

NAG Library Routine Document

F08PEF (DHSEQR)

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

F08PEF (DHSEQR) computes all the eigenvalues and, optionally, the Schur factorization of a real Hessenberg matrix or a real general matrix which has been reduced to Hessenberg form.

2 Specification

```
SUBROUTINE F08PEF(JOB, COMPZ, N, ILO, IHI, H, LDH, WR, WI, Z, LDZ, WORK,
1                LWORK, INFO)
    INTEGER          N, ILO, IHI, LDH, LDZ, LWORK, INFO
    double precision H(LDH,*), WR(*), WI(*), Z(LDZ,*), WORK(max(1,LWORK))
    CHARACTER*1      JOB, COMPZ
```

The routine may be called by its LAPACK name ***dhseqr***.

3 Description

F08PEF (DHSEQR) computes all the eigenvalues and, optionally, the Schur factorization of a real upper Hessenberg matrix H :

$$H = ZTZ^T,$$

where T is an upper quasi-triangular matrix (the Schur form of H), and Z is the orthogonal matrix whose columns are the Schur vectors z_i . See Section 8 for details of the structure of T .

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

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

In this case, after F08NEF (DGEHRD) has been called to reduce A to Hessenberg form, F08NFF (DORGHR) must be called to form Q explicitly; Q is then passed to F08PEF (DHSEQR), which must be called with $\text{COMPZ} = 'V'$.

The routine can also take advantage of a previous call to F08NHF (DGEBAL) 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 F08PEF (DHSEQR) directly. Also, F08NJF (DGEBAL) must be called after this routine to permute the Schur vectors of the balanced matrix to those of the original matrix. If F08NHF (DGEBAL) 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, F08NHF (DGEBAL) must **not** be called with $\text{JOB} = 'S'$ or $'B'$, because the balancing transformation is not orthogonal.

F08PEF (DHSEQR) 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 factor ± 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.
 JOB = 'E'
 Eigenvalues only are required.
 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.
 COMPZ = 'N'
 No Schur vectors are computed (and the array Z is not referenced).
 COMPZ = 'I'
 The Schur vectors of H are computed (and the array Z is initialized by the routine).
 COMPZ = 'V'
 The Schur vectors of A are computed (and the array Z must contain the matrix Q on entry).
Constraint: COMPZ = 'N', 'V' or 'I'.
- 3: N – INTEGER *Input*
On entry: n , the order of the matrix H .
Constraint: $N \geq 0$.
- 4: ILO – INTEGER *Input*
- 5: IHI – INTEGER *Input*
On entry: if the matrix A has been balanced by F08NHF (DGEBAL), then ILO and IHI must contain the values returned by that routine. Otherwise, ILO must be set to 1 and IHI to N.
Constraint: $ILO \geq 1$ and $\min(ILO, N) \leq IHI \leq N$.
- 6: H(LDH,*) – **double precision** array *Input/Output*
Note: the second dimension of the array H must be at least $\max(1, N)$.
On entry: the n by n upper Hessenberg matrix H , as returned by F08NEF (DGEHRD).
On exit: if JOB = 'E', the array contains no useful information.
 If JOB = 'S', H is overwritten by the upper quasi-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 F08PEF (DHSEQR) is called.
Constraint: $LDH \geq \max(1, N)$.
- 8: WR(*) – **double precision** array *Output*
9: WI(*) – **double precision** array *Output*
Note: the dimension of the arrays WR and WI must be at least $\max(1, N)$.
On exit: the real and imaginary parts, respectively, of the computed eigenvalues, unless $INFO > 0$ (in which case see Section 6). Complex conjugate pairs of eigenvalues appear consecutively with the eigenvalue having positive imaginary part first. The eigenvalues are stored in the same order as on the diagonal of the Schur form T (if computed); see Section 8 for details.
- 10: Z(LDZ,*) – **double precision** array *Input/Output*
Note: the second dimension of the array Z must be at least $\max(1, N)$ if $COMPZ = 'V'$ or $'I'$ and at least 1 if $COMPZ = 'N'$.
On entry: if $COMPZ = 'V'$, Z must contain the orthogonal 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 orthogonal matrix of the required Schur vectors, unless $INFO > 0$.
If $COMPZ = 'N'$, Z is not referenced.
- 11: LDZ – INTEGER *Input*
On entry: the first dimension of the array Z as declared in the (sub)program from which F08PEF (DHSEQR) is called.
Constraints:
if $COMPZ = 'V'$ or $'I'$, $LDZ \geq \max(1, N)$;
if $COMPZ = 'N'$, $LDZ \geq 1$.
- 12: WORK($\max(1, LWORK)$) – **double precision** array *Workspace*
On exit: if $INFO = 0$, WORK(1) contains the minimum value of LWORK required for optimal performance.
- 13: LWORK – INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F08PEF (DHSEQR) is called, unless $LWORK = -1$, in which case a workspace query is assumed and the routine only calculates the minimum dimension of WORK.
Constraint: $LWORK \geq \max(1, N)$ or $LWORK = -1$.
- 14: 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 $INFO = -i$, argument i 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 $\text{INFO} = i$, elements $1, 2, \dots, \text{ILO} - 1$ and $i + 1, i + 2, \dots, n$ of WR and WI contain the real and imaginary parts of contain the eigenvalues which have been found.

