E01SEF – NAG Fortran Library Routine Document

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

E01SEF generates a two-dimensional surface interpolating a set of scattered data points, using a modified Shepard method.

2 Specification

```
    SUBROUTINE E01SEF(M, X, Y, F, RNW, RNQ, NW, NQ, FNODES, MINNQ,
   1                  WRK, IFAIL)
    INTEGER           M, NW, NQ, MINNQ, IFAIL
    real              X(M), Y(M), F(M), RNW, RNQ, FNODES(5*M), WRK(6*M)
```

3 Description

This routine constructs an interpolating surface $F(x,y)$ through 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 derivatives.

The basic Shepard method, described in [2], interpolates the input data with the weighted mean

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

where $w_r(x,y) = \frac{1}{d_r^2}$ 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 routine uses a modification due to Franke and Nielson described in [1], 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 f_r above is replaced by a function $f_r(x,y)$, 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. If there are less than 5 other points within distance R_q from (x_r, y_r), the quadratic is replaced by a linear function. In cases of rank-deficiency, the minimum norm solution is computed.

The user may specify values for R_w and R_q, but it is usually easier to choose instead two integers N_w and N_q, from which the routine will compute R_w and R_q. These integers can be thought of as the average numbers of data points lying within distances R_w and R_q respectively from each node. Default values are provided, and advice on alternatives is given in Section 8.2.

The interpolant $F(x,y)$ generated by this routine can subsequently be evaluated for any point (x,y) in the domain of the data by a call to E01SFF.

4 References

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

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

5 Parameters

1: M — INTEGER *Input*

On entry: m, the number of data points.

Constraint: M ≥ 3.

2: X(M) — ***real*** array *Input*

3: Y(M) — ***real*** array *Input*

4: F(M) — ***real*** array *Input*

On entry: the co-ordinates of the rth data point, for $r = 1, 2, \ldots, m$. The order of the data points is immaterial.

Constraint: each of the (X(r),Y(r)) pairs must be unique.

5: RNW — ***real*** *Input/Output*

6: RNQ — ***real*** *Input/Output*

On entry: suitable values for the radii R_w and R_q, described in Section 3.

Constraint: RNQ ≤ 0 or $0 <$ RNW $\leq$ RNQ.

On exit: if RNQ is set less than or equal to zero on entry, then default values for both of them will be computed from the parameters NW and NQ, and RNW and RNQ will contain these values on exit.

7: NW — INTEGER *Input*

8: NQ — INTEGER *Input*

On entry: if RNQ > 0.0 and RNW > 0.0, then NW and NQ are not referenced by the routine. Otherwise, NW and NQ must specify suitable values for the integers N_w and N_q described in Section 3.

If NQ is less than or equal to zero on entry, then default values for both of them, namely NW $= 9$ and NQ $= 18$, will be used.

Constraint: NQ ≤ 0 or $0 <$ NW $\leq$ NQ.

9: FNODES(5∗M) — ***real*** array *Output*

On exit: the coefficients of the constructed quadratic nodal functions. These are in a form suitable for passing to E01SFF.

10: MINNQ — INTEGER *Output*

On exit: the minimum number of data points that lie within radius RNQ of any node, and thus define a nodal function. If MINNQ is very small (say, less than 5), then the interpolant may be unsatisfactory in regions where the data points are sparse.

11: WRK(6∗M) — ***real*** array *Workspace*

12: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL $= 0$ unless the routine detects an error (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 detected by the routine:

IFAIL = 1

On entry, M < 3.

IFAIL = 2

On entry, RNQ > 0 and either RNW > RNQ or RNW $\leq$ 0.

IFAIL = 3

On entry, NQ > 0 and either NW > NQ or NW $\leq$ 0.

IFAIL = 4

On entry, (X(i),Y(i)) is equal to (X(j),Y(j)) for some $i \neq j$.

7 Accuracy

On successful exit, the computational errors should be negligible in most situations but the user should always check the computed surface for acceptability, by drawing contours for instance. The surface always interpolates the input data exactly.

8 Further Comments

8.1 Timing

The time taken for a call of E01SEF is approximately proportional to the number of data points, m, provided that N_q is of the same order as its default value (18). However, if N_q is increased so that the method becomes more global, the time taken becomes approximately proportional to m^2.

8.2 Choice of $\mathrm{N_w}$ and $\mathrm{N_q}$

Note first that the radii R_w and R_q, described in Section 3, are computed as $\frac{D}{2}\sqrt{\frac{N_w}{m}}$ and $\frac{D}{2}\sqrt{\frac{N_q}{m}}$ respectively, where D is the maximum distance between any pair of data points.

