NAG Library Routine Document

F08WCF (DGGEV3)

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

F08WCF (DGGEV3) computes for a pair of n by n real nonsymmetric matrices (A, B) the generalized eigenvalues and, optionally, the left and/or right generalized eigenvectors using the QZ algorithm.

2 Specification

```
SUBROUTINE F08WCF (JOBVL, JOBVR, N, A, LDA, B, LDB, ALPHAR, ALPHAI,         &
                   BETA, VL, LDVL, VR, LDVR, WORK, LWORK, INFO)

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

The routine may be called by its LAPACK name ***dggev3***.

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

$$Av_j = \lambda_j Bv_j.$$

The left eigenvector u_j corresponding to the 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 generalized eigenproblem $Ax = \lambda Bx$, where A and B are real, square matrices, are determined using the QZ algorithm. The QZ algorithm consists of four stages:

1. A is reduced to upper Hessenberg form and at the same time B is reduced to upper triangular form.
2. A is further reduced to quasi-triangular form while the triangular form of B is maintained. This is the real generalized Schur form of the pair (A, B).
3. The quasi-triangular form of A is reduced to triangular form and the eigenvalues extracted. 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. Pairs of complex eigenvalues occur with α_j/β_j and α_{j+1}/β_{j+1} complex conjugates, even though α_j and α_{j+1} are not conjugate.
4. 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: $\mathrm{N} \geq 0$.

4: $\mathrm{A}(\mathrm{LDA},*)$ – REAL (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 F08WCF (DGGEV3) is called.

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

6: $\mathrm{B}(\mathrm{LDB},*)$ – REAL (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 F08WCF (DGGEV3) is called.

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

8: ALPHAR(N) – REAL (KIND=nag_wp) array *Output*

On exit: the element ALPHAR(j) contains the real part of α_j.

9: ALPHAI(N) – REAL (KIND=nag_wp) array *Output*

On exit: the element ALPHAI(j) contains the imaginary part of α_j.

10: BETA(N) – REAL (KIND=nag_wp) array *Output*

On exit: $(\text{ALPHAR}(j) + \text{ALPHAI}(j) \times i)/\text{BETA}(j)$, for $j = 1, 2, \ldots, \text{N}$, will be the generalized eigenvalues.

If ALPHAI(j) is zero, then the jth eigenvalue is real; if positive, then the jth and $(j+1)$st eigenvalues are a complex conjugate pair, with ALPHAI($j+1$) negative.

Note: the quotients $\text{ALPHAR}(j)/\text{BETA}(j)$ and $\text{ALPHAI}(j)/\text{BETA}(j)$ 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 $\|B\|_2$.

11: VL(LDVL, $*$) – REAL (KIND=nag_wp) array *Output*

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

If the jth eigenvalue is real, then $u_j = \text{VL}(:, j)$, the jth column of VL.

If the jth and $(j+1)$th eigenvalues form a complex conjugate pair, then $u_j = \text{VL}(:, j) + i \times \text{VL}(:, j+1)$ and $u(j+1) = \text{VL}(:, j) - i \times \text{VL}(:, j+1)$. Each eigenvector will be scaled so the largest component has $|\text{real part}| + |\text{imag. part}| = 1$.

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

12: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08WCF (DGGEV3) is called.

Constraints:

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

13: VR(LDVR, $*$) – REAL (KIND=nag_wp) array *Output*

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

If the jth eigenvalue is real, then $v_j = \text{VR}(:, j)$, the jth column of VR.

If the jth and $(j+1)$th eigenvalues form a complex conjugate pair, then $v_j = \text{VR}(:, j) + i \times \text{VR}(:, j+1)$ and $v_{j+1} = \text{VR}(:, j) - i \times \text{VR}(:, j+1)$. Each eigenvector will be scaled so the largest component has $|\text{real part}| + |\text{imag. part}| = 1$.

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

14: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08WCF (DGGEV3) is called.

Constraints:

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

15: WORK(max(1, LWORK)) – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO $= 0$, WORK(1) contains the minimum value of LWORK required for optimal performance.

16: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08WCF (DGGEV3) 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, $6 \times nb \times \mathrm{N}$, where nb is the optimal ***block size***.

Constraint: LWORK $\geq \max(1, 8 \times \mathrm{N})$.

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 ALPHAR(j), ALPHAI(j) and BETA(j) 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 F08YKF (DTGEVC) 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

F08WCF (DGGEV3) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08WCF (DGGEV3) 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 complex analogue of this routine is F08WQF (ZGGEV3).

10 Example

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

$$A = \begin{pmatrix} 3.9 & 12.5 & -34.5 & -0.5 \\ 4.3 & 21.5 & -47.5 & 7.5 \\ 4.3 & 21.5 & -43.5 & 3.5 \\ 4.4 & 26.0 & -46.0 & 6.0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1.0 & 2.0 & -3.0 & 1.0 \\ 1.0 & 3.0 & -5.0 & 4.0 \\ 1.0 & 3.0 & -4.0 & 3.0 \\ 1.0 & 3.0 & -4.0 & 4.0 \end{pmatrix}.$$

10.1 Program Text

