NAG Library Routine Document

F08WNF (ZGGEV)

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

F08WNF (ZGGEV) computes for a pair of n by n complex nonsymmetric matrices (A, B) the generalized eigenvalues and, optionally, the left and/or right generalized eigenvectors using the QZ algorithm.

2 Specification

```
SUBROUTINE F08WNF (JOBVL, JOBVR, N, A, LDA, B, LDB, ALPHA, BETA, VL, LDVL,      &
                   VR, LDVR, WORK, LWORK, RWORK, INFO)

INTEGER                 N, LDA, LDB, LDVL, LDVR, LWORK, INFO
REAL (KIND=nag_wp)      RWORK(max(1,8*N))
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*), ALPHA(N), BETA(N), VL(LDVL,*),        &
                        VR(LDVR,*), WORK(max(1,LWORK))
CHARACTER(1)            JOBVL, JOBVR
```

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

3 Description

A generalized eigenvalue for a pair of matrices (A, B) is a scalar λ or a ratio $\alpha/\beta = \lambda$, such that $A - \lambda B$ is singular. It is usually represented as the pair (α, β), as there is a reasonable interpretation for $\beta = 0$, and even for both being zero.

The right generalized eigenvector v_j corresponding to the generalized eigenvalue λ_j of (A, B) satisfies

$$Av_j = \lambda_j B v_j.$$

The left generalized eigenvector u_j corresponding to the generalized eigenvalue λ_j of (A, B) satisfies

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

where u_j^{H} is the conjugate-transpose of u_j.

All the eigenvalues and, if required, all the eigenvectors of the complex generalized eigenproblem $Ax = \lambda Bx$, where A and B are complex, square matrices, are determined using the QZ algorithm. The complex QZ algorithm consists of three stages:

1. A is reduced to upper Hessenberg form (with real, non-negative subdiagonal elements) and at the same time B is reduced to upper triangular form.

2. A is further reduced to triangular form while the triangular form of B is maintained and the diagonal elements of B are made real and non-negative. This is the generalized Schur form of the pair (A, B).

   This routine does not actually produce the eigenvalues λ_j, but instead returns α_j and β_j such that

   $$\lambda_j = \alpha_j/\beta_j, \qquad j = 1, 2, \ldots, n.$$

   The division by β_j becomes your responsibility, since β_j may be zero, indicating an infinite eigenvalue.

3. If the eigenvectors are required they are obtained from the triangular matrices and then transformed back into the original coordinate system.

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

Wilkinson J H (1979) Kronecker's canonical form and the QZ algorithm *Linear Algebra Appl.* **28** 285–303

5 Parameters

1: JOBVL – CHARACTER(1) *Input*

On entry: if JOBVL = 'N', do not compute the left generalized eigenvectors.

If JOBVL = 'V', compute the left generalized eigenvectors.

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

2: JOBVR – CHARACTER(1) *Input*

On entry: if JOBVR = 'N', do not compute the right generalized eigenvectors.

If JOBVR = 'V', compute the right generalized eigenvectors.

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

3: N – INTEGER *Input*

On entry: n, the order of the matrices A and B.

Constraint: N ≥ 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 matrix A in the pair (A, B).

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 F08WNF (ZGGEV) is called.

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

6: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the matrix B in the pair (A, B).

On exit: B has been overwritten.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08WNF (ZGGEV) is called.

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

8: ALPHA(N) – COMPLEX (KIND=nag_wp) array *Output*

On exit: see the description of BETA.

9: BETA(N) – COMPLEX (KIND=nag_wp) array *Output*

On exit: ALPHA(j)/BETA(j), for $j = 1, 2, \ldots, \mathrm{N}$, will be the generalized eigenvalues.

Note: the quotients ALPHA(j)/BETA(j) may easily overflow or underflow, and BETA(j) may even be zero. Thus, you should avoid naively computing the ratio α_j/β_j. However, $\max|\alpha_j|$ will always be less than and usually comparable with $\|\mathrm{A}\|_2$ in magnitude, and $\max|\beta_j|$ will always be less than and usually comparable with $\|\mathrm{B}\|_2$.

