NAG Toolbox

nag_wav_1d_multi_fwd (c09cc)

1 Purpose

nag_wav_1d_multi_fwd (c09cc) computes the one-dimensional multi-level discrete wavelet transform (DWT). The initialization function nag_wav_1d_init (c09aa) must be called first to set up the DWT options.

2 Syntax

```
[c, dwtlev, icomm, ifail] = nag_wav_1d_multi_fwd(x, lenc, nwl, icomm, 'n', n)
[c, dwtlev, icomm, ifail] = c09cc(x, lenc, nwl, icomm, 'n', n)
```

3 Description

nag_wav_1d_multi_fwd (c09cc) computes the multi-level DWT of one-dimensional data. For a given wavelet and end extension method, nag_wav_1d_multi_fwd (c09cc) will compute a multi-level transform of a data array, x_i, for $i = 1, 2, \dots, n$, using a specified number, n_{fwd}, of levels. The number of levels specified, n_{fwd}, must be no more than the value $l_{\max}$ returned in **nwlmax** by the initialization function nag_wav_1d_init (c09aa) for the given problem. The transform is returned as a set of coefficients for the different levels (packed into a single array) and a representation of the multi-level structure.

The notation used here assigns level 0 to the input dataset, x, with level 1 being the first set of coefficients computed, with the detail coefficients, d_1, being stored while the approximation coefficients, a_1, are used as the input to a repeat of the wavelet transform. This process is continued until, at level n_{fwd}, both the detail coefficients, $d_{n_{\text{fwd}}}$, and the approximation coefficients, $a_{n_{\text{fwd}}}$ are retained. The output array, C, stores these sets of coefficients in reverse order, starting with $a_{n_{\text{fwd}}}$ followed by $d_{n_{\text{fwd}}}, d_{n_{\text{fwd}}-1}, \dots, d_1$.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

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

x contains the one-dimensional input dataset x_i, for $i = 1, 2, \dots, n$.

2: **lenc** – INTEGER

The dimension of the array **c**. **c** must be large enough to contain the number, n_c, of wavelet coefficients. The maximum value of n_c is returned in **nwc** by the call to the initialization function nag_wav_1d_init (c09aa) and corresponds to the DWT being continued for the maximum number of levels possible for the given data set. When the number of levels, n_{fwd}, is chosen to be less than the maximum, then n_c is correspondingly smaller and **lenc** can be reduced by noting that the number of coefficients at each level is given by $\lceil \bar{n}/2 \rceil$ for **mode** $=$ 'P' in nag_wav_1d_init (c09aa) and $\lfloor (\bar{n} + n_f - 1)/2 \rfloor$ for **mode** $=$ 'H','W','Z', where $\bar{n}$ is the number of input data at that level and n_f is the filter length provided by the call to nag_wav_1d_init (c09aa). At the final level the storage is doubled to contain the set of approximation coefficients.

Constraint: **lenc** $\geq n_c$, where n_c is the number of approximation and detail coefficients that correspond to a transform with **nwlmax** levels.

3: **nwl** – INTEGER

The number of levels, n_{fwd}, in the multi-level resolution to be performed.

Constraint: $1 \leq$ **nwl** $\leq l_{\max}$, where $l_{\max}$ is the value returned in **nwlmax** (the maximum number of levels) by the call to the initialization function nag_wav_1d_init (c09aa).

4: **icomm**(**100**) – INTEGER array

Contains details of the discrete wavelet transform and the problem dimension as setup in the call to the initialization function nag_wav_1d_init (c09aa).

5.2 Optional Input Parameters

1: **n** – INTEGER

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

The number of elements, n, in the data array x.

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

5.3 Output Parameters

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

Let $q(i)$ denote the number of coefficients (of each type) produced by the wavelet transform at level i, for $i = n_{\text{fwd}}, n_{\text{fwd}} - 1, \ldots, 1$. These values are returned in **dwtlev**. 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 as follows:

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

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

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

2: **dwtlev**(**nwl** + **1**) – INTEGER array

The number of transform coefficients at each level. **dwtlev**(1) and **dwtlev**(2) contain the number, $q(n_{\text{fwd}})$, of approximation and detail coefficients respectively, for the final level of resolution (these are equal); **dwtlev**(i) contains the number of detail coefficients, $q(n_{\text{fwd}} - i + 2)$, for the $(n_{\text{fwd}} - i + 2)$th level, for $i = 3, 4, \ldots, n_{\text{fwd}} + 1$.

3: **icomm**(**100**) – INTEGER array

Contains additional information on the computed transform.

4: **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 $= 1$

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

ifail $= 3$

lenc is too small.

ifail $= 5$

Constraint: **nwl** ≥ 1.

On entry, **nwl** is larger than the maximum number of levels returned by the initialization function.

ifail $= 7$

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

Either the initialization function 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.

