NAG Library Routine Document

F08WAF (DGGEV)

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

F08WAF (DGGEV) 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 F08WAF (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 ***dggev***.

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

$$Av_j = \lambda_j B v_j.$$

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 (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: $\mathrm{N} \geq 0$.

4: A(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 F08WAF (DGGEV) is called.

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

6: B(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 F08WAF (DGGEV) 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 F08WAF (DGGEV) 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 F08WAF (DGGEV) is called.

Constraints:

if JOBVR = 'V', LDVR $\geq \max(1, \text{N})$;
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 F08WAF (DGGEV) 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* × N, where *nb* is the optimal ***block size***.

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

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 ALPHAR(j), ALPHAI(j), and BETA(j) should be correct for $j =$ INFO $+ 1, \ldots,$ N.

INFO $=$ N $+ 1$

Unexpected error returned from F08XEF (DHGEQZ).

INFO $=$ N $+ 2$

Error returned from F08YKF (DTGEVC).

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 complex analogue of this routine is F08WNF (ZGGEV).

9 Example

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

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

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 f08wafe

!     F08WAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dggev, nag_wp, x02amf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: eig
      Real (Kind=nag_wp)               :: small
      Integer                          :: i, info, j, lda, ldb, ldvr, lwork, n
      Logical                          :: pair
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alphai(:), alphar(:),       &
                                          b(:,:), beta(:), vr(:,:), work(:)
      Real (Kind=nag_wp)               :: dummy(1,1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, cmplx, max, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08WAF 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))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of dggev is f08waf
      Call dggev('No left vectors','Vectors (right)',n,a,lda,b,ldb,alphar, &
        alphai,beta,dummy,1,vr,ldvr,dummy,lwork,info)

!     Make sure that there is enough workspace for blocksize nb.
      lwork = max((nb+7)*n,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 dggev is f08waf
      Call dggev('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 DGGEV. INFO =', info
      Else
        small = x02amf()
        pair = .False.
        Do j = 1, n
          Write (nout,*)
          If ((abs(alphar(j))+abs(alphai(j)))*small>=abs(beta(j))) Then
            Write (nout,99998) 'Eigenvalue(', j, ')', &
              ' is numerically infinite or undetermined', 'ALPHAR(', j, &
              ') = ', alphar(j), ', ALPHAI(', j, ') = ', alphai(j), ', BETA(', &
              j, ') = ', beta(j)
          Else
            If (alphai(j)==0.0E0_nag_wp) Then
              Write (nout,99997) 'Eigenvalue(', j, ') = ', alphar(j)/beta(j)
            Else
              eig = cmplx(alphar(j),alphai(j),kind=nag_wp)/ &
                cmplx(beta(j),kind=nag_wp)
              Write (nout,99996) 'Eigenvalue(', j, ') = ', eig
            End If
          End If
          Write (nout,*)
          Write (nout,99995) 'Eigenvector(', j, ')'
          If (alphai(j)==0.0E0_nag_wp) Then
            Write (nout,99994)(vr(i,j),i=1,n)
          Else
            If (pair) Then
              Write (nout,99993)(vr(i,j-1),-vr(i,j),i=1,n)
            Else
              Write (nout,99993)(vr(i,j),vr(i,j+1),i=1,n)
            End If
            pair = .Not. pair
          End If
        End Do

      End If

99999 Format (1X,A,I4)
99998 Format (1X,A,I2,2A/1X,2(A,I2,A,1P,E11.4,3X),A,I2,A,1P,E11.4)
99997 Format (1X,A,I2,A,1P,E11.4)
99996 Format (1X,A,I2,A,'(',1P,E11.4,',',1P,E11.4,')')
99995 Format (1X,A,I2,A)
99994 Format (1X,1P,E11.4)
99993 Format (1X,'(',1P,E11.4,',',1P,E11.4,')')
    End Program f08wafe
```

9.2 Program Data

```
F08WAF 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
```

9.3 Program Results

```
 F08WAF Example Program Results

 Eigenvalue( 1) =  2.0000E+00

 Eigenvector( 1)
  1.0000E+00
  5.7143E-03
  6.2857E-02
  6.2857E-02
```

```
 Eigenvalue( 2) = ( 3.0000E+00, 4.0000E+00)

 Eigenvector( 2)
 (-4.3979E-01,-5.6021E-01)
 (-8.7958E-02,-1.1204E-01)
 (-1.4241E-01, 3.1418E-03)
 (-1.4241E-01, 3.1418E-03)

 Eigenvalue( 3) = ( 3.0000E+00,-4.0000E+00)

 Eigenvector( 3)
 (-4.3979E-01, 5.6021E-01)
 (-8.7958E-02, 1.1204E-01)
 (-1.4241E-01,-3.1418E-03)
 (-1.4241E-01,-3.1418E-03)

 Eigenvalue( 4) =  4.0000E+00

 Eigenvector( 4)
 -1.0000E+00
 -1.1111E-02
  3.3333E-02
 -1.5556E-01
```