NAG Toolbox

nag_fit_1dcheb_arb (e02ad)

1 Purpose

nag_fit_1dcheb_arb (e02ad) computes weighted least squares polynomial approximations to an arbitrary set of data points.

2 Syntax

```
[a, s, ifail] = nag_fit_1dcheb_arb(kplus1, x, y, w, 'm', m)
[a, s, ifail] = e02ad(kplus1, x, y, w, 'm', m)
```

3 Description

nag_fit_1dcheb_arb (e02ad) determines least squares polynomial approximations of degrees $0, 1, \ldots, k$ to the set of data points (x_r, y_r) with weights w_r, for $r = 1, 2, \ldots, m$.

The approximation of degree i has the property that it minimizes σ_i the sum of squares of the weighted residuals ϵ_r, where

$$\epsilon_r = w_r(y_r - f_r)$$

and f_r is the value of the polynomial of degree i at the rth data point.

Each polynomial is represented in Chebyshev series form with normalized argument $\bar{x}$. This argument lies in the range -1 to $+1$ and is related to the original variable x by the linear transformation

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

Here $x_{\max}$ and $x_{\min}$ are respectively the largest and smallest values of x_r. The polynomial approximation of degree i is represented as

$$\tfrac{1}{2}a_{i+1,1}T_0(\bar{x}) + a_{i+1,2}T_1(\bar{x}) + a_{i+1,3}T_2(\bar{x}) + \cdots + a_{i+1,i+1}T_i(\bar{x}),$$

where $T_j(\bar{x})$, for $j = 0, 1, \ldots, i$, are the Chebyshev polynomials of the first kind of degree j with argument $(\bar{x})$.

For $i = 0, 1, \ldots, k$, the function produces the values of $a_{i+1,j+1}$, for $j = 0, 1, \ldots, i$, together with the value of the root-mean-square residual $s_i = \sqrt{\sigma_i/(m - i - 1)}$. In the case $m = i + 1$ the function sets the value of s_i to zero.

The method employed is due to Forsythe (1957) and is based on the generation of a set of polynomials orthogonal with respect to summation over the normalized dataset. The extensions due to Clenshaw (1960) to represent these polynomials as well as the approximating polynomials in their Chebyshev series forms are incorporated. The modifications suggested by Reinsch and Gentleman (see Gentleman (1969)) to the method originally employed by Clenshaw for evaluating the orthogonal polynomials from their Chebyshev series representations 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 Chebyshev series representations of the polynomial approximations should be carried out using nag_fit_1dcheb_eval (e02ae).

4 References

Clenshaw C W (1960) Curve fitting with a digital computer *Comput. J.* **2** 170–173

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

Forsythe G E (1957) Generation and use of orthogonal polynomials for data fitting with a digital computer *J. Soc. Indust. Appl. Math.* **5** 74–88

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

Hayes J G (ed.) (1970) *Numerical Approximation to Functions and Data* Athlone Press, London

5 Parameters

5.1 Compulsory Input Parameters

1: **kplus1** – INTEGER

$k+1$, where k is the maximum degree required.

Constraint: $0 < \mathbf{kplus1} \leq mdist$, where $mdist$ is the number of distinct x values in the data.

2: **x**(**m**) – REAL (KIND=nag_wp) array

The values x_r of the independent variable, for $r = 1, 2, \ldots, m$.

Constraint: the values must be supplied in nondecreasing order with $\mathbf{x}(\mathbf{m}) > \mathbf{x}(1)$.

3: **y**(**m**) – REAL (KIND=nag_wp) array

The values y_r of the dependent variable, for $r = 1, 2, \ldots, m$.

4: **w**(**m**) – REAL (KIND=nag_wp) array

The set of weights, w_r, for $r = 1, 2, \ldots, m$. For advice on the choice of weights, see Section 2.1.2 in the E02 Chapter Introduction.

Constraint: $\mathbf{w}(r) > 0.0$, for $r = 1, 2, \ldots, m$.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the dimension of the arrays **x**, **y**, **w**. (An error is raised if these dimensions are not equal.)

The number m of data points.

Constraint: $\mathbf{m} \geq mdist \geq 2$, where $mdist$ is the number of distinct x values in the data.

5.3 Output Parameters

1: **a**(lda, **kplus1**) – REAL (KIND=nag_wp) array

The coefficients of $T_j(\bar{x})$ in the approximating polynomial of degree i. $\mathbf{a}(i+1, j+1)$ contains the coefficient $a_{i+1,j+1}$, for $i = 0, 1, \ldots, k$ and $j = 0, 1, \ldots, i$.

2: **s**(**kplus1**) – REAL (KIND=nag_wp) array

s$(i+1)$ contains the root-mean-square residual s_i, for $i = 0, 1, \ldots, k$, as described in Section 3. For the interpretation of the values of the s_i and their use in selecting an appropriate degree, see Section 3.1 in the E02 Chapter Introduction.