10: 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 generalized eigenvectors u_j are stored one after another in the columns of VL, in the same order as the corresponding eigenvalues. Each eigenvector will be scaled so the largest component will have |real part| + |imag. part| = 1.

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

11: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08WNF (ZGGEV) is called.

Constraints:

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

12: 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 generalized eigenvectors v_j are stored one after another in the columns of VR, in the same order as the corresponding eigenvalues. Each eigenvector will be scaled so the largest component will have |real part| + |imag. part| = 1.

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

13: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08WNF (ZGGEV) is called.

Constraints:

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

14: 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.

15: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08WNF (ZGGEV) 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 workspace by, say, $nb \times \mathrm{N}$, where nb is the optimal ***block size***.

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

16: RWORK($\max(1, 8 \times \mathrm{N})$) – REAL (KIND=nag_wp) array *Workspace*

17: 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.

INFO $= 1$ to N

The QZ iteration failed. No eigenvectors have been calculated, but ALPHA(j) and BETA(j) should be correct for $j = \mathrm{INFO} + 1, \dots, \mathrm{N}$.

INFO $= \mathrm{N} + 1$

Unexpected error returned from F08XSF (ZHGEQZ).

INFO $= \mathrm{N} + 2$

Error returned from F08YXF (ZTGEVC).

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrices $(A + E)$ and $(B + F)$, where

$$\|(E, F)\|_F = O(\epsilon)\|(A, B)\|_F,$$

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* (1999) for further details.

Note: interpretation of results obtained with the QZ algorithm often requires a clear understanding of the effects of small changes in the original data. These effects are reviewed in Wilkinson (1979), in relation to the significance of small values of α_j and β_j. It should be noted that if α_j and β_j are **both** small for any j, it may be that no reliance can be placed on **any** of the computed eigenvalues $\lambda_i = \alpha_i / \beta_i$. You are recommended to study Wilkinson (1979) and, if in difficulty, to seek expert advice on determining the sensitivity of the eigenvalues to perturbations in the data.

8 Further Comments

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

The real analogue of this routine is F08WAF (DGGEV).

9 Example

This example finds all the eigenvalues and right eigenvectors of the matrix pair (A, B), where

$$A = \begin{pmatrix} -21.10 - 22.50i & 53.50 - 50.50i & -34.50 + 127.50i & 7.50 + 0.50i \\ -0.46 - 7.78i & -3.50 - 37.50i & -15.50 + 58.50i & -10.50 - 1.50i \\ 4.30 - 5.50i & 39.70 - 17.10i & -68.50 + 12.50i & -7.50 - 3.50i \\ 5.50 + 4.40i & 14.40 + 43.30i & -32.50 - 46.00i & -19.00 - 32.50i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.00 - 5.00i & 1.60 + 1.20i & -3.00 + 0.00i & 0.00 - 1.00i \\ 0.80 - 0.60i & 3.00 - 5.00i & -4.00 + 3.00i & -2.40 - 3.20i \\ 1.00 + 0.00i & 2.40 + 1.80i & -4.00 - 5.00i & 0.00 - 3.00i \\ 0.00 + 1.00i & -1.80 + 2.40i & 0.00 - 4.00i & 4.00 - 5.00i \end{pmatrix}.$$

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.

