NAG Fortran Library Routine Document

C06FFF

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

C06FFF computes the discrete Fourier transform of one variable in a multivariate sequence of complex data values.

2 Specification

```
SUBROUTINE C06FFF(NDIM, L, ND, N, X, Y, WORK, LWORK, IFAIL)
INTEGER           NDIM, L, ND(NDIM), N, LWORK, IFAIL
real              X(N), Y(N), WORK(LWORK)
```

3 Description

This routine computes the discrete Fourier transform of one variable (the lth say) in a multivariate sequence of complex data values $z_{j_1 j_2 \ldots j_m}$, where $j_1 = 0, 1, \ldots, n_1 - 1$, $j_2 = 0, 1, \ldots, n_2 - 1$, and so on. Thus the individual dimensions are $n_1, n_2, \ldots, n_m$, and the total number of data values is $n = n_1 \times n_2 \times \ldots \times n_m$.

The routine computes n/n_l one-dimensional transforms defined by:

$$\hat{z}_{j_1 \ldots k_l \ldots j_m} = \frac{1}{\sqrt{n_l}} \sum_{j_l=0}^{n_l-1} z_{j_1 \ldots j_l \ldots j_m} \times \exp\left(-\frac{2\pi i j_l k_l}{n_l}\right)$$

where $k_l = 0, 1, \ldots, n_l - 1$.

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

To compute the inverse discrete Fourier transforms, defined with $\exp\left(+\frac{2\pi i j_l k_l}{n_l}\right)$ in the above formula instead of $\exp\left(-\frac{2\pi i j_l k_l}{n_l}\right)$, this routine should be preceded and followed by calls of C06GCF to form the complex conjugates of the data values and the transform.

The data values must be supplied in a pair of one-dimensional arrays (real and imaginary parts separately), in accordance with the Fortran convention for storing multi-dimensional data (i.e., with the first subscript j_1 varying most rapidly).

This routine calls C06FCF to perform one-dimensional discrete Fourier transforms by the fast Fourier transform (FFT) algorithm in Brigham (1974), and hence there are some restrictions on the values of n_l (See Section 5.)

4 References

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

5 Parameters

1: NDIM – INTEGER *Input*

On entry: the number of dimensions (or variables) in the multivariate data, m.

Constraint: NDIM ≥ 1.

2: L – INTEGER *Input*

On entry: the index of the variable (or dimension) on which the discrete Fourier transform is to be performed, l.

Constraint: $1 \leq \mathrm{L} \leq \mathrm{NDIM}$.

3: ND(NDIM) – INTEGER array *Input*

On entry: ND(i) must contain n_i (the dimension of the ith variable), for $i = 1, 2, \ldots, m$. The largest prime factor of ND(l) must not exceed 19, and the total number of prime factors of ND(l), counting repetitions, must not exceed 20.

Constraint: $\mathrm{ND}(i) \geq 1$ for all i.

4: N – INTEGER *Input*

On entry: the total number of data values, n.

Constraint: $\mathrm{N} = \mathrm{ND}(1) \times \mathrm{ND}(2) \times \ldots \times \mathrm{ND}(\mathrm{NDIM})$.

5: X(N) – ***real*** array *Input/Output*

On entry: $\mathrm{X}(1 + j_1 + n_1 j_2 + n_1 n_2 j_3 + \ldots)$ must contain the real part of the complex data value $z_{j_1 j_2 \ldots j_m}$, for $0 \leq j_1 < n_1$, $0 \leq j_2 < n_2, \ldots$; i.e., the values are stored in consecutive elements of the array according to the Fortran convention for storing multi-dimensional arrays.

On exit: the real parts of the corresponding elements of the computed transform.

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

On entry: the imaginary parts of the complex data values, stored in the same way as the real parts in the array X.

On exit: the imaginary parts of the corresponding elements of the computed transform.

7: WORK(LWORK) – ***real*** array *Workspace*
8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which C06FFF is called.

Constraint: $\mathrm{LWORK} \geq 3 \times \mathrm{ND(L)}$.

9: 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

NDIM < 1.

IFAIL = 2

$N \neq ND(1) \times ND(2) \times \ldots \times ND(NDIM)$.

IFAIL = 3

$L < 1$ or $L > NDIM$.

IFAIL = $10 \times L + 1$

At least one of the prime factors of ND(L) is greater than 19.

IFAIL = $10 \times L + 2$

ND(L) has more than 20 prime factors.

IFAIL = $10 \times L + 3$

$ND(L) < 1$.

IFAIL = $10 \times L + 4$

$LWORK < 3 \times ND(L)$.

