NAG Library Routine Document

F08NPF (ZGEEVX)

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

F08NPF (ZGEEVX) computes the eigenvalues and, optionally, the left and/or right eigenvectors for an n by n complex nonsymmetric matrix A.

Optionally, it also computes a balancing transformation to improve the conditioning of the eigenvalues and eigenvectors, reciprocal condition numbers for the eigenvalues, and reciprocal condition numbers for the right eigenvectors.

2 Specification

```
SUBROUTINE F08NPF (BALANC, JOBVL, JOBVR, SENSE, N, A, LDA, W, VL, LDVL,     &
                   VR, LDVR, ILO, IHI, SCALE, ABNRM, RCONDE, RCONDV,         &
                   WORK, LWORK, RWORK, INFO)
INTEGER                 N, LDA, LDVL, LDVR, ILO, IHI, LWORK, INFO
REAL (KIND=nag_wp)      SCALE(*), ABNRM, RCONDE(*), RCONDV(*), RWORK(*)
COMPLEX (KIND=nag_wp) A(LDA,*), W(*), VL(LDVL,*), VR(LDVR,*),               &
                        WORK(max(1,LWORK))
CHARACTER(1)            BALANC, JOBVL, JOBVR, SENSE
```

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

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.

Balancing a matrix means permuting the rows and columns to make it more nearly upper triangular, and applying a diagonal similarity transformation DAD^{-1}, where D is a diagonal matrix, with the aim of making its rows and columns closer in norm and the condition numbers of its eigenvalues and eigenvectors smaller. The computed reciprocal condition numbers correspond to the balanced matrix. Permuting rows and columns will not change the condition numbers (in exact arithmetic) but diagonal scaling will. For further explanation of balancing, see Section 4.8.1.2 of Anderson *et al.* (1999).

Following the optional balancing, 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: BALANC – CHARACTER(1) *Input*

On entry: indicates how the input matrix should be diagonally scaled and/or permuted to improve the conditioning of its eigenvalues.

BALANC = 'N'
: Do not diagonally scale or permute.

BALANC = 'P'
: Perform permutations to make the matrix more nearly upper triangular. Do not diagonally scale.

BALANC = 'S'
: Diagonally scale the matrix, i.e., replace A by DAD^{-1}, where D is a diagonal matrix chosen to make the rows and columns of A more equal in norm. Do not permute.

BALANC = 'B'
: Both diagonally scale and permute A.

Computed reciprocal condition numbers will be for the matrix after balancing and/or permuting. Permuting does not change condition numbers (in exact arithmetic), but balancing does.

Constraint: BALANC = 'N', 'P', 'S' or 'B'.

2: 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.

If SENSE = 'E' or 'B', JOBVL must be set to JOBVL = 'V'.

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

3: 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.

If SENSE = 'E' or 'B', JOBVR must be set to JOBVR = 'V'.

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

4: SENSE – CHARACTER(1) *Input*

On entry: determines which reciprocal condition numbers are computed.

SENSE = 'N'
: None are computed.

SENSE = 'E'
: Computed for eigenvalues only.

SENSE = 'V'
: Computed for right eigenvectors only.

SENSE = 'B'
: Computed for eigenvalues and right eigenvectors.

If SENSE = 'E' or 'B', both left and right eigenvectors must also be computed (JOBVL = 'V' and JOBVR = 'V').

Constraint: SENSE = 'N', 'E', 'V' or 'B'.

5: N – INTEGER *Input*

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

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

6: $\mathrm{A}(\mathrm{LDA},*)$ – 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. If JOBVL = 'V' or JOBVR = 'V', A contains the Schur form of the balanced version of the matrix A.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NPF (ZGEEVX) is called.

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

8: $\mathrm{W}(*)$ – 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.

9: $\mathrm{VL}(\mathrm{LDVL},*)$ – 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.

10: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08NPF (ZGEEVX) is called.

