{y}_2$ such that $\bar{y}_{1p} = \bar{y}_{2p} = \bar{y}_{1q} = \bar{y}_{2q} = 0$, where p and $q(p \neq q)$ are such that

$$|x_{1p}| = \max|x_{1i}| = \|x_1\|_\infty$$

and

$$|x_{1p}x_{2q} - x_{1q}x_{2p}| = \max|x_{1p}x_{2i} - x_{1i}x_{2p}|.$$

From (2.1) we obtain

$$(A-\lambda_1 I)\bar{y}_1 - \mu_{11}x_1 - \mu_{21}x_2 = r_1 + \mu_{11}\bar{y}_1 + \mu_{21}\bar{y}_2, \qquad (2.2)$$

$$(A-\lambda_2 I)\bar{y}_2 - \mu_{12}x_1 - \mu_{22}x_2 = r_2 + \mu_{12}\bar{y}_1 + \mu_{22}\bar{y}_2$$

where $r_i = \lambda_i x_i - Ax_i$, $i=1,2$. Define y_1 and y_2 by

$$y_1 = \bar{y}_1 + \mu_{11}e_p + \mu_{21}e_q,$$

$$y_2 = \bar{y}_2 + \mu_{12}e_p + \mu_{22}e_q$$

so that y_i gives the full information on both μ_{ij} and $\bar{y}_i$, where the p-th component of e_p and the q-th component of e_q are 1 and the rest of them are all zero. Then (2.2) becomes

$$B_i y_i = r_i + y_{ip}\bar{y}_1 + y_{iq}\bar{y}_2, \qquad i=1,2 \qquad (2.3)$$

where B_i is $A-\lambda_i I$ with columns p and q replaced with $-x_1$ and $-x_2$. Now (2.3) can be solved by the iterative procedure

$$B_i y_i^{k+1} = r_i + y_{ip}^k \bar{y}_1^k + y_{iq}^k \bar{y}_2^k, \qquad i=1,2, \tag{2.4}$$

$$y_i^0 = 0.$$

This is the case for two simple eigenvalues. The iterative method (2.4) extends to a set of $s(s=3,4, \cdots, l)$ close eigenvalues. To see this, consider the initial approximate values $\lambda_1, x_1, \ldots, \lambda_s, x_s$, where $x_1, x_2, \ldots, x_s$ are linearly independent. Then we have

$$A[x_1+\bar{y}_1, \ldots, x_s+\bar{y}_s] = [x_1+\bar{y}_1, \ldots, x_s+\bar{y}_s][diag(\lambda_i)+M] \tag{2.5}$$

where $m_{ij} = \mu_{ij}$ and $\bar{y}_i$ and μ_{ij} are expected to be small. We know that $[x_1+\bar{y}_1, \ldots, x_s+\bar{y}_s]$ span the exact invariant subspace of A.

To find $\bar{y}_i$ and μ_{ij}, we first describe how to determine the s elements of $\bar{y}_i$ that are to be zero. Let X be the $s \times l$ matrix with rows x_i. Let this be reduced to upper triangular form using Gaussian elimination with column pivoting. If the pivotal elements are in columns $p_1, p_2, \ldots, p_s$ respectively then these elements are to be zero in $\bar{y}_i$. Define y_i by

$$y_i = \bar{y}_i + \mu_{1i} e_{p_1} + \cdots + \mu_{si} e_{p_s}, \qquad i=1, \ldots, s$$

Then (2.5) becomes

$$B_i y_i = r_i + y_{ip_1} \bar{y}_1 + \cdots + y_{ip_s} \bar{y}_s, \qquad i=1, \ldots, s \tag{2.6}$$

where B_i is $A-\lambda_i I$ with s of its columns replaced by $-x_1, \ldots, -x_s$. (2.6) now can be solved by the iterative procedure

$$B_i y_i^{k+1} = r_i + y_{ip_1}^k \bar{y}_1^k + \cdots + y_{ip_s}^k \bar{y}_s^k, \qquad i=1, \ldots, s,$$

$$y_i^0 = 0.$$

2.1.2 When A is *defective*.

We know that if A is defective then the computed eigenvectors to the relevant eigenvalues will be almost linearly dependent. So consider two well separated approximate generators x_1, x_2 of the invariant subspace instead of the eigenvectors and a 2×2 matrix M. Then

$$A[x_1 , x_2] \approx [x_1 , x_2]M \approx [x_1 , x_2]\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}.$$

We now attempt to determine $\bar{y}_1$, $\bar{y}_2$ and μ_{ij} such that

$$A[x_1+\bar{y}_1 , x_2+\bar{y}_2] = [x_1+\bar{y}_1 , x_2+\bar{y}_2]\begin{bmatrix} m_{11}+\mu_{11} & m_{12}+\mu_{12} \\ m_{21}+\mu_{21} & m_{22}+\mu_{22} \end{bmatrix}. \tag{2.7}$$

A much more effective algorithm for computing $\bar{y}_1$, $\bar{y}_2$ and μ_{ij} can be produced if $m_{21} = 0$. The task of determining generators x_1 and x_2 corresponding to a zero value of m_{21} can be done using the QR algorithm (see [1]).

When $m_{21} = 0$, (2.7) becomes

$$(A-m_{11}I)\bar{y}_1 - \mu_{11}x_1 - \mu_{21}x_2 = r_1 + \mu_{11}\bar{y}_1 + \mu_{21}\bar{y}_2,$$

$$(A-m_{22}I)\bar{y}_2 - \mu_{12}x_1 - \mu_{22}x_2 - m_{12}\bar{y}_1 = r_2 + \mu_{12}\bar{y}_1 + \mu_{22}\bar{y}_2$$

where $r_1 = m_{11}x_1 - Ax_1$ and $r_2 = m_{12}x_1 + m_{22}x_2 - Ax_2$ and both r_1 and r_2 are expected to be small. These equations can be expressed in the simpler form as in (2.3) using notations B_i and y_i. Thus we have obtained the iterative procedure

$$B_1 y_1^{k+1} = r_1 + y_{1p}^k \bar{y}_1^k + y_{1q}^k \bar{y}_2^k,$$

$$B_2 y_2^{k+1} - m_{12}\bar{y}_1^{k+1} = r_2 + y_{2p}^k \bar{y}_1^k + y_{2q}^k \bar{y}_2^k,$$

$$y_i^0 = 0 \ (i = 1,2).$$

Like the nondefective case, we now consider a set of s approximate generators x_1, ... , x_s of the invariant subspace and a $s \times s$ matrix M such that

$$A[x_1,\ldots,x_s] \approx [x_1,\ldots,x_s]M \approx [x_1,\ldots,x_s]\begin{bmatrix} m_{11} & \cdots & m_{1s} \\ \vdots & & \vdots \\ m_{s1} & \cdots & m_{ss} \end{bmatrix}.$$

Then we have

$$A[x_1+\bar{y}_1,\ldots,x_s+\bar{y}_s] = [x_1+\bar{y}_1,\ldots,x_s+\bar{y}_s]\begin{bmatrix} m_{11}+\mu_{11} & \cdots & m_{1s}+\mu_{1s} \\ \vdots & & \vdots \\ m_{s1}+\mu_{s1} & \cdots & m_{ss}+\mu_{ss} \end{bmatrix}. \tag{2.8}$$

Assume that M is triangular (which can be done by using a QR step). Then (2.8) becomes s sets of linear equations:

$$(A-m_{ii}I)\bar{y}_i - \sum_{j=1}^{s}\mu_{ji}x_j - \sum_{j=1}^{i-1}m_{ji}\bar{y}_j = r_i + \sum_{j=1}^{s}\mu_{ji}\bar{y}_j, \qquad i=1,\ldots,s \tag{2.9}$$

where $r_i = \sum_{j=1}^{i} m_{ji}x_j - Ax_i$. Now (2.9) can be solved by the iterative procedure

$$B_i y_i^{k+1} - \sum_{j=1}^{i-1} m_{ji}\bar{y}_j^{k+1} = r_i + \sum_{j=1}^{s} y_{ip_j}\bar{y}_j^{k}, \qquad i=1,\ldots,s,$$

$$y_i^0 = 0$$

where B_i and y_i are analogous to notations in (2.6).

Remark

In both the nondefective and defective cases, Dongarra, Moler and Wilkison try to solve the equation

$$AX = XM \tag{2.10}$$

simultaneously for the $l \times s$ matrix X and the $s \times s$ matrix M ($s=1,\ldots,l$). Since (2.10) is ls equations in $ls+s^2$ unknowns, they fixed s^2 unknowns by using Gaussian elimination of X with column pivoting [4].

If the initial approximation of X has orthonormal columns, by an orthonormal change of basis, X may be written

$$X = \begin{bmatrix} I_s \\ R \end{bmatrix}$$

and (2.10) yields the Riccati equation

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R$$

where $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is now the new transformed matrix [4].

2.2 The method S

Actually the method S which is devised by Stewart is presented not as an iterative method for computing invariant subspaces but as a technique for obtaining error bounds and perturbation bounds for invariant subspaces associated with the eigenvalue problem. However, it works as an iterative method and gives the approximate invariant subspace.

Consider $A \in \mathbf{R}^{l\times l}$ and an orthogonal matrix $X = (X_1 , X_2) \in \mathbf{R}^{l\times l}$ where $X_1 \in \mathbf{R}^{l\times n}$, $X_2 \in \mathbf{R}^{l\times m}$, $l = n+m$. Then

$$X^T AX = \begin{bmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{bmatrix}$$

where $A'_{ij} = X_i^T A X_j$, $i,j=1,2$.

We know that $R(X_1)$ is an invariant subspace of A if and only if $A'_{21} = 0$. Now suppose that A'_{21}, instead of being zero, is merely small then $R(X_1)$ is an approximate invariant subspace. To determine a more accurate invariant subspace of A, we shall attempt to find an orthogonal matrix U such that the first n columns of $Y = XU$ span

the exact invariant subspace of A. Take U in the form

$$U = \begin{pmatrix} I_n & -R^T \\ R & I_m \end{pmatrix} \begin{pmatrix} (I+R^TR)^{-\frac{1}{2}} & 0 \\ 0 & (I+RR^T)^{-\frac{1}{2}} \end{pmatrix} \tag{2.11}$$

where $R \in \mathbf{R}^{m\times n}$. If $Y = (Y_1, Y_2)$ and $B_{ij} = Y_i^T A Y_j$ then a necessary and sufficient condition for $R(Y_1)$ to be the exact invariant subspace of A is that $B_{21} = 0$. Now from (2.11), we can express B_{21} in terms of R, and then the equation $B_{21} = 0$ gives

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R$$

or

$$TR = -A_{21} + \phi(R) \tag{2.12}$$

where the mappings $T, \phi : \mathbf{R}^{m\times n} \to \mathbf{R}^{m\times n}$ are defined by

$$TR = A_{22}R - RA_{11}$$

and

$$\phi(R) = RA_{12}R.$$

Thus (2.12) may be solved by the iterative procedure (suppose that T is invertible)

$$R^{k+1} = T^{-1}(-A_{21} + \phi(R^k)),$$

$$R^0 = 0.$$

Remark

Recall that $\lambda(T) = \lambda(A_{22}) - \lambda(A_{11})$, where $\lambda(T)$, $\lambda(A_{22})$ and $\lambda(A_{11})$ are the sets of eigenvalues of T, A_{22} and A_{11}, respectively. A consequence of this fact is that T is invertible if and only if $\lambda(A_{11}) \cap \lambda(A_{22}) = \varnothing$ [2].

2.3 The method C

The method C which is devised by Chatelin is a computational scheme to refine an approximate invariant subspace using *Newton's method*.

She seeks a $l \times n$ matrix X which satisfies

$$AX = XB \quad , \quad Y^TX = I_n \tag{2.13}$$

where $A \in \mathbf{R}^{l\times l}$, $B \in \mathbf{R}^{n\times n}$ and a fixed full rank matrix $Y \in \mathbf{R}^{l\times n}$. Since $B = Y^TAX$, the equation (2.13) is equivalent to the quadratic equation

$$F(X) = AX - X(Y^TAX) = 0. \tag{2.14}$$

Using the notation

$$J(X)Z = (I_l - XY^T)AZ - Z(Y^TAX),$$

she derived the Newton iteration,

$$X^{k+1} = X^k - J^{-1}(X^k)F(X^k), \quad X^0 = U \ , \ Y^TU = I_n.$$

Now change basis so that $Y = [I_n \ 0]^T$. In the new basis it is easy to see $Y^TX=I_n$ implies X is of the form

$$X = \begin{bmatrix} I_n \\ R \end{bmatrix}$$

and then (2.14) becomes the Riccati equation

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R$$

where $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is the new transformed matrix [4].

Remarks

- The method C has been derived under the assumption that X is associated with a set of well separated eigenvalues of A so that $J^{-1}(X)$ is bounded [3].
- The method S and the method C were originally presented for $A \in C^{lxl}$. However, since we are interested in matrix problems which involve real data, we considered only a real case.

