NAG Library Function Document

nag_wfilt_3d (c09acc)

1 Purpose

nag_wfilt_3d (c09acc) returns the details of the chosen three-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 coefficients and the number of wavelet coefficients in the second and third dimensions for the single-level case. This function must be called before any of the three-dimensional transform functions in this chapter.

2 Specification

```
#include <nag.h>
#include <nagc09.h>
void nag_wfilt_3d (Nag_Wavelet wavnam, Nag_WaveletTransform wtrans,
     Nag_WaveletMode mode, Integer m, Integer n, Integer fr, Integer *nwlmax,
     Integer *nf, Integer *nwct, Integer *nwcn, Integer *nwcfr,
     Integer icomm[], NagError *fail)
```

3 Description

Three-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 ($m \times n \times fr$) of data array A, nag_wfilt_3d (c09acc) 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 wavelet coefficients (over all levels in the multi-level DWT case); n_{cn}, the number of coefficients in the second dimension for a single-level DWT; and n_{cfr}, the number of coefficients in the third 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 three-dimensional transform functions in this chapter.

4 References

None.

5 Arguments

1: **wavnam** – Nag_Wavelet *Input*

On entry: the name of the mother wavelet. See the c09 Chapter Introduction for details.

wavnam = Nag_Haar
Haar wavelet.

wavnam = Nag_Daubechies***n***, where ***n*** $= 2, 3, \ldots, 10$
Daubechies wavelet with ***n*** vanishing moments (2***n*** coefficients). For example, **wavnam** = Nag_Daubechies4 is the name for the Daubechies wavelet with 4 vanishing moments (8 coefficients).

wavnam = Nag_Biorthogonal***x_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** = Nag_Biorthogonal1_1 is the name for the Biorthogonal wavelet of order 1.1.

Constraint: **wavnam** = Nag_Haar, Nag_Daubechies2, Nag_Daubechies3, Nag_Daubechies4, Nag_Daubechies5, Nag_Daubechies6, Nag_Daubechies7, Nag_Daubechies8, Nag_Daubechies9, Nag_Daubechies10, Nag_Biorthogonal1_1, Nag_Biorthogonal1_3, Nag_Biorthogonal1_5, Nag_Biorthogonal2_2, Nag_Biorthogonal2_4, Nag_Biorthogonal2_6, Nag_Biorthogonal2_8, Nag_Biorthogonal3_1, Nag_Biorthogonal3_3, Nag_Biorthogonal3_5 or Nag_Biorthogonal3_7.

2: **wtrans** – Nag_WaveletTransform *Input*

On entry: the type of discrete wavelet transform that is to be applied.

wtrans = Nag_SingleLevel
Single-level decomposition or reconstruction by discrete wavelet transform.

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

Constraint: **wtrans** = Nag_SingleLevel or Nag_MultiLevel.

3: **mode** – Nag_WaveletMode *Input*

On entry: the end extension method.

mode = Nag_Periodic
Periodic end extension.

mode = Nag_HalfPointSymmetric
Half-point symmetric end extension.

mode = Nag_WholePointSymmetric
Whole-point symmetric end extension.

mode = Nag_ZeroPadded
Zero end extension.

Constraint: **mode** = Nag_Periodic, Nag_HalfPointSymmetric, Nag_WholePointSymmetric or Nag_ZeroPadded.

4: **m** – Integer *Input*

On entry: the number of elements, m, in the first dimension (number of rows of each two-dimensional frame) of the input data, A.

Constraint: $\mathbf{m} \geq 2$.

5: **n** – Integer *Input*

On entry: the number of elements, n, in the second dimension (number of columns of each two-dimensional frame) of the input data, A.

Constraint: $\mathbf{n} \geq 2$.

6: **fr** – Integer *Input*

On entry: the number of elements, fr, in the third dimension (number of frames) of the input data, A.

Constraint: $\mathbf{fr} \geq 2$.

7: **nwlmax** – Integer * *Output*

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

If **wtrans** = Nag_SingleLevel, **nwlmax** is not set.

8: **nf** – Integer * *Output*

On exit: 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.

