NAG Library Routine Document

F08PBF (DGEESX)

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

F08PBF (DGEESX) computes the eigenvalues, the real Schur form T, and, optionally, the matrix of Schur vectors Z for an n by n real nonsymmetric matrix A.

2 Specification

```
SUBROUTINE F08PBF (JOBVS, SORT, SELECT, SENSE, N, A, LDA, SDIM, WR, WI,     &
                   VS, LDVS, RCONDE, RCONDV, WORK, LWORK, IWORK, LIWORK,    &
                   BWORK, INFO)
INTEGER            N, LDA, SDIM, LDVS, LWORK, IWORK(max(1,LIWORK)),         &
                   LIWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), WR(*), WI(*), VS(LDVS,*), RCONDE, RCONDV,      &
                   WORK(max(1,LWORK))
LOGICAL            SELECT, BWORK(*)
CHARACTER(1)       JOBVS, SORT, SENSE
EXTERNAL           SELECT
```

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

3 Description

The real Schur factorization of A is given by

$$A = ZTZ^{\mathrm{T}},$$

where Z, the matrix of Schur vectors, is orthogonal and T is the real Schur form. A matrix is in real Schur form if it is upper quasi-triangular with 1 by 1 and 2 by 2 blocks. 2 by 2 blocks will be standardized in the form

$$\begin{bmatrix} a & b \\ c & a \end{bmatrix}$$

where $bc < 0$. The eigenvalues of such a block are $a \pm \sqrt{bc}$.

Optionally, F08PBF (DGEESX) also orders the eigenvalues on the diagonal of the real Schur form so that selected eigenvalues are at the top left; computes a reciprocal condition number for the average of the selected eigenvalues (RCONDE); and computes a reciprocal condition number for the right invariant subspace corresponding to the selected eigenvalues (RCONDV). The leading columns of Z form an orthonormal basis for this invariant subspace.

For further explanation of the reciprocal condition numbers RCONDE and RCONDV, see Section 4.8 of Anderson *et al.* (1999) (where these quantities are called s and sep respectively).

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

5 Arguments

1: JOBVS – CHARACTER(1) *Input*

On entry: if JOBVS = 'N', Schur vectors are not computed.

If JOBVS = 'V', Schur vectors are computed.

Constraint: JOBVS = 'N' or 'V'.

2: SORT – CHARACTER(1) *Input*

On entry: specifies whether or not to order the eigenvalues on the diagonal of the Schur form.

SORT = 'N'
Eigenvalues are not ordered.

SORT = 'S'
Eigenvalues are ordered (see SELECT).

Constraint: SORT = 'N' or 'S'.

3: SELECT – LOGICAL FUNCTION, supplied by the user. *External Procedure*

If SORT = 'S', SELECT is used to select eigenvalues to sort to the top left of the Schur form.

If SORT = 'N', SELECT is not referenced and F08PBF (DGEESX) may be called with the dummy function F08PAZ.

An eigenvalue $\mathrm{WR}(j) + \sqrt{-1} \times \mathrm{WI}(j)$ is selected if $\mathrm{SELECT}(\mathrm{WR}(j), \mathrm{WI}(j))$ is .TRUE.. If either one of a complex conjugate pair of eigenvalues is selected, then both are. Note that a selected complex eigenvalue may no longer satisfy $\mathrm{SELECT}(\mathrm{WR}(j), \mathrm{WI}(j)) =$ .TRUE. after ordering, since ordering may change the value of complex eigenvalues (especially if the eigenvalue is ill-conditioned); in this case INFO is set to $\mathrm{N} + 2$ (see INFO below).

The specification of SELECT is:

```
FUNCTION SELECT (WR, WI)
LOGICAL SELECT
REAL (KIND=nag_wp) WR, WI
```

1: WR – REAL (KIND=nag_wp) *Input*
2: WI – REAL (KIND=nag_wp) *Input*

On entry: the real and imaginary parts of the eigenvalue.

SELECT must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which F08PBF (DGEESX) is called. Arguments denoted as *Input* must **not** be changed by this procedure.

