# NAG Library Routine Document

# C09FCF

**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

C09FCF computes the three-dimensional multi-level discrete wavelet transform (DWT). The initialization routine C09ACF must be called first to set up the DWT options.

## 2 Specification

```
SUBROUTINE C09FCF (M, N, FR, A, LDA, SDA, LENC, C, NWL, DWTLVM, DWTLVN,         &
                   DWTLVFR, ICOMM, IFAIL)

INTEGER            M, N, FR, LDA, SDA, LENC, NWL, DWTLVM(NWL),                  &
                   DWTLVN(NWL), DWTLVFR(NWL), ICOMM(260), IFAIL
REAL (KIND=nag_wp) A(LDA,SDA,FR), C(LENC)
```

## 3 Description

C09FCF computes the multi-level DWT of three-dimensional data. For a given wavelet and end extension method, C09FCF will compute a multi-level transform of a three-dimensional array $A$, using a specified number, $n_l$, of levels. The number of levels specified, $n_l$, must be no more than the value $l_{\max}$ returned in NWL by the initialization routine C09ACF 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 seven sets of detail coefficients are stored at this level while the approximation coefficients are used as the input to a repeat of the wavelet transform at the next level. This process is continued until, at level $n_l$, all eight types of coefficients are stored. All coefficients are packed into a single array.

## 4 References

None.

## 5 Parameters

1: M – INTEGER *Input*

*On entry*: the first dimension of the input data: the number of rows of each two-dimensional frame.

*Constraint*: this must be the same as the value M passed to the initialization routine C09ACF.

2: N – INTEGER *Input*

*On entry*: the second dimension of the input data: the number of columns of each two-dimensional frame.

*Constraint*: this must be the same as the value N passed to the initialization routine C09ACF.

3: FR – INTEGER *Input*

*On entry*: the third dimension of the input data: the number of two-dimensional frames.

*Constraint*: this must be the same as the value FR passed to the initialization routine C09ACF.

4: A(LDA,SDA,FR) – REAL (KIND=nag_wp) array *Input*

*On entry*: the $m$ by $n$ by $fr$ input three-dimensional array $A$.

5: LDA – INTEGER *Input*

*On entry*: the first dimension of the array A as declared in the (sub)program from which C09FCF is called.

*Constraint*: LDA $\geq$ M.

6: SDA – INTEGER *Input*

*On entry*: the second dimension of the array A as declared in the (sub)program from which C09FCF is called.

*Constraint*: SDA $\geq$ N.

7: LENC – INTEGER *Input*

*On entry*: the dimension of the array C as declared in the (sub)program from which C09FCF is called.

*Constraint*: LENC $\geq n_{\mathrm{ct}}$, where $n_{\mathrm{ct}}$ is the total number of wavelet coefficients that correspond to a transform with NWL levels.

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

*On exit*: the coefficients of the discrete wavelet transform.

Let $q(i)$ denote the number of coefficients (of each type) at level $i$, for $i = 1, 2, \ldots, n_l$, such that $q(i) = \mathrm{DWTLVM}(n_l - i + 1) \times \mathrm{DWTLVN}(n_l - i + 1) \times \mathrm{DWTLVFR}(n_l - i + 1)$. Then, letting $k_1 = q(n_l)$ and $k_{j+1} = k_j + q(n_l - \lceil j/7 \rceil + 1)$, for $j = 1, 2, \ldots, 7n_l$, the coefficients are stored in C as follows:

C$(i)$, for $i = 1, 2, \ldots, k_1$
: Contains the level $n_l$ approximation coefficients, $a_{n_l}$. Note that for computational efficiency reasons these coefficients are stored as $\mathrm{DWTLVM}(1) \times \mathrm{DWTLVN}(1) \times \mathrm{DWTLVFR}(1)$ in C.

C$(i)$, for $i = k_j + 1, \ldots, k_{j+1}$
: Contains the level $n_l - \lceil j/7 \rceil + 1$ detail coefficients. These are:

LLH coefficients if $j \bmod 7 = 1$;

LHL coefficients if $j \bmod 7 = 2$;

LHH coefficients if $j \bmod 7 = 3$;

HLL coefficients if $j \bmod 7 = 4$;

HLH coefficients if $j \bmod 7 = 5$;

HHL coefficients if $j \bmod 7 = 6$;

HHH coefficients if $j \bmod 7 = 0$,

for $j = 1, \ldots, 7n_l$.