CHAPTER 3

THE APPLICATIONS OF INVARIANT SUBSPACES

We know that the sensitivity of an eigenvector depends on eigenvalue sensitivity and on the separation of its corresponding eigenvalue from the other eigenvalues. Thus in ill-conditioned eigenproblem, it makes more sense to try to obtain the invariant subspace than an eigenvector. But often it is sufficient to compute the invariant subspace even though the eigenvalue problem is well-conditioned.

In this chapter, we discuss several cases which require the computation of invariant subspaces. The various applications of invariant subspaces are as follows: The separation of time scales in optimal control theory [5], [6], [7], [8]; normal form of the differential equations [9]; and decoupling [10], [11], [12], [13]. Throughout this chapter, we can see how they use invariant subspaces to achieve their desired aim. Moreover, it is obvious that their methods of computing invariant subspaces have been considered by looking at the Riccati equations corresponding to the relevant invariant subspaces.

3.1 The separation of time scales in singularly perturbed systems

First we briefly review some of the theory of the singular perturbation systems and the importance of the separation of time scales.

Time invariant systems with slow and fast modes (singularly perturbed systems) are stiff, and control problems for such systems are often ill-conditioned. These facts have motivated a system simplification approach to get a well-conditioned system which is equivalent to the original system. A well-known method is the separation of time modes.

Consider the singular perturbation system of finite dimensional dynamical systems, that is,

$$\dot{x} = f(x, z, u, \epsilon, t), \quad x \in \mathbf{R}^n \tag{3.1a}$$

$$\epsilon\dot{z} = g(x, z, u, \epsilon, t), \quad z \in \mathbf{R}^m \tag{3.1b}$$

where $u = u(t)$ is the control vector and ϵ represents a small positive parameter.

When we set $\epsilon = 0$, the differential equation (3.1b) becomes

$$g(\bar{x}, \bar{z}, \bar{u}, 0, t) = 0 \tag{3.2}$$

where the bar is used to indicate that the variables belong to a system with $\epsilon = 0$. We will say that the singular perturbation system is in standard form if and only if (3.2) has $k \geq 1$ distinct (isolated) real roots

$$\bar{z} = \phi_i(\bar{x}, \bar{u}, t), \quad i=1,2,...,k.$$

So if the system is in standard form then we can obtain a well-defined n-dimensional reduced model, that is,

$$\dot{\bar{x}} = f(\bar{x}, \phi_i(\bar{x}, \bar{u}, t), \bar{u}, 0, t)$$

and we can rewrite this form more compactly

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}, t). \tag{3.3}$$

This model is called a quasi-steady-state model which is related to a slow mode.

The slow mode solution, or the quasi-steady-state, is approximated by the reduced model (3.3), while the discrepancy between the mode of the reduced model (3.3) and that of the full model (3.1a)-(3.1b) is the fast mode. To see this, let us examine the variable z which has been excluded from the reduced model (3.3) and substituted by its slow mode $\bar{z}$. There may be a large discrepancy between $\bar{z}(t_0)$ and $z(t_0)$, where t_0 is the initial point of t. Thus $\bar{z}$ cannot be a uniform approximation of z. The best that we can expect is that the approximation $z = \bar{z}(t) + O(\epsilon)$ will hold for $t \in [t_1, T]$, where $t_1 > t_0$.

We now consider the behavior of z in an initial (boundary layer) interval $[t_0, t_1]$. We set

$$\epsilon \frac{dz}{dt} = \frac{dz}{d\tau},$$

hence

$$\frac{d\tau}{dt} = \frac{1}{\epsilon},$$

and use $\tau = 0$ at $t = t_0$. The new time variable is

$$\tau = \frac{t - t_0}{\epsilon},$$

and $\tau = 0$ at $t = t_0$. To describe the behavior of z as a function of τ, we use the boundary layer system,

$$\frac{d\hat{z}}{d\tau} = g(x^0, \hat{z}(\tau), u, 0, t_0) \tag{3.4}$$

with z^0 as the initial condition for $\hat{z}(\tau)$, and x^0, t_0 as fixed parameters.

Using these two reduced order models (3.3) and (3.4), we can obtain the uniform approximations of $x(t,\epsilon)$ and $z(t,\epsilon)$.

$$x = \bar{x}(t) + O(\epsilon),$$

$$z = \bar{z}(t) + \hat{z}(\tau) - \bar{z}(t_0) + O(\epsilon), \quad t \in [t_0, T].$$

Also many properties of the singular perturbation system such as controllability and stability can be deduced from the same properties of slow and fast subsystems. Thus it is important to obtain the reduced order models.

We now describe methods for obtaining the reduced order models of linear continuous and discrete time systems which have been suggested by Kokotovic [5], [6] and Phillips [7], [8] respectively. In both cases, they use an invariant subspace as the first n columns of the similarity transformation.

3.1.1 Continuous time case

A model of a linear time-invariant continuous system with slow and fast modes is

$$\begin{bmatrix} \dot{x} \\ \epsilon \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{3.5}$$

where $x \in \mathbf{R}^n$, $z \in \mathbf{R}^m$, $u \in \mathbf{R}^{n+m}$ and ϵ represents small time constant.

Now using the Riccati transformation, we will transform (3.5) into

$$\begin{bmatrix} \dot{\xi} \\ \epsilon \dot{\eta} \end{bmatrix} = \begin{bmatrix} A^1 & 0 \\ 0 & A^2 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} B^1 \\ B^2 \end{bmatrix} u \tag{3.6}$$

where ξ and η are related to a slow mode and a fast mode, respectively.

First, to get a fast mode η, set $\epsilon = 0$. Then $A_{21}x + A_{22}z + B_2u = 0$. Thus if A_{22}^{-1} exists then the steady-state of z is $\bar{z} = -A_{22}^{-1}A_{21}x$ for $u = 0$. So η can be defined as $\eta = z + A_{22}^{-1}A_{21}x + \epsilon Gx = z + (A_{22}^{-1}A_{21} + \epsilon G)x$. Then

$$\begin{bmatrix} x \\ \eta \end{bmatrix} = \begin{bmatrix} I & 0 \\ R & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

where $R = A_{22}^{-1}A_{21} + \epsilon G$, and

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \epsilon \dot{\eta} \end{bmatrix} &= \begin{bmatrix} I & 0 \\ \epsilon R & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -R & I \end{bmatrix} \begin{bmatrix} x \\ \eta \end{bmatrix} + \begin{bmatrix} I & 0 \\ \epsilon R & I \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ &= \begin{bmatrix} A_{11} - A_{12}R & A_{12} \\ 0 & A_{22} + \epsilon R A_{12} \end{bmatrix} \begin{bmatrix} x \\ \eta \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 + \epsilon R B_1 \end{bmatrix} u \end{aligned} \tag{3.7}$$

where

$$A_{22}R - R(\epsilon A_{11}) + R(\epsilon A_{12})R - A_{21} = 0. \tag{3.8}$$

To transform (3.5) into (3.7), we have to find the solution of (3.8). Let $R = R_0 + D$, $R_0 = A_{22}^{-1}A_{21}$, $D = \epsilon G$ and $A_0 = A_{11} - A_{12}R_0$. Then D is a real root of

$$D(\epsilon A_0) - (A_{22}+R_0(\epsilon A_{12}))D - D(\epsilon A_{12})D + R_0(\epsilon A_0) = 0. \quad (3.9)$$

Setting $\epsilon A_0 = \tilde{A}_0$ and $\epsilon A_{12} = \tilde{A}_{12}$ then (3.9) becomes

$$D\tilde{A}_0 - (A_{22}+R_0\tilde{A}_{12})D - D\tilde{A}_{12}D + R_0\tilde{A}_0 = 0. \quad (3.10)$$

The following theorem gives a sufficient condition for the existence and uniqueness of a real root D and establishes a bound for its norm $\|D\|$. It also formulates a convergent iterative method for computing D.

The Kokotovic theorem I

If A_{22} is nonsingular and if

$$\|A_{22}^{-1}\| \leqslant \frac{1}{3}(\|\tilde{A}_0\| + \|\tilde{A}_{12}\|\|R_0\|)^{-1}$$

then a unique real root of (3.10) exists satisfying

$$0 \leqslant \|D\| \leqslant 2\|\tilde{A}_0\|\|R_0\| \,/\, [\|\tilde{A}_0\| + \|\tilde{A}_{12}\|\|R_0\|] .$$

This root is an asymptotically stable equilibrium of the difference equation

$$D_{k+1} = A_{22}^{-1}(R_0\tilde{A}_0 + D_k\tilde{A}_0 - R_0\tilde{A}_{12}D_k - D_k\tilde{A}_{12}D_k), \quad D_0=0.$$

Remarks ([6])

- In practice, we usually use the simpler form of the iterative method which is directly obtained from (3.8):

$$R_{k+1} = A_{22}^{-1}(R_k(\epsilon A_{11}) - R_k(\epsilon A_{12})R_k + A_{21}),$$

$$R_0 = A_{22}^{-1}A_{21}.$$

- After k iterations the relative error is

$$\frac{\|D_k - D\|}{\|D\|} \leqslant [3\|A_{22}^{-1}\|(\|\tilde{A}_0\| + \|\tilde{A}_{12}\|\|R_0\|)]^k = \epsilon^k[3\|A_{22}^{-1}\|(\|A_0\| + \|A_{12}\|\|R_0\|)]^k.$$

Thus the convergence rate of Kokotovic's iterative method depends on ϵ.

Now consider a slow mode ξ. Let $\xi = x - \epsilon(A_{12}A_{22}^{-1} + \epsilon H)\eta$, i.e.

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix} = \begin{pmatrix}I & -\epsilon M\\ 0 & I\end{pmatrix}\begin{pmatrix}x\\ \eta\end{pmatrix}$$

where $M = A_{12}A_{22}^{-1} + \epsilon H$. Then

$$\begin{pmatrix}\dot{\xi}\\ \epsilon\dot{\eta}\end{pmatrix}$$

$$= \begin{pmatrix}I & -M\\ 0 & I\end{pmatrix}\begin{pmatrix}A_{11}-A_{12}R & A_{12}\\ 0 & A_{22}+\epsilon RA_{12}\end{pmatrix}\begin{pmatrix}I & \epsilon M\\ 0 & I\end{pmatrix}\begin{pmatrix}\xi\\ \eta\end{pmatrix} + \begin{pmatrix}I & -M\\ 0 & I\end{pmatrix}\begin{pmatrix}B_1\\ B_2+\epsilon RB_1\end{pmatrix}u$$

$$= \begin{pmatrix}A_{11}-A_{12}R & 0\\ 0 & A_{22}+\epsilon RA_{12}\end{pmatrix}\begin{pmatrix}\xi\\ \eta\end{pmatrix} + \begin{pmatrix}B_1-M(B_2+\epsilon RB_1)\\ B_2+\epsilon RB_1\end{pmatrix}u$$

where

$$\epsilon(A_{11}-A_{12}R)M - M(A_{22}+\epsilon RA_{12}) + A_{12} = 0. \tag{3.11}$$

The Kokotovic theorem II

Under the conditions of theorem I, the solution M of equation (3.11) is the asymptotically stable equilibrium of the difference equation

$$M_{k+1} = [\epsilon(A_{11}-RA_{12})M_k - M_k(\epsilon RA_{12})]A_{22}^{-1} + A_{12}A_{22}^{-1}.$$

Summarily, if we consider the simpler form of (3.5) as

$$\dot{x} = \begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}x + \begin{pmatrix}B_1\\ B_2\end{pmatrix}u \equiv Ax + Bu$$

then under the conditions of Kokotovic theorem I,

$$y = \begin{pmatrix}I-MR & -M\\ R & I\end{pmatrix}x \equiv T^{-1}x$$

transforms A into a block diagonal form ($= T^{-1}AT$), where

$$A_{22}R - RA_{11} + RA_{12}R - A_{21} = 0,$$

$$(A_{11} - A_{12}R)M - M(A_{22} + RA_{12}) + A_{12} = 0.$$

Moreover, this similarity transformation preserves a two time scale property, i.e. the transformed system (3.6) has n small eigenvalues and m large eigenvalues like the original system (3.5).

3.1.2 Discrete time case

A model of a linear time-invariant discrete system is

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k) \tag{3.12}$$

where $x(k) \in \mathbf{R}^n$, $z(k) \in \mathbf{R}^m$ and $u(k) \in \mathbf{R}^{n+m}$. As in the continuous case, there will exist a basis such that (3.12) takes the form

$$\begin{bmatrix} \xi(k+1) \\ \eta(k+1) \end{bmatrix} = \begin{bmatrix} A^1 & 0 \\ 0 & A^2 \end{bmatrix} \begin{bmatrix} \xi(k) \\ \eta(k) \end{bmatrix} + \begin{bmatrix} B^1 \\ B^2 \end{bmatrix} u(k) \tag{3.13}$$

where ξ and η represent a slow mode and a fast mode, respectively.

