NUMERICAL ANALYSIS OF THE LYAPUNOV EQUATION

WITH

APPLICATION TO INTERCONNECTED POWER SYSTEMS

by

Thomas Mac Athay

SB, Michigan Technological University
(1974)

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

and

ELECTRICAL ENGINEER

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1976

Signature of Author Thomas M. Athay
Department of Electrical Engineering and Computer Science, May 14, 1976

Certified by Michael Athans
Thesis Supervisor

Accepted by .....
Chairman, Departmental Committee on Graduate Students

NUMERICAL ANALYSIS OF THE LYAPUNOV EQUATION

WITH

APPLICATION TO INTERCONNECTED POWER SYSTEMS

by

Thomas Mac Athay

Submitted to the Department of Electrical Engineering and Computer Science on May 14, 1976, in partial fulfillment of the requirements for the degree of Master of Science and Electrical Engineer at the Massachusetts Institute of Technology.

ABSTRACT

The Lyapunov equation is fundamental to control theory. A number of numerical solution methods are compared, with special emphasis placed on applicability to large scale system matrices of a general sparse structure. An iterative decoupling algorithm is developed to exploit this special form, and a computer program that realizes this method is reported. An introduction to linear error analysis is included, and a number of results are developed and extended to several Lyapunov equation solution techniques.

Thesis Supervisor: Nils R. Sandell, Jr.

Title: Assistant Professor of Electrical Engineering and Computer Science.

ACKNOWLEDGMENT

I would like to express my gratitude to Professor Nils Sandell, Jr., for supervising this thesis. He stimulated my original interest in the subject and provided valuable intellectual influence and moral support throughout the course of this study.

My association with the Electronic Systems Laboratory and resulting interaction with other members, particularly fellow graduate students, has been a valuable and pleasant experience. I am also grateful for the financial support I received during this period.

Final thanks go to my wife and partner, Roberta Ann, mostly for her love, but also for typing this thesis.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 USE OF LYAPUNOV EQUATION

Modern control theory has in the past two decades experienced a rapid development that is both exciting and significant. Its engineering application has understandably been a slower process, although recently it appears that the scope and diversity of application efforts are increasing. Important and rapidly changing limits to real world applications are computational requirements that arise in both the analysis and design of control systems. Because the theory has much to offer, efforts to understand and perhaps ease computational requirements are well motivated.

The Lyapunov equation arises in many aspects of both the analysis and design of linear control systems. Its solution is important in stability analysis of linear continuous systems [25], in pole assignment [17], when evaluating quadratic integrals which are often used as cost functions in optimal control [19, 25], and when evaluating covariance matrices in filtering and estimation for continuous systems. In addition, the algebraic Riccati equation, which occurs in some important filtering and optimal control problems, can be solved iteratively where each iteration is the solution of a Lyapunov equation [9, 11]. Another situation where the Lyapunov equation arises is in the design of decentralized control systems. Current research in large scale systems and decentralized control in the Electronic Systems

Laboratory is being directed toward physical systems that, although of large dimension, have sparse system matrices with particular structural forms. Examples of such research are the decentralized control of a freeway traffic corridor [12] and of large scale interconnected power systems [14]. In these studies the Lyapunov equation plays an important role, both in analysis and in design.

1.2 SUMMARY OF THESIS

There are many solution methods of the Lyapunov equation. In Chapter Two a number of them are introduced and compared, with special emphasis being placed on two special criteria. These are a method's potential for exploiting a general, sparse system matrix form and an algorithm's efficiency for re-solution. Basically, three properties of an algorithm are used to quantify the analysis: computational speed, storage requirements, and accuracy. The first two are included in Chapter Two, while the latter is covered in Chapter Three. In the final section of Chapter Two an iterative decoupling algorithm is developed that is specifically designed to meet the two criteria of special interest in this thesis.

Chapter Three is concerned with analyzing the error properties of several important solution methods. In order to do this, a general introduction to the method of backward error analysis is presented. This type of analysis is basically comprised of two steps, one of which involves the concept of numerical conditioning while the other

requires obtaining pertubation bounds that represent the effects of round-off error in the computer. Each of these steps are discussed in a separate section. The final section of the chapter is primarily a summary of the results obtained and final remarks regarding the comparison of the solution methods of interest.

Chapter Four contains a description of a computer program that was written to implement the iterative decoupling algorithm mentioned previously. The results of several small examples are briefly presented. In the next section, an unsuccessful application of the iterative decoupling algorithm to a power system example is reported. Finally, some conclusions and suggestions for future research are presented.

In addition, two appendices are included that are referenced at appropriate points in the main text. One is a summary of some pertinent facts of linear algebra, while the other briefly outlines some elementary tools of error analysis that are used in the analyses of Chapter Three.

CHAPTER II

SOLUTION METHODS

2.1 TWO SPECIAL CRITERIA

The Lyapunov equation is the steady state solution, $P(\infty)$, to the linear matrix equation

$$\frac{d}{dt} P(t) = A^T P(t) + P(t)A + Q \qquad A, P, Q \ n\times n \qquad (2.1.1)$$

If the eigenvalues of A are such that $\text{re}\ \lambda_j < 0$, $j = 1, 2, \ldots, n$, then the steady state solution $P(\infty) = P$ of

$$0 = A^T P + PA + Q \qquad (2.1.2)$$

exists and is unique, and is given by the convergent integral

$$P = \int_0^\infty e^{A^T t} Q e^{At}\, dt \qquad (2.1.3)$$

Furthermore, if Q is symmetric and positive definite, then P will also be symmetric and positive definite. It is occasionally convenient to represent equation (2.1.2) in the form

$$L_A : R^{n\times n} \to R^{n\times n} \qquad P \to A^T P + PA$$

A great deal of attention has been given to the numerical solution of the Lyapunov equation. A useful classification of the variety of solution techniques are the groupings of direct, transformation, and iterative methods. The purpose of this chapter is to summarize those methods that are (at least partly) favorable numerically, and to especially consider their application to sparse equations that must be solved many times. The following chapter will analyze in greater detail some of the algorithms introduced in this chapter. In particular, accuracy is not considered in this chapter.

The general sparse structure considered is system matrices of the form

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{1N} \\ A_{21} & A_{22} & A_{23} & A_{2N} \\ A_{31} & A_{32} & A_{33} & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \qquad \begin{matrix} A_{ii}\ n_i \times n_i \\ A_{ij}\ n_i \times n_j \end{matrix}$$

In the modelling of some physical systems of current interest [12, 14] the diagonal blocks represent physical subsystems and may or may not be sparse. In the dynamic power system problem, for instance, the diagonal blocks model the incremental dynamics of a primemover and electrical generator pair and associated mechanical and electrical

(excitation) controls, while the off-diagonal blocks represent the coupling of machine angle and voltage state variables through transmission network. For even a moderate number of interconnected subsystems (say N = 5 to 10) a solution method that efficiently exploits this sparse structure would be valuable.

The other special criteria that is important is a method's economy in solving $L_A(P) = -Q$ many times with the same A matrix but different Q matricies. This is an important consideration in a number of applications. A particular example is an approach, outlined by Sandell and Athans [14], to the design of decentralized control systems with fixed communication structures. In this approach, the optimal (infinite time, quadratic cost criteria) design is the solution to the constrained parameter optimization problem

$$\text{Min}\{\text{tr}Q(z)P\} \tag{2.1.4}$$

$$\text{subject to} \quad A(z)P + PA^T(z) = -R(z) \tag{2.1.5}$$

where $z^T = (z_1, z_2, \ldots, z_p)$ is the vector of parameters that characterize the design. The problem is to search for z* such that $\text{tr}Q(z^*)P(z^*) \leq \text{tr}Q(z)P(z)$. The point here is that the gradient of (2.1.4) with respect to z at $z = z^k$ (k^{th} step in search) is evaluated by resolving a Lyapunov equation p times with the same A matrix but different driving terms, i.e., with $L_A(P) = -Q$, different Q matrices.

2.2 DIRECT SOLUTION METHODS

The matrix equation

$$A^T P + PA = -Q \tag{2.2.1}$$

or $L_A(P) = -Q$, is an equation in $R^{n\times n}$. It can be conveniently rewritten using Kronecker product notation [25] as an equation in R^{n^2}. Let q and p be vectors that correspond to the n^2 elements of Q and P, taken by rows, respectively. Then (2.2.1) becomes

$$(A^T \times I + I \times A^T)p = -q \tag{2.2.2}$$

or $$K_A p = -q \qquad K_A: \ R^{n^2} \to R^{n^2} \tag{2.2.3}$$

Bellman [19] discusses the basic properties of the Kronecker product, a number of which are also contained in section 3.2. Equation (2.2.3) represents n^2 equations in n^2 unknowns, but in most applications $Q = Q^T$, therefore $P = P^T$ and we can rewrite equation (2.2.3) as

$$Cp = -q \tag{2.2.4}$$

where $$C \text{ is } \frac{n(n+1)}{2} \times \frac{n(n+1)}{2}$$

Two simple algorithms for forming the matrix C are presented by Chen and Shieh[7] and Bingulac [8]. Some easy subscript arithmetic yields

a simpler algorithm, however. Let $s = (2n-i)(i-1)/2 + j$. The s^{th} row in (2.2.4), using (2.2.1) is

$$\sum_{k=1}^{n} (a_{ki}p_{kj} + p_{ik}a_{kj}) = -q_s$$

The elements a_{ki} and a_{kj} belong in the r^{th} column of C, with r given by

$$p_{kj}: \quad r = \begin{cases} (2n-k)(k-1)/2 + j & k<j \\ (2n-j)(j-1)/2 + k & k \geq j \end{cases}$$

$$p_{ik}: \quad r = \begin{cases} (2n-i)(i-1)/2 + k & k \geq j \\ (2n-k)(k-1)/2 + i & k<i \end{cases}$$

Once C is formed, any standard algorithm for solving linear equations can be used. Because one often wishes to resolve (2.2.3) with different q vectors, LU decomposition of C is an efficient approach [21]. With this approach, ignoring sparsity, solving equation (2.2.4) for p requires operations (multiplications and divisions) of the order of $n^6/24$, where A is $n \times n$. Once the LU decomposition of C is accomplished, however, computing p given q requires operations of the order of $n^4/4$. The memory requirement is approximately $n^4/4$ words.

The above operation counts are pessimistic. Even if A is full (n^2 elements), C will have a known sparse structure and $n^2 + [n(n-1)/2][2n-1]$ elements. Let α_c be the number of elements in C and define a sparsity ratio β_c: $\beta_c = \alpha_c/n_c^2$, $0 \leq \beta_c \leq 1$. Then

for a given A(n×n) with a sparsity ratio β_A, $\beta_C \doteq 4\beta_A/n$. This strongly suggests that sparse matrix techniques should be considered when using the direct method. Experience in using a sparse matrix package that includes an optimal ordering algorithm for minimizing the fill-in of the triangular factors of C has resulted in an operation count for the direct method of

$$ops = {}^{1}/_{3}\beta_{Lu}^{2} n_c^3 + {}^{3}/_{2}\beta_{Lu} n_c^2 \doteq 6\beta_A^2 n^4 + {}^{9}/_{2}\beta_A n^3$$

Resolving (2.2.4) for new q requires approximately $3\beta_A n^3$ operations. Although significantly faster, a sparse direct method algorithm is considerably more complex to code. An advantage of either direct method is that a solution can be iteratively refined, with each iteration similar to resolving for new q, to a specified accuracy if the Kronecker matrix is at least moderately well conditioned. This issue will be covered in more detail in the following chapter.

An idea that is standard in the solution of large, sparse sets of linear equations and that exploits the same general sparse structure of interest was studied as a potential solution method for the Lyapunov equation. Again, consider solving Ax = b for x when A has the form:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ A_{N1} & \cdots & & A_{NN} \end{bmatrix}$$

Each subsystem (A_{ii}) is a stability matrix of dimension n. Let the total number of interconnections between the subsystems equal K. Then we can split A into two parts:

$$A = B + UV^T \qquad \text{where } B = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{NN} \end{bmatrix}$$

$U\ (nN \times K)$, $V\ (nN \times K)$

then [28] $$A^{-1} = B^{-1} - Y \Lambda Z^T \qquad (2.2.5)$$

where $Y = B^{-1}U$

$Z = V^T B^{-1}$

$\Lambda = (I + V^T B^{-1} U)^{-1}$

Note that Λ is (K x K) matrix; the dimension of Λ will depend on the geometry of the interconnections, but $\dim\Lambda \leq K$. This idea can be reformulated into an algorithm that involves an LU decomposition of B (i.e., the subsystems individually) and of Λ, and several steps of forward and back substitution. The higher order terms in the operation count are

$$Nn^3/3 + kNn^2 + k^3/3.$$

Although this count is reasonable for the solution of Ax = b, when the algorithm is extended to the Lyapunov equation, the corresponding operation count (for the definitions of N, n, and k) is roughly

$$N^2n^6/24 + kN^3(n^4/4 + 4K^2n^3)$$

which is considerably worse than the sparsity-exploiting direct method.

2.3 TRANSFORMATION SOLUTION METHODS

Transformation solution methods of the equation

$$A^TP + PA = -Q \tag{2.3.1}$$

generally involve introducing a similarity transformation on the state space such that the new system matrix $\tilde{A} = N^{-1}AN$ is in a canonical form and the transformed equation

$$\tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} = -\tilde{Q} \tag{2.3.2}$$

is more easily solvable. This section will consider two solution methods that utilize the companion and Schur canonical forms.

The use of the companion form actually characterizes a number of different solution methods. A number of methods view the problem in the frequency domain. The solution to equation (2.3.1) represents

the variance of the system $x(t) = A^T x(t) + W(t)$ in steady state, where $W(t)$ is stationary, zero mean white noise with spectral density matrix Q. The variance is also given by an integral of the form

$$P = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} H(s)H(-s)ds \qquad (2.3.3)$$

where $H(s)$ is the system transfer function matrix. Astrom gives a very compact solution for the variance of the output of a single input, single output system using results from spectral factorization theory [32]. Hagander has made the extension to multivariable systems [1]. Others have implicitly used the companion form to solve equation (2.3.1), e.g., Muller [3].

Molinari [4] has developed an algorithm that uses the companion form explicitly. Consider:

$$TAT^{-1} = C_A, \; S = T^{-T}PT^{-1}, \; R = -T^{-T}QT^{-1}$$

$$\det [sI - A] = S^n + a_n S^{n-1} + a_{n-1} + S^{n-2} + \ldots . \; a_2 S + a_1$$

$$\text{Then } A^T P + PA = -Q \text{ becomes } C_A{}^T S + SC_A = R \qquad (2.3.3)$$

C_A is of the form:

$$\begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & 1 \\ -a_1 & -a_2 & -a_3 & \cdots & & -a_n \end{bmatrix}$$

Letting $C_A^T S = U$ and $SC_A = V$, equation (2.3.3) becomes $U + V = R$, where

$$u_{ij} = -a_i s_{nj} \qquad i=1 \quad j=1, 2, \ldots, n$$

$$u_{ij} = s_{i-1,\, j} - a_i s_{nj} \qquad i=2, 3, \ldots, n \quad j=1, 2, \ldots, n$$

$$v_{ij} = -a_i s_{in} \qquad i=1 \quad i=1, 2, \ldots, n$$

$$v_{ij} = s_{i,\, j-1} - a_j s_{in} \qquad j=2, 3, \ldots, n \quad i=1, 2, \ldots, n$$

This set of equations can be rewritten by forming a new equation that results from defining an alternating summation along a diagonal $i + j =$ constant. That is

$$\sum_{j=\sigma}^{\Theta} (-1)^{j+1} [u_{jk} + v_{jk}] = \sum_{j=\sigma}^{\Theta} (-1)^{j+1} r_{jk} \qquad i=1, 2, \ldots (2n-1)$$

$$\sigma = \begin{cases} 1 & i \leq n \\ i+1-n & i>n \end{cases} \qquad \Theta = \begin{cases} i & i \leq n \\ n & i>n \end{cases} \qquad (2.3.4)$$

For i even in (2.3.4) the equation reduces to the zero identity, and for i odd all terms cancel except those in S_{ni} (i = 1, 2, ..., n). So equation (2.3.4) reduces to a set of n equations for the last row of S. These equations can be written in an interesting way. Let each sum of the right hand side of equation (2.3.4) be h_i (i = 1, 2, ..., 2n-1; $h_i = 0$ for i even). Define $2g_i = h_{2i-1}$ (i = 1, 2, ..., n). Then equation (2.3.4) becomes Hx = g, where H is the Hurwitz matrix, i.e.,

$$\begin{bmatrix} a_1 & & & & & \\ a_3 & a_2 & a_1 & & & \\ a_5 & a_4 & a_3 & a_2 & & \\ & & \ddots & \ddots & & \\ 0 & 0 & a_{n+1} & a_n & a_{n-1} & a_{n-2} \\ 0 & 0 & 0 & & 1 & a_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ \cdot \\ g_n \end{bmatrix} \qquad (2.3.5)$$

$$S_{ni} = (-1)^i X_i \qquad i = 1, 2, \ldots, n$$

Once the last row of S is obtained in this way, the remaining rows are found from the simple recursion

$$s_r(j-1) = r_r(j) + a_j y_r(n) - C_a^T y_r(j) \qquad (2.3.6)$$

$$j = n, n-1, \ldots, 2$$

Molinari gives a total operation count of $5n^3$, with storage requirements of the order of $4n^2 + 4n$. Aspects of this solution approach will be analyzed in the following chapter; in general, however, this algorithm is superior in terms of speed and storage requirements. Applying this algorithm directly when the system matrix A has the general sparse structure of interest (2.1) may be difficult, however. First, the companion form only exists when A is non-derogatory (discussed in Appendix A). Some other difficulties, which are numerical rather than theoretical, will be discussed in the next chapter. Roughly 3/5 of the total operation counts involve solving $R = T^{-T}QT^{-1}$ for R and $P = T^{T}ST$ for P. In addition, although T will generally reflect the sparseness of A and be sparse, T^{-1} tends to become full. To resolve for new Q matrices with the Molinari algorithm requires $\sim 3n^3$ operations; only the transformation of A is not redone. It is interesting to note that resolving equations (2.3.5) and (2.3.6) for S with different R matrices requires less than $2n^2$ operations.

