NAG Toolbox for MATLAB

e01sb

1 Purpose

e01sb evaluates at a given point the two-dimensional interpolant function computed by e01sa.

2 Syntax

```
[pf, ifail] = e01sb(x, y, f, triang, grads, px, py, 'm', m)
```

3 Description

e01sb takes as input the parameters defining the interpolant $F(x,y)$ of a set of scattered data points (x_r,y_r,f_r), for $r = 1,2,\ldots,m$, as computed by e01sa, and evaluates the interpolant at the point (px,py).

If (px,py) is equal to (x_r,y_r) for some value of r, the returned value will be equal to f_r.

If (px,py) is not equal to (x_r,y_r) for any r, the derivatives in **grads** will be used to compute the interpolant. A triangle is sought which contains the point (px,py), and the vertices of the triangle along with the partial derivatives and f_r values at the vertices are used to compute the value $F(px,py)$. If the point (px,py) lies outside the triangulation defined by the input parameters, the returned value is obtained by extrapolation. In this case, the interpolating function **f** is extended linearly beyond the triangulation boundary. The method is described in more detail in Renka and Cline 1984 and the code is derived from Renka 1984.

e01sb must only be called after a call to e01sa.

4 References

Renka R L 1984 Algorithm 624: Triangulation and interpolation of arbitrarily distributed points in the plane *ACM Trans. Math. Software* **10** 440–442

Renka R L and Cline A K 1984 A triangle-based C^1 interpolation method *Rocky Mountain J. Math.* **14** 223–237

5 Parameters

5.1 Compulsory Input Parameters

1: **x(m) – double array**
2: **y(m) – double array**
3: **f(m) – double array**
4: **triang(7 × m) – int32 array**
5: **grads(2,m) – double array**

m, **x**, **y**, **f**, **triang** and **grads** must be unchanged from the previous call of e01sa.

6: **px – double scalar**
7: **py – double scalar**

The point (px,py) at which the interpolant is to be evaluated.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, **x**, **y**, **f**, **triang** and **grads** must be unchanged from the previous call of e01sa.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **pf – double scalar**

The value of the interpolant evaluated at the point (px, py).

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, $\mathbf{m} < 3$.

ifail $= 2$

On entry, the triangulation information held in the array **triang** does not specify a valid triangulation of the data points. **triang** may have been corrupted since the call to e01sa.

ifail $= 3$

The evaluation point (**px**,**py**) lies outside the nodal triangulation, and the value returned in **pf** is computed by extrapolation.

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Further Comments

The time taken for a call of e01sb is approximately proportional to the number of data points, m.

The results returned by this function are particularly suitable for applications such as graph plotting, producing a smooth surface from a number of scattered points.

