NAG Library Routine Document

C09EAF

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

C09EAF computes the two-dimensional discrete wavelet transform (DWT) at a single level. The initialization routine C09ABF must be called first to set up the DWT options.

2 Specification

```
SUBROUTINE C09EAF (M, N, A, LDA, CA, LDCA, CH, LDCH, CV, LDCV, CD, LDCD,    &
                   ICOMM, IFAIL)

INTEGER            M, N, LDA, LDCA, LDCH, LDCV, LDCD, ICOMM(180), IFAIL
REAL (KIND=nag_wp) A(LDA,N), CA(LDCA,*), CH(LDCH,*), CV(LDCV,*),            &
                   CD(LDCD,*)
```

3 Description

C09EAF computes the two-dimensional DWT of a given input data array, considered as a matrix A, 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, A, first over columns and then to the result over rows. The matrix of approximation (or smooth) coefficients, C_a, is produced by the low pass filter over columns and rows; the matrix of horizontal coefficients, C_h, is produced by the high pass filter over columns and the low pass filter over rows; the matrix of vertical coefficients, C_v, is produced by the low pass filter over columns and the high pass filter over rows; and the matrix of diagonal coefficients, C_d, is produced by the high pass filter over columns and rows. 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 for C_a, C_h, C_v, and C_d together and the number of columns of each coefficients matrix, n_{cn}, are returned by the initialization routine C09ABF. These values can be used to calculate the number of rows of each coefficients matrix, n_{cm}, using the formula $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(4n_{\mathrm{cn}})$.

4 References

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

5 Parameters

1: M – INTEGER *Input*

On entry: number of rows, m, of data matrix A.

Constraint: this must be the same as the value M passed to the initialization routine C09ABF.

2: N – INTEGER *Input*

On entry: number of columns, n, of data matrix A.

Constraint: this must be the same as the value N passed to the initialization routine C09ABF.

3: $\mathrm{A}(\mathrm{LDA},\mathrm{N})$ – REAL (KIND=nag_wp) array *Input*

On entry: the m by n data matrix A.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which C09EAF is called.

Constraint: LDA $\geq$ M.

5: CA(LDCA, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array CA must be at least n_{cn} where n_{cn} is the parameter NWCN returned by routine C09ABF.

On exit: contains the n_{cm} by n_{cn} matrix of approximation coefficients, C_a.

6: LDCA – INTEGER *Input*

On entry: the first dimension of the array CA as declared in the (sub)program from which C09EAF is called.

Constraint: LDCA $\geq n_{\mathrm{cm}}$ where $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(4n_{\mathrm{cn}})$ and n_{cn}, n_{ct} are returned by the initialization routine C09ABF.

7: CH(LDCH, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array CH must be at least n_{cn} where n_{cn} is the parameter NWCN returned by routine C09ABF.

On exit: contains the n_{cm} by n_{cn} matrix of horizontal coefficients, C_h.

8: LDCH – INTEGER *Input*

On entry: the first dimension of the array CH as declared in the (sub)program from which C09EAF is called.

Constraint: LDCH $\geq n_{\mathrm{cm}}$ where $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(4n_{\mathrm{cn}})$ and n_{cn}, n_{ct} are returned by the initialization routine C09ABF.

9: CV(LDCV, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array CV must be at least n_{cn} where n_{cn} is the parameter NWCN returned by routine C09ABF.

On exit: contains the n_{cm} by n_{cn} matrix of vertical coefficients, C_v.

10: LDCV – INTEGER *Input*

On entry: the first dimension of the array CV as declared in the (sub)program from which C09EAF is called.

Constraint: LDCV $\geq n_{\mathrm{cm}}$ where $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(4n_{\mathrm{cn}})$ and n_{cn}, n_{ct} are returned by the initialization routine C09ABF.

11: CD(LDCD, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array CD must be at least n_{cn} where n_{cn} is the parameter NWCN returned by routine C09ABF.

On exit: contains the n_{cm} by n_{cn} matrix of diagonal coefficients, C_d.

12: LDCD – INTEGER *Input*

On entry: the first dimension of the array CD as declared in the (sub)program from which C09EAF is called.