Note that for computational efficiency reasons these coefficients are stored as $\mathrm{DWTLVFR}(\lceil j/7 \rceil) \times \mathrm{DWTLVM}(\lceil j/7 \rceil) \times \mathrm{DWTLVN}(\lceil j/7 \rceil)$ in C.

See Section 9 for details of how to access each set of coefficients in order to perform extraction from C following a call to this routine, or insertion into C before a call to the three-dimensional multi-level inverse routine C09FDF.

9: NWL – INTEGER *Input*

*On entry*: the number of levels, $n_l$, in the multi-level resolution to be performed.

*Constraint*: $1 \leq \text{NWL} \leq l_{\max}$, where $l_{\max}$ is the value returned in NWL (the maximum number of levels) by the call to the initialization routine C09ACF.

10: DWTLVM(NWL) – INTEGER array *Output*

*On exit*: 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 $(n_l - i + 1)$th level of resolution, for $i = 1, 2, \ldots, n_l$.

11: DWTLVN(NWL) – INTEGER array *Output*

*On exit*: 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 $(n_l - i + 1)$th level of resolution, for $i = 1, 2, \ldots, n_l$.

12: DWTLVFR(NWL) – INTEGER array *Output*

*On exit*: the number of coefficients in the third dimension for each coefficient type at each level. DWTLVFR($i$) contains the number of coefficients in the third dimension (for each coefficient type computed) at the $(n_l - i + 1)$th level of resolution, for $i = 1, 2, \ldots, n_l$.

13: ICOMM(260) – 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 C09ACF.

*On exit*: contains additional information on the computed transform.

14: 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, FR = ⟨*value*⟩.
Constraint: FR = ⟨*value*⟩, the value of FR on initialization (see C09ACF).

On entry, M = ⟨*value*⟩.
Constraint: M = ⟨*value*⟩, the value of M on initialization (see C09ACF).

On entry, N = ⟨*value*⟩.
Constraint: N = ⟨*value*⟩, the value of N on initialization (see C09ACF).

IFAIL = 2

On entry, LDA = ⟨*value*⟩ and M = ⟨*value*⟩.
Constraint: LDA ≥ M.

On entry, SDA = ⟨*value*⟩ and N = ⟨*value*⟩.
Constraint: SDA ≥ N.

IFAIL = 3

On entry, LENC = ⟨*value*⟩.
Constraint: LENC ≥ ⟨*value*⟩, the total number of coefficents to be generated.

IFAIL = 5

On entry, NWL = ⟨*value*⟩.
Constraint: NWL ≥ 1.

On entry, NWL = ⟨*value*⟩ and NWL = ⟨*value*⟩ in C09ACF.
Constraint: NWL ≤ NWL in C09ACF.

IFAIL = 6

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

The initialization routine was called with WTRANS = 'S'.

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 example program shows how the wavelet coefficients at each level can be extracted from the output array C. Denoising can be carried out by applying a thresholding operation to the detail coefficients at every level. If $c_{ij}$ is a detail coefficient then $\hat{c}_{ij} = c_{ij} + \sigma\epsilon_{ij}$ and $\sigma\epsilon_{ij}$ is the transformed noise term. If some threshold parameter $\alpha$ 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 C09FDF 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.

## 9 Example

This example computes the three-dimensional multi-level discrete wavelet decomposition for $7 \times 6 \times 5$ input data using the biorthogonal wavelet of order 1.1 (set WAVNAM = 'BIOR1.1' in C09ACF) with periodic end extension, prints a selected set of wavelet coefficients and then reconstructs and verifies that the reconstruction matches the original data.