Another transformation, that to a real Schur form, has been applied to the solution of the Lyapunov equation by Bartels and Stewart [2]. As in the companion form approach, a similarity transformation is applied to the system matrix A (i.e., on the state space), but in this case the transformation is orthogonal as well. The real Schur form immediately yields the eigenvalues of A, and the algorithms that affect the transformation are often used for this purpose. Some of the implications of this fact are important in error analysis, and hence will be considered in the next chapter.

Again, the equation

$$A^T P + PA = Q \tag{2.3.7}$$

is transformed to the equation

$$B^T X + XB = C \tag{2.3.8}$$

where $B = U^T A U$ $X = U^T P U$ $C = -C^T Q U$.

Note that $U^T U = U U^T = I$, which for real U is the definition of an orthogonal matrix. The matrix B (real Schur) is of the form:

$$B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ & B_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & B_{n-1,n} \\ & & & B_{nn} \end{bmatrix}$$

each B_{KK} $K = 1, \ldots, n$ is of, at most, dimension two.

By partitioning X and C conformally with B, equation (2.3.8) can be solved recursively using the equation

$$B_{ii}^T X_{ij} + X_{ij} B_{jj} = C_{ij} - \sum_{k=1}^{i-1} B_{ki}^T X_{kj} - \sum_{k=1}^{j-1} X_{ik} B_{kj} \tag{2.3.9}$$

$$i = 1, 2, \ldots, n \qquad j = 1, 2, \ldots, n$$

X_{ij} is, at most, a 2 x 2 matrix and can be easily found using the direct method (2.2).

The transformation to real Schur form involves two distinct steps. First A is transformed to the Hessenberg matrix H,

$$N^TAN = H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdot & \cdot & \cdot & h_{1n} \\ h_{21} & h_{22} & h_{23} & & & & \cdot \\ 0 & h_{32} & h_{33} & & & & \cdot \\ & 0 & h_{43} & h_{44} & & & \cdot \\ & & 0 & \cdot & \cdot & & \cdot \\ & & & & \cdot & \cdot & h_{n-1,n} \\ & & & 0 & h_{n,n-1} & & h_{nn} \end{bmatrix}$$

The Hessenberg form is important in a variety of numerical problems and a number of algorithms exist to form it [18, 22, 24]. Bartels and Stewart use Householder's method, which is an "exact" transformation with an operation count of the order of $2n^3$. The real Schur form matrix, B, is obtained from H using the QR algorithm, which is an iterative (and inexact) procedure where each iteration is an orthogonal similarity transformation. Because this step is in effect solving the eigenvalue problem, it can be a numerically difficult one. The operation count is of the order of $4\sigma n^3$, with σ the number of iterations before a specified tolerance is met. The rest of the solution is straight forward. Evaluating $C = -U^TQU$ and $P = UXU^T$ takes $3n^3$ operations (less

if A has known sparsity). The count on the recursion (2.3.9) depends on the number of complex eigenvalues of A; assuming one half are complex the count is n^3. An average total count using the real Schur form then, is $(4\sigma + 2)n^3 + (4n^3 + 15n^2)$ while storage requirements are of the order of $3n^2$. Of the total operation count, the second term in brackets is the amount required to resolve the problem for different Q matrices.

The transformation of A to real Schur form is clearly the key to this solution method. Because this step solves the eigenvalue problem, the method has little or no potential for directly exploiting the general sparse structure of interest. In addition, the solution of the eigenvalue problem is typically limited to maximum dimensions on the order of 150-200 for numerical reasons that will become more clear in the next chapter. Nonetheless, it is an interesting general solution technique, and of course particularly attractive if the eigenvalues of A are also sought.

2.4 ITERATIVE SOLUTION METHODS

The solution of $A^TP + PA = -Q$ for stable A can be written as

$$P = \int_0^\infty e^{A^T t} Q e^{At} dt \qquad (2.4.1)$$

This can be computed using the approximation

$$P(t+\Delta) = e^{A^T\Delta} P(t) e^{A\Delta} + \int_0^\Delta e^{A^T t} Q e^{At} dt \triangleq \Phi^T P(t) \Phi + \Gamma$$

then $$P = \sum_{k=0}^{\infty} (\Phi^T)^k \Gamma \Phi^k \qquad (2.4.2)$$

An accelerated version of equation (2.4.2) generates the series much faster, that is

$$P_{k+1} = (\Phi^T)^{2^k} P_k \Phi^{2^k} + P_k \qquad P_0 = \Gamma \qquad (2.4.3)$$

Kleinman characterizes this as a safe approach; the operation count is proportional to n^4, which must be repeated for new Q matrices. Davidson and Man [5] applied the Crank-Nicolson numerical integration method using a formula similar to equation (2.4.3), but with different approximations to Φ and Γ. They report operation counts of the order of $(3\sigma + 4)n^3$, with σ the number of iteration steps of equation (2.4.3). The iteration must be redone for new Q.

Another approximation for Φ results from introducing the bilinear transformation [16],

$$A \rightarrow \Phi = -(A + aI)(A - aI)^{-1}$$

$$Q \rightarrow \Gamma = (A^T - aI)^{-1} Q (A - aI)^{-1} \frac{1}{2a}$$

The solution can then be generated using the accelerated iteration equation (2.4.3). The operation count is similar to the Davidson and Man method. Smith [6] used this approach and the previous one for large ($n \leq 146$), lightly damped systems and found the bilinear

approach superior in terms of accuracy. Hagander [1] also favors this approach over the other iterative methods. However, the bilinear transformation algorithm needs to be redone for new Q matrices, and in terms of the other special requirement, exploitation of sparsity, it is not particularly favorable. This is because Φ will almost never be sparse even when A is.

2.5 ITERATIVE DECOUPLING METHOD

The algorithm discussed in the last section are essentially general purpose solution methods. An iterative decoupling approach naturally suggests itself when the general sparse system matrix structure is directly exploited. Consider a partitioned matrix of the form previously mentioned (i.e., off-diagonal submatrices very sparse)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ \\ A_{21} & A_{22} \end{bmatrix} \qquad \begin{matrix} A_{11}(n_1 \times n_1),\ A_{22}(n_2 \times n_2) \\ N = 2 \end{matrix}$$

The K_A matrix that is $A^T \times I + I \times A^T$, does not have the same general structure of A unless the P and Q matrices are similarly partitioned and K_A is formed in the following way:

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \qquad Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

let P_{ij} = vector of $n_i n_j$ elements of P_{ij} taken row-wise q_{ij} is similarly defined

and p = vector of N^2 P_{ij} sub-vectors that correspond to taking the P_{ij} blocks row-wise, $p \in R^{n^2}$

Then $A^T P + PA = -Q$ becomes

$$(A^T \times I)p + (I \times A^T)p = -q \tag{2.5.1}$$

The first term of equation (2.5.1) is of the form:

$$\begin{bmatrix} A_{11}^T \times I_1 & & A_{21}^T \times I_1 & \\ & A_{11}^T \times I_2 & & A_{21}^T \times I_2 \\ A_{12}^T \times I_1 & & A_{22}^T \times I_1 & \\ & A_{12}^T \times I_2 & & A_{22}^T \times I_2 \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{12} \\ P_{21} \\ P_{22} \end{bmatrix} \tag{2.5.2}$$

The second term of equation (2.5.1) is of the form:

$$\begin{bmatrix} I_1 \times A_{11}^T & I_1 \times A_{21}^T & & \\ I_1 \times A_{12}^T & I_1 \times A_{22}^T & & \\ & & I_2 \times A_{11}^T & I_2 \times A_{21}^T \\ & & I_2 \times A_{12}^T & I_2 \times A_{22}^T \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{12} \\ P_{21} \\ P_{22} \end{bmatrix} \tag{2.5.3}$$

Written in this way, the diagonal blocks of K_A include only the diagonal blocks of A, and the same correspondence holds for the off-diagonal blocks. Assuming that Q is symmetric, the equations corresponding to the off-diagonal blocks are redundant, i.e., $P_{ij} = P_{ji}^T$, $Q_{ij} = Q_{ji}^T$. Decomposing the A matrix, $A = A_0 + A_1$

$$A_0 = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{NN} \end{bmatrix} \quad A_{ii}\ n_i \times n_i$$

Then $L_A(P) = L_{A_0}(P) + L_{A_1}(P) = -Q$. The term $L_{A_0}(P)$ corresponds to the diagonal sub-blocks of equations (2.5.2, 2.5.3) and consists of $N(N+1)/2$ uncoupled equations for the block elements of P,

$$A_{ii}^T P_{ij} + P_{ij} A_{jj} = -Q_{ij} \qquad \begin{matrix} i=1, 2, \ldots, N \\ j=i, i+1, \ldots, N \end{matrix} \qquad (2.5.4)$$

The idea, then is to consider the sequence of solutions P^k, where

$$L_{A_0}(P^k) = -Q - L_{A_1}(P^{k-1}) \qquad (2.5.5)$$

Laub discusses this iteration for some general classes of linear operators [29]; for square matrices this method is well-known [23].

Consider solving $Ax = b$ $x \in R^n$, $b \in R^n$. Let $A = L + D + U$, where L is strictly lower, U is strictly upper and D is diagonal (this decomposition can be in an element or block-partitioned sense). The general iteration is

$$A_0 x^k = b - A_1 x^{k-1} \qquad k = 1, 2, \ldots \qquad (2.5.6)$$
$$x^0 = 0$$

If equation (2.5.6) converges, $\lim x^k = x = (A_0 + A_1)^{-1} b$ and assuming A^{-1} and A_0^{-1} exist,

$$(A_0 + A_1)x = b \qquad x = (A_0 + A_1)^{-1} B$$

$$= (I + A_0^{-1} A_1)^{-1} A_0^{-1} b.$$

If $\rho(A_0^{-1} A_1) < 1$, that is $A_0^{-1} A_1$ is a contraction, then

$$x = (I - A_0^{-1} A_1 + (A_0^{-1} A_1)^2 - \ldots) A_0^{-1} b$$

or
$$A_0 x = \sum_{i=0}^{\infty} (-1)^i [A_0^{-1} A_1]^i b.$$

Define

$$x^k = A_0^{-1} \sum_{i=0}^{k-1} [A_0^{-1} A_1]^i b = A_0^{-1} [b - A_1 x^{k-1}]$$

which is equation (2.5.6). The contraction condition is both necessary and sufficient for global convergence, which is of order one. With $A = L + D + U$, the following associations are usually made:

i) $A_0 = D$ $\quad A_1 = L + U$ Jacobi Iteration

ii) $A_0 = L + D$ $\quad A_1 = U$ Gauss-Seidel Iteration

iii) $A_0 = I + wL$ $\quad A_1 = wU - (1-w)I$ $\quad D = I$, "b" = wb

successive overrelaxation ($1 \leq w \leq 2$)

Equation (2.5.5) is of the form of the Jacobi iteration. It is interesting that another conceptual approach, suggested by the form of the A matrix, leads to the same Jacobi iteration. Consider a power series approach, that is:

$$A = \begin{bmatrix} A_{11} & \varepsilon A_{12} & \varepsilon A_{13} & \cdot & \cdot & \varepsilon A_{1N} \\ \varepsilon A_{21} & A_{22} & \varepsilon A_{23} & \cdot & \cdot & \varepsilon A_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \varepsilon A_{N1} & \varepsilon A_{N2} & \cdot & \cdot & \cdot & A_{NN} \end{bmatrix} = A_0 + \varepsilon A_1$$

with $\dfrac{\partial^i}{\partial p^i}[L_A(P)] \equiv p^i$ then $P(\varepsilon) = \sum_{i=0}^{\infty} \dfrac{\varepsilon^i}{i!} p^i(0)$. (2.5.7)

Equation (2.5.7) is the expansion of P about $\varepsilon = 0$, and the desired solution is evaluated with $\varepsilon = 1$. At $\varepsilon = 1$, however, equation (2.5.7)

is simply a rewriting of equation (2.5.5). The power series approach, therefore, has the same conditions for convergence, namely that A_0 is stable and $A_0^{-1}A_1$ is a contraction.

In order to obtain an approximate operation count using the Jacobi iterative method, assume that there are N subsystems, each of dimension n, and that an off-diagonal block of A has at most γ elements. Let the maximum number of elements in an A_{ii} be α, so $\beta = \alpha/n^2$. If the reduced order equations are solved using a transformation method, the total operation count is of the order of

$$10Nn^3 + \sigma[N^2n^3 + \gamma(N^3n + \frac{N^2n^2}{2})]$$

where σ is the number of iterations. At each iteration, cubic terms result from going in and out of the transform frame, raising the intriguing possibility of reduction of the operation count by a power of n for problems that can be set up to make this transformation-retransformation unnecessary. Storage requirements are approximately $N^2n^2 + 7/2\ Nn^2$. If a sparse direct method is used, the count is approximately

$$6\beta^2N^2n^4 + \sigma[N^3n\gamma + 3\beta n^3N^2].$$

The storage requirements are roughly $\beta(6N^2n^3 + 2Nn^2) + 5n^2N^2$.

CHAPTER III

ERROR ANALYSIS

3.1 INTRODUCTION TO ERROR ANALYSIS

Any solution method for the Lyapunov equation can be analyzed in terms of computational speed, storage requirements, and accuracy. The first two are relatively straight forward to estimate and have been covered in the previous chapter. This chapter considers the third issue, error analysis.

Whenever computations are performed on a digital computer, one fundamental problem is to assess the numerical significance of the solutions. Four basic sources of error can be identified for a typical computation in scientific or engineering work:

i) Modelling errors occur whenever mathematical models are used to represent physical processes.

ii) Measurement errors represent the difference between ideal and computed parameters of the mathematical model as well as whatever errors exist in data that is input to the computer.

iii) Truncation errors represent approximations made during the computation to functions that are not simple algebraic relationships. For example, if e^X is approximated by a finite series, then the neglected higher order terms are the truncation error.

iv) Rounding errors represent the affects of the finite word length of the computer.

By definition, the first two types of error are not a property of the algorithm to be analyzed and will not be considered in this chapter. In addition, truncation errors will not be considered, simply because most of the algorithms analyzed here consist of a finite number of steps and therefore do not involve such errors. Two exceptions are the iterative decoupling method and the QR algorithm, which are infinite (iterative) procedures terminated with some appropriate tolerance test. Although the resulting truncation error will not be considered explicitly, the nature of the analysis used will nonetheless yield results that are consistent with those of the "exact" algorithms and allow useful comparisons to be made. For the remainder of this chapter then, the term error will refer to rounding error only.

There are two fundamental approaches to error analysis, namely forward and backward. The forward approach attempts to carry errors made at each computational step forward and compare the final result with the ideal one. The backward approach views the computed result as the exact solution of a perturbed equation. The latter approach is more modern and usually easier to perform and interpret. Its development is primarily due to Wilkinson [18], a well-known authority in several areas of error analysis. Although he has apparently not studied the Lyapunov equation, this chapter relies heavily on his work [18, 24]. The backward approach actually involves two steps,

representing the inherent conditioning of the problem and obtaining bounds for the perturbations of the ideal equation. The formulation of these two steps is naturally dependent on the problem being analyzed, but a simple description illustrates the basic philosophy.

$$\left.\begin{matrix}\text{data}\\ \text{parameters}\end{matrix}\right\} \rightarrow \text{computation} \rightarrow \text{answer}$$

$$u_1, u_2, \ldots, u_m \qquad t \text{ digit approximation} \qquad x_1, x_2, \ldots, x_n$$

The conditioning of the problem is defined to be the sensitivity of relative errors in x to perturbations in the inputs u. If the sensitivity is high, then the problem is ill-conditioned. At each step in the computation, rounding errors are treated as effective perturbations of the inputs. In general, the conditioning depends on the parameters and the general problem being solved, but not on the specific algorithm used or t (computer word length). The effective perturbations, on the other hand, depend strongly on the specific algorithm, the parameters, t, and possibly x.

The remainder of this chapter is comprised of three sections. The first two consider the two steps of backwards analysis, conditioning and perturbation bounds, respectively. The third summarizes and compares the results obtained for the direct, transformation, and iterative decoupling Lyapunov equation solution methods.