4: SENSE – CHARACTER(1) *Input*

On entry: determines which reciprocal condition numbers are computed.

SENSE = 'N'
None are computed.

SENSE = 'E'
Computed for average of selected eigenvalues only.

SENSE = 'V'
Computed for selected right invariant subspace only.

SENSE = 'B'
Computed for both.

If SENSE = 'E', 'V' or 'B', SORT = 'S'.

Constraint: SENSE = 'N', 'E', 'V' or 'B'.

5: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N ≥ 0.

6: A(LDA, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the n by n matrix A.

On exit: A is overwritten by its real Schur form T.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08PBF (DGEESX) is called.

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

8: SDIM – INTEGER *Output*

On exit: if SORT = 'N', SDIM = 0.

If SORT = 'S', SDIM = number of eigenvalues (after sorting) for which SELECT is .TRUE.. (Complex conjugate pairs for which SELECT is .TRUE. for either eigenvalue count as 2.)

9: WR($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: see the description of WI.

10: WI($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: WR and WI contain the real and imaginary parts, respectively, of the computed eigenvalues in the same order that they appear on the diagonal of the output Schur form T. Complex conjugate pairs of eigenvalues will appear consecutively with the eigenvalue having the positive imaginary part first.

11: VS(LDVS, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array VS must be at least max(1, N) if JOBVS = 'V', and at least 1 otherwise.

On exit: if JOBVS = 'V', VS contains the orthogonal matrix Z of Schur vectors.

If JOBVS = 'N', VS is not referenced.

12: LDVS – INTEGER *Input*

On entry: the first dimension of the array VS as declared in the (sub)program from which F08PBF (DGEESX) is called.

Constraints:

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

13: RCONDE – REAL (KIND=nag_wp) *Output*

On exit: if SENSE = 'E' or 'B', contains the reciprocal condition number for the average of the selected eigenvalues.

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

14: RCONDV – REAL (KIND=nag_wp) *Output*

On exit: if SENSE = 'V' or 'B', RCONDV contains the reciprocal condition number for the selected right invariant subspace.

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

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 F08PBF (DGEESX) is called.

If LWORK = −1, a workspace query is assumed; the routine only calculates upper bounds on the optimal sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error messages related to LWORK or LIWORK is issued.

If SENSE = 'E', 'V' or 'B', LWORK ≥ N + 2 × SDIM × (N − SDIM), where SDIM is the number of selected eigenvalues computed by this routine. Note that N + 2 × SDIM × (N − SDIM) ≤ N + N × N/2.

Note also that an error is only returned if LWORK < max(1, 3 × N), but if SENSE = 'E', 'V' or 'B' this may not be large enough.

Suggested value: for optimal performance, LWORK must generally be larger than the minimum; increase the workspace by, say, $nb \times$ N, where nb is the optimal ***block size*** for F08NEF (DGEHRD).

Constraint: LWORK ≥ max(1, 3 × N).

17: IWORK(max(1, LIWORK)) – INTEGER array *Workspace*

On exit: if INFO = 0, IWORK(1) returns the optimal LIWORK.

18: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08PBF (DGEESX) is called.

If LIWORK = −1, a workspace query is assumed; the routine only calculates upper bounds on the optimal sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error messages related to LWORK or LIWORK is issued.

Constraints:

if SENSE = 'V' or 'B', LIWORK ≥ SDIM × (N − SDIM);
otherwise LIWORK ≥ 1.

Note: SDIM × (N − SDIM) ≤ N × N/4. Note also that an error is only returned if LIWORK < 1, but if SENSE = 'V' or 'B' this may not be large enough.

19: BWORK(∗) – LOGICAL array *Workspace*

Note: the dimension of the array BWORK must be at least max(1, N) if SORT ≠ 'N', and at least 1 otherwise.

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

20: 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

If INFO $= i$ and $i \leq$ N, the QR algorithm failed to compute all the eigenvalues.

INFO $=$ N $+ 1$

The eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned).