## 9.1 Program Text

```
    Program c09fcfe

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: c09acf, c09fcf, c09fdf, nag_wp, x02ajf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: eps, esq, frob
      Integer                          :: fr, i, i1, ifail, ilevel,           &
                                          itype_coeffs, j, k, lda, ldb, ldd,  &
                                          lenc, locc, m, n, nf, nwcfr, nwcm,  &
                                          nwcn, nwct, nwl, sda, sdb, sdd,     &
                                          want_coeffs, want_level
      Character (10)                   :: mode, wavnam, wtrans
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:,:), b(:,:,:), c(:), d(:,:,:),  &
                                          e(:,:,:)
      Integer, Allocatable             :: dwtlvfr(:), dwtlvm(:), dwtlvn(:)
      Integer                          :: icomm(260)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, real, sqrt
!     .. Executable Statements ..
      Continue
      Write (nout,*) 'C09FCF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
!     Read problem parameters
      Read (nin,*) m, n, fr
      Read (nin,*) wavnam, mode
      lda = m
      sda = n
      ldb = m
      sdb = n
      Allocate (a(lda,sda,fr),b(ldb,sdb,fr),e(m,n,fr))

      Write (nout,99999) wavnam, mode, m, n, fr

!     Read data array and write it out

      Do j = 1, fr
        Do i = 1, m
          Read (nin,*) a(i,1:n,j)
        End Do
        If (j<fr) Read (nin,*)
      End Do

      Write (nout,*) ' Input Data     A :'
      Do j = 1, fr
        Write (nout,99997) j
        Do i = 1, m
          Write (nout,99998) a(i,1:n,j)
        End Do
      End Do

!     Query wavelet filter dimensions
!     For Multi-Resolution Analysis, decomposition, wtrans = 'M'
      wtrans = 'Multilevel'
      ifail = 0
      Call c09acf(wavnam,wtrans,mode,m,n,fr,nwl,nf,nwct,nwcn,nwcfr,icomm, &
        ifail)

      lenc = nwct
      Allocate (c(lenc),dwtlvm(nwl),dwtlvn(nwl),dwtlvfr(nwl))
```

```
!     Perform Discrete Wavelet transform
      ifail = 0
      Call c09fcf(m,n,fr,a,lda,sda,lenc,c,nwl,dwtlvm,dwtlvn,dwtlvfr,icomm, &
        ifail)

      Write (nout,99996) nwl
      Write (nout,99995)
      Write (nout,99992) dwtlvm(1:nwl)
      Write (nout,99994)
      Write (nout,99992) dwtlvn(1:nwl)
      Write (nout,99993)
      Write (nout,99992) dwtlvfr(1:nwl)

!     Print the first level HLL coefficients
      want_level = 1
      want_coeffs = 4

!     Identify each set of coefficients in C
      Do ilevel = nwl, 1, -1

        If (ilevel/=want_level) Cycle

        nwcm = dwtlvm(nwl-ilevel+1)
        nwcn = dwtlvn(nwl-ilevel+1)
        nwcfr = dwtlvfr(nwl-ilevel+1)

        ldd = nwcm
        sdd = nwcn
        Allocate (d(ldd,sdd,nwcfr))

        Write (nout,99987) ilevel, nwcm, nwcn, nwcfr

        Do itype_coeffs = 0, 7

          If (itype_coeffs/=want_coeffs) Cycle

!         Unless we're looking at the deepest level of nesting, which contains
!         approximation coefficients, advance the pointer on past the preceding
!         levels
          If (ilevel==nwl) Then
            locc = 0
          Else
            locc = 8*dwtlvm(1)*dwtlvn(1)*dwtlvfr(1)
            Do i = ilevel + 1, nwl - 1
              locc = locc + 7*dwtlvm(nwl-i+1)*dwtlvn(nwl-i+1)*dwtlvfr(nwl-i+1)
            End Do
          End If

!         Now decide which coefficient type we are considering
          Select Case (itype_coeffs)
          Case (0)
            If (ilevel==nwl) Then
              Write (nout,99986) 'Approximation coefficients (LLL) '
              locc = locc + 1
            End If
          Case (1)
            Write (nout,99986) 'Detail coefficients (LLH)  '
            If (ilevel==nwl) Then
!             Advance pointer past approximation coefficients
              locc = locc + nwcm*nwcn*nwcfr + 1
            Else
              locc = locc + 1
            End If
          Case (2)
            Write (nout,99986) 'Detail coefficients (LHL)  '
            If (ilevel==nwl) Then
!             Advance pointer past approximation coefficients and 1 set of
!             detail coefficients
              locc = locc + 2*nwcm*nwcn*nwcfr + 1
            Else
!             Advance pointer past 1 set of detail coefficients
```

```
            locc = locc + nwcm*nwcn*nwcfr + 1
          End If
        Case (3)
          Write (nout,99986) 'Detail coefficients (LHH)  '
          If (ilevel==nwl) Then
!           Advance pointer past approximation coefficients and 2 sets of
!           detail coefficients
            locc = locc + 3*nwcm*nwcn*nwcfr + 1
