SHHEIG Users’ Guide

Peter Benner∗, Vasile Sima† and Matthias Voigt‡

June 10, 2015

Abstract

SHHEIG is a software package for solving of skew-Hamiltonian/Hamiltonian eigenproblems using structure-preserving algorithms. In this users’ guide we give an overview of this software and hints on its installation and usage.

1 Introduction

SHHEIG is a Fortran 77 software package for the solution of generalized eigenvalue problems for $2n \times 2n$ skew-Hamiltonian/Hamiltonian matrix pencils $\lambda S - H$. This package provides implementations of structure-preserving algorithms to

1. compute the eigenvalues;
2. reorder the eigenvalues in the computed structured normal forms;
3. compute the stable deflating subspaces, i.e., the deflating subspaces to the eigenvalues with negative real part.

For these tasks there exist subroutines for dealing with problems where the skew-Hamiltonian matrix $S$ is explicitly given, and for problems where only the factor $Z$ of

$$S = JZ^H J^T Z \quad \text{with} \quad J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

is known. Moreover, all subroutines have versions to deal with real and complex data, respectively. In order to exploit the cache more efficiently for larger problems, SHHEIG contains blocked versions of the codes (for explicitly given $S$ only). For a detailed derivation of the algorithms we refer to [2, 1]. For details on the implementation and numerical examples we refer to [3].

∗Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39016 Magdeburg, Germany, e-mail: benner@mpi-magdeburg.mpg.de
†National Institute for Research & Development in Informatics (ICI), Lab. 07, 8–10 Averescu Ave., Bucharest 1, Romania, code 011455, e-mail: vsima@ici.ro
‡Technische Universität Berlin, Sekretariat MA 4-5, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: mvoie@math.tu-berlin.de
2 Contents

The SHHEIG package is distributed via an archive `shheig.tar.gz`. After extraction one obtains the SHHEIG root directory `shheig` containing the following directories and files:

- the directory `src`, containing the Fortran 77 source files;
- the directory `examples`, containing Fortran test programs (`*.f`) and corresponding data input files (`*.dat`) and example output files (`*.res`);
- the directory `doc`, containing the html documentation describing input and output parameters as well as example inputs and outputs of the Fortran routines in `src`;
- the directory `mex`, containing MEX-files (`*.F`) for generating gateway functions that can be used to call the Fortran routines in MATLAB;
- the directory `test`, containing MATLAB functions for testing the functionality and performance of the gateway functions generated by the MEX-files included in `mex`,
- the directory `usersguide`, containing this users’ guide;
- a `makefile` and the file `make.inc` for building the SHHEIG library;
- the documentation index `libindex.html`.
- a `readme` file containing some general hints.

3 Installation and Usage

3.1 Before Installation

SHHEIG routines make calls to subprograms from the state-of-the-art packages SLICOT (Subroutine Library in Control Theory), LAPACK (Linear Algebra Package) and BLAS (Basic Linear Algebra Subprograms). Thus it is necessary to download and install these libraries before building the SHHEIG library.

SLICOT source code and the prebuilt library is freely available for academic users after registration from the SLICOT website (http://slicot.org). The LAPACK and BLAS libraries are freely downloadable from netlib (http://www.netlib.org/). However, for maximum efficiency it is recommended to use machine-specific, optimized versions whenever possible.

3.2 Building the SHHEIG Library

Template make files are provided to help building the SHHEIG Library object files, and to link and run the available example programs calling the SHHEIG Library routines. In order to use these make files on a specific Unix platform, some changes might be needed in the files `make.inc` and/or `makefile` stored in the SHHEIG root directory.

The changes in `make.inc` might define the specific machine (platform) identifier, the compiler, linker, and archiver flags, as well as the location and names
of the SLICOT, LAPACK, and BLAS libraries, which the program files should be linked to. Details are given in the file `make.inc`.

**IMPORTANT:** On 64bit platforms the code must be compiled with the options `-fPIC` and `-fdefault-integer-8`, for instance by setting

```
OPTS = -O2 -fPIC -fdefault-integer-8
```

in `make.inc`.

Changes in the makefile might be needed for using a Fortran 77 compiler, since a Fortran 90/95 compiler is assumed by default to build the executable example programs. (The SHHEIG routines themselves are written in Fortran 77.)

After performing the necessary changes, as suggested in the comments of the make files, the other needed SHHEIG-related files can be generated automatically with the command

```
make
```

issued from the SHHEIG root directory `shheig`.

The first execution of `make` will create the following files:

- the SHHEIG library object files `*.o` in the subdirectory `src` of `shheig`;
- the library file `shheig.a`, in the directory `shheig`;
- the example program object and executable files, in the subdirectory `examples`;
- the files `*.exa`, with the results computed on the local machine, with the same names as the files with data (`*.dat`) and reference results (`*.res`), also in the subdirectory `examples`.

The subsequent executions of `make` will update the files if changes have been performed.

The files `*.exa`, with the computed results may be compared with the reference results. Several types of differences could be noticed, including possible sign changes for some elements, or even different values in some columns and/or rows of the computed matrices. For instance, the matrices of similarity or equivalence transformations could differ from a platform/compiler to another. This does not usually mean that the computed results are wrong.

More details for executing other tasks, e.g., cleaning the subdirectories `src` and `examples`, are given in the files `makefile` included in the directory `shheig` and in the subdirectories `src` and `examples`.

### 3.3 Creating Gateway Functions for Running SHHEIG in MATLAB

For ease of usage, MEX-files contained in the subdirectory `mex` have been written to generate gateway functions that allow the user the call the main subroutines in MATLAB. There are two ways to compile the MEX-files.

The easiest way consists of going to the subdirectory `mex` and typing the command

```
allmex
```
in MATLAB. This command will first call the MATLAB functions slicotmex and shheigmex that generate the SLICOT and SHHEIG libraries using the built-in Fortran compiler of MATLAB. The respective library files are called mexslicot.a and mexshheig.a. Then the function makemex generates the gateway functions from the MEX-files which will then be tested by the function testmex. The functions called by allmex detect which kind of machine is used and will automatically set the correct flags. However, depending on the location of the SLICOT source files, it might be necessary to modify the value of the variable slicot_src in slicotmex.m, or alternatively, use the path of the SLICOT source files as an argument in allmex.

The second method consists of linking the SLICOT and SHHEIG library files generated outside of MATLAB when compiling the MEX-files. To do so it is necessary to modify the values of the variables libslicot and libshheig in the file makemex.m. Then the gateway functions are generated by typing

makemex

and tested by typing
testmex

in MATLAB in the subdirectory mex.

