NAG Toolbox

nag_wav_1d_mxolap_multi_inv (c09dd)

1 Purpose

nag_wav_1d_mxolap_multi_inv (c09dd) computes the inverse one-dimensional multi-level maximal overlap discrete wavelet transform (MODWT). This function reconstructs data from (possibly filtered or otherwise manipulated) wavelet transform coefficients calculated by nag_wav_1d_mxolap_multi_fwd (c09dc) from an original set of data. The initialization function nag_wav_1d_init (c09aa) must be called first to set up the MODWT options.

2 Syntax

```
[y, ifail] = nag_wav_1d_mxolap_multi_inv(nwlinv, keepa, c, n, icomm, 'lenc',
lenc)
```

```
[y, ifail] = c09dd(nwlinv, keepa, c, n, icomm, 'lenc', lenc)
```

3 Description

nag_wav_1d_mxolap_multi_inv (c09dd) performs the inverse operation of nag_wav_1d_mxolap_multi_fwd (c09dc). That is, given a set of wavelet coefficients computed by nag_wav_1d_mxolap_multi_fwd (c09dc) using a MODWT as set up by the initialization function nag_wav_1d_init (c09aa) on a real array of length n, nag_wav_1d_mxolap_multi_inv (c09dd) will reconstruct the data array y_i, for $i = 1, 2, \dots, n$, from which the coefficients were derived.

4 References

Percival D B and Walden A T (2000) *Wavelet Methods for Time Series Analysis* Cambridge University Press

5 Parameters

5.1 Compulsory Input Parameters

1: **nwlinv** – INTEGER

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_wav_1d_mxolap_multi_fwd (c09dc). The data will be reconstructed to level (**nwl** − **nwlinv**), where level 0 is the original input dataset provided to nag_wav_1d_mxolap_multi_fwd (c09dc).

Constraint: $1 \leq$ **nwlinv** $\leq$ **nwl**, where **nwl** is the value used in a preceding call to nag_wav_1d_mxolap_multi_fwd (c09dc).

2: **keepa** – CHARACTER(1)

Determines whether the approximation coefficients are stored in array **c** for every level of the computed transform or else only for the final level. In both cases, the detail coefficients are stored in **c** for every level computed.

keepa = 'A'
    Retain approximation coefficients for all levels computed.

keepa = 'F'
Retain approximation coefficients for only the final level computed.

Constraint: **keepa** = 'A' or 'F'.

3: **c**(**lenc**) – REAL (KIND=nag_wp) array

The coefficients of a multi-level wavelet transform of the dataset.

The coefficients are stored in **c** as follows:

If **keepa** = 'F',

c$(1 : n_a)$
Contains the level n_l approximation coefficients;

c$(n_a + (i-1) \times n_d + 1 : n_a + i \times n_d)$
Contains the level $(n_l - i + 1)$ detail coefficients, for $i = 1, 2, \ldots, n_l$;

If **keepa** = 'A',

c$((i-1) \times n_a + 1 : i \times n_a)$
Contains the level $(n_l - i + 1)$ approximation coefficients, for $i = 1, 2, \ldots, n_l$;

c$(n_l \times n_a + (i-1) \times n_d + 1 : n_l \times n_a + i \times n_d)$
Contains the level i detail coefficients, for $i = 1, 2, \ldots, n_l$.

The values n_a and n_d denote the numbers of approximation and detail coefficients respectively, which are equal. This number is returned as output in **na** from a preceding call to nag_wav_1d_mxolap_multi_fwd (c09dc). See nag_wav_1d_mxolap_multi_fwd (c09dc) for details.

4: **n** – INTEGER

n, the length of the data array, y, to be reconstructed.

Constraint: This must be the same as the value **n** passed to the initialization function nag_wav_1d_init (c09aa).

5: **icomm**(**100**) – INTEGER array

Contains details of the discrete wavelet transform and the problem dimension for the forward transform previously computed by nag_wav_1d_mxolap_multi_fwd (c09dc).

5.2 Optional Input Parameters

1: **lenc** – INTEGER

Default: the dimension of the array **c**.

