NAG Library Function Document

nag_imldwt (c09cdc)

1 Purpose

nag_imldwt (c09cdc) computes the inverse one-dimensional multi-level discrete wavelet transform (DWT). This function reconstructs data from (possibly filtered or otherwise manipulated) wavelet transform coefficients calculated by nag_mldwt (c09ccc) from an original set of data. The initialization function nag_wfilt (c09aac) must be called first to set up the DWT options.

2 Specification

```
#include <nag.h>
#include <nagc09.h>
```

```
void nag_imldwt (Integer nwlinv, Integer lenc, const double c[], Integer n,
     double y[], const Integer icomm[], NagError *fail)
```

3 Description

nag_imldwt (c09cdc) performs the inverse operation of nag_mldwt (c09ccc). That is, given a set of wavelet coefficients, computed up to level n_{fwd} by nag_mldwt (c09ccc) using a DWT as set up by the initialization function nag_wfilt (c09aac), on a real data array of length n, nag_imldwt (c09cdc) 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 nag_mldwt (c09ccc). 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 nag_mldwt (c09ccc). The data will be reconstructed to level (**nwl** − **nwlinv**), where level 0 is the original input dataset provided to nag_mldwt (c09ccc).

Constraint: $1 \leq$ **nwlinv** $\leq n_{\mathrm{fwd}}$, where n_{fwd} is the value used in a preceding call to nag_mldwt (c09ccc).

2: **lenc** – Integer *Input*

On entry: the dimension of the array **c**.

Constraint: **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 nag_mldwt (c09ccc).

3: **c**[**lenc**] – const double *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 $i = n_{\mathrm{fwd}}, n_{\mathrm{fwd}} - 1, \ldots, 1$. Then, setting $k_1 = q(n_{\mathrm{fwd}})$ and $k_{j+1} = k_j + q(n_{\mathrm{fwd}} - j + 1)$, for $j = 1, 2, \ldots, n_{\mathrm{fwd}}$, the coefficients are stored in **c** as follows:

c$[i - 1]$, for $i = 1, 2, \ldots, k_1$
Contains the level n_{fwd} approximation coefficients, $a_{n_{\mathrm{fwd}}}$.

c$[i - 1]$, for $i = k_1 + 1, \ldots, k_2$
Contains the level n_{fwd} detail coefficients $d_{n_{\mathrm{fwd}}}$.

c$[i - 1]$, for $i = k_j + 1, \ldots, k_{j+1}$
Contains the level $n_{\mathrm{fwd}} - j + 1$ detail coefficients, for $j = 2, 3, \ldots, n_{\mathrm{fwd}}$.

The values $q(i)$, for $i = n_{\mathrm{fwd}}, n_{\mathrm{fwd}} - 1, \ldots, 1$, are contained in **dwtlev** which is produced as output by a preceding call to nag_mldwt (c09ccc). See nag_mldwt (c09ccc) 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 nag_mldwt (c09ccc), this must be the same as argument **n** used in the call to nag_mldwt (c09ccc). For a partial reconstruction of **nwlinv** $<$ **nwl**, this must be equal to **dwtlev**[**nwlinv** $+ 1$], as returned from nag_mldwt (c09ccc).

5: **y**[**n**] – double *Output*

On exit: the dataset reconstructed from the multi-level wavelet transform coefficients and the transformation options supplied to the initialization function nag_wfilt (c09aac).

6: **icomm**[**100**] – const Integer *Communication Array*

On entry: contains details of the discrete wavelet transform and the problem dimension for the forward transform previously computed by nag_mldwt (c09ccc).

7: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_ARRAY_DIM_LEN

On entry, **lenc** is set too small: **lenc** $= \langle$*value*$\rangle$.
Constraint: **lenc** $\geq \langle$*value*$\rangle$.

NE_BAD_PARAM

On entry, argument $\langle$*value*$\rangle$ had an illegal value.

NE_INITIALIZATION

Either the initialization function has not been called first or array **icomm** has been corrupted.

Either the initialization function was called with **wtrans** = Nag_SingleLevel or array **icomm** has been corrupted.

On entry, **n** is inconsistent with the value passed to the initialization function: **n** $= \langle$*value*$\rangle$, **n** should be $\langle$*value*$\rangle$.

NE_INT_2

On entry, **nwlinv** $= \langle value \rangle$.
Constraint: **nwlinv** ≥ 1.

On entry, **nwlinv** is larger than the number of levels computed by the preceding call to nag_mldwt (c09ccc): **nwlinv** $= \langle value \rangle$, expected $= \langle value \rangle$.

NE_INTERNAL_ERROR

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.

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

Not applicable.

9 Further Comments

None.

10 Example

See Section 10 in nag_mldwt (c09ccc).