On the Conditioning of the Nonsymmetric Eigenproblem: Theory and Software

Z. Bai, J. Demmel and A. McKenney*
Courant Institute
251 Mercer Str.
New York, NY 10012

December 9 1981

Abstract

This report reviews the theory and practical estimation of condition numbers for the nonsymmetric eigenvalue problem. The report provides a manual for using LAPACK subroutines STRSNA and STRSEN to estimate condition numbers for individual eigenvalues and eigenvectors, multiple (or clustered) eigenvalues, and invariant subspaces.

*The authors acknowledge the financial support of NSF, grant ASC-8715728. The first author acknowledges DARPA, grant F49620-87-C0065. The second author is also a Presidential Young Investigator.
## Contents

1. Introduction .................................................. 3

2. Spectral Projectors and the Separation of Two Matrices .... 8

3. An Upper Bound on $\|E\|_F$ for Global Error Bounds .... 9

4. Conditioning of Eigenvalues .............................. 10
   4.1 Conditioning of Simple Eigenvalues .............. 10
   4.2 Conditioning of Clustered Eigenvalues .......... 11
   4.3 Stability ............................................. 12

5. Conditioning of Right Eigenvectors and Right Invariant Subspaces .......... 12
   5.1 Angles Between Subspaces ....................... 13
   5.2 Conditioning of Right Eigenvectors and Right Invariant Subspaces ............. 14
   5.3 (Block)diagonalizing a Matrix with a Similarity ............. 14

6. Summary: Perturbation Table ................................ 16

7. STRSNA – Estimating the Condition of Individual Eigenpairs .... 17
   7.1 Usage ............................................. 17
   7.2 Example .......................................... 21
   7.3 Outline of the Algorithm ......................... 22

8. STRSEN Estimating the Condition of a Cluster of Eigenvalues .... 25
   8.1 Usage ............................................. 25
   8.2 Example .......................................... 27
   8.3 Outline of the Algorithm ......................... 29

A. Solution of the Sylvester Equation ....................... 31

B. Swapping Diagonal Blocks ................................. 33

C. List of LAPACK Routines for the Nonsymmetric Eigenproblem .... 35
1 Introduction

We review the theory of condition numbers for the nonsymmetric eigenproblem, and describe algorithms for estimating them. We provide a manual for the LAPACK subroutines STRSNA and STRSEN, which compute these condition numbers for matrices in Schur canonical form. We assume the reader is familiar with the basic theory of the nonsymmetric eigenproblem: eigenvalues, right and left eigenvectors, multiple eigenvalues and right and left invariant subspaces.

The condition number of a problem measures the sensitivity of the solution to small changes in the input. We call the problem ill-conditioned if its condition number is large, and ill-posed if its condition number is infinite. We may use condition numbers to bound errors in computed solutions of numerical problems.

We illustrate this with a simple example. It is well known that the condition number for solving a system of linear equations is \( k(A) \equiv \|A\| \cdot \|A^{-1}\| \), where \( \| \cdot \| \) is any matrix operator norm (we will be more specific about norms later). Suppose that linear system \( Ax = b \) is solved via Gaussian elimination with partial pivoting, or some other stable scheme. Let \( \bar{x} \) be the computed solution. Then one may bound the error by:

\[
\frac{\|\bar{x} - x\|}{\|x\|} = O(macheps) \cdot k(A)
\]

where \( macheps \) is the machine precision. The size of the constant implicit in the \( O(\cdot) \) notation depends on the size of the matrix, pivot growth, etc.

Condition numbers may be expensive to compute exactly. For example, computing \( k(A) \) for even the simplest matrix norms is three times as expensive as solving \( Ax = b \) in the first place. Therefore, one usually uses an inexpensive estimate in place of the exact \( k(A) \). For example, a method for estimating \( k(A) \) is included in LINPACK, which costs just \( O(n^2) \) extra beyond the \( O(n^3) \) cost of solving \( Ax = b \). The price one pays for using an estimate is occasional (but hopefully rare) misestimates of \( k(A) \). Years of experience with the LINPACK estimator attest to its reliability, although examples do exist where it can underestimate \( k(A) \) badly.

The codes we discuss for the nonsymmetric eigenproblem will also use such condition estimators. Here, the savings will be even greater than for linear equation solving: an \( O(n^3) \) estimator using \( O(n) \) workspace in place of an \( O(n^6) \) exact solution using \( O(n^4) \) workspace in some cases.
rocal of a condition number for an eigenvector (or invariant subspace). We compute reciprocals of condition numbers to avoid overflow; an infinite or overflowed condition number is indicated by a zero reciprocal. By combining these two values in simple algebraic formulas, a great deal of detailed information about the eigenproblem can be obtained. This report will describe both these basic condition numbers and these formulas.

Our condition numbers will measure the changes in the eigenvalues, right eigenvectors, means of clusters of eigenvalues, and right invariant subspaces of a matrix $A$ when a perturbation $E$ is added to it; our bounds will be functions (usually multiples) of $\|E\|$. This may be used to estimate the error in solutions computed by LAPACK routines because they are backward stable, i.e. they compute the exact eigendecomposition of a matrix $A + E$ where $A$ is the input matrix, and $\|E\| = O(macheps)\|A\|$. We measure changes in eigenvectors and invariant subspaces by their change in angle; we discuss the angle between subspaces in more detail in section 5. Our condition numbers yield both asymptotic bounds, which are accurate only when the norm $\|E\|$ is small, and global bounds, which work for all $\|E\|$ up to a certain upper bound, whose size depends on the problem and may be large or small. We show how to obtain these upper bounds on $\|E\|$ as well.

We illustrate the reason for providing such a variety of bounds with an example. Let $A_\eta$ be 11 by 11 of the following form:

$$A_\eta = \begin{bmatrix} 0 & 1 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 1 \end{bmatrix}$$

Here, blank entries are also zero. Thus, $A_\eta$ is a block diagonal matrix with a 10 by 10 block at the upper left and a 1 by 1 block at the lower right. When $\eta = 0$, the upper left block is a 10 by 10 Jordan block with a single multiple eigenvalue at 0 and a single right eigenvector $v = [1, 0, \ldots, 0]^T$. Such a matrix is called defective. For small nonzero $\eta$ the eigenvalues become distinct numbers all with absolute value $\eta^1$, and eigenvectors which have rotated away from $v$ by about $\eta^1$ radians. When $\eta = 10^{-10}$, $\eta^1 = .1$, a much larger change. In this case we call the eigenvalue at 0 and its associated eigenvector ill-posed, because their sensitivity is not proportional to the norm of the perturbation $\eta$, but a root of $\eta$.

The practical solution to this problem is to consider this matrix as having a cluster of 10 eigenvalues near zero with a single invariant subspace which
is spanned by all their eigenvectors, as well as a single eigenvalue near .5 with its eigenvector. The mean of this cluster of 10 eigenvalues will be much less sensitive to small perturbations than the individual eigenvalues (in fact it will be independent of \( \eta \) in this example). For small enough \( \|E\| \), our asymptotic error bounds will show that the mean of the cluster of 10 eigenvalues near 0 of \( A_0 + E \) is bounded by \( \|E\| \) (see Bound 4 below); i.e. the mean is very well-conditioned. Similarly, the eigenvalue near .5 can also only change by \( \|E\| \) for small enough \( \|E\| \) (Bound 2). The invariant subspaces will also be much less sensitive than the individual eigenvectors. In this example, the right invariant subspace belonging to the cluster of 10 eigenvalues near 0 is spanned by the first 10 columns of the 11 by 11 identity matrix independent of \( \eta \); more generally our bounds will say that for small \( \|E\| \) the right invariant subspace can rotate by at most 2731\( \|E\| \) radians (Bound 6). The eigenvector belonging to the eigenvalue .5 is equally insensitive in this example.

This illustrates our asymptotic error bounds, valid for sufficiently small \( \|E\| \). In contrast, our global bounds give bounds valid for all \( \|E\| \) up to an upper bound which we also estimate. The matrix \( A_0 \) illustrates the source of these upper bounds on \( \|E\| \). Suppose we make \( \eta = 2^{-10} \approx .001 \); then one of the eigenvalues originally at 0 now equals .5, the same as the eigenvalue in the lower right corner. Thus, we can no longer say that this matrix has a cluster of eigenvalues near 0 and one near .5, and so we can no longer talk about the sensitivity of the mean of the cluster. We can also no longer identify a unique eigenvector associated with an eigenvalue near .5; the eigenvectors have become ill-posed. Indeed, with additional arbitrarily small perturbations the two eigenvectors for the eigenvalues at .5 can be made to rotate arbitrarily within a two dimensional subspace, or one of them can even disappear. Thus, only if we bound \( \|E\| \) to be some value less than \( 2^{-10} \) can we hope to have error bounds. For this example, the upper bound computed by our software will be approximately \( 2 \cdot 10^{-4} \) (Bound 1). For \( \|E\| < 2 \cdot 10^{-4} \), our upper bound on the change in the mean of the eigenvalue cluster will be \( 2\|E\| \) (Bound 5), and our bound on how much the right invariant subspace can rotate will be \( \arctan(2731\|E\|/(1 - 5462\|E\|)) \) radians (Bound 7), both close to the asymptotic bounds.