INFO $=$ N $+ 2$

After reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELECT = .TRUE.. This could also be caused by underflow due to scaling.

7 Accuracy

The computed Schur factorization satisfies

$$A + E = ZTZ^{\mathrm{T}},$$

where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***. See Section 4.8 of Anderson *et al.* (1999) for further details.

8 Parallelism and Performance

F08PBF (DGEESX) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08PBF (DGEESX) 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 F08PPF (ZGEESX).

10 Example

This example finds the Schur factorization of the matrix

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix},$$

such that the real positive eigenvalues of A are the top left diagonal elements of the Schur form, T. Estimates of the condition numbers for the selected eigenvalue cluster and corresponding invariant subspace are also returned.

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.

10.1 Program Text

```
!   F08PBF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module f08pbfe_mod

!     F08PBF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                             :: select
!     .. Parameters ..
      Integer, Parameter, Public         :: nb = 64, nin = 5, nout = 6
      Logical, Parameter, Public         :: check_fac = .True.,                  &
                                            print_cond = .False.
    Contains
      Function select(wr,wi)

!       Logical function select for use with DGEESX (F08PBF)
!       Returns the value .TRUE. if the eigenvalue is real and positive

!       .. Function Return Value ..
        Logical                          :: select
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: wi, wr
!       .. Executable Statements ..
        select = (wr>0._nag_wp .And. wi==0._nag_wp)
        Return
      End Function select
    End Module f08pbfe_mod
    Program f08pbfe

!     F08PBF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: dgeesx, dgemm, dlange => f06raf, nag_wp, x02ajf,  &
                             x04caf
      Use f08pbfe_mod, Only: check_fac, nb, nin, nout, print_cond, select
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                 :: alpha, anorm, beta, eps, norm,       &
                                            rconde, rcondv, tol
      Integer                            :: i, ifail, info, lda, ldc, ldd, ldvs, &
                                            liwork, lwork, n, sdim
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), c(:,:), d(:,:), vs(:,:),     &
```

```
                                       wi(:), work(:), wr(:)
      Real (Kind=nag_wp)               :: dummy(1)
      Integer                          :: idum(1)
      Integer, Allocatable             :: iwork(:)
      Logical, Allocatable             :: bwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08PBF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldc = n
      ldd = n
      ldvs = n
      Allocate (a(lda,n),c(ldc,n),d(ldd,n),vs(ldvs,n),wi(n),wr(n),bwork(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
!     The NAG name equivalent of dgeesx is f08pbf
      Call dgeesx('Vectors (Schur)','Sort',select,                              &
        'Both reciprocal condition numbers',n,a,lda,sdim,wr,wi,vs,ldvs,rconde, &
        rcondv,dummy,lwork,idum,liwork,bwork,info)

!     Make sure that there is enough workspace for block size nb.
      liwork = max((n*n)/4,idum(1))
      lwork = max(n*(nb+2+n/2),nint(dummy(1)))
      Allocate (work(lwork),iwork(liwork))

!     Read in the matrix A
      Read (nin,*)(a(i,1:n),i=1,n)

!     Copy A into D
      d(1:n,1:n) = a(1:n,1:n)

!     Print matrix A
!     ifail: behaviour on error exit
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',n,n,a,lda,'Matrix A',ifail)

      Write (nout,*)
      Flush (nout)

!     Find the Frobenius norm of A
!     The NAG name equivalent of the LAPACK auxiliary dlange is f06raf