References


A Interfaces and Call Graphs of the Driver Routines

A.1 The Subroutine DGHFDF

SUBROUTINE DGHFDF( COMPQ, COMPU, ORTH, N, Z, LDZ, B, LDB, FG, $ LDIFG, NEIG, Q, LDQ, U, LDU, ALPHAR, ALPHAI, $ BETA, IWORK, LIWORK, DWORK, LDWORK, BWORK, $ IWARN, INFO )

C PURPOSE
C
C To compute the relevant eigenvalues of a real N-by-N skew-
C Hamiltonian/Hamiltonian pencil aS - bH, with
C
( B F ) ( 0 I )
where the notation $M'$ denotes the transpose of the matrix $M$.

Optionally, if COMPQ = 'C', an orthogonal basis of the right
deflating subspace of $aS - bH$ corresponding to the eigenvalues
with strictly negative real part is computed. Optionally, if
COMPU = 'C', an orthonormal basis of the companion subspace,
range($P_U$) [1], which corresponds to the eigenvalues with strictly
negative real part, is computed.

ARGUMENTS

Mode Parameters

COMPQ CHARACTER*1
Specifies whether to compute the right deflating subspace
corresponding to the eigenvalues of $aS - bH$ with strictly
negative real part.
= 'N': do not compute the deflating subspace;
= 'C': compute the deflating subspace and store it in the
leading subarray of Q.

COMPU CHARACTER*1
Specifies whether to compute the companion subspace
corresponding to the eigenvalues of $aS - bH$ with strictly
negative real part.
= 'N': do not compute the companion subspace;
= 'C': compute the companion subspace and store it in the
leading subarray of U.

ORTH CHARACTER*1
If COMPQ = 'C' and/or COMPU = 'C', specifies the technique
for computing the orthogonal basis of the deflating
subspace, and/or of the companion subspace, as follows:
= 'P': QR factorization with column pivoting;
= 'S': singular value decomposition.
If COMPQ = 'N' and COMPU = 'N', the ORTH value is not
used.

Input/Output Parameters

N (input) INTEGER
The order of the pencil $aS - bH$. $N > 0$, even.

Z (input/output) DOUBLE PRECISION array, dimension (LDZ, N)
On entry, the leading $N$-by-$N$ part of this array must
contain the non-trivial factor $Z$ in the factorization
$S = J Z' J' Z$ of the skew-Hamiltonian matrix $S$.
On exit, if COMPQ = 'C' or COMPU = 'C', the leading
$N$-by-$N$ part of this array contains the transformed upper
triangular matrix $Z_1$ (see METHOD), after moving the
eigenvalues with strictly negative real part to the top
of the pencil (3). The strictly lower triangular part is
not zeroed.
If COMPQ = 'N' and COMPU = 'N', the leading $N$-by-$N$ part of
this array contains the matrix $Z$ obtained by the routine
DGHURV just before the application of the periodic QZ
algorithm. The elements of the $(2,1)$ block, i.e., in the
rows $N/2+1$ to $N$ and in the columns $1$ to $N/2$ are not set to
zero, but are unchanged on exit.
C LDZ INTEGER
C The leading dimension of the array Z. LDZ >= MAX(1, N).
C
C B (input) DOUBLE PRECISION array, dimension (LDB, N/2)
C On entry, the leading N/2-by-N/2 part of this array must
C contain the matrix B.
C
C LDB INTEGER
C The leading dimension of the array B. LDB >= MAX(1, N/2).
C
C FG (input) DOUBLE PRECISION array, dimension (LDFG, N/2+1)
C On entry, the leading N/2-by-N/2 lower triangular part of
C this array must contain the lower triangular part of the
C symmetric matrix G, and the N/2-by-N/2 upper triangular
C part of the submatrix in the columns 2 to N/2+1 of this
C array must contain the upper triangular part of the
C symmetric matrix F.
C
C LDFG INTEGER
C The leading dimension of the array FG.
C LDFG >= MAX(1, N/2).
C
C NEIG (output) INTEGER
C If COMPQ = 'C' or COMPU = 'C', the number of eigenvalues
C in aS - bH with strictly negative real part.
C
C Q (output) DOUBLE PRECISION array, dimension (LDQ, 2*N)
C On exit, if COMPQ = 'C', the leading N-by-NEIG part of
C this array contains an orthogonal basis of the right
C deflating subspace corresponding to the eigenvalues of
C aS - bH with strictly negative real part. The remaining
C part of this array is used as workspace.
C If COMPQ = 'N', this array is not referenced.
C
C LDQ INTEGER
C The leading dimension of the array Q.
C LDQ >= 1, if COMPQ = 'N';
C LDQ >= MAX(1, 2*N), if COMPQ = 'C'.
C
C U (output) DOUBLE PRECISION array, dimension (LDU, 2*N)
C On exit, if COMPU = 'C', the leading N-by-NEIG part of
C this array contains an orthogonal basis of the companion
C subspace corresponding to the eigenvalues of aS - bH with
C strictly negative real part. The remaining part of this
C array is used as workspace.
C If COMPU = 'N', this array is not referenced.
C
C LDU INTEGER
C The leading dimension of the array U.
C LDU >= 1, if COMPU = 'N';
C LDU >= MAX(1, N), if COMPU = 'C'.
C
C ALPHAR (output) DOUBLE PRECISION array, dimension (N/2)
C The real parts of each scalar alpha defining an eigenvalue
C of the pencil aS - bH.
C
C ALPHAI (output) DOUBLE PRECISION array, dimension (N/2)
C The imaginary parts of each scalar alpha defining an
C eigenvalue of the pencil aS - bH.
C If ALPHAI(j) is zero, then the j-th eigenvalue is real.
C BETA (output) DOUBLE PRECISION array, dimension (N/2)
C The scalars beta that define the eigenvalues of the pencil
C aS - bH.
C If INFO = 0, the quantities alpha = (ALPHAR(j),ALPHAI(j)),
C and beta = BETA(j) represent together the j-th eigenvalue
C of the pencil aS - bH, in the form lambda = alpha/beta.
C Since lambda may overflow, the ratios should not, in
general, be computed. Due to the skew-Hamiltonian/
Hamiltonian structure of the pencil, only half of the
spectrum is saved in ALPHAR, ALPHAI and BETA.
C Specifically, only eigenvalues with imaginary parts
greater than or equal to zero are stored; their conjugate
eigenvalues are not stored. If imaginary parts are zero
(i.e., for real eigenvalues), only positive eigenvalues
are stored. The remaining eigenvalues have opposite signs.
C If IWARN = 1, one or more BETA(j) is not representable,
C and the eigenvalues are returned as described below (see
C the description of the argument IWARN).

Workspace

IWORK INTEGER array, dimension (LIWORK)
C On exit, if INFO = -20, IWORK(1) returns the minimum value
C of LIWORK.
C On exit, if INFO = 0 and IWARN = 1, then IWORK(1), ..., 
C IWORK(N/2) return the scaling parameters for the
C eigenvalues of the pencil aS - bH (see IWARN).

