NAG Toolbox for MATLAB

e01sh

1 Purpose

e01sh evaluates the two-dimensional interpolating function generated by e01sg and its first partial derivatives.

2 Syntax

```
[q, qx, qy, ifail] = e01sh(x, y, f, iq, rq, u, v, 'm', m, 'n', n)
```

3 Description

e01sh takes as input the interpolant $Q(x,y)$ of a set of scattered data points (x_r,y_r,f_r), for $r = 1,2,\ldots,m$, as computed by e01sg, and evaluates the interpolant and its first partial derivatives at the set of points (u_i,v_i), for $i = 1,2,\ldots,n$.

e01sh must only be called after a call to e01sg.

This function is derived from the function QS2GRD described by Renka 1988a.

4 References

Renka R J 1988a Algorithm 660: QSHEP2D: Quadratic Shepard method for bivariate interpolation of scattered data *ACM Trans. Math. Software* **14** 149–150

5 Parameters

5.1 Compulsory Input Parameters

1: **x(m) – double array**

2: **y(m) – double array**

3: **f(m) – double array**

m, **x**, **y** and **f** must be the same values as were supplied in the preceding call to e01sg.

4: **iq(liq) – int32 array**

Must be unchanged from the value returned from a previous call to e01sg.

5: **rq(lrq) – double array**

Must be unchanged from the value returned from a previous call to e01sg.

6: **u(n) – double array**

7: **v(n) – double array**

The evaluation points (u_i,v_i), for $i = 1,2,\ldots,n$.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, **x**, **y** and **f** must be the same values as were supplied in the preceding call to e01sg.

2: **n – int32 scalar**

Default: The dimension of the arrays **u**, **v**, **q**, **qx**, **qy**. (An error is raised if these dimensions are not equal.)

n, the number of evaluation points.

Constraint: $\mathbf{n} \geq 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

liq, lrq

5.4 Output Parameters

1: **q(n) – double array**

The values of the interpolant at (u_i, v_i), for $i = 1, 2, \ldots, n$. If any of these evaluation points lie outside the region of definition of the interpolant the corresponding entries in **q** are set to the largest machine representable number (see x02al), and e01sh returns with **ifail** $= 3$.

2: **qx(n) – double array**

3: **qy(n) – double array**

The values of the partial derivatives of the interpolant $Q(x, y)$ at (u_i, v_i), for $i = 1, 2, \ldots, n$. If any of these evaluation points lie outside the region of definition of the interpolant, the corresponding entries in **qx** and **qy** are set to the largest machine representable number (see x02al), and e01sh returns with **ifail** $= 3$.

4: **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, $\mathbf{m} < 6$,
or $\mathbf{liq} < 2 \times \mathbf{m} + 1$,
or $\mathbf{lrq} < 6 \times \mathbf{m} + 5$,
or $\mathbf{n} < 1$.

ifail $= 2$

Values supplied in **iq** or **rq** appear to be invalid. Check that these arrays have not been corrupted between the calls to e01sg and e01sh.

ifail $= 3$

At least one evaluation point lies outside the region of definition of the interpolant. At all such points the corresponding values in **q**, **qx** and **qy** have been set to the largest machine representable number (see x02al).

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Further Comments

The time taken for a call to e01sh will depend in general on the distribution of the data points. If **x** and **y** are approximately uniformly distributed, then the time taken should be only $O(n)$. At worst $O(mn)$ time will be required.

9 Example

```
nw = int32(0);
nq = int32(0);
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];
u = [20;
     6.41;
     7.54;
     9.91;
     12.3];
v = [3.14;
     15.44;
     10.69;
     18.27;
     9.220000000000001];
[iq, rq, ifail] = e01sg(x, y, f, nw, nq);
[q, qx, qy, ifail] = e01sh(x, y, f, iq, rq, u, v)
```

```
q =
   15.8931
   34.0528
   19.3100
   13.6778
   14.5560
qx =
   -1.2837
   -3.6174
   -2.8368
   -1.5854
   -0.6763
qy =
   -0.6347
   -3.5600
    0.8061
   -4.7103
   -0.7753
ifail =
           0
```