

NAG Fortran 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,
1                BETA, VL, LDVL, VR, LDVR, WORK, LWORK, INFO)
    INTEGER
    N, LDA, LDB, LDVL, LDVR, LWORK, INFO
    double precision
    A(LDA,*), B(LDB,*), ALPHAR(*), ALPHAI(*), BETA(*),
1    VL(LDVL,*), VR(LDVR,*), WORK(*)
    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

$$Av_j = \lambda_j Bv_j.$$

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

$$u_j^H A = \lambda_j u_j^H B.$$

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:

- (i) A is reduced to upper Hessenberg form and at the same time B is reduced to upper triangular form.
- (ii) 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) .
- (iii) 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

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

The division by β_j becomes the responsibility of the user, since β_j may be zero, indicating an infinite eigenvalue. Pairs of complex eigenvalues occur with α_j / β_j and $\alpha_{j+1} / \beta_{j+1}$ complex conjugates, even though α_j and α_{j+1} are not conjugate.

- (iv) If the eigenvectors are required they are obtained from the triangular matrices and then transformed back into the original co-ordinate 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 URL: <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.
- 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.
- 3: N – INTEGER *Input*
On entry: n , the order of the matrices A and B .
Constraint: $N \geq 0$.
- 4: A(LDA,*) – **double precision** array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
On entry: the matrix A in the pair (A, B) .
On exit: 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: $LDA \geq \max(1, N)$.
- 6: B(LDB,*) – **double precision** array *Input/Output*
Note: the second dimension of the array B must be at least $\max(1, N)$.
On entry: the matrix B in the pair (A, B) .
On exit: 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: $LDB \geq \max(1, N)$.
- 8: ALPHAR(*) – **double precision** array *Output*
Note: the dimension of the array ALPHAR must be at least $\max(1, N)$.
On exit: the element ALPHAR(j) contains the real part of α_j .

- 9: ALPHAI(*) – **double precision** array Output
Note: the dimension of the array ALPHAI must be at least $\max(1, N)$.
On exit: the element ALPHAI(j) contains the imaginary part of α_j .
- 10: BETA(*) – **double precision** array Output
Note: the dimension of the array BETA must be at least $\max(1, N)$.
On exit: $(\text{ALPHAR}(j) + \text{ALPHAI}(j) \times i) / \text{BETA}(j)$, $j = 1, \dots, N$, will be the generalized eigenvalues. If ALPHAI(j) is zero, then the j th eigenvalue is real; if positive, then the j th 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 over- or underflow, and BETA(j) may even be zero. Thus, the user should avoid naively computing the ratio α_j / β_j . However, $\max |\alpha_j|$ will be always less than and usually comparable with $\|A\|_2$ in magnitude, and $\max |\beta_j|$ always less than and usually comparable with $\|B\|_2$.
- 11: VL(LDVL,*) – **double precision** array Output
Note: the second dimension of the array VL must be at least $\max(1, N)$.
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 j th eigenvalue is real, then $u_j = \text{VL}(:, j)$, the j th column of VL.
If the j th 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 |real part| + |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', $\text{LDVL} \geq \max(1, N)$;
 $\text{LDVL} \geq 1$ otherwise.
- 13: VR(LDVR,*) – **double precision** array Output
Note: the second dimension of the array VR must be at least $\max(1, N)$.
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 j th eigenvalue is real, then $v_j = \text{VR}(:, j)$, the j th column of VR. If the j th 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 |real part| + |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', $\text{LDVR} \geq \max(1, N)$;
 $\text{LDVR} \geq 1$ otherwise.

- 15: WORK(*) – **double precision** array *Workspace*
Note: the dimension of the array WORK must be at least $\max(1, \text{LWORK})$.
On exit: if INFO = 0, WORK(1) returns the optimal LWORK.
- 16: LWORK – INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F08WAF (DGGEV) is called.
 For good performance, LWORK must generally be larger than the minimum; increase workspace by, say, $nb \times N$, where nb is the block size.
 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.
Constraint: $\text{LWORK} \geq \max(1, 8 \times N)$.
- 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 $\text{INFO} = -i$, the i th argument had an illegal value.

