NAG Fortran 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, F08NPF (ZGEEVX) 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,
     1                   VR, LDVR, ILO, IHI, SCALE, ABNRM, RCONDE, RCONDV,
     2                   WORK, LWORK, RWORK, INFO)
      INTEGER            N, LDA, LDVL, LDVR, ILO, IHI, LWORK, INFO
      double precision   SCALE(*), ABNRM, RCONDE(*), RCONDV(*), RWORK(*)
      complex*16         A(LDA,*), W(*), VL(LDVL,*), VR(LDVR,*), WORK(*)
      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

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

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 B 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 URL: 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:

if BALANC = 'N', do not diagonally scale or permute;
if BALANC = 'P', perform permutations to make the matrix more nearly upper triangular. Do not diagonally scale;
if 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;
if 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.

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'.

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'.

4: SENSE – CHARACTER*1 *Input*

On entry: determines which reciprocal condition numbers are computed:

if SENSE = 'N', none are computed;
if SENSE = 'E', computed for eigenvalues only;
if SENSE = 'V', computed for right eigenvectors only;
if 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').

5: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

6: A(LDA,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the n by n matrix A.

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

8: W(∗) – ***complex*16*** array *Output*

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

On exit: contains the computed eigenvalues.

9: VL(LDVL,∗) – ***complex*16*** array *Output*

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

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.

If JOBVL = 'N', VL is not referenced. $u_j = \mathrm{VL}(:, j)$, the jth column of VL.

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

11: VR(LDVR,∗) – ***complex*16*** array *Output*

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

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.

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

$v_j = \mathrm{VR}(:, j)$, the jth column of VR.

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', $\mathrm{LDVR} \geq \max(1, \mathrm{N})$;
$\mathrm{LDVR} \geq 1$ otherwise.

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 $j = 1, \ldots, \mathrm{ILO} - 1$ or $i = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

15: SCALE(∗) – ***double precision*** 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 $j = 1, \ldots, \mathrm{ILO} - 1$;

$\mathrm{SCALE}(j) = d_j$, 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 – ***double precision*** *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($*$) – ***double precision*** array *Output*

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

On exit: $\mathrm{RCONDE}(j)$ is the reciprocal condition number of the jth eigenvalue.

18: RCONDV($*$) – ***double precision*** array *Output*

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

On exit: $\mathrm{RCONDV}(j)$ is the reciprocal condition number of the jth right eigenvector.

19: WORK($*$) – ***complex*16*** array *Workspace*

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

On exit: if INFO $= 0$, $\mathrm{WORK}(1)$ returns the optimal LWORK.

20: LWORK – INTEGER *Input*

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

For good 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).

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.

Constraints:

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

21: RWORK($*$) – ***double precision*** 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

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith argument had an illegal value.

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

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

and ϵ is the ***machine precision***. See Section 4.8 of Anderson *et al.* (1999) for further details.

8 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).

9 Example

To find 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.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08NPF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX
      PARAMETER        (NB=64,NMAX=10)
      INTEGER          LDA, LDVL, LDVR, LWORK
      PARAMETER        (LDA=NMAX,LDVL=NMAX,LDVR=NMAX,LWORK=(NB+1)*NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION ABNRM, EPS, ERBND, RCND, TOL
      INTEGER          I, IHI, ILO, INFO, J, LWKOPT, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), VL(LDVL,NMAX), VR(LDVR,NMAX),
     +                 W(NMAX), WORK(LWORK)
      DOUBLE PRECISION RCONDE(NMAX), RCONDV(NMAX), RWORK(2*NMAX),
     +                 SCALE(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF
      EXTERNAL         X02AJF
*     .. External Subroutines ..
      EXTERNAL         ZGEEVX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08NPF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
```

```
*
*        Read the matrix A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*        Solve the eigenvalue problem
*
         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.EQ.0) THEN
*
*           Compute the machine precision
*
            EPS = X02AJF()
            TOL = EPS*ABNRM
*
*           Print the eigenvalues and vectors, and associated condition
*           number and bounds
*
            DO 20 J = 1, N
*
*              Print information on jth eigenvalue
*
               WRITE (NOUT,*)
               WRITE (NOUT,99999) 'Eigenvalue(', J, ') = ', W(J)
               RCND = RCONDE(J)
               WRITE (NOUT,*)
               WRITE (NOUT,99998) 'Reciprocal condition number = ', RCND
               IF (RCND.GT.0.0D0) THEN
                  ERBND = TOL/RCND
                  WRITE (NOUT,99998) 'Error bound                 = ',
     +              ERBND
               ELSE
                  WRITE (NOUT,*) 'Error bound is infinite'
               END IF
*
*              Print information on jth eigenvector
*
               WRITE (NOUT,*)
               WRITE (NOUT,99997) 'Eigenvector(', J, ')',
     +           (VR(I,J),I=1,N)
               RCND = RCONDV(J)
               WRITE (NOUT,*)
               WRITE (NOUT,99998) 'Reciprocal condition number = ', RCND
               IF (RCND.GT.0.0D0) THEN
                  ERBND = TOL/RCND
                  WRITE (NOUT,99998) 'Error bound                 = ',
     +              ERBND
               ELSE
                  WRITE (NOUT,*) 'Error bound is infinite'
               END IF
   20       CONTINUE
         ELSE
            WRITE (NOUT,*)
            WRITE (NOUT,99996) 'Failure in ZGEEVX. INFO =', INFO
         END IF
*
*        Print workspace information
*
         LWKOPT = WORK(1)
         IF (LWORK.LT.LWKOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Optimum workspace required = ', LWKOPT,
     +        'Workspace provided         = ', LWORK
         END IF
      ELSE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'NMAX too small'
```

```
      END IF
      STOP
*
99999 FORMAT (1X,A,I2,A,'(',1P,E11.4,',',1P,E11.4,')')
99998 FORMAT (1X,A,1P,E8.1)
99997 FORMAT (1X,A,I2,A,/3(1X,'(',1P,E11.4,',',1P,E11.4,')',:))
99996 FORMAT (1X,A,I4)
99995 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 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
```

9.3 Program Results

```
 F08NPF Example Program Results

 Eigenvalue( 1) = (-6.0004E+00,-6.9998E+00)

 Reciprocal condition number =  9.9E-01
 Error bound                 =  1.6E-15

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

 Reciprocal condition number =  8.4E+00
 Error bound                 =  1.9E-16

 Eigenvalue( 2) = (-5.0000E+00, 2.0060E+00)

 Reciprocal condition number =  1.0E+00
 Error bound                 =  1.6E-15

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

 Reciprocal condition number =  8.0E+00
 Error bound                 =  2.0E-16

 Eigenvalue( 3) = ( 7.9982E+00,-9.9637E-01)

 Reciprocal condition number =  9.8E-01
 Error bound                 =  1.6E-15

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

 Reciprocal condition number =  5.8E+00
 Error bound                 =  2.7E-16

 Eigenvalue( 4) = ( 3.0023E+00,-3.9998E+00)

 Reciprocal condition number =  9.8E-01
 Error bound                 =  1.6E-15

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

 Reciprocal condition number =  5.8E+00
```

```
Error bound                  =  2.7E-16
```