NAG Toolbox for MATLAB

c06pu

1 Purpose

c06pu computes the two-dimensional discrete Fourier transform of a bivariate sequence of complex data values (using complex data type).

2 Syntax

```
[x, ifail] = c06pu(direct, m, n, x)
```

3 Description

c06pu computes the two-dimensional discrete Fourier transform of a bivariate sequence of complex data values $z_{j_1 j_2}$, where $j_1 = 0, 1, \ldots, m-1$ and $j_2 = 0, 1, \ldots, n-1$.

The discrete Fourier transform is here defined by

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

where $k_1 = 0, 1, \ldots, m-1$ and $k_2 = 0, 1, \ldots, n-1$.

(Note the scale factor of $\frac{1}{\sqrt{mn}}$ in this definition.) The minus sign is taken in the argument of the exponential within the summation when the forward transform is required, and the plus sign is taken when the backward transform is required.

A call of c06pu with **direct** = 'F' followed by a call with **direct** = 'B' will restore the original data.

This function calls c06pr to perform multiple one-dimensional discrete Fourier transforms by the fast Fourier transform (FFT) algorithm in Brigham 1974.

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: **m – int32 scalar**

m, the first dimension of the transform.

Constraint: $\mathbf{m} \geq 1$.

3: **n – int32 scalar**

n, the second dimension of the transform.

Constraint: $\mathbf{n} \geq 1$.

4: **x(m × n) – complex array**

The complex data values. If **x** is regarded as a two-dimensional array of dimension $(0 : \mathbf{m}-1, 0 : \mathbf{n}-1)$, then $\mathbf{x}(j_1, j_2)$ must contain $z_{j_1 j_2}$.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

work

5.4 Output Parameters

1: **x(m × 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, $\mathbf{m} < 1$.

ifail = 2

On entry, $\mathbf{n} < 1$.

ifail = 3

On entry, **direct** $\neq$ 'F' or 'B'.

ifail = 4

On entry, **n** has more than 30 prime factors.

ifail = 5

On entry, **m** has more than 30 prime factors.

ifail = 6

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 $mn \times \log(mn)$, but also depends on the factorization of the individual dimensions m and n. c06pu 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';
m = int32(3);
n = 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] = c06pu(direct, m, n, 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
```