NAG Toolbox

nag_wav_2d_multi_fwd (c09ec)

1 Purpose

nag_wav_2d_multi_fwd (c09ec) computes the two-dimensional multi-level discrete wavelet transform (DWT). The initialization function nag_wav_2d_init (c09ab) must be called first to set up the DWT options.

2 Syntax

```
[c, dwtlvm, dwtlvn, icomm, ifail] = nag_wav_2d_multi_fwd(a, lenc, nwl, icomm,
'm', m, 'n', n)
```

```
[c, dwtlvm, dwtlvn, icomm, ifail] = c09ec(a, lenc, nwl, icomm, 'm', m, 'n', n)
```

3 Description

nag_wav_2d_multi_fwd (c09ec) computes the multi-level DWT of two-dimensional data. For a given wavelet and end extension method, nag_wav_2d_multi_fwd (c09ec) will compute a multi-level transform of a matrix A, 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_2d_init (c09ab) 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 matrix, A. Level 1 consists of the first set of coefficients computed: the vertical (v_1), horizontal (h_1) and diagonal (d_1) coefficients are stored at this level while the approximation (a_1) coefficients are used as the input to a repeat of the wavelet transform at the next level. This process is continued until, at level n_{fwd}, all four types of coefficients are stored. The output array, C, stores these sets of coefficients in reverse order, starting with $a_{n_{\text{fwd}}}$ followed by $v_{n_{\text{fwd}}}, h_{n_{\text{fwd}}}, d_{n_{\text{fwd}}}, v_{n_{\text{fwd}}-1}, h_{n_{\text{fwd}}-1}, d_{n_{\text{fwd}}-1}, \dots, v_1, h_1, d_1$.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **a**$(lda, \mathbf{n})$ – REAL (KIND=nag_wp) array

lda, the first dimension of the array, must satisfy the constraint $lda \geq \mathbf{m}$.

The m by n data matrix A.

2: **lenc** – INTEGER

The dimension of the array **c**. **c** must be large enough to contain, n_{ct}, wavelet coefficients. The maximum value of n_{ct} is returned in **nwct** by the call to the initialization function nag_wav_2d_init (c09ab) 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, $l_{\max}$, then n_{ct} is correspondingly smaller and **lenc** can be reduced by noting that the vertical, horizontal and diagonal coefficients are stored at every level and that in addition the approximation coefficients are stored for the final level only. The number of coefficients stored at each level is given by $3 \times \lceil \bar{m}/2 \rceil \times \lceil \bar{n}/2 \rceil$ for **mode** $=$ 'P' in nag_wav_2d_init (c09ab) and $3 \times \lfloor (\bar{m} + n_f - 1)/2 \rfloor \times \lfloor (\bar{n} + n_f - 1)/2 \rfloor$ for **mode** $=$ 'H','W','Z', where the input data is of dimension $\bar{m} \times \bar{n}$ at that level and n_f is the filter length **nf** provided by the call to nag_wav_2d_init (c09ab). At the final level the storage is 4/3 times this value to contain the set of approximation coefficients.

Constraint: **lenc** $\geq n_{\mathrm{ct}}$, where n_{ct} is the total number of coefficients that correspond to a transform with **nwl** 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_2d_init (c09ab).

4: **icomm**(**180**) – INTEGER array

Contains details of the discrete wavelet transform and the problem dimension as setup in the call to the initialization function nag_wav_2d_init (c09ab).

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **a**.

Number of rows, m, of data matrix A.

Constraint: this must be the same as the value **m** passed to the initialization function nag_wav_2d_init (c09ab).

2: **n** – INTEGER

Default: the second dimension of the array **a**.

Number of columns, n, of data matrix A.

Constraint: this must be the same as the value **n** passed to the initialization function nag_wav_2d_init (c09ab).

