NAG Library Routine Document

C06EKF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

C06EKF calculates the circular convolution or correlation of two real vectors of period n. (No extra workspace is required.)

2 Specification

```
SUBROUTINE C06EKF (JOB, X, Y, N, IFAIL)

INTEGER             JOB, N, IFAIL
REAL (KIND=nag_wp) X(N), Y(N)
```

3 Description

C06EKF 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 routine calls the same auxiliary routines as C06EAF and C06EBF 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

1: JOB – INTEGER *Input*

On entry: 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) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the elements of one period of the vector x. If X is declared with bounds $(0 : \mathrm{N} - 1)$ in the subroutine from which C06EKF is called, then $\mathrm{X}(j)$ must contain x_j, for $j = 0, 1, \ldots, n - 1$.

On exit: the corresponding elements of the discrete convolution or correlation.

3: Y(N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the elements of one period of the vector y. If Y is declared with bounds $(0 : \mathrm{N} - 1)$ in the subroutine from which C06EKF is called, then $\mathrm{Y}(j)$ must contain y_j, for $j = 0, 1, \ldots, n - 1$.

On exit: the discrete Fourier transform of the convolution or correlation returned in the array X; the transform is stored in Hermitian form. If the components of the transform are:

$$\left.\begin{aligned} \hat{Z}_k &= a_k + ib_k \\ \hat{Z}_{n-k} &= a_k - ib_k \end{aligned}\right\} \quad k = 0, 1, \ldots n/2$$

where b_0 and $b_{n/2}$ when n is even then $\mathrm{X}(k+1)$ holds a_k and $\mathrm{X}(n-k+1)$ holds nonzero b_k (see Section 2.1.2 in the C06 Chapter Introduction).

4: N – INTEGER *Input*

On entry: n, the number of values in one period of the vectors X and Y.

Constraint: $\mathrm{N} > 1$.

5: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

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.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

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

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken is approximately proportional to $n \times \log(n)$, but also depends on the factorization of n. C06EKF 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, C06EKF 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, C06FKF (which requires additional real workspace) is considerably faster.

10 Example

This example reads in the elements of one period of two real vectors x and y, and prints their discrete convolution and correlation (as computed by C06EKF). In realistic computations the number of data values would be much larger.

10.1 Program Text

```
    Program c06ekfe

!     C06EKF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: c06ekf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: ieof, ifail, j, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: xa(:), xb(:), ya(:), yb(:)
!     .. Executable Statements ..
      Write (nout,*) 'C06EKF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
loop: Do
        Read (nin,*,Iostat=ieof) n
        If (ieof<0) Exit loop
        Allocate (xa(0:n-1),xb(0:n-1),ya(0:n-1),yb(0:n-1))
        Read (nin,*)(xa(j),ya(j),j=0,n-1)
        xb(0:n-1) = xa(0:n-1)
        yb(0:n-1) = ya(0:n-1)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call c06ekf(1,xa,ya,n,ifail)

        Call c06ekf(2,xb,yb,n,ifail)

        Write (nout,*)
        Write (nout,*) '        Convolution  Correlation'
        Write (nout,*)
        Write (nout,99999)(j,xa(j),xb(j),j=0,n-1)
        Deallocate (xa,xb,ya,yb)
      End Do loop

99999 Format (1X,I5,2F13.5)
    End Program c06ekfe
```

10.2 Program Data

```
C06EKF Example Program Data
    9                   : n
      1.00      0.50
      1.00      0.50
      1.00      0.50
      1.00      0.50
      1.00      0.00
      0.00      0.00
      0.00      0.00
      0.00      0.00
      0.00      0.00    : xa, ya
```

10.3 Program Results

```
 C06EKF Example Program Results

         Convolution  Correlation

     0      0.50000      2.00000
     1      1.00000      1.50000
     2      1.50000      1.00000
     3      2.00000      0.50000
     4      2.00000      0.00000
     5      1.50000      0.50000
     6      1.00000      1.00000
     7      0.50000      1.50000
     8      0.00000      2.00000
```