In this example, it is easy to identify the clusters by inspection of \( A_0 \). This is not always the case in practice, when the user is confronted with a matrix whose eigenvalues form a cloud rather than well separated clusters. Unfortunately, there is as yet no reliable, automated procedure for clustering eigenvalues; see [8, 9, 28] for discussion. Our software merely provides the tools for evaluating a particular clustering. A good cluster will have a much
less sensitive mean and invariant subspace than any subcluster, and must be
made part of a much larger (or trivial) cluster before it becomes significantly
less sensitive. The 10 zero eigenvalues of $A_0$ satisfy this criterion.

There is a very large literature on perturbation theory for the eigen-
problem. See [3, 8, 9, 11, 15, 16, 22, 24, 25, 27, 28] for various theoretical
bounds. Chan, Feldman and Parlett[6] provided a Fortran routine to com-
pute the condition number of simple eigenvalue in conjunction with EIS-
PACK routines orthes and hqr, but it does not provide any information
about conditioning for eigenvectors and subspaces. Ruhe[20] suggested us-
ing the Golub-Reinsch SVD algorithm to calculate the condition number for
eigenvectors, but this requires $O(n^3)$ flops per eigenvalue-eigenvector pair,
which is too expensive. Van Loan[26] developed an efficient algorithm for
estimating condition numbers of all eigenvalue-eigenvector pairs of a Hes-
 senberg matrix. It only costs $O(n^2)$ flops per eigenpair, assuming that the
eigenvalues are known.

We have developed new algorithms, which assume the matrix has been
reduced to Schur canonical form (real or complex). Reduction to Schur
canonical form is done by LAPACK subroutines sgehrd and shseqr in the
real case, and cgehrd and chseqr in the complex case. Since this reduction
is done via orthogonal (or unitary) similarities, the condition numbers are
identical to those of the original matrix. As we will see, starting with the
matrix in Schur form simplifies many of the algorithms and lets us use
existing condition estimation software for (quasi)triangular matrices [14, 18,
19].

The rest of this report is organized as follows. Section 2 discusses spectral
projectors and the separation of matrices, quantities on which later bounds
are based. Section 3 discusses the upper bound on $\|E\|_F$ for our global error
bounds. Section 4 discusses asymptotic and global bounds for eigenvalues
and means of clusters of eigenvalues. In section 5, we first define the angle be-
tween two subspaces, the quantity bounded by our error bounds. Second, we
present asymptotic and global perturbation bounds for both right eigenvec-
tors and right invariant subspaces. Third, we discuss (block)diagonalizing a
matrix by a similarity. The results in sections 2 through 5 are stated without
proof; references to proofs in the literature are given. A tabular summary of
all bounds is given in section 6. Sections 7 and 8 describe the usage of the
LAPACK routines strsna and strsen for estimating the desired condition
numbers (actually their reciprocals). STRSNA computes the reciprocal condition
numbers of user-specified eigenvalues and/or eigenvectors of the input
matrix. STRSEN computes the reciprocal condition numbers of the mean
and/or invariant subspace of a single user-specified cluster of eigenvalues.
Two examples are provided to show how to use these codes. Outlines of the algorithms are also given.

The first two appendices describe details of the solution of the Sylvester matrix equation and swapping diagonal blocks of a quasitriangular matrix. The third appendix lists the names and basic functions of LAPACK routines needed for the nonsymmetric eigenvalue problem.

We end with some notation we will need later. Capital letters are used to denote matrices, the corresponding lowercase letter with the subscript \( i,j \) referring to the \((i,j)\) component (e.g., \( a_{ij} \) is the \((i,j)\) component of \( A \)). A submatrix of a matrix \( A \) is written as \( A_{ij} \). Vectors are also denoted by lowercase letters and will be clearly indicated in the text. Lowercase Greek letters will denote scalars.

\[ \| x \|_1, \| x \|_2 \text{ and } \| x \|_{\infty} \text{ denote the one-norm, the Euclidean norm, and the infinity-norm, respectively, of the } n \text{-vector } x: \]

\[ \| x \|_1 = \sum_{i=1}^{n} |x_i|, \quad \| x \|_2 = \left( \sum_{i=1}^{n} |x_i|^2 \right)^{1/2}, \quad \| x \|_{\infty} = \max_{1 \leq i \leq n} |x_i|. \]

\( \| T \|_1, \| T \|_2, \| T \|_{\infty}, \| T \|_F \) denote the matrix norms:

\[ \| T \|_1 = \max_j \sum_i |t_{ij}|, \quad \| T \|_2 = \sup_{x \neq 0} \frac{\|Tx\|_2}{\|x\|_2}, \quad \| T \|_{\infty} = \max_i \sum_j |t_{ij}|, \quad \| T \|_F = \left( \sum_{i,j=1}^{n} |t_{ij}|^2 \right)^{1/2}. \]

Note that \( \| \|_2 \) and \( \| \|_F \) are invariant with respect to unitary transformation.

We will throughout let \( \epsilon_2 \) denote \( \| E \|_2 \), and \( \epsilon_F \) denote \( \| E \|_F \), the norms of our perturbation matrix.

The condition number of \( T \) is \( \kappa(T) = \| T \|_2 \| T^{-1} \|_2 \). A subspace spanned by the columns of matrix \( A \) is denoted as \( R(A) \) (the range of matrix \( A \)). \( \lambda(A) \) denotes the set of all eigenvalues of matrix \( A \). \( A \otimes B \) denotes the Kronecker product of two matrices: \( A \otimes B = (a_{ij}B) \).

The Schur matrix (or Schur form) of a real matrix is an orthogonally similar upper quasi-triangular matrix whose 2 by 2 diagonal blocks (if any exist) are of the form

\[
\begin{pmatrix}
\alpha & \beta \\
\gamma & \alpha
\end{pmatrix}
\]
Such a block has complex conjugate eigenvalues $\alpha \pm \mu$ where $\mu^2 = -\beta \gamma$. The Schur form of a complex matrix is a unitarily similar upper triangular matrix.

2 Spectral Projectors and the Separation of Two Matrices

To explain the bounds in later sections, we need to introduce two quantities, the spectral projector $P$ [22, 15], and the separation of two matrices $A$ and $B$, $\text{sep}(A, B)$ [22].

Suppose our cluster contains $m \geq 1$ eigenvalues, counting multiplicities. Assume the $n$ by $n$ matrix $A$ is in Schur canonical form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

(1)

where the eigenvalues of the $m$ by $m$ matrix $A_{11}$ are exactly those in which we are interested. In practice, if the eigenvalues on the diagonal of $A$ are in the wrong order, they are sorted to put the desired ones in the upper left corner as shown by using the subroutine \texttt{STREXC} in Appendix B.

We define the spectral projector, or simply projector $P$ belonging to the eigenvalues of $A_{11}$ as

$$P = \begin{bmatrix} I_m & R \\ 0 & 0 \end{bmatrix}$$

(2)

where $R$ satisfies the system of linear equations

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

(3)

Equation (3) is called a Sylvester equation. Given the Schur canonical form (1), we solve the Sylvester (3) for $R$ using subroutine \texttt{STRSYL} in Appendix A.

$P$ has several important properties. First, the space spanned by its columns is the same as the right invariant subspace of $A$ belonging to $A_{11}$. Second, the space spanned by its rows is the same as the left invariant subspace of $A$ belonging to $A_{11}$. Thus, $P$ describes the spaces spanned by both the left and right eigenvectors belonging to $A_{11}$. Third, its norm $\|P\|_2 = (1 + \|R\|_2^2)^{1/2}$ plays an important role in our error bounds, as we will see.

In practice, we do not use $\|P\|_2$ for $m > 1$ since this is expensive to compute, but rather the cheaper overestimate

$$\|P\|' \equiv (1 + \|R\|_F^2)^{1/2}$$

(4)
The separation \( \text{sep}(A_{11}, A_{22}) \) of the matrices \( A_{11} \) and \( A_{22} \) is defined as the smallest singular value of the linear map in (3) which takes \( X \) to \( A_{11}X - XA_{22} \), i.e.

\[
\text{sep}(A_{11}, A_{22}) = \min_{X \neq 0} \frac{\|A_{11}X - XA_{22}\|_F}{\|X\|_F} \tag{5}
\]

This formulation lets us estimate \( \text{sep} \) using the condition estimator \texttt{SLACON} [14, 18, 19], which estimates the norm of a linear operator \( \|T\|_1 \) given the ability to compute \( Tx \) and \( T^Ty \) quickly for arbitrary \( x \) and \( y \). In our case, multiplying an arbitrary vector by \( T \) means solving the Sylvester equation (3) with an arbitrary right hand side, and multiplying by \( T^T \) means solving the same equation with \( A_{11} \) replaced by \( A_{11}^T \) and \( A_{22} \) replaced by \( A_{22}^T \). Solving either equation costs at most \( O(n^3) \) operations, or as few as \( O(n^2) \) if \( m \ll n \).

Another formulation which in principle permits an exact evaluation of \( \text{sep}(A_{11}, A_{22}) \) is

\[
\text{sep}(A_{11}, A_{22}) = \sigma_{\min}(I_{n-m} \otimes A_{11} - A_{22}^T \otimes I_m) \tag{6}
\]

where \( \sigma_{\min} \) is the smallest singular value. This method is generally impractical, however, because the matrix whose smallest singular value we need is \( m(n - m) \) dimensional, which can be as large as \( n^2/4 \). Thus we would require as much as \( O(n^4) \) extra workspace and \( O(n^6) \) operations, much more than the estimation method of the last paragraph.

