

# NAG Library Routine Document

## C09FAF

**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

C09FAF computes the three-dimensional discrete wavelet transform (DWT) at a single level. The initialization routine C09ACF must be called first to set up the DWT options.

### 2 Specification

```
SUBROUTINE C09FAF (M, N, FR, A, LDA, SDA, LENC, C, ICOMM, IFAIL)
  INTEGER          M, N, FR, LDA, SDA, LENC, ICOMM(260), IFAIL
  REAL (KIND=nag_wp) A(LDA,SDA,FR), C(LENC)
```

### 3 Description

C09FAF computes the three-dimensional DWT of some given three-dimensional input data, considered as a number of two-dimensional frames, at a single level. For a chosen wavelet filter pair, the output coefficients are obtained by applying convolution and downsampling by two to the input data,  $A$ , first over columns, next over rows and finally across frames. The three-dimensional approximation coefficients are produced by the low pass filter over columns, rows and frames. In addition there are 7 sets of three-dimensional detail coefficients, each corresponding to a different order of low pass and high pass filters (see the C09 Chapter Introduction). All coefficients are packed into a single array. To reduce distortion effects at the ends of the data array, several end extension methods are commonly used. Those provided are: periodic or circular convolution end extension, half-point symmetric end extension, whole-point symmetric end extension and zero end extension. The total number,  $n_{ct}$ , of coefficients computed is returned by the initialization routine C09ACF.

### 4 References

Daubechies I (1992) *Ten Lectures on Wavelets* SIAM, Philadelphia

### 5 Arguments

- 1: M – INTEGER *Input*  
*On entry:* 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 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 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  $fr$  three-dimensional input data  $A$ , where  $A_{ijk}$  is stored in  $A(i, j, k)$ .

- 5: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array A as declared in the (sub)program from which C09FAF 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 C09FAF 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 C09FAF is called.  
*Constraint:*  $LENC \geq n_{ct}$ , where  $n_{ct}$  is the total number of wavelet coefficients, as returned by C09ACF.
- 8: C(LENC) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* 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 C09FYF or C09FZF is recommended. For completeness the following description provides details of precisely how the coefficients are stored in C but this information should only be required in rare cases.  
 The 8 sets of coefficients are stored in the following order: approximation coefficients (LLL) first, followed by 7 sets of detail coefficients: LLH, LHL, LHH, HLL, HLH, HHL, HHH, where L indicates the low pass filter, and H the high pass filter being applied to, respectively, the columns of length M, the rows of length N and then the frames of length FR. Note that for computational efficiency reasons each set of coefficients is stored in the order  $n_{cfr} \times n_{cm} \times n_{cn}$  (see output arguments NWCFR, NWCT and NWCN in C09ACF). See Section 10 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 inverse routine C09FBF.
- 9: 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.
- 10: IFAIL – INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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,  $FR = \langle value \rangle$ .

Constraint:  $FR = \langle value \rangle$ , the value of  $FR$  on initialization (see  $C09ACF$ ).

On entry,  $M = \langle value \rangle$ .

Constraint:  $M = \langle value \rangle$ , the value of  $M$  on initialization (see  $C09ACF$ ).

On entry,  $N = \langle value \rangle$ .

Constraint:  $N = \langle value \rangle$ , the value of  $N$  on initialization (see  $C09ACF$ ).

$IFAIL = 2$

On entry,  $LDA = \langle value \rangle$  and  $M = \langle value \rangle$ .

Constraint:  $LDA \geq M$ .

On entry,  $SDA = \langle value \rangle$  and  $N = \langle value \rangle$ .

Constraint:  $SDA \geq N$ .

$IFAIL = 3$

On entry,  $LENC = \langle value \rangle$  and  $n_{ct} = \langle value \rangle$ .

Constraint:  $LENC \geq n_{ct}$ , where  $n_{ct}$  is the number of DWT coefficients returned by  $C09ACF$  in argument  $NWCT$ .

$IFAIL = 6$

Either the communication array  $ICOMM$  has been corrupted or there has not been a prior call to the initialization routine  $C09ACF$ .

The initialization routine was called with  $WTRANS = 'M'$ .