To find such a basis which transforms (3.12) into (3.13) and preserves the two time scale property, first consider the Riccati transformation

$$\begin{bmatrix} x(k) \\ \eta(k) \end{bmatrix} = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix}.$$

Then

$$\begin{bmatrix} x(k+1) \\ \eta(k+1) \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix} \begin{bmatrix} x(k) \\ \eta(k) \end{bmatrix} + \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k)$$

$$= \begin{bmatrix} A_{11}-A_{12}P & A_{12} \\ 0 & A_{22}+PA_{12} \end{bmatrix} \begin{bmatrix} x(k) \\ \eta(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2+PB_1 \end{bmatrix} u(k)$$

where

$$A_{22}P - PA_{11} + PA_{12}P - A_{21} = 0. \tag{3.14}$$

To completely transform (3.14) into a block diagonal form, let

$$\begin{bmatrix}\xi(k)\\ \eta(k)\end{bmatrix} = \begin{bmatrix}I & -Q\\ 0 & I\end{bmatrix}\begin{bmatrix}x(k)\\ \eta(k)\end{bmatrix}$$

Then

$$\begin{bmatrix}\xi(k+1)\\ \eta(k+1)\end{bmatrix}$$

$$= \begin{bmatrix}I & -Q\\ 0 & I\end{bmatrix}\begin{bmatrix}A_{11}-A_{12}P & A_{12}\\ 0 & A_{22}+PA_{12}\end{bmatrix}\begin{bmatrix}I & Q\\ 0 & I\end{bmatrix}\begin{bmatrix}\xi(k)\\ \eta(k)\end{bmatrix} + \begin{bmatrix}I & -Q\\ 0 & I\end{bmatrix}\begin{bmatrix}B_1\\ B_2+PB_1\end{bmatrix}u(k)$$

$$= \begin{bmatrix}A_{11}-A_{12}P & 0\\ 0 & A_{22}+PA_{12}\end{bmatrix}\begin{bmatrix}\xi(k)\\ \eta(k)\end{bmatrix} + \begin{bmatrix}B_1-Q(B_2+PB_1)\\ B_2+PB_1\end{bmatrix}u(k)$$

where

$$(A_{11}-A_{12}P)Q - Q(A_{22}+PA_{12}) + A_{12} = 0. \tag{3.15}$$

We remark that Phillips derived basically the same equations (3.14)-(3.15) such as Kokotovic's equations (3.8)-(3.11). Actually to obtain the reduced order model of a linear discrete time system, Phillips used Kokotvic's iterative method [6].

Using a similarity transformation

$$V^{-1} = \begin{bmatrix}I-QP & -Q\\ P & I\end{bmatrix}.$$

the system (3.12) is transformed into

$$\begin{bmatrix}\xi(k+1)\\ \eta(k+1)\end{bmatrix} = \begin{bmatrix}A^1 & 0\\ 0 & A^2\end{bmatrix}\begin{bmatrix}\xi(k)\\ \eta(k)\end{bmatrix} + \begin{bmatrix}B^1\\ B^2\end{bmatrix}u(k)$$

where

$$A^1 = A_{11} - A_{12}P, \qquad A^2 = A_{22} + PA_{12},$$

$$B^1 = B_1 - Q(B_2+PB_1), \; B^2 = B_2 + PB_1.$$

In both linear continuous and discrete time cases, Kokotovic and Phillips try to transform the original systems into block diagonal forms by similarity transformation. Whenever the similarity transformation brings an original system to a block diagonal form, the two sets of columns of a similarity transformation constitute bases of the eigenspaces represented by the blocks. This means that the first n columns of T (or V) are a basis for the slow eigenspace of (3.5) (or (3.12)) and the remaining m columns are a basis for the fast eigenspace of (3.5) (or (3.12)), i.e. the first n columns of T (or V) span the invariant subspace of (3.5) (or (3.12)) which is correspending to the the slow modes and the remaining m columns of T (or V) span the invariant subspace of (3.5) (or (3.12)) which is corresponding to the fast modes.

3.2 Normal form for the differential equations

Kreiss, Nichols and Brown [9] consider the two-point boundary problem for stiff systems of ordinary differential equations (ODEs). To obtain accurate numerical solutions of such problems, they try to transform an original system into a block diagonal form by using a smooth similarity transformation. During the process of obtaining a desired transformation, they need to compute an invariant subspace and moreover need to solve the Riccati equation.

To see the details of the procedure, consider the two-point boundary value problem for a linear system of n ordinary differential equations

$$\frac{dy}{dx} = A(x)y + f(x), \qquad 0 \leqslant x \leqslant c \tag{3.16a}$$

subject to n linearly independent boundary conditions

$$B_0 y(0) + B_1 y(c) = g \tag{3.16b}$$

where $y^T = (y_1, \cdots, y_n)$ is a vector function with n components and B_0, B_1 and $A(x) \in C^p$ (i.e. the elements of $A(x)$ are p times continuously differentiable) are all

$n \times n$ matrices and the vector $f(x) \in C^r$. We now divide the x-axis into subintervals of variable length h_j with $x_0=0$, $x_i = \sum_{j=0}^{j=i-1} h_j$, $i=2, \cdots, N$ and $x_N=c$. Using $h = \max_j h_j$, Kreiss, Nichols and Brown define that the system (3.16a)-(3.16b) is stiff if $h\|A\| >> 1$.

To transform (3.16a)-(3.16b) into a desired block diagonal form, they first calculate the eigenvalues of $A(x)$ and divide them into sets M^j containing eigenvalues which are of the same order of magnitude. Since the number of sets M^j depends on x, the block structure can be a function of x. The next step is to determine a transformation $S(x)$ such that

$$\bar{A}(x) = S^{-1}(x)A(x)S(x) = \begin{pmatrix} A_r(x) & 0 & 0 \\ 0 & A_{r-1}(x) & 0 \\ & & \\ 0 & 0 & A_0(x) \end{pmatrix}$$

is in block diagonal form. Here the eigenvalues of every $A_j(x)$ are exactly the eigenvalues contained in M^j.

To construct $S(x)$, first start with the interval $0 \leq x \leq c_1$. At $x=0$, we know that there exists a unitary transformation $U(0)$ such that

$$U^H(0)A(0)U(0) = \begin{pmatrix} A_r & A_{r,r-1} & A_{r,0} \\ 0 & A_{r-1} & A_{r-1,0} \\ & & \\ 0 & 0 & A_0 \end{pmatrix}$$

is in block upper triangular form. Then we determine

$$\bar{S}(0) = \begin{pmatrix} I & S_{r,r-1} & & S_{r,0} \\ 0 & I & & S_{r-1,0} \\ & & & \\ 0 & 0 & & I \end{pmatrix}$$

such that

$$\bar{A}(0) = S^{-1}(0)A(0)S(0) = \begin{pmatrix} A_r & 0 & & 0 \\ 0 & A_{r-1} & & 0 \\ & & & \\ 0 & 0 & & A_0 \end{pmatrix}$$

has the desired block form, where $S(0) = U(0)\bar{S}(0)$.

Now consider the transformed matrix

$$\tilde{A}(x) \equiv S^{-1}(0)A(x)S(0) = \bar{A}(0) + B(x), \quad B(0) = 0$$

where

$$B(x) = S^{-1}(0)(A(x)-A(0))S(0) = \begin{pmatrix} B_{r,r} & B_{r,r-1} & & B_{r,0} \\ B_{r-1,r} & B_{r-1,r-1} & & B_{r-1,0} \\ & & & \\ B_{0,r} & B_{0,r-1} & & B_{0,0} \end{pmatrix}.$$

By assumption the eigenvalues of each block are well separated from the eigenvalues of all other blocks. Therefore in the neighborhood of $x=0$, there exists an $\bar{S}(x)$ such that

$$\bar{S}^{-1}(x)\tilde{A}(x)\bar{S}(x) = S^{-1}(x)A(x)S(x) = \begin{pmatrix} A_r(x) & 0 & & 0 \\ 0 & A_{r-1}(x) & & 0 \\ & & & \\ 0 & 0 & & A_0(x) \end{pmatrix}$$

and so $S(x) = S(0)\bar{S}(x)$.

To discuss this transformation $\tilde{S}(x)$ in detail, they considered a couple of lemma.

Lemma 1

Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbf{R}^{n\times n}$ and $A_{22} \in \mathbf{R}^{m\times m}$. Assume that the eigenvalues of A_{11} are disjoint from the eigenvalues of A_{22}. Then the matrix equation

$$A_{11}X - XA_{22} = C$$

has a unique solution.

Lemma 2

Assume that the eigenvalues of A_{11} are disjoint from those of A_{22} and $\|A_{12}\|$ and $\|A_{21}\|$ are sufficiently small. Then there exist R and M such that

$$\begin{bmatrix} I & 0 \\ -R & I \end{bmatrix}\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} I & 0 \\ R & I \end{bmatrix} = \begin{bmatrix} A_{11}+A_{12}R & A_{12} \\ 0 & A_{22}-RA_{12} \end{bmatrix}$$

and

$$\begin{bmatrix} I & -M \\ 0 & I \end{bmatrix}\begin{bmatrix} A_{11}+A_{12}R & A_{12} \\ 0 & A_{22}-RA_{12} \end{bmatrix}\begin{bmatrix} I & M \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_{11}+A_{12}R & 0 \\ 0 & A_{22}-RA_{12} \end{bmatrix}$$

where

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R,$$

$$(A_{11}+A_{12}R)M - M(A_{22}-RA_{12}) = -A_{12}.$$

These results above can be used to construct a transformation $\bar{S}(x)$ which transforms

$$\tilde{A}(x) = \bar{A}(0) + B(x) = \begin{bmatrix} A_r+B_{r,r} & B_{r,r-1} & B_{r,0} \\ B_{r-1,r} & A_{r-1}+B_{r-1,r-1} & B_{r-1,0} \\ & & \\ B_{0,r} & B_{0,r-1} & A_0+B_{0,0} \end{bmatrix} \equiv \begin{bmatrix} A_{11} & \bar{B}_{12} \\ \bar{B}_{21} & A_{22} \end{bmatrix}$$

into a block diagonal form.

If the matrix $\tilde{A}(x)$ satisfies the conditions of lemma 2, then there exists a unique transformation

$$S_r = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \begin{bmatrix} I & Q \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & Q \\ P & I+PQ \end{bmatrix}$$

such that

$$S_r^{-1}\tilde{A}(x)S_r = \begin{bmatrix} A_{11}+\bar{B}_{12}P & 0 \\ 0 & A_{22}-P\bar{B}_{12} \end{bmatrix}$$

Using $A_{22} - P\bar{B}_{12}$ as $\tilde{A}(x)$, the same process can be applied to it and so $\bar{S}(x) = S_r S_{r-1} \ldots S_0$.

Remarks

- In practice, before making the transformations of lemma 2, the matrix $\tilde{A}(x)$ is scaled so that $\bar{B}_{12}$ and $\bar{B}_{21}$ are of the same order of magnitude [9].
- To find $\bar{S}(x)$, it is important whether the eigenvalues sets M^j are well separated or not.

Now one could use $S(c_1)$ as the starting transformation for $c_1 \leqslant x \leqslant c_2$ and repeat the above procedure to obtain $S(x)$. By continuing this procedure, Kreiss, Nichols and Brown determine $S(x)$ for $0 \leqslant x \leqslant c$ and use it to transform the system (3.16a) into

$$\frac{d\bar{y}}{dx} = \bar{A}(x)\bar{y} + H(x)\bar{y} + G(x)$$

with

$$\bar{A}(x) = \begin{bmatrix} A_r(x) & 0 & 0 \\ 0 & A_{r-1}(x) & 0 \\ & & \\ 0 & 0 & A_0(x) \end{bmatrix},$$

$$H = -S^{-1}\frac{dS}{dx}\ ,\ G = S^{-1}f\ ,\ \bar{y} = S^{-1}y$$

on every blocking subinterval $c_i \leqslant x \leqslant c_{i+1}$.

As in section 3.1, they seek a smooth similarity transformation such that the transformed system has a block digonal form. It follows that each set of columns corresponding to a diagonal block span an invariant subspace.

3.3 Decoupling

It is well-known that if initial value problems (IVPs) have increasing fundamental solutions then IVPs are unstable in general. Many people [10], [11], [12] solve boundary value problems (BVPs) having increasing and decreasing modes by decoupling, so that we can split BVPs into two stable IVPs and compute the increasing modes for decreasing time and the decreasing modes for increasing time.

In this section, we consider the general case and the particular case which have been suggested by Dieci-Osborne-Russell [10] and Wilde-Kokotovic [12], respectively. Dieci, Osborne and Russell consider linear two-point boundary value problems for ODEs. By using the Riccati transformation, they formulate BVPs as three IVPs, one of them the Riccati equation. Wilde and Kokotovic consider particular BVPs which appear in optimal control systems. They try to transform BVPs into IVPs by the dichotomy transformation. In both cases, they use the Riccati transformation and the dichotomy transformation as the similarity transformations and the first n columns span the

invariant subspace.

