NAG Library Routine Document

F08NVF (ZGEBAL)

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

F08NVF (ZGEBAL) balances a complex general matrix in order to improve the accuracy of computed eigenvalues and/or eigenvectors.

2 Specification

```
SUBROUTINE F08NVF (JOB, N, A, LDA, ILO, IHI, SCALE, INFO)

INTEGER                 N, LDA, ILO, IHI, INFO
REAL (KIND=nag_wp)      SCALE(N)
COMPLEX (KIND=nag_wp) A(LDA,*)
CHARACTER(1)            JOB
```

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

3 Description

F08NVF (ZGEBAL) balances a complex general matrix A. The term 'balancing' covers two steps, each of which involves a similarity transformation of A. The routine can perform either or both of these steps.

1. The routine first attempts to permute A to block upper triangular form by a similarity transformation:

$$PAP^{\mathrm{T}} = A' = \begin{pmatrix} A'_{11} & A'_{12} & A'_{13} \\ 0 & A'_{22} & A'_{23} \\ 0 & 0 & A'_{33} \end{pmatrix}$$

   where P is a permutation matrix, and A'_{11} and A'_{33} are upper triangular. Then the diagonal elements of A'_{11} and A'_{33} are eigenvalues of A. The rest of the eigenvalues of A are the eigenvalues of the central diagonal block A'_{22}, in rows and columns i_{lo} to i_{hi}. Subsequent operations to compute the eigenvalues of A (or its Schur factorization) need only be applied to these rows and columns; this can save a significant amount of work if $i_{\mathrm{lo}} > 1$ and $i_{\mathrm{hi}} < n$. If no suitable permutation exists (as is often the case), the routine sets $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$, and A'_{22} is the whole of A.

2. The routine applies a diagonal similarity transformation to A', to make the rows and columns of A'_{22} as close in norm as possible:

$$A'' = DA'D^{-1} = \begin{pmatrix} I & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} A'_{11} & A'_{12} & A'_{13} \\ 0 & A'_{22} & A'_{23} \\ 0 & 0 & A'_{33} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & D_{22}^{-1} & 0 \\ 0 & 0 & I \end{pmatrix}.$$

   This scaling can reduce the norm of the matrix (i.e., $\left\|A''_{22}\right\| < \left\|A'_{22}\right\|$) and hence reduce the effect of rounding errors on the accuracy of computed eigenvalues and eigenvectors.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: JOB – CHARACTER(1) *Input*

On entry: indicates whether A is to be permuted and/or scaled (or neither).

JOB = 'N'
A is neither permuted nor scaled (but values are assigned to ILO, IHI and SCALE).

JOB = 'P'
A is permuted but not scaled.

JOB = 'S'
A is scaled but not permuted.

JOB = 'B'
A is both permuted and scaled.

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

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: 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 matrix A.

On exit: A is overwritten by the balanced matrix. If JOB = 'N', A is not referenced.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NVF (ZGEBAL) is called.

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

5: ILO – INTEGER *Output*
6: IHI – INTEGER *Output*

On exit: the values i_{lo} and i_{hi} such that on exit A(i,j) is zero if $i > j$ and $1 \leq j < i_{\mathrm{lo}}$ or $i_{\mathrm{hi}} < i \leq n$.

If JOB = 'N' or 'S', $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$.

7: SCALE(N) – REAL (KIND=nag_wp) array *Output*

On exit: details of the permutations and scaling factors applied to A. More precisely, if p_j is the index of the row and column interchanged with row and column j and d_j is the scaling factor used to balance row and column j then

$$\mathrm{SCALE}(j) = \begin{cases} p_j, & j = 1,2,\ldots,i_{\mathrm{lo}}-1 \\ d_j, & j = i_{\mathrm{lo}}, i_{\mathrm{lo}}+1,\ldots,i_{\mathrm{hi}} \\ p_j, & j = i_{\mathrm{hi}}+1, i_{\mathrm{hi}}+2,\ldots,n. \end{cases} \quad \text{and}$$

The order in which the interchanges are made is n to $i_{\mathrm{hi}}+1$ then 1 to $i_{\mathrm{lo}}-1$.

8: 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 $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

The errors are negligible, compared with those in subsequent computations.

8 Further Comments

If the matrix A is balanced by F08NVF (ZGEBAL), then any eigenvectors computed subsequently are eigenvectors of the matrix A'' (see Section 3) and hence F08NWF (ZGEBAK) **must** then be called to transform them back to eigenvectors of A.

If the Schur vectors of A are required, then this routine must **not** be called with JOB = 'S' or 'B', because then the balancing transformation is not unitary. If this routine is called with JOB = 'P', then any Schur vectors computed subsequently are Schur vectors of the matrix A'', and F08NWF (ZGEBAK) **must** be called (with SIDE = 'R') to transform them back to Schur vectors of A.

The total number of real floating point operations is approximately proportional to n^2.

The real analogue of this routine is F08NHF (DGEBAL).