LIWORK INTEGER
C The dimension of the array IWORK.
C LIWORK >= N + 18, if COMPQ = 'N' and COMPU = 'N';
C LIWORK >= MAX( N + 18, N/2 + 48, 5*N/2 + 1 ), otherwise.

DWORK DOUBLE PRECISION array, dimension (LDWORK)
C On exit, if INFO = 0, DWORK(1) returns the optimal value
C of LDWORK, and DWORK(2) returns the machine base, b.
C On exit, if INFO = -22, DWORK(1) returns the minimum value
C of LDWORK.

LDWORK INTEGER
C The dimension of the array DWORK.
C LDWORK >= c*N**2 + max( N*N + MAX( N/2+252, 432 ),
C MAX(8*N+48,171) ), where
C c = a, if COMPU = 'N',
C c = a+1, if COMPU = 'C', and
C a = 6, if COMPQ = 'N',
C a = 9, if COMPQ = 'C'.
C For good performance LDWORK should be generally larger.
C
C If LDWORK = -1 a workspace query is assumed; the
C routine only calculates the optimal size of the DWORK
C array, returns this value as the first entry of the DWORK
C array, and no error message is issued by XERBLA.

BWORK LOGICAL array, dimension (N/2)

Warning Indicator

IWARN INTEGER
C = 0: no warning;
C = 1: the eigenvalues will under- or overflow if evaluated;
C therefore, the j-th eigenvalue is represented by
the quantities \( \alpha = (\text{ALPHAR}(j), \text{ALPHAI}(j)) \),
\( \beta = \text{BETA}(j) \), and \( \gamma = \text{IWORK}(j) \) in the form
\( \lambda = (\alpha/\beta) \cdot b^{\gamma} \), where \( b \) is the
machine base (often 2.0), returned in \( \text{DWORK}(2) \).

Error Indicator

\textbf{INFO} INTEGER
- 0: successful exit;
- < 0: if \text{INFO} = -i, the i-th argument had an illegal value;
- 1: periodic QZ iteration failed in the routines \text{DGHURV},
\text{DGHFYR} or the SLICOT library routine \text{MB03BB} (QZ
iteration did not converge or computation of the
shifts failed);
- 2: standard QZ iteration failed in the routines \text{DGHFYR}
or \text{DGHFXE} (called by \text{DGHFX})�;
- 3: a numerically singular matrix was found in the routine
\text{DGHFXY} (called by \text{DGHFX})�;
- 4: the singular value decomposition failed in the LAPACK
routine \text{DGESVD} (for ORTH = 'S').

METHOD

First, the decompositions of \( S \) and \( H \) are computed via orthogonal
matrices \( Q_1 \) and \( Q_2 \) and orthogonal symplectic matrices \( U_1 \) and \( U_2 \),
such that
\( Q_1' T U_1 = J' Z' J' U_1 = ( T_{11} \ 0 ) \),
\( ( Z_{11} \ Z_{12} ) \)
\( ( 0 \quad Z_{22} ) \)
\( H = ( H_{11} \ 0 ) \)
\( ( 0 \quad H_{22} ) \)
where \( T_{11}, Z_{11}, Z_{22}, H_{11} \) are upper triangular and \( H_{22} \) is
upper quasi-triangular.

Then, orthogonal matrices \( Q_3, Q_4 \) and \( U_3 \) are found, for the
matrices
\( Z_{11} = ( \begin{array}{cc} T_{22} & 0 \\ 0 & 0 \end{array} ) \), \( Z_{22} = ( \begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} ) \), \( H = ( \begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} ) \),
\( ( \begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} ) \)
such that \( Z_{11} := U_3' Z_{11} Q_4, Z_{22} := U_3' Z_{22} Q_3 \) are upper
triangular and \( H_{11} := Q_3' H Q_4 \) is upper quasi-triangular. The
following matrices are computed:
\( \begin{array}{cc} T_{12} & 0 \\ 0 & H_{12} \end{array} \), \( Q_3' H Q_4 \) is upper quasi-triangular. The

Then, an orthogonal matrix \( Q \) and an orthogonal symplectic matrix \( U \)
are found such that the eigenvalues with strictly negative real
parts of the pencil
are moved to the top of this pencil.

Finally, an orthogonal basis of the right deflating subspace and an orthogonal basis of the companion subspace corresponding to the eigenvalues with strictly negative real part are computed. See also page 11 in [1] for more details.

REFERENCES

Numerical Solution of Real Skew-Hamiltonian/Hamiltonian Eigenproblems.

NUMERICAL ASPECTS

The algorithm is numerically backward stable and needs $O(N^3)$ floating point operations.

FURTHER COMMENTS

This routine does not perform any scaling of the matrices. Scaling might sometimes be useful, and it should be done externally.

Figure 1: Call graph of DGHFDF.

A.2 The Subroutine DGHUDF

SUBROUTINE DGHUDF( COMPQ, ORTH, N, A, LDA, DE, LDDE, B, LDB, FG, 
$ \ LDFG, NEIG, Q, LDQ, ALPHAR, ALPHAI, BETA, 
$ \ IWORK, LIWORK, DWORK, LDWORK, BWORK, INFO )

PURPOSE

To compute the relevant eigenvalues of a real N-by-N skew-Hamiltonian/Hamiltonian pencil $aS - bH$, with

$$( A D ) \quad ( B F )$$
\[ S = \begin{pmatrix} 0 \\ E \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} 0 \\ G \end{pmatrix}, \quad (1) \]

where the notation \( M' \) denotes the transpose of the matrix \( M \).

Optionally, if \( COMPQ = 'C' \), an orthogonal basis of the right deflating subspace of \( aS - bH \) corresponding to the eigenvalues with strictly negative real part is computed.

**ARGUMENTS**

**Mode Parameters**

1. **COMPQ**: CHARACTER*1
   - Specifies whether to compute the right deflating subspace corresponding to the eigenvalues of \( aS - bH \) with strictly negative real part.
   - \( 'N' \): do not compute the deflating subspace;
   - \( 'C' \): compute the deflating subspace and store it in the leading subarray of \( Q \).

2. **ORTH**: CHARACTER*1
   - If \( COMPQ = 'C' \), specifies the technique for computing the orthogonal basis of the deflating subspace, as follows:
   - \( 'P' \): QR factorization with column pivoting;
   - \( 'S' \): singular value decomposition.
   - If \( COMPQ = 'N' \), the ORTH value is not used.

**Input/Output Parameters**

1. **N**: (input) INTEGER
   - The order of the pencil \( aS - bH \). \( N \geq 0 \), even.

2. **A**: (input/output) DOUBLE PRECISION array, dimension \((LDA, N/2)\)
   - On entry, the leading \( N/2 \)-by-\( N/2 \) part of this array must contain the matrix \( A \).
   - On exit, if \( COMPQ = 'C' \), the leading \( N/2 \)-by-\( N/2 \) part of this array contains the upper triangular matrix \( A_{out} \) (see METHOD); otherwise, it contains the upper triangular matrix \( A \) obtained just before the application of the periodic QZ algorithm (see the routine DGHUTR).

