NAG Library Function Document

nag_cwt_1d_real (c09bac)

1 Purpose

nag_cwt_1d_real (c09bac) computes the real, continuous wavelet transform in one dimension.

2 Specification

```
#include <nag.h>
#include <nagc09.h>
```

```
void nag_cwt_1d_real (Nag_Wavelet wavnam, Integer wparam, Integer n,
     const double x[], Integer nscal, const Integer scales[], double c[],
     NagError *fail)
```

3 Description

nag_cwt_1d_real (c09bac) computes the real part of the one-dimensional, continuous wavelet transform

$$C_{s,k} = \int_{\mathbb{R}} x(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-k}{s} \right) dt,$$

of a signal $x(t)$ at scale s and position k, where the signal is sampled discretely at n equidistant points x_i, for $i = 1, 2, \ldots, n$. ψ is the wavelet function, which can be chosen to be the Morlet wavelet, the derivatives of a Gaussian or the Mexican hat wavelet ($*$ denotes the complex conjugate). The integrals of the scaled, shifted wavelet function are approximated and the convolution is then computed.

The mother wavelets supplied for use with this function are defined as follows.

1. The Morlet wavelet (real part) with nondimensional wave number κ is

$$\psi(x) = \frac{1}{\pi^{1/4}} \left(\cos(\kappa x) - e^{-\kappa^2/2} \right) e^{-x^2/2},$$

   where the correction term, $e^{-\kappa^2/2}$ (required to satisfy the admissibility condition) is included.

2. The derivatives of a Gaussian are obtained from

$$\hat{\psi}^{(m)}(x) = \frac{\mathrm{d}^m \left(e^{-x^2} \right)}{\mathrm{d}x^m},$$

   taking $m = 1, \ldots, 8$. These are the Hermite polynomials multiplied by the Gaussian. The sign is then adjusted to give $\hat{\psi}^{(m)}(0) > 0$ when m is even while the sign of the succeeding odd derivative, $\hat{\psi}^{(m+1)}$, is made consistent with the preceding even numbered derivative. They are normalized by the L^2-norm,

$$p_m = \left(\int_{-\infty}^{\infty} \left[\hat{\psi}^{(m)}(x) \right]^2 \mathrm{d}x \right)^{1/2}$$

   The resulting normalized derivatives can be written in terms of the Hermite polynomials, $H_m(x)$, as

$$\psi^{(m)}(x) = \frac{\alpha H_m(x) e^{-x^2}}{p_m},$$

   where

$$\alpha = \begin{cases} 1, & \text{when } m = 0, 3 \bmod 4; \\ -1, & \text{when } m = 1, 2 \bmod 4. \end{cases}$$

Thus, the derivatives of a Gaussian provided here are,

$$\psi^{(1)}(x) = -\left(\frac{2}{\pi}\right)^{1/4} 2xe^{-x^2},$$

$$\psi^{(2)}(x) = -\left(\frac{2}{\pi}\right)^{1/4} \frac{1}{\sqrt{3}}\left(4x^2 - 2\right)e^{-x^2},$$

$$\psi^{(3)}(x) = \left(\frac{2}{\pi}\right)^{1/4} \frac{1}{\sqrt{15}}\left(8x^3 - 12x\right)e^{-x^2},$$

$$\psi^{(4)}(x) = \left(\frac{2}{\pi}\right)^{1/4} \frac{1}{\sqrt{105}}\left(16x^4 - 48x^2 + 12\right)e^{-x^2},$$

$$\psi^{(5)}(x) = -\left(\frac{2}{\pi}\right)^{1/4} \frac{1}{3\sqrt{105}}\left(32x^5 - 160x^3 + 120x\right)e^{-x^2},$$

$$\psi^{(6)}(x) = -\left(\frac{2}{\pi}\right)^{1/4} \frac{1}{3\sqrt{1155}}\left(64x^6 - 480x^4 + 720x^2 - 120\right)e^{-x^2},$$

$$\psi^{(7)}(x) = \left(\frac{2}{\pi}\right)^{1/4} \frac{1}{3\sqrt{15015}}\left(128x^7 - 1344x^5 + 3360x^3 - 1680x\right)e^{-x^2},$$

$$\psi^{(8)}(x) = \left(\frac{2}{\pi}\right)^{1/4} \frac{1}{45\sqrt{1001}}\left(256x^8 - 3584x^6 + 13440x^4 - 13440x^2 + 1680\right)e^{-x^2}.$$

3. The second derivative of a Gaussian is known as the Mexican hat wavelet and is supplied as an additional function in the form

$$\psi(x) = \frac{2}{\left(\sqrt{3}\pi^{1/4}\right)}\left(1 - x^2\right)e^{-x^2/2}.$$

The remaining normalized derivatives of a Gaussian can be expressed as multiples of the exponential $e^{-t^2/2}$ by applying the substitution $x = t/\sqrt{2}$ followed by multiplication with the scaling factor, $1/\sqrt[4]{2}$.

