NAG Library Routine Document

C09CCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

C09CCF computes the one-dimensional multi-level discrete wavelet transform (DWT). The initialization routine C09AAF must be called first to set up the DWT options.

2 Specification

```
SUBROUTINE C09CCF (N, X, LENC, C, NWL, DWTLEV, ICOMM, IFAIL)

INTEGER            N, LENC, NWL, DWTLEV(NWL+1), ICOMM(100), IFAIL
REAL (KIND=nag_wp) X(N), C(LENC)
```

3 Description

C09CCF computes the multi-level DWT of one-dimensional data. For a given wavelet and end extension method, C09CCF 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 routine C09AAF 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

1: N – INTEGER *Input*

On entry: the number of elements, n, in the data array x.

Constraint: this must be the same as the value N passed to the initialization routine C09AAF.

2: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: X contains the one-dimensional input dataset x_i, for $i = 1, 2, \dots, n$.

3: LENC – INTEGER *Input*

On entry: the dimension of the array C as declared in the (sub)program from which C09CCF is called. 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 routine C09AAF 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 C09AAF and $\lfloor (\bar{n} + n_f - 1)/2 \rfloor$ for MODE = 'H', 'W' or '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 C09AAF. 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.

4: C(LENC) – REAL (KIND=nag_wp) array *Output*

On exit: 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 $j = 2, 3, \ldots, n_{\text{fwd}}$.

5: NWL – INTEGER *Input*

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

Constraint: $1 \leq \text{NWL} \leq l_{\max}$, where $l_{\max}$ is the value returned in NWLMAX (the maximum number of levels) by the call to the initialization routine C09AAF.

6: DWTLEV(NWL + 1) – INTEGER array *Output*

On exit: 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 $i = 3, 4, \ldots, n_{\text{fwd}} + 1$.

7: ICOMM(100) – INTEGER array *Communication Array*

On entry: contains details of the discrete wavelet transform and the problem dimension as setup in the call to the initialization routine C09AAF.

On exit: contains additional information on the computed transform.

8: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, N is inconsistent with the value passed to the initialization routine: N $= \langle value \rangle$, N should be $\langle value \rangle$.

IFAIL $= 3$

On entry, LENC is set too small: LENC $= \langle value \rangle$.
Constraint: LENC $\geq \langle value \rangle$.

IFAIL $= 5$

On entry, NWL $= \langle value \rangle$.
Constraint: NWL ≥ 1.

On entry, NWL is larger than the maximum number of levels returned by the initialization routine: NWL $= \langle value \rangle$, maximum $= \langle value \rangle$.

IFAIL $= 7$

Either the initialization routine has not been called first or array ICOMM has been corrupted.

Either the initialization routine 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.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

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 Parallelism and Performance

Not applicable.

9 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 $i = n_{\text{fwd}}, n_{\text{fwd}} - 1, \ldots, 1$ 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 C09CDF 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.

10 Example

This example performs a multi-level resolution of a dataset using the Daubechies wavelet (see WAVNAM = 'DB4' in C09AAF) 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 C09CDF.

10.1 Program Text

```
    Program c09ccfe

!     C09CCF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: c09aaf, c09ccf, c09cdf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                              :: ifail, lenc, n, nf, nnz, nwc, nwl,   &
                                              nwlinv, nwlmax
      Character (10)                       :: mode, wavnam, wtrans
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: c(:), x(:), y(:)
      Integer, Allocatable                 :: dwtlev(:)
      Integer                              :: icomm(100)
!     .. Intrinsic Procedures ..
      Intrinsic                            :: sum
!     .. Executable Statements ..
      Write (nout,*) 'C09CCF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
!     Read problem parameters
      Read (nin,*) n
      Read (nin,*) wavnam, mode
      Allocate (x(n),y(n))

      Write (nout,99999) wavnam, mode, n

!     Read data array and write it out

      Read (nin,*) x(1:n)

      Write (nout,*) ' Input Data      X :'
      Write (nout,99998) x(1:n)

!     Query wavelet filter dimensions
!     For Multi-Resolution Analysis, decomposition, wtrans = 'M'
      wtrans = 'Multilevel'

!     ifail: behaviour on error exit
```

```
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call c09aaf(wavnam,wtrans,mode,n,nwlmax,nf,nwc,icomm,ifail)

      lenc = nwc
      Allocate (c(lenc),dwtlev(nwlmax+1))

      nwl = nwlmax

!     Perform Discrete Wavelet transform
      ifail = 0
      Call c09ccf(n,x,lenc,c,nwl,dwtlev,icomm,ifail)

      Write (nout,99997) nwl
      Write (nout,99996)
      Write (nout,99995) dwtlev(1:nwl+1)
      nnz = sum(dwtlev(1:nwl+1))
      Write (nout,99994)
      Write (nout,99998) c(1:nnz)

!     Reconstruct original data
      nwlinv = nwl

      ifail = 0
      Call c09cdf(nwlinv,lenc,c,n,y,icomm,ifail)

      Write (nout,99993)
      Write (nout,99998) y(1:n)

99999 Format (1X,' MLDWT :: Wavelet : ',A10,', End mode : ',A10,' N = ',I10)
99998 Format (8(F8.4,1X):)
99997 Format (1X,' Number of Levels : ',I10)
99996 Format (1X,' Number of coefficients in each level : ')
99995 Format (8(I8,1X):)
99994 Format (1X,' Wavelet coefficients C : ')
99993 Format (1X,' Reconstruction              Y : ')
    End Program c09ccfe
```

10.2 Program Data

```
C09CCF Example Program Data
 64                                  : n
 DB4  Zero                           : wavnam, mode
 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         : x(1:n)
```

10.3 Program Results

```
 C09CCF Example Program Results
  MLDWT :: Wavelet : DB4       , End mode : Zero       N =         64
  Input Data      X :
  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
```

```
 Number of Levels :          6
 Number of coefficients in each level :
      7        7        8       10       14       21       35
 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
```