NAG Toolbox

nag_fit_1dcheb_glp (e02af)

1 Purpose

nag_fit_1dcheb_glp (e02af) computes the coefficients of a polynomial, in its Chebyshev series form, which interpolates (passes exactly through) data at a special set of points. Least squares polynomial approximations can also be obtained.

2 Syntax

```
[a, ifail] = nag_fit_1dcheb_glp(f, 'nplus1', nplus1)
[a, ifail] = e02af(f, 'nplus1', nplus1)
```

3 Description

nag_fit_1dcheb_glp (e02af) computes the coefficients a_j, for $j = 1, 2, \ldots, n + 1$, in the Chebyshev series

$$\tfrac{1}{2}a_1T_0(\bar{x}) + a_2T_1(\bar{x}) + a_3T_2(\bar{x}) + \cdots + a_{n+1}T_n(\bar{x}),$$

which interpolates the data f_r at the points

$$\bar{x}_r = \cos((r-1)\pi/n), \quad r = 1, 2, \ldots, n+1.$$

Here $T_j(\bar{x})$ denotes the Chebyshev polynomial of the first kind of degree j with argument $\bar{x}$. The use of these points minimizes the risk of unwanted fluctuations in the polynomial and is recommended when the data abscissae can be chosen by you, e.g., when the data is given as a graph. For further advantages of this choice of points, see Clenshaw (1962).

In terms of your original variables, x say, the values of x at which the data f_r are to be provided are

$$x_r = \tfrac{1}{2}(x_{\max} - x_{\min})\cos(\pi(r-1)/n) + \tfrac{1}{2}(x_{\max} + x_{\min}), \quad r = 1, 2, \ldots, n+1$$

where $x_{\max}$ and $x_{\min}$ are respectively the upper and lower ends of the range of x over which you wish to interpolate.

Truncation of the resulting series after the term involving a_{i+1}, say, yields a least squares approximation to the data. This approximation, $p(\bar{x})$, say, is the polynomial of degree i which minimizes

$$\tfrac{1}{2}\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \cdots + \epsilon_n^2 + \tfrac{1}{2}\epsilon_{n+1}^2,$$

where the residual $\epsilon_r = p(\bar{x}_r) - f_r$, for $r = 1, 2, \ldots, n+1$.

The method employed is based on the application of the three-term recurrence relation due to Clenshaw (1955) for the evaluation of the defining expression for the Chebyshev coefficients (see, for example, Clenshaw (1962)). The modifications to this recurrence relation suggested by Reinsch and Gentleman (see Gentleman (1969)) are used to give greater numerical stability.

For further details of the algorithm and its use see Cox (1974) and Cox and Hayes (1973).

Subsequent evaluation of the computed polynomial, perhaps truncated after an appropriate number of terms, should be carried out using nag_fit_1dcheb_eval (e02ae).

4 References

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

Clenshaw C W (1962) Chebyshev Series for Mathematical Functions *Mathematical tables* HMSO

Cox M G (1974) A data-fitting package for the non-specialist user *Software for Numerical Mathematics* (ed D J Evans) Academic Press

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 Parameters

5.1 Compulsory Input Parameters

1: **f**(**nplus1**) – REAL (KIND=nag_wp) array

For $r = 1, 2, \ldots, n + 1$, $\mathbf{f}(r)$ must contain f_r the value of the dependent variable (ordinate) corresponding to the value

$$\bar{x}_r = \cos(\pi(r - 1)/n)$$

of the independent variable (abscissa) $\bar{x}$, or equivalently to the value

$$x(r) = \tfrac{1}{2}(x_{\mathrm{max}} - x_{\mathrm{min}}) \cos(\pi(r - 1)/n) + \tfrac{1}{2}(x_{\mathrm{max}} + x_{\mathrm{min}})$$

of your original variable x. Here x_{max} and x_{min} are respectively the upper and lower ends of the range over which you wish to interpolate.

5.2 Optional Input Parameters

1: **nplus1** – INTEGER

Default: the dimension of the array **f**.

The number $n + 1$ of data points (one greater than the degree n of the interpolating polynomial).

Constraint: **nplus1** ≥ 2.

5.3 Output Parameters

1: **a**(**nplus1**) – REAL (KIND=nag_wp) array

a(j) is the coefficient a_j in the interpolating polynomial, for $j = 1, 2, \ldots, n + 1$.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **nplus1** < 2.

ifail $= -99$

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

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The rounding errors committed are such that the computed coefficients are exact for a slightly perturbed set of ordinates $f_r + \delta f_r$. The ratio of the sum of the absolute values of the δf_r to the sum of the absolute values of the f_r is less than a small multiple of $(n+1)\epsilon$, where ϵ is the ***machine precision***.

8 Further Comments

The time taken is approximately proportional to $(n+1)^2 + 30$.

For choice of degree when using the function for least squares approximation, see Section 3.2 in the E02 Chapter Introduction.

9 Example

Determine the Chebyshev coefficients of the polynomial which interpolates the data $\bar{x}_r, f_r$, for $r = 1, 2, \ldots, 11$, where $\bar{x}_r = \cos(\pi \times (r-1)/10)$ and $f_r = e^{\bar{x}_r}$. Evaluate, for comparison with the values of f_r, the resulting Chebyshev series at $\bar{x}_r$, for $r = 1, 2, \ldots, 11$.

The example program supplied is written in a general form that will enable polynomial interpolations of arbitrary data at the cosine points $\cos(\pi \times (r-1)/n)$, for $r = 1, 2, \ldots, n+1$, to be obtained for any n ($=$ **nplus1** $- 1$). Note that nag_fit_1dcheb_eval (e02ae) is used to evaluate the interpolating polynomial. The program is self-starting in that any number of datasets can be supplied.

9.1 Program Text

```
function e02af_example

fprintf('e02af example results\n\n');

x = -cos([0:10]*pi/10);
f = exp(-x);

[a, ifail] = e02af(f);

disp('Chebyshev coefficients:');
fprintf('%11.7f\n',a);

for i=1:11
  [p(i), ifail] = e02ae(a, -x(i));
end
fprintf('\n');
disp('      x        p(x)');
disp([x' p']);
```

9.2 Program Results

```
e02af example results

Chebyshev coefficients:
  2.5321318
  1.1303182
  0.2714953
  0.0443368
  0.0054742
  0.0005429
  0.0000450
  0.0000032
  0.0000002
  0.0000000
  0.0000000

      x        p(x)
   -1.0000    2.7183
   -0.9511    2.5884
   -0.8090    2.2457
```

```
  -0.5878    1.8000
  -0.3090    1.3621
  -0.0000    1.0000
   0.3090    0.7342
   0.5878    0.5556
   0.8090    0.4453
   0.9511    0.3863
   1.0000    0.3679
```