NAG Fortran Library Routine Document

F08PSF (CHSEQR/ZHSEQR)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

Warning. The specification of the parameter LWORK changed at Mark 20: LWORK is no longer redundant.

1 Purpose

F08PSF (CHSEQR/ZHSEQR) computes all the eigenvalues, and optionally the Schur factorization, of a complex Hessenberg matrix or a complex general matrix which has been reduced to Hessenberg form.

2 Specification

```
 SUBROUTINE F08PSF(JOB, COMPZ, N, ILO, IHI, H, LDH, W, Z, LDZ, WORK,
1                  LWORK, INFO)
 ENTRY      chseqr (JOB, COMPZ, N, ILO, IHI, H, LDH, W, Z, LDZ, WORK,
1                  LWORK, INFO)
 INTEGER           N, ILO, IHI, LDH, LDZ, LWORK, INFO
 complex           H(LDH,*), W(*), Z(LDZ,*), WORK(*)
 CHARACTER*1       JOB, COMPZ
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine computes all the eigenvalues, and optionally the Schur factorization, of a complex upper Hessenberg matrix H:

$$H = ZTZ^H,$$

where T is an upper triangular matrix (the Schur form of H), and Z is the unitary matrix whose columns are the Schur vectors z_i. The diagonal elements of T are the eigenvalues of H.

The routine may also be used to compute the Schur factorization of a complex general matrix A which has been reduced to upper Hessenberg form H:

$$\begin{aligned} A &= QHQ^H, \text{ where } Q \text{ is unitary,} \\ &= (QZ)T(QZ)^H. \end{aligned}$$

In this case, after F08NSF (CGEHRD/ZGEHRD) has been called to reduce A to Hessenberg form, F08NTF (CUNGHR/ZUNGHR) must be called to form Q explicitly; Q is then passed to F08PSF (CHSEQR/ZHSEQR), which must be called with COMPZ = 'V'.

The routine can also take advantage of a previous call to F08NVF (CGEBAL/ZGEBAL) which may have balanced the original matrix before reducing it to Hessenberg form, so that the Hessenberg matrix H has the structure:

$$\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ & H_{22} & H_{23} \\ & & H_{33} \end{pmatrix}$$

where H_{11} and H_{33} are upper triangular. If so, only the central diagonal block H_{22} (in rows and columns i_{lo} to i_{hi}) needs to be further reduced to Schur form (the blocks H_{12} and H_{23} are also affected). Therefore the values of i_{lo} and i_{hi} can be supplied to F08PSF (CHSEQR/ZHSEQR) directly. Also, F08NWF (CGEBAK/ZGEBAK) must be called after this routine to permute the Schur vectors of the balanced matrix to those of the original matrix. If F08NVF (CGEBAL/ZGEBAL) has not been called however, then i_{lo} must be set to 1 and i_{hi} to n. Note that if the Schur factorization of A is required, F08NVF (CGEBAL/ZGEBAL) must **not** be called with JOB = 'S' or 'B', because the balancing transformation is not unitary.

F08PSF (CHSEQR/ZHSEQR) uses a multishift form of the upper Hessenberg QR algorithm, due to Bai and Demmel (1989). The Schur vectors are normalized so that $\|z_i\|_2 = 1$, but are determined only to within a complex factor of absolute value 1.

4 References

Bai Z and Demmel J W (1989) On a block implementation of Hessenberg multishift QR iteration *Internat. J. High Speed Comput.* **1** 97–112

Golub G H and van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: indicates whether eigenvalues only or the Schur form T is required, as follows:

if JOB = 'E', eigenvalues only are required;

if JOB = 'S', the Schur form T is required.

Constraint: JOB = 'E' or 'S'.

2: COMPZ – CHARACTER*1 *Input*

On entry: indicates whether the Schur vectors are to be computed as follows:

if COMPZ = 'N', no Schur vectors are computed (and the array Z is not referenced);

if COMPZ = 'I', the Schur vectors of H are computed (and the array Z is initialised by the routine);

if COMPZ = 'V', the Schur vectors of A are computed (and the array Z must contain the matrix Q on entry).

Constraint: COMPZ = 'N', 'I' or 'V'.

3: N – INTEGER *Input*

On entry: n, the order of the matrix H.

Constraint: N ≥ 0.

4: ILO – INTEGER *Input*
5: IHI – INTEGER *Input*

On entry: if the matrix A has been balanced by F08NVF (CGEBAL/ZGEBAL), then ILO and IHI must contain the values returned by that routine. Otherwise, ILO must be set to 1 and IHI to N.

Constraints:

ILO ≥ 1 and
$\min(\text{ILO}, \text{N}) \leq \text{IHI} \leq \text{N}$.

6: H(LDH,*) – ***complex*** array *Input/Output*

Note: the second dimension of the array H must be at least $\max(1, \text{N})$.

On entry: the n by n upper Hessenberg matrix H, as returned by F08NSF (CGEHRD/ZGEHRD).

On exit: if JOB = 'E', the array contains no useful information. If JOB = 'S', H is overwritten by the upper triangular matrix T from the Schur decomposition (the Schur form) unless INFO > 0.

7: LDH – INTEGER *Input*

On entry: the first dimension of the array H as declared in the (sub)program from which F08PSF (CHSEQR/ZHSEQR) is called.

