

NAG Library Routine Document

C06EBF

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
double precision 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 $j = 1, 2, \dots, n-1$), C06EBF calculates their discrete Fourier transform defined by

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

(Note the scale factor of $\frac{1}{\sqrt{n}}$ in this definition.) The transformed values $\hat{x}_k$ are purely real (see also the C06 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 .

C06EBF uses the fast Fourier transform (FFT) algorithm (see Brigham (1974)). There are some restrictions on the value of n (see Section 5).

4 References

Brigham E O (1974) *The Fast Fourier Transform* Prentice-Hall

5 Parameters

1: X(N) – **double precision** 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 $0 \leq j \leq n/2$, x_j is contained in X(j), and for $1 \leq j \leq (n-1)/2$, y_j is contained in X(n-j). (See also Section 2.1.2 in the C06 Chapter Introduction and Section 9.)

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

2: N – INTEGER *Input*

On entry: n , the number of data values. 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 . If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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

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 parameter, the recommended value is 0 . **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry 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:

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

On entry, $N \leq 1$.

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 is approximately proportional to $n \times \log n$, but also depends on the factorization of n . C06EBF is faster 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, C06EBF 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 , C06FBF (which requires an additional n elements of workspace) is considerably faster.

9 Example

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

9.1 Program Text

```

*      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 ..
      DOUBLE PRECISION 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) = 0.0D0
         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) = 0.0D0
         END IF
         IFAIL = 1
*
*      CALL C06EBF(X,N,IFAIL)
*
      IF (IFAIL.EQ.0) THEN
         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, XX(J), '      ', U(J), V(J)
80        CONTINUE
         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,*)
      WRITE (NOUT,99997) ' ** C06EBF returned with IFAIL = ',
+      IFAIL
      END IF
    ELSE
      WRITE (NOUT,*) 'Invalid value of N'
    END IF
140 CONTINUE
*
99999 FORMAT (1X,I5,F10.5,A,2F10.5)
99998 FORMAT (1X,I5,2F10.5)
99997 FORMAT (1X,A,I5)
      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