3: **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 $= 1$

The weights are not all strictly positive.

ifail $= 2$

The values of **x**(r), for $r = 1, 2, \ldots,$ **m**, are not in nondecreasing order.

ifail $= 3$

All **x**(r) have the same value: thus the normalization of **x** is not possible.

ifail $= 4$

On entry, **kplus1** < 1 (so the maximum degree required is negative)
or **kplus1** $> mdist$, where $mdist$ is the number of distinct x values in the data (so there cannot be a unique solution for degree $k =$ **kplus1** $- 1$).

ifail $= 5$

$lda <$ **kplus1**.

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

No error analysis for the method has been published. Practical experience with the method, however, is generally extremely satisfactory.

8 Further Comments

The time taken is approximately proportional to $m(k+1)(k+11)$.

The approximating polynomials may exhibit undesirable oscillations (particularly near the ends of the range) if the maximum degree k exceeds a critical value which depends on the number of data points m and their relative positions. As a rough guide, for equally-spaced data, this critical value is about $2 \times \sqrt{m}$. For further details see page 60 of Hayes (1970).

9 Example

Determine weighted least squares polynomial approximations of degrees 0, 1, 2 and 3 to a set of 11 prescribed data points. For the approximation of degree 3, tabulate the data and the corresponding values of the approximating polynomial, together with the residual errors, and also the values of the approximating polynomial at points half-way between each pair of adjacent data points.

The example program supplied is written in a general form that will enable polynomial approximations of degrees $0, 1, \ldots, k$ to be obtained to m data points, with arbitrary positive weights, and the approximation of degree k to be tabulated. nag_fit_1dcheb_eval (e02ae) is used to evaluate the approximating polynomial. The program is self-starting in that any number of datasets can be supplied.

9.1 Program Text

```
function e02ad_example

fprintf('e02ad example results\n\n');

kplus1 = nag_int(4);
x = [ 1.0  2.1   3.1   3.9   4.9   5.8   6.5   7.1   7.8   8.4   9.0];
y = [10.4  7.9   4.7   2.5   1.2   2.2   5.1   9.2  16.1  24.5  35.3];
w = [ 1.0  1.0   1.0   1.0   1.0   0.8   0.8   0.7   0.5   0.3   0.2];

[a, s, ifail] = e02ad(kplus1, x, y, w);

fprintf('Degree  residual   Cheyshev coefficients\n');
for degree=0:kplus1-1;
  i = degree + 1;
  fprintf('%4d%12.2e',degree,s(i));
  fprintf('%10.4f',a(i,1:i));
  fprintf('\n');
end

k = kplus1-1;
fprintf('\nPolynomial approximation and residuals for degree %d\n', k);
ak = a(kplus1,1:kplus1);

m = size(x,2);
xh(1:m-1) = (x(1:m-1)+x(2:m))/2;
z(1:2:2*m-1) = x;
z(2:2:2*m-2) = xh;
zcap = -1 + (z-x(1))*(2/(x(m)-x(1)));
fprintf('\n   x      w       y     p(x)     Residual\n')
for i = 1:m
  j = 2*i-1;
  [p, ifail] = e02ae(ak, zcap(j));
  fprintf('%7.4f %5.2f %7.2f %8.4f %10.2e\n',zcap(j),w(i),y(i),p,p-y(i));
  if (i<m);
    [p, ifail] = e02ae(ak, zcap(j+1));
    fprintf('%7.4f %22.4f\n',zcap(j+1),p);
  end
end
```

9.2 Program Results

```
     e02ad example results

Degree  residual   Cheyshev coefficients
   0    4.07e+00   12.1740
   1    4.28e+00   12.2954    0.2740
   2    1.69e+00   20.7345    6.2016    8.1876
   3    6.82e-02   24.1429    9.4065   10.8400    3.0589

Polynomial approximation and residuals for degree 3

   x      w       y     p(x)     Residual
-1.0000  1.00   10.40  10.4461   4.61e-02
-0.8625                9.3106
```

```
-0.7250  1.00    7.90   7.7977  -1.02e-01
-0.6000                 6.2555
-0.4750  1.00    4.70   4.7025   2.52e-03
-0.3750                 3.5488
-0.2750  1.00    2.50   2.5533   5.33e-02
-0.1500                 1.6435
-0.0250  1.00    1.20   1.2390   3.90e-02
 0.0875                 1.4257
 0.2000  0.80    2.20   2.2425   4.25e-02
 0.2875                 3.3803
 0.3750  0.80    5.10   5.0116  -8.84e-02
 0.4500                 6.8400
 0.5250  0.70    9.20   9.0982  -1.02e-01
 0.6125                12.3171
 0.7000  0.50   16.10  16.2123   1.12e-01
 0.7750                20.1266
 0.8500  0.30   24.50  24.6048   1.05e-01
 0.9250                29.6779
 1.0000  0.20   35.30  35.3769   7.69e-02
```