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

The wavelet coefficients at each level can be extracted from the output array **c** using the information contained in **dwtlev** on exit (see the descriptions of **c** and **dwtlev** in Section 5). For example, given an input data set, x, denoising can be carried out by applying a thresholding operation to the detail coefficients at every level. The elements **c**(i), for $i = k_1 + 1, \ldots, k_{n_{\mathrm{fwd}}} + 1$, as described in Section 5, contain the detail coefficients, $\hat{d}_{ij}$, for $i = n_{\mathrm{fwd}}, n_{\mathrm{fwd}} - 1, \ldots, 1$ and $j = 1, 2, \ldots, q(i)$, where $\hat{d}_{ij} = d_{ij} + \sigma\epsilon_{ij}$ and $\sigma\epsilon_{ij}$ is the transformed noise term. If some threshold parameter α is chosen, a simple hard thresholding rule can be applied as

$$\bar{d}_{ij} = \begin{cases} 0, & \text{if } \left|\hat{d}_{ij}\right| \leq \alpha \\ \hat{d}_{ij}, & \text{if } \left|\hat{d}_{ij}\right| > \alpha, \end{cases}$$

taking $\bar{d}_{ij}$ to be an approximation to the required detail coefficient without noise, d_{ij}. The resulting coefficients can then be used as input to nag_wav_1d_multi_inv (c09cd) in order to reconstruct the denoised signal.

See the references given in the introduction to this chapter for a more complete account of wavelet denoising and other applications.

9 Example

This example performs a multi-level resolution of a dataset using the Daubechies wavelet (see **wavnam** = 'DB4' in nag_wav_1d_init (c09aa)) using zero end extensions, the number of levels of resolution, the number of coefficients in each level and the coefficients themselves are reused. The original dataset is then reconstructed using nag_wav_1d_multi_inv (c09cd).

9.1 Program Text

```
     function c09cc_example

fprintf('c09cc example results\n\n');

n = nag_int(64);
wavnam = 'DB4';
mode = 'zero';
wtrans = 'Multilevel';
x = [ 6.5271; 6.512; 6.5016; 6.5237; 6.4625;
6.3496; 6.4025; 6.4035; 6.4407; 6.4746;
6.5095; 6.6551; 6.61; 6.5969; 6.6083;
6.652; 6.7113; 6.7227; 6.7196; 6.7649;
6.7794; 6.8037; 6.8308; 6.7712; 6.7067;
6.769; 6.7068; 6.7024; 6.6463; 6.6098;
6.59; 6.596; 6.5457; 6.547; 6.5797;
6.5895; 6.6275; 6.6795; 6.6598; 6.6925;
6.6873; 6.7223; 6.7205; 6.6843; 6.703;
6.647; 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];
fprintf('\n Input Data:\n');
for i=1:8:double(n)
  fprintf('%8.4f ', x(i:i+8-1));
  fprintf('\n');
end
fprintf('\n');

% Query wavelet filter dimensions
[nwl, nf, nwc, icomm, ifail] = c09aa(wavnam, wtrans, mode, n);

if ifail == nag_int(0)
  % Perform Discrete Wavelet transform
  [c, dwtlev, icomm, ifail] = c09cc(x, nwc, nwl, icomm);

  if ifail == nag_int(0)
    fprintf(' Length of wavelet filter :             %10d\n', nf);
    fprintf(' Number of Levels :                     %10d\n\n', nwl);
    fprintf(' Number of coefficients in each level :\n     ');
    fprintf(' %8d', dwtlev);
    fprintf('\n');
    fprintf(' Total number of wavelet coefficients : %10d\n\n', nwc);
    fprintf(' Wavelet coefficients C : \n');
    for i=1:8:double(nwc)
      if i+8-1 <= numel(c)
        fprintf('%8.4f ', c(i:i+8-1));
      else
        fprintf('%8.4f ', c(i:numel(c)));
      end
      fprintf('\n');
    end
    fprintf('\n');

    % Reconstruct original data
    [y, ifail] = c09cd(nwl, c, n, icomm);

    if ifail == nag_int(0)
      fprintf('\n Reconstruction       Y : \n');
      for i=1:8:double(n)
        fprintf('%8.4f ', y(i:i+8-1));
```

```
        fprintf('\n');
      end
      fprintf('\n');
    end
  end
end
```

9.2 Program Results

```
     c09cc example results

 Input Data:
  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

 Length of wavelet filter :                      8
 Number of Levels :                              6

 Number of coefficients in each level :
             7        7        8       10       14       21       35
 Total number of wavelet coefficients :        102

 Wavelet coefficients C :
  0.0000  -0.0227  -0.3446   2.7574 -10.1970  44.8800  15.9443   0.0010
 -0.4881 -10.2673  11.3258  -1.7469   2.0785  -0.7334  -0.0054  -0.1402
 -5.8980  -1.1527   5.5613   2.1352   0.3203  -0.4004   0.0010   0.5229
  0.5055  -2.7274  -0.0911  -0.2806  -0.3669   2.9467  -0.3799  -0.1552
  0.0218   0.0922   5.4626  -2.1620   0.5196  -0.0287  -0.0199   0.0920
 -0.0134  -0.1298  -5.5168   2.3105  -0.5383  -0.0155   0.3057   0.6186
 -1.5542   0.2682   0.1566   0.0030  -0.0152  -0.0589   0.0126   0.0063
  0.0171  -0.0268   0.0077  -0.0189   0.0207   0.0104  -0.3207  -0.6062
  1.6288  -0.2414  -0.0671   3.1657  -1.1462   0.2785   0.0523  -0.0030
 -0.0270  -0.0442   0.0090   0.0171  -0.0230  -0.0015   0.0213  -0.0402
 -0.0263  -0.0099   0.0021  -0.0250   0.0210  -0.0028  -0.0298  -0.0095
  0.0034   0.0281  -0.0188  -0.0002  -0.0173  -0.0076  -0.0014   0.0184
 -0.0318   0.0048   0.0047  -3.2555   1.1710  -0.2913

 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
```