9: **nwct** – Integer * *Output*

On exit: the total number of wavelet coefficients, n_{ct}, that will be generated. When **wtrans** = Nag_SingleLevel the number of rows required (i.e., the first dimension of each two-dimensional frame) in each of the output coefficient arrays can be calculated as $n_{\text{cm}} = n_{\text{ct}}/(8 \times n_{\text{cn}} \times n_{\text{cfr}})$. When **wtrans** = Nag_MultiLevel the length of the array used to store all of the coefficient matrices must be at least n_{ct}.

10: **nwcn** – Integer * *Output*

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

11: **nwcfr** – Integer * *Output*

On exit: for a single-level transform (**wtrans** = Nag_SingleLevel), the number of coefficients that would be generated in the third dimension, n_{cfr}, for each coefficient type. For a multi-level transform (**wtrans** = Nag_MultiLevel) this is set to 1.

12: **icomm**[**260**] – Integer *Communication Array*

On exit: 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.

13: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **fr** = ⟨*value*⟩.
Constraint: **fr** ≥ 2.

On entry, **m** = ⟨*value*⟩.
Constraint: **m** ≥ 2.

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 2$.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_wfilt_3d (c09acc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes the three-dimensional multi-level resolution for $8 \times 8 \times 8$ input data by a discrete wavelet transform using the Daubechies wavelet with four vanishing moments (see **wavnam** = Nag_Daubechies4 in nag_wfilt_3d (c09acc)) and zero end extension. 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 approximation coefficients before a reconstruction is performed. This example also demonstrates in general how to access any set of coefficients at any level following a multi-level transform.

