NAG Library Routine Document

F08PPF (ZGEESX)

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

F08PPF (ZGEESX) computes the eigenvalues, the Schur form T, and, optionally, the matrix of Schur vectors Z for an n by n complex nonsymmetric matrix A.

2 Specification

```
SUBROUTINE F08PPF (JOBVS, SORT, SELECT, SENSE, N, A, LDA, SDIM, W, VS,     &
                   LDVS, RCONDE, RCONDV, WORK, LWORK, RWORK, BWORK,        &
                   INFO)

INTEGER                 N, LDA, SDIM, LDVS, LWORK, INFO
REAL (KIND=nag_wp)      RCONDE, RCONDV, RWORK(*)
COMPLEX (KIND=nag_wp) A(LDA,*), W(*), VS(LDVS,*), WORK(max(1,LWORK))
LOGICAL                 SELECT, BWORK(*)
CHARACTER(1)            JOBVS, SORT, SENSE
EXTERNAL                SELECT
```

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

3 Description

The Schur factorization of A is given by

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

where Z, the matrix of Schur vectors, is unitary and T is the Schur form. A complex matrix is in Schur form if it is upper triangular.

Optionally, F08PPF (ZGEESX) also orders the eigenvalues on the diagonal of the 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 F08PPF (ZGEESX) may be called with the dummy function F08PNZ.

An eigenvalue W(j) is selected if SELECT(W(j)) is .TRUE..

The specification of SELECT is:

```
FUNCTION SELECT (W)
LOGICAL SELECT
COMPLEX (KIND=nag_wp) W
```

1: W – COMPLEX (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 F08PPF (ZGEESX) 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 $\geq$ 0.

6: A(LDA, ∗) – COMPLEX (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 Schur form T.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08PPF (ZGEESX) is called.

Constraint: LDA ≥ max(1, N).

8: SDIM – INTEGER *Output*

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

If SORT = 'S', SDIM = number of eigenvalues for which SELECT is .TRUE..

9: W(∗) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: contains the computed eigenvalues, in the same order that they appear on the diagonal of the output Schur form T.

10: VS(LDVS, ∗) – COMPLEX (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 unitary matrix Z of Schur vectors.

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

11: LDVS – INTEGER *Input*

On entry: the first dimension of the array VS as declared in the (sub)program from which F08PPF (ZGEESX) is called.

Constraints:

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

12: 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.

13: 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.

14: WORK(max(1, LWORK)) – COMPLEX (KIND=nag_wp) array *Workspace*

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

15: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08PPF (ZGEESX) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates an upper bound on 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.

If SENSE = 'E', 'V' or 'B', LWORK $\geq 2 \times$ SDIM $\times$ (N $-$ SDIM), where SDIM is the number of selected eigenvalues computed by this routine.

Note that $2 \times \text{SDIM} \times (\text{N} - \text{SDIM}) \leq \text{N} \times \text{N}/2$. Note also that an error is only returned if LWORK $< \max(1, 2 \times \text{N})$, 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, $nb \times \text{N}$, where nb is the optimal ***block size*** for F08NSF (ZGEHRD).

Constraint: LWORK $\geq \max(1, 2 \times \text{N})$.

16: RWORK$(*)$ – REAL (KIND=nag_wp) array *Workspace*

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

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

Note: the dimension of the array BWORK must be at least 1 if SORT = 'N', and at least $\max(1, \text{N})$ otherwise.

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

18: 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 $i \leq \text{N}$, the QR algorithm failed to compute all the eigenvalues.

INFO $= \text{N} + 1$

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

INFO $= \text{N} + 2$

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

$$A + E = ZTZ^{\text{H}},$$

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

F08PPF (ZGEESX) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08PPF (ZGEESX) 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 real analogue of this routine is F08PBF (DGEESX).