3.3.1 The general case

Consider the ODE with constant coefficients

$$\dot{y}(t) = Ay(t) + q, \quad 0 < t < 1 \tag{3.17}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad y(t) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

$$A_{11} \in \mathbf{R}^{n\times n}, \; A_{22} \in \mathbf{R}^{m\times m}, \; q_1, y_1 \in \mathbf{R}^n, \; q_2, y_2 \in \mathbf{R}^m.$$

Using the Riccati transformation

$$w(t) := \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} = \begin{bmatrix} I & 0 \\ -R & I \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix},$$

the original system (3.17) is transformed into

$$\begin{bmatrix} \dot{w}_1(t) \\ \dot{w}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11}+A_{12}R & A_{12} \\ 0 & A_{22}-RA_{12} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 - Rq_1 \end{bmatrix} \tag{3.18}$$

where

$$A_{22}R - RA_{11} - RA_{12}R + A_{21} = 0.$$

By the fundamental decoupling theorem (see [11]), if (3.17) has an exponential dichotomy, i.e. there exist a projection matrix P, a moderate constant $K>0$, and $\lambda, \mu \geq 0$ such that

$$\|Y(t)PY^{-1}(s)\| \leq Ke^{-\lambda(t-s)}, \quad 0 \leq s \leq t \leq 1$$

$$\|Y(t)(I-P)Y^{-1}(s)\| \leq Ke^{-\mu(s-t)}, \quad 0 \leq t \leq s \leq 1$$

where $Y(t)$ is a fundamental solution for (3.17), then we can stably compute the increasing modes for decreasing time and the decreasing modes for increasing time in the transformed system (3.18). This means that well-conditioned BVPs with separated

boundary conditions (BCs) can be solved by solving two stable vector IVPs and the Riccati equation.

3.3.2 The particular case

Consider a $2n$-dimensional system with constant coefficient

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix} = \begin{bmatrix}A & -G\\ -H & -A^T\end{bmatrix}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} \tag{3.19}$$

where all matrices A , G and H are in $\mathbf{R}^{n\times n}$ and G and H are symmetric positive definite matrices.

To transform (3.19) into the block diagonal form, Wilde and Kokotovic use the dichotomy transformation. The dichotomy transformation

$$\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} = \begin{bmatrix}I & I\\ P & N\end{bmatrix}\begin{bmatrix}\xi(t)\\ \eta(t)\end{bmatrix} \tag{3.20}$$

is constructed using the symmetric positive definite solution P and the symmetric negative definite solution N of the algebraic Riccati equation

$$A^TX + XA - XGX + H = 0.$$

To analyze this, we need to review some definitions and a theorem which is presented by Wonham [13].

Definition

1. The pair $(A\ ,B)$ is *controllable* if the rank of Γ is n, where $\Gamma(A,B) = [B,AB, \cdots ,A^{n-1}B]$.
2. The pair $(A\ ,B)$ is *stabilizable* if there exists a constant matrix K such that $A - BK$ is stable (i.e. all its eigenvalues have negative real parts).
3. The pair $(C\ ,A)$ is *detectable* if $(A^T\ ,C^T)$ is stabilizable.

Using above definitions, Wonham shows necessary conditions for existence and uniqueness of P and N.

Wonham theorem

Ccnsider the algebraic Riccati equation

$$A^TX + XA - XGX + H = 0 \tag{3.21}$$

where all matrices are in $\mathbf{R}^{n\times n}$ and G and H are symmetric positive difinite matrices. If $(A\ ,\ B)$ is a *stabilizable* pair, where B is a full-rank factqrization of G (i.e. $BB^T = G$ and rank (B) = rank (G)) and $(C\ ,\ A)$ is a *detectable* pair where C is a full-rank factorization of H (i.e. $C^TC = H$ and rank (C) = rank (H)), then (3.21) has unique symmetric positive and negative definite solutions.

□

Under the assumptions of Wonham's theorem, there exists a dichotomy transformation (3.20) which transforms (3.19) into

$$\begin{vmatrix}\dot{\xi}(t)\\ \eta(t)\end{vmatrix} = \begin{vmatrix}A-GP & 0\\ 0 & A-GN\end{vmatrix}\begin{vmatrix}\xi\\ \eta\end{vmatrix} \tag{3.22}$$

The following theorem shows the stability of the transformation of (3.22). The proof is based on the proof of a stability theorem in [10].

Theorem

If (3.19) has an exponential dichotomy then the dichotomy transformation (3.20) transforms (3.19) into (3.22) which has two n-dimensional systems, one asymptotically stable in forward time and the other asymptotically stable in reverse time.

Proof

Let Y be the fundamental solution for (3.19) such that

$$Y(t) = \begin{bmatrix} Y_{11}(t) & Y_{12}(t) \\ Y_{21}(t) & Y_{22}(t) \end{bmatrix}$$

satisfying $Y(0) = T$, $T = \begin{bmatrix} I & I \\ P & N \end{bmatrix}$ and let $W(t) = \begin{bmatrix} W_{11}(t) & 0 \\ 0 & W_{22}(t) \end{bmatrix}$ be the fundamental solution for (3.22) satisfying $W(0) = I$.

Using

$$Y(t) = TW(t) \quad \text{and} \quad T^{-1} = \begin{bmatrix} I+(N-P)^{-1}P & -(N-P)^{-1} \\ -(N-P)^{-1}P & (N-P)^{-1} \end{bmatrix},$$

we can obtain

$$W(t) = \begin{bmatrix} [I+(N-P)^{-1}P]Y_{11}(t)-(N-P)^{-1}Y_{21}(t) & 0 \\ 0 & -(N-P)^{-1}PY_{12}(t)+(N-P)^{-1}Y_{22}(t) \end{bmatrix}$$

where

$$W_{11} = [I+(N-P)^{-1}P]Y_{11}(t) - (N-P)^{-1}Y_{21}(t)$$

and

$$W_{22} = -(N-P)^{-1}PY_{12}(t) + (N-P)^{-1}Y_{22}(t)$$

are invertible.

Writing

$$Y(t)(I-Q)Y^{-1}(s)$$

$$= \begin{bmatrix} Y_{11}(t)W_{11}^{-1}(s)[I+(N-P)^{-1}P] & -Y_{11}(t)W_{11}^{-1}(s)(N-P)^{-1} \\ Y_{21}(t)W_{11}^{-1}(s)[I+(N-P)^{-1}P] & -Y_{21}(t)W_{11}^{-1}(s)(N-P)^{-1} \end{bmatrix},$$

where Q is a projection with rank n.

$$\|W_{11}(t)W_{11}^{-1}(s)\| = \sup_{d\neq 0}\frac{|W_{11}(t)d|}{|W_{11}(s)d|}$$

$$\leqslant \|I+(N-P)^{-1}P\|\|Y_{11}(t)W_{11}^{-1}(s)\| + \|(N-P)^{-1}\|\|Y_{21}(t)W_{11}^{-1}(s)\|$$

$$\leqslant [\|I+(N-P)^{-1}P\| + \|(N-P)^{-1}\|]Ke^{-\mu(s-t)}, \quad t<s.$$

Also

$$Y(t)QY^{-1}(s)$$

$$= \begin{bmatrix} -Y_{12}(t)W_{22}^{-1}(s)(N-P)^{-1}P & Y_{12}(t)W_{22}^{-1}(s)(N-P)^{-1} \\ -Y_{22}(t)W_{22}^{-1}(s)(N-P)^{-1}P & Y_{22}(t)W_{22}^{-1}(s)(N-P)^{-1} \end{bmatrix}$$

Thus

$$\|W_{22}(t)W_{22}^{-1}(s)\| = \sup_{d\neq 0}\frac{|W_{22}(t)d|}{|W_{22}(s)d|}$$

$$\leqslant \|(N-P)^{-1}P\|\|Y_{12}(t)W_{22}^{-1}(s)\| + \|(N-P)^{-1}\|\|Y_{22}(t)W_{22}^{-1}(s)\|$$

$$\leqslant [\|(N-P)^{-1}P\| + \|(N-P)^{-1}\|]Ke^{-\lambda(t-s)}, \quad t\geqslant s.$$

Now consider the BVP consisting of (3.19) and the BCs

$$x(t_0) = x_0\ ,\ x(T) = x_T\ ,\quad t_0\leqslant t\leqslant T. \tag{3.23}$$

To solve the BVP using the dichotomy transformation (3.20), $\xi(t_0)$ and $\eta(T)$ are determined from

$$\begin{bmatrix} x_0 \\ x_T \end{bmatrix} = \begin{bmatrix} I & V_1(t) \\ V_2(t) & I \end{bmatrix}\begin{bmatrix} \xi(t_0) \\ \eta(T) \end{bmatrix}$$

where $V_1(t)$ and $V_2(t)$ are the fundamental solutions of $\dot{\eta}(t) = (A-GN)\eta$ and $\dot{\xi}(t) = (A-GP)\xi$, respectively. However, if (3.19) has an exponential dichotomy (see [12]) and if $[t_0\ ,\ T]$ is sufficiently large then

$$|V_1(t)| << 1 \ , \ |V_2(t)| << 1$$

and

$$\xi(t_0) \approx x_0 \ , \ \eta(T) \approx x_T.$$

Using these BCs, the approximation $\xi(t)$ and $\eta(t)$ are obtained from the independent initial value problems

$$\dot{\xi}(t) = (A-GP)\xi(t), \quad \xi(t_0) = x_0,$$

$$\dot{\eta}(t) = (A-GN)\eta(t), \quad \eta(T) = x_T.$$

Thus the approximation solution of the BVP (3.19), (3.23) is

$$x(t) = \xi(t) + \eta(t),$$

$$z(t) = P\xi(t) + N\eta(t).$$

In a general case, if the BVPs are well-conditioned and have an exponential dichotomy, then solution vectors can be obtained from two IVPs which have asymptotically stable solution vectors, and from the Riccati equation.

CHAPTER 4

THE SCHUR METHOD

The Schur method for solving algebraic Riccati equations which arise from control problems (both continuous and discrete time cases) is devised by Laub [15]. In both cases, he seeks the unique (under suitable assumptions) symmetric nonnegative definite solution by using an orthogonal similarity transformation.

4.1 Continuous time case

Consider the continuous time algebraic Riccati equation

$$A^TX + XA - XBX + C = 0 \tag{4.1}$$

where all matrices are in $\mathbf{R}^{n\times n}$ and B and C are symmetric nonnegative definite matrices. Under the assumptions of Wonham's theorem, the equation (4.1) is known to have a unique symmetric nonnegative definite solution. Of course there can be other solutions to (4.1), but the Schur method tries to find the symmetric nonnegative definite one.

Consider

$$Z = \begin{bmatrix} A & -B \\ -C & -A^T \end{bmatrix} \in \mathbf{R}^{2n\times 2n}.$$

Then the equation (4.1) can be solved by finding an orthogonal matrix U such that

$$U^TZU = S$$

where

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

and

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad S_{ij} \in \mathbf{R}^{n\times n}.$$

Moreover, it is possible to arrange that the real parts of the spectrum of S_{11} are negative, while the real parts of the spectrum of S_{22} are positive. The solution of (4.1) is given by $X = U_{21}U_{11}^{-1}$.

4.2 Discrete time case

Consider the discrete time algebraic Riccati equation

$$A^TXA - X - A^TXB_1(B_2+B_1^TXB_1)^{-1}B_1^TXA + C = 0 \qquad (4.2)$$

where $A, C, X \in \mathbf{R}^{n\times n}$, $B_1 \in \mathbf{R}^{n\times m}$, $B_2 \in \mathbf{R}^{m\times m}$ and $m \leqslant n$. Also C and B_2 are symmetric nonnegative definite matrices.

To have the unique symmetric nonnegative definite solution to (4.2), we need some assumptions which are slightly different from the continuous time case. If $(A\ ,\ B_1)$ is a *stabilizable* pair and $(H\ ,\ A)$ is a *dectable* pair where H is a full-rank factorization of C (i.e. $H^TH = C$ and rank(H) = rank(C)) and A is invertible then (4.2) has a unique symmetric nonnegative definite solution (which the Schur method attempts to compute).

Setting $B = B_1B_2^{-1}B_1^T$, we consider

$$Z = \begin{bmatrix} A+BA^{-T}C & -BA^{-T} \\ -A^{-T}C & A^{-T} \end{bmatrix} \in \mathbf{R}^{2n\times 2n}.$$

Then the equation (4.2) can be solved by computing an orthogonal matrix U such that

$$U^TZU = S$$

where

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

and

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \quad S_{ij} \in \mathbf{R}^{n \times n}.$$

It is possible to arrange, moreover, that the spectrum of S_{11} lies inside the unit circle and that the spectrum of S_{22} lies outside the unit circle. Again the solution is given by $X = U_{21}U_{11}^{-1}$.

Remarks ([15])

- In the discrete time case, the Schur method has required an explicit inversion of A. If this matrix is ill-conditioned, numerical difficulties arise (recently, the new method has been derived by considering the generalized eigenvalue problem. see [21]).
- This method has the storage requirement of at least two $2n \times 2n$ arrays.
- For the computed $X=U_{21}U_{11}^{-1}$, there is no guarantee of symmetry.
- Obviously the first n columns of an orthogonal matrix U span the invariant subspace of Z.

