NAG Library Routine Document

F08FNF (ZHEEV)

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

F08FNF (ZHEEV) computes all the eigenvalues and, optionally, all the eigenvectors of a complex n by n Hermitian matrix A.

2 Specification

```
SUBROUTINE F08FNF (JOBZ, UPLO, N, A, LDA, W, WORK, LWORK, RWORK, INFO)
INTEGER                 N, LDA, LWORK, INFO
REAL (KIND=nag_wp)      W(N), RWORK(*)
COMPLEX (KIND=nag_wp)   A(LDA,*), WORK(max(1,LWORK))
CHARACTER(1)            JOBZ, UPLO
```

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

3 Description

The Hermitian matrix A is first reduced to real tridiagonal form, using unitary similarity transformations, and then the QR algorithm is applied to the tridiagonal matrix to compute the eigenvalues and (optionally) the eigenvectors.

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: JOBZ – CHARACTER(1) *Input*

On entry: indicates whether eigenvectors are computed.

JOBZ = 'N'
Only eigenvalues are computed.

JOBZ = 'V'
Eigenvalues and eigenvectors are computed.

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

2: UPLO – CHARACTER(1) *Input*

On entry: if UPLO = 'U', the upper triangular part of A is stored.

If 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: $N \geq 0$.

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

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

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 JOBZ = 'V', then A contains the orthonormal eigenvectors of the matrix A.

If JOBZ = 'N', then on exit the lower triangle (if UPLO = 'L') or the upper triangle (if UPLO = 'U') of A, including the diagonal, is overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08FNF (ZHEEV) is called.

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

6: W(N) – REAL (KIND=nag_wp) array *Output*

On exit: the eigenvalues in ascending order.

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

On exit: if $INFO = 0$, the real part of $WORK(1)$ contains the minimum value of LWORK required for optimal performance.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08FNF (ZHEEV) is called.

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

Suggested value: for optimal performance, $LWORK \geq (nb + 1) \times N$, where nb is the optimal ***block size*** for F08FSF (ZHETRD).

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

9: $RWORK(*)$ – REAL (KIND=nag_wp) array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 3 \times N - 2)$.

10: 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 $= i$, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

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 Parallelism and Performance

F08FNF (ZHEEV) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08FNF (ZHEEV) 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

Each eigenvector is normalized so that the element of largest absolute value is real.

The total number of floating-point operations is proportional to n^3.

The real analogue of this routine is F08FAF (DSYEV).

10 Example

This example finds all the eigenvalues and eigenvectors of the Hermitian matrix

$$A = \begin{pmatrix} 1 & 2-i & 3-i & 4-i \\ 2+i & 2 & 3-2i & 4-2i \\ 3+i & 3+2i & 3 & 4-3i \\ 4+i & 4+2i & 4+3i & 4 \end{pmatrix},$$

together with approximate error bounds for the computed eigenvalues and eigenvectors.

10.1 Program Text

```
    Program f08fnfe

!     F08FNF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: ddisna, nag_wp, x02ajf, x04daf, zheev, zscal
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: eerrbd, eps
      Integer                          :: i, ifail, info, k, lda, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), work(:)
      Complex (Kind=nag_wp)            :: dummy(1)
```

```
      Real (Kind=nag_wp), Allocatable  :: rcondz(:), rwork(:), w(:), zerrbd(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, cmplx, conjg, max, maxloc,      &
                                          nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08FNF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),rcondz(n),rwork(3*n-2),w(n),zerrbd(n))

!     Use routine workspace query to get optimal workspace.
!     The NAG name equivalent of zheev is f08fnf
      lwork = -1
      Call zheev('Vectors','Upper',n,a,lda,w,dummy,lwork,rwork,info)

!     Make sure that there is enough workspace for block size nb.
      lwork = max((nb+1)*n,nint(real(dummy(1))))
      Allocate (work(lwork))

!     Read the upper triangular part of the matrix A from data file

      Read (nin,*)(a(i,i:n),i=1,n)

!     Solve the Hermitian eigenvalue problem
!     The NAG name equivalent of zheev is f08fnf
      Call zheev('Vectors','Upper',n,a,lda,w,work,lwork,rwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) w(1:n)

        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)

!       Get the machine precision, EPS and compute the approximate
!       error bound for the computed eigenvalues.  Note that for
!       the 2-norm, max( abs(W(i)) ) = norm(A), and since the
!       eigenvalues are returned in descending order
!       max( abs(W(i)) ) = max( abs(W(1)), abs(W(n)))

        eps = x02ajf()
        eerrbd = eps*max(abs(w(1)),abs(w(n)))

!       Call DDISNA (F08FLF) to estimate reciprocal condition
!       numbers for the eigenvectors
        Call ddisna('Eigenvectors',n,n,w,rcondz,info)

!       Compute the error estimates for the eigenvectors

        Do i = 1, n
          zerrbd(i) = eerrbd/rcondz(i)
        End Do
```

```
!       Print the approximate error bounds for the eigenvalues
!       and vectors

        Write (nout,*)
        Write (nout,*) 'Error estimate for the eigenvalues'
        Write (nout,99998) eerrbd
        Write (nout,*)
        Write (nout,*) 'Error estimates for the eigenvectors'
        Write (nout,99998) zerrbd(1:n)
      Else
        Write (nout,99997) 'Failure in ZHEEV. INFO =', info
      End If

99999 Format (3X,(8F8.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4)
    End Program f08fnfe
```

10.2 Program Data

```
F08FNF Example Program Data

  4                                                 :Value of N

 (1.0, 0.0)  (2.0, -1.0)  (3.0, -1.0)  (4.0, -1.0)
             (2.0,  0.0)  (3.0, -2.0)  (4.0, -2.0)
                          (3.0,  0.0)  (4.0, -3.0)
                                       (4.0,  0.0) :End of matrix A
```

10.3 Program Results

```
 F08FNF Example Program Results

 Eigenvalues
    -4.2443 -0.6886  1.1412 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

 Error estimate for the eigenvalues
        1.5E-15

 Error estimates for the eigenvectors
        4.3E-16    8.4E-16    8.4E-16    1.2E-16
```