Constraint: LDH $\geq \max(1, \mathrm{N})$.

8: W($*$) – ***complex*** array *Output*

Note: the dimension of the array W must be at least $\max(1, \mathrm{N})$.

On exit: the computed eigenvalues, unless INFO > 0 (in which case see Section 6). The eigenvalues are stored in the same order as on the diagonal of the Schur form T (if computed).

9: Z(LDZ,*) – ***complex*** array *Input/Output*

Note: the second dimension of the array Z must be at least $\max(1, \mathrm{N})$ if COMPZ = 'V' or 'I' and at least 1 if COMPZ = 'N'.

On entry: if COMPZ = 'V', Z must contain the unitary matrix Q from the reduction to Hessenberg form; if COMPZ = 'I', Z need not be set.

On exit: if COMPZ = 'V' or 'I', Z contains the unitary matrix of the required Schur vectors, unless INFO > 0.

Z is not referenced if COMPZ = 'N'.

10: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08PSF (CHSEQR/ZHSEQR) is called.

Constraints:

LDZ ≥ 1 if COMPZ = 'N',
LDZ $\geq \max(1, \mathrm{N})$ if COMPZ = 'V' or 'I'.

11: WORK($*$) – ***complex*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \mathrm{LWORK})$.

On exit: if INFO = 0, the real part of WORK(1) contains the minimum value of LWORK required for optimum performance.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08PSF (CHSEQR/ZHSEQR) is called, unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the minimum dimension of WORK.

Constraint: LWORK $\geq \max(1, \mathrm{N})$ or LWORK $= -1$.

13: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

The algorithm has failed to find all the eigenvalues after a total of $30 \times (\text{IHI} - \text{ILO} + 1)$ iterations. If INFO $= i$, elements $1, 2, \ldots, \text{ILO} - 1$ and $i + 1, i + 2, \ldots, n$ of W contain the eigenvalues which have been found.

7 Accuracy

The computed Schur factorization is the exact factorization of a nearby matrix $H + E$, where

$$\|E\|_2 = O(\epsilon)\|H\|_2,$$

and ϵ is the ***machine precision***.

If λ_i is an exact eigenvalue, and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$|\tilde{\lambda}_i - \lambda_i| \leq \frac{c(n)\epsilon\|H\|_2}{s_i},$$

where $c(n)$ is a modestly increasing function of n, and s_i is the reciprocal condition number of λ_i. The condition numbers s_i may be computed by calling F08QYF (CTRSNA/ZTRSNA).

8 Further Comments

The total number of real floating-point operations depends on how rapidly the algorithm converges, but is typically about:

$25n^3$ if only eigenvalues are computed;

$35n^3$ if the Schur form is computed;

$70n^3$ if the full Schur factorization is computed.

The real analogue of this routine is F08PEF (SHSEQR/DHSEQR).

9 Example

To compute all the eigenvalues and the Schur factorization of the upper Hessenberg matrix H, where

$$H = \begin{pmatrix} -3.9700 - 5.0400i & -1.1318 - 2.5693i & -4.6027 - 0.1426i & -1.4249 + 1.7330i \\ -5.4797 + 0.0000i & 1.8585 - 1.5502i & 4.4145 - 0.7638i & -0.4805 - 1.1976i \\ 0.0000 + 0.0000i & 6.2673 + 0.0000i & -0.4504 - 0.0290i & -1.3467 + 1.6579i \\ 0.0000 + 0.0000i & 0.0000 + 0.0000i & -3.5000 + 0.0000i & 2.5619 - 3.3708i \end{pmatrix}.$$

See also F08NTF (CUNGHR/ZUNGHR), which illustrates the use of this routine to compute the Schur factorization of a general matrix.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08PSF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDH, LWORK, LDZ
      PARAMETER        (NMAX=8,LDH=NMAX,LWORK=NMAX,LDZ=NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
*     .. Local Arrays ..
      complex          H(LDH,NMAX), W(NMAX), WORK(LWORK), Z(LDZ,NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
```

```
      EXTERNAL         X04DBF, chseqr
*     .. Intrinsic Functions ..
      INTRINSIC        real, imag
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08PSF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read H from data file
*
         READ (NIN,*) ((H(I,J),J=1,N),I=1,N)
*
*        Calculate the eigenvalues and Schur factorization of H
*
         CALL chseqr('Schur form','Initialize Z',N,1,N,H,LDH,W,Z,LDZ,
     +               WORK,LWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (' (',real(W(I)),',',imag(W(I)),')',I=1,
     +        N)
*
*           Print Schur form
*
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04DBF('General',' ',N,N,H,LDH,'Bracketed','F7.4',
     +                  'Schur form','Integer',RLABS,'Integer',CLABS,80,
     +                  0,IFAIL)
*
*           Print Schur vectors
*
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04DBF('General',' ',N,N,Z,LDZ,'Bracketed','F7.4',
     +                  'Schur vectors of H','Integer',RLABS,'Integer',
     +                  CLABS,80,0,IFAIL)
*
         END IF
      END IF
      STOP
*
99999 FORMAT ((3X,4(A,F7.4,A,F7.4,A,:)))
      END
```