CHAPTER 5

THE ITERATIVE METHODS FOR SOLVING ALGEBRAIC RICCATI EQUATIONS

As we have seen, the Riccati equation

$$A_{22}R - RA_{11} = -A_{21} + RA_{12}R$$

which corresponds to the matrix system

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad A_{11} \in \mathbf{R}^{n\times n}, \quad A_{22} \in \mathbf{R}^{m\times m}$$

has been extensively studied in mathematics and engineering. It appears in a rich variety of situations and is used in many fields, for example, in singular perturbation systems, control theory and in general for boundary value problems for ordinary differential equations.

One aspect of Riccati equations that has always been significant and which has received increasing attention is effective algorithms for their reliable numerical solution in the finite arithmetic environment of a digital computer. Reliable numerical algorithms and software now exist for the solution of many problems in linear algebra, for example, for the singular value decomposition, linear least squares problem, and both standard and generalized eigenvalue problems. But this has not been the case until recently for solving Riccati equations and so it is important to compare and analyze the fairly recent numerical methods which have been proposed.

In this chapter we consider three iterative methods, which are as follow:

The linear convergence method ([4])

Given R_0,

$$A_{22}R_i - R_iA_{11} = -A_{21} + R_{i-1}A_{12}R_{i-1}, \quad i=1,2,....$$

Newton's method ([4])

Given R_0,

$$(A_{22} - R_{i-1}A_{12})R_i - R_i(A_{11} + A_{12}R_{i-1}) = -A_{21} - R_{i-1}A_{12}R_{i-1}, \quad i=1,2,....$$

The generalized secant method ([20])

Given R_0, R_{-1},

$$(A_{22} - R_{i-1}A_{12})R_i - R_i(A_{11} + A_{12}R_{i-2}) = -A_{21} - R_{i-1}A_{12}R_{i-2},$$

$$(A_{22} - R_{i-1}A_{12})R_{i+1} - R_{i+1}(A_{11} + A_{12}R_i) = -A_{21} - R_{i-1}A_{12}R_i, \quad i=1,3,5....$$

Since these three methods (called LCM, NEM and SEM, respectively), all require the solution of the equation of same type which is called the *Sylvester* equation

$$AX + XB = C, \tag{5.1}$$

the algorithm for each method has been implemented by using Sylvester equation algorithms.

Sylvester equation algorithms have been devised by Bartels-Stewart [16] and Golub-Nash-Van Loan [17]. Both algorithms are based on the equivalence between (5.1) and

$$(U^TAU)(U^TXV) + (U^TXV)(V^TBV) = U^TCV$$

where U, V are orthogonal matrices and involve five steps :

1. For the *Bartels-Stewart* algorithm, transform A into upper real Schur form $A' = U^T AU$ by an orthogonal similarity transformation U. For the *Golub-Nash-Van Loan* algorithm, transform A into upper Hessenberg form $A' = U^T AU$ by an orthogonal similarity transformation U.
2. Transform B into lower real Schur form $B' = V^T BV$ by an orthogonal similarity transformation V.
3. Compute $C' = U^T CV$.
4. Solve the transformed system $A'X' + X'B' = C'$ for X'.
5. Compute $X = UX'V^T$.

Moreover, solving the Riccati equation in control problems

$$A_{11}^T R + RA_{11} = -A_{21} + RA_{12}R$$

where all matrices are in $\mathbf{R}^{n\times n}$ and A_{12} and A_{21} are symmetric nonnegative definite matrices, LCM and NEM (in this case, we assume that we have a symmetric initial guess) require the solution of same type which is the symmetric case of the Sylvester equation

$$A^T X + XA = C$$

where all matrices are in $\mathbf{R}^{n\times n}$ and C is symmetric. In this case, the Bartels-Stewart algorithm involves only four steps, which are as follows:

1. Transform A into upper real Schur form $A' = U^T AU$ by an orthogonal similarity transformation U.
2. Compute $C' = U^T CU$.
3. Solve the transformed system $A'^T X' + X'A' = C'$ for X'.
4. Compute $X = UX'U^T$.

5.1 Algorithm and operation counts

We describe the algorithms for the linear convergence method, Newton's method and the generalized secant method:

Algorithm I for the linear convergence method (general case)

1. Choose initial guess R.
2. Transform $A = A_{22}$ into upper real Schur form (or Hessenberg form). Transform $B = -A_{11}$ into lower Schur form.
3. Obtain $C = -A_{21} + RA_{12}R$.
4. Perform Sylvester equation algorithm steps 3, 4, 5, update R and return to 3.

Algorithm II for the linear convergence method (symmetric case)

1. Choose symmetric initial guess R.
2. Transform $A = A_{11}$ into upper real Schur form.
3. Obtain $C = -A_{21} + RA_{12}R$.
4. Perform the symmetric version of Sylvester equation algorithm steps 2, 3, 4, update R and return to 3.

Algorithm I for Newton's method (general case)

1. Choose initial guess R.
2. Obtain $A = A_{22} - RA_{12}$.

   Obtain $B = -(A_{11} + A_{12}R)$.

   Obtain $C = -A_{21} - RA_{12}R$.
3. Perform Sylvester equation algorithm steps 1, 2, 3, 4, 5, update R and return to 2.

Algorithm II for Newton's method (symmetric case)

1. Choose symmetric initial guess R.

2. Obtain $A = A_{11} - A_{12}R$.

   Obtain $C = -A_{21} - RA_{12}R$.

3. Perform the symmetric version of Sylvester equation algorithm steps 1, 2, 3, 4, update R and return to 2.

Algorithm for the generalized secant method

1. Choose initial guess R.

2. Obtain $A = A_{22} - RA_{12}$.

   Obtain $B = -(A_{11} + A_{12}R)$.

   Obtain $C = -A_{21} - RA_{12}R$.

3. Perform Sylvester equation algorithm steps 1, 2, 3, 4, 5 and update R.

4. Save A.

   Obtain $B = -(A_{11} + A_{12}R)$.

   Save RA and obtain $C = -A_{21} - RA_{12}R$.

5. Perform Sylvester equation algorithm steps 2, 3, 4, 5 and update R.

6. Save B.

   Obtain $A = A_{22} - RA_{12}$.

   Obtain $C = -A_{21} - RA_{12}R$.

7. Perform Sylvester equation algorithm steps 1, 3, 4, 5, update R and return to 4.

We now give rigorous operation counts of each iterative method for solving the general Riccati equation. Since all three algorithms are based on Sylvester equation algorithms, we first need to check operation counts for the Bartels-Stewart and the Golub-Nash-Van Loan algorithms.

Operation counts are summarized in the following table ([17]).

Table 1

Operation counts for the Bartels-Stewart and the Golub-Nash-Van Loan algorithms

	Bartels-Stewart	Golub-Nash-Van Loan
step 1	$10m^3$	$5/3m^3$
step 2	$10n^3$	$10n^3$
step 3	m^2n+mn^2	m^2n+mn^2
step 4	$1/2(m^2n+mn^2)$	$3m^2n+1/2mn^2$
step 5	m^2n+mn^2	m^2n+mn^2

We require m^2n operations for obtaining $A = A_{22} - RA_{12}$, mn^2 operations for obtaining $B = -(A_{11} + A_{12}R)$ and $2m^2n$ operations for obtaining $C = -A_{21} - RA_{12}R$ (in SEM step 4, we only need m^2n operations since we save RA_{12}). From this, we can now construct operation counts tables for the three iterative methods.

The operation counts using the Bartels-Stewart algorithm are as follows:

Table 2

Operation counts for LCM, NEM and SEM using the Bartels-Stewart algorithm

Method	i=1	$i \geq 2$
LCM	$10(m^3+n^3)+9/2m^2n+5/2mn^2$	$9/2m^2n+5/2mn^2$
NEM	$10(m^3+n^3)+11/2m^2n+7/2mn^2$	$10(m^3+n^3)+11/2m^2n+7/2mn^2$
SEM	$10(m^3+n^3)+11/2m^2n+7/2mn^2$	i=even, $10n^3+7/2(m^2n+mn^2)$ i=odd, $10m^3+11/2m^2n+5/2mn^2$

The counts using the Golub-Nash-Van Loan algorithm are as follows:

Table 3

Operation counts for LCM, NEM and SEM using the Golub-Nash-Van Loan algorithm

Method	$i=1$	$i \geqslant 2$
LCM	$5/3m^3+10n^3+7m^2n+5/2mn^2$	$7m^2n+5/2mn^2$
NEM	$5/3m^3+10n^3+8m^2n+7/2mn^2$	$5/3m^3+10n^3+8m^2n+7/2mn^2$
SEM	$5/3m^3+10n^3+8m^2n+7/2mn^2$	i=even, $10n^3+6m^2n+7/2mn^2$ i=odd, $5/3m^3+8m^2n+5/2mn^2$

Generally, the Golub-Nash-Van Loan algorithm is faster than the Bartels-Stewart algorithm. Since both algorithms have the same accuracy, it is a good idea to use the Golub-Nash-Van Loan algorithm for solving nonsymmetric Riccati equations (especially when $m >> n$).

In the symmetric case, the Bartels-Stewart algorithm only requires $13.5n^3$ instead of $25n^3$ operations. Thus NEM is as fast as SEM. The details of operation counts are summarized in the following table.

Table 4

Operation counts for LCM, NEM and SEM in the symmetric case

Method	$i=1$	$i \geqslant 2$
LCM	$15.5n^3$	$5.5n^3$
NEM	$16.5n^3$	$16.5n^3$
SEM	$23n^3$	i=even, $19.5n^3$ i=odd, $12n^3$

Remarks

- Since the Golub-Nash-Van Loan algorithm requires only upper Hessenberg form, a bit more operations (when compare with the Bartels-Stewart algorithm) are needed for

solving the transformed system $A'X' + X'B' = C'$ (see the step 4 of a table 1 and the counts for LCM of tables 2, 3). However, when we consider the entire process and $m \geqslant n$ the Golub-Nash-Van Loan algorithm is more efficient than the Bartels-Stewart algorithm (in the nonsymmetric case). This can also be assured, for if $m<n$, we merely apply to the transposed problem

$$B^T X^T + X^T A^T = C^T.$$

- In the symmetric case, after the first iteration, the average operation counts for SEM at each iteration is $16n^3$, while NEW requires $16.5n^3$ operations.

5.2 The rate of convergence and condition number

In chapter 2, we mentioned that the Stewart method for computing the invariant subspace was originally presented not as an algorithm but as a technique for doing perturbation theory for invariant subspaces, although it works as an algorithm as well. Using this perturbation theory, convergence criteria for all three methods has been derived. To state these results, we need to define the separation of two matrices **sep**(A_{11},A_{22}). By using Stewart definition, we can denote sep(A_{11},A_{22}) by sep (A_{11},A_{22}) = min$|\Sigma(A_{11})-\Sigma(A_{22})|$ where $\Sigma(A_{11})$ and $\Sigma(A_{22})$ represent the sets of singular values of A_{11} and A_{22} respectively.

Let

$$K = \frac{\|A_{12}\|_F \|A_{21}\|_F}{\text{sep}^2(A_{11},A_{22})}$$

and $E_i = R_i - R$. If $K<1/4$ then LCM converges to a solution R which is the unique solution inside the ball

$$\|R\|_F < 2\frac{\|A_{21}\|_F}{\text{sep}(A_{11},A_{22})}$$

so that $\|E_i\|_F \leqslant C_L\|E_{i-1}\|_F$, where the constant C_L depends on K (see [4]). If $K<1/12$

then NEM converges to a solution R inside the ball given in the linear case so that $\|E_i\|_F \leq C_N\|E_{i-1}\|_F^2$, where the constant C_N depends on K (see [4]). Also if $K<1/12$ then SEM converges to a solution R inside the ball given in the linear case so that $\|E_i\|_F \leq C_S\|E_{i-1}\|_F\|E_{i-2}\|_F$, where the constant C_S depends on K (see [20]). According to these results, the order of convergence for LCM, NEM and SEM are 1, 2 and $\frac{1+\sqrt{5}}{2}$, respectively.

Since we have considered iterative methods for solving Riccati equations as iterative methods for computing invariant subspaces, the parameter K can be interpreted as follows: $\|A_{12}\|_F\|A_{21}\|_F$ measures the quality of the initial approximate invariant subspace, and moreover $\|A_{21}\|_F$ will be zero if and only if the initial approximation is in fact the exact invariant subspace. The quantity $\mathrm{sep}(A_{11},A_{22})$ measures the separation of the spectra of A_{11} and A_{22}. If sep is small it means that the invariant subspaces belonging to the two parts of the spectrum are unstable and hard to compute. If we denote the invariant subspace corresponding to A_{11} by $\begin{bmatrix} I \\ R \end{bmatrix}$, then K will be small if we start with good initial guess to R and if the eigenvalues associated with A_{11} are well separated from the spectrum of A_{22}.

5.3 Advantages and disadvantages

We have seen the algorithms, operation counts and the rate of convergence for each method. Here, we give their advantages and disadvantages.

