NAG Library Routine Document

C09CDF

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

C09CDF computes the inverse one-dimensional multi-level discrete wavelet transform (DWT). This routine reconstructs data from (possibly filtered or otherwise manipulated) wavelet transform coefficients calculated by C09CCF from an original set of data. The initialization routine C09AAF must be called first to set up the DWT options.

2 Specification

```
SUBROUTINE C09CDF (NWLINV, LENC, C, N, Y, ICOMM, IFAIL)
INTEGER            NWLINV, LENC, N, ICOMM(100), IFAIL
REAL (KIND=nag_wp) C(LENC), Y(N)
```

3 Description

C09CDF performs the inverse operation of C09CCF. That is, given a set of wavelet coefficients, computed up to level n_{fwd} by C09CCF using a DWT as set up by the initialization routine C09AAF, on a real data array of length n, C09CDF will reconstruct the data array y_i, for $i = 1, 2, \ldots, n$, from which the coefficients were derived. If the original input dataset is level 0, then it is possible to terminate reconstruction at a higher level by specifying fewer than the number of levels used in the call to C09CCF. This results in a partial reconstruction.

4 References

None.

5 Arguments

1: NWLINV – INTEGER *Input*

On entry: the number of levels to be used in the inverse multi-level transform. The number of levels must be less than or equal to n_{fwd}, which has the value of argument NWL as used in the computation of the wavelet coefficients using C09CCF. The data will be reconstructed to level (NWL – NWLINV), where level 0 is the original input dataset provided to C09CCF.

Constraint: $1 \leq \mathrm{NWLINV} \leq \mathrm{NWL}$, where NWL is the value used in a preceding call to C09CCF.

2: LENC – INTEGER *Input*

On entry: the dimension of the array C as declared in the (sub)program from which C09CDF is called.

Constraint: $\mathrm{LENC} \geq n_c$, where n_c is the total number of coefficients that correspond to a transform with NWLINV levels and is unchanged from the preceding call to C09CCF.

3: C(LENC) – REAL (KIND=nag_wp) array *Input*

On entry: the coefficients of a multi-level wavelet transform of the dataset.

Let $q(i)$ be the number of coefficients (of each type) at level i, for $i = n_{\text{fwd}}, n_{\text{fwd}} - 1, \ldots, 1$. Then, setting $k_1 = q(n_{\text{fwd}})$ and $k_{j+1} = k_j + q(n_{\text{fwd}} - j + 1)$, for $j = 1, 2, \ldots, n_{\text{fwd}}$, the coefficients are stored in C as follows:

$\mathrm{C}(i)$, for $i = 1, 2, \ldots, k_1$
Contains the level n_{fwd} approximation coefficients, $a_{n_{\text{fwd}}}$.

$\mathrm{C}(i)$, for $i = k_1 + 1, \ldots, k_2$
Contains the level n_{fwd} detail coefficients $d_{n_{\text{fwd}}}$.

$\mathrm{C}(i)$, for $i = k_j + 1, \ldots, k_{j+1}$
Contains the level $n_{\text{fwd}} - j + 1$ detail coefficients, for $j = 2, 3, \ldots, n_{\text{fwd}}$.

The values $q(i)$, for $i = n_{\text{fwd}}, n_{\text{fwd}} - 1, \ldots, 1$, are contained in DWTLEV which is produced as output by a preceding call to C09CCF. See C09CCF for details.

4: N – INTEGER *Input*

On entry: n, the length of the data array, y, to be reconstructed. For a full reconstruction of NWL levels, where NWL is as supplied to C09CCF, this must be the same as argument N used in the call to C09CCF. For a partial reconstruction of NWLINV $<$ NWL, this must be equal to DWTLEV(NWLINV + 2), as returned from C09CCF.

5: Y(N) – REAL (KIND=nag_wp) array *Output*

On exit: the dataset reconstructed from the multi-level wavelet transform coefficients and the transformation options supplied to the initialization routine C09AAF.

6: ICOMM(100) – INTEGER array *Communication Array*

On entry: contains details of the discrete wavelet transform and the problem dimension for the forward transform previously computed by C09CCF.

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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

On entry, NWLINV = ⟨*value*⟩.
Constraint: NWLINV $\geq$ 1.

On entry, NWLINV is larger than the number of levels computed by the preceding call to C09CCF: NWLINV = ⟨*value*⟩, expected = ⟨*value*⟩.

IFAIL = 2

On entry, LENC is set too small: LENC = ⟨*value*⟩.
Constraint: LENC ≥ ⟨*value*⟩.

IFAIL = 4

On entry, N is inconsistent with the value passed to the initialization routine: N = ⟨*value*⟩, N should be ⟨*value*⟩.

IFAIL = 6

Either the initialization routine has not been called first or array ICOMM has been corrupted.

Either the initialization routine was called with WTRANS = 'S' or array ICOMM has been corrupted.

IFAIL = −99

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = −399

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The accuracy of the wavelet transform depends only on the floating-point operations used in the convolution and downsampling and should thus be close to ***machine precision***.

8 Parallelism and Performance

C09CDF is not threaded in any implementation.

9 Further Comments

None.

10 Example

See Section 10 in C09CCF.