3.2 CONDITIONING

In order to discuss the numerical conditioning of a linear set of equations, some basic tools of linear algebra are needed. The necessary ones are primarily basic and commonly known. Error analysis particularly relies on the use and manipulation of vector and matrix norms however, so a brief collection of definitions and relationships that will be important later are included here.

A convex set K, $K \varepsilon R^n(C^n)$, is a set of points such that $x, y \varepsilon K$, $0 \leq \lambda \leq 1 \rightarrow \lambda x + (1-\lambda)y \varepsilon K$. A convex body is a closed, bounded, convex set with interior points. K is an equilibrated convex body if

$$k \varepsilon K, \qquad |w| \leq 1 \rightarrow wk \varepsilon K.$$

Notice that in this case the origin is interior to K. The equilibrated convex body of particular interest here is the unit sphere S, i.e.,

$$S = \{x \varepsilon C^n \mid x^H x \leq 1\}.$$

Now let K be an equilibrated convex body. Then the norm of the vector $x \quad R^n(C^n)$ with respect to K is defined [22]

$$\|x\|_K = \inf(\nu \mid \nu \geq 0,\ x \varepsilon \nu K) \tag{3.2.1}$$

and the (least upper) bound of the matrix $A \varepsilon R^{n\times n}(C^{n\times n})$ with respect to K is

$$\text{lub}_K(A) = \inf(\alpha \mid \alpha \geq 0,\ AK \subset \alpha K). \tag{3.2.2}$$

These definitions satisfy the following properties:

i)	$\|x\| > 0$ if $x \neq 0$	v)	$\text{lub}(A) > 0$ if $A \neq 0$
ii)	$\|\alpha x\| = \|\alpha\| \, \|x\|$	vi)	$\text{lub}(\alpha A) = \|\alpha\| \text{lub}(A)$
iii)	$\|x+y\| \leq \|x\| + \|y\|$	vii)	$\text{lub}(A+B) \leq \text{lub}(A) + \text{lub}(B)$
iv)	$\|Ax\| \leq \|x\| \text{lub}(A)$	viii)	$\text{lub}(AB) \leq \text{lub}(A)\text{lub}(B)$

(3.2.3)

where the subscipt K has been omitted for notational simplicity only. For any matrix A, there is at least one $x \neq 0$ such that

$$\|Ax\| = \|x\| \text{lub}(A).$$

A matrix norm is usually defined as any real valued function of the elements of A such that properties v) - viii) are satisfied with $\|\cdot\|$ replacing $\text{lub}(\cdot)$. The most commonly used vector norms are given by

$$\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{1/p} \tag{3.2.4}$$

$$p = 1,\ 2,\ \infty$$

where $\|x\|_p = \max(x_i)$. Analogous matrix norms that are <u>subordinate</u> to a vector norm satisfy

$$\|A\| = \max \|Ax\|, \qquad \|x\| = 1$$

and are computed as

$$\|A\|_1 = \max_j \sum_i |a_{ij}|$$

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|$$

$$\|A\|_2 = (\max\lambda(A^H A))^{1/2}. \qquad \text{(spectral norm of A)}$$

Norms are said to be consistent or compatible if

$$\|Ax\| \leq \|A\| \|x\|.$$

Notice that subordinate norms must be consistent, but consistent norms are not necessarily subordinate. The useful euclidian norm $\|A\|_E$ is consistent with $\|X\|_2$, where

$$\|A\|_E = (\mathrm{tr}(A^H A))^{1/2} = (\sum_{ij}\sum |a_{ij}|^2)^{1/2}.$$

The eigenvalues of $A^H A$ are called the singular values of A, i.e., $\lambda(A^H A) = \sigma^2$. The eigenvalues of $A^H A$ are real and positive, so we can write

$$\sigma_1^2 \geq \sigma_2^2 \geq \ldots \sigma_n^2 > 0$$

from which

$$\|A\|_2 = \sigma_1.$$

Notice that if S is the unit sphere, then

$$\text{lub}_S(A) = \sigma_1 = \|A\|_2. \tag{3.2.5}$$

A (greatest lower) bound for A with respect to the equilibrated convex body K is defined by

$$\text{glb}_K(A) = \sup(\alpha \,|\, \alpha K \subset AK). \tag{3.2.6}$$

Now $\text{glb}_K(A) = 1/\text{lub}_K(A^{-1})$, and in particular for the unit sphere

$$\text{glbs}(A) = \alpha_n = 1/\|A^{-1}\|_2. \tag{3.2.7}$$

A few important relationships between the different matrix norms that are frequently useful in error analysis are summarized below. First, from

$$Ax = \lambda x$$

we see that

$$\|A\|\|X\| \geq \|AX\| = \|\lambda X\| = |\lambda|\|X\|,$$

so $$|\lambda| \leq \|A\| \text{ for any norm.}$$

Using this fact, a number of inequalities can be deduced, i.e.,

$$\|A\|_2^2 = \max\lambda(A^H A) \leq \|A^H A\|_E \leq \|A\|_E^2 = \mathrm{tr}(A^H A) = \sum\lambda(A^H A) \leq n^{1/2}\|A\|_2,$$

or $$\|A\|_2 \leq \|A\|_E \leq n^{1/2}\|A\|_2.$$

Also $$\|A\|_2^2 \leq \|A^H A\|_\infty \leq \|A^H\|_\infty \|A\|_\infty = \|A\|_1 \|A\|_\infty.$$

Denoting $|A|$ as the matrix whose elements are $|a_{ij}|$, notice that

$$\||A|\| = \|A\| \text{ for all norms except } \|\cdot\|_2, \text{ while}$$

$$\|I\| = 1 \quad \text{for } 1, 2, \infty \text{ norms}$$

$$= n^{1/2} \quad \text{for } \|\cdot\|_E.$$

For a linear set of equations the concept of numerical conditioning is expressed with some simple inequalities and quantified by a single number called the condition number. Consider the sensitivity of the solution of the set of equations

$$Ax = b \tag{3.2.8}$$

to variations in A and b. Let h and k be perturbations in x and b, respectively, where

$$A(x + h) = b + k, \tag{3.2.9}$$

or $h = A^{-1}k,$

so $$\|h\| = \|A^{-1}k\| \leq \|A^{-1}\|\|k\|. \tag{3.2.10}$$

From (3.2.9),

$$\|b\| \leq \|A\|\|x\|, \text{ or } \|x\| \geq \|b\|\|A\|^{-1},$$

and combining with (3.2.10) yields

$$\|h\|/\|x\| \leq \|A\|\|A^{-1}\|\|k\|/\|b\|. \tag{3.2.11}$$

Now let E be a perturbation of A and consider

$$(A + E)(x + h) = b \tag{3.2.12}$$

or $(A + E)h = -Ex.$

Now $(A + E) = A(I + A^{-1}E)$, so

$(A + E)^{-1}$ exists if $\lambda(A^{-1}E) < 1.$

In addition,

$$(I + X)^{-1} = I - X + X^2 - X^3 + \ldots \quad \text{if } \|X\| < 1, \ .$$

$$\|(I + X)^{-1}\| = \|I - X + \ldots\| \leq \|I\| + \|X\| + 1/2\|X^2\| + \ldots$$

$$= \frac{1}{1 - \|X\|} \ .$$

Assuming $\|A^{-1}E\| < 1$, and using the above inequality, equation (3.2.12) can be manipulated to yield

$$\frac{\|h\|}{\|X\|} \leq \frac{\|A\|\|A^{-1}\|\|E\|/\|A\|}{1-\|A\|\|A^{-1}\|\|E\|/\|A\|} \ . \qquad (3.2.13)$$

Now equations (3.2.11) and (3.2.13) are true for the spectral norm (a fortiori for the euclidian norm), and in each case the critical qualtity relating the perturbations is defined to be the spectral condition number,

$$k(A) = \|A\|_2\|A^{-1}\|_2 .$$

The relationship of (3.2.13) illustrates clearly the basic idea of backward error analysis applied to the linear matrix equation (3.2.8). The spectral condition number of A relates the sensitivity of the solution accuracy to the "size" of relative perturbations in A. It is a property of A and independent of solution methods, while the

matrix E, which represents the effects of round off errors, will depend explicitly on the specific algorithm used and computer word length. Note that the condition number of A is defined relative to the problem being solved. For instance, if X is the matrix whose columns are the eigenvectors of A (including pseudo eigenvectors if A has repeated roots) then the condition number of A with respect to the eigenvalue problem is defined to be $\|X\|_2\|X^{-1}\|_2$ [24]. In this thesis, conditioning will always mean conditioning with respect to the solution of sets of linear equations.

As an interesting application of K(A), suppose that Ax = b is solved in some way and no rounding errors occur. In this case, each element of A is correct to t digits, so $|e_{ij}| \leq 2^{-t}|a_{ij}|$, or

$$\|E\|_E \leq 2^{-t}\|A\|_E, \quad \|E\|_2 \leq n^{1/2}2^{-t}\|A\|_2,$$

and equation (3.2.13) yields

$$\frac{\|h\|_2}{\|x\|_2} = \frac{k(A)n^{1/2}2^{-t}}{1-k(A)n^{1/2}2^{-t}} . \tag{3.2.14}$$

This shows that unless $n^{1/2}k(A)2^{-t} \ll 1$ the computed solution will be numerically insignificant. A simple calculation illustrates the use of equation (3.2.14). Suppose that k(A) = 100, n = 100, and single precision is used on an IBM-370. In this case, the word length is

32 bits, of which 24 are given to the mantissa in floating point, i.e., t = 24. Then

$$(K(A)n^{1/2})2^{-t} \simeq (K(A)n^{1/2})10^{-t(.30)}$$

$$= (10^{3})10^{-7.2} = 10^{-4.2},$$

so we can expect only 4 or 5 significant figures in our solution!

A condition number for the Lyapunov equation can be similarly defined. The perturbed equation

$$(A + E)^{T}(P + H) + (P + H)(A + E) = -Q \tag{3.2.15}$$

or $$(K_A + K_E)(P_V + H_V) = -Q_V,$$

leads to

$$\frac{\|H_V\|}{\|P_V\|} \leq \frac{\|K_A\|\|K_A^{-1}\|\|K_E\|/\|K_A\|}{1-\|K_A\|\|K_A^{-1}\|\|K_E\|/\|K_A\|}, \tag{3.2.16}$$

and the condition number for the Lyapunov equation is

$$k(L_A) = \|K_A\|_2\|K_A^{-1}\|_2, \quad K_A = A^{T}\times I + I\times A^{T}$$

In the remainder of this section, two topics are covered. First, some properties of the condition number are presented and discussed. The simpler notation of Ax = b will be used, realizing that the

results apply to the Lyapunov equation analogously. Secondly, some properties of the Kronecker matrix are investigated in an attempt to obtain a useful relationship between its condition number and that of A.

The spectral condition number depends on the singular values of A, for combining equations (3.2.5) and (3.2.7) we have

$$k(A) = \|A\|_2\|A^{-1}\|_2 = \text{lubs}(a)/\text{glbs}(A) = \sigma_1/\sigma_n, \qquad (3.2.17)$$

where

$$\sigma_1^2 = \max\lambda(A^HA), \text{ and } \sigma_n^2 = \min\lambda(A^HA).$$

The computation of k(A) then, is a major task and for this reason other norm measures are often used in practice to bound k(A), usually the euclidian norm, i.e.,

$$k(A) = \|A\|_2\|A^{-1}\|_2 \leq \|A\|_E\|A^{-1}\|_E.$$

In this case the obvious difficulty is computing A^{-1}. Suppose, however, that an approximation of A^{-1} is obtained such that

$$I - AC = R \quad , \quad C \approx A^{-1} \ ,$$

or

$$A^{-1}(I - R) = C \ ,$$

$$A^{-1} = C(I - R)^{-1} \ ,$$

then

$$\|A^{-1}\| \leq \frac{\|C\|}{1-\|R\|} \qquad \text{if } \|R\| < 1. \tag{3.2.18}$$

This use of an approximate inverse is apparently a common way in practice to assess the accuracy of a computed solution x [17, 18, 21, 23, 33], but assuming that C is computed when x is, it does not provide an apriori estimate. Now recall the notation introduced with the iterative decoupling solution methods of the previous chapter. (The notation Ax = b is used here, but the results, again, are analogous to $L_A(P) = -Q$.) We had

$$A = A_0 + A_1 \quad , \quad A_0 \text{ (block) diagonal,}$$

and the condition for convergence of the methods was

$$P(A_0^{-1}A_1) < 1,$$

which is certainly true if $\|A_0^{-1}A_1\| < 1$. This is the same condition on R in (3.2.18), however, so

$$\|A^{-1}\| \leq \frac{\|A_0^{-1}\|}{1-\|A_0^{-1}A_1\|} \quad , \quad \|A_0^{-1}A_1\| < 1, \tag{3.2.19}$$

showing that if any of the iterative methods are used to solve Ax = b then (3.2.19), together with the easily computed $\|A\|_E$, provide a useful approximation to the condition number.

Intuitively one may be tempted to use the ratio $\lambda max(A)/\lambda min(A)$ as a measure of the conditioning of A with respect to Ax = b. This can be a very misleading approximation. Certainly

$$\|A\|_2 \geq |\lambda max(A)| = |\lambda_1| ,$$

$$\|A^{-1}\| \geq |\lambda max(A^{-1})| = 1/|\lambda min(A^{-1})| = 1/|\lambda_n|,$$

and
$$k(A) = \sigma_1/\sigma_n \geq |\lambda_1|/|\lambda_n|, \tag{3.2.20}$$

with equality being obtained only for matrices of special form, e.g., symmetric and anti-symmetric matrices. In order to illustrate why an ill-conditioned matrix may <u>not</u> have a small eigenvalue, consider the following:

normalize A such that $\|A\|_2 = 1$, and

let P be the unitary matrix (which always exists) such that $P^H AP = \text{diag}\ (\lambda_i) + T,$

$$= D + T, \text{ T strictly upper triangular.}$$

Now
$$P^H A^{-1} P = (D + T)^{-1} = (I - R + R^2 - \ldots - (-1)^n R^{n-1}) D^{-1},$$

where
$$R = D^{-1} T.$$

We have

$$\|T\|_2 \leq \|T\|_E \leq \|D + T\|_E = \|A\|_E \leq n^{1/2}\|A\|_2 = n^{1/2},$$

so $$\|R\|_2 \leq \|D^{-1}\|_2\|T\|_2 \leq n^{1/2}/|\lambda_n|,$$

and $$\|A^{-1}\|_2 \leq (1 + n^{1/2}/|\lambda_n| + (n^{1/2}/|\lambda_n|)^2 + \ldots + n^{1/2}/|\lambda_n|)^{n-1})1/|\lambda_n|$$

from which

$$|\lambda_n| \leq \frac{n^{1/2(n+1)}}{\|A^{-1}\|_2}^{1/n} \doteq \frac{n^{1/2}}{\|A^{-1}\|_2^{1/n}} \cdot \qquad (3.2.21)$$

This shows that for fixed n, as $\|A^{-1}\|_2$ approaches infinity, $|\lambda_n|$ approaches zero, but <u>very</u> slowly. Because of the direction of the inequality (3.2.20), we can only conclude $|\lambda_1|/|\lambda_n|$ large implies that ill-conditioning is to be expected.

Because the spectral condition number is an important quantity in backwards error analysis, it is unfortunate that it is difficult to compute. The use of an approximate inverse has been mentioned, but this too requires considerable computation. If an a posteriori estimate of a solution's accuracy is sufficient, the most practical procedure probably involves the use of iterative refinement, and this is illustrated in the next section. If a linear equation solution is a step in a more complicated computation, however, neither approach may be practical and an a priori estimate of accuracy would be very useful. For example, with the iterative decoupling solution method for the Lyapunov equation, a simple error analysis (Section 3.4) relates the final solution error to those that occur at each step, so

an a priori estimate of conditioning, combined with a perturbation bound, could be used to choose a convergence tolerance. Additional research into the nature of the condition number, however, has lead to the conclusion that a simple, easily computed approximation for it is unlikely to be found [22, 24, 35, 36, 37, 38, 39]. A few theoretical properties of the condition number are included here that illustrate the practical difficulty, but aid understanding.

