NAG Fortran Library Routine Document

F08WBF (DGGEVX)

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

F08WBF (DGGEVX) 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.

Optionally it also computes a balancing transformation to improve the conditioning of the eigenvalues and eigenvectors, reciprocal condition numbers for the eigenvalues, and reciprocal condition numbers for the right eigenvectors.

2 Specification

```
      SUBROUTINE F08WBF (BALANC, JOBVL, JOBVR, SENSE, N, A, LDA, B, LDB,
     1                   ALPHAR, ALPHAI, BETA, VL, LDVL, VR, LDVR, ILO, IHI,
     2                   LSCALE, RSCALE, ABNRM, BBNRM, RCONDE, RCONDV, WORK,
     3                   LWORK, IWORK, BWORK, INFO)
      INTEGER            N, LDA, LDB, LDVL, LDVR, ILO, IHI, LWORK, IWORK(*),
     1                   INFO
      double precision   A(LDA,*), B(LDB,*), ALPHAR(*), ALPHAI(*), BETA(*),
     1                   VL(LDVL,*), VR(LDVR,*), LSCALE(*), RSCALE(*), ABNRM,
     2                   BBNRM, RCONDE(*), RCONDV(*), WORK(*)
      LOGICAL            BWORK(*)
      CHARACTER*1        BALANC, JOBVL, JOBVR, SENSE
```

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

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

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

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

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

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 α_{j+1}/β_{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.

For details of the balancing option, see Section 3 of the document for F08WHF (DGGBAL).

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: BALANC – CHARACTER*1 *Input*

On entry: specifies the balance option to be performed:

if BALANC = 'N', do not diagonally scale or permute;
if BALANC = 'P', permute only;
if BALANC = 'S', scale only;
if BALANC = 'B', both permute and scale.

Computed reciprocal condition numbers will be for the matrices after permuting and/or balancing. Permuting does not change condition numbers (in exact arithmetic), but balancing does. In the absence of other information, BALANC = 'B' is recommended.

2: JOBVL – CHARACTER*1 *Input*

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

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

3: JOBVR – CHARACTER*1 *Input*

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

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

4: SENSE – CHARACTER*1 *Input*

On entry: determines which reciprocal condition numbers are computed:

if SENSE = 'N', none are computed;
if SENSE = 'E', computed for eigenvalues only;
if SENSE = 'V', computed for eigenvectors only;
if SENSE = 'B', computed for eigenvalues and eigenvectors.

5: N – INTEGER *Input*

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

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

6: A(LDA,$*$) – ***double precision*** 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: has been overwritten. If JOBVL = 'V' or JOBVR = 'V' or both, then A contains the first part of the real Schur form of the 'balanced' versions of the input A and B.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08WBF (DGGEVX) is called.

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

8: $\mathrm{B}(\mathrm{LDB},*)$ – ***double precision*** 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: has been overwritten. If JOBVL = 'V' or JOBVR = 'V' or both, then B contains the second part of the real Schur form of the 'balanced' versions of the input A and B.

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08WBF (DGGEVX) is called.

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

10: $\mathrm{ALPHAR}(*)$ – ***double precision*** array *Output*

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

On exit: the element $\mathrm{ALPHAR}(j)$ contains the real part of α_j.

11: $\mathrm{ALPHAI}(*)$ – ***double precision*** array *Output*

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

On exit: the element $\mathrm{ALPHAI}(j)$ contains the imaginary part of α_j.

12: $\mathrm{BETA}(*)$ – ***double precision*** array *Output*

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

On exit: $(\mathrm{ALPHAR}(j) + \mathrm{ALPHAI}(j) \times i)/\mathrm{BETA}(j)$, $j = 1, \ldots, \mathrm{N}$, will be the generalized eigenvalues. If $\mathrm{ALPHAI}(j)$ is zero, then the jth eigenvalue is real; if positive, then the jth and $(j+1)$st eigenvalues are a complex conjugate pair, with $\mathrm{ALPHAI}(j+1)$ negative.

