NAG Library Routine Document

F08GQF (ZHPEVD)

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 parameters LRWORK and LIWORK changed at Mark 20 in the case where JOB = 'V' and N > 1: the minimum dimension of the array RWORK has been reduced whereas the minimum dimension of the array IWORK has been increased.

1 Purpose

F08GQF (ZHPEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian matrix held in packed storage. If the eigenvectors are requested, then it uses a divide-and-conquer algorithm to compute eigenvalues and eigenvectors. However, if only eigenvalues are required, then it uses the Pal–Walker–Kahan variant of the QL or QR algorithm.

2 Specification

```
 SUBROUTINE F08GQF(JOB, UPLO, N, AP, W, Z, LDZ, WORK, LWORK, RWORK,
1                  LRWORK, IWORK, LIWORK, INFO)
 INTEGER           N, LDZ, LWORK, LRWORK, IWORK(max(1,LIWORK)), LIWORK,
1                  INFO
 double precision  W(*), RWORK(max(1,LRWORK))
 complex*16        AP(*), Z(LDZ,*), WORK(max(1,LWORK))
 CHARACTER*1       JOB, UPLO
```

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

3 Description

F08GQF (ZHPEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian matrix A (held in packed storage). In other words, it can compute the spectral factorization of A as

$$A = Z\Lambda Z^{\mathrm{H}},$$

where Λ is a real diagonal matrix whose diagonal elements are the eigenvalues λ_i, and Z is the (complex) unitary matrix whose columns are the eigenvectors z_i. Thus

$$Az_i = \lambda_i z_i, \qquad i = 1, 2, \ldots, n.$$

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphiahttp://www.netlib.org/lapack/lug

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 computed.

JOB = 'N'
  Only eigenvalues are computed.

JOB = 'V'
Eigenvalues and eigenvectors are computed.

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

2: UPLO – CHARACTER*1 *Input*

On entry: indicates whether the upper or lower triangular part of A is stored.

UPLO = 'U'
The upper triangular part of A is stored.

UPLO = 'L'
The lower triangular part of A is stored.

Constraint: UPLO = 'U' or 'L'.

3: N – INTEGER *Input*

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

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

4: AP($*$) – ***complex*16*** array *Input/Output*

Note: the dimension of the array AP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the n by n Hermitian matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in AP$(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in AP$(i + (2n-j)(j-1)/2)$ for $i \geq j$.

On exit: AP is overwritten by the values generated during the reduction to tridiagonal form. The elements of the diagonal and the off-diagonal of the tridiagonal matrix overwrite the corresponding elements of A.

5: W($*$) – ***double precision*** array *Output*

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

On exit: the eigenvalues of the matrix A in ascending order.

6: Z(LDZ,$*$) – ***complex*16*** array *Output*

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

On exit: if JOB = 'V', Z is overwritten by the unitary matrix Z which contains the eigenvectors of A.

If JOB = 'N', Z is not referenced.

7: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08GQF (ZHPEVD) is called.

Constraints:

if JOB = 'V', $\mathrm{LDZ} \geq \max(1, \mathrm{N})$;
if JOB = 'N', $\mathrm{LDZ} \geq 1$.

8: WORK$(\max(1,\mathrm{LWORK}))$ – ***complex*16*** array *Workspace*

On exit: if INFO $= 0$, WORK(1) contains the required minimal size of LWORK.

9: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08GQF (ZHPEVD) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LWORK} \geq 1$ or $\mathrm{LWORK} = -1$;
if JOB = 'N' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq \mathrm{N}$ or $\mathrm{LWORK} = -1$;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq 2 \times \mathrm{N}$ or $\mathrm{LWORK} = -1$.

10: RWORK$(\max(1,\mathrm{LRWORK}))$ – ***double precision*** array *Workspace*

On exit: if INFO $= 0$, RWORK(1) contains the required minimal size of LRWORK.

11: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which F08GQF (ZHPEVD) is called.

If LRWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the RWORK array, returns this value as the first entry of the RWORK array, and no error message related to LRWORK is issued.

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LRWORK} \geq 1$ or $\mathrm{LRWORK} = -1$;
if JOB = 'N' and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq \mathrm{N}$ or $\mathrm{LRWORK} = -1$;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq 2 \times \mathrm{N}^2 + 5 \times \mathrm{N} + 1$ or $\mathrm{LRWORK} = -1$.

12: IWORK$(\max(1,\mathrm{LIWORK}))$ – INTEGER array *Workspace*

On exit: if INFO $= 0$, IWORK(1) contains the required minimal size of LIWORK.