5.3 Output Parameters

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

The coefficients of the discrete wavelet transform. If you need to access or modify the approximation coefficients or any specific set of detail coefficients then the use of nag_wav_2d_coeff_ext (c09ey) or nag_wav_2d_coeff_ins (c09ez) is recommended. For completeness the following description provides details of precisely how the coefficient are stored in **c** but this information should only be required in rare cases.

Let $q(i)$ denote the number of coefficients (of each type) at level i, for $i = 1, 2, \ldots, n_{\mathrm{fwd}}$, such that $q(i) =$ **dwtlvm**$(n_{\mathrm{fwd}} - i + 1) \times$ **dwtlvn**$(n_{\mathrm{fwd}} - i + 1)$. Then, letting $k_1 = q(n_{\mathrm{fwd}})$ and $k_{j+1} = k_j + q(n_{\mathrm{fwd}} - \lceil j/3 \rceil + 1)$, for $j = 1, 2, \ldots, 3n_{\mathrm{fwd}}$, the coefficients are stored in **c** as follows:

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

c(i), for $i = k_j + 1, \ldots, k_{j+1}$
: Contains the level $n_{\mathrm{fwd}} - \lceil j/3 \rceil + 1$ vertical, horizontal and diagonal coefficients. These are:

vertical coefficients if $j \bmod 3 = 1$;

horizontal coefficients if $j \bmod 3 = 2$;

diagonal coefficients if $j \bmod 3 = 0$,

for $j = 1, \ldots, 3n_{\mathrm{fwd}}$.

2: **dwtlvm**(**nwl**) – INTEGER array

The number of coefficients in the first dimension for each coefficient type at each level. **dwtlvm**(i) contains the number of coefficients in the first dimension (for each coefficient type computed) at the $(n_{\text{fwd}} - i + 1)$th level of resolution, for $i = 1, 2, \ldots, n_{\text{fwd}}$. Thus for the first $n_{\text{fwd}} - 1$ levels of resolution, **dwtlvm**$(n_{\text{fwd}} - i + 1)$ is the size of the first dimension of the matrices of vertical, horizontal and diagonal coefficients computed at this level; for the final level of resolution, **dwtlvm**(1) is the size of the first dimension of the matrices of approximation, vertical, horizontal and diagonal coefficients computed.

3: **dwtlvn**(**nwl**) – INTEGER array

The number of coefficients in the second dimension for each coefficient type at each level. **dwtlvn**(i) contains the number of coefficients in the second dimension (for each coefficient type computed) at the $(n_{\text{fwd}} - i + 1)$th level of resolution, for $i = 1, 2, \ldots, n_{\text{fwd}}$. Thus for the first $n_{\text{fwd}} - 1$ levels of resolution, **dwtlvn**$(n_{\text{fwd}} - i + 1)$ is the size of the second dimension of the matrices of vertical, horizontal and diagonal coefficients computed at this level; for the final level of resolution, **dwtlvn**(1) is the size of the second dimension of the matrices of approximation, vertical, horizontal and diagonal coefficients computed.

4: **icomm**(**180**) – INTEGER array

Contains additional information on the computed transform.

5: **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: **m** $= \langle value \rangle$, the value of **m** on initialization (see nag_wav_2d_init (c09ab)).

Constraint: **n** $= \langle value \rangle$, the value of **n** on initialization (see nag_wav_2d_init (c09ab)).

ifail $= 2$

Constraint: $lda \geq$ **m**.

ifail $= 3$

lenc is too small, the total number of coefficents to be generated.

ifail $= 5$

Constraint: **nwl** $\leq$ **nwlmax** in nag_wav_2d_init (c09ab).