Note: the quotients $\mathrm{ALPHAR}(j)/\mathrm{BETA}(j)$ and $\mathrm{ALPHAI}(j)/\mathrm{BETA}(j)$ may easily over- or underflow, and $\mathrm{BETA}(j)$ 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 $\|\mathrm{A}\|_2$ in magnitude, and $\max|\beta_j|$ always less than and usually comparable with $\|\mathrm{B}\|_2$.

13: $\mathrm{VL}(\mathrm{LDVL},*)$ – ***double precision*** array *Output*

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

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

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

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

14: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08WBF (DGGEVX) is called.

Constraints:

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

15: 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 their eigenvalues. If the jth eigenvalue is real, then $v(j) = \mathrm{VR}(:, j)$, the jth column of VR. If the jth and $(j+1)$th eigenvalues form a complex conjugate pair, then $v_j = \mathrm{VR}(:, j) + i \times \mathrm{VR}(:, j+1)$ and $v_{j+1} = \mathrm{VR}(:, j) - i \times \mathrm{VR}(:, j+1)$.

Each eigenvector will be scaled so the largest component has $|\text{real part}| + |\text{imag. part}| = 1$.

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

16: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08WBF (DGGEVX) is called.

Constraints:

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

17: ILO – INTEGER *Output*
18: IHI – INTEGER *Output*

On exit: ILO and IHI are integer values such that $\mathrm{A}(i, j) = 0$ and $\mathrm{B}(i, j) = 0$ if $i > j$ and $j = 1, \ldots, \mathrm{ILO} - 1$ or $i = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

If BALANC = 'N' or 'S', ILO = 1 and IHI = N.

19: LSCALE($*$) – ***double precision*** array *Output*

Note: the dimension of the array LSCALE must be at least $\max(1, N)$.

On exit: details of the permutations and scaling factors applied to the left side of A and B.

If pl_j is the index of the row interchanged with row j, and dl_j is the scaling factor applied to row j, then:

$\mathrm{LSCALE}(j) = pl_j$ for $j = 1, \ldots, \mathrm{ILO} - 1$;

$\mathrm{LSCALE} = dl_j$ for $j = \mathrm{ILO}, \ldots, \mathrm{IHI}$;

$\mathrm{LSCALE} = pl_j$ for $j = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

The order in which the interchanges are made is N to $\mathrm{IHI} + 1$, then 1 to $\mathrm{ILO} - 1$.

20: RSCALE($*$) – ***double precision*** array *Output*

Note: the dimension of the array RSCALE must be at least $\max(1, N)$.

On exit: details of the permutations and scaling factors applied to the right side of A and B.

If pr_j is the index of the column interchanged with column j, and dr_j is the scaling factor applied to column j, then:

$\mathrm{RSCALE}(j) = pr_j$ for $j = 1, \ldots, \mathrm{ILO} - 1$;
if $\mathrm{RSCALE} = dr_j$ for $j = \mathrm{ILO}, \ldots, \mathrm{IHI}$;
if $\mathrm{RSCALE} = pr_j$ for $j = \mathrm{IHI} + 1, \ldots, \mathrm{N}$.

The order in which the interchanges are made is N to IHI + 1, then 1 to ILO − 1.

21: ABNRM – ***double precision*** *Output*

On exit: the 1-norm of the balanced matrix A.

22: BBNRM – ***double precision*** *Output*

On exit: the 1-norm of the balanced matrix B.

23: RCONDE($*$) – ***double precision*** array *Output*

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

On exit: if SENSE = 'E' or 'B', the reciprocal condition numbers of the eigenvalues, stored in consecutive elements of the array. For a complex conjugate pair of eigenvalues two consecutive elements of RCONDE are set to the same value. Thus $\mathrm{RCONDE}(j)$, $\mathrm{RCONDV}(j)$, and the jth columns of VL and VR all correspond to the jth eigenpair.