          Else
!           Advance pointer past 2 sets of detail coefficients
            locc = locc + 2*nwcm*nwcn*nwcfr + 1
          End If
        Case (4)
          Write (nout,99986) 'Detail coefficients (HLL)  '
          If (ilevel==nwl) Then
!           Advance pointer past approximation coefficients and 3 sets of
!           detail coefficients
            locc = locc + 4*nwcm*nwcn*nwcfr + 1
          Else
!           Advance pointer past 3 sets of detail coefficients
            locc = locc + 3*nwcm*nwcn*nwcfr + 1
          End If
        Case (5)
          Write (nout,99986) 'Detail coefficients (HLH)  '
          If (ilevel==nwl) Then
!           Advance pointer past approximation coefficients and 4 sets of
!           detail coefficients
            locc = locc + 5*nwcm*nwcn*nwcfr + 1
          Else
!           Advance pointer past 4 sets of detail coefficients
            locc = locc + 4*nwcm*nwcn*nwcfr + 1
          End If
        Case (6)
          Write (nout,99986) 'Detail coefficients (HHL)  '
          If (ilevel==nwl) Then
!           Advance pointer past approximation coefficients and 5 sets of
!           detail coefficients
            locc = locc + 6*nwcm*nwcn*nwcfr + 1
          Else
!           Advance pointer past 4 sets of detail coefficients
            locc = locc + 5*nwcm*nwcn*nwcfr + 1
          End If
        Case (7)
          Write (nout,99986) 'Detail coefficients (HHH)  '
          If (ilevel==nwl) Then
!           Advance pointer past approximation coefficients and 6 sets of
!           detail coefficients
            locc = locc + 7*nwcm*nwcn*nwcfr + 1
          Else
!           Advance pointer past 5 sets of detail coefficients
            locc = locc + 6*nwcm*nwcn*nwcfr + 1
          End If
        End Select

        If (itype_coeffs>0 .Or. ilevel==nwl) Then

          If (itype_coeffs==0) Then

!            For a multi level transform approx coeffs stored as nwcm x nwcn x
nwcfr...
            i1 = locc
            Do k = 1, nwcfr
              Do j = 1, nwcn
                Do i = 1, nwcm
                  d(i,j,k) = c(i1)
                  i1 = i1 + 1
                End Do
              End Do
            End Do

          Else
```

```
!             but detail coefficients are stored as ncwfr x nwcm x nwcn
              Do k = 1, nwcfr
                Do j = 1, nwcn
                  Do i = 1, nwcm
                    i1 = locc - 1 + (j-1)*nwcfr*nwcm + (i-1)*nwcfr + k
                    d(i,j,k) = c(i1)
                  End Do
                End Do
              End Do

            End If

!           Print out the selected set of coefficients
            Write (nout,99989) ilevel, itype_coeffs
            Do k = 1, nwcfr
              Write (nout,99988) k
              Do i = 1, nwcm
                Write (nout,99998) d(i,1:nwcn,k)
              End Do
            End Do

          End If
        End Do
        Deallocate (d)
      End Do

!     Reconstruct original data
      ifail = 0
      Call c09fdf(nwl,lenc,c,m,n,fr,b,ldb,sdb,icomm,ifail)

!     Check reconstruction matches original
      eps = 10.0_nag_wp*real(m,kind=nag_wp)*real(n,kind=nag_wp)* &
        real(fr,kind=nag_wp)*x02ajf()

      e(1:m,1:n,1:fr) = b(1:m,1:n,1:fr) - a(1:m,1:n,1:fr)
      frob = 0.0_nag_wp
      Do k = 1, fr
        esq = 0.0_nag_wp
        Do j = 1, n
          Do i = 1, m
            esq = esq + e(i,j,k)**2
          End Do
        End Do
        frob = max(frob,sqrt(esq))
      End Do

      If (frob>eps) Then
        Write (nout,99991)
      Else
        Write (nout,99990)
      End If

99999 Format (1X,' MLDWT :: Wavelet  : ',A/1X,'          End mode : ',A/1X, &
        '          M        : ',I4/1X,'          N        : ',I4/1X, &
        '          FR       : ',I4/)
99998 Format (8(F8.4,1X):)
99997 Format (1X,' Frame ',I2,' : ')
99996 Format (/1X,' Number of Levels : ',I10)
99995 Format (1X,' Number of coefficients in 1st dimension for each level :')
99994 Format (1X,' Number of coefficients in 2nd dimension for each level :')
99993 Format (1X,' Number of coefficients in 3rd dimension for each level :')
99992 Format (8(I8,1X):)
99991 Format (/1X,' Fail: Frobenius norm of B-A, where A is the original '/1X, &
        ' data and B is the reconstrucion, is too large.')
99990 Format (/1X,' Success: the reconstruction matches the original.')
99989 Format (1X,' Level ',I2,', Coefficients ',I2,' : ')
```

```
99988 Format (1X,' Frame ',I2,' : ')
99987 Format (/1X,70('-')/1X,'Level : ',I10,'; output is ',I10,' by ',I10, &
        ' by ',I10/1X,70('-'))
99986 Format (/1X,A)
    End Program c09fcfe
```