Constraint: **nwl** ≥ 1.

ifail $= 7$

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

Either the initialization function was called with **wtrans** = 'S' or **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 **dwtlvm** and **dwtlvn** on exit (see the descriptions of **c**, **dwtlvm** and **dwtlvn** in Section 5). For example, given an input data set, A, denoising can be carried out by applying a thresholding operation to the detail (vertical, horizontal and diagonal) coefficients at every level. The elements $\mathbf{c}(k_1+1)$ to $\mathbf{c}(k_{n_{\mathrm{fwd}}+1})$, as described in Section 5, contain the detail coefficients, $\hat{c}_{ij}$, for $i = n_{\mathrm{fwd}}, n_{\mathrm{fwd}}-1, \ldots, 1$ and $j = 1, 2, \ldots, 3q(i)$, where $q(i)$ is the number of each type of coefficient at level i and $\hat{c}_{ij} = c_{ij} + \sigma\epsilon_{ij}$ 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

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

taking $\bar{c}_{ij}$ to be an approximation to the required detail coefficient without noise, c_{ij}. The resulting coefficients can then be used as input to nag_wav_2d_multi_inv (c09ed) in order to reconstruct the denoised signal. See Section 10 in nag_wav_2d_coeff_ins (c09ez) for a simple example of denoising.

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 transform of a dataset using the Daubechies wavelet (see **wavnam** = 'DB2' in nag_wav_2d_init (c09ab)) using half-point symmetric end extensions, the maximum possible number of levels of resolution, where the number of coefficients in each level and the coefficients themselves are not changed. The original dataset is then reconstructed using nag_wav_2d_multi_inv (c09ed).

9.1 Program Text

```
     function c09ec_example

fprintf('c09ec example results\n\n');

m = nag_int(7);
n = nag_int(8);
a = [3, 7, 9, 1, 9, 9, 1, 0;
     9, 9, 3, 3, 4, 1, 2, 4;
     7, 8, 1, 3, 8, 9, 3, 3;
     1, 1, 1, 1, 2, 8, 4, 0;
     1, 2, 4, 6, 5, 6, 5, 4;
     2, 2, 5, 7, 3, 6, 6, 8;
     7, 9, 3, 1, 3, 4, 7, 2];

fprintf('\nInput data a:\n');
disp(a);

wavnam = 'DB2';
mode = 'Half';
wtrans = 'Multilevel';
[nwl, nf, nwct, nwcn, icomm, ifail] = ...
  c09ab(...
        wavnam, wtrans, mode, m, n);
```

```
lenc = nwct;
% Perform Discrete Wavelet transform
[c, dwtlvm, dwtlvn, icomm, ifail] = c09ec(a, lenc, nwl, icomm);

fprintf('\nLength of wavelet filter : %d\n', nf);
fprintf('Number of Levels :         %d\n', nwl);
fprintf('Number of coefficients in first dimension for each level :\n');
disp(transpose(dwtlvm(1:nwl)));
fprintf('Number of coefficients in second dimension for each level :\n');
disp(transpose(dwtlvn(1:nwl)));

fprintf('\nTotal number of wavelet coefficients : %d\n', nwct);
fprintf('\nWavelet coefficients c :\n');

for ilevel = 1:nwl
  fprintf('------------------------------------------------------\n');
  d1 = dwtlvm(ilevel);
  d2 = dwtlvn(ilevel);
  level = nwl-ilevel+1;
  fprintf('Level %d output is %d by %d\n', level, d1, d2);
  fprintf('------------------------------------------------------\n');

  for itype_coeffs = nag_int(1:4)
    switch itype_coeffs
      case {1}
        if (ilevel == 1)
          fprintf('Approximation coefficients:\n');
        end
      case {2}
        fprintf('Vertical coefficients:\n');
      case {3}
        fprintf('Horizontal coefficients:\n');
      case {4}
        fprintf('Diagonal coefficients:\n');
    end
    if (itype_coeffs>1 || ilevel==1)
      % Use 2D extraction routine c09eyf
      [d, icomm, ifail] = c09ey(...
                              level, itype_coeffs-1, c, icomm);
      disp(d(1:d1,1:d2));
    end
  end
  fprintf('\n');
end

% Reconstruct original data
[b, ifail] = c09ed(nwl, c, m, n, icomm);
fprintf('Reconstruction       b:\n');
disp(b);
```