4 References

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

5 Arguments

1: **wavnam** – Nag_Wavelet *Input*

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

wavnam = Nag_Morlet
Morlet wavelet.

wavnam = Nag_DGauss
Derivative of a Gaussian wavelet.

wavnam = Nag_MexHat
Mexican hat wavelet.

Constraint: **wavnam** = Nag_Morlet, Nag_DGauss or Nag_MexHat.

2: **wparam** – Integer *Input*

On entry: the nondimensional wave number for the Morlet wavelet or the order of the derivative for the Gaussian wavelet. It is not referenced when **wavnam** $=$ Nag_MexHat.

Constraints:

if **wavnam** $=$ Nag_Morlet, $5 \leq$ **wparam** ≤ 20;
if **wavnam** $=$ Nag_DGauss, $1 \leq$ **wparam** ≤ 8.

3: **n** – Integer *Input*

On entry: the size, n, of the input dataset x.

Constraint: **n** ≥ 2.

4: **x**[**n**] – const double *Input*

On entry: **x** contains the input dataset **x**$[j-1] = x_j$, for $j = 1, 2, \ldots, n$.

5: **nscal** – Integer *Input*

On entry: the number of scales to be computed.

Constraint: **nscal** ≥ 1.

6: **scales**[**nscal**] – const Integer *Input*

On entry: the scales at which the transform is to be computed.

Constraint: **scales**$[i-1] \geq 1$, for $i = 1, 2, \ldots,$ **nscal**.

7: **c**[**nscal** $\times$ **n**] – double *Output*

Note: the (i,j)th element of the matrix C is stored in **c**$[(j-1) \times$ **nscal** $+ i - 1]$.

On exit: the transform coefficients at the requested scales, where **c**$[(j-1) \times$ **nscal** $+ i - 1]$ is the transform coefficient $C_{i,j}$ at scale i and position j.

8: **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 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument $\langle$*value*$\rangle$ had an illegal value.

NE_INT

On entry, **n** $= \langle$*value*$\rangle$.
Constraint: **n** ≥ 2.

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

On entry, **wavnam** $=$ Nag_DGauss and **wparam** $= \langle$*value*$\rangle$.
Constraint: if **wavnam** $=$ Nag_DGauss, $1 \leq$ **wparam** ≤ 8.

On entry, **wavnam** $=$ Nag_Morlet and **wparam** $= \langle value \rangle$.
Constraint: if **wavnam** $=$ Nag_Morlet, $5 \leq$ **wparam** ≤ 20.

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 2.7.6 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 2.7.5 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The accuracy of nag_cwt_1d_real (c09bac) is determined by the fact that the convolution must be computed as a discrete approximation to the continuous form. The input signal, x, is taken to be piecewise constant using the supplied discrete values.

8 Parallelism and Performance

nag_cwt_1d_real (c09bac) is not threaded in any implementation.

9 Further Comments

Workspace is internally allocated by nag_cwt_1d_real (c09bac). The total size of these arrays is $2^{13} + (\mathbf{n} + n_k - 1)$ double elements and n_k Integer elements, where $n_k = k \times \max(\mathbf{scales}[i-1])$ and $k = 17$ when **wavnam** $=$ Nag_Morlet or Nag_DGauss and $k = 11$ when **wavnam** $=$ Nag_MexHat.

10 Example

This example computes the continuous wavelet transform of a dataset containing a single nonzero value representing an impulse. The Morlet wavelet is used with wave number $\kappa = 5$ and scales 1, 2, 3, 4.