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

9 Example

This program reads in a bivariate sequence of complex data values and prints the discrete Fourier transform of the second variable. It then performs an inverse transform and prints the sequence so obtained, which may be compared with 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.

```
*     C06FFF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NDIM, NMAX, LWORK
      PARAMETER        (NDIM=2,NMAX=96,LWORK=96)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          IFAIL, L, N
*     .. Local Arrays ..
      real             WORK(LWORK), X(NMAX), Y(NMAX)
      INTEGER          ND(NDIM)
*     .. External Subroutines ..
      EXTERNAL         C06FFF, C06GCF, READXY, WRITXY
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06FFF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
```

```
   20 READ (NIN,*,END=40) ND(1), ND(2), L
      N = ND(1)*ND(2)
      IF (N.GE.1 .AND. N.LE.NMAX) THEN
         CALL READXY(NIN,X,Y,ND(1),ND(2))
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Original data'
         CALL WRITXY(NOUT,X,Y,ND(1),ND(2))
         IFAIL = 0
*
*        Compute transform
         CALL C06FFF(NDIM,L,ND,N,X,Y,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Discrete Fourier transform of variable ', L
         CALL WRITXY(NOUT,X,Y,ND(1),ND(2))
*
*        Compute inverse transform
         CALL C06GCF(Y,N,IFAIL)
         CALL C06FFF(NDIM,L,ND,N,X,Y,WORK,LWORK,IFAIL)
         CALL C06GCF(Y,N,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence as restored by inverse transform'
         CALL WRITXY(NOUT,X,Y,ND(1),ND(2))
         GO TO 20
      ELSE
         WRITE (NOUT,*) 'Invalid value of N'
      END IF
   40 STOP
*
99999 FORMAT (1X,A,I1)
      END
*
      SUBROUTINE READXY(NIN,X,Y,N1,N2)
*     Read 2-dimensional complex data
*     .. Scalar Arguments ..
      INTEGER           N1, N2, NIN
*     .. Array Arguments ..
      real              X(N1,N2), Y(N1,N2)
*     .. Local Scalars ..
      INTEGER           I, J
*     .. Executable Statements ..
      DO 20 I = 1, N1
         READ (NIN,*) (X(I,J),J=1,N2)
         READ (NIN,*) (Y(I,J),J=1,N2)
   20 CONTINUE
      RETURN
      END
*
      SUBROUTINE WRITXY(NOUT,X,Y,N1,N2)
*     Print 2-dimensional complex data
*     .. Scalar Arguments ..
      INTEGER           N1, N2, NOUT
*     .. Array Arguments ..
      real              X(N1,N2), Y(N1,N2)
*     .. Local Scalars ..
      INTEGER           I, J
*     .. Executable Statements ..
      DO 20 I = 1, N1
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Real ', (X(I,J),J=1,N2)
         WRITE (NOUT,99999) 'Imag ', (Y(I,J),J=1,N2)
   20 CONTINUE
      RETURN
*
99999 FORMAT (1X,A,7F10.3,/(6X,7F10.3))
      END
```

9.2 Program Data

```
C06FFF Example Program Data
    3    5    2
     1.000     0.999     0.987     0.936     0.802
     0.000    -0.040    -0.159    -0.352    -0.597
     0.994     0.989     0.963     0.891     0.731
    -0.111    -0.151    -0.268    -0.454    -0.682
     0.903     0.885     0.823     0.694     0.467
    -0.430    -0.466    -0.568    -0.720    -0.884
```

9.3 Program Results

```
 C06FFF Example Program Results

 Original data

 Real      1.000     0.999     0.987     0.936     0.802
 Imag      0.000    -0.040    -0.159    -0.352    -0.597

 Real      0.994     0.989     0.963     0.891     0.731
 Imag     -0.111    -0.151    -0.268    -0.454    -0.682

 Real      0.903     0.885     0.823     0.694     0.467
 Imag     -0.430    -0.466    -0.568    -0.720    -0.884

 Discrete Fourier transform of variable 2

 Real      2.113     0.288     0.126    -0.003    -0.287
 Imag     -0.513    -0.000     0.130     0.190     0.194

 Real      2.043     0.286     0.139     0.018    -0.263
 Imag     -0.745    -0.032     0.115     0.189     0.225

 Real      1.687     0.260     0.170     0.079    -0.176
 Imag     -1.372    -0.125     0.063     0.173     0.299

 Original sequence as restored by inverse transform

 Real      1.000     0.999     0.987     0.936     0.802
 Imag     -0.000    -0.040    -0.159    -0.352    -0.597

 Real      0.994     0.989     0.963     0.891     0.731
 Imag     -0.111    -0.151    -0.268    -0.454    -0.682

 Real      0.903     0.885     0.823     0.694     0.467
 Imag     -0.430    -0.466    -0.568    -0.720    -0.884
```