9.2 Program Results

```
     c09ec example results

Input data a:
     3     7     9     1     9     9     1     0
     9     9     3     3     4     1     2     4
     7     8     1     3     8     9     3     3
     1     1     1     1     2     8     4     0
     1     2     4     6     5     6     5     4
     2     2     5     7     3     6     6     8
     7     9     3     1     3     4     7     2

Length of wavelet filter : 4
Number of Levels :         2
Number of coefficients in first dimension for each level :
                    4                    5

Number of coefficients in second dimension for each level :
                    4                    5
```

```
Total number of wavelet coefficients : 139

Wavelet coefficients c :
-------------------------------------------------------
Level 2 output is 4 by 4
-------------------------------------------------------
Approximation coefficients:
   24.9724   25.6017   20.8900    7.9280
   27.6100   27.0955   18.7941    8.2804
   11.2663   11.0273   19.6410   18.6651
   27.6050   26.6443   14.5913   18.0835

Vertical coefficients:
   -2.5552   -6.1078   -4.0629    8.2136
   -1.6061   -7.2355   -3.3633    7.6075
   -0.2225   -1.6283   -0.5301    3.7415
   -0.9052   -6.5810    0.8023    1.8591

Horizontal coefficients:
   -3.8069   -3.0730    2.1121   -1.8525
   -2.7548   -4.5949   -0.8321   -4.8155
    4.8398    4.5104   -1.5308   -0.6456
   -6.4332   -4.5381    2.4753    6.8224

Diagonal coefficients:
   -0.8978   -0.2326   -1.2515    2.6346
    0.5708   -4.9783   -1.5309    6.4569
   -0.1854   -1.8430    0.2426   -0.0754
    0.0345    7.1864    1.5938   -5.9745

-------------------------------------------------------
Level 1 output is 5 by 5
-------------------------------------------------------
Vertical coefficients:
   -2.5981    4.6471    2.5392   -2.8415   -0.2165
   -1.3203   -0.0592    3.0490   -2.5837    1.0458
   -0.4330   -1.6405   -1.1752    0.2533   -2.3448
   -0.4118   -0.0682   -2.4608   -0.0167    0.4387
   -1.5368   -1.1450   -0.5547    4.5936   -3.6863

Horizontal coefficients:
   -4.3301   -1.8170    0.8023    5.7566   -2.8146
    4.3089    3.6908    0.8349    3.4653    1.7108
   -1.5311   -1.0736    1.5257    0.0212   -0.9608
    2.8873    3.1148   -1.9118   -0.4007   -1.5302
   -2.2377   -2.7611    2.4453   -0.3705    4.3448

Diagonal coefficients:
   -1.5000    4.4151   -0.0057   -0.8236   -1.1250
   -0.1953   -2.9530    1.8840   -1.7635    0.9877
   -0.4330    0.2745    1.1450    0.4632   -0.5547
   -0.3538   -0.3215    0.6462    1.3705   -1.2778
    0.7288    0.4587   -1.8873   -1.8828    2.4028

Reconstruction       b:
    3.0000    7.0000    9.0000    1.0000    9.0000    9.0000    1.0000    0.0000
    9.0000    9.0000    3.0000    3.0000    4.0000    1.0000    2.0000    4.0000
    7.0000    8.0000    1.0000    3.0000    8.0000    9.0000    3.0000    3.0000
    1.0000    1.0000    1.0000    1.0000    2.0000    8.0000    4.0000    0.0000
    1.0000    2.0000    4.0000    6.0000    5.0000    6.0000    5.0000    4.0000
    2.0000    2.0000    5.0000    7.0000    3.0000    6.0000    6.0000    8.0000
    7.0000    9.0000    3.0000    1.0000    3.0000    4.0000    7.0000    2.0000
```