10.1 Program Text

```
/* nag_wfilt_3d (c09acc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */
#include <stdio.h>
#include <math.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagc09.h>
#include <nagx02.h>

#define A(I,J,K) a[I-1 + (J-1)* lda + (K-1)* lda * sda]
#define B(I,J,K) b[I-1 + (J-1)* ldb + (K-1)* ldb * sdb]
#define D(I,J,K) d[I-1 + (J-1)* ldd + (K-1)* ldd * sdd]

int main(void)
{
  /* Scalars */
```

```
  Integer exit_status = 0;
  Integer i, j, k, lda, ldb, ldd, lenc, nwlmax, m, n, fr;
  Integer nwcfr, nwcm, nwcn, nwct, nwl, sda, sdb, sdd, nf;
  Integer want_coeffs, want_level;
  double frob, esq, eps;
  /* Arrays */
  char mode[25], wavnam[25];
  double *a = 0, *b = 0, *c = 0, *d = 0;
  Integer *dwtlvfr = 0, *dwtlvm = 0, *dwtlvn = 0;
  Integer icomm[260];
  /* Nag Types */
  Nag_Wavelet wavnamenum;
  Nag_WaveletMode modenum;
  Nag_MatrixType matrix = Nag_GeneralMatrix;
  Nag_OrderType order = Nag_ColMajor;
  Nag_DiagType diag = Nag_NonUnitDiag;
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_wfilt_3d (c09acc) Example Program Results\n\n");
  fflush(stdout);

  /* Skip heading in data file and read problem parameters */
#ifdef _WIN32
  scanf_s("%*[^\n] %" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n]", &m, &n,
          &fr);
#else
  scanf("%*[^\n] %" NAG_IFMT "%" NAG_IFMT "%" NAG_IFMT "%*[^\n]", &m, &n,
        &fr);
#endif
  lda = m;
  sda = n;
  ldb = m;
  sdb = n;
#ifdef _WIN32
  scanf_s("%24s%24s%*[^\n]\n", wavnam, (unsigned)_countof(wavnam), mode,
          (unsigned)_countof(mode));
#else
  scanf("%24s%24s%*[^\n]\n", wavnam, mode);
#endif

  if (!(a = NAG_ALLOC((lda) * (sda) * (fr), double)) ||
      !(b = NAG_ALLOC((ldb) * (sdb) * (fr), double)))
  {
    printf("Allocation failure\n");
    exit_status = 1;
    goto END;
  }

  printf("Parameters read from file :: \n");
  printf(" MLDWT :: Wavelet  : %s\n", wavnam);
  printf("          End mode : %s\n", mode);
  printf("          m  : %4" NAG_IFMT "\n", m);
  printf("          n  : %4" NAG_IFMT "\n", n);
  printf("          fr : %4" NAG_IFMT "\n\n", fr);

  /* nag_enum_name_to_value (x04nac).
   * Converts NAG enum member name to value
   */
  wavnamenum = (Nag_Wavelet) nag_enum_name_to_value(wavnam);
  modenum = (Nag_WaveletMode) nag_enum_name_to_value(mode);

  /* Read data array */
  for (k = 1; k <= fr; k++) {
    for (i = 1; i <= m; i++) {
#ifdef _WIN32
      for (j = 1; j <= n; j++)
        scanf_s("%lf", &A(i, j, k));
#else
      for (j = 1; j <= n; j++)
```

```
        scanf("%lf", &A(i, j, k));
#endif
    }
#ifdef _WIN32
    scanf_s("%*[^\n] ");
#else
    scanf("%*[^\n] ");
#endif
  }

  /* Print out the input data */
  printf("Input Data :\n\n");
  for (k = 1; k <= fr; k++) {
    /* nag_gen_real_mat_print_comp (x04cbc).