First, Wehl [35] has proven a number of interesting inequalities that relate the singular values and eigenvalues of A. Let

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$$

$$\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2 > 0,$$

then the Wehl inequalities are

$$|\lambda_1||\lambda_2|\ldots|\lambda_k| \leq \sigma_1\sigma_2 \ldots \sigma_k \quad k=1, 2, \ldots, n-1,$$

$$|\lambda_1||\lambda_2|\ldots|\lambda_n| = \sigma_1\sigma_2 \ldots \sigma_k \qquad (3.2.22)$$

and

$$|\lambda_1|^s + |\lambda_2|^s + \ldots + |\lambda_k|^s \leq \sigma_1^s + \sigma_2^s + \ldots \sigma_k^s$$

$$s > 0, \text{ real.}$$

Manipulating (3.2.22) yields a bound for R(A)

$$K(A) \leq \sigma_1^n/\det(A) \leq \|A\|_E^n/\det(A). \tag{3.2.23}$$

This is probably the simplest bound not involving A^{-1} which has been found. A geometrical insight into the spectral condition number results by recognizing that A maps the unit sphere onto an ellipse whose axes lie along the eigenvectors of A^HA. Let S be the unit sphere, and y_k an eigenvector of A^HA. So

$$A^HAy_1 = \sigma_1^2 y_1 \qquad A^HAy_n = \sigma_n^2 y_n$$

while from the definitions of matrix bounds (3.2.2)

$$\max_{x\varepsilon S}\|Ax\| \equiv \|Ax_1^*\| = \sigma_1, \quad \min_{x\varepsilon S}\|Ax\| \equiv \|Ax_n^*\| = \sigma_n$$

then $$Ax_1^* = \sigma_1 y_1, \quad Ax_n^* = \sigma_n y_n. \tag{3.2.24}$$

Again, $K(A) = \sigma_1/\sigma_n$, which is the ratio of the lengths of the major and minor axes of the ellipse AS. Another geometrical representation of the condition number formalizes the intuitive notion that ill-conditioning is related to the distortion of the ellipse (notice that as $\det(A) \to 0$, the ellipse degenerates to a place in at least one dimension). If the angle θ is defined by

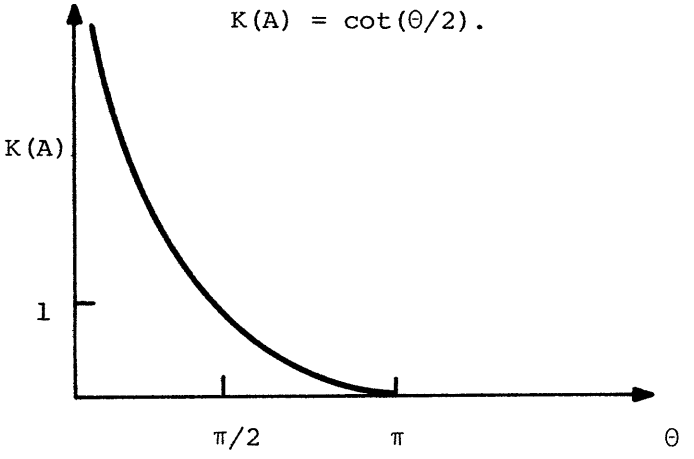
$$(k(A) - k^{-1}(A))/(k(A) + k^{-1}(A)) = \cos\theta \qquad (3.2.25)$$

then the inequality of Wielandt [22] is given by

$$\frac{|x^H A^H A y|^2}{(x^H A^H A x)(y^H A^H A y)} \leq \cos^2\theta \text{ , x, y any orthogonal pair of vectors} \qquad (3.2.26)$$

The geometrical interpretation is that θ is the minimal angle between Ax and Ay, for all orthogonal pairs x and y. Applying a standard trigonometric identity to (3.2.21), we obtain

$$K(A) = \cot(\theta/2). \qquad (3.2.27)$$



This discussion of the condition number is concluded with the following simple example. The mapping of the unit sphere is illustrated, along with the relationships between the singular values, eigenvalues, and norms of A.

$$A = \begin{bmatrix} -1 & a \\ 0 & -2 \end{bmatrix} \qquad K^2(A) = \frac{5+a^2+((a^2+1)(a^2+9))^{1/2}}{5+a^2-((a^2+1)(a^2+9))^{1/2}}$$

It is easy to verify that K(A) is an increasing function of a (for a > 0).

Let a = 3, so

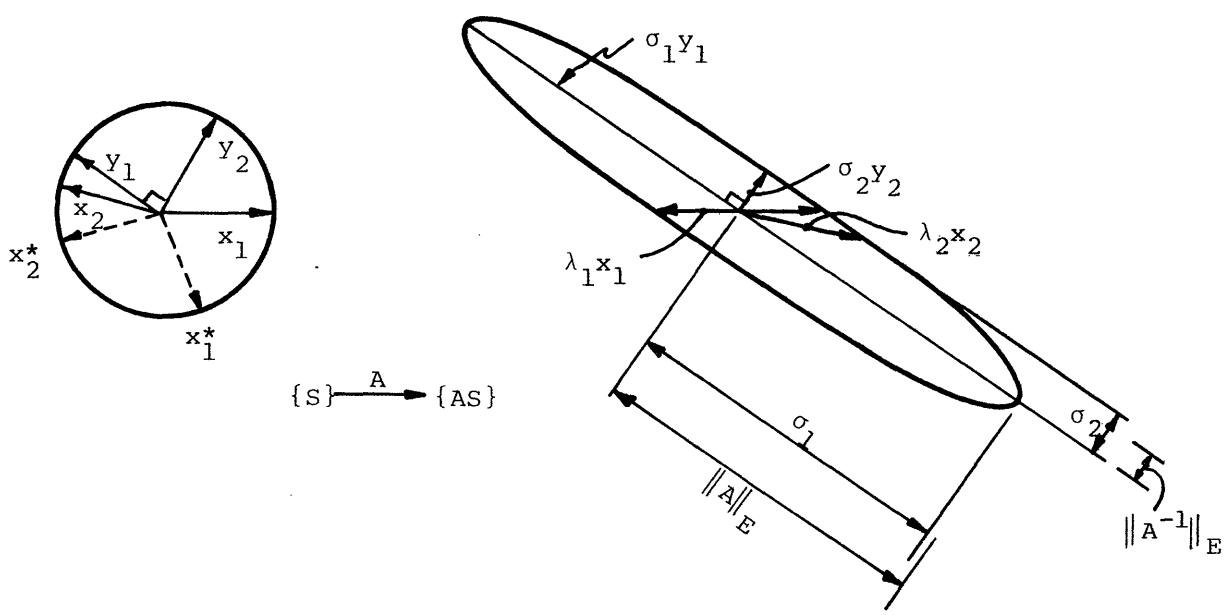
$$A^H A = \begin{bmatrix} 10 & -6 \\ -6 & 4 \end{bmatrix} \qquad \sigma_1^2 = (3.70)^2,\ y_1 = \begin{bmatrix} -.851 \\ .525 \end{bmatrix}$$

$$K(A) = \frac{3.70}{.54} = 6.85 \qquad \sigma_2^2 = (0.54)^2,\ y_2 = \begin{bmatrix} .525 \\ .851 \end{bmatrix} .$$

Similarly, the eigenvalues and eigenvectors of A are

$$\lambda_1 = -1 \quad x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = -2 \quad x_2 = \begin{bmatrix} -.950 \\ .316 \end{bmatrix}$$

and $\|A\|_E = 3.74$ $\|A^{-1}\|_E = 0.53$



Again, it is emphasized that although the proceeding discussion of conditioning was phrased in terms of the equation Ax = b, analogous remarks apply for the Lyapunov equation, whose condition number was defined in equation (3.2.16) to be

$$k(L_A) = \|K_A\|_2 \|K_A^{-1}\|_2 , \qquad K_A = A^T \times I + I \times A^T .$$

Because the Kronecker matrix K_A is $n^2 \times n^2$ however, the practical difficulties in either computing or approximating its condition number are even more severe than in the case of A itself which is n×n. For this reason the properties of the Kronecker matrix were investigated to see if its condition number is related in any simple way to that of A. The conclusion reached is that no such relationship exists, and the purpose of the remainder of this section is to briefly summarize why this is true.

First, the properties of the Kronecker matrix are naturally very dependent on those of the Kronecker product $A \times B$. If A is n×m and B is l×p, then the matrix $A \times B$ is (nl)×(mp) and partitioned into the (mn) blocks $(a_{ij}B)$. Many interesting properties of the Kronecker product are not developed here, but are given by Bellman [19] and Barnett and Storey [30]. The eigenvalues of $A \times B$ are important here, however, so consider

$$A \quad n\times n, \qquad Ax^i = \lambda_i x^i,$$

$$B \quad m\times m, \qquad By^j = \mu_j y^j,$$

then A × B has eigenvalues $\lambda_i\mu_j$ (i=1, 2, ..., n; j=1, 2, ..., m) and eigenvectors $z_{ij} \in R^{nm}$, where

$$z_{ij} = \begin{bmatrix} x_1^i y^j \\ x_2^i y^j \\ \cdot \\ \cdot \\ x_n^i y^j \end{bmatrix} \tag{3.2.28}$$

This result is easy to verify by expanding the defining relationship $(A \times B)z_{ij} = \lambda_i\mu_j z_{ij}$, using the block structure of (A × B). We can use this result to easily find the eigenvalues and eigenvectors of the Kronecker matrix by using a theorem of Frobenius []9], i.e., if the roots of A are λ(A), then the roots of the polynomial function of A, f(n), are λ(f(A)) = f(λ). Consider

$$(In+\varepsilon A)\times(Im+\varepsilon B) = In\times Im+\varepsilon(A\times Im+In\times B)+\varepsilon^2(A\times B).$$

From the above we have

$$\lambda[(In+\varepsilon A)\times(Im+\varepsilon B)] = (1+\varepsilon\lambda_i)(1+\varepsilon\mu_j) = 1+\varepsilon(\lambda_i+\mu_j)+\varepsilon^2\lambda_i\mu_j.$$

and the matrix $(A \times Im + In \times B)$, therefore has eigenvalues $(\lambda_i + \mu_j)$ with associated eigenvectors z_{ij}. By letting $A = A^T$ and $B = A^T$, this result applies to the Kronecker matrix directly.

Now the singular values of $A \times B$ may be found using the relations (3.2.28), i.e.,

$$(A\times B) = [\lambda (A\times B)^H (A\times B)]^{1/2} = [\lambda (A^H A \times B^H B)]^{1/2} = \sigma(A)\sigma(B).$$

The approach used to obtain the eigenvalues of $(A\times Im+In\times B)$ does not extend to its singular values, however. The essential difficulty may be seen by considering

$$(A\times Im+In\times B)^H (A\times Im+In\times B) = (A^H A\times Im+In\times B^H B)+A^H\times B+A\times B^H \quad (3.2.29)$$

If the last two terms were absent, then the singular values of $(A\times Im+In\times B)$ would be $\sigma_i(A)+\sigma_j(B)$ as was the case for the eigenvalues. Although the eigenvalues of each <u>term</u> on the right hand side of (3.2.29) are products or sums of the eigenvalues of A and B, no expression has been found for the eigenvalues of the complete expression. This is basically unsurprising, for the roots of a sum of matrices are essentially unrelated to the roots of each single matrix, except for matrices of very special form [20].

Some insight may be gained from considering the inequality given previously,

$$k(L_A) \geq \frac{|\max\lambda(K_A)|}{|\min\lambda(K_A)|} = \frac{\max|\lambda_i+\lambda_j|}{\min|\lambda_i+\lambda_j|} \cdot \qquad (3.2.30)$$

Suppose that A is 2×2, with a complex pair of eigenvalues $\alpha \pm jw$, $\alpha << w$. Then for A $|\lambda max|/|\lambda min| = k(A) = 1$, while from equation (3.2.30)

$$k(L_A) \geq 2|\alpha+jw|/\alpha \doteq 2w/\alpha >> 1.$$

This is significant because lightly damped poles are common in many engineering systems, and this simple example shows that the condition number of the Lyapunov equation can become very large although the A matrix itself is very well-conditioned!

Finally, consider the commonly used approximation for the condition number,

$$k(L_A) \leq \|K_A\|_E \|K_A^{-1}\|_E.$$

Now it is easy to see that

$$\|K_A\|_E \leq 2n^{1/2}\|A\|_E ,$$

but again no reasonable bound for $\|K_A^{-1}\|_E$ in terms of A can be obtained. Barnett and Storey [30] give an explicit expression for the inverse of the Kronecker matrix K_A, from which a bound can be obtained, but the result is very pessimistic and has no practical value.

3.3 PERTURBATION BOUNDS

In the previous section the first step in the backward error analysis of the linear matrix equation was developed and extended to the Lyapunov equation. This section pursues the second and more difficult step, which is to assess specific algorithms and obtain bounds for the matrix that represents the equivalent perturbations in the elements of A or L_A. In order to obtain a perturbation bound, an algorithm must be analyzed in great detail, i.e., broken down to the level of each addition and multiplication (division). At this level, the effect of a finite word length is assumed to be such that $\overline{x}(\cdot)\overline{y} \equiv (x(\cdot)y)(1+e)$, $|e| \leq 2^{-t}$, where the bar denotes the computed quantity and $(\cdot)$ can be any of the four basic arithmetic operations. With this assumption, a number of intermediate error bounds, e.g., for inner products, can be obtained. These results have been placed in Appendix B, and are the building blocks of the bounds obtained here. It is generally true that this process of successively bounding elementary operations yields conservative results, and this fact motivates using a statistical approach (see [34] for example), but the bounding approach is preferred by Wilkinson and used here. The purpose of this section, which relies heavily on the works of Wilkinson, is to describe simply the algorithms of interest and the essential steps of their error analysis, omitting many of the details. Once the fundamentals of Appendix B and the basic concepts of backward analysis are grasped, the intermediate results follow in a simple, but somewhat tedious, manner.

Although several of the transformation solution methods of the Lyapunov equation are quite different from the direct method, their error analyses are similar and useful comparisons can be made. Both types of methods rely on a series of elementary similarity transformations at some point in the algorithm, so it is natural to consider these elementary operations separately before proceeding. There are two basic types of similarity transformations, unitary and non-unitary or elementary [22, 24]. The basic formulation of a similarily transformation using elementary matrices is

$$X A X^{-1} = B$$

where X is an elementary matrix. There are many types of elementary matrices, two of which will be used later:

i) The matrices I_{ij}, equal to the identity matrix except in rows and columns i and j, which are of the form

$$\begin{array}{c} \begin{bmatrix} 0 & & 1 \\ & & \\ 1 & & 0 \end{bmatrix} \begin{array}{l} \text{row i} \\ \\ \text{row j} \end{array} \\ \text{col. i} \quad \text{col. j} \end{array}$$

Note that $I_{ij}I_{ij} = I$, i.e., it is orthogonal. Premultiplication by I_{ij} interchanges rows i and j and postmultiplication interchanges columns i and j. Any permulation matrix is a product of matrices of this type.

ii) The matrices N_i, equal to the identity matrix except for the i^{th} column, which is (written as a row vector)

$$(0, 0, \ldots, 0, 1, -n_{i+1,i}, -n_{i+2,i}, \ldots, -n_{ni}).$$

The inverse of N_i is obtained by changing the signs of the $n_{k,i}$ elements. An important property is that the product $N_1 N_2 \ldots N_{n-1}$ is a lower triangular matrix with unit diagonal elements, while the $i\text{-}j^{th}$ element is $-n_{ij}$ $(i>j)$.

The practical use of elementary matrices in computations is illustrated by the fact that there always exists an X that is the product of elementary matrices such that

$$Xx = ke_1 \qquad \text{with:} \quad e_1 \text{ unit vector}, \quad k \text{ real constant}, \quad x \text{ arbitrary vector} \tag{3.3.1}$$

The other type of similarity transformations is based on unitary matrices. Two kinds of elementary unitary matrices are used in practice. The first kind is called a plane rotation; the matrix R(p,q) is defined by

$$r_{pp} = e^{j\alpha}\cos\theta \qquad r_{pq} = e^{j\beta}\sin\theta$$
$$r_{qp} = -e^{-j\beta}\sin\theta \qquad r_{qq} = e^{-j\alpha}\cos\theta$$
$$r_{ij} = \delta_{ij} \text{ otherwise} \qquad \alpha, \beta, \theta \text{ real}$$

With $\alpha=\beta=0$, the transformation is equivalent to a rotation in the p,q plane through the angle θ. The four non-trivial elements can be expressed also by

$$r_{pp} = \bar{x}/r \qquad r_{qq} = x/r \qquad r_{pq} = \bar{y}/r \qquad r_{qp} = -y/r$$

$$r^2 = |x|^2 + |y|^2 \tag{3.3.3}$$

The pre (post)-multiplication of A with R(p,q) affects (independently) only rows (columns) p and q. Let $z = R(p,q)x$ and construct R(p,q) using (3.3.3) with x_p for x and x_q for y. Then $z_p = r$, $z_q = 0$, and $z_i = x_i$ otherwise.

The second kind of elementary unitary matrix is called an elementary Hermitian matrix. These are of the form

$$P(r) = I-2w(r)w^H(r), \qquad \text{where } \|w(r)\|_2=1 \tag{3.3.4}$$

and $$w^T(r) = (0, 0, \ldots 0, w(r)_{r+1}, w(r)_{r+2}, \ldots, w(r)n).$$

It is easy to verify that P(r) is Hermitian and unitary. When w(r) is real, P(r) is real, symmetric and orthogonal, Pre (post)-multiplication of A by P(r) affects only rows (columns) r+1 to n and treats each column (row) of A independently. Given a vector x, we can choose w(r) such that P(r)x has zero elements in positions r+2, r+3, ..., n. To do this consider

$$S^2 = |x_{r+1}|^2 = \ldots + |x_n|^2,$$

$$T = (|x_{r+1}|^2 S^2)^{1/2}, \quad H = S^2 + T \tag{3.3.5}$$

$$w^T(r) = (0, 0, \ldots, w_{r+1}(1+S^2/T), x_{r+2}, \ldots x_n).$$

When x is real (3.3.5) can be expressed more simply by

$$S^2 = x_{r+1}^2 + \ldots + x_n^2 \qquad H = S^2 + |x_{r+1}|S,$$

$$w^T(r) = (0, 0, \ldots, w_{r+1} \pm S, x_{r+2}, \ldots x_n), \tag{3.3.6}$$

where $\pm S$ is chosen to have the same signs as x_{r+1}. Notice that the above transformation P(r)x is identical to the succession of transformations R(r+1, r+2), R(r+1, r+3), ..., R(r+1, n) applied to x. Since each column of A is treated independently in forming P(r)A, we can reduce elements r+2, ..., n of any column of A to zero without affecting the first r elements.