\( \text{sep}(A_{11}, A_{22}) \) measures the “separation” of the spectrum of \( A_{11} \) and \( A_{22} \) in the following sense. It is zero if and only if \( A_{11} \) and \( A_{22} \) have a common eigenvalue, and small if there is a small perturbation of either one that makes them have a common eigenvalue. If \( A_{11} \) and \( A_{22} \) are both normal matrices, then \( \text{sep}(A_{11}, A_{22}) \) is just the minimum distance between an eigenvalue of \( A_{11} \) and an eigenvalue of \( A_{22} \).

\texttt{STRSNA} computes \( 1/\|P\|_2 \) (which is always \( \leq 1 \), avoiding the possibility of overflow) and \( \text{sep} \) for user-selected individual eigenvalues (i.e. \( A_{11} \) is 1 by 1). \texttt{STRSEN} computes \( 1/\|P\|' \) and \( \text{sep} \) for a single user-specified cluster with \( m \geq 1 \) eigenvalues.

### 3 An Upper Bound on \( \|E\|_F \) for Global Error Bounds

We discuss the upper bound on \( \|E\|_F \) which limits the applicability of our global bounds in the next two sections. As stated in the introduction, this upper bound occurs because if \( \|E\|_F \) is large enough that the eigenvalue
being considered (or one of the eigenvalues in the cluster being considered) moves and coalesces with another eigenvalue (outside the cluster), then we can no longer uniquely identify the cluster for which we want bounds. Thus, in this section we present lower bounds on the smallest $\|E\|_F$ such that $A + E$ has a multiple eigenvalue (or a multiple eigenvalue involving at least one eigenvalue within the original cluster and one outside).

**Bound 1:** [22, Theorem 4.14] Let $A$, $P$ and $\text{sep}(A_{11},A_{22})$ be defined as in (1), (2) and (5). Then as long as

$$\|E\|_F < \frac{\text{sep}(A_{11},A_{22})}{4 \cdot \|P\|_2} \quad (7)$$

the eigenvalues in the cluster belonging to $A_{11}$ will remain disjoint from the eigenvalues outside the cluster. In particular, the global error bounds of sections 4 and 5 will be guaranteed valid only for $E$ satisfying (7). We may replace $\|P\|_2$ by $\|P\|'$ as defined in (4) to get a slightly smaller upper bound.

Bound 1 can be quite conservative, greatly underestimating the smallest perturbation needed to make eigenvalues coalesce. However, it is nearly exact in some cases (e.g. for 2 by 2 matrices and normal matrices), and a good estimate in many others; see [8, 9] for discussion.

$1/\|P\|'$ (or $1/\|P\|_2$ if $m = 1$) and $\text{sep}(A_{11},A_{22})$ are computed by STRSNA and STRSENN as described in section 2.

## 4 Conditioning of Eigenvalues

In this section, we review how to measure the sensitivity of both simple eigenvalues and clusters of eigenvalues.

### 4.1 Conditioning of Simple Eigenvalues

Let $\lambda$ be a simple eigenvalue of the $n$ by $n$ matrix $A$, with unit right eigenvector $x$ and unit left eigenvector $y$. In other words $Ax = \lambda x$, $y^T A = \lambda y^T$, and $\|x\|_2 = \|y\|_2 = 1$. Let $P$ be the spectral projector for $\lambda$; one may write $P = (x \cdot y^T)/(y^T \cdot x)$. Note that $\|P\|_2 = 1/|y^T x|$, the secant of the angle between $x$ and $y$.

Let $E$ be a perturbation of $A$, and $\epsilon_2 = \|E\|_2$. Let $\lambda'$ be the perturbed eigenvalue of $A + E$. 

10
**Bound 2:** [27, p. 68]

\[ |\lambda' - \lambda| \leq \epsilon_2 \|P\|_2 + O(\epsilon_2^2) \]

The \(O(\epsilon_2^2)\) term indicates this is an asymptotic bound, applicable only for sufficiently small \(\epsilon_2\). This bound is attainable, in the sense that for \(\epsilon_2\) sufficiently small, there exists an \(E\) such that \(|\lambda' - \lambda| = \epsilon_2 \|P\|_2 + O(\epsilon_2^2)\).

There is also a global version of this bound:

**Bound 3:** [3] Suppose \(A\) has all simple eigenvalues \(\lambda_i\). Let \(P_i\) be the corresponding spectral projectors. Then any eigenvalue \(\lambda'\) of \(A + E\) must lie in one of the disks

\[ \{ \lambda : |\lambda - \lambda_i| \leq n\epsilon_2 \|P_i\|_2 \} \]

There is no limit on the size of \(\epsilon_2\) in Bound 3. Note that the sizes of the disks are just \(n\) times larger than in Bound 1, where \(\epsilon_2\) must be small. Bound 3 is an stronger version of what is often called the Bauer-Fike Theorem, although Bauer and Fike proved this stronger version as well. In the weaker Bauer-Fike Theorem, all of the disks have the same radius, approximately equal to the largest radius \(\max_i n\epsilon_2 \|P_i\|_2\) in Bound 3. Note that Bound 3 is only useful when all the radii \(n\epsilon_2 \|P_i\|_2\) are of modest size, since if one or more disks is so large that it intersects all the other disks, there is little information about locations of individual eigenvalues; we only know they lie in the union of all the disks.

The subroutine **STRSNA** can compute \(1/\|P\|_2\) for a user-specified subset of the eigenvalues of \(A\).

### 4.2 Conditioning of Clustered Eigenvalues

It is easiest to think of \(A\) in Schur form (1), with the eigenvalues of \(A_{11}\) being the cluster of interest. We are interested in bounding the perturbation in the average of the eigenvalues of the cluster, which may be written \(A_{11}/m\), the trace of \(A_{11}\) divided by \(m\). Let \(E\) be a perturbation of \(A\), and \(\epsilon_2 = \|E\|_2\).

Let \(\bar{\lambda} = A_{11}/m\) be the mean of the unperturbed eigenvalues, and \(\bar{\lambda}'\) be the mean of the perturbed eigenvalues.

**Bound 4:** [15, sec. II.2.2]

\[ |\bar{\lambda} - \bar{\lambda}'| \leq \epsilon_2 \|P\|_2 + O(\epsilon_2^2) \]
We may substitute $\|P\|'$ of equation (4) for $\|P\|_2$ to get a slightly weaker bound.

The $O(\epsilon_2^2)$ indicates this is an asymptotic bound, applicable only for sufficiently small $\epsilon_2$. This bound is nearly attainable, in the sense that for $\epsilon_2$ sufficiently small, there exists an $E$ such that $|\bar{\lambda} - \bar{\lambda}'| \leq \|P\|_2 \epsilon_2/m + O(\epsilon_2^2)$. When $m = 1$, it is of course identical to the bound in the previous subsection.

Our global bound on $|\bar{\lambda} - \bar{\lambda}'|$ will only be valid for $\|E\|_F$ satisfying Bound 1 of section 2:

**Bound 5:** [22, page 748] Suppose $\|E\|_F$ satisfies Bound 1. Then

$$|\bar{\lambda} - \bar{\lambda}'| \leq 2\epsilon_2 \|P\|_2$$

Thus, the global bound is just twice as large as the asymptotic bound. Again we may substitute $\|P\|'$ of equation (4) for $\|P\|_2$ to get a slightly weaker bound.

**STRSNA** computes $1/\|P\|_2$ for a user-specified set of individual eigenvalues. **STRSE** can compute $1/\|P\|'$ for a single user-specified cluster of $m \geq 1$ eigenvalues.

### 4.3 Stability

When the eigenvalues of a full matrix $A$ have been found from its computed Schur form $\tilde{A}$, the computed $\tilde{s}$ will be those appropriate to $\tilde{A}$. These $\tilde{s}$ can differ substantially from the true $s$. Indeed, when $A$ is defective, $\tilde{A}$ will usually not be, and hence zero $s$ will become nonzero $\tilde{s}$. The reverse situation could occur though this is much less probable. Since some of the $\tilde{s}$ may not even be the correct order of magnitude, it might be felt that our heavy reliance on them is unjustified. Since our algorithms for computing the Schur form and $s(\lambda)$ are backward stable, $s(\lambda)$ is the correct value for a matrix $A + E$ very close to the original matrix $A$. This justifies their use. The same comments apply to the computation and use of sep described in the next section.

### 5 Conditioning of Right Eigenvectors and Right Invariant Subspaces

In this section, we review how to measure the sensitivity of eigenvectors and invariant subspaces. We begin by defining of the angle between two
subspaces, and then use it to describe the conditioning of eigenvectors and invariant subspaces.

