NAG Library Routine Document

F08NNF (ZGEEV)

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

F08NNF (ZGEEV) computes the eigenvalues and, optionally, the left and/or right eigenvectors for an n by n complex nonsymmetric matrix A.

2 Specification

```
SUBROUTINE F08NNF (JOBVL, JOBVR, N, A, LDA, W, VL, LDVL, VR, LDVR, WORK,    &
                   LWORK, RWORK, INFO)
INTEGER                 N, LDA, LDVL, LDVR, LWORK, INFO
REAL (KIND=nag_wp)      RWORK(*)
COMPLEX (KIND=nag_wp) A(LDA,*), W(*), VL(LDVL,*), VR(LDVR,*),               &
                        WORK(max(1,LWORK))
CHARACTER(1)            JOBVL, JOBVR
```

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

3 Description

The right eigenvector v_j of A satisfies

$$Av_j = \lambda_j v_j$$

where λ_j is the jth eigenvalue of A. The left eigenvector u_j of A satisfies

$$u_j^{\mathrm{H}} A = \lambda_j u_j^{\mathrm{H}}$$

where u_j^{H} denotes the conjugate transpose of u_j.

The matrix A is first reduced to upper Hessenberg form by means of unitary similarity transformations, and the QR algorithm is then used to further reduce the matrix to upper triangular Schur form, T, from which the eigenvalues are computed. Optionally, the eigenvectors of T are also computed and backtransformed to those of A.

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 Parameters

1: JOBVL – CHARACTER(1) *Input*

On entry: if JOBVL = 'N', the left eigenvectors of A are not computed.

If JOBVL = 'V', the left eigenvectors of A are computed.

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

2: JOBVR – CHARACTER(1) *Input*

On entry: if JOBVR = 'N', the right eigenvectors of A are not computed.

If JOBVR = 'V', the right eigenvectors of A are computed.

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

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 matrix A.

On exit: A has been overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NNF (ZGEEV) is called.

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

6: W($*$) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: contains the computed eigenvalues.

7: VL(LDVL, $*$) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: if JOBVL = 'V', the left eigenvectors u_j are stored one after another in the columns of VL, in the same order as their corresponding eigenvalues; that is $u_j = \mathrm{VL}(:, j)$, the jth column of VL.

If JOBVL = 'N', VL is not referenced.

8: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08NNF (ZGEEV) is called.

Constraints:

if JOBVL = 'V', $\mathrm{LDVL} \geq \max(1, \mathrm{N})$;
otherwise $\mathrm{LDVL} \geq 1$.

9: VR(LDVR, $*$) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: if JOBVR = 'V', the right eigenvectors v_j are stored one after another in the columns of VR, in the same order as their corresponding eigenvalues; that is $v_j = \mathrm{VR}(:, j)$, the jth column of VR.

If JOBVR = 'N', VR is not referenced.

10: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08NNF (ZGEEV) is called.

Constraints:

if JOBVR = 'V', LDVR $\geq \max(1, \mathrm{N})$;
otherwise LDVR ≥ 1.

11: WORK$(\max(1, \mathrm{LWORK}))$ – 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.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08NNF (ZGEEV) 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 should be generally larger than the minimum, say $\mathrm{N} + nb \times \mathrm{N}$, where nb is the optimal ***block size*** for F08NSF (ZGEHRD).

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

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

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

14: 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 QR algorithm failed to compute all the eigenvalues, and no eigenvectors have been computed; elements $i + 1 : \mathrm{N}$ of W contain eigenvalues which have converged.

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 4.8 of Anderson *et al.* (1999) for further details.

8 Parallelism and Performance

F08NNF (ZGEEV) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08NNF (ZGEEV) 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 to have Euclidean norm equal to unity and the element of largest absolute value real and positive.

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

The real analogue of this routine is F08NAF (DGEEV).

10 Example

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

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix}.$$

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

10.1 Program Text

```
    Program f08nnfe

!     F08NNF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04daf, zgeev
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, ldvr, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), vr(:,:), w(:), work(:)
      Complex (Kind=nag_wp)            :: dummy(1,1)
      Real (Kind=nag_wp), Allocatable  :: rwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08NNF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldvr = n
      Allocate (a(lda,n),vr(ldvr,n),w(n),rwork(2*n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of zgeev is f08nnf
      Call zgeev('No left vectors','Vectors (right)',n,a,lda,w,dummy,1,vr, &
        ldvr,dummy,lwork,rwork,info)
```

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

!     Read the matrix A from data file

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

!     Compute the eigenvalues and right eigenvectors of A

!     The NAG name equivalent of zgeev is f08nnf
      Call zgeev('No left vectors','Vectors (right)',n,a,lda,w,dummy,1,vr, &
        ldvr,work,lwork,rwork,info)

      If (info==0) Then

!       Print solution

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

        Write (nout,*)
        Flush (nout)

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

      Else
        Write (nout,*)
        Write (nout,99998) 'Failure in ZGEEV.  INFO = ', info
      End If

99999 Format ((3X,4(' (',F7.4,',',F7.4,')':)))
99998 Format (1X,A,I4)
    End Program f08nnfe
```

10.2 Program Data

```
F08NNF 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
```

10.3 Program Results

```
 F08NNF 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 -0.3865 -0.1730 -0.0356
     0.0000  0.1732  0.2669 -0.1782

 2  -0.0177 -0.3539  0.6924  0.1264
     0.3036  0.4529  0.0000  0.2666
```

```
 3   0.0875  0.6124  0.3324  0.0129
     0.3115  0.0000  0.4960 -0.2966

 4  -0.0561 -0.0859  0.2504  0.8898
    -0.2906 -0.3284 -0.0147  0.0000
```