NAG Library Routine Document

C06FJF

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

C06FJF computes the multi-dimensional discrete Fourier transform of a multivariate sequence of complex data values.

2 Specification

```
SUBROUTINE C06FJF(NDIM, ND, N, X, Y, WORK, LWORK, IFAIL)
INTEGER            NDIM, ND(NDIM), N, LWORK, IFAIL
double precision   X(N), Y(N), WORK(LWORK)
```

3 Description

C06FJF computes the multi-dimensional discrete Fourier transform of a multi-dimensional sequence of complex data values $z_{j_1 j_2 \cdots j_m}$, 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 $n = n_1 \times n_2 \times \cdots \times n_m$.

The discrete Fourier transform is here defined (e.g., for $m = 2$) by:

$$\hat{z}_{k_1,k_2} = \frac{1}{\sqrt{n}} \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} z_{j_1 j_2} \times \exp\left(-2\pi i\left(\frac{j_1 k_1}{n_1} + \frac{j_2 k_2}{n_2}\right)\right),$$

where $k_1 = 0, 1, \ldots, n_1 - 1$, $k_2 = 0, 1, \ldots, n_2 - 1$.

The extension to higher dimensions is obvious. (Note the scale factor of $\frac{1}{\sqrt{n}}$ in this definition.)

To compute the inverse discrete Fourier transform, defined with $\exp(+2\pi i(\ldots))$ in the above formula instead of $\exp(-2\pi i(\ldots))$, 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 the n_i (see Section 5).

4 References

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

5 Parameters

1: NDIM – INTEGER *Input*

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

Constraint: NDIM ≥ 1.

2: 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 each ND(i) must not exceed 19, and the total number of prime factors of ND(i), counting repetitions, must not exceed 20.

Constraint: ND(i) ≥ 1, for $i = 1, 2, \ldots, \mathrm{NDIM}$.

3: N – INTEGER *Input*

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

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

4: X(N) – ***double precision*** 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 $z_{j_1 j_2 \cdots j_m}$, for $0 \leq j_1 \leq n_1 - 1, 0 \leq j_2 \leq n_2 - 1, \ldots$; 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.

5: Y(N) – ***double precision*** 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.

6: WORK(LWORK) – ***double precision*** array *Workspace*
7: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which C06FJF is called.

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

8: 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: $\mathrm{IFAIL} = 0$ 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 $\mathrm{IFAIL} = 0$ 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, $\mathrm{NDIM} < 1$.

$\mathrm{IFAIL} = 2$

On entry, $\mathrm{N} \neq \mathrm{ND}(1) \times \mathrm{ND}(2) \times \cdots \times \mathrm{ND}(\mathrm{NDIM})$.

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 $= 10 \times l + 3$

On entry, ND(l) < 1.

IFAIL $= 10 \times l + 4$

On entry, 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 is approximately proportional to $n \times \log n$, but also depends on the factorization of the individual dimensions ND(i). C06FJF is faster if the only prime factors are 2, 3 or 5; and fastest of all if they are powers of 2.

9 Example

This example reads in a bivariate sequence of complex data values and prints the two-dimensional Fourier transform. It then performs an inverse transform and prints the sequence so obtained, which may be compared to the original data values.

9.1 Program Text

```
*     C06FJF 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, N
*     .. Local Arrays ..
      DOUBLE PRECISION WORK(LWORK), X(NMAX), Y(NMAX)
      INTEGER          ND(NDIM)
*     .. External Subroutines ..
      EXTERNAL         C06FJF, C06GCF, READXY, WRITXY
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06FJF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=40) ND(1), ND(2)
      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 values'
         CALL WRITXY(NOUT,X,Y,ND(1),ND(2))
         IFAIL = 1
*
*        Compute transform
         CALL C06FJF(NDIM,ND,N,X,Y,WORK,LWORK,IFAIL)
*
```

```
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Components of discrete Fourier transform'
            CALL WRITXY(NOUT,X,Y,ND(1),ND(2))
*
*           Compute inverse transform
            CALL C06GCF(Y,N,IFAIL)
            CALL C06FJF(NDIM,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,*)
            WRITE (NOUT,99999) ' ** C06FJF returned with IFAIL = ',
     +        IFAIL
         END IF
      ELSE
         WRITE (NOUT,*) 'Invalid value of N'
      END IF
   40 CONTINUE
*
99999 FORMAT (1X,A,I5)
      END
*
      SUBROUTINE READXY(NIN,X,Y,N1,N2)
*     Read 2-dimensional complex data
*     .. Scalar Arguments ..
      INTEGER           N1, N2, NIN
*     .. Array Arguments ..
      DOUBLE PRECISION 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 ..
      DOUBLE PRECISION 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

```
C06FJF Example Program Data
    3    5
     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

```
 C06FJF Example Program Results

 Original data values

 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

 Components of discrete Fourier transform

 Real      3.373     0.481     0.251     0.054    -0.419
 Imag     -1.519    -0.091     0.178     0.319     0.415

 Real      0.457     0.055     0.009    -0.022    -0.076
 Imag      0.137     0.032     0.039     0.036     0.004

 Real     -0.170    -0.037    -0.042    -0.038    -0.002
 Imag      0.493     0.058     0.008    -0.025    -0.083

 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
```