5.1 Angles Between Subspaces

Let $\theta(x, y)$ denote the acute angle between two nonzero $n$-vectors $x$ and $y$. We may write $\cos \theta(x, y) = \frac{|x^T y|}{\|x\|_2 \|y\|_2}$. We wish to generalize this to the (maximum) angle between two $m > 1$ dimensional subspaces, which we denote $\mathcal{X}$ and $\mathcal{Y}$. One way to define this angle is as

$$\theta_{\text{max}}(\mathcal{X}, \mathcal{Y}) = \max_{x \not= 0} \min_{y \not= 0} \theta(x, y) (= \max_{y \not= 0} \min_{x \not= 0} \theta(x, y))$$

A more computational definition of $\theta_{\text{max}}(\mathcal{X}, \mathcal{Y})$ is the following. Suppose $\mathcal{X}$ is spanned by the columns of the $n$ by $m$ orthonormal matrix $X$, and similarly $\mathcal{Y}$ is spanned by the columns of the $n$ by $m$ orthonormal matrix $Y$. Then

$$\theta_{\text{max}}(\mathcal{X}, \mathcal{Y}) = \arccos \sigma_{\text{min}}(Y^T X)$$

Our bounds of the next two sections will bound $\theta_{\text{max}}(\mathcal{X}, \mathcal{X}')$ where $\mathcal{X}$ is an unperturbed invariant subspace, and $\mathcal{X}'$ is a perturbed invariant subspace.

We may also define the minimum angle between $\mathcal{X}$ and $\mathcal{Y}$ as

$$\theta_{\text{min}}(\mathcal{X}, \mathcal{Y}) = \min_{x \not= 0} \min_{y \not= 0} \theta(x, y) (= \min_{y \not= 0} \min_{x \not= 0} \theta(x, y))$$

This may be reexpressed computationally as

$$\theta_{\text{min}}(\mathcal{X}, \mathcal{Y}) = \arccos \sigma_{\text{max}}(Y^T X)$$

The norms of the spectral projectors $\|P\|_2$ introduced in section 2 have a simple interpretation in terms of angles between subspaces. Let $P$ be the spectral projector for the eigenvalue cluster with right invariant subspace $\mathcal{R}$ and left invariant subspace $\mathcal{L}$. Let $\mathcal{R}_c$ be the complementary right invariant subspace (the subspace for the other eigenvalues) and $\mathcal{L}_c$ be the complementary left invariant subspace. Then

$$\|P\|_2 = \csc \theta_{\text{min}}(\mathcal{R}, \mathcal{R}_c) = \csc \theta_{\text{min}}(\mathcal{L}, \mathcal{L}_c)$$

$$\|P\|_2 = \sec \theta_{\text{max}}(\mathcal{R}, \mathcal{L}) = \sec \theta_{\text{max}}(\mathcal{R}_c, \mathcal{L}_c)$$

In other words, as $\|P\|_2$ grows and the cluster becomes more ill-conditioned, the minimum angle between complementary right (or complementary left) subspaces shrinks. Also, the maximum angle between corresponding left and right invariant subspaces grows.
5.2 Conditioning of Right Eigenvectors and Right Invariant Subspaces

We assume $A$ is in Schur canonical form (1), with the eigenvalues of $A_{11}$ being the cluster whose right invariant subspace $R$ we are interested in. Let $E$ be a perturbation of $A$, and $\epsilon_F = \|E\|_F$. Let $R'$ be the perturbed right invariant subspace of $A + E$.

**Bound 6:** [8, Lemma 7.8]

$$\theta_{\max}(R, R') \leq \frac{2\epsilon_F}{\text{sep}(A_{11}, A_{22})} + O(\epsilon_F^2)$$

The $O(\epsilon_F^2)$ indicates that this is an asymptotic bound, applicable only for sufficiently small $\epsilon_F$. It is nearly attainable for sufficiently small $\epsilon_F$.

**Bound 7:** [8, Lemma 7.8] Suppose $\|E\|_F$ satisfies Bound 1. Then

$$\theta_{\max}(R, R') \leq \arctan\left(\frac{2\epsilon_F}{\text{sep}(A_{11}, A_{22}) - 4\|P\|_2^2\epsilon_F}\right)$$

$\|P\|_2$ may be replaced by $\|P\|'$ of equation (4) to obtain a slightly weaker bound.

Bounds 6 and 7 imply that sep is the reciprocal of the condition number for eigenvectors and invariant subspaces. Routines STRSNA and STRSEN compute sep for individual eigenvectors and a given invariant subspace, respectively.

5.3 (Block)diagonalizing a Matrix with a Similarity

Occasionally one wishes to find a matrix $V$ which diagonalizes $A$: $V^{-1}AV = \Lambda$, where $\Lambda$ is a diagonal matrix with the eigenvalues on the diagonal. This may be useful for computing functions of matrices. For example, to exponentiate a matrix one may use the identity $\exp(A) = V \exp(\Lambda)V^{-1} = V \text{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n})V^{-1}$. The accuracy of such an algorithm depends on the condition number $\kappa(V)$ of $V$. The columns of $V$ must be eigenvectors of $A$, but their norms are arbitrary; we would like to choose these norms to minimize $\kappa(V)$. The next bound gives a nearly optimal choice of the norms of the columns of $V$, and bounds the resulting $\kappa(V)$.  

14
Bound 8: [7] Suppose $A$ has distinct eigenvalues $\lambda_i$ with corresponding right eigenvectors $v_i$, where we assume $\|v_i\|_2 = 1$, and projectors $P_i$. Let $V = [v_1, \ldots, v_n]$ be the matrix of these eigenvectors. Let $V' = [\alpha_1 v_1, \ldots, \alpha_n v_n]$ be another matrix where the $\alpha_i$ are arbitrary nonzero scalars. Then

$$\max_i \|P_i\|_2 \leq \kappa(V')$$

Also

$$\max_i \|P_i\|_2 \leq \kappa(V) \leq n \cdot \max_i \|P_i\|_2$$

In other words choosing the columns of $V$ to have norm 1 nearly minimizes $\kappa(V)$ over all matrices whose columns are eigenvectors.

A variation on diagonalization is block-diagonalization, where we ask only that $V^{-1}AV = B$ be block diagonal, where the diagonal blocks $B_{ii}$ of $B$ have specified eigenvalues (which are all disjoint subsets of the eigenvalues of $A$). Suppose $B_{ii}$ is located in rows and columns $j$ through $k$ of $B$. Then columns $j$ through $k$ of $V$ must span the right invariant subspace of $A$ corresponding to the eigenvalues of $B$. Let $V_i$ denote these columns of $V$. Just as we could choose the norm of each column of $V$ when we wanted to diagonalize $A$, here we have the freedom to choose $V_i$ to be any basis of the right invariant subspace we like. Again, we would like to choose the basis which minimizes $\kappa(V)$. The next bound says how to do this.

Bound 9: [7] Let the set $\lambda(A)$ of eigenvalues of $A$ be written as a disjoint union $\bigcup_{i=1}^b S_i$ of $b$ sets of eigenvalues $S_i$. Let $n_i$ be the number of eigenvalues in $S_i$, counting multiplicities. Let $P_i$ be the projector corresponding to $S_i$, and $R_i$ the corresponding right invariant subspace. Let $V_i$ be any matrix whose $n_i$ columns form an orthonormal basis of $R_i$, and write $V = [V_1, \ldots, V_b]$. Then $V^{-1}AV = B$ is block diagonal where diagonal block $B_{ii}$ has eigenvalues $S_i$. Let $V'_i$ be an arbitrary matrix whose $n_i$ columns form a basis of $R_i$, and write $V' = [V'_1, \ldots, V'_b]$. $V'^{-1}AV' = B'$ is also block diagonal where diagonal block $B'_{ii}$ has eigenvalues $S_i$. Then

$$\max_i \|P_i\|_2 \leq \kappa(V')$$

Also

$$\max_i \|P_i\|_2 \leq \kappa(V) \leq b \cdot \max_i \|P_i\|_2$$

In other words choosing the columns of $V_i$ which span $R_i$ to be orthonormal nearly minimizes $\kappa(V)$ over all block-diagonalizing similarities which reduce $A$ to diagonal blocks with the same eigenvalues in each block.
6 Summary: Perturbation Table

For convenience, the bounds presented in the preceding sections are summarized in the following table. The notation is as follows. We assume the matrix $A$ is in Schur canonical form (1). $P$ denotes the spectral projector associated with with eigenvalue(s) of $A_{11}$ defined in (2). sep will be shorthand for $\text{sep}(A_{11}, A_{22})$, defined in (5). $\lambda$ will denote the unperturbed eigenvalue if $A_{11}$ is 1 by 1, and if $A_{11}$ is larger $\bar{\lambda}$ will denote the unperturbed mean of its eigenvalues. $\lambda'$ and $\bar{\lambda}'$ will denote the perturbed values of $\lambda$ and $\bar{\lambda}$, respectively, for $A + E$. $\mathcal{R}$ denotes the unperturbed right invariant subspace corresponding to $A_{11}$, and $\mathcal{R}'$ denotes its perturbed counterpart of $A + E$. $\theta$ will denote $\theta_{\max}(\mathcal{R}, \mathcal{R}')$, the angular perturbation of the right invariant subspace as defined in (8) or (9). $\epsilon_2$ will denote $\|E\|_2$ and $\epsilon_F$ will denote $\|E\|_F$, norms of the perturbation $E$. In the table, each asymptotic bound has a $+O(\epsilon^2)$ term which is not written. Superscripts in parentheses on each bound indicate which Bound in the body of text they are. The superscript $\dagger$ indicates that the bound applies only if $\epsilon_F < \text{sep}/(4\|P\|_2)$ (Bound 1).

