

## NAG Toolbox

### nag\_wav\_2d\_init (c09ab)

#### 1 Purpose

`nag_wav_2d_init` (c09ab) returns the details of the chosen two-dimensional discrete wavelet filter. For a chosen mother wavelet, discrete wavelet transform type (single-level or multi-level DWT) and end extension method, this function returns the maximum number of levels of resolution (appropriate to a multi-level transform), the filter length, the total number of approximation, horizontal, vertical and diagonal coefficients and the number of coefficients in the second dimension for the single-level case. This function must be called before any of the two-dimensional transform functions in this chapter.

#### 2 Syntax

```
[nwlmax, nf, nwct, nwc, icomm, ifail] = nag_wav_2d_init(wavnam, wtrans, mode,
m, n)
[nwlmax, nf, nwct, nwc, icomm, ifail] = c09ab(wavnam, wtrans, mode, m, n)
```

#### 3 Description

Two-dimensional discrete wavelet transforms (DWT) are characterised by the mother wavelet, the end extension method and whether multiresolution analysis is to be performed. For the selected combination of choices for these three characteristics, and for given dimensions ( $m \times n$ ) of data matrix  $A$ , `nag_wav_2d_init` (c09ab) returns the dimension details for the transform determined by this combination. The dimension details are:  $l_{\max}$ , the maximum number of levels of resolution that would be computed were a multi-level DWT applied;  $n_f$ , the filter length;  $n_{ct}$  the total number of approximation, horizontal, vertical and diagonal coefficients (over all levels in the multi-level DWT case); and  $n_{cn}$ , the number of coefficients in the second dimension for a single-level DWT. These values are also stored in the communication array **icomm**, as are the input choices, so that they may be conveniently communicated to the two-dimensional transform functions in this chapter.

#### 4 References

None.

#### 5 Parameters

##### 5.1 Compulsory Input Parameters

1: **wavnam** – CHARACTER(\*)

The name of the mother wavelet. See the C09 Chapter Introduction for details.

**wavnam** = 'HAAR'  
Haar wavelet.

**wavnam** = 'DB $n$ ', where  $n = 2, 3, \dots, 10$

Daubechies wavelet with  $n$  vanishing moments ( $2n$  coefficients). For example, **wavnam** = 'DB4' is the name for the Daubechies wavelet with 4 vanishing moments (8 coefficients).

**wavnam** = 'BIORx.y', where **x.y** can be one of 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5 or 3.7  
Biorthogonal wavelet of order **x.y**. For example **wavnam** = 'BIOR3.1' is the name for the biorthogonal wavelet of order 3.1.

*Constraint:* **wavnam** = 'HAAR', 'DB2', 'DB3', 'DB4', 'DB5', 'DB6', 'DB7', 'DB8', 'DB9', 'DB10', 'BIOR1.1', 'BIOR1.3', 'BIOR1.5', 'BIOR2.2', 'BIOR2.4', 'BIOR2.6', 'BIOR2.8', 'BIOR3.1', 'BIOR3.3', 'BIOR3.5' or 'BIOR3.7'.

2: **wtrans** – CHARACTER(1)

The type of discrete wavelet transform that is to be applied.

**wtrans** = 'S'

Single-level decomposition or reconstruction by discrete wavelet transform.

**wtrans** = 'M'

Multiresolution, by a multi-level DWT or its inverse.

*Constraint:* **wtrans** = 'S' or 'M'.

3: **mode** – CHARACTER(1)

The end extension method.

**mode** = 'P'

Periodic end extension.

**mode** = 'H'

Half-point symmetric end extension.

**mode** = 'W'

Whole-point symmetric end extension.

**mode** = 'Z'

Zero end extension.

*Constraint:* **mode** = 'P', 'H', 'W' or 'Z'.

4: **m** – INTEGER

The number of elements,  $m$ , in the first dimension (number of rows of data matrix  $A$ ) of the input data.

*Constraint:*  $m \geq 2$ .

5: **n** – INTEGER

The number of elements,  $n$ , in the second dimension (number of columns of data matrix  $A$ ) of the input data.

*Constraint:*  $n \geq 2$ .

## 5.2 Optional Input Parameters

None.

## 5.3 Output Parameters

1: **nwlmax** – INTEGER

The maximum number of levels of resolution,  $l_{\max}$ , that can be computed if a multi-level discrete wavelet transform is applied (**wtrans** = 'M'). It is such that  $2^{l_{\max}} \leq \min(m, n) < 2^{l_{\max}+1}$ , for  $l_{\max}$  an integer.

If **wtrans** = 'S', **nwlmax** is not set.

2: **nf** – INTEGER

The filter length,  $n_f$ , for the supplied mother wavelet. This is used to determine the number of coefficients to be generated by the chosen transform.

3: **nwct** – INTEGER

The total number of wavelet coefficients,  $n_{ct}$ , that will be generated. When **wtrans** = 'S' the number of rows required in each of the output coefficient matrices can be calculated as  $n_{cm} = n_{ct}/(4n_{cn})$ . When **wtrans** = 'M' the length of the array used to store all of the coefficient matrices must be at least  $n_{ct}$ .

4: **nwcn** – INTEGER

For a single-level transform (**wtrans** = 'S'), the number of coefficients that would be generated in the second dimension,  $n_{cn}$ , for each coefficient type. For a multi-level transform (**wtrans** = 'M') this is set to 1.

5: **icomm(180)** – INTEGER array

Contains details of the wavelet transform and the problem dimension which is to be communicated to the two-dimensional discrete transform functions in this chapter.

6: **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, **wavnam** =  $\langle value \rangle$  was an illegal value.

**ifail** = 2

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

**ifail** = 3

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

**ifail** = 4

Constraint: **m**  $\geq$  2.

Constraint: **n**  $\geq$  2.

**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

Not applicable.

## 8 Further Comments

None.

## 9 Example

This example computes the two-dimensional multi-level resolution for a  $6 \times 6$  matrix by a discrete wavelet transform using the Haar wavelet with whole-point symmetric end extensions. The number of levels of transformation actually performed is one less than the maximum possible. This number of levels, the length of the wavelet filter, the total number of coefficients and the number of coefficients in each dimension for each level are printed along with the vertical detail coefficients from the first level, before a reconstruction is performed.

### 9.1 Program Text

```
function c09ab_example

fprintf('c09ab example results\n\n');

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

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

wavnam = 'Haar';
mode = 'Whole';
wtrans = 'Multilevel';
[nwl, nf, nwct, nwn, icomm, ifail] = ...
    c09ab(...
        wavnam, wtrans, mode, m, n);

lenc = nwct;
% Calculate one less than the max possible number of levels
nwl = nwl - 1;

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

% c09ab returns nwct based on max levels
% so recalculate based on nwl levels
nwct = 3*sum(dwtlvm(1:nwl).*dwtlvn(1:nwl));
nwct = nwct + dwtlvm(1)*dwtlvn(1);

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');
% print c using groups of eight
ngroups = nwct/8;
nleft = mod(nwct,8);
for i=0:ngroups-1
    fprintf('%8.4f', c(8*i+1:8*i+8));
    fprintf('\n');
end
```



2.0000	5.0000	9.0000	4.0000	4.0000	2.0000	5.0000	9.0000
1.0000	8.0000	3.0000	3.0000	5.0000	3.0000	8.0000	3.0000
8.0000	1.0000	6.0000	4.0000	6.0000	1.0000	1.0000	6.0000
8.0000	1.0000	1.0000	1.0000	2.0000	3.0000	1.0000	6.0000
9.0000	2.0000	2.0000	4.0000	6.0000	1.0000	2.0000	9.0000

---