## 9.2 Program Data

```
C09FCF Example Program Data
 7, 6, 5        : m, n, fr
 Bior1.1 period : wavnam, mode
 3.0000  2.0000  2.0000  2.0000  1.0000  1.0000
 2.0000  9.0000  1.0000  2.0000  1.0000  3.0000
 2.0000  5.0000  1.0000  2.0000  1.0000  1.0000
 1.0000  6.0000  2.0000  2.0000  7.0000  2.0000
 5.0000  3.0000  2.0000  2.0000  4.0000  7.0000
 2.0000  2.0000  1.0000  1.0000  2.0000  1.0000
 6.0000  2.0000  1.0000  3.0000  6.0000  9.0000

 2.0000  1.0000  5.0000  1.0000  2.0000  3.0000
 2.0000  9.0000  5.0000  2.0000  1.0000  2.0000
 2.0000  3.0000  2.0000  7.0000  1.0000  1.0000
 2.0000  1.0000  1.0000  2.0000  3.0000  1.0000
 2.0000  1.0000  2.0000  8.0000  3.0000  3.0000
 1.0000  4.0000  5.0000  1.0000  2.0000  7.0000
 8.0000  1.0000  3.0000  9.0000  1.0000  2.0000

 3.0000  1.0000  4.0000  1.0000  1.0000  1.0000
 1.0000  1.0000  2.0000  1.0000  2.0000  6.0000
 4.0000  1.0000  7.0000  2.0000  5.0000  6.0000
 3.0000  2.0000  1.0000  5.0000  9.0000  5.0000
 1.0000  1.0000  2.0000  2.0000  2.0000  1.0000
 2.0000  6.0000  3.0000  9.0000  5.0000  1.0000
 1.0000  1.0000  8.0000  2.0000  1.0000  3.0000

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

 5.0000  2.0000  1.0000  2.0000  1.0000  1.0000
 3.0000  1.0000  9.0000  1.0000  2.0000  1.0000
 2.0000  3.0000  1.0000  1.0000  7.0000  2.0000
 7.0000  2.0000  2.0000  6.0000  1.0000  1.0000
 5.0000  1.0000  7.0000  2.0000  1.0000  1.0000
 2.0000  1.0000  3.0000  2.0000  2.0000  1.0000
 5.0000  3.0000  9.0000  1.0000  4.0000  1.0000
```