```
    Program f08wcfe

!     F08WCF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dggev3, m01def, m01edf, nag_wp, x02ajf, x04caf,   &
                             x04daf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: zero = 0.0_nag_wp
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: scal
      Integer                          :: i, ifail, info, j, k, lda, ldb,      &
                                          ldvr, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: eigval(:), eigvec(:,:)
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alphai(:), alphar(:),        &
                                          b(:,:), beta(:), vr(:,:), work(:)
      Real (Kind=nag_wp)               :: dummy(1,1)
      Integer, Allocatable             :: irank(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, all, cmplx, conjg, maxloc, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08WCF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldvr = n
      Allocate (a(lda,n),alphai(n),alphar(n),b(ldb,n),beta(n),vr(ldvr,n),      &
        eigvec(n,n),eigval(n),irank(n))
```

```
!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of dggev3 is f08wcf
      Call dggev3('No left vectors','Vectors (right)',n,a,lda,b,ldb,alphar,    &
        alphai,beta,dummy,1,vr,ldvr,dummy,lwork,info)

      lwork = nint(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 dggev3 is f08wcf
      Call dggev3('No left vectors','Vectors (right)',n,a,lda,b,ldb,alphar,    &
        alphai,beta,dummy,1,vr,ldvr,work,lwork,info)
      If (info>0) Then
        Write (nout,*)
        Write (nout,99999) 'Failure in DGGEV3. INFO =', info
        Go To 100
      End If

!     Re-normalize the eigenvectors, largest absolute element real
      j = 0
      Do i = 1, n
        If (alphai(i)==zero) Then
          eigvec(1:n,i) = cmplx(vr(1:n,i),zero,kind=nag_wp)
        Else If (j==0) Then
          eigvec(1:n,i) = cmplx(vr(1:n,i),vr(1:n,i+1),kind=nag_wp)
          j = 1
        Else
          eigvec(1:n,i) = cmplx(vr(1:n,i-1),-vr(1:n,i),kind=nag_wp)
          j = 0
        End If
        work(1:n) = abs(eigvec(1:n,i))
        k = maxloc(work(1:n),1)
        scal = conjg(eigvec(k,i))/abs(eigvec(k,i))
        eigvec(1:n,i) = eigvec(1:n,i)*scal
      End Do

!     If eigenvalues are finite, order by descending absolute values
      If (all(abs(beta(1:n))>x02ajf())) Then
!       add small amount to alphai to distinguish conjugates
        alphai(1:n) = alphai(1:n) + x02ajf()*10.0_nag_wp
        eigval(1:n) = cmplx(alphar(1:n),alphai(1:n),kind=nag_wp)
        eigval(1:n) = eigval(1:n)/beta(1:n)
        work(1:n) = abs(eigval(1:n))
        ifail = 0
        Call m01def(work,n,1,n,1,1,'Descending',irank,ifail)
        Call m01edf(eigval,1,n,irank,ifail)

!       Print ordered eigenvalues
        ifail = 0
        Call x04daf('Gen',' ',1,n,eigval,1,'Eigenvalues:',ifail)

!       Order the eigenvectors in the same way and print
        Do j = 1, n
          eigval(1:n) = eigvec(j,1:n)
          Call m01edf(eigval,1,n,irank,ifail)
          eigvec(j,1:n) = eigval(1:n)
        End Do

        Write (nout,*)
        ifail = 0
        Call x04daf('Gen',' ',n,n,eigvec,n,'Right Eigenvectors (columns):',    &
          ifail)
      Else
        Write (nout,*) 'Some of the eigenvalues are infinite'
```

```
        Write (nout,*)
        ifail = 0
        Call x04caf('Gen',' ',1,n,alphar,1,'Alpha (real):',ifail)
        Call x04caf('Gen',' ',1,n,alphai,1,'Alpha (imag):',ifail)
        Call x04caf('Gen',' ',1,n,beta,1,'Beta:',ifail)
      End If
100   Continue

99999 Format (1X,A,I4)
    End Program f08wcfe
```

10.2 Program Data

```
F08WCF Example Program Data
   4                        :Value of N
   3.9  12.5 -34.5  -0.5
   4.3  21.5 -47.5   7.5
   4.3  21.5 -43.5   3.5
   4.4  26.0 -46.0   6.0 :End of matrix A
   1.0   2.0  -3.0   1.0
   1.0   3.0  -5.0   4.0
   1.0   3.0  -4.0   3.0
   1.0   3.0  -4.0   4.0 :End of matrix B
```

10.3 Program Results

```
 F08WCF Example Program Results
 Eigenvalues:
            1           2           3           4
 1      3.0000      3.0000      4.0000      2.0000
        4.0000     -4.0000      0.0000      0.0000

 Right Eigenvectors (columns):
         1       2       3       4
 1   0.7122  0.7122  1.0000  1.0000
     0.0000  0.0000  0.0000  0.0000

 2   0.1424  0.1424  0.0111  0.0057
     0.0000 -0.0000  0.0000  0.0000

 3   0.0855  0.0855 -0.0333  0.0629
    -0.1140  0.1140 -0.0000  0.0000

 4   0.0855  0.0855  0.1556  0.0629
    -0.1140  0.1140  0.0000  0.0000
```