Constraints:

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

11: $\mathrm{VR}(\mathrm{LDVR},*)$ – 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.

12: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08NPF (ZGEEVX) is called.

Constraints:

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

13: ILO – INTEGER *Output*
14: IHI – INTEGER *Output*

On exit: ILO and IHI are integer values determined when A was balanced. The balanced A has $a_{ij} = 0$ if $i > j$ and $j = 1, 2, \ldots, \mathrm{ILO} - 1$ or $i = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

15: SCALE($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: details of the permutations and scaling factors applied when balancing A.

If p_j is the index of the row and column interchanged with row and column j, and d_j is the scaling factor applied to row and column j, then

$\mathrm{SCALE}(j) = p_j$, for $j = 1, 2, \ldots, \mathrm{ILO} - 1$;

$\mathrm{SCALE}(j) = d_j$, for $j = \mathrm{ILO}, \ldots, \mathrm{IHI}$;

$\mathrm{SCALE}(j) = p_j$, for $j = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

The order in which the interchanges are made is N to $\mathrm{IHI} + 1$, then 1 to $\mathrm{ILO} - 1$.

16: ABNRM – REAL (KIND=nag_wp) *Output*

On exit: the 1-norm of the balanced matrix (the maximum of the sum of absolute values of elements of any column).

17: RCONDE($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: RCONDE(j) is the reciprocal condition number of the jth eigenvalue.

18: RCONDV($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: RCONDV(j) is the reciprocal condition number of the jth right eigenvector.

19: 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.

20: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08NPF (ZGEEVX) 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 must generally be larger than the minimum, increase LWORK by, say, $\mathrm{N} \times nb$, where nb is the optimal ***block size*** for F08NEF (DGEHRD).

Constraints:

if SENSE = 'N' or 'E', LWORK $\geq \max(1, 2 \times \mathrm{N})$;
if SENSE = 'V' or 'B', LWORK $\geq \max(1, \mathrm{N} \times \mathrm{N} + 2 \times \mathrm{N})$.

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

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

22: 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 or condition numbers have been computed; elements $1 : \mathrm{ILO} - 1$ and $i + 1 : \mathrm{N}$ 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

F08NPF (ZGEEVX) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08NPF (ZGEEVX) 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 F08NBF (DGEEVX).

10 Example

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

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

together with estimates of the condition number and forward error bounds for each eigenvalue and eigenvector. The option to balance the matrix is used. In order to compute the condition numbers of the eigenvalues, the left eigenvectors also have to be computed, but they are not printed out in this example.

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 f08npfe

!     F08NPF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ajf, zgeevx
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: abnrm, eps, tol
      Integer                          :: i, ihi, ilo, info, j, lda, ldvl,     &
                                          ldvr, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), vl(:,:), vr(:,:), w(:),    &
                                            work(:)
      Complex (Kind=nag_wp)            :: dummy(1)
      Real (Kind=nag_wp), Allocatable  :: rconde(:), rcondv(:), rwork(:),      &
                                          scale(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08NPF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldvl = n
      ldvr = n
      Allocate (a(lda,n),vl(ldvl,n),vr(ldvr,n),w(n),rconde(n),rcondv(n), &
        rwork(2*n),scale(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of zgeevx is f08npf
      Call zgeevx('Balance','Vectors (left)','Vectors (right)', &
        'Both reciprocal condition numbers',n,a,lda,w,vl,ldvl,vr,ldvr,ilo,ihi, &
        scale,abnrm,rconde,rcondv,dummy,lwork,rwork,info)

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

!     Read the matrix A from data file

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

!     Solve the eigenvalue problem

!     The NAG name equivalent of zgeevx is f08npf
      Call zgeevx('Balance','Vectors (left)','Vectors (right)', &
        'Both reciprocal condition numbers',n,a,lda,w,vl,ldvl,vr,ldvr,ilo,ihi, &
        scale,abnrm,rconde,rcondv,work,lwork,rwork,info)