In matrix problems similarity transformations are a major part of many numerical algorithms [24]. The two basic types differ in an important way that may affect the numerical stability of the algorithm. The difference is that unitary similarity transformations preserve the conditioning of the original matrix while elementary transformations, in general, do not. Let U be unitary and consider

$$U\,A\,U^H = B. \qquad U^H U = I$$

Now

$$\|B\|_2^2 = \|UAU^H\|_2^2 = \max\lambda\,(UAU^H)^H(UAU^H)$$

$$= \max\lambda\,(UA^HAU^H) = \max\lambda\,(A^HA) = \|A\|_2^2\,, \qquad (3.3.7)$$

so the spectral norm is invarient under unitary similarity transformations. The difficulty with elementary similarity transformations can be illustrated simply. Equation (3.3.6) shows how to construct a P(r) such that for

$$y = P(r)x, \quad y_k = 0 \qquad k = r+2, \ldots, n.$$

Suppose we attempt the same transformation with an elementary matrix N_r, i.e.,

$$y = N_r x \qquad (3.3.8)$$

Clearly the appropriate elements of N_r must be

$$n_{k,\,r} = x_k/x_r\,, \quad k>r\,,$$

and the transformation breaks down if $x_r = 0$ and $x_k \neq 0$ for some $k>r$. Generally, if x_r is small relative to some x_k, then the rounding errors become large. (The details of why this is true are outlined in Appendix B.) This difficulty of numerical instability can be greatly

reduced using the familiar operation of pivoting, which is affected using the orthogonal matrices I_{ij} mentioned previously. To illustrate this, and to complete the transformation (equation 3.3.8), let q be the smallest integer such that

$$|x_q| = \max_{i \geq r} |x_i|$$

Then let $z = I_{rq}x$ and again choose N_r so that $N_r z$ has zero elements in positions r+1, ...n. We have

$$n_{ir} = z_i/z_r , \qquad |n_{ir}| \leq 1$$

and the difficulty is avoided. The combination I_{ra} and N_r is called a stabilized elementary matrix.

Theoretically then, unitary similarity transformations are superior with respect to numerical stability. Wilkinson (and others, e.g. [17, 22]) states that in practice stabilized elementary transformations are almost as stable as unitary transformations, provided the pivoting strategy is successful. The latter fact complicates the a priori error analysis, however.

The basic properties of elementary similarity transformation have been introduced. In the remainder of this section algorithms are formulated and perturbation bounds are given for the following problems:

i) $Ax = b$

ii) Transformation to Hessenberg form

iii) Transformation to real Schur form

iv) Transformation to Companion form.

The results of these analyses will then be extended to direct and transformation Lyapunov solution methods in the final section of this chapter.

If x is a solution of

$$Ax = b \tag{3.3.9}$$

and S is a square, nonsingular matrix, then it is also a solution of

$$SAx = Sb \tag{3.3.10}$$

The basic idea is to construct a simple matrix S such that SA is upper triangular. This is accomplished in a series of steps, each of which produces a set of equations equivalent to equation (3.3.9). At the $(r-1)^{th}$ step we have

$$A_{r-1}x = b_{r-1} \tag{3.3.11}$$

where A_{r-1} is upper triangular in the first r-1 columns. The r^{th} step consists of finding an elementary matrix of the form N_r such that N_rA_{r-1}

is upper triangular in the first r columns and is identical to A_{r-1} in the first (r-1) rows and columns. Thus, the r^{th} step is precisely the problem covered previously (e.g., equation 3.3.8). If we ignore the pivoting issue for now and denote the elements of A at the r^{th} step a_{ij}^{r}, then the critical elements of N_r are given by

$$n_{ir} = a_{ir}^{r-1}/a_{rr}^{r-1} \quad , \tag{3.3.12}$$

and

$$A_r = N_r A_{r-1} \quad , \quad b_r = N_r b_{r-1}. \tag{3.3.13}$$

Combining equations (3.3.13) for r=1 to n-1, we have

$$N_{n-1} \cdots N_2 N_1 A_0 = A_{n-1} \quad , \quad N_{n-1} \cdots N_2 N_1 b_0 = b_{n-1} \quad ,$$

where $A_0 = A$ is the original matrix. Recalling the property of the product $N_{n-1} \cdots N_2 N_1$ mentioned previously, define

$$L = N_1^{-1} N_2^{-1} \cdots N_{n-1}^{-1} \qquad \text{(unit lower trangular matrix)}$$

where

$$\ell_{ij} = n_{ij} \qquad i>j \tag{3.3.14}$$

$$= \delta_{ij} \text{ otherwise.}$$

The arguments made earlier regarding pivoting definately apply here, but it can be readily verified that the inclusion of a matrix I_{rq} at the r^{th} step does not affect the essential result, which is

$$A_0 x = LA_{n-1} x \equiv LUx = b. \tag{3.3.15}$$

L and U are lower and upper triangular matrices, respectively, so x may be found by performing simple forward and back substitution. Of course rounding errors occur at each step, so the computed L and U correspond to the exact triangularization of $A_0 + E$, i.e.,

$$LU = A_0 + E. \tag{3.3.16}$$

For n = 3,

$$L = \begin{bmatrix} 1 & & \\ n_{21} & 1 & \\ n_{31} & n_{32} & 1 \end{bmatrix} \qquad U = \begin{bmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 \\ & a_{22}^2 & a_{23}^2 \\ & & a_{33}^3 \end{bmatrix}$$

This basic bounds outlined in Appendix B can be applied for each step in the process to construct the perturbation matrix E. If the pivoting strategy is successful, $|n_{ij}| \leq 1$. If the maximum element of A_r is denoted by g, then

$$|E| \leq 2g2^{-t} \begin{bmatrix} 0 & 0 & 0 & . & . & . & 0 & 0 \\ 1 & 1 & 1 & . & . & . & 1 & 1 \\ 1 & 2 & 2 & . & . & . & 2 & 2 \\ 1 & 2 & 3 & . & . & . & 3 & 3 \\ . & . & . & . & . & . & . & . \\ 1 & 2 & 3 & . & . & . & n-1 & n+1 \end{bmatrix} \tag{3.3.17}$$

Similar analyses can be applied to the solution of the triangular sets of equations

$$(L + \delta L)y = b \qquad (U + \delta U)x = y \qquad (3.3.18)$$

Before collecting these results, however, it is worthwhile to repeat an argument made by Wilkinson [18] regarding the solution of triangular sets of equations that has important practical consequences. To illustrate this consider solving $(L + \delta L)y = b$. Assume for now that $\|\delta L\| \leq n2^{-t}\|L\|$. Equation (3.2.13) then provides the estimate

$$\frac{\|y-L^{-1}b\|}{\|L^{-1}b\|} \leq \frac{n2^{-t}\|L\|\|L^{-1}\|}{1-n2^{-t}\|L\|\|L^{-1}\|} \qquad (3.3.19)$$

This implies that if $\|L\|\|L^{-1}\|$ is large, then the accuracy of the computed solution will deteriorate. However, in practice, large errors do not occur in the solution of triangular sets of equations and Wilkinson has found that a realistic estimate is

$$\frac{\|y-L^{-1}b\|}{\|L^{-1}b\|} \leq f(n)2^{-t} \quad , \qquad (3.3.20)$$

where f(n) is a linear function of n and the accuracy is <u>not</u> affected by the condition number of the matrix L. This contrasts greatly with the general case of solving Ax = b, and it is important to realize that this is a special property of triangular sets of equations [18]. The

practical consequence for the general case is that when b is such that the corresponding solution reveals the ill-conditioning of A, it is the errors due to the LU factorization which limit the accuracy of the solution of $Ax = b$.

Combining equations (3.3.16, 3.3.18), we have

$$(A + E + L\delta U + U\delta L + \delta L\delta U)x = b,$$

$$\text{or} \qquad (A + K)x = b. \tag{3.3.21}$$

Taking the norm of equation (3.3.17) and of analogous results for the other perturbation bounds yields

$$\|E\|_\infty \leq 2g2^{-t}(n/2 + 1)(n - 1)$$

$$\|\delta L\|_\infty \leq 1/2(n^2 + n + 2)2^{-t}$$

$$\|\delta U\|_\infty \leq g/2(n^2 + n + 2)2^{-t}$$

$$\|L\|_\infty \leq n$$

$$\|U\|_\infty \leq gn,$$

so $$\|K\|_\infty \leq 2^{-t}g(2n^2 + n^3).$$

Again $g = \max|a_{ij}|$, but Wilkinson has found that $\max|a_{ij}^r| \leq 8\max|a_{ij}|$ for almost all matrices, when pivoting is used. So if A is initially normalized such that $|a_{ij}| \leq 1/8$, then g is essentially 1. In addition he states

that this bound is pessimistic, which is to be expected as statistical variations have been neglected throughout. For practical purposes, the expected error is

$$\|K\|_\infty \leq gn2^{-t} \tag{3.2.22}$$

with inner product accumulation. If accumulation is not available, the factor n becomes $n^{3/2}$. Before beginning the analysis of the reduction to Hessenberg form, a few concluding remarks are appropriate regarding the solution of Ax = b. First, the factorization algorithm presented was chosen to emphasize the use of similarity transformations because they conceptually link all the analyses of this section. Algorithms which carry out this process in a computationally different and more efficient manner are well known [21]; in addition, if accumulation of inner products is available, a direct LU factorization scheme provides a better bound for E, namely $\|E\|_\infty \leq g \cdot n2^{-t}$. The result of equation (3.2.22) still applies, however. Secondly, it is important to clearly recognize the difference between residuals and accuracy. In order to illustrate this, call the computed solution x', so that the residual vector is

$$r' = b - Ax' \tag{3.3.23}$$

Combining equations (3.3.21, 3.3.22) then

$$\|r'\|_\infty = \|b - Ax'\|_\infty \leq gn2^{-t}\|x'\| \ .$$

Thus r' is bound to be small relative to x' <u>regardless</u> of the accuracy of x'. For an a priori accuracy estimate we use equation (3.2.13). Of course one conclusion of the last section was that finding the actual condition number is a major computation, and that a reliable a priori estimate of it is not generally available (except for the use of an approximate inverse). Another approach, which is more practical, employs iterative refinement of the solution.

Define a sequence of back-substitutions using the computed L and U by

$$r^s = b - Ax^s$$

$$x^{s+1} = x^s + (LU)^{-1}r^s \ . \qquad (3.3.24)$$

Wilkinson proves that if

$$\|A^{-1}\|\|E\| < 2^{-p} \ , \quad p>1$$

then $$\|x^{s+1} - x\| \leq \frac{2^{-p}}{p-2^{-p}} \|x^s - s\| \ . \qquad (3.3.25)$$

For $p \geq 2$, at least p significant bits are gained per iteration. The additional work is $0(1/n)$ times the original factorization, but it is <u>essential</u> that the residuals are accumulated in double precision. For most A matrices a single refinement is sufficient, in which case output

of the quantity $\|x^2 - x^1\|_\infty / \|x^2\|_\infty$ provides a reliable assessment of the accuracy of x^2.

The second problem for which perturbation bounds are required is that of transforming a general matrix A to Hessenberg form, H. Recall that a Hessenberg matrix is of the form

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & & \vdots \\ 0 & h_{32} & h_{33} & & \vdots \\ & 0 & h_{43} & h_{44} & \\ & & 0 & \ddots & \ddots \\ & & & \ddots & h_{n-1,n} \\ & & & 0 \; h_{n,n-1} & h_{nn} \end{bmatrix} .$$

For symmetric matrices a tridiagonal matrix is analogous to H, and methods that affect the transformation are conceptually identical. The transformation is carried out in a series of n-1 steps (A n×n); at the beginning of the r^{th} step we have

$$A_{r-1} = \left[\begin{array}{c|c} H_{r-1} & C_{r-1} \\ \hline 0 \; b_{r-1} & B_{r-1} \end{array}\right] \begin{matrix} r \\ n-r \end{matrix} \qquad (3.3.26)$$

$$\begin{matrix} r & n-r \end{matrix}$$

b_{r-1} is $(n - r \times 1)$,

and the problem is to construct a matrix X_r such that

$$A_r = X_r A_{r-1} X_r^{-1} \tag{3.3.27}$$

Any of the transformation matrices introduced at the beginning of this section can be used. The use of plane rotation matrices is referred to as Given's method, while Householder's method [22, 24] employs the unitary Hermitian transformation matrices. Householder's method will be used to illustrate the basic process, so X_r is replaced by P_r, i.e.,

$$P_r = \begin{bmatrix} I & & \\ & & r \\ & & \\ & Q_r & n-r \end{bmatrix}$$

where $Q_r = I - 2v_r v_r^T$ and we assume that the elements of A are real (P_r is orthogonal). The result of the multiplications (3.3.27) with P_r is

$$\begin{aligned} H_r &= H_{r-1} \quad , \qquad C_r = C_{r-1} Q_r \\ b_r &= Q_r b_{r-1} \quad , \qquad B_r = Q_r B_{r-1} Q_r \ . \end{aligned} \tag{3.3.29}$$

The method relies on constructing a Q_r such that $b_r = Q_r b_{r-1}$ has zero elements in positions r+2, r+3, ..., n. This construction was given

previously by equation (3.3.6). Thus, proceeding from r=1 to r=n-1, we have

$$P_{n-1}P_{n-2} \cdots P_1 A P_1 \cdots P_{n-2}P_{n-1} = PAP = H \qquad (3.3.30)$$

Again, rounding errors occur at each step and the aim of backward error analysis is to derive perturbation bounds, viewing the computed H as the exact transformation of a perturbed A matrix, i.e.,

$$H = P(A + E)P. \qquad (3.3.31)$$

Wilkinson [24] does this in great detail for a number of variations of the basic algorithm. The following development, which is valid for any unitary transformation, is both illustrative of the common features of his approach and useful for obtaining a number of results because of the generality. The matrices E, X, Y, and Z will denote perturbation matrices; other matrices are assumed exact unless over-lined with a bar, which symbolizes a computed quantity. Finally, although the analysis is for unitary similarity transformations, we assume that the P_r matrices are orthogonal merely to simplify the notation.

So, $A_r = P_r A_{r-1} P_r$ is the exact transformation at the r^{th} step, and

$$\bar{A}_r = \bar{P}_r \bar{A}_{r-1} \bar{P}_r + E_r \qquad (3.3.32)$$

where $\bar{P}_r = P_r + X_r$.

For any of the different algorithms, we can compute a constant a that depends on the specific arithmetic operations used in forming P_r such that $\|X_r\|_2 \leq a2^{-t}$. Now

$$\bar{A}_r = (P_r + X_r)\bar{A}_{r-1}(P_r + X_r) + E_r \tag{3.3.33}$$

$$= P_r\bar{A}_{r-1}P_r + Y_r$$

where $$Y_r = X_r\bar{A}_{r-1}P_r + P_r\bar{A}_{r-1}X_r + X_r\bar{A}_{r-1}X_r + E_r \ . \tag{3.3.34}$$

Let r = 1, 2, ..., n

$$\bar{A}_n = G_1A_0G_1 + G_2Y_1G_2 + \ldots G_nY_{n-1}G_n + Y_n \tag{3.3.35}$$

where $G_r = P_rP_{r-1} \ldots P_n$,

or $\bar{A}_n = G_1A_0G_1 + Y$,

or $$\bar{A}_n = G_1(A_0 + Z)G_1 \tag{3.3.36}$$

where $Z = L_nY_nL_n + L_{n-1}Y_{n-1}L_{n-1} + \ldots + L_1Y_1L_1$

$L_r = P_1P_2 \ldots P_r$.

