NAG Toolbox for MATLAB

c06ek

1 Purpose

c06ek calculates the circular convolution or correlation of two real vectors of period n. No extra workspace is required.

2 Syntax

```
[x, y, ifail] = c06ek(job, x, y, 'n', n)
```

3 Description

c06ek computes:

if **job** $= 1$, the discrete **convolution** of x and y, defined by

$$z_k = \sum_{j=0}^{n-1} x_j y_{k-j} = \sum_{j=0}^{n-1} x_{k-j} y_j;$$

if **job** $= 2$, the discrete **correlation** of x and y defined by

$$w_k = \sum_{j=0}^{n-1} x_j y_{k+j}.$$

Here x and y are real vectors, assumed to be periodic, with period n, i.e., $x_j = x_{j\pm n} = x_{j\pm 2n} = \ldots$; z and w are then also periodic with period n.

Note: this usage of the terms 'convolution' and 'correlation' is taken from Brigham 1974. The term 'convolution' is sometimes used to denote both these computations.

If $\hat{x}$, $\hat{y}$, $\hat{z}$ and $\hat{w}$ are the discrete Fourier transforms of these sequences, i.e.,

$$\hat{x}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \times \exp\left(-i\frac{2\pi jk}{n}\right), \text{ etc.,}$$

then $\hat{z}_k = \sqrt{n}.\hat{x}_k\hat{y}_k$ and $\hat{w}_k = \sqrt{n}.\bar{\hat{x}}_k\hat{y}_k$ (the bar denoting complex conjugate).

This function calls the same auxiliary functions as c06ea and c06eb to compute discrete Fourier transforms, and there are some restrictions on the value of n.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **job – int32 scalar**

The computation to be performed.

job $= 1$

$$z_k = \sum_{j=0}^{n-1} x_j y_{k-j} \text{ (convolution);}$$

job $= 2$

$$w_k = \sum_{j=0}^{n-1} x_j y_{k+j} \text{ (correlation).}$$

Constraint: **job** $= 1$ or 2.

2: **x**(**n**) – **double array**

The elements of one period of the vector x. If **x** is declared with bounds $(0 : \mathbf{n} - 1)$ in the (sub)program from which c06ek is called, then $\mathbf{x}(j)$ must contain x_j, for $j = 0, 1, \ldots, n - 1$.

3: **y**(**n**) – **double array**

The elements of one period of the vector y. If **y** is declared with bounds $(0 : \mathbf{n} - 1)$ in the (sub)program from which c06ek is called, then $\mathbf{y}(j)$ must contain y_j, for $j = 0, 1, \ldots, n - 1$.

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

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

n, the number of values in one period of the vectors **x** and **y**. The largest prime factor of **n** must not exceed 19, and the total number of prime factors of **n**, counting repetitions, must not exceed 20.

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

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **x**(**n**) – **double array**

The corresponding elements of the discrete convolution or correlation.

2: **y**(**n**) – **double array**

The discrete Fourier transform of the convolution or correlation returned in the array **x**; the transform is stored in Hermitian form, exactly as described in the document for c06ea.

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 $= 1$

At least one of the prime factors of **n** is greater than 19.

ifail = 2

n has more than 20 prime factors.

ifail = 3

On entry, **n** ≤ 1.

ifail = 4

On entry, **job** $\neq 1$ or 2.

7 Accuracy

The results should be accurate to within a small multiple of the ***machine precision***.

8 Further Comments

The time taken is approximately proportional to $n \times \log n$, but also depends on the factorization of n. c06ek is faster if the only prime factors of n are 2, 3 or 5; and fastest of all if n is a power of 2.

On the other hand, c06ek is particularly slow if n has several unpaired prime factors, i.e., if the 'square-free' part of n has several factors. For such values of n, c06fk (which requires an additional n elements of workspace) is considerably faster.

9 Example

```
job = int32(1);
x = [1;
     1;
     1;
     1;
     1;
     0;
     0;
     0;
     0];
y = [0.5;
     0.5;
     0.5;
     0.5;
     0;
     0;
     0;
     0;
     0];
[xOut, yOut, ifail] = c06ek(job, x, y)
```

```
xOut =
    0.5000
    1.0000
    1.5000
    2.0000
    2.0000
    1.5000
    1.0000
    0.5000
    0.0000
yOut =
    3.3333
   -1.0585
   -0.0082
    0.0833
    0.0667
```

```
   -0.0243
   -0.1443
   -0.0465
   -0.8882
ifail =
           0
```