$IFAIL = -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

$IFAIL = -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

$IFAIL = -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation 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

$C09FAF$  is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

None.

## 10 Example

This example computes the three-dimensional discrete wavelet decomposition for  $5 \times 4 \times 3$  input data using the Haar wavelet, WAVNAM = 'HAAR', with half point end extension, prints the wavelet coefficients and then reconstructs the original data using C09FBF. This example also demonstrates in general how to access any set of coefficients following a single level transform.

### 10.1 Program Text

```

Program c09fafa

!      C09FAF Example Program Text
!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: c09acf, c09faf, c09fbf, c09fyf, nag_wp
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Integer                    :: cindex, fr, i, ifail, j, lda, ldb,    &
                             ldd, lenc, m, n, nf, nwcfr, nwcm,    &
                             nwcn, nwct, nwl, sda, sdb, sdd
Character (12)              :: mode, wavnam, wtrans
Character (33)              :: title
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:,:,:), b(:,:,:), c(:), d(:,:,:)
Integer                    :: icomm(260)
Character (3)               :: cpass(0:7)
!      .. Executable Statements ..
!      Continue
!      Write (nout,*) 'C09FAF Example Program Results'

!      Skip heading in data file
!      Read (nin,*)
!      Read problem parameters.
!      Read (nin,*) m, n, fr
!      Read (nin,*) wavnam, mode
!      Write (nout,99999) wavnam, mode

!      lda = m
!      sda = n
!      Allocate (a(lda,sda,fr))
!      ldb = m
!      sdb = n
!      Allocate (b(ldb,sdb,fr))

!      Read data array
!      Do j = 1, fr
!         Read (nin,*)
!         Read (nin,*)(a(i,1:n,j),i=1,m)
!      End Do

!      Write (nout,99998) 'Input Data'
!      Do j = 1, fr
!         Write (nout,99996) j
!         Do i = 1, m
!            Write (nout,99997) a(i,1:n,j)

```

```

      End Do
    End Do

!   Query wavelet filter dimensions
    wtrans = 'Single Level'

!   ifail: behaviour on error exit
!   =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
    ifail = 0
    Call c09acf(wavnam,wtrans,mode,m,n,fr,nwl,nf,nwct,nwcn,nwcf,icomm,      &
      ifail)
    nwcm = nwct/(8*nwcn*nwcf)
    lenc = nwct
    Allocate (c(lenc))

!   3D DWT decomposition
    ifail = 0
    Call c09faf(m,n,fr,a,lda,sda,lenc,c,icomm,ifail)

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

!   Loop over low/high passes from LLL to HHH
    cpass(0:7) = ('LLL','LLH','LHL','LHH','HLL','HLH','HHL','HHH')
    Do cindex = 0, 7
      If (cindex==0) Then
        title = 'Approximation coefficients (LLL)'
      Else
        title = 'Detail coefficients (' // cpass(cindex) // ')'
      End If

!   Extract coefficients
      Call c09fyf(0,cindex,lenc,c,d,ldd,sdd,icomm,ifail)

      Write (nout,99992) title
      Write (nout,99995)('Frame ',j,j=1,nwcf)
      Write (nout,99994) cindex, (d(1,1:nwcn,j),j=1,nwcf)
      Do i = 2, nwcm
        Write (nout,99993)(d(i,1:nwcn,j),j=1,nwcf)
      End Do
    End Do

!   3D DWT reconstruction
    ifail = 0
    Call c09fbf(m,n,fr,lenc,c,b,ldb,sdb,icomm,ifail)

    Write (nout,99998) 'Output Data                                B'
    Do j = 1, fr
      Write (nout,99996) j
      Do i = 1, m
        Write (nout,99997) b(i,1:n,j)
      End Do
    End Do

99999 Format (/,1X,'DWT ::',/,1X,'                               Wavelet : ',A,/,1X,      &
  '                               End mode: ',A)
99998 Format (/,1X,A,' : ')
99997 Format (1X,8(F8.4,1X),:)
99996 Format (1X,'Frame ',I2,' : ')
99995 Format (11X,6(10X,A,I2))
99994 Format (4X,I4,6X,8(1X,F8.4))
99993 Format (14X,8(1X,F8.4))
99992 Format (/,1X,A)
    End Program c09faf