13: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08GQF (ZHPEVD) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the IWORK array, returns this value as the first entry of the IWORK array, and no error message related to LIWORK is issued.

Constraints:

if JOB = 'N' or $\mathrm{N} \leq 1$, $\mathrm{LIWORK} \geq 1$ or $\mathrm{LIWORK} = -1$;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LIWORK} \geq 5 \times \mathrm{N} + 3$ or $\mathrm{LIWORK} = -1$.

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

if INFO $= i$ and JOB = 'N', the algorithm failed to converge; i elements of an intermediate tridiagonal form did not converge to zero; if INFO $= i$ and JOB = 'V', then the algorithm failed to compute an eigenvalue while working on the submatrix lying in rows and column $i/(\mathrm{N}+1)$ through $\mathrm{mod}(i, \mathrm{N}+1)$.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $(A+E)$, where

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

and ϵ is the ***machine precision***. See Section 4.7 of Anderson *et al.* (1999) for further details.

8 Further Comments

The real analogue of this routine is F08GCF (DSPEVD).

9 Example

This example computes all the eigenvalues and eigenvectors of the Hermitian matrix A, where

$$A = \begin{pmatrix} 1.0+0.0i & 2.0-1.0i & 3.0-1.0i & 4.0-1.0i \\ 2.0+1.0i & 2.0+0.0i & 3.0-2.0i & 4.0-2.0i \\ 3.0+1.0i & 3.0+2.0i & 3.0+0.0i & 4.0-3.0i \\ 4.0+1.0i & 4.0+2.0i & 4.0+3.0i & 4.0+0.0i \end{pmatrix}.$$

9.1 Program Text

```
*     F08GQF Example Program Text
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDZ
      PARAMETER        (NMAX=8,LDZ=NMAX)
      INTEGER          LWORK, LIWORK, LRWORK, MMAX
      PARAMETER        (LWORK=2*NMAX,LIWORK=5*NMAX+3,
     +                 LRWORK=2*NMAX*NMAX+5*NMAX+1,MMAX=NMAX*(NMAX+1)/2)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
      CHARACTER        JOB, UPLO
*     .. Local Arrays ..
      COMPLEX *16      AP(MMAX), WORK(LWORK), Z(LDZ,NMAX)
      DOUBLE PRECISION RWORK(LRWORK), W(NMAX)
      INTEGER          IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         X04DAF, ZHPEVD
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08GQF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
         READ (NIN,*) UPLO
*
```

```
*        Read A from data file
*
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((AP(I+J*(J-1)/2),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((AP(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
         END IF
*
         READ (NIN,*) JOB
*
*        Calculate all the eigenvalues and eigenvectors of A
*
         CALL ZHPEVD(JOB,UPLO,N,AP,W,Z,LDZ,WORK,LWORK,RWORK,LRWORK,
     +               IWORK,LIWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
*
*           Print eigenvalues and eigenvectors
*
            WRITE (NOUT,*) 'Eigenvalues'
            DO 20 I = 1, N
               WRITE (NOUT,99999) I, W(I)
   20       CONTINUE
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04DAF('General',' ',N,N,Z,LDZ,'Eigenvectors',IFAIL)
*
         END IF
      END IF
*
99999 FORMAT (3X,I5,5X,2F8.4)
      END
```

9.2 Program Data

```
F08GQF Example Program Data
  4                                                     :Value of N
  'L'                                                   :Value of UPLO
  (1.0, 0.0)
  (2.0, 1.0)  (2.0, 0.0)
  (3.0, 1.0)  (3.0, 2.0)  (3.0, 0.0)
  (4.0, 1.0)  (4.0, 2.0)  (4.0, 3.0)  (4.0, 0.0)     :End of matrix A
  'V'                                                   :Value of JOB
```

9.3 Program Results

```
 F08GQF Example Program Results

 Eigenvalues
       1      -4.2443
       2      -0.6886
       3       1.1412
       4      13.7916

 Eigenvectors
          1       2       3       4
 1   0.4836  0.6470 -0.4456 -0.3859
     0.0000  0.0000  0.0000  0.0000

 2   0.2912 -0.4984 -0.0230 -0.4441
    -0.3618 -0.1130 -0.5702  0.0156
```

```
 3  -0.3163  0.2949  0.5331 -0.5173
    -0.3696  0.3165  0.1317 -0.0844

 4  -0.4447 -0.2241 -0.3510 -0.5277
     0.3406 -0.2878  0.2261 -0.3168
```