From the operation counts tables, it is obvious that LCM [4] is much cheaper than NEM and SEM at each iteration. Especially when $m >> n$, because this method then only needs $O(m^2)$ operations after the first iteration.

In all three methods, it is important to choose a good initial guess R_0. We cannot guarantee that LCM always converges, even if we give an apparently accurate initial approximation (see **Example 6,7**).

Consider nonsymmetric Riccati equation $A_{22}R - RA_{11} = -A_{21} + RA_{12}R$. For NEM [4], the coefficient matrices $(A_{22} - R_{i-1}A_{12})$ and $(A_{11} + A_{12}R_{i-1})$ vary from step to step, making it necessary to preprocess at each step. Thus each step could take $O(m^3)$ operations, which is quite expensive.

SEM [20] consider this cost problem and moreover the order of convergence $(\frac{1+\sqrt{5}}{2} \approx 1.6)$ is between LCM and NEM so SEM has superlinear convergence properties as NEM.

To compare the relative cost of SEM and NEM, we now construct another table (for when $m >> n$).

Table 5

Operation counts for NEM and SEM (when $m >> n$)

Method	i=1	$i \geqslant 2$
NEM	$O(m^3)$	$O(m^3)$
SEM	$O(m^3)$	i=even, $O(m^2)$ i=odd, $O(m^3)$

From tables 2, 3 and 5, it is obvious that the cost for SEM is less than the cost for NEM at each iteration.

Solving the symmetric Riccati equation $A_{11}^T R + RA_{11} = -A_{21} + RA_{12}R$ which arises in control problems, NEM is as fast as SEM at each iteration provided a symmetric starting value is known. Since NEM converges faster than SEM, NEM is more efficient than SEM. However, if we get the initial guess from a QR-type process, such as

the Schur method, then there is no guarantee of symmetry. Thus in this case SEM is still competitive with NEM.

We know that LCM, NEM and SEM have been devised by using Sylvester equation algorithms. The Bartels-Stewart and the Golub-Nash-Van Loan algorithms are derived under the assumption that the Sylvester equation

$$AX + XB = C$$

has a unique solution, that is, $\lambda(A) \cap \lambda(-B) = \emptyset$, where $\lambda(A)$ and $\lambda(-B)$ are the set of eigenvalues of A and $-B$, respectively. Thus if there is a coalescence among the eigenvalues of A and $-B$ then Sylvester equation algorithms are unstable. It means that if $\lambda(A_{22}) \cap \lambda(A_{11}) \neq \emptyset$ then LCM is unstable, since LCM is trying to solve the Sylvester equation

$$A_{22}R + R(-A_{11}) = -A_{21} + RA_{12}R.$$

Also, SEM and NEM are unstable if $\lambda(A'_{22}) \cap \lambda(A'_{11}) \neq \emptyset$, because the Sylvester equation which they are trying to solve becomes

$$A'_{22}R + RA'_{11} = -A_{21} + RA_{12}R$$

where $A'_{22} = A_{22} - RA_{12}$, $A'_{11} = -(A_{11} + A_{12}R)$ (for SEM, $A'_{11} = -(A_{11} + A_{12}R')$).

5.4 Numerical examples and comparisons

We now present a number of examples to illustrate various points discussed in previous sections and to compare the iterative methods with Kokotovic's iterative method and the Schur method (called KIM and SCM, respectively), which are known to be reliable and efficient methods for solving Riccati equations in singular perturbation and control problems, respectively. All computations have been performed using the FORTRAN-G compiler and double precision on an IBM 3081 (our machine precision is near 10^{-16}).

5.4.1 Examples from singular perturbation problems.

Consider the simple example of a singularly perturbed system with m large and n small eigenvalues

$$\begin{pmatrix} \dot{x} \\ \epsilon \dot{z} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix}, \quad A_{11} \in \mathbf{R}^{n \times n}, A_{22} \in \mathbf{R}^{m \times m}$$

and Kokotovic's iterative method [6]:

$$R_i = A_{22}^{-1}(R_{i-1}A_{11} - R_{i-1}A_{12}R_{i-1} + A_{21}), \quad i=1,2,\ldots,$$

(If a singular perturbation system has n large and m small eigenvalues then we use $R_i = (A_{22}R_{i-1} + R_{i-1}A_{12}R_{i-1} - A_{21})A_{11}^{-1}$, $i=1,2,\ldots$). We know that KIM is efficient, in solving Riccati equations arising from singular perturbation problems, since the rate of convergence is proportional to ϵ, and after inversion, we need only $3m^2n+mn^2$ operations at each iteration.

We now choose several examples from singular perturbation systems to compare the three iterative methods with KIM. All computation have been performed by using both the Bartels-Stewart and the Golub-Nash-Van Loan algorithms with the initial guess $R_0 = 0$. Moreover, to get 7 digits accuracy, we use the relative error as the stop criterion, i.e. if $\| \frac{R_N - R_{N-1}}{R_N} \|_F < 10^{-7}$ then N is the final number of iterations.

These following examples show that SEM and NEM are very competitive with KIM. First each example and a computed solution are presented.

Example 1 :Kokotovic [6]

Consider the continuous model of a power system

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

where $A_{11} \in \mathbf{R}^{2\times 2}$, $A_{22} \in \mathbf{R}^{3\times 3}$, $x \in \mathbf{R}^2$, $z \in \mathbf{R}^3$ and

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} -0.11 & 0.02 & 0.03 & 0.0 & 0.02 \\ 0.0 & -0.17 & 0.0 & 0.0 & 0.17 \\ 0.0 & 2.0 & -4.0 & 0.0 & 0.0 \\ -4.0 & 0.0 & 0.0 & -2.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.75 & -5.0 \end{bmatrix}$$

KIM and the three iterative methods, by using both the Bartels-Stewart algorithm and the Golub-Nash-Van Loan algorithm, all give the same solution

$$R = \begin{bmatrix} -0.04840394 & -0.5203825 \\ 2.154609 & -0.04306733 \\ 2.106174 & -0.05827145 \end{bmatrix}.$$

Example 2 :Phillips [7]

Consider the discrete model of a steam power system

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix}$$

where $A_{11} \in \mathbf{R}^{2\times 2}$, $A_{22} \in \mathbf{R}^{3\times 3}$, $x(k) \in \mathbf{R}^2$, $z(k) \in \mathbf{R}^3$ and

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 0.9014 & 0.1179 & 0.0525 & 0.0167 & 0.02104 \\ -0.0196 & 0.8743 & 0.0 & 0.025 & 0.02934 \\ -0.0071 & 0.7342 & 0.20175 & 0.013 & 0.21067 \\ -0.75 & -0.0557 & -0.032 & 0.19357 & -0.014076 \\ -0.306 & -0.01694 & -0.011 & 0.14278 & 0.013217 \end{bmatrix}$$

All methods give the same solution

$$R = \begin{bmatrix} 0.09316662 & -1.140467 \\ 1.083789 & -0.1514285 \\ 0.5308208 & -0.1025373 \end{bmatrix}.$$

Example 3 :Phillips [7]

Consider the discrete model of a power system

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix}$$

where $A_{11}, A_{22} \in \mathbf{R}^{4\times4}$, $x(k), z(k) \in \mathbf{R}^4$ and

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 0.835 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.096 & 0.861 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.029 \\ -0.002 & -0.005 & 0.882 & -0.253 & 0.041 & -0.003 & -0.025 & -0.001 \\ 0.007 & 0.014 & -0.029 & 0.928 & 0.0 & 0.006 & 0.059 & 0.002 \\ -0.03 & -0.061 & 2.028 & -2.303 & 0.088 & -0.021 & -0.224 & -0.008 \\ 0.048 & 0.758 & 0.0 & 0.0 & 0.0 & 0.165 & 0.0 & 0.023 \\ -0.012 & -0.027 & 1.209 & -1.4 & 0.161 & -0.013 & 0.156 & 0.006 \\ 0.815 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.011 \end{bmatrix}$$

All methods give the same solution

$$R = \begin{bmatrix} 0.01451242 & 0.03516997 & -2.079860 & 1.813171 \\ 0.09707658 & -1.089080 & 0.0 & 0.0 \\ -0.02188941 & -0.01141115 & -2.255242 & 1.524221 \\ -0.9890777 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Example 4 :Phillips [18]

Consider the continuous model of a power system

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

where $A_{11}, A_{22} \in \mathbf{R}^{4\times4}$, $x, z \in \mathbf{R}^4$ and

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} -5.0 & 0.0 & 0.0 & 0.0 & 4.75 & 0.0 & 0.0 & 0.0 \\ 0.0 & -2.0 & 0.0 & 0.0 & 0.0 & -2.0 & 0.0 & 0.0 \\ -0.08 & -0.11 & -3.99 & -0.93 & 0.0 & -0.07 & 10.0 & -9.1 \\ 0.0 & 0.0 & 1.32 & -1.39 & 0.0 & 0.0 & 0.0 & -0.28 \\ 0.0 & 0.0 & 0.0 & 0.0 & -0.2 & 0.0 & 0.0 & 0.0 \\ 0.17 & 0.0 & 0.0 & 0.0 & 0.0 & -0.17 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & -0.5 & 0.0 \\ 0.01 & 0.01 & -0.06 & 0.12 & 0.0 & 0.01 & 0.0 & -0.11 \end{pmatrix}$$

All methods give the same solution

$$R = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.03519669 & 0.0 & 0.0 & 0.0 \\ -0.0006385651 & -0.001647461 & 0.05102651 & -0.002348209 \\ 0.001882734 & 0.004962614 & 0.01556921 & 0.09525143 \end{pmatrix}$$

To compare each method, we construct a table which shows how many iterations N we need to get 7 digits accuracy and which gives the condition number $K = \dfrac{\|A_{12}\|_F \|A_{21}\|_F}{\mathrm{sep}^2(A_{11}, A_{22})}$ and the residual r.

Table 6

Numerical results I for examples 1, 2, 3 and 4

Ex	K	KIM		LCM		SEM		NEM	
		N	r	N	r	N	r	N	r
1	0.08604	9	1.0-7	7	1.0-7	5	1.0-14	4	1.0-14
2	0.2224	15	1.0-7	10	1.0-8	5	1.0-13	4	1.0-14
3	1.2552	16	1.0-7	11	1.0-7	6	1.0-14	4	1.0-14
4	3.5253	15	1.0-8	13	1.0-8	7	1.0-15	5	1.0-15

N: The number of iterations for 7 digits accuracy

$$r = \|A_{22}R_N - R_N A_{11} + R_N A_{12} R_N - A_{21}\|_F$$

From table 6, we see that SEM and NEM solutions have much better accuracy.

It is not easy to give general results for the actual cost, since this depends on the number of iterations and the size of matrix. But if we need very accurate solutions of Riccati equations then SEM and NEM are generally a bit cheaper here than KIM (but if ϵ is very small then they are not, see example 5). In each example, we also determine the number of iterations required to make the residual less than 10^{-14} and then compute operation counts and CPU time when using the Golub-Nash-Van Loan algorithm. The numerical results are summarized in the following table.

Table 7

Numerical results II for examples 1, 2, 3 and 4

Ex	KIM			LCM			SEM			NEM		
	N	Op	Time	N	Op	Time	N	Op	Time	N	Op	Time
1	16	1083	0.0054	13	2153	0.013	5	1209	0.0065	4	1244	0.0073
2	29	1922	0.01	18	2933	0.023	6	1439	0.009	4	1244	0.0068
3	32	8265	0.027	22	14112	0.063	6	6764	0.022	4	5888	0.016
4	26	6729	0.021	25	15936	0.07	7	7538	0.024	5	7360	0.022

N: The number of iterations satisfying $r < 10^{-14}$

$$r = \|A_{22}R_N - R_N A_{11} + R_N A_{12} R_N - A_{21}\|_F$$

Op: Operation counts

Time: CPU time (sec)

We now construct tables to show the rate of convergence for each method, where

$$\text{relerr} = \frac{\|R_i - R_{i-1}\|_F}{\|R_i\|_F}.$$

Table 8

The rate of convergence for KIM

i	relerr (Ex1)	relerr (Ex2)	relerr (Ex3)	relerr (Ex4)
1	.1+1	.1+1	.1+1	.1+1
2	.84-1	.21+0	.19+0	.19+0
3	.65-2	.62-1	.51-1	.76-1
4	.71-3	.21-1	.15-1	.22-1
5	.16-3	.71-2	.48-2	.50-2
6	.23-4	.23-2	.16-2	.11-2
7	.23-5	.75-3	.62-3	.48-3
8	.15-6	.23-3	.24-3	.21-3
9	.14-7	.67-4	.94-4	.71-4
10		.19-4	.35-4	.18-4
11		.54-5	.12-4	.36-5
12		.16-5	.41-5	.99-6
13		.50-6	.13-5	.49-6
14		.17-6	.43-6	.19-6
15		.61-7	.15-6	.59-7
16			.56-7	

Table 9

The rate of convergence for LCM

i	relerr (Ex1)	relerr (Ex2)	relerr (Ex3)	relerr (Ex4)
1	.1+1	.1+1	.1+1	.1+1
2	.29-1	.82-1	.63-1	.38+0
3	.20-2	.11-1	.93-2	.21-1
4	.22-3	.18-2	.17-2	.20-1
5	.64-5	.27-3	.52-3	.33-2
6	.85-6	.37-4	.13-3	.90-3
7	.53-7	.63-5	.24-4	.35-3
8		.83-6	.45-5	.25-4
9		.14-6	.13-5	.24-4
10		.21-7	.34-6	.42-5
11			.61-7	.10-5
12				.43-6
13				.30-7

Table 10

The rate of convergence for SEM

i	relerr (Ex1)	relerr (Ex2)	relerr (Ex3)	relerr (Ex4)
1	.1+1	.1+1	.1+1	.1+1
2	.28-1	.84-1	.63-1	.53+0
3	.66-3	.59-2	.92-2	.21+0
4	.13-5	.38-4	.11-3	.16-1
5	.77-10	.15-7	.11-6	.95-3
6			.13-11	.54-5
7				.16-8

Table 11

The rate of convergence for NEM

i	relerr (Ex1)	relerr (Ex2)	relerr (Ex3)	relerr (Ex4)
1	.1+1	.1+1	.1+1	.1+1
2	.28-1	.86-1	.70-1	.40+0
3	.62-4	.55-3	.27-3	.24-1
4	.27-9	.20-7	.16-7	.15-3
5				.72-8

The next example illustrates the fact that KIM is extremely useful for singularly perturbed systems with small ϵ.