<table>
<thead>
<tr>
<th></th>
<th>Asymptotic Bounds</th>
<th>Global Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple eigenvalue</td>
<td>$</td>
<td>\lambda - \lambda'</td>
</tr>
<tr>
<td>Eigenvalue Cluster</td>
<td>$</td>
<td>\bar{\lambda} - \bar{\lambda}'</td>
</tr>
<tr>
<td>Invariant subspace</td>
<td>$\theta \leq \frac{2\epsilon_F}{\text{sep}}^{(6)}$</td>
<td>$\theta \leq \arctan\left(\frac{2\epsilon_F}{\text{sep}-4|P|_2\epsilon_F}\right)^{\dagger(7)}$</td>
</tr>
</tbody>
</table>

In addition, Bound 8 says that the nearly best conditioned matrix $V$ such that $V^{-1}AV$ is diagonal has as its columns the eigenvectors of $A$ all with norm equal to 1. The condition number $\kappa(V)$ of this $V$ satisfies $\max_i \|P_i\|_2 \leq \kappa(V) \geq n \cdot \max_i \|P_i\|_2$, where $P_i$ is the projector corresponding to eigenvalue $\lambda_i$.

Bound 9 describes a nearly best conditioned matrix $V$ such that $V^{-1}AV = B$ is block diagonal, such that the 6 diagonal blocks of $B$ have specified eigenvalues. This nearly best $V$ may be written $V = [V_1, \ldots, V_6]$ where $V_i$ is any
orthonormal basis of the right invariant subspace of $A$ corresponding to the
eigenvalues in the $i$-th diagonal block of $B$. The condition number $\kappa(V)$ of
$V$ satisfies $\max_i \|P_i\|_2 \leq \kappa(V) \leq b \cdot \max_i \|P_i\|_2$, where $P_i$ is the projector
corresponding to eigenvalues of the $i$-th diagonal block of $B$.

In summary, we see that all the condition numbers for the nonsymmetric
eigenproblem depend on the quantities $\|P\|_2$ and sep. The use LAPACK
subroutines STRSNA and STRSEN to estimate these quantities is discussed in
the following sections.

7  STRSNA – Estimating the Condition of Individual
   Eigenpairs

In this section, we show how to use LAPACK subroutine STRSNA for esti-
mating condition numbers of selected eigenvalues and eigenvectors. We also
describe the algorithms used. The variable SEP corresponds to the sep of the
preceding sections, and the variable S corresponds to $1/\|P\|_2$. We return
$1/\|P\|_2$ instead of $\|P\|_2$ since $1/\|P\|_2$ is between zero and one, thus avoiding
overflow problems for very ill-conditioned eigenvalues.

7.1 Usage

Single precision.

CALL STRSNA(SELECT,N,T,LDT,RE,LDRE,LE,LDLE,S,SEP,
$ MM,M,WORK,LDWORK,X,V,B,ISGN,INFO)
*$
* .. Scalar Arguments ..
INTEGER N,LDT,LDRE,LDLE,MM,M,LDWORK,INFO
* *
* .. Array Arguments ..
LOGICAL SELECT(*)
INTEGER ISGN(*)
REAL S(*), SEP(*)
REAL T(LDT,*), RE(LDRE,*), LE(LDLE,*)
REAL WORK(LDWORK,*), X(*), V(*), B(*)
*
* Arguments
* =========
* * SELECT - LOGICAL      array if DIMENSION (N).
On entry, SELECT specifies the eigenpair whose condition numbers are to be estimated. The condition number corresponding to the J-th eigenpair is specified by setting SELECT(J) to .TRUE..
Not modified.

* N - INTEGER
On entry, N specifies the order of the matrix T. N must be at least zero.
Not modified.

* T - REAL array of DIMENSION (LDT,N).
On entry, T contains an upper quasi-triangular matrix in Schur canonical form. This means that the diagonal entries of 2 by 2 diagonal blocks must be equal.
Not modified.

* LDT - INTEGER
On entry, LDT specifies the first dimension of T as declared in the calling (sub)program. LDT must be at least max(1, N).
Not modified.

* RE - REAL array of DIMENSION (LDRE,MM).
On entry, RE contains the real and imaginary parts of the selected right eigenvectors computed by STREVC or SHSEIN.
If the next selected eigenvalue is real, the next column of RE contains its eigenvector. If the next selected eigenvalue is complex, the next two columns of RE contain the real and imaginary parts of its eigenvector.
Not modified.

* LDRE - INTEGER
On entry, LDRE specifies the leading dimension of RE as declared in the calling (sub)program. LDRE must be at least max(1, N).
Not modified.

* LE - REAL array of DIMENSION (LDLE,MM).
On entry, LE contains the real and imaginary parts of the
* selected left eigenvectors computed by STREVC or SHSEIN.
* If the next selected eigenvalue is real, the next column
* of RE contains its eigenvector. If the next selected
* eigenvalue is complex, the next two columns of LE contain
* the real and imaginary parts of its eigenvector.
* Not modified.
*
* LDLE - INTEGER
* On entry, LDLE specifies the leading dimension of LE as
* declared in the calling (sub)program. LDLE must be at
* least max(1, N).
* Not modified.
*
* S - REAL array of DIMENSION(MM).
* On exit, S contains the reciprocals of the condition
* numbers of the selected eigenvalues. If the Jth and
* (J+1)st eigenpairs are complex conjugate, then both
* S(J) and S(J+1) will be set (and equal).
*
* SEP - REAL array of DIMENSION(MM).
* On exit, SEP contains the estimated reciprocals of the
* condition numbers of the selected right eigenvectors.
* If the Jth and (J+1)st eigenpairs are complex conjugate,
* then both SEP(J) and SEP(J+1) will be set (and equal).
*
* MM - INTEGER
* On entry, MM should be set to an upper bound for the
* length of arrays S(*) and SEP(*) required to store the
* reciprocal condition numbers to be estimated. Note that
* for a complex conjugate eigenpair, we need two locations
* for S and SEP. This means S(J), SEP(J), RE(J), and LE(J)
* correspond to the same eigenvalue for all J.
* Not modified.
*
* M - INTEGER
* On exit, M is the size of arrays S(*) and SEP(*) actually
* used to store condition numbers.
*
* WORK - REAL array of DIMENSION(LDWORK,N)
* Workspace.
* LDWORK - INTEGER
* On entry, LDWORK specifies the leading dimension of WORK
* as declared in the calling (sub)program. LDWORK must be
* at least max(1, N).
* Not modified.
*
* X - REAL array of DIMENSION(2*(N-1)).
* Workspace.
*
* V - REAL array of DIMENSION(2*(N-1)).
* Workspace.
*
* B - REAL array of DIMENSION(N)
* Workspace.
*
* ISGN - INTEGER array of DIMENSION(2*(N-1))
* Workspace.
*
* INFO - INTEGER
* On exit, INFO is set to
* 0 for normal return,
* -K input argument number K is illegal.
* N+1 the assigned length of S and SEP too small.
*
**

Double precision. The calling sequence of the double precision routine DTRSNA is the same as that of the corresponding single precision “S” subroutine except that all the real variables are double precision.

Complex. The calling sequence of the single precision complex is essentially the same as STRSNA, except that the T, RE, WORK, X, V arrays are complex, and the integer array ISGN is real.

Double precision complex. The calling sequence of the double precision complex routine ZTRSNA is the same as that of the corresponding single precision “C” subroutine except that all the real variables are double precision and all the complex variables are double precision complex.
7.2 Example

