NAG Toolbox

nag_fit_2dcheb_eval (e02cb)

1 Purpose

nag_fit_2dcheb_eval (e02cb) evaluates a bivariate polynomial from the rectangular array of coefficients in its double Chebyshev series representation.

2 Syntax

```
[ff, ifail] = nag_fit_2dcheb_eval(mfirst, k, l, x, xmin, xmax, y, ymin, ymax, a,
'mlast', mlast)
```

```
[ff, ifail] = e02cb(mfirst, k, l, x, xmin, xmax, y, ymin, ymax, a, 'mlast',
mlast)
```

3 Description

This function evaluates a bivariate polynomial (represented in double Chebyshev form) of degree k in one variable, $\bar{x}$, and degree l in the other, $\bar{y}$. The range of both variables is -1 to $+1$. However, these normalized variables will usually have been derived (as when the polynomial has been computed by nag_fit_2dcheb_lines (e02ca), for example) from your original variables x and y by the transformations

$$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{(x_{\max} - x_{\min})} \quad \text{and} \quad \bar{y} = \frac{2y - (y_{\max} + y_{\min})}{(y_{\max} - y_{\min})}.$$

(Here $x_{\min}$ and $x_{\max}$ are the ends of the range of x which has been transformed to the range -1 to $+1$ of $\bar{x}$. $y_{\min}$ and $y_{\max}$ are correspondingly for y. See Section 9). For this reason, the function has been designed to accept values of x and y rather than $\bar{x}$ and $\bar{y}$, and so requires values of $x_{\min}$, etc. to be supplied by you. In fact, for the sake of efficiency in appropriate cases, the function evaluates the polynomial for a sequence of values of x, all associated with the same value of y.

The double Chebyshev series can be written as

$$\sum_{i=0}^{k} \sum_{j=0}^{l} a_{ij} T_i(\bar{x}) T_j(\bar{y}),$$

where $T_i(\bar{x})$ is the Chebyshev polynomial of the first kind of degree i and argument $\bar{x}$, and $T_j(\bar{y})$ is similarly defined. However the standard convention, followed in this function, is that coefficients in the above expression which have either i or j zero are written $\frac{1}{2}a_{ij}$, instead of simply a_{ij}, and the coefficient with both i and j zero is written $\frac{1}{4}a_{0,0}$.

The function first forms $c_i = \sum_{j=0}^{l} a_{ij} T_j(\bar{y})$, with $a_{i,0}$ replaced by $\frac{1}{2}a_{i,0}$, for each of $i = 0, 1, \ldots, k$. The value of the double series is then obtained for each value of x, by summing $c_i \times T_i(\bar{x})$, with c_0 replaced by $\frac{1}{2}c_0$, over $i = 0, 1, \ldots, k$. The Clenshaw three term recurrence (see Clenshaw (1955)) with modifications due to Reinsch and Gentleman (1969) is used to form the sums.

4 References

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

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: **mfirst** – INTEGER

The index of the first and last x value in the array x at which the evaluation is required respectively (see Section 9).

Constraint: **mlast** $\geq$ **mfirst**.

2: **k** – INTEGER
3: **l** – INTEGER

The degree k of x and l of y, respectively, in the polynomial.

Constraint: **k** ≥ 0 and **l** ≥ 0.

4: **x**(**mlast**) – REAL (KIND=nag_wp) array

x(i), for $i =$ **mfirst**$, \ldots,$ **mlast**, must contain the x values at which the evaluation is required.

Constraint: **xmin** $\leq$ **x**$(i) \leq$ **xmax**, for all i.

5: **xmin** – REAL (KIND=nag_wp)
6: **xmax** – REAL (KIND=nag_wp)

The lower and upper ends, x_{min} and x_{max}, of the range of the variable x (see Section 3).

The values of **xmin** and **xmax** may depend on the value of y (e.g., when the polynomial has been derived using nag_fit_2dcheb_lines (e02ca)).

Constraint: **xmax** $>$ **xmin**.

7: **y** – REAL (KIND=nag_wp)

The value of the y coordinate of all the points at which the evaluation is required.

Constraint: **ymin** $\leq$ **y** $\leq$ **ymax**.

8: **ymin** – REAL (KIND=nag_wp)
9: **ymax** – REAL (KIND=nag_wp)

The lower and upper ends, y_{min} and y_{max}, of the range of the variable y (see Section 3).

Constraint: **ymax** $>$ **ymin**.

10: **a**(na) – REAL (KIND=nag_wp) array

na, the dimension of the array, must satisfy the constraint $na \geq (\mathbf{k} + 1) \times (\mathbf{l} + 1)$, the number of coefficients in a polynomial of the specified degree.

The Chebyshev coefficients of the polynomial. The coefficient a_{ij} defined according to the standard convention (see Section 3) must be in **a**$(i \times (l + 1) + j + 1)$.

5.2 Optional Input Parameters

1: **mlast** – INTEGER

Default: For **mlast**, the dimension of the array **x**.

The index of the first and last x value in the array x at which the evaluation is required respectively (see Section 9).

Constraint: **mlast** $\geq$ **mfirst**.

5.3 Output Parameters

1: **ff**(**mlast**) – REAL (KIND=nag_wp) array

ff(i) gives the value of the polynomial at the point (x_i, y), for $i = $ **mfirst**$, \dots ,$ **mlast**.

