nag_convolution_real (c06ekc)

1. Purpose

nag_convolution_real (c06ekc) calculates the circular convolution or correlation of two real vectors of period n.

2. Specification

```
#include <nag.h>
#include <nagc06.h>

void nag_convolution_real(Nag_VectorOp operation, Integer n, double x[],
     double y[], NagError *fail)
```

3. Description

This function computes:

if **operation** = **Nag_Convolution**, 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 **operation** = **Nag_Correlation**, 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.

$$\hat{x}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \exp\left(-i2\pi \frac{jk}{n}\right)$$

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 nag_fft_real (c06eac) and nag_fft_hermitian (c06ebc) to compute discrete Fourier transforms, and there are some restrictions on the value of n.

4. Parameters

operation

Input: the computation to be performed:

if **operation** = **Nag_Convolution**, $z_k = \sum_{j=0}^{n-1} x_j y_{k-j}$;

if **operation** = **Nag_Correlation**, $w_k = \sum_{j=0}^{n-1} x_j y_{k+j}$;

Constraint: **operation** = **Nag_Convolution** or **Nag_Correlation**.

n

Input: the number of values, n, in one period of the vectors **x** and **y**.

Constraint: **n** > 1. 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.

x[n]

Input: the elements of one period of the vector x. $\mathbf{x}[j]$ must contain x_j, for $j = 0, 1, \ldots, n-1$.
Output: the corresponding elements of the discrete convolution or correlation.

y[n]

Input: the elements of one period of the vector y. $\mathbf{y}[j]$ must contain y_j, for $j = 0, 1, \ldots, n-1$.
Output: 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 nag_fft_real (c06eac).

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_C06_FACTOR_GT

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

NE_C06_TOO_MANY_FACTORS

n has more than 20 prime factors.

NE_INT_ARG_LE

On entry, **n** must not be less than or equal to 1: **n** = ⟨*value*⟩.

NE_BAD_PARAM

On entry, parameter **operation** had an illegal value.

6. Further Comments

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

The function is particularly slow if n has several unpaired prime factors, i.e., if the 'square-free' part of n has several factors.

6.1. Accuracy

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

6.2. References

Brigham E O (1974) *The Fast Fourier Transform* Prentice-Hall.

7. See Also

nag_fft_real (c06eac)

8. Example

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

8.1. Program Text

```
/* nag_convolution_real(c06ekc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagc06.h>
```

```
#define NMAX 64

main()
{
  Integer j, n;
  double xa[NMAX], xb[NMAX], ya[NMAX], yb[NMAX];

  Vprintf("c06ekc Example Program Results\n");
  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  while (scanf("%ld", &n)!=EOF)
    if (n>1 && n<=NMAX)
      {
        for (j = 0; j<n; ++j)
          {
            Vscanf("%lf%lf", &xa[j], &ya[j]);
            xb[j] = xa[j];
            yb[j] = ya[j];
          }
        c06ekc(Nag_Convolution, n, xa, ya, NAGERR_DEFAULT);
        c06ekc(Nag_Correlation, n, xb, yb, NAGERR_DEFAULT);
        Vprintf("\n          Convolution   Correlation\n\n");
        for (j = 0; j<n; ++j)
          Vprintf("%5ld %13.5f %13.5f\n", j, xa[j], xb[j]);
      }
    else
      {
        Vfprintf(stderr,"\n n = %ld which is an invalid value of n.\n", n);
        exit(EXIT_FAILURE);
      }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
c06ekc Example Program Data
    9
      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
```

8.3. Program Results

```
c06ekc 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
```