NAG Toolbox for MATLAB

c06pj

1 Purpose

c06pj computes the multi-dimensional discrete Fourier transform of a multivariate sequence of complex data values.

2 Syntax

```
[x, ifail] = c06pj(direct, nd, x, 'ndim', ndim, 'n', n)
```

3 Description

c06pj computes the multi-dimensional discrete Fourier transform of a multi-dimensional 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 discrete Fourier transform is here defined (e.g., for $m = 2$) by:

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

where $k_1 = 0, 1, \ldots, n_1 - 1$, $k_2 = 0, 1, \ldots, n_2 - 1$. The plus or minus sign in the argument of the exponential terms in the above definition determine the direction of the transform: a minus sign defines the **forward** direction and a plus sign defines the **backward** direction.

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

A call of c06pj with **direct** = 'F' followed by a call with **direct** = 'B' will restore the original data.

The data values must be supplied in a one-dimensional array using column-major storage ordering of multi-dimensional data (i.e., with the first subscript j_1 varying most rapidly).

This function calls c06pr to perform one-dimensional discrete Fourier transforms. Hence, the function uses a variant of the fast Fourier transform (FFT) algorithm (see Brigham 1974) known as the Stockham self-sorting algorithm, which is described in Temperton 1983b.

4 References

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

Temperton C 1983b Self-sorting mixed-radix fast Fourier transforms *J. Comput. Phys.* **52** 1–23

5 Parameters

5.1 Compulsory Input Parameters

1: **direct – string**

If the **F**orward transform as defined in Section 3 is to be computed, then **direct** must be set equal to 'F'.

If the **B**ackward transform is to be computed then **direct** must be set equal to 'B'.

Constraint: **direct** = 'F' or 'B'.

2: **nd**(**ndim**) – **int32 array**

The elements of **nd** must contain the dimensions of the **ndim** variables; that is, **nd**(i) must contain the dimension of the ith variable.

Constraints:

nd$(i) \geq 1$, for $i = 1, 2, \ldots,$ **ndim**;
nd(i) must have less than 31 prime factors (counting repetitions), for $i = 1, 2, \ldots,$ **ndim**.

3: **x**(**n**) – **complex array**

The complex data values. Data values are stored in **x** using column-major ordering for storing multi-dimensional arrays; that is, $z_{j_1 j_2 \cdots j_m}$ is stored in $\mathbf{x}(1 + j_1 + n_1 j_2 + n_1 n_2 j_3 + \cdots)$.

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 array **x**.

n, the total number of data values.

Constraint: **n** must equal the product of the first **ndim** elements of the array **nd**

5.3 Input Parameters Omitted from the MATLAB Interface

work, lwork

5.4 Output Parameters

1: **x**(**n**) – **complex array**

The corresponding elements of the computed transform.

2: **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, **direct** $\neq$ 'F' or 'B'.

ifail $= 3$

On entry, at least one of the first **ndim** elements of **nd** is less than 1.

ifail $= 4$

On entry, **n** does not equal the product of the first **ndim** elements of **nd**.

ifail = 5

On entry, **lwork** is too small. The minimum amount of workspace required is returned in **work**(1).

ifail = 6

On entry, **nd**(i) has more than 30 prime factors for some i.

ifail = 7

An unexpected error has occurred in an internal call. Check all (sub)program 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 the individual dimensions **nd**(i). c06pj is somewhat faster than average if their only prime factors are 2, 3 or 5; and fastest of all if they are powers of 2.

9 Example

```
direct = 'F';
nd = [int32(3);
     int32(5)];
x = [complex(1, +0);
     complex(0.994, -0.111);
     complex(0.903, -0.43);
     complex(0.999, -0.04);
     complex(0.989, -0.151);
     complex(0.885, -0.466);
     complex(0.987, -0.159);
     complex(0.963, -0.268);
     complex(0.823, -0.5679999999999999);
     complex(0.9360000000000001, -0.352);
     complex(0.891, -0.454);
     complex(0.694, -0.72);
     complex(0.802, -0.597);
     complex(0.731, -0.6820000000000001);
     complex(0.467, -0.884)];
[xOut, ifail] = c06pj(direct, nd, x)
```

```
xOut =
   3.3731 - 1.5187i
   0.4565 + 0.1368i
  -0.1705 + 0.4927i
   0.4814 - 0.0907i
   0.0549 + 0.0317i
  -0.0375 + 0.0584i
   0.2507 + 0.1776i
   0.0093 + 0.0389i
  -0.0423 + 0.0082i
   0.0543 + 0.3188i
  -0.0217 + 0.0356i
  -0.0377 - 0.0255i
  -0.4194 + 0.4145i
  -0.0759 + 0.0045i
  -0.0022 - 0.0829i
ifail =
           0
```