Certainly

$$\|Y\|_2 \leq \|Y_n\|_2 + \|Y_{n-1}\|_2 + \ldots + \|Y_1\|_2 ,$$

and using $\|X_r\|_2 \leq a2^{-t}$ as well as the invariance of the spectral norm under unitary similarity transformations, equation (3.3.34) implies

$$\|Y_r\|_2 \leq \|\bar{A}_{r-1}\|_2(2a2^{-t} + a^2 2^{-2t}) + \|E_r\|_2 \qquad (3.3.37)$$

while (3.3.33) yields

$$\|\bar{A}_r\|_2 \leq \|\bar{A}_{r-1}\|_2 + \|Y_r\|_2 \leq (1+a2^{-t})^2\|\bar{A}_{r-1}\|_2 + \|E_r\|_2 \qquad (3.3.38)$$

Considering equation (3.3.32), we can analyze the specific algorithm of interest to find a bound of the form

$$\|E_r\|_2 \leq f(r, n)2^{-t}\|\bar{A}_{r-1}\|_2$$

where $f(r, n)$ is a simple function of r and n in general, although in some cases it is a constant. Using this, equations (3.3.37, 3.3.38) become

$$\|Y_r\|_2 \leq (2a2^{-t} + a^2 2^{-2t} + f(p,n)2^{-t})\|\bar{A}_{r-1}\|_2$$

$$\|\bar{A}_r\|_2 \leq ((1+a2^{-t})^2 + f(p, n)2^{-t})\|\bar{A}_{r-1}\|_2 .$$

Finally, for

$$\bar{A}_n = G_1(A_0 + Z)G ,$$

$$\|Z\|_2 \leq 2^{-t}\|A_0\|_2 \sum_{r=1}^{n} \{[2a+a^2 2^{-t}+f(r,n)] \prod_{i=1}^{r} [(1+a2^{-t})^2+f(i,n)2^{-t}]\} .$$

Applying the basic results of Appendix B to the Householder method one can find that f(i, n) should be of the form f(i, n) = k(n-i). Shoving this into the above result yields a complicated series which blows up rapidly as n gets very large, but in the useful range the result is (for a computer without accumulation of inner products)

$$\|Z\|_2 \leq \|A_0\|_2 (k_1 n^2 + k_2 n^4 2^{-t}) 2^{-t} \quad . \tag{3.3.39}$$

If accumulation of inner products is available, replace n^2 with n and n^4 with n^2.

The third problem for which perturbation bounds are required is that of transforming a matrix in Hessenberg form to real Schur form. Recall that the real Schur form is a generalization of the triangular form wherein 2×2 blocks along the diagonal correspond to complex conjugate pairs of eigenvalues. Of course, this transformation solves the eigenvalue problem, which is of considerable importance and difficulty. As a result, a thorough understanding and analysis of the Bartels and Steward Lyapunov equation solution method requires a preliminary study of the eigenvalue problem, which would be in itself a substantial thesis topic.

There exist many ways to obtain the Schur form which are variations of two basic methods, the LR and QR algorithms [24]. Both are inexact procedures in which an infinite sequence of similarity transformations are successively applied to a general matrix A; the former

relies on elementary matrices to do this while the latter employs unitary transformations. Neither algorithm actually requires the Hessenberg form, but the computational advantages of obtaining it for the general A first are great. As was mentioned previously, elementary transformations are almost as numerically stable as unitary ones if they can be stabilized. A stabilized process is one in which the elements of the transformation matrices are strictly bounded. For the LR algorithm numerical stability and speed of convergence are conflicting properties because preserving the Hessenberg form at each step (least number of elements below the diagonal, i.e., fewest operations) eliminates pivoting options and vice versa. The QR algorithm is, therefore, generally "better", and will be used to illustrate the basic reduction. It is defined by the simple recursion

$$A_s = Q_s R_s \ , \quad A_{s+1} = Q_s^H A_s Q_s = R_s Q_s \ , \tag{3.3.40}$$

where Q_s is unitary and R_s is upper triangular. Manipulating (3.3.40),

$$A_{s+1} = (Q_s^H Q_{s-1}^H \cdots Q_1^H) A_1 (Q_1 Q_2 \cdots Q_s) \ ,$$

or

$$(Q_1 Q_2 \cdots Q_s) A_{s+1} = A_1 (Q_1 Q_2 \cdots Q_s) \ . \tag{3.3.41}$$

Denoting $(Q_1 Q_2 \cdots Q_s) = P_s$ and $(R_s R_{s-1} \cdots R_1) = U_s$, we have

$$P_s U_s = (Q_1 \cdots Q_{s-1})(Q_s R_s)(R_{s-1} \cdots R_1)$$

$$= (Q_1 \cdots Q_{s-1}) A_s (R_{s-1} \cdots R_1)$$

$$= A_1 (Q_1 \cdots Q_{s-1})(R_{s-1} \cdots R_1)$$

$$= A_1 P_{s-1} U_{s-1} \cdot$$

Repeating for $P_{s-1}U_{s-1}$, $P_{s-2}U_{s-2}$, etc. yields

$$P_s U_s = A_1^s \cdot$$

Therefore $P_s U_s$ is the corresponding factorization of A_1^s, and in either case Wilkinson [24] shows that the factorization is unique if the diagonal elements of the R_s are taken to be positive. Computationally, each iteration (3.3.40) involves two essential steps. The first step is to construct an orthogonal Q_s such that (assume A real)

$$Q_s^T A_s = R_s \ , \qquad (3.3.42)$$

where R_s is upper triangular. Notice that this step is analogous to the triangular factorization covered previously (equation 3.3.15). Now it is easy to verify that if A_1 is in Hessenberg form, so is each A_s. Therefore, Q_s^T can be constructed as a product of (n-1) plane rotations in the (1, 2), (2, 3), ...k (n-1, n) planes, i.e.,

$$Q_s^T = R(n-1, n)R(n-2, n-1) \ldots R(1, 2) \qquad (3.3.43)$$

where the R(p, q) are defined by equation (3.3.3). The second step is to then successively post-multiply R_s with the transposes of R(p, q),

$$R_s(R^T(1, 2)R^T(2, 3) \ldots R^T(n-1, n)) = R_sQ_s = A_{s+1} \quad .(3.3.44)$$

The A_s tend in the limit to a matrix of the form

$$\begin{bmatrix} X_1 & x & \cdot & \cdot & \cdot & x \\ & X_2 & x & \cdot & \cdot & x \\ & & X_3 & & & \cdot \\ & & & \cdot & & \cdot \\ & & & & \cdot & \cdot \\ & & & & X_{p-1} & x \\ & & & & & X_p \end{bmatrix} \quad X_i(n_i \times n_i)$$

The dimension of each X_i is equal to the number of distinct eigenvalues of equal modulus. An interesting property of the convergence is that the eigenvalues of the X_i converge to eigenvalues of A, while elements above the (block) diagonal do <u>not</u> tend to a strict limit, but may change from iteration to iteration by a factor ultimately of modulus unity [22, 24].

The analysis developed previously, equations (3.3.32, 3.3.38), can be applied to the QR algorithm in order to obtain perturbation

bounds. In terms of the notation used there, however, the summation was over columns of A, but is now taken over steps of the QR algorithm. In this case f(s, n) must also be computed with the same procedure, realizing that the successive A_s are in Hessenberg form. Accomplishing this, we find

$$f(s, n) \doteq k_1 n \tag{3.3.45}$$

$$a \doteq k_2 n$$

from which the final perturbation bound is

$$\bar{A}_s = P_s^T(A_0 + Z_s)P_s \ , \ A_0 \text{ in Hessenberg form,}$$

$$\|Z_s\| \leq \|A_0\| [n \cdot s(K_3 + K_2^2 n 2^{-t}) 2^{-t} + n^2 s^2 (K_3^2 + K_2^2 K_3 n 2^{-t}) 2^{-2t}] \ , \tag{3.3.46}$$

$$K_3 = K_1 + 2K_2 \ .$$

Note that this bound neglects terms that go to infinity as the product ns gets very large, but is applicable in the useful range of ns.

The final problem of this section is that of transforming a general matrix A to Companion (or Frobenius) form. As usual, several methods may be used. A well-known algorithm is that of Danilewski [31], which uses elementary matrices to affect the transformation. The difficulty with this method can be seen by considering the r^{th} step:

$$\begin{bmatrix} 1 & & & -n_1 & & \\ & 1 & & \cdot & & \\ & & \cdot & \cdot & & \\ & & & \cdot & & \\ & & & 1 \;\; -n_r & & \\ & & & n_{r+1} & & \\ & & & \cdot \;\; 1 & & \\ & & & \cdot & \cdot & \\ & & & -n_n & & 1 \end{bmatrix} \begin{bmatrix} 0 & & & a^r_{1r} & \cdot \;\; \cdot & a^r_{1n} \\ 1 & 0 & & \cdot & & \cdot \\ & 1 & & \cdot & & \cdot \\ & & \cdot & \cdot & & \cdot \\ & & \cdot \;\; 0 & a^r_{rr} & & \cdot \\ & & & 1 \;\; a^r_{r+1,r} & & \cdot \\ & & & \cdot & & \cdot \\ & & & \cdot & & \cdot \\ & & & a^r_{n,r} & \cdot \;\; \cdot & a^r_{nn} \end{bmatrix} \tag{3.3.47}$$

Clearly the appropriate choices for the critical elements of N_{r+1} are $n_k = a^r_{kr}/a^r_{r+1,r}$ $(k \neq r+1)$, $n_{r+1} = 1/a^r_{r+1,r}$. If pivoting is used (it should be), our choice of pivotal elements is restricted to $a^r_{k,r}$ $k = r+1, \ldots, n$. We can insure $|n_k| < 1$ only for $k = r+1, r+2, \ldots, n$. so the transformation can not be stabilized. Because of this, a priori perturbation bounds can not be obtained.

Wilkinson criticezes the Danilewski algorithm [24], and proposes a two-step procedure that improves, but does not completely eliminate, the numerical instability. First, the matrix A is transformed to Hennesberg form using either unitary or stabilized elementary matrices. The matrix H is then reduced to companion form with a series of unstabilized elementary transformations. An algorithm for the second step can be obtained directly from the form of the similarity transformation, MH = CM, i.e., for n = 5

$$\begin{bmatrix} m_{11} & m_{12} & \cdot & \cdot & m_{15} \\ & m_{22} & \cdot & \cdot & \cdot \\ & & m_{33} & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & m_{55} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & \cdot & \cdot & h_{15} \\ h_{21} & h_{22} & \cdot & \cdot & h_{25} \\ & h_{32} & \cdot & \cdot & h_{35} \\ & & h_{43} & \cdot & h_{45} \\ & & & h_{54} & h_{55} \end{bmatrix} = \begin{bmatrix} 1 & & & & p_1 \\ & 1 & & & p_2 \\ & & 1 & & p_3 \\ & & & 1 & p_4 \\ & & & & 1 \; p_5 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & \cdot & \cdot & m_{15} \\ & m_{22} & \cdot & \cdot & \cdot \\ & & m_{33} & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & m_{55} \end{bmatrix}$$

(3.3.48)

The rows of M may be determined successively ($m_{11} = 1$, free choice), $r = 1, 2, \ldots, n$ with

$$m_{r,j+1} = (m_{k-1,j} - \sum_{k=r}^{j} m_{r,k} h_{k,j}) 1/h_{j+1,j} \qquad (3.3.49)$$

$$j = r-1, \ldots, n-1.$$

Then $p_r = (m_{r-1,n} - \sum_{k=r}^{n} m_{r,k} h_{k,n}) 1/m_{n,n}$.

If any of the $h_{j+1,j}$ are zero or very small, the $m_{r,j+1}$ become very large and large rounding errors will result. Some criteria should be used to set those elements to zero and therefore decompose H into a direct sum of block Hessenberg matrices, each of which may then be transformed individually. Unfortunately, perturbation bounds can not be obtained for this second step in the reduction. An interesting idea [24] which could further improve the numerical stability of this step is to modify the first step such that the sub-diagonal elements

are either 0 or 1. Theoretically, this is possible as the modified Hessenberg form must be exactly similar to the corresponding Frobenius canonical form. Note that if no zeros appear on the subdiagonal, then the matrix must be non-derrogatory and the Frobenius form degenerates to the Companion form. The following algorithm, which is a modification of an algorithm developed in [24], illustrates an approach that may solve this problem. Stabilized elementary transformations are used to form a matrix N such that

$$AN = NH \tag{3.3.50}$$

where N is lower triangular except in the first column, which is e_1 (unit vector). If the r^{th} column of the products in (3.3.50) are equated, we have

$$\left.\begin{aligned} \sum_{k=r}^{n} a_{ij} n_{kr} &= \sum_{k=2}^{i} n_{ik} h_{kr} \qquad i=2,\ 3,\ \ldots,\ r \\ &= \sum_{k=2}^{r} n_{ik} h_{kr} + n_{i,r+1} \qquad i=r+1,\ \ldots,\ n \end{aligned}\right\} \quad r=2,\ 3,\ \ldots,\ n$$

and at the r^{th} step the r^{th} column of H and the $(r+1)^{th}$ column of N can be obtained.

Consider the situation at the r^{th} step for n=5 and r=3, where critical elements are represented as they might be stored in the computer,

$$\begin{bmatrix} h_{11} & h_{12} & a_{13} & a_{14} & a_{15} \\ n_{22} & h_{22} & a_{23} & a_{24} & a_{25} \\ n_{32} & n_{33} & a_{33} & a_{34} & a_{35} \\ n_{42} & n_{43} & a_{43} & a_{44} & a_{45} \\ n_{52} & n_{53} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

$\uparrow$

r^{th} step

i) determine r^{th} column of H

$$\left.\begin{array}{l} h_{ir} = (\sum_{k=r}^{n} a_{ik} n_{kr}) 1/n_{ii} \\ \text{store in } a_{ir} \end{array}\right\} i=2, 3, \ldots, r$$

ii) compute (3.3.51)

$$\left.\begin{array}{l} n_{i,r+1} = \sum_{k=r}^{n} a_{ik} n_{kr} - \sum_{k=2}^{r} n_{ik} h_{kr} \\ \text{store in } a_{ir} \end{array}\right\} i=r+1, \ldots, n-1$$

iii) let (r+1)' be the first integer such that

$$|n_{i(r+1)'}| = \max_{i>r} |n_{i,r+1}|$$

exchange rows and columns (r+1) and (r+1)'

iv) if $|n_{r+1',r+1'}| \leq \varepsilon$ (some tolerance), then take the r^{th} subdiagonal of H to be zero and set $n_{(r+1)',(r+1)'}$ (which is now in position (r+1, r)) to 1.

Assuming that the Hessenberg form is constructed with subdiagonal elements zero or unity, then the Frobenius form can be obtained as a direct sum of Companion forms. Each companion form can then be obtained with a slight modification of equations (3.3.48, 3.3.49), where now both the diagonal elements of M and the subdiagonal elements of each reduced order Hessenberg matrix are unity. Numerically, the modification is significant because no divisions are required for the second step. An error analysis of this modified procedure yields a perturbation bound of the form

$$C = M(H + E)M^{-1} ,$$

$$|E| \leq 2g2^{-t} \begin{bmatrix} 1 & 2 & . & . & . & . & n-1 & n \\ 0 & 1 & 2 & . & . & . & n-2 & n-1 \\ & 0 & & . & & & . & . \\ & & & . & . & & & \\ & & & & . & . & . & . \\ & & & & & . & 0 & 1 \end{bmatrix} \qquad (3.3.52)$$

where $g = \max_{r,i,j} |h^r_{i,j}|$.

Notice that this result is similar to that for LU decomposition except for the factor g. This factor complicates the a priori analysis

and reflects the fact that, at one point or another, reduction of a general matrix to companion form requires a series of unstabilized elementary transformations. In order to obtain this bound, it was assumed that the modifications suggested above can be successfully carried out. If the additional assumption is made that the perturbation bound of the reduction of A to Hessenberg form using stabilized elementary transformations is approximately the same as the bound previously obtained for unitary transformations, then the bounds for each step of the two step reduction of A to companion form can be combined. From equation (3.3.39)

$$H = G(A + Z)G,$$

$$\|Z\|_2 \leq \|A\|_2(K_1 n^2 + k_2 n^4 2^{-t})2^{-t} ,$$

and from equation (3.3.52)

$$C = M(H + E)M^{-1} ,$$

$$\|E\|_1 = \|E\|_\infty \doteq g n^2 2^{-t} .$$

Using the fact that the spectral norm is invarient under unitary similarity transformations and that for any matrix B

$$\|B\|_2^2 \leq \|B^H B\|_\infty \leq \|B^H\|_\infty \|B\|_\infty = \|B\|_1 \|B\|_\infty ,$$

then

$$C = MG(A + X)GM^{-1} = D(A + X)D^{-1} ,$$

$$\|X\|_2 \leq (\|A\|_2 K_1 n^2 + gn^2 + \|A\|_2 K_2 n^{4_2-t}) 2^{-t} , \quad (3.3.53)$$

where $g = \max_{r,i,j} |h_{ij}^r|$.

3.4 SUMMARY OF RESULTS

The purpose of this section is to apply the main results of the analyses of the previous sections to the following Lyapunov equation solution techniques: direct method, transformation methods based on the Schur and Companion canonical forms, and the iterative decoupling method. It is intended to be essentially self-contained, but it is important to realize that error analysis does not yield exact results, and that a number of assumptions and qualifying remarks made previously are not repeated here.

Recall that the technique of backward error analysis assumes that the computed solution exactly satisfies the perturbed equation

$$(A^T + E^T)(P + \delta P) + (P + \delta P)(A + E) = -Q \quad (3.4.1)$$

where the matrix E represents the affects of round-off errors that occur during each step of the particular algorithm of interest. The matrix δP represents the resulting solution error, and its relative size is given by equation (3.2.16).