     * Prints out a matrix.
     */
    fflush(stdout);
    nag_gen_real_mat_print_comp(order, matrix, diag, m, n, &A(1, 1, k), lda,
                                "%6.2f", " ", Nag_NoLabels, 0,
                                Nag_NoLabels, 0, 80, 0, 0, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_gen_real_mat_print_comp (x04cbc).\n%s\n",
             fail.message);
      exit_status = 2;
      goto END;
    }
    printf("\n");
  }

  /*
   * nag_wfilt_3d (c09acc).
   * Three-dimensional wavelet filter initialization
   */
  nag_wfilt_3d(wavnamenum, Nag_MultiLevel, modenum, m, n, fr, &nwlmax, &nf,
               &nwct, &nwcn, &nwcfr, icomm, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_wfilt_3d (c09acc).\n%s\n", fail.message);
    exit_status = 3;
    goto END;
  }

  /* Transform one less than the max possible number of levels. */
  nwl = nwlmax - 1;
  lenc = nwct;
  if (!(c = NAG_ALLOC((lenc), double)) ||
      !(dwtlvm = NAG_ALLOC((nwl), Integer)) ||
      !(dwtlvn = NAG_ALLOC((nwl), Integer)) ||
      !(dwtlvfr = NAG_ALLOC((nwl), Integer)))
  {
    printf("Allocation failure\n");
    exit_status = 4;
    goto END;
  }

  /* nag_mldwt_3d (c09fcc).
   * Three-dimensional multi-level discrete wavelet transform
   */
  nag_mldwt_3d(m, n, fr, a, lda, sda, lenc, c,
               nwl, dwtlvm, dwtlvn, dwtlvfr, icomm, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_mldwt_3d (c09fcc).\n%s\n", fail.message);
    exit_status = 5;
    goto END;
  }

  /* nag_wfilt_3d (c09acc) returns nwct based on max levels, so recalculate
   * for the number of levels required, nwl.
   */
  nwct = dwtlvm[0] * dwtlvn[0] * dwtlvfr[0];
  for (i = 0; i < nwl; i++)
    nwct += 7 * dwtlvm[i] * dwtlvn[i] * dwtlvfr[i];
```

```
  printf("Number of Levels :         %4" NAG_IFMT "\n", nwl);
  printf("Length of wavelet filter : %4" NAG_IFMT "\n", nf);
  printf("Total number of wavelet coefficients : %4" NAG_IFMT "\n", nwct);
  printf("Number of coefficients in 1st dimension for each level:\n");
  for (i = 0; i < nwl; i++) {
    printf("%4" NAG_IFMT "%s", dwtlvm[i], i + 1 % 8 ? "" : "\n");
  }
  printf("\nNumber of coefficients in 2nd dimension for each level:\n");
  for (i = 0; i < nwl; i++) {
    printf("%4" NAG_IFMT "%s", dwtlvn[i], i + 1 % 8 ? "" : "\n");
  }
  printf("\nNumber of coefficients in 3rd dimension for each level:\n");
  for (i = 0; i < nwl; i++) {
    printf("%4" NAG_IFMT "%s", dwtlvfr[i], i + 1 % 8 ? "" : "\n");
  }
  printf("\n\n");

  /* Select the deepest level. */
  want_level = nwl;
  /* Select the approximation coefficients. */
  want_coeffs = 0;

  /* Use the extraction routine c09afc to retrieve the required
   * coefficients.
   */
  nwcm = dwtlvm[nwl - want_level];
  nwcn = dwtlvn[nwl - want_level];
  nwcfr = dwtlvfr[nwl - want_level];
