F02GAF – NAG Fortran Library Routine Document

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

F02GAF computes all the eigenvalues, and optionally the Schur factorization, of a complex general matrix.

2 Specification

```
      SUBROUTINE F02GAF(JOB, N, A, LDA, W, Z, LDZ, RWORK, WORK, LWORK,
     1                  IFAIL)
      INTEGER           N, LDA, LDZ, LWORK, IFAIL
      real              RWORK(*)
      complex           A(LDA,*), W(*), 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 complex general matrix A:

$$A = ZTZ^H,$$

where T is an upper triangular matrix, and Z is a unitary 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 F08QUF (CTRSEN/ZTRSEN). Other reorderings may be achieved by calls to F08QTF (CTREXC/ZTREXC).

4 References

[1] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), 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: N ≥ 0.

3: A(LDA,∗) — ***complex*** 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 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 F02GAF is called.

Constraint: LDA ≥ max(1,N).

5: W(∗) — ***complex*** array *Output*

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

On exit: the computed eigenvalues. If JOB = 'S' or 'V', the eigenvalues occur in the same order as on the diagonal of T.

6: Z(LDZ,∗) — ***complex*** 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 unitary matrix Z of Schur vectors.

Z is not referenced if JOB = 'N' or 'S'.

7: LDZ — INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F02GAF is called.

Constraints:

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

8: RWORK(∗) — ***real*** array *Workspace*

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

9: WORK(LWORK) — ***complex*** array *Workspace*

10: LWORK — INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02GAF is called. On some high-performance computers, increasing the dimension of WORK will enable the routine to run faster; a value of 64 × N should allow near-optimal performance on almost all machines.

Constraint: LWORK ≥ max(1,2×N).

11: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

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 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(1,2×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

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

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 F08QYF (CTRSNA/ZTRSNA).

8 Further Comments

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

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

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix}.$$

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.

```
*     F02GAF 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 ..
      complex          A(LDA,NMAX), W(NMAX), WORK(LWORK), Z(LDZ,NMAX)
      real             RWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F02GAF, X04DBF
*     .. Intrinsic Functions ..
      INTRINSIC        imag, real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02GAF 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 F02GAF('Vectors',N,A,LDA,W,Z,LDZ,RWORK,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99999) (' (',real(W(I)),',',imag(W(I)),')',I=1,N)
         WRITE (NOUT,*)
*
         CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed','F7.4',
     +               'Schur form','Integer',RLABS,'Integer',CLABS,80,0,
     +               IFAIL)
*
         WRITE (NOUT,*)
*
         CALL X04DBF('General',' ',N,N,Z,LDZ,'Bracketed','F7.4',
     +               'Schur vectors','Integer',RLABS,'Integer',CLABS,80,
     +               0,IFAIL)
*
      END IF
      STOP
*
99999 FORMAT (3X,4(A,F7.4,A,F7.4,A,:))
      END
```

9.2 Program Data

```
F02GAF Example Program Data
  4                                                           :Value of N
 (-3.97,-5.04) (-4.11, 3.70) (-0.34, 1.01) ( 1.29,-0.86)
 ( 0.34,-1.50) ( 1.52,-0.43) ( 1.88,-5.38) ( 3.36, 0.65)
 ( 3.31,-3.85) ( 2.50, 3.45) ( 0.88,-1.08) ( 0.64,-1.48)
 (-1.10, 0.82) ( 1.81,-1.59) ( 3.25, 1.33) ( 1.57,-3.44)   :End of matrix A
```

9.3 Program Results

```
F02GAF 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) (-0.4701,-0.2119) ( 0.0438, 0.5124) (-0.9097,-0.0925)
2  ( 0.0000, 0.0000) (-5.0000, 2.0060) ( 0.7150,-0.1028) (-0.0580, 0.2575)
3  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 7.9982,-0.9964) (-0.2232,-1.0549)
4  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 3.0023,-3.9998)

Schur vectors
                   1                 2                 3                 4
1  ( 0.8457, 0.0000) (-0.3613, 0.1351) (-0.1755, 0.2297) ( 0.1099,-0.2007)
2  (-0.0177, 0.3036) (-0.3366, 0.4660) ( 0.7228, 0.0000) ( 0.0336, 0.2312)
3  ( 0.0875, 0.3115) ( 0.6311, 0.0000) ( 0.2871, 0.4999) ( 0.0944,-0.3947)
4  (-0.0561,-0.2906) (-0.1045,-0.3339) ( 0.2476, 0.0195) ( 0.8534, 0.0000)
```