NAG Toolbox

nag_sum_fft_cosine (c06rf)

1 Purpose

nag_sum_fft_cosine (c06rf) computes the discrete Fourier cosine transforms of m sequences of real data values. The elements of each sequence and its transform are stored contiguously.

2 Syntax

```
[x, ifail] = nag_sum_fft_cosine(n, x, 'm', m)
[x, ifail] = c06rf(n, x, 'm', m)
```

3 Description

Given m sequences of $n+1$ real data values x_j^p, for $j=0,1,\ldots,n$ and $p=1,2,\ldots,m$, nag_sum_fft_cosine (c06rf) simultaneously calculates the Fourier cosine transforms of all the sequences defined by

$$\hat{x}_k^p = \sqrt{\tfrac{2}{n}}\left(\tfrac{1}{2}x_0^p + \sum_{j=1}^{n-1} x_j^p \times \cos\left(jk\frac{\pi}{n}\right) + \tfrac{1}{2}(-1)^k x_n^p\right), \quad k=0,1,\ldots,n \text{ and } p=1,2,\ldots,m.$$

(Note the scale factor $\sqrt{\frac{2}{n}}$ in this definition.)

This transform is also known as type-I DCT.

Since the Fourier cosine transform defined above is its own inverse, two consecutive calls of nag_sum_fft_cosine (c06rf) will restore the original data.

The transform calculated by this function can be used to solve Poisson's equation when the derivative of the solution is specified at both left and right boundaries (see Swarztrauber (1977)).

The function uses a variant of the fast Fourier transform (FFT) algorithm (see Brigham (1974)) known as the Stockham self-sorting algorithm, described in Temperton (1983), together with pre- and post-processing stages described in Swarztrauber (1982). Special coding is provided for the factors 2, 3, 4 and 5.

4 References

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

Swarztrauber P N (1977) The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle *SIAM Rev.* **19(3)** 490–501

Swarztrauber P N (1982) Vectorizing the FFT's *Parallel Computation* (ed G Rodrique) 51–83 Academic Press

Temperton C (1983) Fast mixed-radix real Fourier transforms *J. Comput. Phys.* **52** 340–350

5 Parameters

5.1 Compulsory Input Parameters

1: **n** – INTEGER

One less than the number of real values in each sequence, i.e., the number of values in each sequence is $n + 1$.

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

2: $\mathbf{x}(\mathbf{0} : \mathbf{n}, \mathbf{m})$ – REAL (KIND=nag_wp) array

The data values of the pth sequence to be transformed, denoted by x_j^p, for $j = 0, 1, \ldots, n$ and $p = 1, 2, \ldots, m$, must be stored in $\mathbf{x}(j, p)$.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: 1

m, the number of sequences to be transformed.

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

5.3 Output Parameters

1: $\mathbf{x}(\mathbf{0} : \mathbf{n}, \mathbf{m})$ – REAL (KIND=nag_wp) array

The $(n + 1)$ components of the pth Fourier cosine transform, denoted by $\hat{x}_k^p$, for $k = 0, 1, \ldots, n$ and $p = 1, 2, \ldots, m$, are stored in $\mathbf{x}(k, p)$, overwriting the corresponding original values.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

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

ifail $= 2$

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

ifail $= 3$

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

ifail $= -99$

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

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

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 nag_sum_fft_cosine (c06rf) is approximately proportional to $nm\log(n)$, but also depends on the factors of n. nag_sum_fft_cosine (c06rf) 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. Workspace of order $O(n)$ is internally allocated by this function.

9 Example

This example reads in sequences of real data values and prints their Fourier cosine transforms (as computed by nag_sum_fft_cosine (c06rf)). It then calls nag_sum_fft_cosine (c06rf) again and prints the results which may be compared with the original sequence.

9.1 Program Text

```
     function c06rf_example

fprintf('c06rf example results\n\n');

% Discrete cosine transform of 3 sequences of length 7
m = nag_int(3);
n = nag_int(6);
x = [ 0.3854   0.5417   0.9172;
      0.6772   0.2983   0.0644;
      0.1138   0.1181   0.6037;
      0.6751   0.7255   0.6430;
      0.6362   0.8638   0.0428;
      0.1424   0.8723   0.4815;
      0.9562   0.4936   0.2057];

[x, ifail] = c06rf(n,x);
disp('X under discrete cosine transform:');
disp(x);

% Reconstruct using same transform
[x, ifail] = c06rf(n,x);
disp('X reconstructed under second cosine transform:');
disp(x);
```

9.2 Program Results

```
     c06rf example results

X under discrete cosine transform:
    1.6833    1.9605    1.3838
   -0.0482   -0.4884    0.1588
    0.0176   -0.0655   -0.0761
    0.1368    0.4444   -0.1184
    0.3240    0.0964    0.3512
   -0.5830    0.0856    0.5759
   -0.0427   -0.2289    0.0110

X reconstructed under second cosine transform:
    0.3854    0.5417    0.9172
    0.6772    0.2983    0.0644
    0.1138    0.1181    0.6037
    0.6751    0.7255    0.6430
    0.6362    0.8638    0.0428
    0.1424    0.8723    0.4815
    0.9562    0.4936    0.2057
```