  ldd = nwcm;
  sdd = nwcn;
  if (!(d = NAG_ALLOC((ldd) * (sdd) * (nwcfr), double)))
  {
    printf("Allocation failure\n");
    exit_status = 6;
    goto END;
  }

  /* nag_wav_3d_coeff_ext (c09fcc).
   * Extract the desired coefficients.
   */
  nag_wav_3d_coeff_ext(want_level, want_coeffs, lenc, c, d, ldd, sdd, icomm,
                       &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_wav_3d_coeff_ext (c09afc).\n%s\n", fail.message);
    exit_status = 7;
    goto END;
  }

  /* Print the details of the level */
  printf("-----------------------------------------------------\n");
  printf("Level : %4" NAG_IFMT "", want_level);
  printf("; output is %4" NAG_IFMT "", nwcm);
  printf(" by %4" NAG_IFMT "", nwcn);
  printf(" by %4" NAG_IFMT "\n", nwcfr);
  printf("-----------------------------------------------------\n\n");

  /* Print out the selected set of coefficients */
  switch (want_coeffs) {
  case 0:
    printf("Approximation coefficients (LLL)\n");
    break;
  case 1:
    printf("Detail coefficients (LLH)\n");
    break;
  case 2:
    printf("Detail coefficients (LHL)\n");
    break;
  case 3:
    printf("Detail coefficients (LHH)\n");
    break;
```

```
  case 4:
    printf("Detail coefficients (HLL)\n");
    break;
  case 5:
    printf("Detail coefficients (HLH)\n");
    break;
  case 6:
    printf("Detail coefficients (HHL)\n");
    break;
  case 7:
    printf("Detail coefficients (HHH)\n");
    break;
  }

  printf("Level %4" NAG_IFMT ":\n", want_level);
  for (k = 1; k <= nwcfr; k++) {
    printf(" Frame %4" NAG_IFMT " :\n", k);
    for (i = 1; i <= nwcm; i++) {
      for (j = 1; j <= nwcn; j++) {
        printf("%9.3f%s", D(i, j, k), j % 8 ? "" : "\n");
      }
      printf("\n");
    }
  }

  /* nag_imldwt_3d (c09fdc).
   * Three-dimensional inverse multi-level discrete wavelet transform
   */
  nag_imldwt_3d(nwl, lenc, c, m, n, fr, b, ldb, sdb, icomm, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_imldwt_3d (c09fdc).\n%s\n", fail.message);
    exit_status = 8;
    goto END;
  }

  /* Check reconstruction matches original */
  eps = 40.0 * (double) (m + n + fr) * nag_machine_precision;

  frob = 0.0;
  for (k = 1; k <= fr; k++) {
    esq = 0.0;
    for (j = 1; j <= n; j++) {
      for (i = 1; i <= m; i++)
        esq = esq + pow(B(i, j, k) - A(i, j, k), 2);
    }
    frob = MAX(frob, sqrt(esq));
  }

  if (frob > eps) {
    printf("\nFail: Frobenius norm of B-A, where A is the original \n"
           "data and B is the reconstrucion, is too large.\n");
  }
  else {
    printf("\nSuccess: the reconstruction matches the original.\n");
  }

END:
  NAG_FREE(a);
  NAG_FREE(b);
  NAG_FREE(c);
  NAG_FREE(d);
  NAG_FREE(dwtlvfr);
  NAG_FREE(dwtlvm);
  NAG_FREE(dwtlvn);
  return exit_status;
}
```