## 9.3 Program Results

```
 C09FCF Example Program Results

  MLDWT :: Wavelet   : Bior1.1
           End mode : period
           M        :    7
           N        :    6
           FR       :    5

  Input Data     A :
  Frame  1 :
  3.0000   2.0000   2.0000   2.0000   1.0000   1.0000
  2.0000   9.0000   1.0000   2.0000   1.0000   3.0000
  2.0000   5.0000   1.0000   2.0000   1.0000   1.0000
  1.0000   6.0000   2.0000   2.0000   7.0000   2.0000
  5.0000   3.0000   2.0000   2.0000   4.0000   7.0000
  2.0000   2.0000   1.0000   1.0000   2.0000   1.0000
  6.0000   2.0000   1.0000   3.0000   6.0000   9.0000
```

```
 Frame  2 :
 2.0000   1.0000   5.0000   1.0000   2.0000   3.0000
 2.0000   9.0000   5.0000   2.0000   1.0000   2.0000
 2.0000   3.0000   2.0000   7.0000   1.0000   1.0000
 2.0000   1.0000   1.0000   2.0000   3.0000   1.0000
 2.0000   1.0000   2.0000   8.0000   3.0000   3.0000
 1.0000   4.0000   5.0000   1.0000   2.0000   7.0000
 8.0000   1.0000   3.0000   9.0000   1.0000   2.0000
 Frame  3 :
 3.0000   1.0000   4.0000   1.0000   1.0000   1.0000
 1.0000   1.0000   2.0000   1.0000   2.0000   6.0000
 4.0000   1.0000   7.0000   2.0000   5.0000   6.0000
 3.0000   2.0000   1.0000   5.0000   9.0000   5.0000
 1.0000   1.0000   2.0000   2.0000   2.0000   1.0000
 2.0000   6.0000   3.0000   9.0000   5.0000   1.0000
 1.0000   1.0000   8.0000   2.0000   1.0000   3.0000
 Frame  4 :
 5.0000   8.0000   1.0000   2.0000   2.0000   1.0000
 1.0000   2.0000   2.0000   9.0000   2.0000   9.0000
 2.0000   2.0000   2.0000   1.0000   1.0000   3.0000
 1.0000   1.0000   1.0000   5.0000   1.0000   2.0000
 3.0000   2.0000   8.0000   1.0000   9.0000   2.0000
 2.0000   1.0000   9.0000   1.0000   2.0000   2.0000
 3.0000   6.0000   5.0000   3.0000   2.0000   2.0000
 Frame  5 :
 5.0000   2.0000   1.0000   2.0000   1.0000   1.0000
 3.0000   1.0000   9.0000   1.0000   2.0000   1.0000
 2.0000   3.0000   1.0000   1.0000   7.0000   2.0000
 7.0000   2.0000   2.0000   6.0000   1.0000   1.0000
 5.0000   1.0000   7.0000   2.0000   1.0000   1.0000
 2.0000   1.0000   3.0000   2.0000   2.0000   1.0000
 5.0000   3.0000   9.0000   1.0000   4.0000   1.0000

 Number of Levels :          2
 Number of coefficients in 1st dimension for each level :
      2        4
 Number of coefficients in 2nd dimension for each level :
      2        3
 Number of coefficients in 3rd dimension for each level :
      2        3

----------------------------------------------------------------------
Level :          1; output is          4 by          3 by          3
----------------------------------------------------------------------

Detail coefficients (HLL)
 Level  1, Coefficients  4 :
 Frame  1 :
-4.9497   0.0000   0.0000
 0.7071   1.7678  -3.1820
 0.7071   2.1213   1.7678
 0.0000   0.0000   0.0000
 Frame  2 :
 4.2426  -2.1213  -4.9497
 0.7071  -0.0000  -0.7071
-1.4142  -3.1820   1.4142
 0.0000   0.0000   0.0000
 Frame  3 :
 2.1213  -4.9497  -0.7071
-2.8284  -4.2426   4.9497
 2.1213   2.8284  -0.7071
 0.0000   0.0000   0.0000

 Success: the reconstruction matches the original.
```