Example 5 :Allemong, Kokotovic [19]

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 1/\epsilon A_{21} & 1/\epsilon A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbf{R}^{2\times 2}$, $A_{22} \in \mathbf{R}^{5\times 5}$, $x \in \mathbf{R}^2$, $z \in \mathbf{R}^5$, $\epsilon = 0.1$ and

$$\begin{bmatrix} A_{11} & A_{12} \\ \frac{1}{\epsilon}A_{21} & \frac{1}{\epsilon}A_{22} \end{bmatrix} = \begin{bmatrix} -0.58 & 0.0 & 0.0 & -0.27 & 0.0 & 0.2 & 0.0 \\ 0.0 & -1.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & -5.0 & 2.1 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 377 & 0.0 & 0.0 \\ -0.14 & 0.0 & 0.14 & -0.2 & -0.28 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.08 & 2.0 \\ -173 & 66.7 & -116 & 40.9 & 0.0 & -66.7 & -16.7 \end{bmatrix}.$$

By simple multiplication, we can also contruct examples for $\epsilon<0.1$. The results are in the table below.

Table 12

Numerical results for example 5

ϵ	K	KIM		LCM		SEM		NEM	
		N	r	N	r	N	r	N	r
1.0-1	246.4	11	1.0-8	13	1.0-7	6	1.0-12	5	1.0-13
1.0-2	5833.8	6	1.0-10	6	1.0-8	4	1.0-11	4	1.0-11
1.0-4	35.2	3	1.0-11	3	1.0-7	3	1.0-10	3	1.0-10

N: The number of iterations for 7 digits accuracy

$$r = \|A_{22}R_N - R_N A_{11} + R_N A_{12} R_N - A_{12}\|_F$$

The above table shows that KIM needs the same number of iterations to get 7 digits accuracy as the other three methods provided $\epsilon \lesssim 10^{-4}$. Thus, in this case (i.e. a singulary perturbated system with small ϵ), KIM is the best method for solving Riccati equations. Futhermore, the iterative methods work fine, even though K is pretty big.

We have performed all computations for the three iterative methods by using both the Bartels-Stewart and the Golub-Nash-Van Loan algorithms. As expected, computed solutions for both algorithms are identical. Since if we use the Golub-Nash-Van Loan algorithm then we can save some operation counts (see **Table 2, Table 3**), it is a good idea to use it for solving Riccati equations in singular perturbation problems.

5.4.2 Examples from control problems

Consider the continuous time algebraic Riccati equation arising from control problems

$$A^TR + RA - RBR + C = 0$$

where all matrices are in $\mathbf{R}^{n\times n}$ and B and C are symmetric nonnegative definite matrices. We know that the Schur method considers

$$Z = \begin{bmatrix} A & -B \\ -C & -A^T \end{bmatrix} \in \mathbf{R}^{2n\times 2n}$$

and reduces Z to the real Schur form such that

$$U^TZU = S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}.$$

The solution is given by $R = U_{21}U_{11}^{-1}$.

To transform Z into the real Schur form, we generally need $10(2n)^3$ operations and moreover we need $\frac{4}{3}n^3$ for obtaining $R = U_{21}U_{11}^{-1}$. We see that the total number of operations required is at least $81n^3$. Should the ordering of the real Schur form require, say, 25 percent more operations than the unordered real Schur form, we have about $101n^3$ for the entire process.

Remark

Recall that SEM and NEM require approximately $16n^3$ and $16.5n^3$ at each iteration, respectively (at the first iteration, SEM requires $23n^3$). If SEM and NEM need about 5 iterations to get the solution which has the same accuracy as that of SCM, then we can say both methods are competitive with SCM (for the case that any particular ordering of the real Schur form is not needed).

We now consider a few examples to compare the three iterative methods to SCM. We know that the Bartels-Stewart algorithm is more efficient than the Golub-Nash-Van Loan algorithm for LCM and NEM. Thus all computations for them have been performed by using the Bartels-Stewart algorithm. However, we use the Golub-Nash-Van Loan algorithm for SEM.

The following simple continuous-time example is used to illustrate the effect of an ill-conditioned problem (problems where the data can be perturbed slightly but the resulting change in the solution is large) and an initial guess.

Example 6 :Arnold, Laub [14]

$$\dot{x} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} \epsilon \\ 0 \end{bmatrix} u$$

$$y = (1 \ , \ 1)x$$

$$\text{minimize} \quad \int_0^\infty (y^T y + u^T u)dt.$$

The applicable algebraic Riccati equation is

$$A^T R + RA - RBR + C = 0 \tag{5.2}$$

where $\epsilon = 10^{-m}$ and

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} \epsilon^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

When we solve the algebraic Riccati equation (5.2) with an initial guess R_0=0, LCM, NEM and SEM give the same stable solution. The numerical results are summarized in the following table.

Table 13

Numerical results I for example 6

m	LCM		SEM		NEM	
	N	r	N	r	N	r
0	36	1.0-12	8	1.0-15	6	1.0-15
2	5	1.0-16	4	1.0-16	3	1.0-16
4	3	1.0-16	3	1.0-16	3	1.0-16
6	3	1.0-16	3	1.0-16	3	1.0-16
10	2	1.0-16	2	1.0-16	2	1.0-16

N: The number of iterations for 13 digits accuracy

$$r = \|A^T R_N + R_N A - R_N B R_N + C\|_F$$

Recall that SCM has been devised to get the unique symmetric nonnegative definite solution, and a symmetric matrix is nonnegative definite if and only if all eigenvalues are nonnegative. Our computed solutions which are derived by iterative schemes with a zero initial guess are symmetric, but since some of the eigenvalues are negative, we know that these stable solutions are not the desired solutions.

The true symmetric nonnegative definite solution for R can be hand-calculated as

$$R = \begin{bmatrix} \dfrac{1+\sqrt{1+\epsilon^2}}{\epsilon^2} & \dfrac{1}{2+\sqrt{1+\epsilon^2}} \\ \dfrac{1}{2+\sqrt{1+\epsilon^2}} & \dfrac{1}{4} - \dfrac{\epsilon^2}{4(2+\sqrt{1+\epsilon^2})^2} \end{bmatrix}.$$

Note that as $\epsilon \to 0$, the (1,1) element of R tends to infinity. It suggests that we need another initial guess with a large (1,1) element.

We solve the Riccati equation (5.2) with $R_0 = \begin{bmatrix} R_0(1,1) & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$. SEM and NEM give the unique symmetric nonnegative definite solution (as does SCM). Since the unique symmetric nonnegative definite solution to this example is unstable as $\epsilon \to 0$, the solution accuracy is degenerating. When ϵ=1.0, the problem is well-conditioned and we

have an accurate initial approximation, but LCM gives overflow. The numerical results of interest are summarized in the following table.

Table 14

Numerical results II for example 6

m	$R_0(1,1)$	LCM	SEM		NEM		SCM
			N	r	N	r	r
0	1.0+1	overflow	10	1.0-14	7	1.0-14	1.0-14
2	1.0+5	overflow	11	1.0-11	8	1.0-11	1.0-11
4	1.0+9	overflow	11	1.0-6	8	1.0-7	1.0-6
6	1.0+13	overflow	11	1.0-3	8	1.0-3	1.0-3
8	1.0+17	overflow		1.0+1		1.0+1	1.0+1

N: The number of iterations satisfying r

$$r = \|A^T R_N + R_N A - R_N B R_N + C\|_F$$

We can see that SEM and NEM have the same accuracy as SCM. The execution times of both methods are greater than for SCM. However, we expect more careful implementations of the two iterative methods to be faster than the current versions.

The following example illustrates the effect of ill-conditioning of A_{12} with respect to inversion. Recall that A_{12} measures the quality of the initial approximation.

Example 7 :Arnold, Laub [14]

$$\dot{x} = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.02 \end{bmatrix} x + \begin{bmatrix} 0.1 & 0 \\ 0.001 & 0.01 \end{bmatrix} u$$

$$y = (10,\ 100)x$$

$$\text{minimize} \int_0^\infty (y^T y + u^T \begin{bmatrix} 1+\epsilon & 1 \\ 1 & 1 \end{bmatrix} u) dt, \quad \epsilon = 10^{-m}$$

The applicable algebraic Riccati equation is

$$A^TR + RA - RBD^{-1}B^TR + C = 0$$

where

$$A = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.02 \end{bmatrix}, \quad C = \begin{bmatrix} 100 & 1000 \\ 1000 & 10000 \end{bmatrix},$$

$$BD^{-1}B^T = \begin{bmatrix} 0.1 & 0 \\ 0.001 & 0.01 \end{bmatrix} \begin{bmatrix} 1+\epsilon & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.1 & 0.001 \\ 0 & 0.01 \end{bmatrix}.$$

We solve this problem by using the identity matrix I as an initial guess. SEM and NEM generate the unique symmetric nonnegative definite solution, even though LCM gets overflow all the time, The numerical results of interest are summarized in the following table.

Table 15

Numerical results for example 7

m	SEM		NEM		SCM
	N	r	N	r	r
0	14	1.0-11	13	1.0-11	1.0-10
2	9	1.0-10	12	1.0-10	1.0-9
4	12	1.0-8	11	1.0-8	1.0-7
6	16	1.0-7	11	1.0-6	1.0-5
8	20	1.0-4	14	1.0-4	1.0-3
10	25	1.0-2	17	1.0-3	1.0-1

N: The number of iterations satisfying r

$$r = \|A^TR_N + R_NA - R_NBD^{-1}B^TR_N + C\|_F$$

SEM and NEM give the same accurate solution as SCM. But we have difficulty in obtaining a good initial guess as $\epsilon \to 0$.

CHAPTER 6

STEEPEST DESCENT TECHNIQUES

We have seen that SEM and NEM are competitive with well-known KIM and SCM. Their weakness is the usual requirement that an accurate initial approximation to the solution is needed to ensure convergence. This leads us to consider the steepest descent method which is used to find sufficiently accurate starting approximations for iterative methods.

6.1 The matrix algorithms of steepest descent type

Consider the Riccati equation

$$F(R) = A_{22}R - RA_{11} - RA_{12}R + A_{21} = 0. \tag{6.1}$$

The Riccati equation (6.1) will have a solution precisely when the function $G:\mathbf{R}^{n\times m} \to \mathbf{R}$ defined by $G(R) = \|F(R)\|_F^2$ has the minimal value zero. To minimize G, we use the method of steepest descent which determines a local minimum for a function of the form $G:\mathbf{R}^r \to \mathbf{R}$.

The basis of this method for finding a local minimum for a function G is the natural idea of choosing R to be in the direction of the greatest decrease in the value of G, namely, $-\nabla G(x)$ where $\nabla G(x)$ denotes the gradient of $G(x)$, direction of the gradient. The numerical details of a matrix algorithm of steepest descent type are as follows:

Algorithm I

1. Given an initial approximation R_0, evaluate $G(R_0) = \|F(R_0)\|_F^2$ and $\nabla G(R_0)$.

2. Make $\nabla G(R_0)$ a unit vector.

3. Choose α_3 such that $G(R_0) \geq G(R_0 - \alpha_3 \nabla G(R_0))$.

4. Set $\alpha_1=0$, $\alpha_2=\alpha_3/2$ and determine the quadratic polynomial that interpolates $G(R_0-\alpha\nabla G(R_0))$ at $\alpha=\alpha_1$, α_2 and α_3.

5. Determine a critical point α_0 of this quadratic.

6. Define α to be the number that minimizes this quadratic.

7. Return to 1 with R_0 replaced by $R=R_0-\alpha\nabla G(R_0)$.

Since we need $3m^2n+mn^2$ operations to get $G(R) = \|F(R)\|_F^2$, steps 4 and 5 which give us an optimum value of α can be too costly, so we use a modified algorithm.