The following program segment illustrates the use of the single precision subroutine to estimate selected reciprocal condition numbers of the eigenvalues and eigenvectors of a general matrix. The program first reduces a general matrix to upper Hessenberg form by SGEHRD, and then computes the Schur decomposition by the multishift QR iteration (SHSEQR). After that the user should input the logical array SELECT to specify the eigenpairs whose condition numbers will be estimated, and STREVC is called to compute the selected right and left eigenvectors and compactly store them in array RE and LE. Finally STRSNA is called to return the desired reciprocal condition numbers.

```fortran
PROGRAM TEST
INTEGER ISGN(100)
LOGICAL SELECT(50)
REAL A(50,50), WR(50), WI(50), RE(50,50), LE(50,50)
REAL S(50), SEP(50), RWORK(50)
REAL B(50), X(100), V(100), WORK(50,150), U(50,150)
INTEGER N, LDA, LDRE, LDLE, LDWORK, M, I, J, MAXN, INFO
INTEGER NBLCK1, NSHIFT, NBLCK2
REAL DUMMY
PARAMETER (LDA = 50, LDRE = 50, LDLE = 50, LDU = 50)
PARAMETER (LDWORK = 50, MAXN = 150)

* * Input data:
* N: the order of matrix A.
* A: N by N array to store the input matrix.
* NBLCK1: the blocksize used in Hessenberg reduction.
* NSHIFT: the number of shifts used in multishift QR algorithm.
* NBLCK2: the blocksize used in bulge chasing in QR iteration.
*
READ(*,*) N
DO 10 I = 1,N
   READ(*,*) (A(I,J),J = 1,N)
10 CONTINUE
READ(*,*) NBLCK1
READ(*,*) NSHIFT
READ(*,*) NBLCK2
*
* Compute Schur decomposition.
*
CALL XENVIR('BLOCK',NBLCK1)
CALL SGEHRD('H', N, A, LDA, DUMMY, DUMMY, WR, WORK, LDWORK,
$ MAXN, INFO)
CALL XENVIR('SHIFT', NSHIFT)
CALL XENVIR('BLOCK', NBLCK2)
CALL SHSEQR('S', N, A, LDA, DUMMY, DUMMY, WR, WI, U, LDU,
$ MAXN, WORK, LDWORK, MAXN, INFO)
DO 20 I = 1, N
    WRITE(*,*) WR(I),WI(I)
20 CONTINUE
*
* Input logical array SELECT to specify the eigenpairs whose
* condition numbers will be estimated.
*
DO 30 I = 1,N
    READ(*,*) SELECT(I)
30 CONTINUE
*
* Compute the selected eigenvectors
*
CALL STREVC('B', SELECT, N, A, LDA, RE, LDRE, LE, LDLE,
$ N, M, RWORK, INFO)
*
* Estimate the selected condition numbers of eigenpairs
*
CALL STRSNA(SELECT, N, A, LDA, RE, LDRE, LE, LDLE, S, SEP,
$ N, M, WORK, LDWORK, X, V, B, ISGN, INFO)
*
* Print output
*
DO 40 I = 1,M
    WRITE(*,*) S(I),SEP(I)
40 CONTINUE
STOP
END

7.3 Outline of the Algorithm

The STRSNA routine is designed to estimate the reciprocals of condition numbers of the selected eigenvalue-eigenvector pairs of a Schur canonical matrix
Logical array SELECT specifies the condition numbers to be estimated. The arrays RE and LE are used in STREVC to compactly store the selected right and left eigenvectors respectively. Then RE and LE are used in STRSNA to compute the reciprocal condition number for individual eigenvalues:

\[ s(\lambda) = \frac{\|r^Hl\|}{\|r\|_2 \|l\|_2} \]

where \( r \) and \( l \) are the right and left eigenvectors of \( T \) corresponding to the eigenvalue \( \lambda \). Note that for complex eigenvalues, the next two columns of RE (LE) store the real and imaginary parts of the right (left) eigenvectors, respectively. We see that computing the reciprocals of condition number of an eigenvalue costs \( O(n) \) operations.

Variable \( S(*) \) contains the reciprocals of condition numbers of the selected eigenvalues.

For estimating the reciprocals of condition numbers of the associated right eigenvectors, STRSNA first calls subroutine STREXC to swap the diagonal blocks of matrix \( T \) by orthogonal transformation to the form:

\[ QTQ^T = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} \]

where the \( n_1 \) by \( n_1 \) matrix \( T_{11} \) is 1 by 1 or 2 by 2 depending on whether the eigenvalue is real or complex. If \( T_{11} \) is a 1 by 1 block \( \lambda \), we have

\[ \text{sep}(\lambda) = \min_{\|x\|_2=1} \| (\lambda I - T_{22})x \|_2 \]

If \( T_{11} \) is a 2 by 2 block, then we use a unitary rotation to triangularize the 2 by 2 block to get

\[ \begin{pmatrix} \lambda & t_{12} \\ 0 & \tilde{T}_{22} \end{pmatrix} \]

whence

\[ \text{sep}(\lambda) = \min_{\|x\|_2=1} \| (\lambda - \tilde{T}_{22})x \|_2. \]

In both cases sep can be estimated by estimating the one-norm of

\[ K^{-1} = (T' - \lambda)^{-1} \]

because of the relationship

\[ \frac{1}{\sqrt{n-1}} \| K^{-1} \|_1 \leq \frac{1}{\text{sep}(\lambda)} = \| K^{-1} \|_2 \leq \sqrt{n-1} \| K^{-1} \|_1 \]

23
Estimating the one-norm of $\|K^{-1}\|_1$ can be done by calling SLACON via a reverse communication interface. This means one must provide the solution vectors $x$ and $y$ of the quasi-triangular systems:

$$Kx = z, \quad K^Ty = z$$

where $z$ is determined by SLACON. This is the function of the subroutine SLAQTR. Note that $K$ is a complex matrix if $\lambda$ is a complex eigenvalue, but it is of the form

$$K = C + iD$$

where the real part $C$ is a real upper quasi-triangular matrix, and the imaginary part is

$$D = \begin{pmatrix} x & x & \cdots & x \\ x & \ddots & & \\ & & x \\ \end{pmatrix}$$

Hence we can easily solve the complex systems

$$(C + iD)(p + iq) = (e + if), \quad (C + iD)^H(g + ih) = (e + if)$$

in real arithmetic, and use $2(n-1)$ length vectors $x = \begin{pmatrix} p \\ q \end{pmatrix}$ and $y = \begin{pmatrix} g \\ h \end{pmatrix}$ as the input vectors of SLACON. We also use the fact that for any complex matrix $C + iD$,

$$\frac{1}{\sqrt{2}} \| \begin{pmatrix} C & -D \\ D & C \end{pmatrix} \|_1 \leq \|C + iD\|_1 \leq \| \begin{pmatrix} C & -D \\ D & C \end{pmatrix} \|_1.$$ 

The cost of the algorithm depends on the location of the selected eigenvalues along the diagonal of the input matrix. Swapping adjacent diagonal blocks costs $O(n)$, so moving an eigenvalue at diagonal position $k$ to the upper left costs $O(kn)$ operations. Since it requires about $O(n^2)$ to solve a quasi-triangular system, estimating the condition number of an eigenvector costs $O(n^2)$ operations once the eigenvalue is in the upper left corner. Therefore the total cost is $O(n^3)$ per eigenvector condition number.

The variable SEP(*) contains the estimated reciprocal condition numbers of the selected eigenvectors.
8 STRSEN Estimating the Condition of a Cluster of Eigenvalues

In this section, we first show the usage of LAPACK subroutine STRSEN for estimating the reciprocal condition number of a specified multiple (or clustered) eigenvalue and its corresponding invariant subspace, and then outline the algorithm.

8.1 Usage

Single precision

CALL STRSEN(SELECT,N,T,LDT,S,SEP,WORK,LDWORK,NWORK, $ X,V,ISGN,INFO)

* .. Scalar Arguments ..
INTEGER N, LDT, LDWORK, NWORK, INFO
REAL S, SEP

* .. Array Arguments ..
LOGICAL SELECT(*)
INTEGER ISGN(*)
REAL T(LDT,*), WORK(LDWORK,*), X(*), V(*)

* Arguments
* =========

* SELECT - LOGICAL array if DIMENSION (N)
* On entry, SELECT specifies the 1 by 1 or 2 by 2 diagonal blocks in the eigenvalue cluster. For 2 by 2 blocks,
* the first index of SELECT must be set to .TRUE. if the block to be collected. Complex conjugate eigenvalues will
* either both be inside the cluster, or both outside.
* On exit, SELECT may have been altered. If the elements of SELECT corresponding to a 2 by 2 block were each initially
* set to .TRUE., the program resets the second one to .FALSE..

* N - INTEGER
* On entry, N specifies the order of the matrix T. N must be at least zero.
T - REAL array of DIMENSION(LDT,N)
On entry, T contains an upper quasi-triangular matrix in Schur canonical form. This means that the diagonal entries of 2 by 2 diagonal blocks must be equal.
Not modified.

LDT - INTEGER
On entry, LDT specifies the first dimension of T as declared in the calling (sub)program. LDT must be at least max(1, N).
Not modified.

S - REAL
On exit, S is a lower bound on the reciprocal of the condition number for the selected cluster of eigenvalues. S cannot underestimate the true reciprocal condition number by more than a factor of sqrt(N).

SEP - REAL
On exit, SEP is the estimated reciprocal of the condition number of the corresponding invariant subspace.

WORK - REAL array of DIMENSION(LDWORK, N2)
Workspace, where N2 is the number of eigenvalues in the cluster, counting multiplicities.

LDWORK - INTEGER
On entry, LDWORK specifies the first dimension of WORK as declared in the calling (sub)program. LDWORK must be at least max(1, N-N2).
Not modified.

NWORK - INTEGER
On entry, NWORK specifies the largest possible columns of working array U. NWORK should be larger than N2.
Not modified.

X - REAL array of DIMENSION(N1*N2)
Double precision. The calling sequence of the double precision routine DTRSEN is the same as that of the corresponding single precision “S” subroutine except that all the real variables are double precision.

Complex. The calling sequence of the single precision complex routine is essentially the same as STRSNA, except that the T, RE, WORK, X, V arrays are complex. Integer array ISGN is not needed.

Double precision complex. The calling sequence of the double precision complex routine ZTRSEN is the same as that of the corresponding single precision complex “C” subroutine except that all the real variables are double precision and all the complex variables are double precision complex.

8.2 Example

The following program segment illustrates the use of the single precision subroutine to estimate the reciprocal condition numbers of a specified cluster of eigenvalues and its corresponding invariant subspace for a general matrix. The program first reduces a general matrix to upper Hessenberg form by SGEHRD, and then computes the Schur decomposition by the multishift QR algorithm (SHSEQR). After that the user should input the logical array SELECT to specify the eigenvalues in the cluster. Then STRSEN is called to estimate the reciprocal condition numbers.