INFO > 0 and $\text{INFO} \leq N$

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

INFO > N

= N + 1: QZ iteration failed in F08XEF (DHGEQZ).

= N + 2: error return 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 $\lambda_i = \alpha_i / \beta_i$. The user is 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

To find all the eigenvalues and right eigenvectors of the matrix pair (A, B) , where

$$A = \begin{pmatrix} 3.9 & 12.5 & -34.5 & -0.5 \\ 4.3 & 21.5 & -47.5 & 7.5 \\ 4.3 & 21.5 & -43.5 & 3.5 \\ 4.4 & 26.0 & -46.0 & 6.0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.0 & 2.0 & -3.0 & 1.0 \\ 1.0 & 3.0 & -5.0 & 4.0 \\ 1.0 & 3.0 & -4.0 & 3.0 \\ 1.0 & 3.0 & -4.0 & 4.0 \end{pmatrix}.$$

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

Note: the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      F08WAF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX
      PARAMETER        (NB=64,NMAX=10)
      INTEGER          LDA, LDB, LDVR, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LDVR=NMAX,LWORK=NMAX*(7+NB))
*      .. Local Scalars ..
      COMPLEX *16      EIG
      DOUBLE PRECISION SMALL
      INTEGER          I, INFO, J, LWKOPT, N
*      .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), ALPHAI(NMAX), ALPHAR(NMAX),
+      B(LDB,NMAX), BETA(NMAX), DUMMY(1,1),
+      VR(LDVR,NMAX), WORK(LWORK)
*      .. External Functions ..
      DOUBLE PRECISION X02AMF
      EXTERNAL          X02AMF
*      .. External Subroutines ..
      EXTERNAL          DGGEV
*      .. Intrinsic Functions ..
      INTRINSIC          ABS, CMPLX
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F08WAF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*          Read in the matrices A and B
*
*          READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*          READ (NIN,*) ((B(I,J),J=1,N),I=1,N)
*
*          Solve the generalized eigenvalue problem
*
*          CALL DGGEV('No left vectors','Vectors (right)',N,A,LDA,B,LDB,
+      ALPHAR,ALPHAI,BETA,DUMMY,1,VR,LDVR,WORK,LWORK,INFO)
*
      IF (INFO.GT.0) THEN
          WRITE (NOUT,*)
          WRITE (NOUT,99999) 'Failure in DGGEV. INFO =', INFO
      END IF
      99999

```

```

      ELSE
        SMALL = X02AMF()
        DO 20 J = 1, N
          WRITE (NOUT,*)
          IF ((ABS(ALPHAR(J))+ABS(ALPHAI(J)))*SMALL.GE.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).EQ.0.0D0) THEN
              WRITE (NOUT,99997) 'Eigenvalue(', J, ') = ',
+                ALPHAR(J)/BETA(J)
            ELSE
              EIG = CMPLX(ALPHAR(J),ALPHAI(J))/BETA(J)
              WRITE (NOUT,99996) 'Eigenvalue(', J, ') = ', EIG
            END IF
          END IF
          WRITE (NOUT,*)
          WRITE (NOUT,99995) 'Eigenvector(', J, ') '
          IF (ALPHAI(J).EQ.0.0D0) THEN
            WRITE (NOUT,99994) (VR(I,J),I=1,N)
          ELSE IF (ALPHAI(J).GT.0.0D0) THEN
            WRITE (NOUT,99993) (VR(I,J),VR(I,J+1),I=1,N)
          ELSE
            WRITE (NOUT,99993) (VR(I,J-1),-VR(I,J),I=1,N)
          END IF
20      CONTINUE
*
        LWKOPT = WORK(1)
        IF (LWORK.LT.LWKOPT) THEN
          WRITE (NOUT,*)
          WRITE (NOUT,99992) 'Optimum workspace required = ',
+            LWKOPT, 'Workspace provided      = ', LWORK
        END IF
      END IF
    ELSE
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'NMAX too small'
    END IF
  STOP
*
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,')')
99992 FORMAT (1X,A,I5,/1X,A,I5)
END

```

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