3. **LDA**: INTEGER
   - The leading dimension of the array \( A \). \( LDA \geq \max(1, N/2) \).

4. **DE**: (input/output) DOUBLE PRECISION array, dimension \((LDDE, N/2+1)\)
   - On entry, the leading \( N/2 \)-by-\( N/2 \) lower triangular part of this array must contain the lower triangular part of the skew-symmetric matrix \( E \), and the \( N/2 \)-by-\( N/2 \) upper triangular part of the submatrix in the columns 2 to \( N/2+1 \) of this array must contain the upper triangular part of the skew-symmetric matrix \( D \).
   - The entries on the diagonal and the first superdiagonal of this array need not be set, but are assumed to be zero.
   - On exit, if \( COMPQ = 'C' \), the leading \( N/2 \)-by-\( N/2 \) lower triangular part and the first superdiagonal contain the transpose of the upper quasi-triangular matrix \( C_{out} \) (see METHOD), and the \( (N/2-1) \)-by-\( (N/2-1) \) upper triangular part of the submatrix in the columns 3 to \( N/2+1 \) of this array contains the strictly upper triangular part of the skew-symmetric matrix \( D_{out} \) (see METHOD), without the main
diagonal, which is zero. On exit, if COMPQ = 'N', the leading N/2-by-N/2 lower
triangular part and the first superdiagonal contain the
transpose of the upper Hessenberg matrix C2, and the
(N/2-1)-by-(N/2-1) upper triangular part of the submatrix
in the columns 3 to N/2+1 of this array contains the
strictly upper triangular part of the skew-symmetric
matrix D (without the main diagonal) just before the
application of the periodic QZ algorithm.

LDDE INTEGER
The leading dimension of the array DE.
LDDE => MAX(1, N/2).

B (input/output) DOUBLE PRECISION array, dimension
(LDB, N/2)
On entry, the leading N/2-by-N/2 part of this array must
contain the matrix B.
On exit, if COMPQ = 'C', the leading N/2-by-N/2 part of
this array contains the upper triangular matrix C1out
(see METHOD); otherwise, it contains the upper triangular
matrix C1 obtained just before the application of the
periodic QZ algorithm.

LDB INTEGER
The leading dimension of the array B. LDB => MAX(1, N/2).

FG (input/output) DOUBLE PRECISION array, dimension
(LDFG, N/2+1)
On entry, the leading N/2-by-N/2 lower triangular part of
this array must contain the lower triangular part of the
symmetric matrix G, and the N/2-by-N/2 upper triangular
part of the submatrix in the columns 2 to N/2+1 of this
array must contain the upper triangular part of the
symmetric matrix F.
On exit, if COMPQ = 'C', the leading N/2-by-N/2 part of
the submatrix in the columns 2 to N/2+1 of this array
contains the matrix Vout (see METHOD); otherwise, it
contains the matrix V obtained just before the application
of the periodic QZ algorithm.

LDFG INTEGER
The leading dimension of the array FG.
LDFG => MAX(1, N/2).

NEIG (output) INTEGER
If COMPQ = 'C', the number of eigenvalues in aS - bH with
strictly negative real part.

Q (output) DOUBLE PRECISION array, dimension (LDQ, 2*N)
On exit, if COMPQ = 'C', the leading N-by-NEIG part of
this array contains an orthogonal basis of the right
deflating subspace corresponding to the eigenvalues of
aA - bB with strictly negative real part. The remaining
part of this array is used as workspace.
If COMPQ = 'N', this array is not referenced.

LDQ INTEGER
The leading dimension of the array Q.
LDQ => 1, if COMPQ = 'N';
LDQ => MAX(1, 2*N), if COMPQ = 'C'.

11
ALPHAR (output) DOUBLE PRECISION array, dimension (N/2)
The real parts of each scalar alpha defining an eigenvalue of the pencil aS - bH.

ALPHAI (output) DOUBLE PRECISION array, dimension (N/2)
The imaginary parts of each scalar alpha defining an eigenvalue of the pencil aS - bH.
If ALPHAI(j) is zero, then the j-th eigenvalue is real.

BETA (output) DOUBLE PRECISION array, dimension (N/2)
The scalars beta that define the eigenvalues of the pencil aS - bH.
Together, the quantities alpha = (ALPHAR(j),ALPHAI(j)) and beta = BETA(j) represent the j-th eigenvalue of the pencil aS - bH, in the form lambda = alpha/beta. Since lambda may overflow, the ratios should not, in general, be computed.
Due to the skew-Hamiltonian/Hamiltonian structure of the pencil, for every eigenvalue lambda, -lambda is also an eigenvalue, and thus it has only to be saved once in ALPHAR, ALPHAI and BETA.
Specifically, only eigenvalues with imaginary parts greater than or equal to zero are stored; their conjugate eigenvalues are not stored. If imaginary parts are zero (i.e., for real eigenvalues), only positive eigenvalues are stored. The remaining eigenvalues have opposite signs.

Workspace

IWORK INTEGER array, dimension (LIWORK)
On exit, if INFO = -19, IWORK(1) returns the minimum value of LIWORK.

LIWORK INTEGER
The dimension of the array IWORK.
LIWORK >= MAX( 32, N + 12, 2*N + 1 ).

DWORK DOUBLE PRECISION array, dimension (LDWORK)
On exit, if INFO = 0, DWORK(1) returns the optimal LDWORK.
On exit, if INFO = -21, DWORK(1) returns the minimum value of LDWORK.

LDWORK INTEGER
The dimension of the array DWORK.
LDWORK >= 3*(N/2)**2 + N**2 + MAX( MAX( N, 32 ) + 4,
4*(N+1) ), if COMPQ = 'N',
where the last term can be reduced to 4*N if N/2 is odd;
LDWORK >= 8*N**2 + MAX( 8*N + 32, N/2 + 168, 272 ),
if COMPQ = 'C'.
For good performance LDWORK should be generally larger.
If LDWORK = -1 a workspace query is assumed; the routine only calculates the optimal size of the DWORK array, returns this value as the first entry of the DWORK array, and no error message is issued by XERBLA.

BWORK LOGICAL array, dimension (N/2)

Error Indicator

INFO INTEGER
= 0: successful exit;
< 0: if INFO = -i, the i-th argument had an illegal value;
1: periodic QZ iteration failed in the routines DGHUTR or DGHUYR (QZ iteration did not converge or computation of the shifts failed);  
2: standard QZ iteration failed in the routines DGHUYR or DGHUEX (called by DGHUXC);  
3: a numerically singular matrix was found in the routine DGHUEY (called by DGHUXC);  
4: the singular value decomposition failed in the LAPACK routine DGESVD (for ORTH = 'S').

