NAG Library Function Document

nag_1d_cheb_eval2 (e02akc)

1 Purpose

nag_1d_cheb_eval2 (e02akc) evaluates a polynomial from its Chebyshev series representation, allowing an arbitrary index increment for accessing the array of coefficients.

2 Specification

```
#include <nag.h>
#include <nage02.h>
void nag_1d_cheb_eval2 (Integer n, double xmin, double xmax,
     const double a[], Integer ia1, double x, double *result, NagError *fail)
```

3 Description

If supplied with the coefficients a_i, for $i = 0, 1, \ldots, n$, of a polynomial $p(\bar{x})$ of degree n, where

$$p(\bar{x}) = \tfrac{1}{2}a_0 + a_1 T_1(\bar{x}) + \cdots + a_n T_n(\bar{x}),$$

nag_1d_cheb_eval2 (e02akc) returns the value of $p(\bar{x})$ at a user-specified value of the variable x. Here $T_j(\bar{x})$ denotes the Chebyshev polynomial of the first kind of degree j with argument $\bar{x}$. It is assumed that the independent variable $\bar{x}$ in the interval $[-1, +1]$ was obtained from your original variable x in the interval $[x_{\min}, x_{\max}]$ by the linear transformation

$$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}}.$$

The coefficients a_i may be supplied in the array **a**, with any increment between the indices of array elements which contain successive coefficients. This enables the function to be used in surface fitting and other applications, in which the array might have two or more dimensions.

The method employed is based on the three-term recurrence relation due to Clenshaw (see Clenshaw (1955)), with modifications due to Reinsch and Gentleman (see Gentleman (1969)). For further details of the algorithm and its use see Cox (1973) and Cox and Hayes (1973).

4 References

Clenshaw C W (1955) A note on the summation of Chebyshev series *Math. Tables Aids Comput.* **9** 118–120

Cox M G (1973) A data-fitting package for the non-specialist user *NPL Report NAC 40* National Physical Laboratory

Cox M G and Hayes J G (1973) Curve fitting: a guide and suite of algorithms for the non-specialist user *NPL Report NAC26* National Physical Laboratory

Gentleman W M (1969) An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients *Comput. J.* **12** 160–165

5 Arguments

1: **n** – Integer *Input*

On entry: n, the degree of the given polynomial $p(\bar{x})$.

Constraint: $\mathbf{n} \geq 0$.

2: **xmin** – double *Input*
3: **xmax** – double *Input*

On entry: the lower and upper end points respectively of the interval $[x_{\min}, x_{\max}]$. The Chebyshev series representation is in terms of the normalized variable $\bar{x}$, where

$$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}}.$$

Constraint: **xmin** $<$ **xmax**.

4: **a**$[dim]$ – const double *Input*

Note: the dimension, *dim*, of the array **a** must be at least $((\mathbf{n} + 1 - 1) \times \mathbf{ia1} + 1)$.

On entry: the Chebyshev coefficients of the polynomial $p(\bar{x})$. Specifically, element $i \times \mathbf{ia1}$ must contain the coefficient a_i, for $i = 0, 1, \ldots, n$. Only these $n + 1$ elements will be accessed.

5: **ia1** – Integer *Input*

On entry: the index increment of **a**. Most frequently, the Chebyshev coefficients are stored in adjacent elements of **a**, and **ia1** must be set to 1. However, if, for example, they are stored in **a**$[0]$, **a**$[3]$, **a**$[6]$, $\ldots$, then the value of **ia1** must be 3.

Constraint: **ia1** ≥ 1.

6: **x** – double *Input*

On entry: the argument x at which the polynomial is to be evaluated.

Constraint: **xmin** $\leq$ **x** $\leq$ **xmax**.

7: **result** – double * *Output*

On exit: the value of the polynomial $p(\bar{x})$.

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

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

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

NE_INT

On entry, **ia1** $= \langle value \rangle$.
Constraint: **ia1** ≥ 1.

On entry, **n** $+ 1 = \langle value \rangle$.
Constraint: **n** $+ 1 \geq 1$.

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 0.

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 the Essential Introduction 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 the Essential Introduction for further information.

NE_REAL_2

On entry, **xmax** $= \langle value \rangle$ and **xmin** $= \langle value \rangle$.
Constraint: **xmax** $>$ **xmin**.

NE_REAL_3

On entry, **x** does not lie in [**xmin**, **xmax**]: **x** $= \langle value \rangle$, **xmin** $= \langle value \rangle$ and **xmax** $= \langle value \rangle$.

7 Accuracy

The rounding errors are such that the computed value of the polynomial is exact for a slightly perturbed set of coefficients $a_i + \delta a_i$. The ratio of the sum of the absolute values of the δa_i to the sum of the absolute values of the a_i is less than a small multiple of $(n+1) \times$ ***machine precision***.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken is approximately proportional to $n + 1$.

10 Example

Suppose a polynomial has been computed in Chebyshev series form to fit data over the interval $[-0.5, 2.5]$. The following program evaluates the polynomial at 4 equally spaced points over the interval. (For the purposes of this example, **xmin**, **xmax** and the Chebyshev coefficients are supplied . Normally a program would first read in or generate data and compute the fitted polynomial.)

10.1 Program Text

```
/* nag_1d_cheb_eval2 (e02akc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nage02.h>

int main(void)
{
  /* Initialized data */
  const double xmin = -0.5;
  const double xmax = 2.5;
  const double a[7] = { 2.53213, 1.13032, 0.2715, 0.04434, 0.00547, 5.4e-4,
                        4e-5 };

  /* Scalars */
  double       p, x;
  Integer      exit_status, i, m, n, one;
```

```
  NagError     fail;

  INIT_FAIL(fail);

  exit_status = 0;
  printf("nag_1d_cheb_eval2 (e02akc) Example Program Results\n");

  n = 6;
  one = 1;

  printf("\n");
  printf("   i   Argument  Value of polynomial\n");

  m = 4;
  for (i = 1; i <= m; ++i)
    {
      x = (xmin * (double)(m - i) + xmax * (double)(i - 1)) /
          (double)(m - 1);
      /* nag_1d_cheb_eval2 (e02akc).
       * Evaluation of fitted polynomial in one variable from
       * Chebyshev series form
       */
      nag_1d_cheb_eval2(n, xmin, xmax, a, one, x, &p, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_1d_cheb_eval2 (e02akc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%4"NAG_IFMT"%10.4f    %9.4f\n", i, x, p);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_1d_cheb_eval2 (e02akc) Example Program Results

   i   Argument  Value of polynomial
   1   -0.5000       0.3679
   2    0.5000       0.7165
   3    1.5000       1.3956
   4    2.5000       2.7183
```

Example Program
Evaluation of Chebyshev Representation of Polynomial