# NAG Library Routine Document

# F08FQF (ZHEEVD)

**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 arguments 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

F08FQF (ZHEEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian matrix. 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 F08FQF (JOB, UPLO, N, A, LDA, W, WORK, LWORK, RWORK, LRWORK,    &
                   IWORK, LIWORK, INFO)
INTEGER                 N, LDA, LWORK, LRWORK, IWORK(max(1,LIWORK)),       &
                        LIWORK, INFO
REAL (KIND=nag_wp)      W(*), RWORK(max(1,LRWORK))
COMPLEX (KIND=nag_wp) A(LDA,*), WORK(max(1,LWORK))
CHARACTER(1)            JOB, UPLO
```

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

## 3 Description

F08FQF (ZHEEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian matrix $A$. In other words, it can compute the spectral factorization of $A$ as

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

where $\Lambda$ is a real diagonal matrix whose diagonal elements are the eigenvalues $\lambda_i$, and $Z$ is the (complex) unitary matrix whose columns are the eigenvectors $z_i$. Thus

$$A z_i = \lambda_i z_i, \quad i = 1, 2, \ldots, n.$$

## 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, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Arguments

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: A(LDA, $*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

*On entry*: the $n$ by $n$ Hermitian matrix $A$.

If UPLO = 'U', the upper triangular part of $A$ must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of $A$ must be stored and the elements of the array above the diagonal are not referenced.

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

5: LDA – INTEGER *Input*

*On entry*: the first dimension of the array A as declared in the (sub)program from which F08FQF (ZHEEVD) is called.

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

6: W($*$) – REAL (KIND=nag_wp) 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.

7: WORK($\max(1, \mathrm{LWORK})$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

8: LWORK – INTEGER *Input*

*On entry*: the dimension of the array WORK as declared in the (sub)program from which F08FQF (ZHEEVD) is called.

If $\mathrm{LWORK} = -1$, 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 $\mathrm{N} > 1$, $\mathrm{LWORK} \geq \mathrm{N} + 1$ or $\mathrm{LWORK} = -1$;

if JOB = 'V' and $N > 1$, $LWORK \geq N \times (N + 2)$ or $LWORK = -1$.

9: RWORK(max(1, LRWORK)) – REAL (KIND=nag_wp) array *Workspace*

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

10: LRWORK – INTEGER *Input*

*On entry*: the dimension of the array RWORK as declared in the (sub)program from which F08FQF (ZHEEVD) 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 $N \leq 1$, $LRWORK \geq 1$ or $LRWORK = -1$;
if JOB = 'N' and $N > 1$, $LRWORK \geq N$ or $LRWORK = -1$;
if JOB = 'V' and $N > 1$, $LRWORK \geq 2 \times N^2 + 5 \times N + 1$ or $LRWORK = -1$.

11: IWORK(max(1, LIWORK)) – INTEGER array *Workspace*

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

12: LIWORK – INTEGER *Input*

*On entry*: the dimension of the array IWORK as declared in the (sub)program from which F08FQF (ZHEEVD) 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 $N \leq 1$, $LIWORK \geq 1$ or $LIWORK = -1$;
if JOB = 'N' and $N > 1$, $LIWORK \geq 1$ or $LIWORK = -1$;
if JOB = 'V' and $N > 1$, $LIWORK \geq 5 \times N + 3$ or $LIWORK = -1$.

13: INFO – INTEGER *Output*

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

## 6 Error Indicators and Warnings

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 $i/(N + 1)$ through $i \bmod (N + 1)$.

## 7 Accuracy

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

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

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

## 8 Parallelism and Performance

F08FQF (ZHEEVD) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08FQF (ZHEEVD) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

The real analogue of this routine is F08FCF (DSYEVD).

## 10 Example

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

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

The example program for F08FQF (ZHEEVD) illustrates the computation of error bounds for the eigenvalues and eigenvectors.

### 10.1 Program Text

```
    Program f08fqfe

!     F08FQF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04daf, zheevd, zscal
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, k, lda, liwork,      &
                                          lrwork, lwork, n
      Character (1)                    :: job, uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), work(:)
      Real (Kind=nag_wp), Allocatable  :: rwork(:), w(:)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, cmplx, conjg, maxloc
!     .. Executable Statements ..
      Write (nout,*) 'F08FQF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
```

```
      lda = n
      liwork = 5*n + 3
      lrwork = 2*n*n + 5*n + 1
      lwork = n*(n+2)
      Allocate (a(lda,n),work(lwork),rwork(lrwork),w(n),iwork(liwork))
      Read (nin,*) uplo

!     Read A from data file

      If (uplo=='U') Then
        Read (nin,*)(a(i,i:n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)(a(i,1:i),i=1,n)
      End If

      Read (nin,*) job

!     Calculate all the eigenvalues and eigenvectors of A
!     The NAG name equivalent of zheevd is f08fqf
      Call zheevd(job,uplo,n,a,lda,w,work,lwork,rwork,lrwork,iwork,liwork,     &
        info)

      Write (nout,*)
      If (info>0) Then
        Write (nout,*) 'Failure to converge.'
      Else

!       Print eigenvalues and eigenvectors

        Write (nout,*) 'Eigenvalues'
        Do i = 1, n
          Write (nout,99999) i, w(i)
        End Do
        Write (nout,*)
        Flush (nout)

!       Normalize the eigenvectors so that the element of largest absolute
!       value is real.
        Do i = 1, n
          rwork(1:n) = abs(a(1:n,i))
          k = maxloc(rwork(1:n),1)
          Call zscal(n,conjg(a(k,i))/cmplx(abs(a(k,i)),kind=nag_wp),a(1,i),1)
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04daf('General',' ',n,n,a,lda,'Eigenvectors',ifail)

      End If

99999 Format (3X,I5,5X,2F8.4)
    End Program f08fqfe
```

## 10.2 Program Data

```
F08FQF 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
```

### 10.3 Program Results

```
F08FQF Example Program Results

Eigenvalues
      1      -4.2443
      2      -0.6886
      3       1.1412
      4      13.7916

Eigenvectors
         1       2       3       4
1  -0.3839  0.6470  0.0179  0.3309
   -0.2941  0.0000 -0.4453 -0.1986

2  -0.4512 -0.4984  0.5706  0.3728
    0.1102 -0.1130  0.0000 -0.2419

3   0.0263  0.2949 -0.1530  0.4870
    0.4857  0.3165  0.5273 -0.1938

4   0.5602 -0.2241 -0.2118  0.6155
    0.0000 -0.2878 -0.3598  0.0000
```