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, **mfirst** $>$ **mlast**,
or **k** < 0,
or **l** < 0,
or $na < (\mathbf{k} + 1) \times (\mathbf{l} + 1)$,
or $nwork < \mathbf{k} + 1$.

ifail $= 2$

On entry, **ymin** $\geq$ **ymax**,
or **y** $<$ **ymin**,
or **y** $>$ **ymax**.

ifail $= 3$

On entry, **xmin** $\geq$ **xmax**,
or **x**$(i) <$ **xmin**, or **x**$(i) >$ **xmax**, for some $i = $ **mfirst**, **mfirst** $+ 1, \dots ,$ **mlast**.

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 method is numerically stable in the sense that the computed values of the polynomial are exact for a set of coefficients which differ from those supplied by only a modest multiple of ***machine precision***.

8 Further Comments

The time taken is approximately proportional to $(k + 1) \times (m + l + 1)$, where $m = $ **mlast** $-$ **mfirst** $+ 1$, the number of points at which the evaluation is required.

This function is suitable for evaluating the polynomial surface fits produced by the function nag_fit_2dcheb_lines (e02ca), which provides the double array **a** in the required form. For this use, the values of $y_{\min}$ and $y_{\max}$ supplied to the present function must be the same as those supplied to nag_fit_2dcheb_lines (e02ca). The same applies to $x_{\min}$ and $x_{\max}$ if they are independent of y. If they vary with y, their values must be consistent with those supplied to nag_fit_2dcheb_lines (e02ca) (see Section 9 in nag_fit_2dcheb_lines (e02ca)).

The arguments **mfirst** and **mlast** are intended to permit the selection of a segment of the array **x** which is to be associated with a particular value of y, when, for example, other segments of **x** are associated with other values of y. Such a case arises when, after using nag_fit_2dcheb_lines (e02ca) to fit a set of data, you wish to evaluate the resulting polynomial at all the data values. In this case, if the arguments **x**, **y**, **mfirst** and **mlast** of the present function are set respectively (in terms of arguments of nag_fit_2dcheb_lines (e02ca)) to **x**, $\mathbf{y}(S)$, $1 + \sum_{i=1}^{s-1} \mathbf{m}(i)$ and $\sum_{i=1}^{s} \mathbf{m}(i)$, the function will compute values of the polynomial surface at all data points which have $\mathbf{y}(S)$ as their y coordinate (from which values the residuals of the fit may be derived).

9 Example

This example reads data in the following order, using the notation of the argument list above:

N **k** **l**
$\mathbf{a}(i)$, for $i = 1, 2, \ldots, (\mathbf{k}+1) \times (\mathbf{l}+1)$
ymin **ymax**
$\mathbf{y}(i)$ $M(i)$ $\mathbf{xmin}(i)$ $\mathbf{xmax}(i)$ $\mathrm{X1}(i)$ $\mathrm{XM}(i)$, for $i = 1, 2, \ldots, N$.

For each line $\mathbf{y} = \mathbf{y}(i)$ the polynomial is evaluated at $M(i)$ equispaced points between $\mathrm{X1}(i)$ and $\mathrm{XM}(i)$ inclusive.

9.1 Program Text

```
     function e02cb_example

fprintf('e02cb example results\n\n');

% domain
dx   = 4/19;
x    = [0.5:dx:4.5];
y    = [0:dx:4];
xmin = 0.1; xmax = 4.5;
ymin = 0;   ymax = 4;

% Fit characteristics and coefficients
k = nag_int(3);
l = nag_int(2);
a = [15.3482   5.15073  -2.20140   1.14719  -0.64419   0.30464 ...
     -0.4901  -0.00314  -6.69912   0.00153   3.00033  -0.00022];

% Evaluations on mesh
mfirst = nag_int(1);
mlast  = nag_int(20);

% Evaluate fit
for i = 1:mlast
  [fit(:,i), ifail] = e02cb( ...
                              mfirst, k, l, x, xmin, xmax, y(i), ...
                              ymin, ymax, a, 'mlast', mlast);
  sol = [x; fit(:,i)'];
end

fprintf('\nThe bivariate polynomial fit values for y = %5.1f are:\n',y(mlast));
sol = [x; fit(:,mlast)'];
fprintf('      x       p(x,y=4)\n', sol);
fprintf('%11.4f%11.4f\n', sol);

fig1 = figure;
meshc(y,x,fit);
title('Least-squares bi-variate polynomial fit');
xlabel('x');
ylabel('y');
zlabel('p(x,y)');
```

9.2 Program Results

```
     e02cb example results

The bivariate polynomial fit values for y =   4.0 are:
      x       p(x,y=4)
     0.5000     3.5575
     0.7105     6.8145
     0.9211     9.3405
     1.1316    11.1986
     1.3421    12.4520
     1.5526    13.1637
     1.7632    13.3970
     1.9737    13.2148
     2.1842    12.6803
     2.3947    11.8566
     2.6053    10.8069
     2.8158     9.5942
     3.0263     8.2816
     3.2368     6.9323
     3.4474     5.6094
     3.6579     4.3760
     3.8684     3.2951
     4.0789     2.4300
     4.2895     1.8437
     4.5000     1.5993
```