METHOD

First, the decompositions of $S$ and $H$ are computed via orthogonal transformations $Q_1$ and $Q_2$ as follows:

\[
\begin{align*}
Q_1' S \quad J_1' &= \begin{pmatrix} A_{out} & D_{out} \\ 0 & A_{out}' \end{pmatrix}, \\
J_1' Q_2' \quad J S \quad Q_2 &= \begin{pmatrix} 0 & A_{out}' \\ B_{out} & 0 \end{pmatrix} =: T, \\
Q_1' H \quad Q_2 &= \begin{pmatrix} C_{1out} & V_{out} \\ 0 & C_{2out}' \end{pmatrix},
\end{align*}
\]

and $A_{out}$, $B_{out}$, $C_{1out}$ are upper triangular, $C_{2out}$ is upper quasi-triangular and $D_{out}$ and $F_{out}$ are skew-symmetric.

Then, orthogonal matrices $Q_3$ and $Q_4$ are found, for the extended matrices

\[
\begin{align*}
S_e &= \begin{pmatrix} A_{out} & 0 \\ 0 & C_{1out} \end{pmatrix} \quad \text{and} \quad H_e &= \begin{pmatrix} 0 & B_{out} \\ -C_{2out} & 0 \end{pmatrix},
\end{align*}
\]

such that $S_{11} := Q_4' S_e Q_3$ is upper triangular and $H_{11} := Q_4' H_e Q_3$ is upper quasi-triangular. The following matrices are computed:

\[
S_{12} := Q_4' \begin{pmatrix} 0 & V_{out} \\ V_{out}' & 0 \end{pmatrix} Q_4 \quad \text{and} \quad H_{12} := Q_4' \begin{pmatrix} 0 & 0 \\ 0 & -H_{11}' \end{pmatrix} Q_4.
\]

Then, an orthogonal matrix $Q$ is found such that the eigenvalues with strictly negative real parts of the pencil

\[
\begin{pmatrix} S_{11} & S_{12} \\ 0 & H_{11} \end{pmatrix} a ( ) - b ( )
\]

are moved to the top of this pencil.

Finally, an orthogonal basis of the right deflating subspace corresponding to the eigenvalues with strictly negative real part is computed. See also page 12 in [1] for more details.

REFERENCES

Numerical Solution of Real Skew-Hamiltonian/Hamiltonian...
A.3 The Subroutine ZGHFDF

SUBROUTINE ZGHFDF( COMPQ, COMPU, ORTH, N, Z, LDZ, B, LDB, FG, $ LDFG, NEIG, D, LDD, C, LDC, Q, LDQ, U, LDU, $ ALPHAR, ALPHAI, BETA, IWORK, LIWORK, DWORK, $ LDWORK, ZWORK, LWORK, BWORK, INFO )

PURPOSE

To compute the eigenvalues of a complex N-by-N skew-Hamiltonian/
Hamiltonian pencil aS - bH, with

\[
S = J Z^T J^T Z \quad \text{and} \quad H = \begin{pmatrix} G & -B^T \\ B & -G \end{pmatrix},
\]

where

\[
J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}
\]

The structured Schur form of the embedded real skew-Hamiltonian/
skew-Hamiltonian pencil, aB_S - bB_T, with B_S = J B_Z^T J^T B_Z,

\[
B_Z = \begin{pmatrix} \text{Re}(Z_{11}) & -\text{Im}(Z_{11}) & \text{Re}(Z_{12}) & -\text{Im}(Z_{12}) \\ \text{Im}(Z_{11}) & \text{Re}(Z_{11}) & \text{Im}(Z_{12}) & \text{Re}(Z_{12}) \end{pmatrix}
\]
is determined and used to compute the eigenvalues. Optionally, if
COMPQ = 'C', an orthonormal basis of the right deflating subspace,
Def_-(S, H), of the pencil aS - bH in (1), corresponding to the
eigenvalues with strictly negative real part, is computed. Namely,
after transforming aB_S - bB_H, in the factored form, by unitary
matrices, we have B_Sout = J B_Zout' J' B_Zout,
and the eigenvalues with strictly negative real part of the
complex pencil aB_Sout - bB_Hout are moved to the top. The
notation M' denotes the conjugate transpose of the matrix M.
Optionally, if COMPU = 'C', an orthonormal basis of the companion
subspace, \( \text{range}(P_U) \setminus \{1\} \), which corresponds to the eigenvalues
with negative real part, is computed. The embedding doubles the
multiplicities of the eigenvalues of the pencil aS - bH.

ARGUMENTS

Mode Parameters

COMPQ CHARACTER*1
Specifies whether to compute the right deflating subspace
corresponding to the eigenvalues of aS - bH with strictly
negative real part.
- 'N': do not compute the deflating subspace;
- 'C': compute the deflating subspace and store it in the
  leading subarray of Q.

COMPU CHARACTER*1
Specifies whether to compute the companion subspace
corresponding to the eigenvalues of aS - bH with strictly
negative real part.
- 'N': do not compute the companion subspace;
- 'C': compute the companion subspace and store it in the
  leading subarray of U.

ORTH CHARACTER*1
If COMPQ = 'C' or COMPU = 'C', specifies the technique for
computing the orthonormal bases of the deflating subspace
and companion subspace, as follows:
- 'P': QR factorization with column pivoting;
- 'S': singular value decomposition.
If COMPQ = 'N' and COMPU = 'N', the ORTH value is not used.
Input/Output Parameters

\(N\)  (input) INTEGER
Order of the pencil \(aS - bH\). \(N \geq 0\), even.

\(Z\)  (input/output) COMPLEX*16 array, dimension (LDZ, N)
On entry, the leading \(N\)-by-\(N\) part of this array must contain the non-trivial factor \(Z\) in the factorization \(S = JZ'J'Z\) of the skew-Hamiltonian matrix \(S\).
On exit, if \(COMPQ = 'C'\) or \(COMPU = 'C'\), the leading \(N\)-by-\(N\) part of this array contains the upper triangular matrix \(BA\) in (3) (see also METHOD). The strictly lower triangular part is not zeroed.
If \(COMPQ = 'N'\) and \(COMPU = 'N'\), this array is unchanged on exit.

LDZ INTEGER
The leading dimension of the array \(Z\). \(LDZ \geq \max(1, N)\).

\(B\)  (input/output) COMPLEX*16 array, dimension (LDB, N)
On entry, the leading \(N/2\)-by-\(N/2\) part of this array must contain the matrix \(B\).
On exit, if \(COMPQ = 'C'\) or \(COMPU = 'C'\), the leading \(N\)-by-\(N\) part of this array contains the upper triangular matrix \(BB\) in (3) (see also METHOD). The strictly lower triangular part is not zeroed.
If \(COMPQ = 'N'\) and \(COMPU = 'N'\), this array is unchanged on exit.