9 Example

This example computes all the eigenvalues and right eigenvectors of the matrix A, where

$$A = \begin{pmatrix} 1.50 - 2.75i & 0.00 + 0.00i & 0.00 + 0.00i & 0.00 + 0.00i \\ -8.06 - 1.24i & -2.50 - 0.50i & 0.00 + 0.00i & -0.75 + 0.50i \\ -2.09 + 7.56i & 1.39 + 3.97i & -1.25 + 0.75i & -4.82 - 5.67i \\ 6.18 + 9.79i & -0.92 - 0.62i & 0.00 + 0.00i & -2.50 - 0.50i \end{pmatrix}.$$

The program first calls F08NVF (ZGEBAL) to balance the matrix; it then computes the Schur factorization of the balanced matrix, by reduction to Hessenberg form and the QR algorithm. Then it calls F08QXF (ZTREVC) to compute the right eigenvectors of the balanced matrix, and finally calls F08NWF (ZGEBAK) to transform the eigenvectors back to eigenvectors of the original matrix A.

9.1 Program Text

```
    Program f08nvfe

!     F08NVF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgebak, zgebal, zgehrd, zhseqr,   &
                             ztrevc, zunghr
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)             :: firstnz
      Integer                           :: i, ifail, ihi, ilo, info, j, lda,    &
                                           ldh, ldvl, ldvr, lwork, m, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), h(:,:), tau(:), vl(:,:),   &
                                            vr(:,:), w(:), work(:)
```

```
      Real (Kind=nag_wp), Allocatable  :: rwork(:), scale(:)
      Logical                          :: select(1)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: aimag, real
!     .. Executable Statements ..
      Write (nout,*) 'F08NVF Example Program Results'
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldvl = 1
      lda = n
      ldh = n
      ldvr = n
      lwork = 64*n
      Allocate (a(lda,n),h(ldh,n),tau(n),vl(ldvl,1),vr(ldvr,n),w(n), &
        work(lwork),rwork(n),scale(n))

!     Read A from data file
      Read (nin,*)(a(i,1:n),i=1,n)

!     Balance A
!     The NAG name equivalent of zgebal is f08nvf
      Call zgebal('Both',n,a,lda,ilo,ihi,scale,info)

!     Reduce A to upper Hessenberg form H = (Q**H)*A*Q
!     The NAG name equivalent of zgehrd is f08nsf
      Call zgehrd(n,ilo,ihi,a,lda,tau,work,lwork,info)

!     Copy A to H and VR
      h(1:n,1:n) = a(1:n,1:n)
      vr(1:n,1:n) = a(1:n,1:n)

!     Form Q explicitly, storing the result in VR
!     The NAG name equivalent of zunghr is f08ntf
      Call zunghr(n,1,n,vr,ldvr,tau,work,lwork,info)

!     Calculate the eigenvalues and Schur factorization of A
!     The NAG name equivalent of zhseqr is f08psf
      Call zhseqr('Schur form','Vectors',n,ilo,ihi,h,ldh,w,vr,ldvr,work,lwork, &
        info)

      Write (nout,*)
      If (info>0) Then
        Write (nout,*) 'Failure to converge.'
      Else
        Write (nout,*) 'Eigenvalues'
        Write (nout,99999)(' (',real(w(i)),',',aimag(w(i)),')',i=1,n)
        Flush (nout)

!       Calculate the eigenvectors of A, storing the result in VR
!       The NAG name equivalent of ztrevc is f08qxf
        Call ztrevc('Right','Backtransform',select,n,h,ldh,vl,ldvl,vr,ldvr,n, &
          m,work,rwork,info)

!       The NAG name equivalent of zgebak is f08nwf
        Call zgebak('Both','Right',n,ilo,ihi,scale,m,vr,ldvr,info)

!       Print eigenvectors

        Write (nout,*)
        Flush (nout)

!       Normalize the eigenvectors
        Do i = 1, m
          Do j = n, 1, -1
            If (vr(j,i)/=(0._nag_wp,0._nag_wp)) Then
              firstnz = vr(j,i)
            End If
          End Do
```

```
          vr(1:n,i) = vr(1:n,i)/firstnz
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,m,vr,ldvr,'Bracketed','F7.4', &
          'Contents of array VR','Integer',rlabs,'Integer',clabs,80,0,ifail)

      End If

99999 Format ((3X,4(A,F7.4,A,F7.4,A:)))
    End Program f08nvfe
```

9.2 Program Data

```
F08NVF Example Program Data
  4                                                          :Value of N
 ( 1.50,-2.75) ( 0.00, 0.00) ( 0.00, 0.00) ( 0.00, 0.00)
 (-8.06,-1.24) (-2.50,-0.50) ( 0.00, 0.00) (-0.75, 0.50)
 (-2.09, 7.56) ( 1.39, 3.97) (-1.25, 0.75) (-4.82,-5.67)
 ( 6.18, 9.79) (-0.92,-0.62) ( 0.00, 0.00) (-2.50,-0.50)   :End of matrix A
```

9.3 Program Results

```
 F08NVF Example Program Results

 Eigenvalues
    (-1.2500, 0.7500) (-1.5000,-0.4975) (-3.5000,-0.5025) ( 1.5000,-2.7500)

 Contents of array VR
                     1                  2                  3                  4
 1  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000,-0.0000) ( 1.0000, 0.0000)
 2  ( 0.0000, 0.0000) ( 1.0000,-0.0000) ( 1.0000, 0.0000) (-1.4269,-1.6873)
 3  ( 1.0000, 0.0000) (-9.7405,-0.0846) ( 0.6466, 1.5212) ( 5.3497, 1.5369)
 4  ( 0.0000, 0.0000) (-0.9215,-0.6177) ( 0.9215, 0.6177) (-0.0819, 3.0107)
```