9 Example

```
x = [11.16;
     12.85;
     19.85;
     19.72;
     15.91;
     0;
     20.87;
```

```
     3.45;
     14.26;
     17.43;
     22.8;
     7.58;
     25;
     0;
     9.66;
     5.22;
     17.25;
     25;
     12.13;
     22.23;
     11.52;
     15.2;
     7.54;
     17.32;
     2.14;
     0.51;
     22.69;
     5.47;
     21.67;
     3.31];
y = [1.24;
     3.06;
     10.72;
     1.39;
     7.74;
     20;
     20;
     12.78;
     17.87;
     3.46;
     12.39;
     1.98;
     11.87;
     0;
     20;
     14.66;
     19.57;
     3.87;
     10.79;
     6.21;
     8.529999999999999;
     0;
     10.69;
     13.78;
     15.03;
     8.369999999999999;
     19.63;
     17.13;
     14.36;
     0.33];
f = [22.15;
     22.11;
     7.97;
     16.83;
     15.3;
     34.6;
     5.74;
     41.24;
     10.74;
     18.6;
     5.47;
     29.87;
     4.4;
     58.2;
     4.73;
     40.36;
     6.43;
```

```
     8.74;
     13.71;
     10.25;
     15.74;
     21.6;
     19.31;
     12.11;
     53.1;
     49.43;
     3.25;
     28.63;
     5.52;
     44.08];
triang = [int32(2);
     int32(12);
     int32(30);
     int32(22);
     int32(5);
     int32(21);
     int32(12);
     int32(1);
     int32(22);
     int32(10);
     int32(5);
     int32(20);
     int32(11);
     int32(29);
     int32(24);
     int32(18);
     int32(20);
     int32(10);
     int32(22);
     int32(0);
     int32(21);
     int32(2);
     int32(10);
     int32(20);
     int32(3);
     int32(24);
     int32(19);
     int32(14);
     int32(26);
     int32(25);
     int32(28);
     int32(15);
     int32(0);
     int32(15);
     int32(17);
     int32(29);
     int32(27);
     int32(0);
     int32(16);
     int32(25);
     int32(26);
     int32(23);
     int32(17);
     int32(15);
     int32(19);
     int32(24);
     int32(2);
     int32(22);
     int32(4);
     int32(20);
     int32(5);
     int32(29);
     int32(3);
     int32(20);
     int32(13);
     int32(30);
     int32(1);
```

```
      int32(2);
      int32(21);
      int32(23);
      int32(26);
      int32(27);
      int32(29);
      int32(11);
      int32(20);
      int32(18);
      int32(0);
      int32(22);
      int32(30);
      int32(26);
      int32(6);
      int32(0);
      int32(6);
      int32(28);
      int32(16);
      int32(19);
      int32(9);
      int32(17);
      int32(7);
      int32(0);
      int32(25);
      int32(8);
      int32(23);
      int32(19);
      int32(15);
      int32(28);
      int32(24);
      int32(29);
      int32(7);
      int32(15);
      int32(9);
      int32(13);
      int32(20);
      int32(4);
      int32(0);
      int32(15);
      int32(16);
      int32(23);
      int32(21);
      int32(5);
      int32(24);
      int32(9);
      int32(4);
      int32(18);
      int32(13);
      int32(11);
      int32(3);
      int32(5);
      int32(10);
      int32(5);
      int32(19);
      int32(23);
      int32(12);
      int32(2);
      int32(4);
      int32(10);
      int32(2);
      int32(1);
      int32(30);
      int32(14);
      int32(0);
      int32(19);
      int32(16);
      int32(8);
      int32(26);
      int32(12);
      int32(21);
```

```
      int32(5);
      int32(3);
      int32(29);
      int32(17);
      int32(9);
      int32(19);
      int32(16);
      int32(28);
      int32(6);
      int32(26);
      int32(8);
      int32(6);
      int32(14);
      int32(30);
      int32(12);
      int32(23);
      int32(8);
      int32(25);
      int32(7);
      int32(29);
      int32(13);
      int32(0);
      int32(6);
      int32(25);
      int32(16);
      int32(15);
      int32(3);
      int32(11);
      int32(13);
      int32(27);
      int32(7);
      int32(17);
      int32(24);
      int32(12);
      int32(26);
      int32(14);
      int32(22);
      int32(1);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
      int32(4);
      int32(10);
      int32(15);
      int32(20);
      int32(27);
      int32(33);
      int32(38);
      int32(42);
      int32(46);
      int32(51);
      int32(55);
      int32(61);
      int32(67);
      int32(72);
      int32(80);
      int32(86);
      int32(91);
```

```
     int32(95);
     int32(102);
     int32(109);
     int32(114);
     int32(121);
     int32(127);
     int32(133);
     int32(138);
     int32(145);
     int32(149);
     int32(153);
     int32(160);
     int32(165)];
grads = [-1.140405342767156, -0.4357198622693418, -1.161724005615109, -
1.276972207328971, ...
        -0.6278486024553142,  -3.132801874919405,  -0.6231009766427308,  -
4.116234348980095, ...
     -0.1496036246803416, -1.025200901835296, -0.53304481969507, ...
         -2.419284698532712,  -0.5941515752170861,  -4.305315582208205,  -
1.21077192708639, ...
         -4.277295971546778,  -0.2323837579939513,  -1.217150958273922,  -
0.7483935341782484, ...
        -1.125839379039441,  -0.1296490597603071,  -0.4934166408380433,  -
2.967901396065827, ...
     -1.159094130013523, -4.983420210060235, -4.415649278298132, ...
          -1.701023037644137,  -3.865170800158547,  -1.11444168181528,  -
3.558088598721557;
          1.102423493716169,  -0.4119937359850634,  -0.1996927058704321,  -
0.4772010898085318, ...
           -1.275982165225702,   -4.867253488763668,   0.4749949233182674,
4.049830932684454, ...
        -1.406413588708735,  -0.6115621920093465,  -0.4112033481718157,  -
1.536494551329491, ...
        -0.5379193744247506,  -0.7193844302771463,  -4.702587279573724,  -
0.4552259462405117, ...
     -1.356265978508783, -0.6099290850050068, 0.03756094871403919, ...
          -0.9677121666033058,  -1.392218980145819,  -0.4379968769089029,
1.890879686709509, ...
     -0.05775955106958167, -0.5507852746920535, 0.1560569669466433, ...
        -0.2133612182466448,  -6.007036910499079,  -0.3749392692394107,  -
0.9579636671711963];
px = 3;
py = 17;
[pf, ifail] = e01sb(x, y, f, triang, grads, px, py)
```

```
pf =
   41.2455
ifail =
           0
```