NAG Toolbox

nag_interp_2d_scat_shep (e01sg)

1 Purpose

nag_interp_2d_scat_shep (e01sg) generates a two-dimensional interpolant to a set of scattered data points, using a modified Shepard method.

2 Syntax

```
[iq, rq, ifail] = nag_interp_2d_scat_shep(x, y, f, nw, nq, 'm', m)
[iq, rq, ifail] = e01sg(x, y, f, nw, nq, 'm', m)
```

3 Description

nag_interp_2d_scat_shep (e01sg) constructs a smooth function $Q(x,y)$ which interpolates a set of m scattered data points (x_r, y_r, f_r), for $r = 1, 2, \ldots, m$, using a modification of Shepard's method. The surface is continuous and has continuous first partial derivatives.

The basic Shepard (1968) method interpolates the input data with the weighted mean

$$Q(x,y) = \frac{\sum_{r=1}^{m} w_r(x,y) q_r}{\sum_{r=1}^{m} w_r(x,y)},$$

where $q_r = f_r$, $w_r(x,y) = \frac{1}{d_r^2}$ and $d_r^2 = (x - x_r)^2 + (y - y_r)^2$.

The basic method is global in that the interpolated value at any point depends on all the data, but this function uses a modification (see Franke and Nielson (1980) and Renka (1988a)), whereby the method becomes local by adjusting each $w_r(x,y)$ to be zero outside a circle with centre (x_r, y_r) and some radius R_w. Also, to improve the performance of the basic method, each q_r above is replaced by a function $q_r(x,y)$, which is a quadratic fitted by weighted least squares to data local to (x_r, y_r) and forced to interpolate (x_r, y_r, f_r). In this context, a point (x,y) is defined to be local to another point if it lies within some distance R_q of it. Computation of these quadratics constitutes the main work done by this function.

The efficiency of the function is further enhanced by using a cell method for nearest neighbour searching due to Bentley and Friedman (1979).

The radii R_w and R_q are chosen to be just large enough to include N_w and N_q data points, respectively, for user-supplied constants N_w and N_q. Default values of these arguments are provided by the function, and advice on alternatives is given in Section 9.2.

This function is derived from the function QSHEP2 described by Renka (1988b).

Values of the interpolant $Q(x,y)$ generated by this function, and its first partial derivatives, can subsequently be evaluated for points in the domain of the data by a call to nag_interp_2d_scat_shep_eval (e01sh).

4 References

Bentley J L and Friedman J H (1979) Data structures for range searching *ACM Comput. Surv.* **11** 397–409

Franke R and Nielson G (1980) Smooth interpolation of large sets of scattered data *Internat. J. Num. Methods Engrg.* **15** 1691–1704

Renka R J (1988a) Multivariate interpolation of large sets of scattered data *ACM Trans. Math. Software* **14** 139–148

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

Shepard D (1968) A two-dimensional interpolation function for irregularly spaced data *Proc. 23rd Nat. Conf. ACM* 517–523 Brandon/Systems Press Inc., Princeton

5 Parameters

5.1 Compulsory Input Parameters

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

The Cartesian coordinates of the data points (x_r, y_r), for $r = 1, 2, \ldots, m$.

Constraint: these coordinates must be distinct, and must not all be collinear.

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

f(r) must be set to the data value f_r, for $r = 1, 2, \ldots, m$.

4: **nw** – INTEGER

The number N_w of data points that determines each radius of influence R_w, appearing in the definition of each of the weights w_r, for $r = 1, 2, \ldots, m$ (see Section 3). Note that R_w is different for each weight. If $\mathbf{nw} \leq 0$ the default value $\mathbf{nw} = \min(19, \mathbf{m} - 1)$ is used instead.

Constraint: $\mathbf{nw} \leq \min(40, \mathbf{m} - 1)$.

5: **nq** – INTEGER

The number N_q of data points to be used in the least squares fit for coefficients defining the nodal functions $q_r(x, y)$ (see Section 3). If $\mathbf{nq} \leq 0$ the default value $\mathbf{nq} = \min(13, \mathbf{m} - 1)$ is used instead.

Constraint: $\mathbf{nq} \leq 0$ or $5 \leq \mathbf{nq} \leq \min(40, \mathbf{m} - 1)$.

5.2 Optional Input Parameters

1: **m** – INTEGER

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

m, the number of data points.

Constraint: $\mathbf{m} \geq 6$.

5.3 Output Parameters

1: **iq**(liq) – INTEGER array

$liq = 2 \times \mathbf{m} + 1$.

Integer data defining the interpolant $Q(x, y)$.

2: **rq**(lrq) – REAL (KIND=nag_wp) array

$lrq = 6 \times \mathbf{m} + 5$.

Real data defining the interpolant $Q(x, y)$.