Now from that equation, we see that unless

$$\|K_A\|\|K_A^{-1}\|\|K_E\|/\|K_A\| << 1$$

the solution will be very inaccurate, so assuming that this is true, equation (3.2.16) becomes

$$\|\delta P\|/\|P\| \leq k(L_a)\|K_E\|/\|K_A\| \quad , \tag{3.4.2}$$

where $K_A = A^T \times I + I \times A^T$ and $\|K_A\|_2\|K_A^{-1}\|_2 \equiv k(L_A)$, the spectral condition number of the Kronecker matrix. Technically, equation (3.4.2) is valid only for the 2-norm, due to the definition of the condition number, but we will use whatever norm is most convenient. This is reasonable because it is the ratio on the right hand side of (3.4.2) that is of primary interest here. In addition, some solution methods do not involve the Kronecker matrix, so the relationships

$$\begin{aligned} n^{1/2}\|X\|_E \leq \|K_X\|_E \leq 2n^{1/2}\|X\|_E \ , \\ \|X\|_2 \leq \|K_X\|_2 \leq 2\|X\|_2 \end{aligned} \tag{3.4.3}$$

will be used. Notice that in this case, (3.4.2) becomes

$$|\delta P|/|P| \leq 2k(L_A)|E|/|A| \ . \tag{3.4.4}$$

Finally, the results will be developed assuming that the computer used does not accumulate inner products (e.g., IBM-370), although the appropriate modifications are given if accumulation is available. (See Appendix B for explanation of inner product accumulation.)

Direct Method

First, re-write equation (3.4.1) using the Kronecker notation,

$$(K_A + K_E)(p + \delta p) = -q.$$

Now, from equation (3.2.22), a bound for K_E that accounts for errors made in the LU factorization and in solving the resulting sets of triangular equations is

$$\|K_E\|_E \leq g(n^2/2)^{3/2}2^{-t} \leq \frac{g}{2} n^3 2^{-t} . \qquad (3.4.5)$$

Using (3.4.2, 3.4.3), we have

$$\|\delta p\|_E / \|p\|_E \leq k(L_A)(\frac{g}{2} n^{5/2} 2^{-t}) / \|A\|_E , \qquad (3.4.6)$$

where $g = \max |(K_A)^r_{ij}| \leq 8 \max|(K_A)^0_{ij}| \leq 16 \max|a_{ij}|$. The first inequality reflects the modest growth of the elements of the matrix being factored when pivoting is used (which is essential) and the second follows easily from the formulation of the Kronecker matrix. If accumulation is used, replace $n^{5/2}$ by $n^{3/2}$. It is important to

realize that this bound actually includes a statistical factor suggested by Wilkinson [18] and explained more fully in the previous section.

Equation (3.4.6) is the main result, but an interesting extension, although not entirely consistent with the bounding approach used here, facilitates a comparison with the Bartels and Stewart method analysis that will follow. The basic idea is that if an assumption is made on a statistical distribution of the magnitudes of the elements of A, then the factor g can be related to the euclidian norm and eliminated in equation (3.4.6). To illustrate, let $g = 16 \max|a_{ij}|$ and each element of A be such that $|a_{ij}| = x_{ij} \frac{g}{16}$, where the x_{ij} are independent random variables equally distributed on (0, 1). This formulation is technically incorrect (i.e., g is a random variable now, etc.) but informally at least the expected value of the euclidian norm of A is

$$E[|A|_E] = ng/16\sqrt{3}$$

and using this in equation (3.4.6) we have

$$|\delta P|_E/|P|_E \leq k(L_A) 8\sqrt{3}\, n^{3/2} 2^{-t} \quad . \tag{3.4.7}$$

Bartels and Stewart Method

Letting R denote the product of orthogonal matrices that affected the transformation of A to Hessenberg to real Schur form, the exact Lyapunov equation becomes

$$(RA^TR^T)(RPR^T) + (RPR^T)(RAR^T) = -RQR^T \tag{3.4.8}$$

or $$A_s^TY + YA_s = -C$$

collecting the lower order terms of equations (3.3.39, 3.3.46), we have that the computed Schur matrix $\bar{A}_s$ is exactly similar to the perturbed equation

$$\bar{A}_s = R(A + Z_s)R^T \quad ,$$

$$\|Z_s\|_2 \leq \|A\|_2(k_1n^2 + k_2ns)a^{-t} \tag{3.4.9}$$

where s is the number of iterations of the QR algorithm. Comparing equations (3.2.11, 3.2.13), we see that as far as relative perburbations in the solution are concerned, the errors that occur in forming RQR^T can be added to those that result from transforming A. In addition, assuming that the transformations are applied to Q at each step of the algorithm, the resulting errors are similar to those of (3.4.9), i.e.,

$$\bar{C} = R(Q + \delta Q)R^T \quad ,$$

$$\|\delta Q\|_2 \leq \|Q\|_2(k_1n^2 + k_2ns)2^{-t} \quad . \tag{3.4.10}$$

The next source of error occurs in solving for Y. This step is essentially that of solving a block triangular set of equations,

where each block is at most 2×2. Consider ignoring these errors temporarily. Then, combining equations (3.4.4, 3.4.9, 3.4.10)

$$\|\delta Y\|_2/\|Y\|_2 \leq 4k(L_A)(k_1 n^2 + k_2 ns)2^{-t} \qquad (3.4.11)$$

Now is the previous chapter we found that the solution of a triangular set of equations produces a low relative error that is practice does not depend on the condition number of the triangular matrix, e.g., from equation (3.3.20) the solution of $Lx = b$ yields the very satisfactory bound $\|\delta X\| \leq Kn\|X\|$. Because the 2×2 blocks can be solved explicitly, it seems reasonable for this analysis to assume that we effectively obtain Y from the solution of a sparse set of triangular equations of dimension $n^2/2$ with a relative error bound (ignoring sparsity) on the order of $\|\delta Y\| \leq \frac{k}{2} n^2 \|Y\|$. For even moderate $k(L_A)$ in (3.4.11), this term is relatively unimportant.

The final step in the algorithm is to compute

$$\bar{P} = \bar{R}^T(Y + \delta Y)\bar{R} = P + \delta P,$$

where the first order perturbation terms in δP are

$$\delta P = \delta R^T Y R + R^T Y \delta R + R^T \delta Y R \ . \qquad (3.4.12)$$

In taking the norm of equation (3.4.12), we find that the contribution due to the first two terms is small compared to the last one, so the

final result is essentially that of (3.4.11), i.e.,

$$\|\delta Y\|_2/\|Y\|_2 \leq 4k(L_A)(k_1 n^2 + k_2 ns)2^{-t} \qquad (3.4.13)$$

A final comment on this result is that we expect s, the number of QR iterations, to be some linear function of n, and that accumulation of inner products reduces the $k_1 n^2$ term to $k_3 n$ if Householder's method is used to affect the transformation to Hessenberg form.

Companion Form Methods

Unfortunately, a complete error analysis of Lyapunov equation solution methods that rely on the Companion form cannot be obtained. There are several reasons for this. First, the various comments made in the previous section apply here, and they all basically reflect the fact that at some point in transforming A to Companion form unstabilized elementary transformations must be used. Molinari's algorithm was criticized in particular for ignoring this difficulty entirely. Several improvements were suggested, however, and they led to the intermediate result (3.3.53)

$$\begin{aligned} \bar{C} &= D(A + X)D^{-1} , \\ \|X\|_2 &\leq (k_1 n^2 \|A\|_2 + gn^2)2^{-t} , \\ g &= \max|h_{ij}^r| , \end{aligned} \qquad (3.4.14)$$

where H is the Hessenberg matrix and r denotes the r^{th} step in transforming H to C. Now the next step in this solution method is to form the Hurwitz matrix W, which is constructed from the computed elements of C, i.e., the coefficients of the characteristic polynomial of A. W is then factored, from which the last row of $Y = D^{-T}PD^{-1}$ can be obtained. The difficulty is that the method of backward analysis tells us that $\bar{C}$ is exactly similar to a perturbed A matrix (i.e., (3.4.14)), but this gives no indication of the relative accuracy of the elements of $\bar{C}$. The perturbations in these elements affect not only the analysis of the factorization of W and solution of $Y_r(n)$ (last row), but that of the recursion used to obtain the remaining rows $Y_r(j)$, j=n-1, n-2, ..., as well.

Although several analyses of these steps were performed using a combination of forward and backward techniques, the results are not very consistent and are therefore not reported here. The subjective conclusion reached, however, is that this Lyapunov equation solution method is probably less accurate than the two analyzed previously.

Iterative Decoupling Algorithm

For this analysis, the notation Ax = b will be used to illustrate the approach because it is simpler. The result is then easily extended to the Laypunov equation.

Again, using the backward approach, the k^{th} iteration can be viewed as

$$(A_0 + E)(x^k + \delta x^k) = b - A_1(x^{k-1} + \delta x^{k-1}) \quad . \quad (3.4.15)$$

Consider the first step, i.e.,

$$(A_0 + E)(x' + \delta x') = b,$$

or $$\delta x' = -(I + A_0^{-1}E)^{-1}A_0^{-1}Ex^1 \ .$$

From equation (3.2.13)

$$\|\delta x'\| \leq k(A_0)^{\|E\|}/\|A_0\| \ \|x'\|$$

where we assume that $k(A_0)^{\|E\|}/\|A_0\| << 1$. (Note that we can effectively force this by using iterative refinement if necessary.) Now let $r = k(A_0)^{\|E\|}/\|A_0\|$, $s = \|A_0^{-1}A_1\|$, and consider the second step,

$$(A_0 = E)(x^2 + \delta x^2) = b - A_1(x' + \delta x') \ ,$$

or $$\delta x^2 = -(I + A_0^{-1}E)^{-1}A_0^{-1}Ex^2 - (I + A_0^{-1}E)^{-1}A_0^{-1}A_1\delta x' \ .$$

and $$\|\delta x^2\| \leq r\|x^2\| + sr\|x'\| .$$

Continuing in this fashion, we find

$$\|\delta x^k\| \leq r\|x^k\| + sr\|x^{k-1}\| + \ldots s^{k-1}r\|x'\|$$

or $$\|\delta x^k\| \leq r\|x^*\|(1 + s + s^2 + \ldots + s^{k-1}) = r\|x^*\|\frac{1-s^k}{1-s} \ ,$$

$s < 1$, where $\|x^*\| = \max_k \|x^k\|$.

The desired result is

$$\|\delta x^k\| / \|x^*\| \leq \frac{k(A_0)\|E\| / \|A_0\|}{(1 - \|A_0^{-1}A_1\|)} \quad . \tag{3.4.16}$$

Now the condition $s = \|A_0^{-1}A_1\| < 1$ is sufficient for the convergence of the algorithm, so expressing the series $1 + s + s^2 + \ldots + s^{k-1}$ in the above form is reasonable, but the bound (3.4.16) may be very pessimistic if s is near 1. Equation (3.4.16) is valid for the 2-norm, and the matrices E and A_0 are assumed to be block diagonal, i.e.,

$$E \quad \text{diag}(E_i) \ , \quad A_0 = \text{diag}(A_i) \qquad i=1, 2, \ldots, N$$

$$A_i \quad n_i \times n_i$$

so $$\|E\|_2 = \max_i \|E_i\|_2 \ , \quad \|A_0\|_2 = \max_i \|A_i\|_2 \equiv \|A_i^*\|_2$$

and $$k(A_0) \leq \max_i \|A_i\|_2 / \min_j \|A_j^{-1}\|_2 \quad . \tag{3.4.17}$$

Extending this to the Lyapunov equation, we see that the bound will be similar to that obtained for the direct method previously, except that the numerator in (3.4.16) depends only on the <u>reduced order equations</u>. Suppose that equality is obtained in (3.4.17), so that the i^{th} subsystem (i=j) has the largest condition number, $k(L_{A_i})$. Let $n_m = \max_i n_i$. From (3.4.5)

$$\|K_E\|_2 \leq |K_E|_E \leq n_m^{1/2} |K_E\|_2 \leq \frac{g}{2} n_m^{7/2_2-t}$$

so combining (3.4.6, 3.4.16), the final result is

$$\|\delta P\|_{2/\|P\|_2} \leq \frac{k(L_{A_i})(\frac{g}{2} n_m^{3}2^{-t})/\|A_i^*\|_2}{(1 - \|A_0^{-1}A_1\|)} \quad . \qquad (3.4.18)$$

Although the notation is somewhat cumbersome, the essential point here is that for the iterative decoupling algorithm, it is the errors made in solving the reduced order equations, along with the contraction condition, that limit the accuracy of the final solution.

CHAPTER IV

APPLICATION

4.1 ITERATIVE DECOUPLING PROGRAM

A Fortran computer program that realizes the iterative decoupling algorithm has been written. A general purpose subroutine for solving the Lyapunov equation by the direct method is used to solve the reduced order equations. Sparsity coding techniques have been employed; in particular, a commercial sparse matrix package is used to perform optimal ordering of the Kronecker matrices to make the LU factorizations as sparse as possible, within certain constraints on the relative magnitude of pivotal elements that are important for numerical stability. Although N(N+1)/2 Kronecker matrices must be constructed and factored, the LU factorization is first done symbolically and then numerically. This is useful when some of the diagonal blocks of A share a similar structure because the symbolic factorization need not be repeated for those Kronecker matrices that involve the similar, but not necessarily identical subsystems.

The algorithm has been tested initially with relatively small system matrices, so that the centralized solution can be computed and iteratively refined to a specified accuracy to provide a reliable check. The same accuracy tolerance is used to terminate the decoupling iteration. Specifically, the iteration terminates when each element of the diagonal blocks of P^k, i.e., P^k_{ii}, i=1, ..., N, changes in value from that of the previous step by less than TOL. Note that this is an accuracy tolerance.

In the following example TOL = 10^{-4} and the solution is the steady state covariance of the system $\dot{x}(t) = Ax(t) + \omega(t)$, $E\{\omega(t)\omega'(J)\} =$ $= I\delta(t-J)$.

$$A = \left[\begin{array}{ccc|ccc|ccc} -3 & & & & & & \alpha_1 & & \\ & -2 & -2 & & & & & & \\ 1 & 2 & -2 & & & & & & \\ \hline & & & -4 & & 1 & & & -\alpha_2 \\ \alpha_1 & & & & -3 & -2 & & & \\ & & & & 2 & -3 & & & \\ \hline & & -\alpha_1 & & & & -2 & & \\ & & & & & & & -1 & -.5 \\ & & & & & & & .5 & -1 \end{array}\right]$$

The coupling elements α_1, α_2 were varied as $\alpha_1 = \frac{1}{2}k$, $\alpha_2 = k$ for $k = 0, 1, 2, \ldots, 7$. As k increases, the elements in the diagonal blocks of P move increasingly away from the initial decentralized solution (k=0), and the number of iterations required for convergence naturally grows accordingly. This range of coupling elements was sufficient to vary some of the solution values by two order of magnitude. The number of iterations varied from 2 for k=1 to 9 for k=7. The first solution, k=1, had an execution time of approximately .6 seconds, most of which is forming the Kronecker matrices and factoring them. Each additional solution executed in less than .06 second. The computer used was an IBM-370/168.

4.2 POWER SYSTEM APPLICATION

In this section, another numerical experiment using the iterative decoupling algorithm is reported. Unfortunately the algorithm failed to converge for the system matrices used,so the purpose of this section is to briefly describe what was attempted and why it did not work.

A particular problem of current interest in the study of power system dynamic behavior is that of obtaining transient stability equivalents. In one approach to this problem [42], an important preliminary step is to identify coherent groups of generators, i.e., machines that tend to swing together under the influence of a severe network disturbance. A reliable, but time-consuming, method of making this identification is to simply run a number of large transient stability programs, and visually compare plots of the rotor angle responses of all the generators of interest. Now in some cases simple, linearized machine models may be sufficient for the purpose of identifying coherent groups, and in this case, the solution of a Lyapunov equation provides valuable information. For example, let x_i and x_j be the rotor angles of machines i and j, and suppose the matrix Q is null except for $q_{ij}=q_{ji}=1$. Then

$$M = \int_0^\infty x^T Q x dt = tr(X_0 P) \quad , \tag{4.2.1}$$

where P is the solution of the Lyapunov equation, provides a measure of the coherency of machines i and j.

In this experiment, a simple three machine-infinite bus system was used, where each machine was represented by a constant voltage behind transient reactance, i.e., two state swing equation model. When linearized, these equations are of the form:

$$M_1\Delta\dot{w}_1 + {}^{\Delta w_1}/_{R_1 w_0} = -Y_{11}\delta_1 + Y_{12}\delta_2 + Y_{13}\delta_3$$

$$\dot{\delta}_1 = \Delta w_1$$

$$M_2\Delta\dot{w}_2 + {}^{\Delta w_2}/_{R_2 w_0} = Y_{12}\delta_1 - Y_{22}\delta_2 + Y_{23}\delta_3 \qquad (4.2.2)$$

$$\dot{\delta}_2 = \Delta w_2$$

$$M_3\Delta\dot{w}_3 + {}^{\Delta w_3}/_{R_3 w_0} = Y_{13}\delta_1 + Y_{23}\delta_2 - Y_{33}\delta_3$$

where δ_i = perturbation of machine i's rotor angle from operating point

R_i = droop of machine i

Y_{ij} = transfer admittance between machines i and j

Y_{ii} = self admittance of machine i

Lee [41] studied this same system, and the per unit values used here were taken from his work. He did not include the damping term, but this is necessary in order for the Lyapunov equation solution to exist. A typical set of values used was:

$$M_1 = .1326 \qquad Y_{11} = 2.20 \qquad Y_{12} = 1.0$$
$$M_2 = .1592 \qquad Y_{22} = 2.60 \qquad Y_{13} = .90 \qquad (4.2.3)$$
$$M_3 = .1194 \qquad Y_{33} = 2.30 \qquad Y_{23} = 1.2$$
$$R_1 = R_2 = R_3 = .01$$

With parameters on the order of (4.2.3), the iterative decoupling algorithm did not converge, but slowly moved away from the initial decentralized solution. The reason is that the necessary condition for convergence of the algorithm, i.e., $\rho(L_{A_0}^{-1}L_{A_1}) < 1$, is not satisfied for these typical values. It is interesting to note, however, that $\rho(A_0^{-1}A_1)$ is less than one. In order to see if any simple normalization of the elements of A might help, $L_{A_0}^{-1}L_{A_1}$ was computed symbolically. To illustrate the difficulty, consider that the row norm of this product is of the form:

$$\|L_{A_0}^{-1}L_{A_1}\|_\infty = {}^{Y_{ij}}/_{M_i}(1 + w_0R_iM_i + {}^{1}/_{w_0R_iY_{ii}}) .$$

We can see that normalizing A is useless and, although only a sufficient condition for convergence, that the values of (4.2.3) must be drastically changed to make this quantity less than 1.