10.1 Program Text

```
/* nag_cwt_1d_real (c09bac) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */
#include <nag.h>
#include <nag_stdlib.h>
#include <nagc09.h>
#include <nagx04.h>

int main(void)
{
  /* Scalars */
  Integer exit_status = 0;
  Integer i, n, nscal, wparam;
  /* Arrays */
  Integer *scales = 0;
  double *c = 0, *x = 0;
  char *clabs = 0, **clabsc = 0;
  char wavnam[20];
  /* NAG types */
  NagError fail;
  Nag_Wavelet wavnamenum;

  INIT_FAIL(fail);

  printf("nag_cwt_1d_real (c09bac) Example Program Results\n\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  /* Read problem parameters */
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%" NAG_IFMT "", &n, &nscal);
#else
  scanf("%" NAG_IFMT "%" NAG_IFMT "", &n, &nscal);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  if (!(c = NAG_ALLOC((nscal) * (n), double)) ||
      !(scales = NAG_ALLOC((nscal), Integer)) ||
      !(x = NAG_ALLOC((n), double)) ||
      !(clabs = NAG_ALLOC(10 * 10, char)) || !(clabsc = NAG_ALLOC(10, char *))
         )
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }
#ifdef _WIN32
  scanf_s("%19s", wavnam, (unsigned)_countof(wavnam));
#else
  scanf("%19s", wavnam);
#endif
  /* nag_enum_name_to_value (x04nac).
   * Converts NAG enum member name to value
   */
  wavnamenum = (Nag_Wavelet) nag_enum_name_to_value(wavnam);
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%*[^\n]", &wparam);
#else
  scanf("%" NAG_IFMT "%*[^\n]", &wparam);
#endif
  printf("Parameters read from file ::\n");
  printf("    Wavelet : %20s,  wparam : %" NAG_IFMT "\n", wavnam, wparam);
  printf("          n : %20" NAG_IFMT ",   nscal : %" NAG_IFMT "\n", n,
         nscal);

  /* Read data array and write it out */
#ifdef _WIN32
  for (i = 0; i < nscal; i++)
    scanf_s("%" NAG_IFMT "", &scales[i]);
#else
  for (i = 0; i < nscal; i++)
    scanf("%" NAG_IFMT "", &scales[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
#ifdef _WIN32
  for (i = 0; i < n; i++)
    scanf_s("%lf", &x[i]);
#else
  for (i = 0; i < n; i++)
```

```
    scanf("%lf", &x[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
  printf("Input Data ::\n");
  printf("     Scales :");
  for (i = 0; i < nscal; i++)
    printf("%9" NAG_IFMT "", scales[i]);

  printf("\n          x :");
  for (i = 0; i < n; i++)
    printf("%9.3f%s", x[i], (i + 1) % 5 ? "" : "\n             ");
  printf("\n");
  /* nag_cwt_1d_real (c09bac).
   * One-dimensional real continuous wavelet transform
   */
  nag_cwt_1d_real(wavnamenum, wparam, n, x, nscal, scales, c, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_cwt_1d_real (c09bac).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

  printf("Number of Scales :%9" NAG_IFMT "\n\n", nscal);
  fflush(stdout);

  /* print coefficients for each scale as columns of a matrix. This requires
   * printing c as a row major matrix with principal dimension nscal using
   * nag_gen_real_mat_print_comp (x04cbc).
   */
  for (i = 0; i < nscal; i++)
#ifdef _WIN32
    sprintf_s(&clabs[10 * i], 10, "scale %3" NAG_IFMT "", scales[i]);
#else
    sprintf(&clabs[10 * i], "scale %3" NAG_IFMT "", scales[i]);
#endif
  for (i = 0; i < nscal; i++)
    clabsc[i] = &clabs[i * 10];
  nag_gen_real_mat_print_comp(Nag_RowMajor, Nag_GeneralMatrix,
                              Nag_NonUnitDiag, n, nscal, c, nscal, "%13.4e",
                              "Wavelet coefficients C ::", Nag_NoLabels, 0,
                              Nag_CharacterLabels, (const char **) clabsc, 80,
                              0, NULL, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_gen_real_mat_print_comp (x04cbc).\n%s\n",
           fail.message);
    exit_status = 6;
    goto END;
  }

END:
  NAG_FREE(c);
  NAG_FREE(x);
  NAG_FREE(scales);
  NAG_FREE(clabs);
  NAG_FREE(clabsc);
  return exit_status;
}
```

10.2 Program Data

```
nag_cwt_1d_real (c09bac) Example Program Data
 10        4            : n, nscal
 Nag_Morlet     5       : wavnam, wparam
 1    2    3    4       : scales(1:nscal)
 0.0  0.0  0.0  0.0  1.0
 0.0  0.0  0.0  0.0  0.0 : x[]
```

10.3 Program Results

```
nag_cwt_1d_real (c09bac) Example Program Results

Parameters read from file ::
    Wavelet :           Nag_Morlet,  wparam : 5
          n :                   10,   nscal : 4
Input Data ::
     Scales :        1        2        3        4
          x :    0.000    0.000    0.000    0.000    1.000
                 0.000    0.000    0.000    0.000    0.000

Number of Scales :        4

 Wavelet coefficients C ::
      scale   1     scale   2     scale   3     scale   4
    -1.7651e-05    1.5012e-04    5.2331e-02    1.4454e-01
    -1.3643e-03   -5.8141e-02    1.7057e-01   -8.4364e-02
     4.6511e-03    1.8442e-01   -1.4891e-01   -2.8870e-01
     8.9294e-02   -2.6380e-01   -2.6822e-01   -9.4993e-02
    -9.2563e-02    1.3289e-01    2.5680e-01    2.8293e-01
    -9.2563e-02    1.3289e-01    2.5680e-01    2.8293e-01
     8.9294e-02   -2.6380e-01   -2.6822e-01   -9.4993e-02
     4.6511e-03    1.8442e-01   -1.4891e-01   -2.8870e-01
    -1.3643e-03   -5.8141e-02    1.7057e-01   -8.4364e-02
    -1.7651e-05    1.5012e-04    5.2331e-02    1.4454e-01
```