

# NAG Library Routine Document

## F08FPF (ZHEEVX)

**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

F08FPF (ZHEEVX) computes selected eigenvalues and, optionally, eigenvectors of a complex  $n$  by  $n$  Hermitian matrix  $A$ . Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

### 2 Specification

```
SUBROUTINE F08FPF (JOBZ, RANGE, UPLO, N, A, LDA, VL, VU, IL, IU, ABSTOL, &
                  M, W, Z, LDZ, WORK, LWORK, RWORK, IWORK, JFAIL, INFO)
INTEGER          N, LDA, IL, IU, M, LDZ, LWORK, IWORK(*), &
                  JFAIL(*), INFO
REAL (KIND=nag_wp) VL, VU, ABSTOL, W(*), RWORK(*)
COMPLEX (KIND=nag_wp) A(LDA,*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)     JOBZ, RANGE, UPLO
```

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

### 3 Description

The Hermitian matrix  $A$  is first reduced to real tridiagonal form, using unitary similarity transformations. The required eigenvalues and eigenvectors are then computed from the tridiagonal matrix; the method used depends upon whether all, or selected, eigenvalues and eigenvectors are required.

### 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>

Demmel J W and Kahan W (1990) Accurate singular values of bidiagonal matrices *SIAM J. Sci. Statist. Comput.* **11** 873–912

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: RANGE – CHARACTER(1) *Input*

*On entry:* if RANGE = 'A', all eigenvalues will be found.

If RANGE = 'V', all eigenvalues in the half-open interval  $(VL, VU]$  will be found.

If RANGE = 'I', the ILth to IUth eigenvalues will be found.

*Constraint:* RANGE = 'A', 'V' or 'I'.

- 3: 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'.
- 4: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 5: 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:* the lower triangle (if UPLO = 'L') or the upper triangle (if UPLO = 'U') of  $A$ , including the diagonal, is overwritten.
- 6: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F08FPF (ZHEEVX) is called.  
*Constraint:*  $LDA \geq \max(1, N)$ .
- 7: VL – REAL (KIND=nag\_wp) *Input*  
 8: VU – REAL (KIND=nag\_wp) *Input*  
*On entry:* if RANGE = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.  
 If RANGE = 'A' or 'I', VL and VU are not referenced.  
*Constraint:* if RANGE = 'V',  $VL < VU$ .
- 9: IL – INTEGER *Input*  
 10: IU – INTEGER *Input*  
*On entry:* if RANGE = 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be returned.  
 If RANGE = 'A' or 'V', IL and IU are not referenced.  
*Constraints:*  
 if RANGE = 'I' and  $N = 0$ ,  $IL = 1$  and  $IU = 0$ ;  
 if RANGE = 'I' and  $N > 0$ ,  $1 \leq IL \leq IU \leq N$ .
- 11: ABSTOL – REAL (KIND=nag\_wp) *Input*  
*On entry:* the absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval  $[a, b]$  of width less than or equal to

$$\text{ABSTOL} + \epsilon \max(|a|, |b|),$$

where  $\epsilon$  is the *machine precision*. If ABSTOL is less than or equal to zero, then  $\epsilon \|T\|_1$  will be used in its place, where  $T$  is the tridiagonal matrix obtained by reducing  $A$  to tridiagonal form. Eigenvalues will be computed most accurately when ABSTOL is set to twice the underflow threshold  $2 \times \text{X02AMF}(\ )$ , not zero. If this routine returns with  $\text{INFO} > 0$ , indicating that some eigenvectors did not converge, try setting ABSTOL to  $2 \times \text{X02AMF}(\ )$ . See Demmel and Kahan (1990).

- 12: M – INTEGER *Output*  
*On exit:* the total number of eigenvalues found.  $0 \leq M \leq N$ .  
 If RANGE = 'A',  $M = N$ .  
 If RANGE = 'I',  $M = \text{IU} - \text{IL} + 1$ .
- 13: W(\*) – REAL (KIND=nag\_wp) array *Output*  
**Note:** the dimension of the array W must be at least  $\max(1, N)$ .  
*On exit:* the first M elements contain the selected eigenvalues in ascending order.
- 14: Z(LDZ, \*) – COMPLEX (KIND=nag\_wp) array *Output*  
**Note:** the second dimension of the array Z must be at least  $\max(1, M)$  if JOBZ = 'V', and at least 1 otherwise.  
*On exit:* if JOBZ = 'V', then  
     if  $\text{INFO} = 0$ , the first M columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the  $i$ th column of Z holding the eigenvector associated with  $W(i)$ ;  
     if an eigenvector fails to converge ( $\text{INFO} > 0$ ), then that column of Z contains the latest approximation to the eigenvector, and the index of the eigenvector is returned in JFAIL.  
 If JOBZ = 'N', Z is not referenced.  
**Note:** you must ensure that at least  $\max(1, M)$  columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound of at least N must be used.
- 15: LDZ – INTEGER *Input*  
*On entry:* the first dimension of the array Z as declared in the (sub)program from which F08FPF (ZHEEVX) is called.  
*Constraints:*  
     if JOBZ = 'V',  $\text{LDZ} \geq \max(1, N)$ ;  
     otherwise  $\text{LDZ} \geq 1$ .
- 16: WORK(max(1, LWORK)) – COMPLEX (KIND=nag\_wp) array *Workspace*  
*On exit:* if  $\text{INFO} = 0$ , the real part of WORK(1) contains the minimum value of LWORK required for optimal performance.
- 17: LWORK – INTEGER *Input*  
*On entry:* the dimension of the array WORK as declared in the (sub)program from which F08FPF (ZHEEVX) 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 largest optimal **block size** for F08FSF (ZHETRD) and for F08FUF (ZUNMTR).