If SENSE = 'V', RCONDE is not referenced.

24: RCONDV($*$) – ***double precision*** array *Output*

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

On exit: if SENSE = 'V' or 'B', the estimated reciprocal condition numbers of the eigenvectors, stored in consecutive elements of the array. For a complex eigenvector two consecutive elements of RCONDV are set to the same value.

If SENSE = 'E', RCONDV is not referenced.

25: WORK($*$) – ***double precision*** array *Workspace*

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

On exit: if INFO = 0, $\mathrm{WORK}(1)$ returns the optimal LWORK.

26: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08WBF (DGGEVX) is called.

For good performance, LWORK must generally be larger than the minimum; increase workspace by, say, $nb \times \mathrm{N}$, 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: $\mathrm{LWORK} \geq \max(1, 8 \times \mathrm{N})$.

27: IWORK($*$) – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least $\mathrm{N} + 6$.

If SENSE = 'E', IWORK is not referenced.

28: BWORK($*$) – LOGICAL array *Workspace*

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

If SENSE = 'N', BWORK is not referenced.

29: 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 $= -i$, the ith argument had an illegal value.

INFO = 1leqN

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

INFO $>$ N

$= \mathrm{N} + 1$: other than QZ iteration failed in F08XEF (DHGEQZ).

$= \mathrm{N} + 2$: error return from F08YKF (DTGEVC).

7 Accuracy

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

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

and ϵ is the ***machine precision***.

An approximate error bound on the chordal distance between the ith computed generalized eigenvalue w and the corresponding exact eigenvalue λ is

$$\epsilon \times \|\mathrm{ABNRM}, \mathrm{BBNRM}\|_2/\mathrm{RCONDE}(i).$$

An approximate error bound for the angle between the ith computed eigenvector VL(i) or VR(i) is given by

$$\epsilon \times \|\mathrm{ABNRM}, \mathrm{BBNRM}\|_2/\mathrm{RCONDV}(i).$$

For further explanation of the reciprocal condition numbers RCONDE and RCONDV, see Section 4.11 of Anderson *et al.* (1999).

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 F08WPF (ZGGEVX).

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

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

together with estimates of the condition number and forward error bounds for each eigenvalue and eigenvector. The option to balance the matrix pair is used.

