NAG Fortran Library Routine Document

F02GBF

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

F02GBF computes all the eigenvalues, and optionally all the eigenvectors, of a complex general matrix.

2 Specification

```
SUBROUTINE F02GBF(JOB, N, A, LDA, W, V, LDV, RWORK, WORK, LWORK, IFAIL)
INTEGER            N, LDA, LDV, LWORK, IFAIL
real               RWORK(*)
complex            A(LDA,*), W(*), V(LDV,*), WORK(LWORK)
CHARACTER*1        JOB
```

3 Description

This routine computes all the eigenvalues, and optionally all the right eigenvectors, of a complex general matrix A:

$$Ax_i = \lambda_i x_i, \quad i = 1, 2, \ldots, n.$$

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 eigenvectors are to be computed as follows:

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

if JOB = 'V', then eigenvalues and eigenvectors are computed.

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

2: N – INTEGER *Input*

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

Constraint: N $\geq$ 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 = 'V', A contains the Schur form of the balanced input matrix A' (see Section 8); 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 F02GBF is called.

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

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

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

On exit: the computed eigenvalues.

6: V(LDV,*) – ***complex*** array *Output*

Note: the second dimension of the array V must be at least $\max(1, \text{N})$ if JOB = 'V' and at least 1 otherwise.

On exit: if JOB = 'V', V contains the eigenvectors, with the ith column holding the eigenvector associated with the eigenvalue λ_i (stored in W(i)).

V is not referenced if JOB = 'N'.

7: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F02GBF is called.

Constraints:

LDV ≥ 1, if JOB = 'N',
LDV $\geq \max(1, \text{N})$, if JOB = 'V'.

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

Note: the dimension of the array RWORK must be at least $\max(1, 2 \times \text{N})$.

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 F02GBF is called. On some high-performance computers, increasing the dimension of WORK will enable the routine to run faster; a value of $64 \times \text{N}$ should allow near-optimal performance on almost all machines.

Constraint: LWORK $\geq \max(1, 2 \times \text{N})$.

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' or 'V',
or N < 0,
or LDA < max(1, N),
or LDV < 1, or LDV < N and JOB = 'V',
or LWORK < max(1, 2 × N).

IFAIL = 2

The QR algorithm failed to compute all the eigenvalues.

7 Accuracy

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, ϵ is the ***machine precision***, and s_i is the reciprocal condition number of λ_i; A' is the balanced form of the original matrix A (see Section 8), and $\|A'\| \leq \|A\|$.

If x_i is the corresponding exact eigenvector, and $\tilde{x}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{x}_i, x_i)$ between them is bounded as follows:

$$\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\epsilon\|A'\|_2}{sep_i},$$

where sep_i is the reciprocal condition number of x_i.

The condition numbers s_i and sep_i may be computed by calling F08QYF (CTRSNA/ZTRSNA), using the Schur form of the balanced matrix A' which is returned in the array A when JOB = 'V'.

8 Further Comments

The routine calls routines from LAPACK in Chapter F08. It first balances the matrix, using a diagonal similarity transformation to reduce its norm; and then reduces the balanced matrix A' to upper Hessenberg form H, using a unitary similarity transformation: $A' = QHQ^H$. If only eigenvalues are required, the routine uses the Hessenberg QR algorithm to compute the eigenvalues. If the eigenvectors are required, the routine first forms the unitary matrix Q that was used in the reduction to Hessenberg form; it then uses the Hessenberg QR algorithm to compute the Schur factorization of A' as $A' = ZTZ^H$. It computes the right eigenvectors of T by backward substitution, pre-multiplies them by Z to form the eigenvectors of A'; and finally transforms the eigenvectors to those of the original matrix A.

Each eigenvector x is normalized so that $\|x\|_2 = 1$, 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 all the eigenvalues and eigenvectors 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.

```
*     F02GBF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDV, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LDV=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*     .. Local Arrays ..
      complex          A(LDA,NMAX), V(LDV,NMAX), W(NMAX), WORK(LWORK)
      real             RWORK(2*NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F02GBF, X04DBF
*     .. Intrinsic Functions ..
      INTRINSIC        imag, real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02GBF 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 eigenvalues and eigenvectors of A
*
         IFAIL = 0
*
         CALL F02GBF('Vectors',N,A,LDA,W,V,LDV,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,V,LDV,'Bracketed','F7.4',
     +               'Eigenvectors','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

```
F02GBF 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

```
F02GBF Example Program Results

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

Eigenvectors
                   1                 2                 3                 4
1  ( 0.8457, 0.0000) (-0.3865, 0.1732) (-0.1730, 0.2669) (-0.0356,-0.1782)
2  (-0.0177, 0.3036) (-0.3539, 0.4529) ( 0.6924, 0.0000) ( 0.1264, 0.2666)
3  ( 0.0875, 0.3115) ( 0.6124, 0.0000) ( 0.3324, 0.4960) ( 0.0129,-0.2966)
4  (-0.0561,-0.2906) (-0.0859,-0.3284) ( 0.2504,-0.0147) ( 0.8898, 0.0000)
```