

# NAG Library Routine Document

## F08PNF (ZGEES)

**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

F08PNF (ZGEES) 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 F08PNF (JOBVS, SORT, SELECT, N, A, LDA, SDIM, W, VS, LDVS, WORK, &
                  LWORK, RWORK, BWORK, INFO)

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

```

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

### 3 Description

The Schur factorization of  $A$  is given by

$$A = ZTZ^H,$$

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, F08PNF (ZGEES) also orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left. The leading columns of  $Z$  form an orthonormal basis for the invariant subspace corresponding to the selected eigenvalues.

### 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 Parameters

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 F08PNF (ZGEES) 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 F08PNF (ZGEES) is called. Parameters denoted as *Input* must **not** be changed by this procedure.
- 4: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 5: 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  $T$ .
- 6: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F08PNF (ZGEES) is called.  
*Constraint:*  $LDA \geq \max(1, N)$ .
- 7: SDIM – INTEGER *Output*  
*On exit:* if SORT = 'N', SDIM = 0.  
 If SORT = 'S', SDIM = number of eigenvalues for which SELECT is .TRUE..
- 8: 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  $T$ .

- 9: VS(LDVS,\*) – COMPLEX (KIND=nag\_wp) array Output  
**Note:** the second dimension of the array VS must be at least  $\max(1, N)$  if  $\text{JOBVS} = 'V'$ , and at least 1 otherwise.  
*On exit:* if  $\text{JOBVS} = 'V'$ , VS contains the unitary matrix  $Z$  of Schur vectors.  
 If  $\text{JOBVS} = 'N'$ , VS is not referenced.
- 10: LDVS – INTEGER Input  
*On entry:* the first dimension of the array VS as declared in the (sub)program from which F08PNF (ZGEES) is called.  
*Constraints:*  
     if  $\text{JOBVS} = 'V'$ ,  $\text{LDVS} \geq \max(1, N)$ ;  
     otherwise  $\text{LDVS} \geq 1$ .
- 11: WORK( $\max(1, \text{LWORK})$ ) – COMPLEX (KIND=nag\_wp) array Workspace  
*On exit:* if  $\text{INFO} = 0$ , the real part of  $\text{WORK}(1)$  contains the minimum value of LWORK required for optimal performance.
- 12: LWORK – INTEGER Input  
*On entry:* the dimension of the array WORK as declared in the (sub)program from which F08PNF (ZGEES) is called.  
 If  $\text{LWORK} = -1$ , 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, say  $2 \times N + nb \times N$ , where  $nb$  is the optimal **block size** for F08NSF (ZGEHRD).  
*Constraint:*  $\text{LWORK} \geq \max(1, 2 \times N)$ .
- 13: RWORK(\*) – REAL (KIND=nag\_wp) array Workspace  
**Note:** the dimension of the array RWORK must be at least  $\max(1, N)$ .
- 14: BWORK(\*) – LOGICAL array Workspace  
**Note:** the dimension of the array BWORK must be at least 1 if  $\text{SORT} = 'N'$ , and at least  $\max(1, N)$  otherwise.  
 If  $\text{SORT} = 'N'$ , BWORK is not referenced.
- 15: INFO – INTEGER Output  
*On exit:*  $\text{INFO} = 0$  unless the routine detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\text{INFO} < 0$

If  $\text{INFO} = -i$ , argument  $i$  had an illegal value. An explanatory message is output, and execution of the program is terminated.

$\text{INFO} = 1$  to  $N$

If  $\text{INFO} = i$  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

$$A + E = ZTZ^H,$$

where

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

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

## 8 Further Comments

The total number of floating point operations is proportional to  $n^3$ .

The real analogue of this routine is F08PAF (DGEES).

## 9 Example

This example finds the Schur factorization of the matrix

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

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 f08pnfe

!      F08PNF Example Program Text
!
!      Mark 24 Release. NAG Copyright 2012.
!
!      .. Use Statements ..
!      Use nag_library, Only: f08pnz, nag_wp, x02ajf, x04dbf, zgees, zgemm,      &
!                               zlange => f06uaf
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nb = 64, nin = 5, nout = 6
!      .. Local Scalars ..
!      Complex (Kind=nag_wp)       :: alpha, beta
!      Real (Kind=nag_wp)          :: norm
!      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)       :: wdum(1)
!      Real (Kind=nag_wp), Allocatable :: rwork(:)

```

```

Logical                :: dummy(1)
Character (1)         :: clabs(1), rlabs(1)
! .. Intrinsic Procedures ..
Intrinsic              :: cmplx, max, nint, real
! .. Executable Statements ..
Write (nout,*) 'F08PNF 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),vs(ldvs,n),c(ldc,n),d(ldd,n),w(n),rwork(n))

! Use routine workspace query to get optimal workspace.
lwork = -1
! The NAG name equivalent of zgees is f08pnf
Call zgees('Vectors (Schur)', 'No sort', f08pnz, n, a, lda, sdim, w, vs, ldvs, &
  wdum, lwork, rwork, dummy, info)

! Make sure that there is enough workspace for blocksize nb.
lwork = max((nb+1)*n, nint(real(wdum(1))))
Allocate (work(lwork))

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

! Copy A into D
d(1:ldd,1:n) = a(1:lda,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 Schur factorization

! The NAG name equivalent of zgees is f08pnf
Call zgees('Vectors (Schur)', 'No sort', f08pnz, n, a, lda, sdim, w, vs, ldvs, &
  work, lwork, rwork, dummy, info)

If (info>0) Then
  Write (nout,99999) 'Failure in ZGEES. INFO =', info
Else

! Compute  $A - Z^*T^*Z^*H$  from Schur 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())**0.8_nag_wp) Then
  Write (nout,*) 'Norm of A-(Z* $T^*Z^*H$ ) is much greater than 0.'
  Write (nout,*) 'Schur factorization has failed.'
Else
! Print eigenvalues.
Write (nout,*) 'Eigenvalues'

```

```

        Write (nout,99998)(w(i),i=1,n)
      End If
    End If

99999 Format (1X,A,I4)
99998 Format ((3X,4(' ',F7.4,',',',',F7.4,')':)))
      End Program f08pnfe

```

## 9.2 Program Data

F08PNF 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

```

## 9.3 Program Results

F08PNF 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)

```

Eigenvalues

```

(-6.0004,-6.9998) (-5.0000, 2.0060) ( 7.9982,-0.9964) ( 3.0023,-3.9998)

```

---