10.2 Program Data

```
nag_wfilt_3d (c09acc) Example Program Data

   8     8     8                                  : m, n, fr

 Nag_Daubechies4
 Nag_ZeroPadded                                   : wavnam, mode

  10.0  31.0   4.0  10.0  13.0  15.0   4.0   6.0
  26.0  24.0   3.0  18.0  17.0  22.0  20.0   5.0
   6.0   5.0   6.0  11.0  22.0  23.0  23.0   1.0
   9.0  15.0  18.0   1.0  30.0  24.0   8.0   1.0
  18.0   4.0  26.0  20.0  31.0  21.0   4.0   6.0
  25.0  23.0  25.0  14.0  13.0   3.0   3.0  29.0
  22.0  29.0   7.0  29.0  13.0  31.0   3.0  12.0
  22.0   3.0  30.0   5.0  10.0   4.0   1.0  19.0 : frame 1

   1.0   2.0  14.0  31.0  19.0  28.0   6.0  15.0
  26.0  25.0  25.0   4.0   5.0  15.0  24.0   5.0
   1.0  29.0   8.0  18.0  22.0  18.0  31.0  23.0
   8.0   4.0  16.0  21.0  14.0   2.0   2.0  21.0
  10.0   3.0  14.0   3.0  25.0  10.0  24.0  15.0
   3.0  16.0  26.0  21.0  16.0  19.0  25.0  27.0
  28.0  29.0   1.0  20.0   3.0  24.0  31.0  28.0
  31.0  28.0  14.0  30.0  13.0  29.0  20.0   4.0 : frame 2

  31.0  26.0  23.0   5.0  22.0   1.0  16.0   8.0
  21.0   1.0  29.0  10.0  23.0  14.0   9.0   3.0
  20.0  10.0  11.0  22.0  26.0  31.0   3.0  21.0
   9.0  24.0  19.0   3.0   4.0   1.0  13.0  29.0
  18.0  16.0   5.0   6.0   9.0  16.0   8.0  16.0
  32.0  19.0  32.0   1.0   6.0   4.0   1.0  17.0
  29.0  29.0   2.0  29.0  27.0  25.0  31.0   6.0
  28.0  15.0  15.0  22.0  18.0   1.0  18.0  14.0 : frame 3

  15.0   9.0   4.0  14.0  26.0  10.0   3.0  28.0
  21.0  24.0  32.0  27.0   1.0  27.0   8.0  16.0
  10.0  27.0  29.0  15.0  13.0   1.0   5.0  16.0
   4.0   1.0   8.0  31.0  14.0   6.0   5.0  27.0
   1.0  19.0  11.0  31.0  12.0  31.0  17.0  26.0
  27.0   1.0  16.0   6.0  18.0   2.0  17.0  17.0
  30.0   9.0  15.0  32.0  32.0  29.0  16.0   2.0
   3.0  11.0  26.0   2.0  23.0   8.0  10.0  31.0 : frame 4

  12.0   7.0   6.0  12.0   1.0  13.0  30.0  26.0
  27.0  27.0  20.0  16.0  30.0  28.0  13.0  30.0
  29.0  15.0  15.0   5.0   1.0  13.0  31.0   2.0
  31.0  21.0  27.0  30.0   8.0   7.0  11.0   3.0
  17.0   4.0   6.0   1.0   9.0  25.0   3.0  15.0
  12.0  18.0  16.0   5.0   9.0  16.0   6.0  13.0
   3.0   5.0  26.0  30.0  19.0  11.0  32.0  24.0
   6.0  16.0   7.0  15.0  31.0  10.0  20.0  14.0 : frame 5

  20.0   7.0  17.0  11.0   4.0  21.0  25.0  17.0
  18.0  22.0  22.0   6.0   1.0   5.0  15.0  17.0
  25.0  24.0  16.0  13.0  19.0  16.0  23.0  10.0
   1.0  31.0   5.0  13.0  11.0  12.0   1.0  18.0
   1.0  27.0   9.0   5.0  29.0  26.0  23.0  13.0
   2.0  17.0  17.0  14.0  31.0  21.0  16.0   5.0
  26.0  21.0  10.0  21.0   9.0  11.0   1.0  15.0
   8.0  15.0  18.0   4.0  16.0   9.0   3.0  29.0 : frame 6

  26.0   2.0  30.0  26.0   7.0   4.0   9.0   1.0
  15.0   2.0  10.0  22.0  16.0  15.0   4.0   3.0
   4.0   7.0  32.0  27.0   7.0   5.0  17.0   4.0
  22.0  30.0   6.0  18.0  32.0   2.0   1.0  31.0
  15.0  19.0  20.0  12.0  10.0  28.0  27.0   3.0
  26.0  31.0  21.0   2.0  27.0  10.0  22.0  13.0
  32.0   3.0  27.0  23.0   1.0  11.0   4.0  26.0
   3.0   1.0  31.0  21.0  27.0  21.0  14.0   9.0 : frame 7
```

```
   2.0  16.0  16.0  23.0  23.0   9.0  27.0  12.0
  15.0  17.0  20.0  27.0   5.0   4.0  18.0  16.0
  29.0  32.0  20.0   8.0  14.0  32.0  11.0   4.0
  28.0   1.0  15.0  19.0  14.0   9.0  30.0  18.0
  20.0   2.0   8.0  11.0  20.0  24.0  14.0   3.0
  18.0  15.0  16.0   3.0  23.0   1.0  19.0  31.0
  32.0  27.0  28.0   9.0  15.0  23.0   9.0  13.0
   1.0  24.0  30.0   4.0  18.0  11.0   1.0  22.0 : frame 8
```