```
PROGRAM TEST
INTEGER ISGN(50)
```
REAL A(50,50), WR(50), WI(50)
REAL X(100), V(100), U(50,150), WORK(50,150)
LOGICAL SELECT(50)
INTEGER N, LDA, LDWORK, M, I, J, MAXN, INFO
INTEGER NBLCK1, NSHIFT, NBLCK2
REAL S, SEP, DUMMY
PARAMETER (LDA = 50, LDU = 50, LDWORK = 50, MAXN = 150)

* *
* Input data:
* N: the order of matrix A.
* A: N by N array to store the input matrix.
* NBLCK1: the blocksize used in Hessenberg reduction.
* NSHIFT: the number of shifts used in the multishift QR algorithm.
* NBLCK2: the blocksize used in bulge chasing in QR iteration.
* *
READ(*,*) N
DO 10 I = 1,N
   READ(*,*) (A(I,J),J = 1,N)
10 CONTINUE
READ(*,*) NBLCK1
READ(*,*) NSHIFT
READ(*,*) NBLCK2
* *
* Compute Schur decomposition.
* *
CALL XENVIR('BLOCK', NBLCK1)
CALL SGEHRD('H', N, A, LDA, DUM, DUM, WR, WORK, LDWORK,$
   MAXN, INFO)
CALL XENVIR('SHIFT', NSHIFT)
CALL XENVIR('BLOCK', NBLCK2)
CALL SHSEQR('S', N, A, LDA, DUM, DUM, WR, WI, U, LDU,$
   MAXN, WORK, LDWORK, MAXN, INFO)
DO 20 I = 1, N
   WRITE(*,*) WR(I),WI(I)
20 CONTINUE
* *
* Input the SELECT to specify the eigenvalues to be collected
* together, then the condition numbers of the corresponding
* invariant subspace will be estimated.
* 
28
DO 30 I = 1,N
    READ(*,*) SELECT(I)
30      CONTINUE
*
CALL STRSEN(SELECT, N, A, LDA, S, SEP, WORK, LDWORK,
$    ISGN, X, V, INFO)
*
* Print output
*
WRITE(*,*) S,SEP
STOP
END

8.3 Outline of the Algorithm

STRSEN routine estimates the reciprocal condition numbers of specified multiple (or clustered) eigenvalues and their corresponding invariant subspace for a real Schur canonical matrix $T$.

Logical array SELECT specifies the 1 by 1 (for real eigenvalues) or 2 by 2 (for complex conjugate eigenvalues) diagonal blocks that are to be collected together to form the desired cluster. Note that for 2 by 2 diagonal blocks, the first index of SELECT must be set to .TRUE. if the block to be collected. For real matrices, complex conjugate eigenpairs will both be selected if one is selected.

STRSEN first calls subroutine STREX2 to collect the selected diagonal blocks by orthogonal transformation to the top-left corner of $T$ such that

$$ QTQ^T = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} $$

where the selected blocks have been collected in $n_1$ by $n_1$ matrix $T_{11}$. Then STRSEN computes the projector $P$ on the invariant subspace associated with $T_{11}$. It is known that

$$ P = \begin{pmatrix} I & -R \\ 0 & 0 \end{pmatrix}, $$

where $R$ is the solution of the Sylvester equation

$$ T_{11}R - RT_{22} = T_{12}. $$

This is done by subroutine STRSYL. The program tests to avoid overflow if $\|R\|$ is very large, returning zero as the reciprocal condition number.
The return value $S$ of \texttt{STRSEN} is the lower bound $(1 + \|R\|^2)\|P\|^{-1/2}$ on the reciprocal of $\|P\|_2$. It cannot underestimate $\|P\|_2^{-1}$ by more than a factor of $n^{1/2}$.

Finally, \texttt{STRSEN} estimates the separation of $T_{11}$ and $T_{22}$. We know that this can be estimated by the one-norm of

$$K^{-1} = (I_{n_2} \otimes T_{11} - T_{22} \otimes I_{n_1})^{-1}, \quad n = n_1 + n_2$$

because of the relationship

$$\frac{1}{\sqrt{n_1 n_2}} \|K^{-1}\|_1 \leq \frac{1}{\text{sep}(T_{11}, T_{22})} = \|K^{-1}\|_2 \leq \sqrt{n_1 n_2} \|K^{-1}\|_1.$$  

This is done by calling \texttt{SLACON} via a reverse communication interface, providing the solution vectors $x$ and $y$ of the equations:

$$Kx = z, \quad K^Ty = z$$

where $z$ is determined by \texttt{SLACON}. This means we must solve the Sylvester equations:

$$T_{11}X - XT_{22} = Z \quad T_{11}^TY - YT_{22}^T = Z$$

This is again done by the subroutine \texttt{STRSYL}.

The return value \texttt{SEP} of \texttt{STRSEN} is the estimated (upper bound) of $\text{sep}(T_{11}, T_{22})$.

Swapping adjacent diagonal blocks on the diagonal of the input matrix costs $O(n)$, so swapping $n_1$ selected eigenvalues to the top left corner costs at most $O(n_1 n_2^2)$ (and as little as nothing if they are already at the top left corner). Once the desired eigenvalues are at the top left, solving either above Sylvester equation costs $O(n_1^2 n_2 + n_1 n_2^2)$ operations. Therefore the condition number estimation of a cluster of eigenvalues and their corresponding invariant subspace costs at most $O(n^3)$ operations, or as few as $O(n^2)$ if $n_1 \ll n_2$. 

30
A Solution of the Sylvester Equation

Considerable attention in the literature has been paid to solving the Sylvester equation. Among proposed solutions, Bartels and Stewart’s method[2], and Golub, Nash and Van Loan’s method[12] are direct matrix factorization methods. In this appendix, we discuss the method originally presented by Bartels and Stewart, and the associated routine STRSYL.

The Sylvester matrix equation is of the form

\[ \text{op}(A)X - X\text{op}(B) = sC \]  \hspace{1cm} (10)

where \( A, B \) and \( C \) are real \( m \times m, n \times n \) and \( m \times n \) matrices respectively. \( \text{op}(A) = A \) or \( A^T \) is a transpose option. \( s \) is a scaling factor \((\leq 1)\) which is so chosen so that \( X \) can be computed without overflow. We will also suppose that \( A \) and \( B \) are in upper quasi-triangular form, otherwise we should compute the Schur decomposition of \( A \) and \( B \) (by SGEHRD and SHSEQR),

\[ U^T A U = R, \quad V^T B V = S \]  \hspace{1cm} (11)

where \( R \) and \( S \) are upper quasi-triangular, and \( U \) and \( V \) are orthogonal. The reductions (A.11) lead to the system (A.10).

It is well known that (A.10) has a unique solution if and only if there are no common eigenvalues of \( A \) and \( B \).

Let \( p \) be the number of 1 by 1 and 2 by 2 blocks along the diagonal of \( A \), and let \( q \) be the number of the 1 by 1 and 2 by 2 blocks along the diagonal of \( B \). Partition the \( A, B, X \) and \( C \) conformally. If \( \text{op}(A) = A \) and \( \text{op}(B) = B \) then the \( ij \)th block \( X_{ij} \) satisfies

\[ A_{ii} X_{ij} - X_{ij} B_{jj} = s_1 (C_{ij} - C'_{ij}) \]  \hspace{1cm} (12)

where

\[ C'_{ij} = \sum_{k=i+1}^{p} A_{ik} X_{kj} - \sum_{l=1}^{i-1} X_{il} B_{lj} \]

Note that since \( A_{ii} \) and \( B_{jj} \) are each 1 by 1 or 2 by 2, the solution of (3) can be obtained by solving a linear system of order at most four. That can be solved easily. Once calculated, the solution \( X_{ij} \) can be stored in the locations occupied by \( C_{ij} \), which is no longer needed. The solution matrix \( X \) can be successively solved column by column starting from bottom-left corner of \( X \), i.e., in order \( X_{p1}, X_{p-1,1}, \ldots, X_{11}, X_{p2}, \ldots, X_{1q} \).
For solving equation (A.12), we note that if $A_{ii}$ and/or $B_{jj}$ are balanced $2 \times 2$ blocks, i.e., they are of the form

$$
\begin{pmatrix}
\alpha & \beta \\
\gamma & \alpha 
\end{pmatrix}
$$

then $\alpha \pm \mu$ are the eigenvalues where $\beta \gamma = -\mu^2$. Equation (A.12) can then be expressed as a $2 \times 2$ linear system

$$(H - w)Y = s_1F$$

Here $H$ is an $na$ by $na$ real matrix ($na = 1, 2$), $w$ is real or complex, $Y$ and $F$ are $na$ by 1 matrices which are real if $w$ is real and complex if $w$ is complex, and $s_1$ is a local scaling factor ($\leq 1$) which is so chosen that $Y$ can be computed without overflow (see SLALN2 and SLASY2). In particular, if $A_{ii}$ and $B_{jj}$ have the same eigenvalues, $s_1$ is set to 0.

Similarly, if $\text{op}(A) = A^T$ and $\text{op}(B) = B$, the $ij$th block $X_{ij}$ of the solution $X$ can be successively solved column by column starting from top-left corner of $X$, i.e., in order $X_{11}$, $X_{21}$, $\ldots$, $X_{p1}$, $X_{12}$, $\ldots$, $X_{pq}$.

If $\text{op}(A) = A$ and $\text{op}(B) = B^T$, the $ij$th block $X_{ij}$ of the solution $X$ can be successively solved column by column starting from bottom-right corner of $X$, i.e., in order $X_{pq}$, $X_{p-1,q}$, $\ldots$, $X_{1q}$, $X_{p,q-1}$, $\ldots$, $X_{11}$.