4.3 CONCLUSIONS

The Lyapunov equation is both theoretically interesting and practically useful. Although commonly associated with stability theory, the various physical interpretations of its solution and relationship to the evaluation of quadratic integrals make it a basic tool in a number of areas of control theory. Many different methods can be used to solve it and a number of these were discussed in Chapter Two. The iterative decoupling algorithm developed in that chapter is basically an original contribution of this thesis, although the idea upon which it is based is not new. It is a special purpose solution method with several desirable properties that requires more development in order to assess its real potential. A suggestion for future work here is to extend the algorithm to the over-relaxation scheme, as this would add flexibility to the method.

The error analyses of Chapter Three are based heavily on the works of others, primarily Wilkinson, although several of the results are original in their specific application and extension to the Lyapunov equation. One important conclusion here is that the bounding approach is primarily useful for comparing different algorithms and the results obtained should not be interpreted too literally. This is especially true regarding a priori accuracy estimates. There is little doubt that theoretical analysis and numerical experience are both necessary in order to perform useful error analyses. For this reason, an obvious

suggestion for future research would be a systematic, well-organized set of numerical experiments designed to correlate and refine some of the bounds obtained here.

APPENDIX A

The purpose of this appendix is to summarize a few properties of, and relationships between, the canonical forms used in the main text, with special emphasis on the existence conditions of the Companion canonical form. This material is standard and may be found in most linear algebra texts, and for this reason, the account is brief and factual.

Let the matrix A be an element of $C^{n\times n}$. A scaler $\lambda\varepsilon C$ is called an eigenvalue of A if there exists a non-zero vector $x\varepsilon C^n$ such that $Ax = \lambda x$, and the vector x is called an eigenvector of A associated with the eigenvalue λ. The eigenvalues are the roots of the characteristic equation of A, which is a polynomial of degree n given by $\det(\lambda I-A) = 0$.

If the eigenvalues of A are distinct, then the n eigenvectors of A are linearly independent and form a basis for C^n. In this case, the matrix P, whose columns are the eigenvectors, induces a similarity transformation on C^n such that $P^{-1}AP = \text{diag}(\lambda_i)$.

Suppose that A has r distinct eigenvalues, $\lambda_1,\lambda_2, \ldots _r$ with multiplicities $m_1, m_2, \ldots, m_r$, where $\sum_{k=1}^{r} m_k = n$. Then the generalization of the diagonal form is the Jordon canonical form, i.e., there exists a matrix H such that $H^{-1}AH = J$, where J is in Jordon form and is the direct sum of p ($p\geq r$) Jordon sub-blocks $J_k(\lambda_i)$. As an example, for n=6, p=4, m_1=4, m_2=1, m_3=1:

$$J = \begin{bmatrix} J_2(\lambda_1) & & & & \\ & J_2(\lambda_1) & & & \\ & & J_1(\lambda_2) & & \\ & & & \lambda_1(\lambda_3) & \end{bmatrix} \quad . \tag{A.1}$$

The general form of a Jordon sub-block $J_k(\lambda_i)$ is

$$\begin{bmatrix} \lambda_i & 1 & & & & \\ & \lambda_i & 1 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \cdot & 1 \\ & & & & & \lambda_i \end{bmatrix} \quad (k \times k \text{ matrix})$$

Now, the number of independent eigenvectors of A is equal to the number of Jordon sub-blocks. For the above example, the eigenvalues of J are e_1, e_3, e_5, and e_6, while those of A are He_1, He_3, He_5, and He_6. Note that, apart from the ordering of the sub-blocks, the transformation is unique.

The elementary divisors of A are the p polynomials $\det(J_k(\lambda_i - \lambda))$, and the minimal polynomial of A is the product of those elementary divisors that correspond to the Jordon blocks of largest dimension of each distinct eigenvalue, i.e., for the above example the minimal polynomial is

$$m(\lambda) = \det(J_2(\lambda_1-\lambda))\det(J_1(\lambda_2-\lambda))\det(J_1(\lambda_3-\lambda)).$$

Now a matrix with distinct eigenvalues must necessarily have linear elementary divisors, while one with one or more non-linear elementary divisors is called defective. If there is more than one Jordon sub-block for any distinct eigenvalue, then the matrix is called derogatory, and in this case the degree of the minimal polynomial is less than n.

If the characteristic polynomial of A is

$$\det(\lambda I-A) = \lambda^n + a_n\lambda^{n-1} + a_{n-1}\lambda^{n-2} + \ldots + a_2\lambda + a_1,$$

then the Companion canonical form C_A is

$$\begin{bmatrix} 0 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & 1 & \\ & & & & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & & -a_2 & -a_1 \end{bmatrix} \qquad (A.2)$$

A matrix A is similar to a matrix in Companion form only if it is non-derogatory. Such a matrix is also said to be cyclic of period n. For those interested in control theory, an equivalent statement is that the matrix A is non-derogatory (and hence, a similar companion

form exists) iff there exists a vector b such that the pair (A, b) is completely controllable.

The generalization of the Companion form for derogatory matrices is the Frobenius (or Rational) canonical form, which is the direct sum of m sub-blocks of dimension n_i, i=1, 2, ..., m, and each sub-block is of the form of (A.2). For the example (A.1) m=2, n_1=4, and n_2=2. Any matrix A is similar to a matrix in Forbenius form, and in this case, m is the smallest integer such that there exists a B (n×m) such that the pair (A, B) is completely controllable.

APPENDIX B

Round-off errors in algebraic computations occur because real numbers must be represented with a finite number of digits or bits. Matrix computations, however complicated, are performed by a series of elementary algebraic operations. In this appendix, some basic results for the fundamental arithmetic operations as performed on a digital computer are given [18]. Only the case of floating point arithmetic is considered.

In floating point, the real number x is represented as

$$x = 2^{b}(a); \quad b \text{ integer}, \ -1 \leq a \leq -1/2 \text{ or } 1/2 \leq a \leq 1.$$

Consider the addition of two scalers, x_1 and x_2. Define:

$fl(x_1 + x_2)$	computed quantity
$x_1 + x_2$	exact quantity
t	number of digits assigned to mantissa.

In the bounding approach of error analysis [18], it is assumed that the rounding errors of the elementary operations are such that

$$fl(x_1 + x_2) \equiv (x_1 + x_2)(1 + e)$$

$$fl(x_1 x_2) \equiv x_1 x_2 (1 + e) \qquad |e| \leq 2^{-t}$$

$$fl(x_1/x_2) \equiv x_1/x_2 (1 + e)$$

Using these assumptions, similar results can be obtained which will be useful in later sections. In order to illustrate the procedure, consider the computation

$$s_n = fl(x_1 + x_2 + \ldots + x_n)$$

let $s_1 = fl(x_1)$

$$s_r = fl(s_{r-1} + x_r) = (s_{r-1} + x_r)(1 + e) \qquad r=2, 3, \ldots, n$$

then $$s_n = fl(x_1 + x_2 + \ldots x_n) = x_1(1+e) + x_2(1+e) + \ldots + x_n(1+e_n)$$

where $$(1 - t^{-t})^{n-r+1} \leq (1 + e_r) \leq (1 + 2^{-t})^{n-r+1} .$$

Now, a bound of the form $(1 - 2^{-t})^r \leq (1 + e) \leq (1 + 2^{-t})^r$ arises frequently and is somewhat inconvenient. With the very reasonable assumption that $r2^{-t} < 0.1$, it can be replaced with $|e| \leq r2^{-t}1$, where $2^{-t}1 = 1.06\ 2^{-t}$. So, for the above result, we have $|e_r| \leq (n-r+1)2^{-t}1$. Notice that the bound depends on the order of summation; the best procedure is to sum the smallest terms first.

In a similar manner we may obtain the following

$$p_n = fl(x_1x_2 \ldots x_n) = x_1x_2 \ldots x_n(1+e_2)(1+e_3) \ldots (1+e_n)$$

$$= x_1x_2 \ldots x_n(1+E) \qquad |E| < (n-1)2^{-t_1}$$

$$s_n = fl(x_1y_1+x_2y_2+ \ldots x_ny_n) = x_1y_1(1+e_1)+x_2y_2(1+e_2)+\ldots+x_ny_n(1+e_n)$$

$$|e_r| < (n-r+2)2^{-t_1}$$

The results for the extended sum and inner product assumed that the machine does not accumulate with a 2t-digit mantissa. As far as round-off errors are concerned, accumulation is definitely an advantage. In a machine with this feature, intermediate results in a series of elementary operations are <u>not</u> rounded to t-digits, i.e., the working registers that contain the intermediate results carry a 2t-digit mantissa. For a machine that accumulates the operations are denoted $fl_2(\cdot)$, and comparing the following bounds with those given previously illustrates the significance of accumulation. (Note that higher level languages, e.g., Fortran, on the IBM-370, do not have this capability.)

$$fl_2(x_1+x_2+\ldots+x_n) = [x_1(1+e_1)+x_2(1+e_2)+\ldots+x_n(1+e_n)](1+e)$$

$$|e| \leq 2^{-t} \qquad |e_r| < \frac{3}{2}(n+1-r)a^{-2t_2}$$

where $$2^{-2t_2} = 1.06\ 2^{-2t}$$

$$f\ell_2(x_1y_1+x_2y_2+\ldots+x_ny_n) = [x_1y_1(1+e_1)+x_2y_2(1+e_2)+\ldots+x_ny_n(1+e_n)](1+e)$$

$$|e| \leq 2^{-t} \qquad |e_r| < \frac{3}{2}(n+2-r)2^{-2t_2}$$

Some other useful results:

$$B = f\ell(kA) \qquad \|B - kA\|_E \leq |k|2^{-t}\|A\|_E \qquad k\text{-scaler}$$

$$y = f\ell(Ax) = Ax + e \qquad \|e_2\| \leq 2^{-t}n\|A\|_E\|x\|_2$$

$$C = f\ell(AB) = AB + E \qquad \|E\|_E \leq 2^{-t}n\|A\|_E\|B\|_E$$

$$C = f\ell_2(AB) = AB + E \qquad \|E\|_E \leq 2^{-t}\|AB\|_E + \frac{3}{2}n\|A\|_E\|B\|_E$$

REFERENCES

1. P. Hagander, "Numerical Solution of $A^TS + SA + Q = 0$," Information Sciences, 4, 1972.

2. R.H. Bartels and G.W. Stewart, "Solution of the Matrix Equation $AX + XB = C$," Communications of the ACM, Vol. 15, No. 9, September, 1972.

3. P.C. Muller, "Solution of the Matrix Equations $AX + XB = -Q$ and $S^TX + XS = -Q$," SIAMJ Appl. Math, Vol. 18, No. 3, May, 1970.

4. B.P. Molinari, "Algebraic solution of Matrix Linear Equations in Control Theory," Proc. IEEE, Vol. 116, No. 10, October, 1969.

5. E.J. Davison and F.T. Man, "The Numerical Solution of $A'Q + QA = -C$," IEEE Trans. AC-13, August, 1968.

6. P.G. Smith, "Numerical Solution of the Matrix Equation $AX + XA^T + B = 0$," IEEE Trans. AC-16, June, 1971.

7. C.F. Chen and L.S. Shieh, "A Note on Expanding $PA + A^TP = -Q$," IEEE Trans. AC-13, February, 1968.

8. S.P. Bingulac, "An Alternative Approach to Expanding $PA + A^TP = -Q$," IEEE Trans. AC-15, February, 1970.

9. D.L. Kleinman, "On An Iterative Technique for Riccati Equation Computations," IEEE Trans. AC-13, February, 1968.

10. N.R. Sandell, Jr., P. Varaiya, and M. Athans, "A Survey of Decentralized Methods for Large Scale Systems," to be published.

11. N.R. Sandell, Jr., "On Newton's Method for Riccati Equation Solution," IEEE Trans. AC-19, June, 1974.

12. D. Looze, Decentralized Control of a Freeway Traffic Corridor, M.S. Thesis, M.I.T., August, 1975.

13. P.V. Kokotovic, "Feedback Design of Large Linear Systems," in Feedback Systems, edited by J.B. Cruz, Jr., McGraw-Hill, N.Y., 1972.

14. N.R. Sandell, Jr., and M. Athans, "A Think Piece for Three-Year Research Plan for the Development of Decentralized Strategies for the Control of Interconnected Power Systems," Electronic Systems Laboratory Report, M.I.T., January, 1975.

15. M. Athans, "The Role and Use of the Stochastic Linear - Quadratic - Gaussian Problem in Control System Design," IEEE Trans. AC-16, December, 1971.

16. R.A. Smith, "Matrix Calculations for Lyapunov Quadratic Forms," J. Differential Equations, 2, April, 1966, pp. 208-17.

17. Rall, Louis B. (Ed.), Error in Digital Computation, Vol, I. John Wiley & Sons, 1965.

18. Wilkinson, J.H., Rounding Errors in Algebraic Processes. Prentice-Hall, 1963.

19. Bellman, Richard, Introduction to Matrix Analysis. McGraw-Hill, 1960.

20. Schneider, Hans (Ed.), Recent Advances in Matrix Theory. University of Wisconsin Press.

21. Forsythe, George and Moler, Cleve, Computer Solution of Linear Algebraic Systems. Prentice-Hall, 1967.

22. Householder, A.S., The Theory of Matrices in Numerical Analysis. Blaisdell Publishing Company, 1965.

23. Varga, R.S., Matrix Iterative Analysis. Prentice-Hall, 1962.

24. Wilkinson, J.H., The Algebraic Eigenvalue Problem. Clarendon Press, 1965.

25. Brockett, Roger W., Finite Dimensional Linear Systems. John Wiley and Sons, Inc., 1970.

26. Byerly, R.T. and Kimbark, E.W. (Eds.), Stability of Large Electric Power Systems. IEEE Press, New York, 1974.

27. Elgerd, O.I., Electric Energy Systems Theory: An Introduction. McGram-Hill.

28. Selby, S.M. (Ed.), Standard Mathematical Tables. CRC Press, Inc., 1974.

29. Laub, Alan J., Decentralization of Optimization Problems by Iterative Coordination. Center for Comparative Studies in Technological Development and Social Change, University of Minnesota, 1974.

30. Barnett, S. and Storey C., Matrix Methods in Stability Theory. Thomas Nelson & Sons, 1970.

31. Faddeeva, V.N., Computational Methods of Linear Algebra. Dover Publications, Inc., 1959.

32. Astrom, K.J., Introduction to Stochastic Control Theory. Academic Press, 1970.

33. Gastinel, Noel, Linear Numerical Analysis. Academic Press, 1970.

34. Oppenheim, Alan V., Papers on Digital Signal Processing. M.I.T. Press, 1973.

35. Wehl, H., "Inequalities Between Two Kinds of Eigenvalues of a Linear Transformation," Proc. Nat. Acad. Sci., Vol. 35, pp. 408-11, 1949.

36. Von Neumann, J. and Goldstine, H.H., "Numerical Inverting of Matrices of High Order," Bull. Amer. Math. Soc., Vol. 53, pp. 1021-1099, 1947.

37. Fan, K., "On a Theorem of Wehl," Proc. Nat. Acad. Sci., Vol. 36, pp. 760-66, 1951.

38. Marcus, M., "Remark on a Norm Inequality for Square Matrices," Proc. Amer. Math. Soc., Vol. 6, pp. 117-19, 1955.

39. Kato, T., "·", Journal of Phys. Soc. of Japan, Vol. 4, pp. 334-339, 1949.

40. Jameson, A., "Solution of the Equation AX + XB = C by the Inversion of an M x M or N x N Matrix," SIAMJ Appl. Math., Vol. 16, pp. 1020-1023, 1968.

41. Lee, Stephen, Transient Stability Equivalents for Power System Planning, EPSEL Report No. 38, M.I.T., June, 1972.

42. deMello, R., Podmore, R., Stanton, K.N., Coherency Based Dynamic Equivalents for Transient Stability Studies, Electric Power Research Institute, Final Report EPRI 904, January, 1975.