NAG Toolbox

nag_wav_2d_multi_inv (c09ed)

1 Purpose

nag_wav_2d_multi_inv (c09ed) computes the inverse two-dimensional multi-level discrete wavelet transform (DWT). This function reconstructs data from (possibly filtered or otherwise manipulated) wavelet transform coefficients calculated by nag_wav_2d_multi_fwd (c09ec) from an original input matrix. The initialization function nag_wav_2d_init (c09ab) must be called first to set up the DWT options.

2 Syntax

```
[b, ifail] = nag_wav_2d_multi_inv(nwlinv, c, m, n, icomm, 'lenc', lenc)
[b, ifail] = c09ed(nwlinv, c, m, n, icomm, 'lenc', lenc)
```

3 Description

nag_wav_2d_multi_inv (c09ed) performs the inverse operation of nag_wav_2d_multi_fwd (c09ec). That is, given a set of wavelet coefficients, computed up to level n_{fwd} by nag_wav_2d_multi_fwd (c09ec) using a DWT as set up by the initialization function nag_wav_2d_init (c09ab), on a real matrix, A, nag_wav_2d_multi_inv (c09ed) will reconstruct A. The reconstructed matrix is referred to as B in the following since it will not be identical to A when the DWT coefficients have been filtered or otherwise manipulated prior to reconstruction. If the original input matrix 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_wav_2d_multi_fwd (c09ec). This results in a partial reconstruction.

4 References

None.

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_2d_multi_fwd (c09ec). The data will be reconstructed to level (**nwl** – **nwlinv**), where level 0 is the original input dataset provided to nag_wav_2d_multi_fwd (c09ec).

Constraint: $1 \leq$ **nwlinv** $\leq$ **nwl**, where **nwl** is the value used in a preceding call to nag_wav_2d_multi_fwd (c09ec).

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

The coefficients of a multi-level wavelet transform of the original matrix, A, which may have been filtered or otherwise manipulated.

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 $k_{j+1} = k_j + q(n_{\mathrm{fwd}} - \lceil j/3 \rceil + 1)$, 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 mod $3 = 1$;

horizontal coefficients if j mod $3 = 2$;

diagonal coefficients if j mod $3 = 0$,

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

Note that the coefficients in **c** may be extracted according to level and type into two-dimensional arrays using nag_wav_2d_coeff_ext (c09ey), and inserted using nag_wav_2d_coeff_ins (c09ez).

3: **m** – INTEGER

The number of elements, m, in the first dimension of the reconstructed matrix B. For a full reconstruction of **nwl** levels, where **nwl** is as supplied to nag_wav_2d_multi_fwd (c09ec), this must be the same as argument **m** used in the call to nag_wav_2d_multi_fwd (c09ec). For a partial reconstruction of **nwlinv** $<$ **nwl** levels, this must be equal to **dwtlvm**(**nwlinv** $+ 1$), as returned from nag_wav_2d_multi_fwd (c09ec).

4: **n** – INTEGER

The number of elements, n, in the second dimension of the reconstructed matrix B. For a full reconstruction of **nwl** levels, where **nwl** is as supplied to nag_wav_3d_multi_fwd (c09fc), this must be the same as argument **n** used in the call to nag_wav_2d_multi_fwd (c09ec). For a partial reconstruction of **nwlinv** $<$ **nwl**, this must be equal to **dwtlvn**(**nwlinv** $+ 1$), as returned from nag_wav_2d_multi_fwd (c09ec).

5: **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: **lenc** – INTEGER

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

The dimension of the array **c**.

Constraint: **lenc** $\geq n_{\mathrm{ct}}$, where n_{ct} is the total number of coefficients that correspond to a transform with **nwlinv** levels and is unchanged from the preceding call to nag_wav_2d_multi_fwd (c09ec).

5.3 Output Parameters

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

The m by n reconstructed matrix, B, based on the input multi-level wavelet transform coefficients and the transform options supplied to the initialization function nag_wav_2d_init (c09ab).

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** $\leq n_{\text{fwd}}$.

Constraint: **nwlinv** ≥ 1.

ifail $= 2$

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

ifail $= 3$

lenc is too small, the total number of coefficients generated by the preceding call to nag_wav_2d_multi_fwd (c09ec).

ifail $= 4$

m is too small, the number of coefficients in the first dimension at the required level of reconstruction.

n is too small, the number of coefficients in the second dimension at the required level of reconstruction.

ifail $= 6$

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

None.

9 Example

See Section 10 in nag_wav_2d_multi_fwd (c09ec).

9.1 Program Text

```
     function c09ed_example

fprintf('c09ed example results\n\n');

m = nag_int(7);
n = nag_int(8);
wavnam = 'DB2';
mode = 'Half';
wtrans = 'Multilevel';
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);
[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:double(nwl))));
fprintf('Number of coefficients in second dimension for each level :\n');
disp(transpose(dwtlvn(1:double(nwl))));

fprintf('\nTotal number of wavelet coefficients : %d\n', nwct);
fprintf('\nWavelet coefficients c :\n');
jstart = 1;
for ilevel = 1:double(nwl)
  fprintf('-------------------------------------------------------\n');
  fprintf('Level %d output is %d by %d\n', ...
     nwl-ilevel+1, dwtlvm(ilevel), dwtlvn(ilevel));
  fprintf('-------------------------------------------------------\n');

  iskip = double(dwtlvm(ilevel));
  i2 = iskip*double(dwtlvn(ilevel)) - 1;

  for itype_coeffs = 1:4
    switch itype_coeffs
      case {1}
        if (ilevel == nwl)
          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)
      for i1 = jstart:jstart+iskip-1
        fprintf('%8.4f',c(i1:iskip:i1+i2));
        fprintf('\n');
      end
      jstart = jstart + i2 + 1;
    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

```
     c09ed 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
-------------------------------------------------------
 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
-------------------------------------------------------
Approximation coefficients:
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
```