NAG Library Routine Document

F03BNF

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

F03BNF computes the determinant of a complex n by n matrix A. F07ARF (ZGETRF) must be called first to supply the matrix A in factorized form.

2 Specification

```
SUBROUTINE F03BNF (N, A, LDA, IPIV, D, ID, IFAIL)
INTEGER                 N, LDA, IPIV(N), ID(2), IFAIL
COMPLEX (KIND=nag_wp) A(LDA,*), D
```

3 Description

F03BNF computes the determinant of a complex n by n matrix A that has been factorized by a call to F07ARF (ZGETRF). The determinant of A is the product of the diagonal elements of U with the correct sign determined by the row interchanges.

4 References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag

5 Arguments

1: N – INTEGER *Input*

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

Constraint: N > 0.

2: A(LDA, $*$) – COMPLEX (KIND=nag_wp) array *Input*

Note: the second dimension of the array A must be at least N.

On entry: the n by n matrix A in factorized form as returned by F07ARF (ZGETRF).

3: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F03BNF is called.

Constraint: LDA $\geq$ N.

4: IPIV(N) – INTEGER array *Input*

On entry: the row interchanges used to factorize matrix A as returned by F07ARF (ZGETRF).

5: D – COMPLEX (KIND=nag_wp) *Output*

On exit: the mantissa of the real and imaginary parts of the determinant.

6: ID(2) – INTEGER array *Output*

On exit: the exponents for the real and imaginary parts of the determinant. The determinant, $d = (d_r, d_i)$, is returned as $d_r = D_r \times 2^j$ and $d_i = D_i \times 2^k$, where $\mathrm{D} = (D_r, D_i)$ and j and k are stored in the first and second elements respectively of the array ID on successful exit.

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{N} = \langle value \rangle$.
Constraint: $\mathrm{N} \geq 1$.

$\mathrm{IFAIL} = 3$

On entry, $\mathrm{LDA} = \langle value \rangle$ and $\mathrm{N} = \langle value \rangle$.
Constraint: $\mathrm{LDA} \geq \mathrm{N}$.

$\mathrm{IFAIL} = 4$

The matrix A is approximately singular.

$\mathrm{IFAIL} = -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

$\mathrm{IFAIL} = -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

$\mathrm{IFAIL} = -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The accuracy of the determinant depends on the conditioning of the original matrix. For a detailed error analysis, see page 107 of Wilkinson and Reinsch (1971).

8 Parallelism and Performance

F03BNF is not threaded in any implementation.

9 Further Comments

The time taken by F03BNF is approximately proportional to n.

10 Example

This example calculates the determinant of the complex matrix

$$\begin{pmatrix} 1 & 1+2i & 2+10i \\ 1+i & 3i & -5+14i \\ 1+i & 5i & -8+20i \end{pmatrix}.$$

10.1 Program Text

```
    Program f03bnfe

!     F03BNF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f03bnf, nag_wp, x04daf, zgetrf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: two = 2.0_nag_wp
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: d
      Integer                          :: i, ifail, info, lda, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:)
      Integer                          :: id(2)
      Integer, Allocatable             :: ipiv(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: aimag, real
!     .. Executable Statements ..
      Write (nout,*) 'F03BNF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),ipiv(n))

      Read (nin,*)(a(i,1:n),i=1,n)

!     LU Factorize A
!     The NAG name equivalent of zgetrf is f07arf
      Call zgetrf(n,n,a,lda,ipiv,info)
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04daf('g','n',n,n,a,lda,'Array A after factorization',ifail)

      Write (nout,*)
      Write (nout,*) 'Pivots'
      Write (nout,99999) ipiv(1:n)
      Write (nout,*)
      Flush (nout)
```

```
      ifail = 0
      Call f03bnf(n,a,lda,ipiv,d,id,ifail)

      Write (nout,99998) d, id
      Write (nout,*)
      Write (nout,99997) two**id(1)*real(d), two**id(2)*aimag(d)

99999 Format (1X,8(1X,I13))
99998 Format (1X,'D = (',F13.5,',',F13.5,'), ID = (',I0,', ',I0,')')
99997 Format (1X,'Value of determinant = (',E13.5,',',E13.5,')')
    End Program f03bnfe
```

10.2 Program Data

```
F03BNF Example Program Data
  3
 (1.0, 0.0)   (1.0, 2.0)   (2.0,10.0)
 (1.0, 1.0)   (0.0, 3.0)  (-5.0,14.0)
 (1.0, 1.0)   (0.0, 5.0)  (-8.0,20.0)
```

10.3 Program Results

```
 F03BNF Example Program Results

 Array A after factorization
             1          2          3
 1      1.0000     0.0000    -5.0000
        1.0000     3.0000    14.0000

 2      1.0000     0.0000    -3.0000
        0.0000     2.0000     6.0000

 3      0.5000     0.2500    -0.2500
       -0.5000     0.2500    -0.2500

 Pivots
              2             3             3

 D = (      0.06250,      0.00000), ID = (4, 0)

 Value of determinant = (  0.10000E+01,  0.00000E+00)
```