NAG Toolbox for MATLAB

e02cb

1 Purpose

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

2 Syntax

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

3 Description

This (sub)program 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 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 8). For this reason, the (sub)program 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 (sub)program, 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 (sub)program 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 – int32 scalar**

the index of the first and last x value in the array x at which the evaluation is required respectively (see Section 8).

Constraint: $\mathbf{mlast} \geq \mathbf{mfirst}$.

2: **k – int32 scalar**

3: **l – int32 scalar**

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

Constraint: $\mathbf{k} \geq 0$ and $\mathbf{l} \geq 0$.

4: **x(mlast) – double array**

$\mathbf{x}(i)$, for $i = \mathbf{mfirst}, \mathbf{mfirst} + 1, \ldots, \mathbf{mlast}$, must contain the x values at which the evaluation is required.

Constraint: $\mathbf{xmin} \leq \mathbf{x}(i) \leq \mathbf{xmax}$, for all i.

5: **xmin – double scalar**

6: **xmax – double scalar**

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 e02ca).

Constraint: $\mathbf{xmax} > \mathbf{xmin}$.

7: **y – double scalar**

The value of the y co-ordinate of all the points at which the evaluation is required.

Constraint: $\mathbf{ymin} \leq \mathbf{y} \leq \mathbf{ymax}$.

8: **ymin – double scalar**

9: **ymax – double scalar**

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

Constraint: $\mathbf{ymax} > \mathbf{ymin}$.

10: **a(na) – double array**

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

5.2 Optional Input Parameters

1: **mlast – int32 scalar**

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

the index of the first and last x value in the array x at which the evaluation is required respectively (see Section 8).

Constraint: $\mathbf{mlast} \geq \mathbf{mfirst}$.

5.3 Input Parameters Omitted from the MATLAB Interface

na, work, nwork

5.4 Output Parameters

1: **ff(mlast) – double array**

ff(i) gives the value of the polynomial at the point (x_i, y), for $i = \mathbf{mfirst}, \mathbf{mfirst} + 1, \ldots, \mathbf{mlast}$.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

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

ifail $= 2$

On entry, $\mathbf{ymin} \geq \mathbf{ymax}$,
or $\mathbf{y} < \mathbf{ymin}$,
or $\mathbf{y} > \mathbf{ymax}$.

ifail $= 3$

On entry, $\mathbf{xmin} \geq \mathbf{xmax}$,
or $\mathbf{x}(i) < \mathbf{xmin}$, or $\mathbf{x}(i) > \mathbf{xmax}$, for some $i = \mathbf{mfirst}, \mathbf{mfirst} + 1, \ldots, \mathbf{mlast}$.

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 $m = \mathbf{mlast} - \mathbf{mfirst} + 1$, the number of points at which the evaluation is required.

This (sub)program is suitable for evaluating the polynomial surface fits produced by the (sub)program 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 (sub)program must be the same as those supplied to 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 e02ca (see Section 8 of the document for e02ca).

The parameters **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 e02ca to fit a set of data, you wish to evaluate the resulting polynomial at all the data values. In this case, if the parameters **x**, **y**, **mfirst** and **mlast** of the present function are set respectively (in terms of parameters of 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 co-ordinate (from which values the residuals of the fit may be derived).

9 Example

```
mfirst = int32(1);
k = int32(3);
l = int32(2);
x = [0.5;
     1;
     1.5;
     2;
     2.5;
     3;
     3.5;
     4;
     4.5];
xmin = 0.1;
xmax = 4.5;
y = 1;
ymin = 0;
ymax = 4;
a = [15.3482;
     5.15073;
     0.1014;
     1.14719;
     0.14419;
     -0.10464;
     0.04901;
     -0.00314;
     -0.00699;
     0.00153;
     -0.00033;
     -0.00022];
[ff, ifail] = e02cb(mfirst, k, l, x, xmin, xmax, y, ymin, ymax, a)
```

```
ff =
    2.0812
    2.1888
    2.3018
    2.4204
    2.5450
    2.6758
    2.8131
    2.9572
    3.1084
ifail =
           0
```