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

18: RWORK(\*) – REAL (KIND=nag\_wp) array Workspace

**Note:** the dimension of the array RWORK must be at least  $\max(1, 7 \times N)$ .

19: IWORK(\*) – INTEGER array Workspace

**Note:** the dimension of the array IWORK must be at least  $\max(1, 5 \times N)$ .

20: JFAIL(\*) – INTEGER array Output

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

*On exit:* if JOBZ = 'V', then

if INFO = 0, the first M elements of JFAIL are zero;

if INFO > 0, JFAIL contains the indices of the eigenvectors that failed to converge.

If JOBZ = 'N', JFAIL is not referenced.

21: 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

The algorithm failed to converge;  $\langle value \rangle$  eigenvectors did not converge. Their indices are stored in array JFAIL.

## 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

F08FPF (ZHEEVX) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08FPF (ZHEEVX) 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 total number of floating-point operations is proportional to  $n^3$ .

The real analogue of this routine is F08FBF (DSYEVX).

## 10 Example

This example finds the eigenvalues in the half-open interval  $(-2,2]$ , and the corresponding eigenvectors, of the Hermitian matrix

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

### 10.1 Program Text

```

Program f08fpfe

!      F08FPF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, x04daf, zheevx, zscal
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Real (Kind=nag_wp), Parameter      :: zero = 0.0E+0_nag_wp
Integer, Parameter                 :: nb = 64, nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)                 :: abstol, vl, vu
Integer                             :: i, ifail, il, info, iu, k, lda, ldz, &
                                     lwork, m, n
!      .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: a(:,,:), work(:), z(:,,:)
Complex (Kind=nag_wp)               :: dummy(1)
Real (Kind=nag_wp), Allocatable     :: rwork(:), w(:)
Integer, Allocatable                 :: iwork(:), jfail(:)
!      .. Intrinsic Procedures ..
Intrinsic                             :: abs, cmplx, conjg, max, maxloc,      &
                                     nint, real

!      .. Executable Statements ..
Write (nout,*) 'F08FPF Example Program Results'
Write (nout,*)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n
lda = n
ldz = n
m = n
Allocate (a(lda,n),z(ldz,m),rwork(7*n),w(n),iwork(5*n),jfail(n))

!      Read the lower and upper bounds of the interval to be searched.
Read (nin,*) vl, vu

!      Use routine workspace query to get optimal workspace.
lwork = -1
!      The NAG name equivalent of zheevx is f08fpf
Call zheevx('Vectors','Values in range','Upper',n,a,lda,vl,vu,il,iu,      &
           abstol,m,w,z,ldz,dummy,lwork,rwork,iwork,jfail,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.

```

```

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

!      Set the absolute error tolerance for eigenvalues.  With ABSTOL
!      set to zero, the default value is used instead

      abstol = zero

!      Solve the Hermitian eigenvalue problem

!      The NAG name equivalent of zheevx is f08fpf
      Call zheevx('Vectors','Values in range','Upper',n,a,lda,vl,vu,il,iu,      &
        abstol,m,w,z,ldz,work,lwork,rwork,iwork,jfail,info)

      If (info>=0) Then

!      Print solution

      Write (nout,99999) 'Number of eigenvalues found =', m
      Write (nout,*)
      Write (nout,*) 'Eigenvalues'
      Write (nout,99998) w(1:m)
      Flush (nout)

!      Normalize the eigenvectors so that the element of largest absolute
!      value is real.
      Do i = 1, m
        rwork(1:n) = abs(z(1:n,i))
        k = maxloc(rwork(1:n),1)
        Call zscal(n,conjg(z(k,i))/cplx(abs(z(k,i)),kind=nag_wp),z(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,m,z,ldz,'Selected eigenvectors',ifail)

      If (info>0) Then
        Write (nout,99999) 'INFO eigenvectors failed to converge, INFO =',      &
          info
        Write (nout,*) 'Indices of eigenvectors that did not converge'
        Write (nout,99997) jfail(1:m)
      End If
    Else
      Write (nout,99999) 'Failure in ZHEEVX. INFO =', info
    End If

99999 Format (1X,A,I5)
99998 Format (3X,(8F8.4))
99997 Format (3X,(8I8))
      End Program f08fpfe

```

## 10.2 Program Data

F08FPF Example Program Data

```

  4                                     :Value of N
-2.0      2.0                         :Values of VL and VU
(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

F08FPF Example Program Results

Number of eigenvalues found = 2

Eigenvalues  
-0.6886 1.1412

Selected eigenvectors

	1	2
1	0.6470	0.0179
	0.0000	-0.4453
2	-0.4984	0.5706
	-0.1130	0.0000
3	0.2949	-0.1530
	0.3165	0.5273
4	-0.2241	-0.2118
	-0.2878	-0.3598

---