LDB INTEGER
The leading dimension of the array \(B\). \(LDB \geq \max(1, N)\).

\(FG\)  (input/output) COMPLEX*16 array, dimension (LDFG, N)
On entry, the leading \(N/2\)-by-\(N/2\) lower triangular part of this array must contain the lower triangular part of the Hermitian matrix \(G\), and the \(N/2\)-by-\(N/2\) upper triangular part of the submatrix in the columns \(2\) to \(N/2+1\) of this array must contain the upper triangular part of the Hermitian matrix \(F\).
On exit, if \(COMPQ = 'C'\) or \(COMPU = 'C'\), the leading \(N\)-by-\(N\) part of this array contains the Hermitian matrix \(BF\) in (3) (see also METHOD). The strictly lower triangular part of the input matrix is preserved. The diagonal elements might have tiny imaginary parts.
If \(COMPQ = 'N'\) and \(COMPU = 'N'\), this array is unchanged on exit.

LDFG INTEGER
The leading dimension of the array \(FG\). \(LDFG \geq \max(1, N)\).

\(NEIG\)  (output) INTEGER
If \(COMPQ = 'C'\) or \(COMPU = 'C'\), the number of eigenvalues in \(aS - bH\) with strictly negative real part.

\(D\)  (output) COMPLEX*16 array, dimension (LDD, N)
If \(COMPQ = 'C'\) or \(COMPU = 'C'\), the leading \(N\)-by-\(N\) part of this array contains the matrix \(BD\) in (3) (see METHOD).
If \(COMPQ = 'N'\) and \(COMPU = 'N'\), this array is not referenced.

LDD INTEGER
The leading dimension of the array D.
LDG >= 1,  if COMPQ = 'N' and COMPU = 'N';
LDG >= MAX(1, N), if COMPQ = 'C' or COMPU = 'C'.

C (output) COMPLEX*16 array, dimension (LDG, N)
If COMPQ = 'C' or COMPU = 'C', the leading N-by-N part of
this array contains the lower triangular matrix B in (3)
(see also METHOD). The strictly upper triangular part is
not zeroed.
If COMPQ = 'N' and COMPU = 'N', this array is not
referenced.

LDC INTEGER
The leading dimension of the array C.
LDC >= 1,  if COMPQ = 'N' and COMPU = 'N';
LDC >= MAX(1, N), if COMPQ = 'C' or COMPU = 'C'.

C Q (output) COMPLEX*16 array, dimension (LDQ, 2*N)
On exit, if COMPQ = 'C', the leading N-by-NEIG part of
this array contains an orthonormal basis of the right
deflating subspace corresponding to the eigenvalues of the
pencil aS - bH with strictly negative real part.
The remaining entries are meaningless.
If COMPQ = 'N', this array is not referenced.

LDQ INTEGER
The leading dimension of the array Q.
LDQ >= 1,  if COMPQ = 'N';
LDQ >= MAX(1, 2*N), if COMPQ = 'C'.

C U (output) COMPLEX*16 array, dimension (LDU, 2*N)
On exit, if COMPU = 'C', the leading N-by-NEIG part of
this array contains an orthonormal basis of the companion
subspace corresponding to the eigenvalues of the
class aS - bH with strictly negative real part. The
remaining entries are meaningless.
If COMPU = 'N', this array is not referenced.

LDU INTEGER
The leading dimension of the array U.
LDU >= 1,  if COMPU = 'N';
LDU >= MAX(1, N), if COMPU = 'C'.

C ALPHAR (output) DOUBLE PRECISION array, dimension (N)
The real parts of each scalar alpha defining an eigenvalue
of the pencil aS - bH.

C ALPHAI (output) DOUBLE PRECISION array, dimension (N)
The imaginary parts of each scalar alpha defining an
eigenvalue of the pencil aS - bH.
If ALPHAI(j) is zero, then the j-th eigenvalue is real.

C BETA (output) DOUBLE PRECISION array, dimension (N)
The scalars beta that define the eigenvalues of the pencil
aS - bH.
Together, the quantities alpha = (ALPHAR(j), ALPHAI(j)) and
beta = BETA(j) represent the j-th eigenvalue of the pencil
aS - bH, in the form lambda = alpha/beta. Since lambda may
overflow, the ratios should not, in general, be computed.

Workspace

17
IWORK INTEGER array, dimension (LIWORK)
C
LIWORK INTEGER
The dimension of the array IWORK. LIWORK >= 2*N+9.
C
DWORK DOUBLE PRECISION array, dimension (LDWORK)
On exit, if INFO = 0, DWORK(1) returns the optimal LDWORK.
On exit, if INFO = -26, DWORK(1) returns the minimum value of LDWORK.
C
LDWORK INTEGER
The dimension of the array DWORK.
LDWORK >= c*N**2 + N + MAX(2*N, 24) + 3, where
c = 18, if COMPU = 'C';
c = 16, if COMPQ = 'C' and COMPU = 'N';
c = 13, if COMPQ = 'N' and COMPU = 'N'.
For good performance LDWORK should be generally larger.
C
If LDWORK = -1, then a workspace query is assumed;
the routine only calculates the optimal size of the DWORK array, returns this value as the first entry of the DWORK array, and no error message related to LDWORK is issued by XERBLA.
C
ZWORK COMPLEX*16 array, dimension (LZWORK)
On exit, if INFO = 0, ZWORK(1) returns the optimal LZWORK.
On exit, if INFO = -28, ZWORK(1) returns the minimum value of LZWORK.
C
LZWORK INTEGER
The dimension of the array ZWORK.
LZWORK >= 8*N + 28, if COMPQ = 'C';
LZWORK >= 6*N + 28, if COMPQ = 'N' and COMPU = 'C';
LZWORK >= 1, if COMPQ = 'N' and COMPU = 'N'.
For good performance LZWORK should be generally larger.
C
If LZWORK = -1, then a workspace query is assumed;
the routine only calculates the optimal size of the ZWORK array, returns this value as the first entry of the ZWORK array, and no error message related to LZWORK is issued by XERBLA.
C
BWORK LOGICAL array, dimension (LBWORK)
LBWORK >= 0, if COMPQ = 'N' and COMPU = 'N';
LBWORK >= N, if COMPQ = 'C' or COMPU = 'C'.
C
Error Indicator
INFO INTEGER
= 0: successful exit;
< 0: if INFO = -i, the i-th argument had an illegal value;
= 1: the algorithm was not able to reveal information about the eigenvalues from the 2-by-2 blocks in the SLICOT Library routine MB03BD (called by DGHFST);
= 2: periodic QZ iteration failed in the SLICOT Library routines MB03HD or MB03EZ when trying to triangularize the 2-by-2 blocks;
= 3: the singular value decomposition failed in the LAPACK routine ZGESVD (for ORTH = 'S').
C
METHOD
First $T = iH$ is set. Then, the embeddings, $B_Z$ and $B_T$, of the matrices $S$ and $T$, are determined and, subsequently, the routine DGHFST is applied to compute the structured Schur form, i.e., the factorizations
\[
B_Z = U' B_Z Q = \begin{pmatrix} BZ11 & BZ12 \\ 0 & BZ22 \end{pmatrix} \quad \text{and} \quad B_T = J Q' J' B_T Q = \begin{pmatrix} T11 & T12 \\ 0 & T11' \end{pmatrix},
\]
where $Q$ is real orthogonal, $U$ is real orthogonal symplectic, $BZ11$, $BZ22'$ are upper triangular, and $T11$ is upper quasi-triangular.