Algorithm II

1. Given an initial approximation R_0, evaluate $G(R_0) = \|F(R_0)\|_F^2$ and $\nabla G(R_0)$.

2. Make $\nabla G(R_0)$ a unit vector.

3. Choose α such that $G(R_0) \geq G(R_0-\alpha\nabla G(R_0))$.

4. Return to 1 with R_0 replaced by $R=R_0-\alpha\nabla G(R_0)$.

The steepest descent method generally converges only linearly to the solution, but its convergence is global in nature. As a consequence, we use these steepest descent type algorithms to find sufficiently accurate initial guesses for SEM and NEM.

6.2 Numerical examples

We present a few examples which need an accurate initial guess. Actually, we failed to get the solutions of these examples using the iterative methods without using algorithms of steepest descent type. To get a solution set, first we perform the matrix algorithm of steepest descent type with $R_0=I$, tolerance=10^{-1} and the maximum

number of iterations=5 and then we use iterative methods. Since the problems below are of control problems, we again use the Bartels-Stewart algorithm for LCM and NEM and the Golub-Nash-Van Loan algorithm for SEM.

Example 8 :Laub [15]

Consider higher order Riccati equations arising from position and velocity control for a string of high-speed vehicles. For a string of N vehicles, it is necessary to solve the Riccati equation

$$A_N^T R_N + R_N A_N - R_N B_N D_N^{-1} B_N^T R_N + C_N = 0$$

where all matrices are of order $n=2N-1$ and are given by

$$A_N = \begin{vmatrix} A_{11} & A_{12} & & & & \\ & A_{22} & A_{23} & & & \\ & & & & & \\ & & & A_{N-2,N-2} & A_{N-2,N-1} & \\ & & & & A_{N-1,N-1} & \begin{vmatrix} 0 \\ -1 \end{vmatrix} \\ & & & & & -1 \end{vmatrix}$$

where

$$A_{i,i} = \begin{vmatrix} -1 & 0 \\ 1 & 0 \end{vmatrix}, \quad A_{i,i+1} = \begin{vmatrix} 0 & 0 \\ -1 & 0 \end{vmatrix}$$

and

$$B_N D_N^{-1} B_N^T = diag(1, 0, 1, 0, \ldots, 0, 1),$$

$$C_N = diag(0, 10, 0, 10, \ldots, 10, 0).$$

For the cases of 5, 10 and 20 vehicles, we try both algorithms I and II of steepest descent type. In both cases, SEM and NEM give the same solution which is identical to the solution of SCM, and moreover this solution has the same accuracy as SCM. The

numerical results of interest are summarized in the following table.

Table 16

Numerical results I for example 8

N	SEM		NEM		SCM
	L	r	L	r	r
5	8	1.0-12	6	1.0-12	1.0-12
10	10	1.0-12	8	1.0-12	1.0-12
20	10	1.0-11	9	1.0-11	1.0-12

L:The number of iterations satisfying r

$$r = \|A^T R_L + R_L A - R_L B D^{-1} B^T R_L + C\|_F$$

We also check the CPU time of the matrix algorithms of steepest descent type. The numerical results are summarized in the following table.

Table 17

Numerical results II for example 8

N	Algorithm I	Algorithm II
5	0.37	0.15
10	3.12	1.25
20	26	10.2

Algorithm II save approximately 60 percent of the CPU time of algorithm I. To compare with SCM we check the CPU time of the iterative methods and add to CPU time of algorithm II. The total CPU time of each method is summarized in the following table.

Table 18

Numerical results III for example 8

N	SEM	NEM	SCM
5	0.61	0.56	0.28
10	5.60	5.05	2.0
20	44.6	43.2	13.9

Example 9 :Laub [14]

This example involves circulant matrices. Consider

$$A^T R + RA - RBR + C = 0$$

where all matrices are of order n=64 and are given by

$$A = \begin{pmatrix} -2 & 1 & 0 & & & 0 & 1 \\ 1 & -2 & 1 & 0 & & & 0 \\ 0 & 1 & \ddots & \ddots & & & \\ & 0 & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & & 0 \\ 0 & & & & \ddots & & 1 \\ 1 & 0 & & & 0 & 1 & -2 \end{pmatrix}$$

and $B = I$, $C = I$. The matrices A, B and C are all circulant so the Riccati solution $R \in \mathbf{R}^{n \times n}$ is known to be circulant of the form

$$R = \begin{pmatrix} r_0 & r_{n-1} & \cdots & r_1 \\ r_1 & r_0 & \cdots & r_2 \\ & & & \\ & & & \\ r_{n-1} & r_{n-2} & \cdots & r_0 \end{pmatrix}$$

We perform both algorithms I and II of steepest descent type. In both cases, SEM and NEM give the same circulant solution which is identical to the solution of SCM and moreover the solution for both methods is as accurate as for SCM. As in example 8, we

compute the residual and check the CPU time for both algorithms of steepest descent type. The numerical results are summarized in the following tables.

Table 19

Numerical results I for example 9

SEM		NEM		SCM
N	r	N	r	r
5	1.0-11	4	1.0-12	1.0-12

N: The number of iterations satisfying r

$$r = \|A^T R_N + R_N A - R_N B R_N + C\|_F$$

Table 20

Numerical results II for example 9

Algorithm I	Algorithm II
113.8	44.5

Using table 20, we also check CPU time for SEM and NEM. SEM and NEM required approximately 108s of CPU time and 101s of CPU time respectively. SCM required approximately 57s of CPU time.

CHAPTER 7

CONCLUSIONS

In this thesis, we have implemented numerical methods for solving algebraic Riccati equations. The numerical results in the previous chapters indicate that SEM and NEM are comparable to KIM and SCM. In fact, with a modified matrix algorithm of steepest descent type which is used to find a sufficiently accurate starting value, SEM and NEM perform as well as SCM for solving Riccati equations of large order.

If we want to get very accurate solutions of Riccati equations arising from singular perturbation problems then both iterative methods are faster than KIM except for the case in which ϵ is very small. The execution times of SEM and NEM are greater than SCM. However, we expect more careful implementations of both methods to be much faster than the current versions.

The new iterative method, namely SEM, works as well as NEM. Solving nonsymmetric Riccati equations, SEM is considerably faster than NEM at each iteration. However, in the symmetric case, NEM is at least as fast as SEM provided a symmetric initial value is known.

We know that a modified matrix algorithm of steepest descent type (algorithm II) is relatively expensive, although this method in theory always converges. A way to improve the efficiency of iterative methods is to find a reliable and efficient method to give a good initial approximation. This fact requires further research.

As a final consideration and research, we have also considered the generalized Riccati equation

$$A_{22}R - LA_{11} = -A_{21} + LA_{12}R,$$

$$B_{22}R - LB_{11} = -B_{21} + LB_{12}R$$

corresponding to the generalized eigenproblem

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} - \lambda \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

where $A_{11}, B_{11} \in \mathbf{R}^{n\times n}$, $A_{22}, B_{22} \in \mathbf{R}^{m\times m}$ and $R, L \in \mathbf{R}^{m\times n}$.

As for the standard Riccati equation, LCM, NEM and SEM can be used (see [4] and [20]). These three iterative methods require the solution of the equation of same type, say, the generalized Sylvester equation:

$$A_1R - LB_1 = C_1, \tag{7.1}$$

$$A_2R - LB_2 = C_2$$

where $A_1, A_2 \in \mathbf{R}^{m\times m}$, $B_1, B_2 \in \mathbf{R}^{n\times n}$ and $R, L, C_1, C_2 \in \mathbf{R}^{m\times n}$.

In order to solve the generalized Sylvester equation, We also implemented the new algorithm called the generalized Bartels-Stewart algorithm (in short, GBS). The main idea of GBS is based on the equivalence between the problems (7.1) and

$$(Q_1^TA_1Z_1)(Z_1^TRZ_2) - (Q_1^TLQ_2)(Q_2^TB_1Z_2) = Q_1^TC_1Z_2, \tag{7.2}$$

$$(Q_1^TA_2Z_1)(Z_1^TRZ_2) - (Q_1^TLQ_2)(Q_2^TB_2Z_2) = Q_1^TC_2Z_2$$

where (Q_1, Z_1) and (Q_2, Z_2) are orthogonal matrices which are found from the QZ algorithm for the two pencils, $A_1 - \lambda A_2$ and $B_1 - \lambda B_2$ respectively. Now A_1 and B_1 are reduced to upper real Schur forms $Q_1^TA_1Z_1$ and $Q_2^TB_1Z_2$, respectively and also A_2 and B_2 are transformed into upper triangular forms $Q_1^TA_2Z_1$ and $Q_2^TB_2Z_2$, respectively. Thus we can solve this transformed systems (7.2) by modifying the Bartels-Stewart algorithm.

Using SEM and NEM with GBS, we solved a few discrete time algebraic Riccati equations in control problems ([15] and [21]) and obtained solutions which are identical

to those of SCM. However, further study is needed to prove the efficiency of SEM and NEM for solving generalized Riccati equations. All things considered, it seems that SEM and NEM have the potential to solve a large class of problems.

REFERENCES

[1] J.J.Dongarra, C.B.Moler, J.H.Wilkinson ; "Improving the accuracy of computed eigenvalues and eigenvectors", SIAM J. Num. Anal. vol 20, no 1, Feb. 1983, pp23-45

[2] G.W.Stewart ; "Error and perturbation bounds for subspaces associated with certain eigenvalues problems", SIAM Review, vol 15, no 4, Oct. 1973 , pp727-764

[3] F.Chatelin , "Simultaneous Newton's iteration for the eigenproblem", Computing, Suppl. 5, 1984, pp67-74

[4] J.W.Demmel ; "Three methods for refining estimates of invariant subspaces", submitted to Computing

[5] P.V.Kokotovic ; "Applications of singular perturbation techniques to control problems", SIAM Review, vol 26, no 4, Oct. 1984, pp501-550

[6] P.V.Kokotovic ; "A Riccati equation for block-diagonalization of ill-conditioned systems", IEEE Trans. Auto. Control, vol AC-20, Dec. 1975, pp812-814

[7] R.G.Phillips ; "Reduced order modelling and control of two-time-scale discrete systems", INT J. Control, vol 31, no 4, 1980, pp765-780

[8] R.G.Phillips ; "The equivalence of time-scale decomposition techniques used in the analysis and design of linear systems", INT J. Control, vol 37, no 6, 1983, pp1239-1257

[9] H.Kreiss, N.K.Nichols, D.L.Brown ; "Stiff two-point boundary value problems", SIAM J. Num. Anal. vol 23, no 2, Apr. 1986, pp325-368

[10] L.Dieci, M.R.Osborne, R.D.Russell; "A Riccati transformation method for solving BVPs I: theoretical aspects", submitted to SIAM J. Num. Anal.

[11] U.M.Ascher, R.M.M.Mattheij, R.D.Russell; "Numerical solution of Boundary value

problems for ordinary differential equation", Prentice-Hall, 1987

[12] R.R.Wilde, P.V.Kokotovic ; "A dichotomy in linear control theory", IEEE Trans. Auto. control, vol AC-17, Jun. 1972, pp382-383

[13] W.M.Wonham ; "On a matrix Riccati equation of stochastic control", SIAM J. Control, vol 6, no 4, 1968, pp681-697

[14] W.F.Arnold, A.J.Laub ; "Generalized eigenproblem algorithms and software for algebraic Riccati equations", Proc. IEEE, vol 72, no 12, Dec. 1984, pp1746-1754

[15] A.J.Laub ; "A Schur method for solving algebraic Riccati equations", IEEE Trans. Auto. Control, vol AC-24, no 6, Dec. 1979, pp913-921

[16] R.H.Bartels, G.W.Stewart ; "Solution of the matrix equation $AX+XB=C$", CACM, vol 15, no 9, Sep. 1972, pp820-826

[17] G.H.Golub, S.Nash, C.F.Van Loan ; "A Hessenberg-Schur method for the problem $AX+XB=C$", IEEE Trans. Auto. Control, vol AC-24, no 6, Dec. 1979, pp909-913

[18] R G.Phillips ; "A two-stage design of linear feedback controls", IEEE Trans. Auto. Control, vol AC-25, Dec. 1980, pp1220-1223

[19] J.J.Allemong, P.V.Kokotovic ; "Eigensensitivities in reduced order modelling", IEEE Trans. Auto. Control, vol AC-25, no 4, Aug. 1980, pp821-822

[20] L.Dieci, R.D.Russell ; "On the computation of invariant subspaces", submitted to Num. Math.

[21] T.Pappas, A.J.Laub, N.R.Sandell ; "On the numerical solution of the discrete-time algebraic Riccati equation", IEEE Trans Auto. Control, vol AC-25, no 4, Aug. 1980, pp631-641

[22] G.H.Golub, C.F.Van Loan ; "Matrix computations", The Johns Hopkins University Press, 1983

[23] R.L.Burden, J.D.Faires ; "Numerical analysis (third edition)", PWS Publishers, 1985