Constraint: LDCD $\geq n_{\mathrm{cm}}$ where $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(4n_{\mathrm{cn}})$ and n_{cn}, n_{ct} are returned by the initialization routine C09ABF.

13: ICOMM(180) – 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 C09ABF.

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: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{M} = \langle value \rangle$.
Constraint: $\text{M} = \langle value \rangle$, the value of M on initialization (see C09ABF).

On entry, $\text{N} = \langle value \rangle$.
Constraint: $\text{N} = \langle value \rangle$, the value of N on initialization (see C09ABF).

$\text{IFAIL} = 2$

On entry, $\text{LDA} = \langle value \rangle$ and $\text{M} = \langle value \rangle$.
Constraint: $\text{LDA} \geq \text{M}$.

$\text{IFAIL} = 3$

On entry, $\text{LDCA} = \langle value \rangle$.
Constraint: $\text{LDCA} \geq \langle value \rangle$, the number of wavelet coefficients in the first dimension.

On entry, $\text{LDCD} = \langle value \rangle$.
Constraint: $\text{LDCD} \geq \langle value \rangle$, the number of wavelet coefficients in the first dimension.

On entry, $\text{LDCH} = \langle value \rangle$.
Constraint: $\text{LDCH} \geq \langle value \rangle$, the number of wavelet coefficients in the first dimension.

On entry, $\text{LDCV} = \langle value \rangle$.
Constraint: $\text{LDCV} \geq \langle value \rangle$, the number of wavelet coefficients in the first dimension.

$\text{IFAIL} = 6$

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

Either the initialization routine was called with WTRANS = 'M' or ICOMM has been corrupted.

$\text{IFAIL} = -99$

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

C09EAF 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 two-dimensional discrete wavelet decomposition for a 6×6 input matrix using the Daubechies wavelet, WAVNAM = 'DB4', with half point symmetric end extension.

10.1 Program Text

```
    Program c09eafe

!     C09EAF Example Program Text
!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: c09abf, c09eaf, c09ebf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, ldb, ldca, ldcd,      &
                                          ldch, ldcv, m, n, nf, nwcm, nwcn,    &
                                          nwct, nwl
      Character (12)                   :: mode, wavnam, wtrans
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), ca(:,:), cd(:,:),    &
                                          ch(:,:), cv(:,:)
      Integer                          :: icomm(180)
!     .. Executable Statements ..
      Write (nout,*) 'C09EAF Example Program Results'

!     Skip heading in data file
      Read (nin,*)
!     Read problem parameters.
      Read (nin,*) m, n
      Read (nin,*) wavnam, mode
```

```
      Write (nout,99999) wavnam, mode

      lda = m
      ldb = m
      Allocate (a(lda,n),b(ldb,n))

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

      Write (nout,99998) 'Input Data                    A'
      Do i = 1, m
        Write (nout,99997) a(i,1:n)
      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 c09abf(wavnam,wtrans,mode,m,n,nwl,nf,nwct,nwcn,icomm,ifail)
      nwcm = nwct/(4*nwcn)

      Allocate (ca(nwcm,nwcn),cd(nwcm,nwcn),cv(nwcm,nwcn),ch(nwcm,nwcn))
      ldca = nwcm
      ldch = nwcm
      ldcv = nwcm
      ldcd = nwcm

      ifail = 0
      Call c09eaf(m,n,a,lda,ca,ldca,ch,ldch,cv,ldcv,cd,ldcd,icomm,ifail)

      Write (nout,99998) 'Approximation coefficients   CA'
      Do i = 1, nwcm
        Write (nout,99997) ca(i,1:nwcn)
      End Do
      Write (nout,99998) 'Diagonal coefficients        CD'
      Do i = 1, nwcm
        Write (nout,99997) cd(i,1:nwcn)
      End Do
      Write (nout,99998) 'Horizontal coefficients      CH'
      Do i = 1, nwcm
        Write (nout,99997) ch(i,1:nwcn)
      End Do
      Write (nout,99998) 'Vertical coefficients        CV'
      Do i = 1, nwcm
        Write (nout,99997) cv(i,1:nwcn)
      End Do

      ifail = 0
      Call c09ebf(m,n,ca,ldca,ch,ldch,cv,ldcv,cd,ldcd,b,ldb,icomm,ifail)

      Write (nout,99998) 'Reconstruction                B'
      Do i = 1, m
        Write (nout,99997) b(i,1:n)
      End Do

99999 Format (/1X,'DWT ::'/1X,'       Wavelet : ',A/1X,'       End mode: ',A)
99998 Format (/1X,A,' : ')
99997 Format (1X,8(F8.4,1X):)
    End Program c09eafe
```