10 Example

This example finds the Schur factorization of the matrix

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix},$$

such that the eigenvalues of A with positive real part of 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

```
!   F08PPF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module f08ppfe_mod

!     F08PPF 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(w)

!       Logical function select for use with ZGEESX (F08PPF)
```

```
!       Returns the value .TRUE. if the real part of the eigenvalue
!       w is positive.

!       .. Function Return Value ..
        Logical                       :: select
!       .. Scalar Arguments ..
        Complex (Kind=nag_wp), Intent (In) :: w
!       .. Intrinsic Procedures ..
        Intrinsic                     :: real
!       .. Executable Statements ..
        select = (real(w)>0._nag_wp)
        Return
      End Function select
    End Module f08ppfe_mod
    Program f08ppfe

!     F08PPF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ajf, x04dbf, zgeesx, zgemm,           &
                             zlange => f06uaf
      Use f08ppfe_mod, Only: check_fac, nb, nin, nout, print_cond, select
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Complex (Kind=nag_wp)           :: alpha, beta
      Real (Kind=nag_wp)              :: anorm, eps, norm, rconde, rcondv,    &
                                         tol
      Integer                         :: i, ifail, info, lda, ldc, ldd, ldvs, &
                                         lwork, n, sdim
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), c(:,:), d(:,:), vs(:,:),  &
                                         w(:), work(:)
      Complex (Kind=nag_wp)           :: dummy(1)
      Real (Kind=nag_wp), Allocatable :: rwork(:)
      Logical, Allocatable            :: bwork(:)
      Character (1)                   :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: cmplx, max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08PPF 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),w(n),rwork(n),bwork(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of zgeesx is f08ppf
      Call zgeesx('Vectors (Schur)','Sort',select,                            &
        'Both reciprocal condition numbers',n,a,lda,sdim,w,vs,ldvs,rconde,     &
        rcondv,dummy,lwork,rwork,bwork,info)

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

!     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 x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.4','Matrix A',       &
        'Integer',rlabs,'Integer',clabs,80,0,ifail)

      Write (nout,*)
      Flush (nout)

!     Find the Frobenius norm of A
!     The NAG name equivalent of the LAPACK auxiliary zlange is f06uaf
      anorm = zlange('Frobenius',n,n,a,lda,rwork)

!     Find the Schur factorization of A
!     The NAG name equivalent of zgeesx is f08ppf
      Call zgeesx('Vectors (Schur)','Sort',select,                             &
        'Both reciprocal condition numbers',n,a,lda,sdim,w,vs,ldvs,rconde,      &
        rcondv,work,lwork,rwork,bwork,info)

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

      If (check_fac) Then
!       Compute A - Z*T*Z^H from the factorization of A and store in matrix D
!       The NAG name equivalent of zgemm is f06zaf
        alpha = cmplx(1,kind=nag_wp)
        beta = cmplx(0,kind=nag_wp)
        Call zgemm('N','N',n,n,n,alpha,vs,ldvs,a,lda,beta,c,ldc)
        alpha = cmplx(-1,kind=nag_wp)
        beta = cmplx(1,kind=nag_wp)
        Call zgemm('N','C',n,n,n,alpha,c,ldc,vs,ldvs,beta,d,ldd)

!       Find norm of matrix D and print warning if it is too large
!       f06uaf is the NAG name equivalent of the LAPACK auxiliary zlange
        norm = zlange('O',ldd,n,d,ldd,rwork)
        If (norm>x02ajf()**0.5_nag_wp) Then
          Write (nout,*) 'Norm of A-(Z*T*Z^H) 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)(i,w(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,I4,2X,' (',F7.4,',',F7.4,')',:)
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 f08ppfe
```

10.2 Program Data

```
F08PPF Example Program Data

  4                                                          :Value of N

(-3.97, -5.04)  (-4.11,  3.70)  (-0.34,  1.01)  ( 1.29, -0.86)
( 0.34, -1.50)  ( 1.52, -0.43)  ( 1.88, -5.38)  ( 3.36,  0.65)
( 3.31, -3.85)  ( 2.50,  3.45)  ( 0.88, -1.08)  ( 0.64, -1.48)
(-1.10,  0.82)  ( 1.81, -1.59)  ( 3.25,  1.33)  ( 1.57, -3.44) :End of matrix A
```

10.3 Program Results

```
 F08PPF Example Program Results

 Matrix A
                     1                  2                  3                  4
 1  (-3.9700,-5.0400) (-4.1100, 3.7000) (-0.3400, 1.0100) ( 1.2900,-0.8600)
 2  ( 0.3400,-1.5000) ( 1.5200,-0.4300) ( 1.8800,-5.3800) ( 3.3600, 0.6500)
 3  ( 3.3100,-3.8500) ( 2.5000, 3.4500) ( 0.8800,-1.0800) ( 0.6400,-1.4800)
 4  (-1.1000, 0.8200) ( 1.8100,-1.5900) ( 3.2500, 1.3300) ( 1.5700,-3.4400)

 Number of eigenvalues for which SELECT is true =    2
 (dimension of invariant subspace)

 Selected eigenvalues
    1   ( 7.9982,-0.9964)
    2   ( 3.0023,-3.9998)
```