NAG Toolbox for MATLAB

c06pr

1 Purpose

c06pr computes the discrete Fourier transforms of m sequences, each containing n complex data values.

2 Syntax

```
[x, ifail] = c06pr(direct, m, n, x)
```

3 Description

Given m sequences of n complex data values z_j^p, for $j = 0, 1, \ldots, n-1$ and $p = 1, 2, \ldots, m$, c06pr simultaneously calculates the (**forward** or **backward**) discrete Fourier transforms of all the sequences defined by

$$\hat{z}_k^p = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} z_j^p \times \exp\left(\pm i \frac{2\pi jk}{n}\right), \qquad k = 0, 1, \ldots, n-1; \qquad p = 1, 2, \ldots, m.$$

(Note the scale factor $\frac{1}{\sqrt{n}}$ 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 c06pr with **direct** = 'F' followed by a call with **direct** = 'B' will restore the original data.

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. Special code is provided for the factors 2, 3, 4 and 5.

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 number of sequences to be transformed.

Constraint: **m** ≥ 1.

3: **n – int32 scalar**

n, the number of complex values in each sequence.

Constraint: **n** ≥ 1.

4: **x(m × n) – complex array**

The complex data must be stored in **x** as if in a two-dimensional array of dimension $(1 : \mathbf{m}, 0 : \mathbf{n} - 1)$; each of the m sequences is stored in a **row** of each array. In other words, if the elements of the pth sequence to be transformed are denoted by z_j^p, for $j = 0, 1, \ldots, n - 1$, then $\mathbf{x}(j \times \mathbf{m} + p)$ must contain z_j^p.

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**

Contains the complex transforms.

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 $= 5$

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 by c06pr is approximately proportional to $nm \log n$, but also depends on the factors of n. c06pr is fastest if the only prime factors of n are 2, 3 and 5, and is particularly slow if n is a large prime, or has large prime factors.

9 Example

```
direct = 'F';
m = int32(3);
n = int32(6);
x = [complex(0.3854, +0.5417);
     complex(0.9172, +0.9089);
     complex(0.1156, +0.6214);
     complex(0.6772, +0.2983);
     complex(0.0644, +0.3118);
     complex(0.06850000000000001, +0.8681);
     complex(0.1138, +0.1181);
     complex(0.6037, +0.3465);
     complex(0.206, +0.706);
     complex(0.6751, +0.7255);
     complex(0.643, +0.6198);
     complex(0.863, +0.8652);
     complex(0.6362, +0.8638);
     complex(0.0428, +0.2668);
     complex(0.6967, +0.919);
     complex(0.1424, +0.8723);
     complex(0.4815, +0.1614);
     complex(0.2792, +0.3355)];
[xOut, ifail] = c06pr(direct, m, n, x)
```

```
xOut =
   1.0737 + 1.3961i
   1.1237 + 1.0677i
   0.9100 + 1.7617i
  -0.5706 - 0.0409i
   0.1728 + 0.0386i
  -0.3054 + 0.0624i
   0.1733 - 0.2958i
   0.4185 + 0.7481i
   0.4079 - 0.0695i
  -0.1467 - 0.1521i
   0.1530 + 0.1752i
  -0.0785 + 0.0725i
   0.0518 + 0.4517i
   0.3686 + 0.0565i
  -0.1193 + 0.1285i
   0.3625 - 0.0321i
   0.0101 + 0.1403i
  -0.5314 - 0.4335i
ifail =
           0
```