10.3 Program Results

```
nag_wfilt_3d (c09acc) Example Program Results

Parameters read from file ::
 MLDWT :: Wavelet  : Nag_Daubechies4
          End mode : Nag_ZeroPadded
          m  :    8
          n  :    8
          fr :    8

Input Data :

   10.00  31.00   4.00  10.00  13.00  15.00   4.00   6.00
   26.00  24.00   3.00  18.00  17.00  22.00  20.00   5.00
    6.00   5.00   6.00  11.00  22.00  23.00  23.00   1.00
    9.00  15.00  18.00   1.00  30.00  24.00   8.00   1.00
   18.00   4.00  26.00  20.00  31.00  21.00   4.00   6.00
   25.00  23.00  25.00  14.00  13.00   3.00   3.00  29.00
   22.00  29.00   7.00  29.00  13.00  31.00   3.00  12.00
   22.00   3.00  30.00   5.00  10.00   4.00   1.00  19.00

    1.00   2.00  14.00  31.00  19.00  28.00   6.00  15.00
   26.00  25.00  25.00   4.00   5.00  15.00  24.00   5.00
    1.00  29.00   8.00  18.00  22.00  18.00  31.00  23.00
    8.00   4.00  16.00  21.00  14.00   2.00   2.00  21.00
   10.00   3.00  14.00   3.00  25.00  10.00  24.00  15.00
    3.00  16.00  26.00  21.00  16.00  19.00  25.00  27.00
   28.00  29.00   1.00  20.00   3.00  24.00  31.00  28.00
   31.00  28.00  14.00  30.00  13.00  29.00  20.00   4.00

   31.00  26.00  23.00   5.00  22.00   1.00  16.00   8.00
   21.00   1.00  29.00  10.00  23.00  14.00   9.00   3.00
   20.00  10.00  11.00  22.00  26.00  31.00   3.00  21.00
    9.00  24.00  19.00   3.00   4.00   1.00  13.00  29.00
   18.00  16.00   5.00   6.00   9.00  16.00   8.00  16.00
   32.00  19.00  32.00   1.00   6.00   4.00   1.00  17.00
   29.00  29.00   2.00  29.00  27.00  25.00  31.00   6.00
   28.00  15.00  15.00  22.00  18.00   1.00  18.00  14.00

   15.00   9.00   4.00  14.00  26.00  10.00   3.00  28.00
   21.00  24.00  32.00  27.00   1.00  27.00   8.00  16.00
   10.00  27.00  29.00  15.00  13.00   1.00   5.00  16.00
    4.00   1.00   8.00  31.00  14.00   6.00   5.00  27.00
    1.00  19.00  11.00  31.00  12.00  31.00  17.00  26.00
   27.00   1.00  16.00   6.00  18.00   2.00  17.00  17.00
   30.00   9.00  15.00  32.00  32.00  29.00  16.00   2.00
    3.00  11.00  26.00   2.00  23.00   8.00  10.00  31.00

   12.00   7.00   6.00  12.00   1.00  13.00  30.00  26.00
   27.00  27.00  20.00  16.00  30.00  28.00  13.00  30.00
   29.00  15.00  15.00   5.00   1.00  13.00  31.00   2.00
   31.00  21.00  27.00  30.00   8.00   7.00  11.00   3.00
   17.00   4.00   6.00   1.00   9.00  25.00   3.00  15.00
   12.00  18.00  16.00   5.00   9.00  16.00   6.00  13.00
    3.00   5.00  26.00  30.00  19.00  11.00  32.00  24.00
    6.00  16.00   7.00  15.00  31.00  10.00  20.00  14.00

   20.00   7.00  17.00  11.00   4.00  21.00  25.00  17.00
   18.00  22.00  22.00   6.00   1.00   5.00  15.00  17.00
```

```
   25.00  24.00  16.00  13.00  19.00  16.00  23.00  10.00
    1.00  31.00   5.00  13.00  11.00  12.00   1.00  18.00
    1.00  27.00   9.00   5.00  29.00  26.00  23.00  13.00
    2.00  17.00  17.00  14.00  31.00  21.00  16.00   5.00
   26.00  21.00  10.00  21.00   9.00  11.00   1.00  15.00
    8.00  15.00  18.00   4.00  16.00   9.00   3.00  29.00

   26.00   2.00  30.00  26.00   7.00   4.00   9.00   1.00
   15.00   2.00  10.00  22.00  16.00  15.00   4.00   3.00
    4.00   7.00  32.00  27.00   7.00   5.00  17.00   4.00
   22.00  30.00   6.00  18.00  32.00   2.00   1.00  31.00
   15.00  19.00  20.00  12.00  10.00  28.00  27.00   3.00
   26.00  31.00  21.00   2.00  27.00  10.00  22.00  13.00
   32.00   3.00  27.00  23.00   1.00  11.00   4.00  26.00
    3.00   1.00  31.00  21.00  27.00  21.00  14.00   9.00

    2.00  16.00  16.00  23.00  23.00   9.00  27.00  12.00
   15.00  17.00  20.00  27.00   5.00   4.00  18.00  16.00
   29.00  32.00  20.00   8.00  14.00  32.00  11.00   4.00
   28.00   1.00  15.00  19.00  14.00   9.00  30.00  18.00
   20.00   2.00   8.00  11.00  20.00  24.00  14.00   3.00
   18.00  15.00  16.00   3.00  23.00   1.00  19.00  31.00
   32.00  27.00  28.00   9.00  15.00  23.00   9.00  13.00