10.2 Program Data

```
C09EAF Example Program Data
 6, 6              : m,n
 DB4  Half : wavnam, mode
  8.0000   7.0000   3.0000   3.0000   1.0000   1.0000
  4.0000   6.0000   1.0000   5.0000   2.0000   9.0000
  8.0000   1.0000   4.0000   9.0000   3.0000   7.0000
  9.0000   3.0000   8.0000   2.0000   4.0000   3.0000
  1.0000   3.0000   7.0000   1.0000   5.0000   2.0000
  4.0000   3.0000   7.0000   7.0000   6.0000   1.0000
```

10.3 Program Results

```
 C09EAF Example Program Results

 DWT ::
        Wavelet : DB4
        End mode: Half

 Input Data                    A :
   8.0000   7.0000   3.0000   3.0000   1.0000   1.0000
   4.0000   6.0000   1.0000   5.0000   2.0000   9.0000
   8.0000   1.0000   4.0000   9.0000   3.0000   7.0000
   9.0000   3.0000   8.0000   2.0000   4.0000   3.0000
   1.0000   3.0000   7.0000   1.0000   5.0000   2.0000
   4.0000   3.0000   7.0000   7.0000   6.0000   1.0000

 Approximation coefficients   CA :
   6.3591  10.3477   8.0995  10.3210   8.7587   3.5783
  11.5754   6.3762  12.1704   7.4521   8.6977  14.8535
   2.0630   8.4499  15.4726  12.1764   3.8920   2.7112
  10.2143   6.2445  13.8571   8.1060   7.7701  13.2127
   6.3353   8.7805  10.2727  10.0472   6.8614   7.5814
  11.7141  11.1018   5.2923   8.1272  14.5540   2.5729

 Diagonal coefficients        CD :
   0.4777   1.0230  -0.3147   0.0625   0.0831  -1.3316
   1.0689   1.5671  -2.1422   0.5565   1.7593  -2.8097
  -0.9555  -1.9276   0.9195  -0.2228  -0.5125   2.6989
   0.2899   0.4453  -0.5695   0.1541   0.4749  -0.7946
   0.4944   1.4145   0.3488  -0.1187  -0.6212  -1.5177
  -1.3753  -2.5224   1.7581  -0.4316  -1.1835   3.7547

 Horizontal coefficients      CH :
   0.4100  -0.1827   1.5354   0.0784   0.8101  -1.3594
   2.3496  -0.9422   2.3780  -1.0540   2.7743  -2.2648
  -1.2690   0.0152  -6.9338  -1.7435  -1.6917   1.2388
   0.6317  -0.0969   2.3300   0.4637   0.6365  -0.1162
  -0.2343   0.3923   5.5457   2.1818   0.2103  -0.8573
  -1.8880   0.8142  -4.8552   0.0736  -2.7395   3.3590

 Vertical coefficients        CV :
   1.5365   5.9678   3.4309  -1.0585  -5.0275  -4.8492
   0.6779  -0.0294  -5.3274   1.6483   4.8689  -1.8383
  -1.1065  -2.8791   0.1535   0.0982   0.8417   2.8923
  -0.1359  -2.6633  -5.8549   1.8440   6.2403   0.5697
   1.4244   5.2140   1.6410  -0.4669  -3.2369  -4.5757
   1.0288   2.2521   0.0574  -0.1359  -0.5170  -2.6854

 Reconstruction                B :
   8.0000   7.0000   3.0000   3.0000   1.0000   1.0000
   4.0000   6.0000   1.0000   5.0000   2.0000   9.0000
   8.0000   1.0000   4.0000   9.0000   3.0000   7.0000
   9.0000   3.0000   8.0000   2.0000   4.0000   3.0000
   1.0000   3.0000   7.0000   1.0000   5.0000   2.0000
   4.0000   3.0000   7.0000   7.0000   6.0000   1.0000
```