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.

```
*     F08WBF 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*NB+2*NMAX*NMAX)
*     .. Local Scalars ..
      COMPLEX *16      EIG
      DOUBLE PRECISION ABNORM, ABNRM, BBNRM, EPS, ERBND, RCND, SMALL,
     +                 TOL
      INTEGER          I, IHI, ILO, INFO, J, LWKOPT, N
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), ALPHAI(NMAX), ALPHAR(NMAX),
     +                 B(LDB,NMAX), BETA(NMAX), DUMMY(1,1),
     +                 LSCALE(NMAX), RCONDE(NMAX), RCONDV(NMAX),
     +                 RSCALE(NMAX), VR(LDVR,NMAX), WORK(LWORK)
      INTEGER          IWORK(NMAX+6)
      LOGICAL          BWORK(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION F06BNF, X02AJF, X02AMF
      EXTERNAL         F06BNF, X02AJF, X02AMF
*     .. External Subroutines ..
      EXTERNAL         DGGEVX
*     .. Intrinsic Functions ..
      INTRINSIC        ABS, CMPLX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08WBF 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 DGGEVX('Balance','No vectors (left)','Vectors (right)',
     +               'Both reciprocal condition numbers',N,A,LDA,B,LDB,
     +               ALPHAR,ALPHAI,BETA,DUMMY,1,VR,LDVR,ILO,IHI,LSCALE,
     +               RSCALE,ABNRM,BBNRM,RCONDE,RCONDV,WORK,LWORK,IWORK,
     +               BWORK,INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*)
```

```
            WRITE (NOUT,99999) 'Failure in DGGEVX. INFO =', INFO
         ELSE
*
*           Compute the machine precision, the safe range parameter
*           SMALL and sqrt(ABNRM**2+BBNRM**2)
*
            EPS = X02AJF()
            SMALL = X02AMF()
            ABNORM = F06BNF(ABNRM,BBNRM)
            TOL = EPS*ABNORM
*
*           Print out eigenvalues and vectors and associated condition
*           number and bounds
*
            DO 20 J = 1, N
*
*              Print out information on the jth eigenvalue
*
               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
               RCND = RCONDE(J)
               WRITE (NOUT,*)
               WRITE (NOUT,99995) 'Reciprocal condition number = ', RCND
               IF (RCND.GT.0.0D0) THEN
                  ERBND = TOL/RCND
                  WRITE (NOUT,99995) 'Error bound                 = ',
     +              ERBND
               ELSE
                  WRITE (NOUT,*) 'Error bound is infinite'
               END IF
*
*              Print out information on the jth eigenvector
*
               WRITE (NOUT,*)
               WRITE (NOUT,99994) 'Eigenvector(', J, ')'
               IF (ALPHAI(J).EQ.0.0D0) THEN
                  WRITE (NOUT,99993) (VR(I,J),I=1,N)
               ELSE IF (ALPHAI(J).GT.0.0D0) THEN
                  WRITE (NOUT,99992) (VR(I,J),VR(I,J+1),I=1,N)
               ELSE
                  WRITE (NOUT,99992) (VR(I,J-1),-VR(I,J),I=1,N)
               END IF
               RCND = RCONDV(J)
               WRITE (NOUT,*)
               WRITE (NOUT,99995) 'Reciprocal condition number = ', RCND
               IF (RCND.GT.0.0D0) THEN
                  ERBND = TOL/RCND
                  WRITE (NOUT,99995) 'Error bound                 = ',
     +              ERBND
               ELSE
                  WRITE (NOUT,*) 'Error bound is infinite'
               END IF
   20       CONTINUE
*
            LWKOPT = WORK(1)
            IF (LWORK.LT.LWKOPT) THEN
               WRITE (NOUT,*)
```

```
               WRITE (NOUT,99991) '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),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,1P,E8.1)
99994 FORMAT (1X,A,I2,A)
99993 FORMAT (1X,1P,E11.4)
99992 FORMAT (1X,'(',1P,E11.4,',',1P,E11.4,')')
99991 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08WBF 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

```
 F08WBF Example Program Results

 Eigenvalue( 1) =  2.0000E+00

 Reciprocal condition number =  9.5E-02
 Error bound                 =  2.5E-14

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

 Reciprocal condition number =  1.3E-01
 Error bound                 =  1.9E-14

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

 Reciprocal condition number =  1.7E-01
 Error bound                 =  1.4E-14

 Eigenvector( 2)
 (-4.2550E-01,-5.7450E-01)
 (-8.5099E-02,-1.1490E-01)
 (-1.4298E-01,-8.6125E-04)
 (-1.4298E-01,-8.6125E-04)

 Reciprocal condition number =  3.8E-02
 Error bound                 =  6.2E-14

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

 Reciprocal condition number =  1.7E-01
 Error bound                 =  1.4E-14
```

```
Eigenvector( 3)
(-4.2550E-01, 5.7450E-01)
(-8.5099E-02, 1.1490E-01)
(-1.4298E-01, 8.6125E-04)
(-1.4298E-01, 8.6125E-04)

Reciprocal condition number =  3.8E-02
Error bound                 =  6.2E-14

Eigenvalue( 4) =  4.0000E+00

Reciprocal condition number =  5.1E-01
Error bound                 =  4.6E-15

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

Reciprocal condition number =  7.1E-02
Error bound                 =  3.3E-14
```