      If (info==0) Then

!       Compute the machine precision

        eps = x02ajf()
```

```
        tol = eps*abnrm

!       Print the eigenvalues and vectors, and associated condition
!       number and bounds

        Write (nout,*)
        Write (nout,*) 'Eigenvalues'
        Write (nout,*)
        Write (nout,*) '         Eigenvalue           rcond    error'

        Do j = 1, n

!         Print information on jth eigenvalue

          If (rconde(j)>0.0_nag_wp) Then
            If (tol/rconde(j)<10.0_nag_wp*eps) Then
              Write (nout,99999) j, w(j), rconde(j), '-'
            Else
              Write (nout,99998) j, w(j), rconde(j), tol/rconde(j)
            End If
          Else
            Write (nout,99999) j, w(j), rconde(j), 'Inf'
          End If

        End Do

        Write (nout,*)
        Write (nout,*) 'Eigenvectors'
        Write (nout,*)
        Write (nout,*) '         Eigenvector          rcond    error'

        Do j = 1, n

!         Print information on jth eigenvector

          Write (nout,*)

!         Make first real part component be positive
          If (real(vr(1,j))<0.0_nag_wp) Then
            vr(1:n,j) = -vr(1:n,j)
          End If
          If (rcondv(j)>0.0_nag_wp) Then
            If (tol/rcondv(j)<10.0_nag_wp*eps) Then
              Write (nout,99999) j, vr(1,j), rcondv(j), '-'
            Else
              Write (nout,99998) j, vr(1,j), rcondv(j), tol/rcondv(j)
            End If
          Else
            Write (nout,99999) j, vr(1,j), rcondv(j), 'Inf'
          End If

          Write (nout,99997) vr(2:n,j)

        End Do
        Write (nout,*)
        Write (nout,*) 'Errors below 10*machine precision are not displayed'
      Else
        Write (nout,*)
        Write (nout,99996) 'Failure in ZGEEVX. INFO =', info
      End If

99999 Format (1X,I2,1X,'(',1P,E11.4,',',E11.4,')',1X,0P,F7.4,4X,A)
99998 Format (1X,I2,1X,'(',1P,E11.4,',',E11.4,')',1X,0P,F7.4,1X,1P,E8.1)
99997 Format (1X,3X,'(',1P,E11.4,',',E11.4,')')
99996 Format (1X,A,I4)

    End Program f08npfe
```

10.2 Program Data

```
F08NPF 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

```
 F08NPF Example Program Results

 Eigenvalues

          Eigenvalue           rcond    error
  1 (-6.0004E+00,-6.9998E+00)  0.9932  1.6E-15
  2 (-5.0000E+00, 2.0060E+00)  0.9964  1.6E-15
  3 ( 7.9982E+00,-9.9637E-01)  0.9814  1.6E-15
  4 ( 3.0023E+00,-3.9998E+00)  0.9779  1.6E-15

 Eigenvectors

          Eigenvector          rcond    error

  1 ( 8.4572E-01, 0.0000E+00)  8.4011    -
    (-1.7723E-02, 3.0361E-01)
    ( 8.7521E-02, 3.1145E-01)
    (-5.6147E-02,-2.9060E-01)

  2 ( 3.8655E-01,-1.7323E-01)  8.0214    -
    ( 3.5393E-01,-4.5288E-01)
    (-6.1237E-01,-0.0000E+00)
    ( 8.5928E-02, 3.2836E-01)

  3 ( 1.7297E-01,-2.6690E-01)  5.8292    -
    (-6.9242E-01,-0.0000E+00)
    (-3.3240E-01,-4.9598E-01)
    (-2.5039E-01, 1.4655E-02)

  4 ( 3.5614E-02, 1.7822E-01)  5.8292    -
    (-1.2637E-01,-2.6663E-01)
    (-1.2933E-02, 2.9657E-01)
    (-8.8982E-01,-0.0000E+00)

 Errors below 10*machine precision are not displayed
```