NAG Fortran Library Routine Document

F02EAF

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.

1 Purpose

F02EAF computes all the eigenvalues, and optionally the Schur factorization, of a real general matrix.

2 Specification

```
SUBROUTINE F02EAF(JOB, N, A, LDA, WR, WI, Z, LDZ, WORK, LWORK, IFAIL)
INTEGER            N, LDA, LDZ, LWORK, IFAIL
real               A(LDA,*), WR(*), WI(*), Z(LDZ,*), WORK(LWORK)
CHARACTER*1        JOB
```

3 Description

This routine computes all the eigenvalues, and optionally the Schur form or the complete Schur factorization, of a real general matrix A:

$$A = ZTZ^T,$$

where T is an upper quasi-triangular matrix, and Z is an orthogonal matrix. T is called the *Schur form* of A, and the columns of Z are called the *Schur vectors*.

If it is desired to order the Schur factorization so that specified eigenvalues occur in the leading positions on the diagonal of T, then this routine may be followed by a call of F08QGF (STRSEN/DTRSEN). Other reorderings may be achieved by calls to F08QFF (STREXC/DTREXC).

4 References

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 the Schur form and Schur vectors are to be computed as follows:

if JOB = 'N', then only eigenvalues are computed;

if JOB = 'S', then eigenvalues and the Schur form T are computed;

if JOB = 'V', then eigenvalues, the Schur form and the Schur vectors are computed.

Constraint: JOB = 'N', 'S' or 'V'.

2: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

3: A(LDA,*) – ***real*** array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the n by n general matrix A.

On exit: if JOB = 'S' or 'V', A contains the upper quasi-triangular matrix T, the Schur form of A. If JOB = 'N', the contents of A are overwritten.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02EAF is called.

Constraint: LDA $\geq$ max(1, N).

5: WR(*) – ***real*** array *Output*

Note: the dimension of the array WR must be at least max(1, N).

6: WI(*) – ***real*** array *Output*

Note: the dimension of the array WI must be at least max(1, N).

On exit: the real and imaginary parts, respectively, of the computed eigenvalues. Complex conjugate pairs of eigenvalues are stored in consecutive elements of the arrays, with the eigenvalue having positive imaginary part first. If JOB = 'S' or 'V', the eigenvalues occur in the same order as on the diagonal of T, with complex conjugate pairs corresponding to 2 by 2 diagonal blocks.

7: Z(LDZ,*) – ***real*** array *Output*

Note: the second dimension of the array Z must be at least max(1, N) if JOB=V and at least 1 otherwise.

On exit: If JOB = 'V', Z contains the orthogonal matrix Z of Schur vectors. Z is not referenced if JOB = 'N' or 'S'.

8: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F02EAF is called.

Constraints:

LDZ ≥ 1 if JOB = 'N' or 'S',
LDZ $\geq$ max(1, N) if JOB = 'V'.

9: WORK(LWORK) – ***real*** array *Workspace*
10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02EAF is called. On some high-performance computers, increasing the dimension of WORK will enable the routine to run faster; a value of $64 \times$ N should allow near-optimal performance on almost all machines.

Constraint: LWORK $\geq$ max(1, 3 $\times$ N).

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, JOB ≠ 'N', 'S' or 'V',
or N < 0,
or LDA < max(1, N),
or LDZ < 1, or LDZ < N and JOB = 'V',
or LWORK < max(1, 3 × N).

IFAIL = 2

The QR algorithm failed to compute all the eigenvalues.

7 Accuracy

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

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

and ϵ is the ***machine precision***.

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

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

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 (STRSNA/DTRSNA).

8 Further Comments

The routine calls routines from LAPACK in Chapter F08. It first reduces A to upper Hessenberg form H, using an orthogonal similarity transformation: $A = QHQ^T$. If only eigenvalues or the Schur form are required, the routine uses the upper Hessenberg QR algorithm to compute the eigenvalues or Schur form of H. If the Schur vectors are required, the routine first forms the orthogonal matrix Q that was used in the reduction to Hessenberg form; it then uses the QR algorithm to reduce H to T, using further orthogonal transformations: $H = STS^T$, and at the same time accumulates the matrix of Schur vectors $Z = QS$.

If all the computed eigenvalues are real, then T is upper triangular, and the diagonal elements of T are the eigenvalues of A. If some of the computed eigenvalues form complex conjugate pairs, then T has 2 by 2 diagonal blocks. Each block has the form

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

where $\beta\gamma < 0$, and the corresponding eigenvalues are $\alpha \pm \sqrt{\beta\gamma}$.

Each Schur vector z is normalized so that $\|z\|_2 = 1$, and the element of largest absolute value is positive.

The time taken by the routine is approximately proportional to n^3.

9 Example

To compute the Schur factorization of the matrix A, where

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix}.$$

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.

```
*     F02EAF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDZ, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LDZ=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*     .. Local Arrays ..
      real             A(LDA,NMAX), WI(NMAX), WORK(LWORK), WR(NMAX),
     +                 Z(LDZ,NMAX)
*     .. External Subroutines ..
      EXTERNAL         F02EAF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02EAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*        Compute Schur factorization of A
*
         IFAIL = 0
*
         CALL F02EAF('Vectors',N,A,LDA,WR,WI,Z,LDZ,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99999) (' (',WR(I),',',WI(I),')',I=1,N)
         WRITE (NOUT,*)
*
         CALL X04CAF('General',' ',N,N,A,LDA,'Schur form',IFAIL)
*
         WRITE (NOUT,*)
*
         CALL X04CAF('General',' ',N,N,Z,LDZ,'Schur vectors',IFAIL)
*
      END IF
      STOP
*
99999 FORMAT (1X,A,F8.4,A,F8.4,A)
      END
```

9.2 Program Data

```
F02EAF Example Program Data
  4                                    :Value of N
  0.35   0.45  -0.14  -0.17
  0.09   0.07  -0.54   0.35
 -0.44  -0.33  -0.03   0.17
  0.25  -0.32  -0.13   0.11   :End of matrix A
```

9.3 Program Results

```
 F02EAF 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 -0.0060 -0.1144  0.0336
 2  -0.0000 -0.0994  0.6483 -0.2026
 3   0.0000 -0.2478 -0.0994  0.3474
 4   0.0000  0.0000 -0.0000 -0.1007

 Schur vectors
          1       2       3       4
 1   0.6551 -0.3450  0.1037  0.6641
 2   0.5236  0.6141 -0.5807 -0.1068
 3  -0.5362  0.2935 -0.3073  0.7293
 4   0.0956  0.6463  0.7467  0.1249
```