NAG Toolbox

nag_wav_1d_sngl_fwd (c09ca)

1 Purpose

nag_wav_1d_sngl_fwd (c09ca) computes the one-dimensional discrete wavelet transform (DWT) at a single level. The initialization function nag_wav_1d_init (c09aa) must be called first to set up the DWT options.

2 Syntax

```
[ca, cd, icomm, ifail] = nag_wav_1d_sngl_fwd(x, lenc, icomm, 'n', n)
[ca, cd, icomm, ifail] = c09ca(x, lenc, icomm, 'n', n)
```

3 Description

nag_wav_1d_sngl_fwd (c09ca) computes the one-dimensional DWT of a given input data array, x_i, for $i = 1, 2, \ldots, n$, at a single level. For a chosen wavelet filter pair, the output coefficients are obtained by applying convolution and downsampling by two to the input, x. The approximation (or smooth) coefficients, C_a, are produced by the low pass filter and the detail coefficients, C_d, by the high pass filter. To reduce distortion effects at the ends of the data array, several end extension methods are commonly used. Those provided are: periodic or circular convolution end extension, half-point symmetric end extension, whole-point symmetric end extension or zero end extension. The number n_c, of coefficients C_a or C_d is returned by the initialization function nag_wav_1d_init (c09aa).

4 References

Daubechies I (1992) *Ten Lectures on Wavelets* SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

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

x contains the input dataset x_i, for $i = 1, 2, \ldots, n$.

2: **lenc** – INTEGER

The dimension of the arrays **ca** and **cd**. this must be at least the number, n_c, of approximation coefficients, C_a, and detail coefficients, C_d, of the discrete wavelet transform as returned in **nwc** by the call to the initialization function nag_wav_1d_init (c09aa).

Constraint: **lenc** $\geq n_c$, where n_c is the value returned in **nwc** by the call to the initialization function nag_wav_1d_init (c09aa).

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

ca(i) contains the ith approximation coefficient, $C_a(i)$, for $i = 1, 2, \ldots, n_c$.

2: **cd**(**lenc**) – REAL (KIND=nag_wp) array

cd(i) contains the ith detail coefficient, $C_d(i)$, for $i = 1, 2, \ldots, n_c$.

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 $= 3$

On entry, array dimension **lenc** not large enough.

ifail $= 6$

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

Either the initialization function was called with **wtrans** = 'M' 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

None.

9 Example

This example computes the one-dimensional discrete wavelet decomposition for 8 values using the Daubechies wavelet, **wavnam** = 'DB4', with zero end extension.

9.1 Program Text

```
     function c09ca_example

fprintf('c09ca example results\n\n');

n = nag_int(8);
wavnam = 'DB4';
mode = 'zero';
wtrans = 'Single Level';
x = [1; 3; 5; 7; 6; 4; 5; 2];
fprintf('\n Input Data:\n');
fprintf('%8.4f ', x);
fprintf('\n');

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

if ifail == nag_int(0)
  % Compute the transform
  [ca, cd, icomm, ifail] = c09ca(x, nwc, icomm);

  if ifail == nag_int(0)
    fprintf(' Approximation coefficients CA :\n');
    fprintf('%8.4f ', ca);
    fprintf('\n');
    fprintf(' Detail coefficients        CD :\n');
    fprintf('%8.4f ', cd);
    fprintf('\n');

    % Reconstruct original data
    [y, ifail] = c09cb(ca, cd, n, icomm);

    if ifail == nag_int(0)
      fprintf(' Reconstruction       Y : \n');
      fprintf('%8.4f ', y);
      fprintf('\n');
    end
  end
end
```

9.2 Program Results

```
     c09ca example results

 Input Data:
  1.0000   3.0000   5.0000   7.0000   6.0000   4.0000   5.0000   2.0000
 Approximation coefficients CA :
  0.0011  -0.0043  -0.0174   4.4778   8.9557   7.3401   2.5816
 Detail coefficients        CD :
  0.0237   0.0410  -0.5966   1.7763  -0.7517   0.3332  -0.1188
 Reconstruction       Y :
  1.0000   3.0000   5.0000   7.0000   6.0000   4.0000   5.0000   2.0000
```