If $\text{op}(A) = A^T$ and $\text{op}(B) = B^T$, the $ij$th block $X_{ij}$ of the solution $X$ can be successively solved column by column starting from top-right corner of $X$, i.e., in order $X_{1q}$, $X_{2q}$, $\ldots$, $X_{pq}$, $X_{1,q-1}$, $\ldots$, $X_{p1}$.

Using the above different substitution orderings enables one to work on the input matrices directly rather than to transpose the input matrices.

The overall number of flops for the above substitution solution is

$$2.5(m^2n + mn^2)$$

where we have assumed that $A$ and $B$ are already in Schur form.

The program may be used to iteratively refine of the computed solution $X_1$ of (A.10): let the residual matrix $R_1 = C - AX_1 + X_1B$ be computed in double precision and rerounded to single precision. Use the same program to solve the system $AX_2 - X_2B = R_1$. Then $X_1 + X_2$ will in general be a more accurate approximation. This process may be iterated. This iteration is analogous to the iterative refinement of approximate solutions of linear systems as described by Wilkinson[27, p. 255]. (This is not done in STRSEN and STRSNA.)
B Swapping Diagonal Blocks

The crux of swapping a selected block of a real Schur form to a specified position along the diagonal (subroutine STREXC), or collecting selected blocks together (subroutine STREX2) is the swapping of adjacent blocks by an orthogonal similarity transformation (subroutine SLAEXC). Stewart [23] developed an adjacent block swapping algorithm using one or two QR steps with a pre-determined shift to force the ordering of the eigenvalues of the new blocks. More recently, Ng and Parlett [17] present a more straightforward algorithm for the same task. The presentation in this appendix is based on Ng and Parlett’s work. We discuss in more detail the treatment of pathological cases.

Consider a submatrix of the form

\[
\begin{pmatrix}
T_{11} & T_{12} \\
0 & T_{22}
\end{pmatrix}
\]

where \(T_{11}\) is a \(p\) by \(p\) matrix, and \(T_{22}\) is a \(q\) by \(q\) matrix, \(p, q = 1\) or \(2\), and assume that \(T_{11}\) and \(T_{22}\) have no eigenvalue in common. Moreover, we assume that if either is a 2 by 2 matrix, it has been standardized (i.e., it has identical diagonal entries.) Now, we want to find an orthogonal matrix \(Q\) which swaps \(T_{11}\) and \(T_{22}\), i.e.,

\[
Q \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} Q^T = \begin{pmatrix} \tilde{T}_{22} & \tilde{T}_{12} \\ 0 & \tilde{T}_{11} \end{pmatrix}
\]

where \(\tilde{T}_{ii}\) is similar to \(T_{ii}\), \(i = 1, 2\).

Since \(T_{11}\) and \(T_{22}\) have no eigenvalue in common, it follows that there exists a unique \(p \times q\) matrix \(X\) such that

\[
T_{11}X - XT_{22} = T_{12}.
\]

Hence

\[
\begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} = \begin{pmatrix} I_p & -X \\ 0 & I_q \end{pmatrix} \begin{pmatrix} T_{11} & 0 \\ 0 & T_{22} \end{pmatrix} \begin{pmatrix} I_p & X \\ 0 & I_q \end{pmatrix} = \begin{pmatrix} -X & I_p \\ I_q & 0 \end{pmatrix} \begin{pmatrix} T_{22} & 0 \\ 0 & T_{11} \end{pmatrix} \begin{pmatrix} 0 & I_q \\ I_p & X \end{pmatrix}
\]

We see that it is easy to find an orthogonal \((p + q) \times (p + q)\) matrix \(Q\) such that

\[
Q \begin{pmatrix} -X \\ I_q \end{pmatrix} = \begin{pmatrix} M_2 \\ 0 \end{pmatrix}
\]

\(13\)
with some invertible \( q \times q M_2 \), e.g., using Householder matrices to do the QR decomposition.

Let \( Q \) premultiply and postmultiply the original matrix, yielding
\[
Q \begin{pmatrix} T_{11} & T_{12} \\ T_{22} \end{pmatrix} Q^T = Q \begin{pmatrix} -X & I_p \\ I_q & 0 \end{pmatrix} \begin{pmatrix} T_{22} & 0 \\ 0 & T_{11} \end{pmatrix} \begin{pmatrix} 0 & I_q \\ I_p & X \end{pmatrix} Q^T
\]
\[
= \begin{pmatrix} M_2 & W \\ 0 & M_1 \end{pmatrix} \begin{pmatrix} T_{22} & 0 \\ 0 & T_{11} \end{pmatrix} \begin{pmatrix} M_2 & W \\ 0 & M_1 \end{pmatrix}^{-1}
\]
\[
= \begin{pmatrix} M_2 T_{22} M_2^{-1} & T_{12}' \\ 0 & M_1 T_{11} M_1^{-1} \end{pmatrix}.
\]

\( T_{11} \) and \( T_{22} \) have been swapped.

The above considerations are summed up in the following steps.

1. Solve \( T_{11} X - X T_{22} = sT_{12} \). \( s \) is a scale factor introduced to avoid overflow.

2. Check if the magnitude of \( \| X \| \) exceeds the square root of the overflow threshold. In this case \( T_{11} \) and \( T_{22} \) are too close to swap, so we exit.

3. Use a Householder matrix \( Q \) to do the QR decomposition of \( (X I)^T \) and update \( T \) by \( Q \): \( QTQ^T \).

4. Accumulate the orthogonal transformations if desired.

5. To preserve the standard Schur form, make the diagonal elements equal in each 2 by 2 block using orthogonal transformations.

6. Accumulate the orthogonal transformations if desired.

Several comments should be made. First, the solution of the matrix equation \( T_{11} X - X T_{22} = sT_{12} \) has been discussed in detail in Appendix A, the routine \texttt{SLASY2}. Second, there is no danger in working with \( X \) of large norm \textit{provided} that \( \| X \|^2 \) does not overflow. Moreover if \( \| X \|^2 \) does overflow then the blocks should not be swapped because a tiny perturbation will cause \( T_{11} \) and \( T_{22} \) to have at least one common eigenvalue\cite{9}. Hence in step 2, we check the norm of \( X \), and if \( X \) satisfies
\[
\frac{s \cdot \max(\| T_{11} \|, \| T_{22} \|)}{\| X \| + s} < \epsilon,
\]
where \( \epsilon \) is the machine precision, then the two blocks are regarded as too close to swap.
List of LAPACK Routines for the Nonsymmetric Eigenproblem

LAPACK main routines for the nonsymmetric eigenproblem:

**SGEBAL** Balance an input general matrix and isolate eigenvalues whenever possible.

**SGEBAK** Form the eigenvectors for a general matrix by back transforming those of the corresponding balanced matrix determined by **SGEBAL**.

**SGEHRD** Reduce a general matrix to an upper Hessenberg matrix.

**SHSEQR** Compute the eigenvalues of an upper Hessenberg matrix by the multishift QR algorithm, and return the Schur form, accumulating the orthogonal matrix if desired.

**STREVC** Compute selected right and/or left eigenvectors of a Schur matrix.

**SHSEIN** Compute selected right and/or left eigenvectors of a Hessenberg matrix by inverse iteration.

**SORGC3** Overwrite a matrix containing Householder vectors stored in the strictly lower part by the orthogonal matrix they represent.

**STRSNA** Estimate selected reciprocal condition numbers of individual eigenpairs of Schur matrix.

**STRSEN** Estimate selected reciprocal condition numbers of a multiple (or cluster of) eigenvalues and their corresponding invariant subspace of a Schur matrix.

**STRSYL** Solve the Sylvester equation with coefficient matrices in Schur form.

**STREXC** Swap a selected diagonal 1 by 1 or 2 by 2 block of a Schur matrix to a specified position.

**STREX2** Collect several selected diagonal 1 by 1 or 2 by 2 blocks of a Schur matrix to the top-left or bottom right corner.

LAPACK auxiliary routines for the nonsymmetric eigenproblem:

**SLAHRD** Chase a k by k bulge of an upper Hessenberg matrix one block down from a specified column number.
SLAHQR  BLAS 1 based QR routine to compute the eigenvalues of an upper Hessenberg matrix, and return the Schur form, accumulating the orthogonal matrix if desired.

SLATRS  Mixed subroutine of BLAS 1 and BLAS 2 to solve triangular equations while avoiding overflow.

SLAQTR  Solving real or complex quasi-triangular systems where the real part is quasi-triangular, and the imaginary part is of a special form.

SLALN2  Solve a 2 by 2 linear equation.

SLAE2   Compute the eigenvalues of a 2 by 2 nonsymmetric matrix.

SLAEXC  Swap adjacent diagonal 1 by 1 or 2 by 2 blocks of a Schur matrix.

SLASY2  Solve the Sylvester equation with coefficient matrices up to 2 by 2.

SLAEQU  Equalize the diagonal elements of a 2 by 2 block with an orthogonal similarity.

Other LAPACK routines, auxiliary routines, functions called by eigensystem subroutines (except Level 1, 2 or 3 BLAS routines).
    XERBLA, LSAME, R1MACH, ENVIR
    SLACON, SLACPY, SLAZRO, SLARFG, SLARF, SLANHS, SLAPY2, SLAPY3
References