If JOB = 'E', then on exit, the remaining unconverged eigenvalues are the eigenvalues of the upper Hessenberg matrix $\hat{H}$, formed from $H(\text{ILO} : \text{INFO}, \text{ILO} : \text{INFO})$, i.e., the ILO through INFO rows and columns of the final output matrix H .

If JOB = 'S', then on exit

$$(*) \quad H_i U = U \tilde{H}$$

for some matrix U , where H_i is the input upper Hessenberg matrix and $\tilde{H}$ is an upper Hessenberg matrix formed from $H(\text{INFO} + 1 : \text{IHI}, \text{INFO} + 1 : \text{IHI})$.

If COMPZ = 'V', then on exit

$$Z_{\text{out}} = Z_{\text{in}} U$$

where U is defined in (*) (regardless of the value of JOB).

If COMPZ = 'T', then on exit

$$Z_{\text{out}} = U$$

where U is defined in (*) (regardless of the value of JOB).

If INFO > 0 and COMPZ = 'N', then Z is not accessed.

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 F08QLF (DTRSNA).

8 Further Comments

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

$7n^3$ if only eigenvalues are computed;

$10n^3$ if the Schur form is computed;

$20n^3$ if the full Schur factorization is computed.

The Schur form T has the following structure (referred to as **canonical** Schur form).

If all the computed eigenvalues are real, T is upper triangular, and the diagonal elements of T are the eigenvalues; $\text{WR}(i) = t_{ii}$, for $i = 1, 2, \dots, n$ and $\text{WI}(i) = 0.0$.

If some of the computed eigenvalues form complex conjugate pairs, then T has 2 by 2 diagonal blocks. Each diagonal block has the form

$$\begin{pmatrix} t_{ii} & t_{i,i+1} \\ t_{i+1,i} & t_{i+1,i+1} \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \alpha \end{pmatrix}$$

where $\beta\gamma < 0$. The corresponding eigenvalues are $\alpha \pm \sqrt{\beta\gamma}$; $\text{WR}(i) = \text{WR}(i+1) = \alpha$; $\text{WI}(i) = +\sqrt{|\beta\gamma|}$; $\text{WI}(i+1) = -\text{WI}(i)$.

The complex analogue of this routine is F08PSF (ZHSEQR).

9 Example

This example computes all the eigenvalues and the Schur factorization of the upper Hessenberg matrix H , where

$$H = \begin{pmatrix} 0.3500 & -0.1160 & -0.3886 & -0.2942 \\ -0.5140 & 0.1225 & 0.1004 & 0.1126 \\ 0.0000 & 0.6443 & -0.1357 & -0.0977 \\ 0.0000 & 0.0000 & 0.4262 & 0.1632 \end{pmatrix}.$$

See also the example for F08NFF (DORGHR), which illustrates the use of this routine to compute the Schur factorization of a general matrix.

9.1 Program Text

```
*      F08PEF 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 ..
      DOUBLE PRECISION H(LDH,NMAX), WI(NMAX), WORK(LWORK), WR(NMAX),
+      Z(LDZ,NMAX)
*      .. External Subroutines ..
      EXTERNAL         DHSEQR, X04CAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F08PEF 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 DHSEQR('Schur form','Initialize Z',N,1,N,H,LDH,WR,WI,Z,
+      LDZ,WORK,LWORK,INFO)
*
      WRITE (NOUT,*)
      IF (INFO.GT.0) THEN
        WRITE (NOUT,*) 'Failure to converge.'
      ELSE
        WRITE (NOUT,*) 'Eigenvalues'
        WRITE (NOUT,99999) (' (' ,WR(I) ,',',WI(I) ,')' ,I=1,N)
*
*      Print Schur form
*
      WRITE (NOUT,*)
      IFAIL = 0
*
      CALL X04CAF('General',' ',N,N,H,LDH,'Schur form',IFAIL)
*
*      Print Schur vectors
*
      WRITE (NOUT,*)
```

```

      IFAIL = 0
*
      CALL X04CAF('General',' ',N,N,Z,LDZ,'Schur vectors of H',
+              IFAIL)
*
      END IF
      END IF
*
99999 FORMAT (1X,A,F8.4,A,F8.4,A)
      END

```

9.2 Program Data

F08PEF Example Program Data

4				:Value of N
0.3500	-0.1160	-0.3886	-0.2942	
-0.5140	0.1225	0.1004	0.1126	
0.0000	0.6443	-0.1357	-0.0977	
0.0000	0.0000	0.4262	0.1632	:End of matrix H

9.3 Program Results

F08PEF Example Program Results

Eigenvalues

```

( 0.7995, 0.0000)
(-0.0994, 0.4008)
(-0.0994, -0.4008)
(-0.1007, 0.0000)

```

Schur form

	1	2	3	4
1	0.7995	0.0061	-0.1144	-0.0335
2	0.0000	-0.0994	-0.6483	-0.2026
3	0.0000	0.2477	-0.0994	-0.3474
4	0.0000	0.0000	0.0000	-0.1007

Schur vectors of H

	1	2	3	4
1	-0.6551	-0.3450	-0.1036	0.6641
2	0.5972	-0.1706	0.5246	0.5823
3	0.3845	-0.7143	-0.5789	-0.0821
4	0.2576	0.5845	-0.6156	0.4616