C06EBF – NAG Fortran Library Routine Document

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

C06EBF calculates the discrete Fourier transform of a Hermitian sequence of n complex data values. (No extra workspace required.)

2 Specification

```
SUBROUTINE C06EBF(X, N, IFAIL)
INTEGER           N, IFAIL
real              X(N)
```

3 Description

Given a Hermitian sequence of n complex data values z_j (i.e., a sequence such that z_0 is real and z_{n-j} is the complex conjugate of z_j, for $j = 1, 2, \ldots, n-1$) this routine calculates their discrete Fourier transform defined by:

$$\hat{x}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} z_j \times \exp\left(-i\frac{2\pi jk}{n}\right), \qquad k = 0, 1, \ldots, n-1.$$

(Note the scale factor of $\frac{1}{\sqrt{n}}$ in this definition.) The transformed values $\hat{x}_k$ are purely real (see also the Chapter Introduction).

To compute the inverse discrete Fourier transform defined by:

$$\hat{y}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} z_j \times \exp\left(+i\frac{2\pi jk}{n}\right),$$

this routine should be preceded by a call of C06GBF to form the complex conjugates of the z_j.

The routine uses the fast Fourier transform (FFT) algorithm (Brigham [1]). There are some restrictions on the value of n (see Section 5).

4 References

[1] Brigham E O (1973) *The Fast Fourier Transform* Prentice–Hall

5 Parameters

1: X(N) — ***real*** array *Input/Output*

On entry: the sequence to be transformed stored in Hermitian form. If the data values z_j are written as $x_j + iy_j$, and if X is declared with bounds (0:N−1) in the subroutine from which C06EBF is called, then for $0 \le j \le n/2$, x_j is contained in X(j), and for $1 \le j \le (n-1)/2$, y_j is contained in X($n-j$). (See also Section 2.1.2 of the Chapter Introduction and the Example Program.)

On exit: the components of the discrete Fourier transform $\hat{x}_k$. If X is declared with bounds (0:N−1) in the (sub)program from which C06EBF is called, then $\hat{x}_k$ is stored in X(k), for $k = 0, 1, \ldots, n-1$.

2: N — INTEGER *Input*

On entry: the number of data values, n. The largest prime factor of N must not exceed 19, and the total number of prime factors of N, counting repetitions, must not exceed 20.

Constraint: N > 1.

3: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

At least one of the prime factors of N is greater than 19.

IFAIL = 2

N has more than 20 prime factors.

IFAIL = 3

N $\leq$ 1.

IFAIL = 4

An unexpected error has occurred in an internal call. Check all subroutine calls and array dimensions. Seek expert help.

7 Accuracy

Some indication of accuracy can be obtained by performing a subsequent inverse transform and comparing the results with the original sequence (in exact arithmetic they would be identical).

8 Further Comments

The time taken by the routine is approximately proportional to $n \times \log n$, but also depends on the factorization of n. The routine is somewhat faster than average if the only prime factors of n are 2, 3 or 5; and fastest of all if n is a power of 2.

On the other hand, the routine is particularly slow if n has several unpaired prime factors, i.e., if the 'square-free' part of n has several factors. For such values of n, routine C06FBF (which requires an additional n elements of workspace) is considerably faster.

9 Example

This program reads in a sequence of real data values which is assumed to be a Hermitian sequence of complex data values stored in Hermitian form. The input sequence is expanded into a full complex sequence and printed alongside the original sequence. The discrete Fourier transform (as computed by C06EBF) is printed out.

The program then performs an inverse transform using C06EAF and C06GBF, and prints the sequence so obtained alongside the original data values.

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.

```
*     C06EBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX
```

```
      PARAMETER        (NMAX=20)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          IFAIL, J, N, N2, NJ
*     .. Local Arrays ..
      real             U(0:NMAX-1), V(0:NMAX-1), X(0:NMAX-1),
     +                 XX(0:NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         C06EAF, C06EBF, C06GBF
*     .. Intrinsic Functions ..
      INTRINSIC        MOD
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06EBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=140) N
      IF (N.GT.1 .AND. N.LE.NMAX) THEN
         DO 40 J = 0, N - 1
            READ (NIN,*) X(J)
            XX(J) = X(J)
   40    CONTINUE
         U(0) = X(0)
         V(0) = 0.0e0
         N2 = (N-1)/2
         DO 60 J = 1, N2
            NJ = N - J
            U(J) = X(J)
            U(NJ) = X(J)
            V(J) = X(NJ)
            V(NJ) = -X(NJ)
   60    CONTINUE
         IF (MOD(N,2).EQ.0) THEN
            U(N2+1) = X(N2+1)
            V(N2+1) = 0.0e0
         END IF
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence and corresponding complex sequence'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '         Data           Real      Imag'
         WRITE (NOUT,*)
         DO 80 J = 0, N - 1
            WRITE (NOUT,99999) J, X(J), '     ', U(J), V(J)
   80    CONTINUE
         IFAIL = 0
*
         CALL C06EBF(X,N,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Components of discrete Fourier transform'
         WRITE (NOUT,*)
         DO 100 J = 0, N - 1
            WRITE (NOUT,99999) J, X(J)
  100    CONTINUE
*
         CALL C06EAF(X,N,IFAIL)
         CALL C06GBF(X,N,IFAIL)
```

```
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence as restored by inverse transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '        Original  Restored'
         WRITE (NOUT,*)
         DO 120 J = 0, N - 1
            WRITE (NOUT,99998) J, XX(J), X(J)
  120    CONTINUE
         GO TO 20
      ELSE
         WRITE (NOUT,*) 'Invalid value of N'
      END IF
  140 STOP
*
99999 FORMAT (1X,I5,F10.5,A,2F10.5)
99998 FORMAT (1X,I5,2F10.5)
      END
```

9.2 Program Data

```
C06EBF Example Program Data
    7
  0.34907
  0.54890
  0.74776
  0.94459
  1.13850
  1.32850
  1.51370
```

9.3 Program Results

```
 C06EBF Example Program Results

 Original sequence and corresponding complex sequence

          Data           Real      Imag

     0   0.34907       0.34907   0.00000
     1   0.54890       0.54890   1.51370
     2   0.74776       0.74776   1.32850
     3   0.94459       0.94459   1.13850
     4   1.13850       0.94459  -1.13850
     5   1.32850       0.74776  -1.32850
     6   1.51370       0.54890  -1.51370

 Components of discrete Fourier transform

     0   1.82616
     1   1.86862
     2  -0.01750
     3   0.50200
     4  -0.59873
     5  -0.03144
     6  -2.62557
```

```
Original sequence as restored by inverse transform

        Original  Restored

    0   0.34907   0.34907
    1   0.54890   0.54890
    2   0.74776   0.74776
    3   0.94459   0.94459
    4   1.13850   1.13850
    5   1.32850   1.32850
    6   1.51370   1.51370
```