NAG Library Routine Document

E01SGF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

E01SGF generates a two-dimensional interpolant to a set of scattered data points, using a modified Shepard method.

2 Specification

```
SUBROUTINE E01SGF (M, X, Y, F, NW, NQ, IQ, LIQ, RQ, LRQ, IFAIL)
INTEGER            M, NW, NQ, IQ(LIQ), LIQ, LRQ, IFAIL
REAL (KIND=nag_wp) X(M), Y(M), F(M), RQ(LRQ)
```

3 Description

E01SGF 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 $q_r = f_r$, $w_r(x,y) = \dfrac{1}{d_r^2}$ 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 routine 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 routine.

The efficiency of the routine 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 parameters are provided by the routine, and advice on alternatives is given in Section 9.2.

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

Values of the interpolant $Q(x,y)$ generated by this routine, and its first partial derivatives, can subsequently be evaluated for points in the domain of the data by a call to E01SHF.

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

1: M – INTEGER *Input*

On entry: m, the number of data points.

Constraint: $\mathrm{M} \geq 6$.

2: X(M) – REAL (KIND=nag_wp) array *Input*

3: Y(M) – REAL (KIND=nag_wp) array *Input*

On entry: 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.

4: F(M) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{F}(r)$ must be set to the data value f_r, for $r = 1, 2, \ldots, m$.

5: NW – INTEGER *Input*

On entry: 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 $\mathrm{NW} \leq 0$ the default value $\mathrm{NW} = \min(19, \mathrm{M} - 1)$ is used instead.

Constraint: $\mathrm{NW} \leq \min(40, \mathrm{M} - 1)$.

6: NQ – INTEGER *Input*

On entry: 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 $\mathrm{NQ} \leq 0$ the default value $\mathrm{NQ} = \min(13, \mathrm{M} - 1)$ is used instead.

Constraint: $\mathrm{NQ} \leq 0$ or $5 \leq \mathrm{NQ} \leq \min(40, \mathrm{M} - 1)$.

7: IQ(LIQ) – INTEGER array *Output*

On exit: integer data defining the interpolant $Q(x, y)$.

8: LIQ – INTEGER *Input*

On entry: the dimension of the array IQ as declared in the (sub)program from which E01SGF is called.

Constraint: $\mathrm{LIQ} \geq 2 \times \mathrm{M} + 1$.

9: RQ(LRQ) – REAL (KIND=nag_wp) array *Output*

On exit: real data defining the interpolant $Q(x, y)$.

10: LRQ – INTEGER *Input*

On entry: the dimension of the array RQ as declared in the (sub)program from which E01SGF is called.

Constraint: LRQ $\geq 6 \times \mathrm{M} + 5$.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, $\mathrm{M} < 6$,
or $0 < \mathrm{NQ} < 5$,
or $\mathrm{NQ} > \min(40, \mathrm{M} - 1)$,
or $\mathrm{NW} > \min(40, \mathrm{M} - 1)$,
or $\mathrm{LIQ} < 2 \times \mathrm{M} + 1$,
or $\mathrm{LRQ} < 6 \times \mathrm{M} + 5$.

IFAIL $= 2$

On entry, $(\mathrm{X}(i), \mathrm{Y}(i)) = (\mathrm{X}(j), \mathrm{Y}(j))$ 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.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

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

8 Parallelism and Performance

E01SGF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

E01SGF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

9.1 Timing

The time taken for a call to E01SGF 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 $O(\mathrm{M})$. At worst $O\left(\mathrm{M}^2\right)$ time will be required.

9.2 Choice of N_w and N_q

Default values of the parameters N_w and N_q may be selected by calling E01SGF with $\mathrm{NW} \leq 0$ and $\mathrm{NQ} \leq 0$. These default values may well be satisfactory for many applications.

If non-default values are required they must be supplied to E01SGF through positive values of NW and NQ. Increasing these parameters makes the method less local. This may increase the accuracy of the resulting interpolant at the expense of increased computational cost. The default values $\mathrm{NW} = \min(19, \mathrm{M}-1)$ and $\mathrm{NQ} = \min(13, \mathrm{M}-1)$ 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 parameter values. For further advice on the choice of these parameters see Renka (1988a).

10 Example

This program reads in a set of 30 data points and calls E01SGF to construct an interpolating function $Q(x, y)$. It then calls E01SHF 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.

10.1 Program Text

```
    Program e01sgfe

!     E01SGF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: e01sgf, e01shf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, liq, lrq, m, n, nq, nw
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: f(:), q(:), qx(:), qy(:), rq(:),     &
                                          u(:), v(:), x(:), y(:)
      Integer, Allocatable             :: iq(:)
!     .. Executable Statements ..
```

```
      Write (nout,*) 'E01SGF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

!     Input the number of data points

      Read (nin,*) m
      liq = 2*m + 1
      lrq = 6*m + 5
      Allocate (x(m),y(m),f(m),iq(liq),rq(lrq))

      Do i = 1, m
        Read (nin,*) x(i), y(i), f(i)
      End Do

!     Generate the interpolant.

      nq = 0
      nw = 0

      ifail = 0
      Call e01sgf(m,x,y,f,nw,nq,iq,liq,rq,lrq,ifail)

!     Input the number of evaluation points.

      Read (nin,*) n
      Allocate (u(n),v(n),q(n),qx(n),qy(n))

      Do i = 1, n
        Read (nin,*) u(i), v(i)
      End Do

!     Evaluate the interpolant using E01SHF.

      ifail = 0
      Call e01shf(m,x,y,f,iq,liq,rq,lrq,n,u,v,q,qx,qy,ifail)

      Write (nout,*)
      Write (nout,*) '     I      U(I)      V(I)      Q(I)'

      Do i = 1, n
        Write (nout,99999) i, u(i), v(i), q(i)
      End Do

99999 Format (1X,I6,3F10.2)
    End Program e01sgfe
```

10.2 Program Data

```
E01SGF Example Program Data
 30                        M, the number of data points
 11.16    1.24   22.15   X, Y, F data point definition
 12.85    3.06   22.11
 19.85   10.72    7.97
 19.72    1.39   16.83
 15.91    7.74   15.30
  0.00   20.00   34.60
 20.87   20.00    5.74
  3.45   12.78   41.24
 14.26   17.87   10.74
 17.43    3.46   18.60
 22.80   12.39    5.47
  7.58    1.98   29.87
 25.00   11.87    4.40
  0.00    0.00   58.20
  9.66   20.00    4.73
  5.22   14.66   40.36
 17.25   19.57    6.43
 25.00    3.87    8.74
```

```
 12.13   10.79   13.71
 22.23    6.21   10.25
 11.52    8.53   15.74
 15.20    0.00   21.60
  7.54   10.69   19.31
 17.32   13.78   12.11
  2.14   15.03   53.10
  0.51    8.37   49.43
 22.69   19.63    3.25
  5.47   17.13   28.63
 21.67   14.36    5.52
  3.31    0.33   44.08   End of data points
5                        N, the number of evaluation points
20.00    3.14           U, V evaluation point definition
 6.41   15.44
 7.54   10.69
 9.91   18.27
12.30    9.22           End of evaluation points
```

10.3 Program Results

```
E01SGF Example Program Results

     I      U(I)      V(I)      Q(I)
     1     20.00      3.14     15.89
     2      6.41     15.44     34.05
     3      7.54     10.69     19.31
     4      9.91     18.27     13.68
     5     12.30      9.22     14.56
```