      anorm = dlange('Frobenius',n,n,a,lda,work)

!     Find the Schur factorization of A
!     The NAG name equivalent of dgeesx is f08pbf
      Call dgeesx('Vectors (Schur)','Sort',select,                              &
        'Both reciprocal condition numbers',n,a,lda,sdim,wr,wi,vs,ldvs,rconde, &
        rcondv,work,lwork,iwork,liwork,bwork,info)

      If (info/=0 .And. info/=(n+2)) Then
        Write (nout,99993) 'Failure in DGEESX. INFO =', info
        Go To 100
      End If

      If (check_fac) Then
!       Compute A - Z*T*Z^T from the factorization of A and store in matrix D
!       The NAG name equivalent of dgemm is f06yaf
        alpha = 1.0_nag_wp
        beta = 0.0_nag_wp
        Call dgemm('N','N',n,n,n,alpha,vs,ldvs,a,lda,beta,c,ldc)
        alpha = -1.0_nag_wp
        beta = 1.0_nag_wp
```

```
        Call dgemm('N','T',n,n,n,alpha,c,ldc,vs,ldvs,beta,d,ldd)

!       Find norm of matrix D and print warning if it is too large
!       f06raf is the NAG name equivalent of the LAPACK auxiliary dlange
        norm = dlange('O',ldd,n,d,ldd,work)
        If (norm>x02ajf()**0.8_nag_wp) Then
          Write (nout,*) 'Norm of A-(Z*T*Z^T) is much greater than 0.'
          Write (nout,*) 'Schur factorization has failed.'
          Go To 100
        End If
      End If

!     Print solution
      Write (nout,99999) 'Number of eigenvalues for which SELECT is true = ',  &
        sdim, '(dimension of invariant subspace)'

      Write (nout,*)
!     Print eigenvalues.
      Write (nout,*) 'Selected eigenvalues'
      Write (nout,99998)(' (',wr(i),',',wi(i),')',i=1,sdim)
      Write (nout,*)

      If (info==(n+2)) Then
        Write (nout,99997) '***Note that rounding errors mean ',               &
          'that leading eigenvalues in the Schur form',                        &
          'no longer satisfy SELECT = .TRUE.'
        Write (nout,*)
      End If
      Flush (nout)

      If (print_cond) Then
!       Print out the reciprocal condition numbers
        Write (nout,99996) 'Reciprocal of projection norm onto the invariant', &
          'subspace for the selected eigenvalues', 'RCONDE = ', rconde
        Write (nout,*)
        Write (nout,99995)                                                     &
          'Reciprocal condition number for the invariant subspace',            &
          'RCONDV = ', rcondv

!       Compute the machine precision
        eps = x02ajf()
        tol = eps*anorm

!       Print out the approximate asymptotic error bound on the
!       average absolute error of the selected eigenvalues given by
!       eps*norm(A)/RCONDE
        Write (nout,*)
        Write (nout,99994) 'Approximate asymptotic error bound for selected ', &
          'eigenvalues   = ', tol/rconde

!       Print out an approximate asymptotic bound on the maximum
!       angular error in the computed invariant subspace given by
!       eps*norm(A)/RCONDV
        Write (nout,99994)                                                     &
          'Approximate asymptotic error bound for the invariant ',             &
          'subspace = ', tol/rcondv
      End If
100   Continue

99999 Format (1X,A,I4,/,1X,A)
99998 Format (1X,A,F8.4,A,F8.4,A)
99997 Format (1X,2A,/,1X,A)
99996 Format (1X,A,/,1X,A,/,1X,A,1P,E8.1)
99995 Format (1X,A,/,1X,A,1P,E8.1)
99994 Format (1X,2A,1P,E8.1)
99993 Format (1X,A,I4)
    End Program f08pbfe
```

10.2 Program Data

```
F08PBF Example Program Data

  4                          :Value of N

  0.35   0.45  -0.14  -0.17
  0.09   0.07  -0.54   0.35
 -0.44  -0.33  -0.03   0.17
  0.25  -0.32  -0.13   0.11 :End of matrix A
```

10.3 Program Results

```
 F08PBF Example Program Results

 Matrix A
          1       2       3       4
 1   0.3500  0.4500 -0.1400 -0.1700
 2   0.0900  0.0700 -0.5400  0.3500
 3  -0.4400 -0.3300 -0.0300  0.1700
 4   0.2500 -0.3200 -0.1300  0.1100

 Number of eigenvalues for which SELECT is true =    1
 (dimension of invariant subspace)

 Selected eigenvalues
  (  0.7995,  0.0000)
```