```

## 10.2 Program Data

C09FAF Example Program Data

```
5      4      3      : m, n, fr
Haar   half      : wavnam, mode
```

```
3.0000 2.0000 2.0000 2.0000
2.0000 9.0000 1.0000 2.0000
2.0000 5.0000 1.0000 2.0000
1.0000 6.0000 2.0000 2.0000
5.0000 3.0000 2.0000 2.0000 : Frame 1
```

```
2.0000 1.0000 5.0000 1.0000
2.0000 9.0000 5.0000 2.0000
2.0000 3.0000 2.0000 7.0000
2.0000 1.0000 1.0000 2.0000
2.0000 1.0000 2.0000 8.0000 : Frame 2
```

```
3.0000 1.0000 4.0000 1.0000
1.0000 1.0000 2.0000 1.0000
4.0000 1.0000 7.0000 2.0000
3.0000 2.0000 1.0000 5.0000
1.0000 1.0000 2.0000 2.0000 : Frame 3
```

## 10.3 Program Results

C09FAF Example Program Results

DWT ::

```
Wavelet : Haar
End mode: half
```

```
Input Data          A :
Frame 1 :
 3.0000  2.0000  2.0000  2.0000
 2.0000  9.0000  1.0000  2.0000
 2.0000  5.0000  1.0000  2.0000
 1.0000  6.0000  2.0000  2.0000
 5.0000  3.0000  2.0000  2.0000
Frame 2 :
 2.0000  1.0000  5.0000  1.0000
 2.0000  9.0000  5.0000  2.0000
 2.0000  3.0000  2.0000  7.0000
 2.0000  1.0000  1.0000  2.0000
 2.0000  1.0000  2.0000  8.0000
Frame 3 :
 3.0000  1.0000  4.0000  1.0000
 1.0000  1.0000  2.0000  1.0000
 4.0000  1.0000  7.0000  2.0000
 3.0000  2.0000  1.0000  5.0000
 1.0000  1.0000  2.0000  2.0000
```

Approximation coefficients (LLL)

```
          Frame 1          Frame 2
0      10.6066  7.0711  4.2426  5.6569
       7.7782  6.7175  7.0711  10.6066
       7.7782  9.8995  2.8284  5.6569
```

Detail coefficients (LLH)

```
          Frame 1          Frame 2
1      0.7071 -2.1213  0.0000  0.0000
       2.1213 -1.7678  0.0000  0.0000
       3.5355 -4.2426  0.0000  0.0000
```

Detail coefficients (LHL)

```
          Frame 1          Frame 2
2     -4.2426  2.1213  1.4142  2.8284
       -2.8284 -2.4749  2.8284  0.7071
       2.1213 -4.2426  0.0000  0.0000
```

## Detail coefficients (LHH)

	Frame 1		Frame 2	
3	0.0000	-2.8284	0.0000	0.0000
	-2.8284	1.7678	0.0000	0.0000
	0.7071	4.2426	0.0000	0.0000

## Detail coefficients (HLL)

	Frame 1		Frame 2	
4	-4.9497	0.0000	1.4142	1.4142
	0.7071	1.7678	-0.0000	2.1213
	0.0000	0.0000	0.0000	0.0000

## Detail coefficients (HLH)

	Frame 1		Frame 2	
5	0.7071	0.7071	0.0000	0.0000
	-0.7071	-2.4749	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000

## Detail coefficients (HHL)

	Frame 1		Frame 2	
6	5.6569	0.7071	1.4142	1.4142
	0.0000	-1.7678	1.4142	6.3640
	0.0000	0.0000	0.0000	0.0000

## Detail coefficients (HHH)

	Frame 1		Frame 2	
7	0.0000	0.0000	0.0000	0.0000
	1.4142	1.0607	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000

## Output Data

B :

## Frame 1 :

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

## Frame 2 :

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

## Frame 3 :

3.0000	1.0000	4.0000	1.0000
1.0000	1.0000	2.0000	1.0000
4.0000	1.0000	7.0000	2.0000
3.0000	2.0000	1.0000	5.0000
1.0000	1.0000	2.0000	2.0000

---