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 $= 1$

On entry, $\mathbf{m} < 6$,
or $0 < \mathbf{nq} < 5$,
or $\mathbf{nq} > \min(40, \mathbf{m} - 1)$,
or $\mathbf{nw} > \min(40, \mathbf{m} - 1)$,
or $liq < 2 \times \mathbf{m} + 1$,
or $lrq < 6 \times \mathbf{m} + 5$.

ifail $= 2$

On entry, $(\mathbf{x}(i), \mathbf{y}(i)) = (\mathbf{x}(j), \mathbf{y}(j))$ for some $i \neq j$.

ifail $= 3$

On entry, all the data points are collinear. No unique solution exists.

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

On successful exit, the function generated interpolates the input data exactly and has quadratic accuracy.

8 Further Comments

8.1 Timing

The time taken for a call to nag_interp_2d_scat_shep (e01sg) will depend in general on the distribution of the data points. If **x** and **y** are uniformly randomly distributed, then the time taken should be $O(\mathbf{m})$. At worst $O(\mathbf{m}^2)$ time will be required.

8.2 Choice of N_w and N_q

Default values of the arguments N_w and N_q may be selected by calling nag_interp_2d_scat_shep (e01sg) with $\mathbf{nw} \leq 0$ and $\mathbf{nq} \leq 0$. These default values may well be satisfactory for many applications.

If non-default values are required they must be supplied to nag_interp_2d_scat_shep (e01sg) through positive values of **nw** and **nq**. Increasing these arguments makes the method less local. This may increase the accuracy of the resulting interpolant at the expense of increased computational cost. The default values $\mathbf{nw} = \min(19, \mathbf{m} - 1)$ and $\mathbf{nq} = \min(13, \mathbf{m} - 1)$ have been chosen on the basis of experimental results reported in Renka (1988a). In these experiments the error norm was found to vary smoothly with N_w and N_q, generally increasing monotonically and slowly with distance from the optimal pair. The method is not therefore thought to be particularly sensitive to the argument values. For further advice on the choice of these arguments see Renka (1988a).

9 Example

This program reads in a set of 30 data points and calls nag_interp_2d_scat_shep (e01sg) to construct an interpolating function $Q(x, y)$. It then calls nag_interp_2d_scat_shep_eval (e01sh) to evaluate the interpolant at a set of points.

Note that this example is not typical of a realistic problem: the number of data points would normally be larger.

9.1 Program Text

```
     function e01sg_example

fprintf('e01sg example results\n\n');

% Scattered Grid Data
x = [11.16; 12.85; 19.85; 19.72; 15.91;  0.00; 20.87;  3.45; 14.26; ...
     17.43; 22.80;  7.58; 25.00;  0.00;  9.66;  5.22; 17.25; 25.00; ...
     12.13; 22.23; 11.52; 15.20;  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; 20.00; 20.00; 12.78; 17.87; ...
      3.46; 12.39;  1.98; 11.87;  0.00; 20.00; 14.66; 19.57;  3.87; ...
     10.79;  6.21;  8.53;  0.00; 10.69; 13.78; 15.03;  8.37; 19.63; ...
     17.13; 14.36; 0.33];
f = [22.15; 22.11;  7.97; 16.83; 15.30; 34.60;  5.74; 41.24; 10.74; ...
     18.60;  5.47; 29.87;  4.40; 58.20;  4.73; 40.36;  6.43;  8.74; ...
     13.71; 10.25; 15.74; 21.60; 19.31; 12.11; 53.10; 49.43;  3.25; ...
     28.63;  5.52; 44.08];

% Generate interpolant
nw = nag_int(0);
nq = nag_int(0);
[iq, rq, ifail] = e01sg(x, y, f, nw, nq);

% Interpolation points
u = [20.00;  6.41;  7.54;  9.91; 12.30];
v = [ 3.14; 15.44; 10.69; 18.27;  9.22];

% Interpolate at interpolation points
[q, qx, qy, ifail] = e01sh(x, y, f, iq, rq, u, v);

fprintf('Interpolated values Q and its derivatives at (u,v)\n');
fprintf('     u      v      q      qx     qy\n');
for i = 1:size(u,1)
  fprintf('%7.2f%7.2f%7.2f%7.2f%7.2f\n', u(i), v(i), q(i), qx(i), qy(i));
end
```

9.2 Program Results

```
     e01sg example results

Interpolated values Q and its derivatives at (u,v)
     u      v      q      qx     qy
  20.00   3.14  15.89  -1.28  -0.63
   6.41  15.44  34.05  -3.62  -3.56
   7.54  10.69  19.31  -2.84   0.81
   9.91  18.27  13.68  -1.59  -4.71
  12.30   9.22  14.56  -0.68  -0.78
```