NAG Toolbox for MATLAB

c06ff

1 Purpose

c06ff computes the discrete Fourier transform of one variable in a multivariate sequence of complex data values.

2 Syntax

```
[x, y, ifail] = c06ff(l, nd, x, y, 'ndim', ndim, 'n', n)
```

3 Description

c06ff 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 function 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 function should be preceded and followed by calls of c06gc 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 function calls c06fc 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

5.1 Compulsory Input Parameters

1: **l – int32 scalar**

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

Constraint: $1 \leq \mathbf{l} \leq \mathbf{ndim}$.

2: **nd(ndim) – int32 array**

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: **nd**$(i) \geq 1$, for $i = 1, 2, \ldots,$ **ndim**.

3: **x(n) – double array**

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 $0 \leq j_1 \leq n_1 - 1, 0 \leq j_2 \leq n_2 - 1, \ldots$; i.e., the values are stored in consecutive elements of the array according to the Fortran convention for storing multi-dimensional arrays.

4: **y(n) – double array**

The imaginary parts of the complex data values, stored in the same way as the real parts in the array **x**.

5.2 Optional Input Parameters

1: **ndim – int32 scalar**

Default: The dimension of the array **nd**.

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

Constraint: **ndim** ≥ 1.

2: **n – int32 scalar**

Default: The dimension of the arrays **x**, **y**. (An error is raised if these dimensions are not equal.)

n, the total number of data values.

Constraint: **n** $=$ **nd**$(1) \times$ **nd**$(2) \times \ldots \times$ **nd**(**ndim**).

5.3 Input Parameters Omitted from the MATLAB Interface

work, lwork

5.4 Output Parameters

1: **x(n) – double array**

The real parts of the corresponding elements of the computed transform.

2: **y(n) – double array**

The imaginary parts of the corresponding elements of the computed transform.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **ndim** < 1.

ifail $= 2$

On entry, $\mathbf{n} \neq \mathbf{nd}(1) \times \mathbf{nd}(2) \times \ldots \times \mathbf{nd}(\mathbf{ndim})$.

ifail $= 3$

On entry, $\mathbf{l} < 1$ or $\mathbf{l} > \mathbf{ndim}$.

ifail $= 10 \times l + 1$

At least one of the prime factors of $\mathbf{nd}(l)$ is greater than 19.

ifail $= 10 \times l + 2$

$\mathbf{nd}(l)$ has more than 20 prime factors.

ifail $= 10 \times l + 3$

On entry, $\mathbf{nd}(l) < 1$.

ifail $= 10 \times l + 4$

On entry, $\mathbf{lwork} < 3 \times \mathbf{nd}(l)$.

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 $n \times \log n_l$, but also depends on the factorization of n_l. c06ff 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

```
l = int32(2);
nd = [int32(3);
      int32(5)];
x = [1;
      0.994;
      0.903;
      0.999;
      0.989;
      0.885;
      0.987;
      0.963;
      0.823;
      0.9360000000000001;
      0.891;
      0.694;
      0.802;
      0.731;
      0.467];
y = [0;
      -0.111;
      -0.43;
      -0.04;
      -0.151;
      -0.466;
      -0.159;
      -0.268;
```

```
     -0.5679999999999999;
     -0.352;
     -0.454;
     -0.72;
     -0.597;
     -0.6820000000000001;
     -0.884];
[xOut, yOut, ifail] = c06ff(l, nd, x, y)
```

```
xOut =
    2.1126
    2.0429
    1.6869
    0.2880
    0.2862
    0.2596
    0.1257
    0.1389
    0.1695
   -0.0030
    0.0180
    0.0791
   -0.2873
   -0.2633
   -0.1759
yOut =
   -0.5134
   -0.7451
   -1.3721
   -0.0003
   -0.0322
   -0.1246
    0.1298
    0.1148
    0.0631
    0.1899
    0.1892
    0.1731
    0.1940
    0.2251
    0.2988
ifail =
           0
```