Second, the routine ZGHFXC is applied, to compute a unitary matrix $Q$ and a unitary symplectic matrix $U$, such that
\[
U' B_Z Q = \begin{pmatrix} Z11 & Z12 \\ 0 & Z22 \end{pmatrix}, \quad J Q' J'(-i*B_T) Q = \begin{pmatrix} H11 & H12 \\ 0 & -H11' \end{pmatrix},
\]
with $Z11$, $Z22'$, $H11$ upper triangular, and such that the spectrum $\text{Spec}_{-}(J B_Z' J' B_Z, -i*B_T)$ is contained in the spectrum of the 2*NEIG-by-2*NEIG leading principal subpencil $aZ22'*Z11 - bH11$.

Finally, the right deflating subspace and the companion subspace are computed. See also page 21 in [1] for more details.

REFERENCES


NUMERICAL ASPECTS

The algorithm is numerically backward stable and needs $O(N)$ complex floating point operations.

FURTHER COMMENTS

This routine does not perform any scaling of the matrices. Scaling might sometimes be useful, and it should be done externally.

A.4 The Subroutine ZGHUDF

SUBROUTINE ZGHUDF( COMPQ, ORTH, N, A, LDA, DE, LDDE, B, LDB, FG, 
LDFG, NEIG, Q, LDQ, ALPHAR, ALPHAI, BETA, 
IWORK, DWORK, LDWORK, ZWORK, LZWORK, BDWORK, 
... )
$INFO$

PURPOSE

To compute the eigenvalues of a complex N-by-N skew-Hamiltonian/
Hamiltonian pencil $aS - bH$, with

\[
\begin{pmatrix}
A & D \\
E & A'
\end{pmatrix}
\begin{pmatrix}
B & F \\
G & -B'
\end{pmatrix}
\]

$S = (\cdot)$ and $H = (\cdot)$. \hspace{1cm} (1)

The structured Schur form of the embedded real skew-Hamiltonian/
skew-Hamiltonian pencil $aB_S - bB_T$, defined as

\[
\begin{pmatrix}
\text{Re}(A) & -\text{Im}(A) \\
\text{Im}(A) & \text{Re}(A)
\end{pmatrix}
\begin{pmatrix}
\text{Re}(D) & -\text{Im}(D) \\
\text{Im}(D) & \text{Re}(D)
\end{pmatrix}
\]

$B_S = \ldots$, and

\[
\begin{pmatrix}
\text{Re}(E) & -\text{Im}(E) \\
\text{Im}(E) & \text{Re}(E)
\end{pmatrix}
\begin{pmatrix}
\text{Re}(A') & \text{Im}(A') \\
-\text{Im}(A') & \text{Re}(A')
\end{pmatrix}
\]

\[
\begin{pmatrix}
-\text{Im}(B) & -\text{Re}(B) \\
\text{Re}(B) & -\text{Im}(B)
\end{pmatrix}
\begin{pmatrix}
-\text{Im}(F) & -\text{Re}(F) \\
\text{Re}(F) & -\text{Im}(F)
\end{pmatrix}
\]

$B_T = \ldots$, $T = iH$, \hspace{1cm} (2)

is determined and used to compute the eigenvalues. The notation $M'$
denotes the conjugate transpose of the matrix $M$. Optionally, \hspace{1cm} (3)

if COMPQ = 'C', an orthonormal basis of the right deflating
subspace of the pencil $aS - bH$, corresponding to the eigenvalues
with strictly negative real part, is computed. Namely, after
transforming $aB_S - bB_H$ by unitary matrices, we have

\[
\begin{pmatrix}
BA & BD \\
RE & BF
\end{pmatrix}
\begin{pmatrix}
B_{Sout} = (\cdot) \\
B_{Hout} = (\cdot)
\end{pmatrix}
\]
and the eigenvalues with strictly negative real part of the
complex pencil \( aS - bH \) are moved to the top. The
embedding doubles the multiplicities of the eigenvalues of the
cell \( aS - bH \).

**ARGUMENTS**

**Mode Parameters**

**COMPQ CHARACTER**
- Specifies whether to compute the deflating subspace corresponding to the eigenvalues of \( aS - bH \) with strictly negative real part.
- \( 'N' \): do not compute the deflating subspace; compute the eigenvalues only;
- \( 'C' \): compute the deflating subspace and store it in the leading subarray of \( Q \).

**ORTH CHARACTER**
- If \( COMPQ = 'C' \), specifies the technique for computing an orthonormal basis of the deflating subspace, as follows:
- \( 'P' \): QR factorization with column pivoting;
- \( 'S' \): singular value decomposition.
- If \( COMPQ = 'N' \), the ORTH value is not used.

**Input/Output Parameters**

**N (input) INTEGER**
- The order of the pencil \( aS - bH \). \( N \geq 0 \), even.

**A (input/output) COMPLEX**
- On entry, the leading \( N/2 \)-by-\( N/2 \) part of this array must contain the matrix \( A \).
- On exit, if \( COMPQ = 'C' \), the leading \( N \)-by-\( N \) part of this array contains the upper triangular matrix \( BA \) in (3) (see also METHOD). The strictly lower triangular part is not zeroed; it is preserved in the leading \( N/2 \)-by-\( N/2 \) part.
- If \( COMPQ = 'N' \), this array is unchanged on exit.

**LDA INTEGER**
- The leading dimension of the array \( A \). \( LDA \geq MAX(1, N) \).

**DE (input/output) COMPLEX**
- On entry, the leading \( N/2 \)-by-\( N/2 \) lower triangular part of this array must contain the lower triangular part of the skew-Hermitian matrix \( E \), and the \( N/2 \)-by-\( N/2 \) upper triangular part of the submatrix in the columns 2 to \( N/2+1 \) of this array must contain the upper triangular part of the skew-Hermitian matrix \( D \).
- On exit, if \( COMPQ = 'C' \), the leading \( N \)-by-\( N \) part of this array contains the skew-Hermitian matrix \( BD \) in (3) (see also METHOD). The strictly lower triangular part of the input matrix is preserved.
- If \( COMPQ = 'N' \), this array is unchanged on exit.