The dimension of the array **c**.

Constraints:

if **keepa** = 'F', **lenc** $\geq (n_l + 1) \times n_a$;
if **keepa** = 'A', **lenc** $\geq 2 \times n_l \times n_a$, where n_a is the number of approximation or detail coefficients at each level and is unchanged from the preceding call to nag_wav_1d_mxolap_multi_fwd (c09dc).

5.3 Output Parameters

1: **y**(**n**) – REAL (KIND=nag_wp) array

The dataset reconstructed from the multi-level wavelet transform coefficients and the transformation options supplied to the initialization function nag_wav_1d_init (c09aa).

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: **nwlinv** ≥ 1.

On entry, **nwlinv** is larger than the number of levels computed by the preceding call to nag_wav_1d_mxolap_multi_fwd (c09dc).

ifail $= 2$

On entry, **keepa** $= \langle value \rangle$ was an illegal value.

ifail $= 3$

Constraint: **lenc** $\geq \langle value \rangle$.

ifail $= 5$

On entry, **n** is inconsistent with the value passed to the initialization function.

ifail $= 7$

On entry, the initialization function nag_wav_1d_init (c09aa) has not been called first or it has not been called with **wtrans** = 'U', or the communication array **icomm** has become corrupted.

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

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 Further Comments

None.

9 Example

See Section 10 in nag_wav_1d_mxolap_multi_fwd (c09dc).

9.1 Program Text

```
function c09dd_example

fprintf('c09dd example results\n\n');

% Decompose x using maximal overlap discrete wavelet over 2 levels

n      = nag_int(64);
x      = [6.5271 6.5120 6.5016 6.5237 6.4625 6.3496 6.4025 6.4035 ...
          6.4407 6.4746 6.5095 6.6551 6.6100 6.5969 6.6083 6.6520 ...
          6.7113 6.7227 6.7196 6.7649 6.7794 6.8037 6.8308 6.7712 ...
          6.7067 6.7690 6.7068 6.7024 6.6463 6.6098 6.5900 6.5960 ...
          6.5457 6.5470 6.5797 6.5895 6.6275 6.6795 6.6598 6.6925 ...
          6.6873 6.7223 6.7205 6.6843 6.7030 6.6470 6.6008 6.6061 ...
          6.6097 6.6485 6.6394 6.6571 6.6357 6.6224 6.6073 6.6075 ...
          6.6379 6.6294 6.5906 6.6258 6.6369 6.6515 6.6826 6.7042];

wavnam = 'DB4';
mode   = 'Periodic';
wtrans = 'U';
keepa = 'All';
fprintf(' MLMODWT :: Wavelet : %10s, End mode : %10s, n = %10d\n',...
          wavnam, mode, n);
fprintf('         :: Keepa   : %10s\n\n',keepa);

% Setup for wavelet
[nwlmax, nf, nwc, icomm, ifail] = c09aa(wavnam, wtrans, mode, n);

% Compute decomposition over two levels
nwl = nag_int(2);
lenc = 2*n*nwl;
[c, na,  icomm, ifail] = c09dc(x, keepa, lenc, nwl, icomm);

fprintf(' Number of Levels                      : %10d\n',nwl);
fprintf(' Number of coefficients in each level : %10d\n\n',na);
fprintf(' Wavelet coefficients C : \n');
fprintf('%8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f\n',c)

% Reconstruct
[y, ifail] = c09dd(nwl, keepa, c, n, icomm);

fprintf('\n Reconstruction       Y : \n')
fprintf('%8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f\n',y)
```

