NAG Library Routine Document

F08WQF (ZGGEV3)

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

F08WQF (ZGGEV3) 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 F08WQF (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 ***zggev3***.

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

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

   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 (2012) *Matrix Computations* (4th 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 Arguments

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$(1, \text{N})$.

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 F08WQF (ZGGEV3) is called.

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

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

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

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 F08WQF (ZGGEV3) is called.

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

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 $\|A\|_2$ in magnitude, and $\max|\beta_j|$ will always be less than and usually comparable with $\|B\|_2$.

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 F08WQF (ZGGEV3) 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 F08WQF (ZGGEV3) is called.

Constraints:

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

14: WORK(max(1, 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 F08WQF (ZGGEV3) 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, $nb \times (\mathrm{N} \times 6)$, where nb is the optimal ***block size***.

Constraint: 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

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 and BETA should be correct from element $\langle value \rangle$.

INFO $= \mathrm{N} + 1$

The QZ iteration failed with an unexpected error, please contact NAG.

INFO $= \mathrm{N} + 2$

A failure occurred in F08YXF (ZTGEVC) while computing generalized eigenvectors.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for 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 Parallelism and Performance

F08WQF (ZGGEV3) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08WQF (ZGGEV3) 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

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

The real analogue of this routine is F08WCF (DGGEV3).

10 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

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

10.1 Program Text

```
    Program f08wqfe

!     F08WQF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ajf, x04daf, zggev3
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: scal
      Integer                          :: i, ifail, info, k, 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, all, conjg, maxloc, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08WQF Example Program Results'
      Write (nout,*)
!     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 zggev3 is f08wqf
      Call zggev3('No left vectors','Vectors (right)',n,a,lda,b,ldb,alpha,     &
        beta,dummy,1,vr,ldvr,dummy,lwork,rwork,info)

      lwork = 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 zggev3 is f08wqf
      Call zggev3('No left vectors','Vectors (right)',n,a,lda,b,ldb,alpha,     &
        beta,dummy,1,vr,ldvr,work,lwork,rwork,info)

      If (info>0) Then
        Write (nout,*)
        Write (nout,99999) 'Failure in ZGGEV3. INFO =', info
      Else If (all(abs(beta(1:n))>x02ajf())) Then
!       Re-normalize the eigenvectors, largest absolute element real
        Do i = 1, n
          rwork(1:n) = abs(vr(1:n,i))
          k = maxloc(rwork(1:n),1)
          scal = conjg(vr(k,i))/rwork(k)
          vr(1:n,i) = vr(1:n,i)*scal
        End Do
        alpha(1:n) = alpha(1:n)/beta(1:n)
        ifail = 0
        Call x04daf('Gen',' ',1,n,alpha,1,'Eigenvalues:',ifail)
        Write (nout,*)
        Call x04daf('Gen',' ',n,n,vr,ldvr,'Right Eigenvectors (columns):',     &
          ifail)
      Else
        Write (nout,*) 'Some of the eigenvalues are infinite.'
        Write (nout,*)
        ifail = 0
        Call x04daf('Gen',' ',1,n,alpha,1,'Alpha',ifail)
        Call x04daf('Gen',' ',1,n,beta,1,'Beta',ifail)
      End If

99999 Format (1X,A,I4)
    End Program f08wqfe
```

10.2 Program Data

```
F08WQF 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
```

10.3 Program Results

```
 F08WQF Example Program Results

 Eigenvalues:
             1           2           3           4
 1      3.0000      2.0000      3.0000      4.0000
       -9.0000     -5.0000     -1.0000     -5.0000

 Right Eigenvectors (columns):
          1       2       3       4
 1   0.8424  0.7342  0.9778  0.9111
     0.0000  0.0000  0.0000  0.0000

 2   0.1348  0.0034  0.1564  0.0081
    -0.1011 -0.0025 -0.1173 -0.0061
```

```
 3   0.1011  0.0461  0.1173 -0.0304
     0.1348  0.0000 -0.1564 -0.0000

 4  -0.1348 -0.0000  0.1564 -0.0000
     0.1011  0.0461  0.1173  0.1417
```