Default values $N_w = 9$ and $N_q = 18$ work quite well when the data points are fairly uniformly distributed. However, for data having some regions with relatively few points or for small data sets ($m < 25$), a larger value of N_w may be needed. This is to ensure a reasonable number of data points within a distance R_w of each node, and to avoid some regions in the data area being left outside all the discs of radius R_w on which the weights $w_r(x,y)$ are non-zero. Maintaining N_q approximately equal to $2N_w$ is usually an advantage.

Note however that increasing N_w and N_q does not improve the quality of the interpolant in all cases. It does increase the computational cost and makes the method less local.

9 Example

This program reads in a set of 30 data points and calls E01SEF to construct an interpolating surface. It then calls E01SFF to evaluate the interpolant at a sample of points on a rectangular grid.

Note that this example is not typical of a realistic problem: the number of data points would normally be larger, and the interpolant would need to be evaluated on a finer grid to obtain an accurate plot, say.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E01SEF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=100,NMAX=25)
*     .. Local Scalars ..
      real             RNQ, RNW, XHI, XLO, YHI, YLO
      INTEGER          I, IFAIL, J, M, MINNQ, NQ, NW, NX, NY
*     .. Local Arrays ..
      real             F(MMAX), FNODES(5*MMAX), PF(NMAX), PX(NMAX),
     +                 PY(NMAX), WRK(6*MMAX), X(MMAX), Y(MMAX)
*     .. External Subroutines ..
      EXTERNAL         E01SEF, E01SFF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01SEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*     Input the number of nodes.
      READ (NIN,*) M
      IF (M.GE.1 .AND. M.LE.MMAX) THEN
*        Input the nodes (X,Y) and heights, F.
         DO 20 I = 1, M
            READ (NIN,*) X(I), Y(I), F(I)
   20    CONTINUE
*        Compute the nodal function coefficients.
         RNQ = 0.0e0
         NQ = 0
         IFAIL = 0
*
         CALL E01SEF(M,X,Y,F,RNW,RNQ,NW,NQ,FNODES,MINNQ,WRK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99997) '    RNW =', RNW, '  RNQ =', RNQ,
     +     '  MINNQ =', MINNQ
         WRITE (NOUT,*)
*        Evaluate the interpolant on a rectangular grid at NX*NY points
*        over the domain (XLO to XHI) x (YLO to YHI).
         READ (NIN,*) NX, XLO, XHI
         READ (NIN,*) NY, YLO, YHI
         IF (NX.LE.NMAX .AND. NY.LE.NMAX) THEN
            DO 40 I = 1, NX
               PX(I) = (real(NX-I)/(NX-1))*XLO + (real(I-1)/(NX-1))*XHI
   40       CONTINUE
            DO 60 I = 1, NY
               PY(I) = (real(NY-I)/(NY-1))*YLO + (real(I-1)/(NY-1))*YHI
   60       CONTINUE
            WRITE (NOUT,99999) '          X', (PX(I),I=1,NX)
            WRITE (NOUT,*) '     Y'
            DO 100 I = NY, 1, -1
               DO 80 J = 1, NX
```

```
                  IFAIL = 0
*
                  CALL E01SFF(M,X,Y,F,RNW,FNODES,PX(J),PY(I),PF(J),
     +                        IFAIL)
*
   80          CONTINUE
               WRITE (NOUT,99998) PY(I), (PF(J),J=1,NX)
  100       CONTINUE
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,7F8.2)
99998 FORMAT (1X,F8.2,3X,7F8.2)
99997 FORMAT (1X,A,F8.2,A,F8.2,A,I3)
      END
```

9.2 Program Data

```
E01SEF 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
   7    3.0   21.0           Grid definition, X axis
   6    2.0   17.0           Grid definition, Y axis
```

9.3 Program Results

```
E01SEF Example Program Results

    RNW =    8.22  RNQ =   11.62  MINNQ =  4

          X    3.00    6.00    9.00   12.00   15.00   18.00   21.00
     Y
   17.00      39.27   27.87   21.90   14.43   12.06    9.48    5.26
   14.00      47.21   37.79   25.17   14.43   13.29   11.34    6.23
   11.00      37.41   24.65   16.31   13.76   12.75   10.40    6.98
    8.00      35.43   20.22   18.25   15.72   15.56   13.02    9.63
    5.00      40.83   32.47   25.15   21.36   18.98   16.81   12.60
    2.00      44.56   34.46   26.48   22.30   21.09   18.70   15.07
```