9.1 Program Text

```
    Program f08wnfe

!     F08WNF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02amf, zggev
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: small
      Integer                          :: i, info, j, lda, ldb, ldvr, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), alpha(:), b(:,:), beta(:), &
                                          vr(:,:), work(:)
      Complex (Kind=nag_wp)            :: dummy(1,1)
      Real (Kind=nag_wp), Allocatable  :: rwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08WNF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldvr = n
      Allocate (a(lda,n),alpha(n),b(ldb,n),beta(n),vr(ldvr,n),rwork(8*n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of zggev is f08wnf
      Call zggev('No left vectors','Vectors (right)',n,a,lda,b,ldb,alpha,beta, &
        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 in the matrices A and B

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

!     Solve the generalized eigenvalue problem

!     The NAG name equivalent of zggev is f08wnf
      Call zggev('No left vectors','Vectors (right)',n,a,lda,b,ldb,alpha,beta, &
        dummy,1,vr,ldvr,work,lwork,rwork,info)

!     Normalize the eigenvectors
      Do i = 1, n
        vr(1:n,i) = vr(1:n,i)/vr(1,i)
      End Do
```

```
      If (info>0) Then
        Write (nout,*)
        Write (nout,99999) 'Failure in ZGGEV. INFO =', info
      Else
        small = x02amf()
        Do j = 1, n
          Write (nout,*)
          If ((abs(alpha(j)))*small>=abs(beta(j))) Then
            Write (nout,99998) 'Eigenvalue(', j, ')', &
              ' is numerically infinite or undetermined', 'ALPHA(', j, ') = ', &
              alpha(j), ', BETA(', j, ') = ', beta(j)
          Else
            Write (nout,99997) 'Eigenvalue(', j, ') = ', alpha(j)/beta(j)
          End If
          Write (nout,*)
          Write (nout,99996) 'Eigenvector(', j, ')', (vr(i,j),i=1,n)
        End Do

      End If

99999 Format (1X,A,I4)
99998 Format (1X,A,I2,2A/1X,2(A,I2,A,'(',1P,E11.4,',',1P,E11.4,')'))
99997 Format (1X,A,I2,A,'(',1P,E11.4,',',1P,E11.4,')')
99996 Format (1X,A,I2,A/3(1X,'(',1P,E11.4,',',1P,E11.4,')':))
    End Program f08wnfe
```

9.2 Program Data

```
F08WNF Example Program Data
  4                                                         : Value of N
  (-21.10,-22.50) ( 53.50,-50.50) (-34.50,127.50) (  7.50,  0.50)
  ( -0.46, -7.78) ( -3.50,-37.50) (-15.50, 58.50) (-10.50, -1.50)
  (  4.30, -5.50) ( 39.70,-17.10) (-68.50, 12.50) ( -7.50, -3.50)
  (  5.50,  4.40) ( 14.40, 43.30) (-32.50,-46.00) (-19.00,-32.50) : End of A
  (  1.00, -5.00) (  1.60,  1.20) ( -3.00,  0.00) (  0.00, -1.00)
  (  0.80, -0.60) (  3.00, -5.00) ( -4.00,  3.00) ( -2.40, -3.20)
  (  1.00,  0.00) (  2.40,  1.80) ( -4.00, -5.00) (  0.00, -3.00)
  (  0.00,  1.00) ( -1.80,  2.40) (  0.00, -4.00) (  4.00, -5.00) : End of B
```

9.3 Program Results

```
 F08WNF Example Program Results

 Eigenvalue( 1) = ( 3.0000E+00,-9.0000E+00)

 Eigenvector( 1)
 ( 1.0000E+00,-1.0839E-17) ( 1.6000E-01,-1.2000E-01) ( 1.2000E-01, 1.6000E-01)
 (-1.6000E-01, 1.2000E-01)

 Eigenvalue( 2) = ( 2.0000E+00,-5.0000E+00)

 Eigenvector( 2)
 ( 1.0000E+00,-5.4042E-18) ( 4.5714E-03,-3.4286E-03) ( 6.2857E-02, 8.1974E-17)
 (-8.5613E-17, 6.2857E-02)

 Eigenvalue( 3) = ( 3.0000E+00,-1.0000E+00)

 Eigenvector( 3)
 ( 1.0000E+00,-7.1362E-19) ( 1.6000E-01,-1.2000E-01) ( 1.2000E-01,-1.6000E-01)
 ( 1.6000E-01, 1.2000E-01)

 Eigenvalue( 4) = ( 4.0000E+00,-5.0000E+00)

 Eigenvector( 4)
 ( 1.0000E+00,-4.1134E-18) ( 8.8889E-03,-6.6667E-03) (-3.3333E-02,-3.6029E-17)
 (-2.0098E-16, 1.5556E-01)
```