    1.00  24.00  30.00   4.00  18.00  11.00   1.00  22.00

Number of Levels :            2
Length of wavelet filter :    8
Total number of wavelet coefficients : 5145
Number of coefficients in 1st dimension for each level:
   7   7
Number of coefficients in 2nd dimension for each level:
   7   7
Number of coefficients in 3rd dimension for each level:
   7   7

-------------------------------------------------------
Level :    2; output is    7 by    7 by    7
-------------------------------------------------------

Approximation coefficients (LLL)
Level    2:
 Frame    1 :
   -0.000   -0.000    0.000    0.000    0.000    0.000    0.000
   -0.000   -0.000    0.000   -0.000    0.000   -0.001   -0.000
    0.000    0.000   -0.000   -0.000   -0.002    0.004   -0.000
   -0.000   -0.000   -0.000    0.002    0.003   -0.012    0.001
    0.000   -0.000   -0.002    0.001    0.093    0.116    0.000
    0.000   -0.001    0.001   -0.006    0.158    0.093    0.010
    0.000   -0.000    0.000   -0.001    0.012    0.006    0.001
 Frame    2 :
   -0.000    0.000    0.000   -0.000   -0.001   -0.001   -0.000
    0.000   -0.000    0.000   -0.001    0.003    0.004    0.000
    0.000   -0.000   -0.001    0.003    0.013   -0.006   -0.003
   -0.000    0.000    0.003   -0.007   -0.071    0.007    0.015
   -0.000    0.004   -0.015    0.041   -0.368   -0.343   -0.068
   -0.001    0.000    0.024   -0.087   -0.499   -0.581   -0.067
   -0.000   -0.000    0.005   -0.013   -0.080   -0.073   -0.005
 Frame    3 :
    0.000    0.000   -0.000    0.001    0.001    0.003    0.000
   -0.000    0.000   -0.001    0.004   -0.022    0.001   -0.001
   -0.000   -0.001    0.007   -0.013    0.045   -0.073    0.007
    0.001    0.003   -0.014   -0.001    0.087    0.326   -0.049
    0.001   -0.017    0.069   -0.068    0.591   -0.172    0.394
    0.002    0.012   -0.122    0.419   -0.527    1.229    0.162
    0.000    0.003   -0.018    0.040    0.115    0.282    0.010
 Frame    4 :
   -0.000   -0.000    0.001   -0.003    0.006   -0.010   -0.003
    0.000   -0.001    0.004   -0.011    0.095   -0.018   -0.001
    0.000    0.006   -0.030    0.059   -0.392    0.365    0.013
   -0.002   -0.016    0.068   -0.064    0.537   -1.457    0.030
   -0.007    0.059   -0.149   -0.105   -2.969    0.111   -1.419
```

```
   -0.002   -0.042    0.260   -0.728    2.468   -4.177   -0.512
    0.000   -0.008    0.027   -0.021   -0.122   -0.998   -0.071
 Frame    5 :
    0.000   -0.000   -0.001   -0.001    0.080    0.101    0.014
   -0.001    0.003   -0.002   -0.002   -0.530   -0.571   -0.044
   -0.001   -0.016    0.080   -0.186    0.418    0.493    0.009
    0.010    0.052   -0.414    1.126    0.611   -0.004   -0.129
    0.083   -0.472    0.959   -2.951   84.849   91.369   10.175
    0.160   -0.319   -0.896    1.855  106.190  117.275   12.990
    0.021   -0.021   -0.218    0.496   12.532   12.975    1.342
 Frame    6 :
    0.000   -0.000   -0.001    0.000    0.095    0.134    0.016
   -0.001    0.005   -0.005    0.001   -0.701   -0.367   -0.023
   -0.001   -0.013    0.035   -0.040    1.395   -0.223   -0.140
    0.003    0.017   -0.025   -0.049   -3.242   -0.351    0.328
    0.137   -0.480   -0.144    0.607  105.581  101.777   10.072
    0.136   -0.613    0.874   -2.862  121.107  124.422   13.705
    0.007   -0.094    0.431   -1.415   12.937   13.126    1.602
 Frame    7 :
    0.000   -0.000    0.001   -0.002    0.013    0.016    0.001
   -0.000    0.001    0.000   -0.004   -0.081   -0.038   -0.002
    0.001    0.000   -0.021    0.082    0.085   -0.027   -0.015
   -0.003   -0.007    0.104   -0.349    0.014   -0.131    0.029
    0.018   -0.036   -0.097    0.142   11.444   11.628    0.978
    0.019   -0.076    0.023    0.104   13.727   13.307    1.563
    0.000   -0.016    0.075   -0.204    1.629    1.283    0.155

Success: the reconstruction matches the original.
```