9.2 Program Results

```
c09dd example results

 MLMODWT :: Wavelet :        DB4, End mode :   Periodic, n =         64
         :: Keepa   :        All

 Number of Levels                      :          2
 Number of coefficients in each level :         64

 Wavelet coefficients C :
  6.6448   6.6505   6.6415   6.6090   6.5631   6.5119   6.4657   6.4371
  6.4162   6.4041   6.4062   6.4235   6.4652   6.5191   6.5744   6.6170
  6.6375   6.6496   6.6575   6.6741   6.7038   6.7335   6.7633   6.7849
  6.7939   6.7970   6.7868   6.7649   6.7407   6.7102   6.6814   6.6571
  6.6269   6.5993   6.5773   6.5598   6.5574   6.5688   6.5881   6.6173
  6.6492   6.6741   6.6941   6.7052   6.7078   6.7083   6.7001   6.6842
  6.6616   6.6338   6.6146   6.6072   6.6139   6.6306   6.6428   6.6459
  6.6384   6.6252   6.6147   6.6113   6.6143   6.6189   6.6264   6.6361
  6.6719   6.5883   6.4958   6.4890   6.5103   6.4695   6.3900   6.3656
  6.4065   6.4444   6.4727   6.5273   6.6057   6.6409   6.6102   6.6001
  6.6469   6.7019   6.7288   6.7330   6.7501   6.7824   6.8064   6.8147
  6.7846   6.7332   6.7239   6.7297   6.6971   6.6508   6.6127   6.5897
  6.5818   6.5636   6.5476   6.5657   6.5980   6.6284   6.6627   6.6803
  6.6821   6.6941   6.7131   6.7182   6.7020   6.6824   6.6562   6.6140
```

```
 6.5942   6.6126   6.6378   6.6502   6.6498   6.6403   6.6233   6.6086
 6.6099   6.6260   6.6300   6.6112   6.6094   6.6358   6.6581   6.6778
 0.0107   0.0084   0.0003  -0.0065  -0.0000   0.0196   0.0191  -0.0152
-0.0369  -0.0291  -0.0131   0.0227   0.0461   0.0005  -0.0488  -0.0145
 0.0518   0.0503  -0.0038  -0.0243  -0.0087  -0.0111  -0.0316  -0.0191
 0.0323   0.0461  -0.0001  -0.0300  -0.0107   0.0164   0.0112  -0.0156
-0.0225  -0.0091   0.0090   0.0244   0.0050  -0.0281  -0.0150   0.0146
 0.0145   0.0034  -0.0019   0.0058   0.0188   0.0074  -0.0133  -0.0127
-0.0062  -0.0008   0.0077   0.0022  -0.0151  -0.0192  -0.0041   0.0091
 0.0136   0.0230   0.0203  -0.0081  -0.0274  -0.0179  -0.0013   0.0074
-0.0150   0.0126   0.0048  -0.0276  -0.0227   0.0639  -0.0184  -0.0048
-0.0303   0.0180   0.0327  -0.0343   0.0119  -0.0046   0.0167   0.0025
-0.0524   0.0369   0.0029   0.0055  -0.0070  -0.0134   0.0099   0.0088
-0.0095   0.0103  -0.0114  -0.0181   0.0269   0.0132  -0.0371   0.0250
-0.0186   0.0138   0.0022  -0.0058  -0.0112   0.0207  -0.0058  -0.0054
 0.0115  -0.0089  -0.0106   0.0180  -0.0096   0.0107  -0.0156   0.0068
 0.0074  -0.0242   0.0169   0.0075  -0.0045   0.0031  -0.0108   0.0092
-0.0115   0.0061  -0.0002   0.0078  -0.0012  -0.0168   0.0074   0.0157

Reconstruction       Y :
 6.5271   6.5120   6.5016   6.5237   6.4625   6.3496   6.4025   6.4035
 6.4407   6.4746   6.5095   6.6551   6.6100   6.5969   6.6083   6.6520
 6.7113   6.7227   6.7196   6.7649   6.7794   6.8037   6.8308   6.7712
 6.7067   6.7690   6.7068   6.7024   6.6463   6.6098   6.5900   6.5960
 6.5457   6.5470   6.5797   6.5895   6.6275   6.6795   6.6598   6.6925
 6.6873   6.7223   6.7205   6.6843   6.7030   6.6470   6.6008   6.6061
 6.6097   6.6485   6.6394   6.6571   6.6357   6.6224   6.6073   6.6075
 6.6379   6.6294   6.5906   6.6258   6.6369   6.6515   6.6826   6.7042
```