9.2 Program Data

```
F08PSF Example Program Data
  4                                                             :Value of N
 (-3.9700,-5.0400) (-1.1318,-2.5693) (-4.6027,-0.1426) (-1.4249, 1.7330)
 (-5.4797, 0.0000) ( 1.8585,-1.5502) ( 4.4145,-0.7638) (-0.4805,-1.1976)
 ( 0.0000, 0.0000) ( 6.2673, 0.0000) (-0.4504,-0.0290) (-1.3467, 1.6579)
 ( 0.0000, 0.0000) ( 0.0000, 0.0000) (-3.5000, 0.0000) ( 2.5619,-3.3708)
                                                        :End of matrix H
```

9.3 Program Results

```
 F08PSF Example Program Results

 Eigenvalues
    (-6.0004,-6.9998) (-5.0000, 2.0060) ( 7.9982,-0.9964) ( 3.0023,-3.9998)

 Schur form
                     1                 2                 3                 4
 1  (-6.0004,-6.9998) (-0.2080, 0.4719) (-0.4829, 0.1768) ( 0.1301, 0.9052)
 2  ( 0.0000, 0.0000) (-5.0000, 2.0060) (-0.6653, 0.2814) ( 0.0038, 0.2639)
 3  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 7.9982,-0.9964) ( 0.2004, 1.0595)
 4  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 3.0023,-3.9998)

 Schur vectors of H
                     1                 2                 3                 4
 1  ( 0.8457, 0.0000) ( 0.1380, 0.3602) (-0.2677,-0.1091) (-0.2213,-0.0582)
 2  ( 0.2824,-0.3304) (-0.4612, 0.2075) ( 0.6846, 0.0000) ( 0.2927, 0.0320)
 3  ( 0.0748, 0.2800) ( 0.7239, 0.0000) ( 0.5924,-0.0189) (-0.0229, 0.2005)
 4  ( 0.0670, 0.0860) ( 0.2169, 0.1560) (-0.2745, 0.1454) ( 0.9057, 0.0000)
```