**LDDE INTEGER**
- The leading dimension of the array \( DE \). \( LDDE \geq MAX(1, N) \).

**B (input/output) COMPLEX**
- On entry, the leading \( N/2 \)-by-\( N/2 \) part of this array must...
contain the matrix B.
On exit, if COMPQ = 'C', the leading N-by-N part of this
array contains the upper triangular matrix BB in (3) (see
also METHOD). The strictly lower triangular part is not
zeroed; the elements below the first subdiagonal of the
input matrix are preserved.
If COMPQ = 'N', this array is unchanged on exit.

LDB INTEGER
The leading dimension of the array B. LDB >= MAX(1, N).

FG (input/output) COMPLEX*16 array, dimension (LDFG, N)
On entry, the leading N/2-by-N/2 lower triangular part of
this array must contain the lower triangular part of the
Hermitian matrix G, and the N/2-by-N/2 upper triangular
part of the submatrix in the columns 2 to N/2+1 of this
array must contain the upper triangular part of the
Hermitian matrix F.
On exit, if COMPQ = 'C', the leading N-by-N part of this
array contains the Hermitian matrix BF in (3) (see also
METHOD). The strictly lower triangular part of the input
matrix is preserved. The diagonal elements might have tiny
imaginary parts.
If COMPQ = 'N', this array is unchanged on exit.

LDFG INTEGER
The leading dimension of the array FG. LDFG >= MAX(1, N).

NEIG (output) INTEGER
If COMPQ = 'C', the number of eigenvalues in aS - bH with

Q (output) COMPLEX*16 array, dimension (LDQ, 2*N)
On exit, if COMPQ = 'C', the leading N-by-NEIG part of
this array contains an orthonormal basis of the right
deflating subspace corresponding to the eigenvalues of the
pencil aS - bH with strictly negative real part.
The remaining entries are meaningless.
If COMPQ = 'N', this array is not referenced.

LDQ INTEGER
The leading dimension of the array Q.
LDQ >= 1, if COMPQ = 'N';
LDQ >= MAX(1, 2*N), if COMPQ = 'C'.

ALPHAR (output) DOUBLE PRECISION array, dimension (N)
The real parts of each scalar alpha defining an eigenvalue
of the pencil aS - bH.

ALPHAI (output) DOUBLE PRECISION array, dimension (N)
The imaginary parts of each scalar alpha defining an
eigenvalue of the pencil aS - bH.
If ALPHAI(j) is zero, then the j-th eigenvalue is real.

BETA (output) DOUBLE PRECISION array, dimension (N)
The scalars beta that define the eigenvalues of the pencil
aS - bH.
Together, the quantities alpha = (ALPHAR(j),ALPHAI(j)) and
beta = BETA(j) represent the j-th eigenvalue of the pencil
aS - bH, in the form lambda = alpha/beta. Since lambda may
overflow, the ratios should not, in general, be computed.

Workspace

22
C IWORK INTEGER array, dimension (LIWORK)
LIWORK >= N, if ORTH = 'P';
LIWORK >= 0, if ORTH <> 'P'.

C DWORK DOUBLE PRECISION array, dimension (LDWORK)
On exit, if INFO = 0, DWORK(1) returns the optimal LDWORK.
On exit, if INFO = -20, DWORK(1) returns the minimum value
of LDWORK.

C LDWORK INTEGER
The dimension of the array DWORK.
LDWORK >= MAX( 1, 5*N*N + 3*N ), if COMPQ = 'N';
LDWORK >= MAX( 1, 11*N*N + 2*N ), if COMPQ = 'C'.
For good performance LDWORK should be generally larger.

If LDWORK = -1, then a workspace query is assumed;
the routine only calculates the optimal size of the
DWORK array, returns this value as the first entry of
the DWORK array, and no error message related to LDWORK
is issued by XERBLA.

C ZWORK COMPLEX*16 array, dimension (LZWORK)
On exit, if INFO = 0, ZWORK(1) returns the optimal LZWORK.
On exit, if INFO = -22, ZWORK(1) returns the minimum
value of LZWORK.

C LZWORK INTEGER
The dimension of the array ZWORK.
LZWORK >= 1, if COMPQ = 'N';
LZWORK >= 8*N + 4, if COMPQ = 'C'.
For good performance LZWORK should be generally larger.

If LZWORK = -1, then a workspace query is assumed;
the routine only calculates the optimal size of the
ZWORK array, returns this value as the first entry of
the ZWORK array, and no error message related to LZWORK
is issued by XERBLA.

C BWORK LOGICAL array, dimension (LBWORK)
LBWORK >= 0, if COMPQ = 'N';
LBWORK >= N, if COMPQ = 'C'.

Error Indicator
C INFO INTEGER
= 0: successful exit;
< 0: if INFO = -i, the i-th argument had an illegal value;
= 1: QZ iteration failed in the routine DGHUST (QZ
iteration did not converge or computation of the
shifts failed);
= 2: QZ iteration failed in the LAPACK routine ZHGEQZ when
trying to triangularize the 2-by-2 blocks;
= 3: the singular value decomposition failed in the LAPACK
routine ZGESVD (for ORTH = 'S').

METHOD
First, T = i*H is set. Then, the embeddings, B_S and B_T, of the
matrices S and T, are determined and, subsequently, the routine
DGHUST is applied to compute the structured Schur form, i.e., the
factorizations
Second, the routine ZGHUXC is applied, to compute a unitary matrix $Q$, such that

$\begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{11}' \end{pmatrix} = \begin{pmatrix} J \end{pmatrix} Q' \begin{pmatrix} J' \end{pmatrix} B_S Q = \begin{pmatrix} \ast \end{pmatrix} =: B_{S_{\text{out}}}$,

$\begin{pmatrix} H_{11} & H_{12} \\ 0 & -H_{11}' \end{pmatrix} = \begin{pmatrix} J \end{pmatrix} Q' \begin{pmatrix}-iB_T\end{pmatrix} Q = \begin{pmatrix} \ast \end{pmatrix} =: B_{H_{\text{out}}}$,

with $S_{11}$, $H_{11}$ upper triangular, and such that $\text{Spec}_{-\epsilon}(B_S, -iB_T)$ is contained in the spectrum of the $2\times\text{NEIG}$-by-$2\times\text{NEIG}$ leading principal subpencil $aS_{11} - bH_{11}$.

Finally, the right deflating subspace is computed. See also page 22 in [1] for more details.

REFERENCES

Numerical Computation of Deflating Subspaces of Embedded Hamiltonian Pencils.

NUMERICAL ASPECTS

The algorithm is numerically backward stable and needs $O(N^3)$ complex floating point operations.

FURTHER COMMENTS

This routine does not perform any scaling of the matrices. Scaling might sometimes be useful, and it should be done externally.
Figure 4: Call graph of ZGHUDF.