NAG Fortran Library Routine Document

C06ECF

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

C06ECF calculates the discrete Fourier transform of a sequence of n complex data values. (No extra workspace required.)

2 Specification

```
SUBROUTINE C06ECF(X, Y, N, IFAIL)
INTEGER            N, IFAIL
real               X(N), Y(N)
```

3 Description

Given a sequence of n complex data values z_j, for $j = 0, 1, \ldots, n-1$, this routine calculates their discrete Fourier transform defined by:

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

(Note the scale factor of $\frac{1}{\sqrt{n}}$ in this definition.)

To compute the inverse discrete Fourier transform defined by:

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

this routine should be preceded and followed by calls of C06GCF to form the complex conjugates of the z_j and the $\hat{z}_k$.

The routine uses the fast Fourier transform (FFT) algorithm (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) – ***real*** array *Input/Output*

On entry: if X is declared with bounds $(0 : \mathrm{N} - 1)$ in the (sub)program from which C06ECF is called, then $\mathrm{X}(j)$ must contain x_j, the real part of z_j, for $j = 0, 1, \ldots, n-1$.

On exit: the real parts a_k of the components of the discrete Fourier transform. If X is declared with bounds $(0 : \mathrm{N} - 1)$ in the (sub)program from which C06ECF is called, then a_k is contained in $\mathrm{X}(k)$, for $k = 0, 1, \ldots, n-1$.

2: Y(N) – ***real*** array *Input/Output*

On entry: if Y is declared with bounds $(0 : \mathrm{N} - 1)$ in the (sub)program from which C06ECF is called, then $\mathrm{Y}(j)$ must contain y_j, the imaginary part of z_j, for $j = 0, 1, \ldots, n-1$.

On exit: the imaginary parts b_k of the components of the discrete Fourier transform. If Y is declared with bounds $(0 : \mathrm{N} - 1)$ in the (sub)program from which C06ECF is called, then b_k is contained in $\mathrm{Y}(k)$, for $k = 0, 1, \ldots, n - 1$.

3: 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: $\mathrm{N} > 1$.

4: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 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, for users 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 $= 3$

$\mathrm{N} \leq 1$.

IFAIL $= 4$

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 C06FCF (which requires an additional n ***real*** elements of workspace) is considerably faster.

9 Example

This program reads in a sequence of complex data values and prints their discrete Fourier transform.

It then performs an inverse transform using C06GCF and C06ECF, 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.

```
*     C06ECF 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
*     .. Local Arrays ..
      real             X(0:NMAX-1), XX(0:NMAX-1), Y(0:NMAX-1),
     +                 YY(0:NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         C06ECF, C06GCF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06ECF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=100) N
      IF (N.GT.1 .AND. N.LE.NMAX) THEN
         DO 40 J = 0, N - 1
            READ (NIN,*) X(J), Y(J)
            XX(J) = X(J)
            YY(J) = Y(J)
   40    CONTINUE
         IFAIL = 0
*
         CALL C06ECF(X,Y,N,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Components of discrete Fourier transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '          Real      Imag'
         WRITE (NOUT,*)
         DO 60 J = 0, N - 1
            WRITE (NOUT,99999) J, X(J), Y(J)
   60    CONTINUE
*
         CALL C06GCF(Y,N,IFAIL)
         CALL C06ECF(X,Y,N,IFAIL)
         CALL C06GCF(Y,N,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence as restored by inverse transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '             Original                 Restored'
         WRITE (NOUT,*)
     +     '          Real      Imag           Real      Imag'
         WRITE (NOUT,*)
         DO 80 J = 0, N - 1
            WRITE (NOUT,99999) J, XX(J), YY(J), X(J), Y(J)
   80    CONTINUE
         GO TO 20
      ELSE
         WRITE (NOUT,*) 'Invalid value of N'
      END IF
  100 STOP
*
99999 FORMAT (1X,I5,2F10.5,5X,2F10.5)
      END
```

9.2 Program Data

```
C06ECF Example Program Data
    7
  0.34907  -0.37168
  0.54890  -0.35669
  0.74776  -0.31175
  0.94459  -0.23702
  1.13850  -0.13274
  1.32850   0.00074
  1.51370   0.16298
```

9.3 Program Results

```
 C06ECF Example Program Results

 Components of discrete Fourier transform

           Real      Imag

     0   2.48361  -0.47100
     1  -0.55180   0.49684
     2  -0.36711   0.09756
     3  -0.28767  -0.05865
     4  -0.22506  -0.17477
     5  -0.14825  -0.30840
     6   0.01983  -0.56496

 Original sequence as restored by inverse transform

              Original                 Restored
           Real      Imag           Real      Imag

     0   0.34907  -0.37168        0.34907  -0.37168
     1   0.54890  -0.35669        0.54890  -0.35669
     2   0.74776  -0.31175        0.74776  -0.31175
     3   0.94459  -0.23702        0.94459  -0.23702
     4   1.13850  -0.13274        1.13850  -0.13274
     5   1.32850   0.00074        1.32850   0.00074
     6   1.51370   0.16298        1.51370   0.16298
```