E01SAF – 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

E01SAF generates a two-dimensional surface interpolating a set of scattered data points, using the method of Renka and Cline.

2 Specification

```
SUBROUTINE E01SAF(M, X, Y, F, TRIANG, GRADS, IFAIL)
INTEGER            M, TRIANG(7*M), IFAIL
real               X(M), Y(M), F(M), GRADS(2,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 method due to Renka and Cline. In the (x,y) plane, the data points must be distinct. The constructed surface is continuous and has continuous first derivatives.

The method involves firstly creating a triangulation with all the (x,y) data points as nodes, the triangulation being as nearly equiangular as possible (see Cline and Renka [1]). Then gradients in the x- and y-directions are estimated at node r, for $r = 1,2,\ldots,m$, as the partial derivatives of a quadratic function of x and y which interpolates the data value f_r, and which fits the data values at nearby nodes (those within a certain distance chosen by the algorithm) in a weighted least-squares sense. The weights are chosen such that closer nodes have more influence than more distant nodes on derivative estimates at node r. The computed partial derivatives, with the f_r values, at the three nodes of each triangle define a piecewise polynomial surface of a certain form which is the interpolant on that triangle. See Renka and Cline [4] for more detailed information on the algorithm, a development of that by Lawson [2]. The code is derived from Renka [3].

The interpolant $F(x,y)$ can subsequently be evaluated at any point (x,y) inside or outside the domain of the data by a call to E01SBF. Points outside the domain are evaluated by extrapolation.

4 References

[1] Cline A K and Renka R L (1984) A storage-efficient method for construction of a Thiessen triangulation *Rocky Mountain J. Math.* **14** 119–139

[2] Lawson C L (1977) Software for C^1 surface interpolation *Mathematical Software III* (ed J R Rice) Academic Press 161–194

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

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

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 data points are accepted in any order, but see Section 8.

Constraint: the (x, y) nodes must not all be collinear, and each node must be unique.

5: TRIANG(7∗M) — INTEGER array *Output*

On exit: a data structure defining the computed triangulation, in a form suitable for passing to E01SBF.

6: GRADS(2,M) — ***real*** array *Output*

On exit: the estimated partial derivatives at the nodes, in a form suitable for passing to E01SBF. The derivatives at node r with respect to x and y are contained in GRADS$(1, r)$ and GRADS$(2, r)$ respectively, for $r = 1, 2, \ldots, m$.

7: 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, all the (X,Y) pairs are collinear.

IFAIL = 3

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

The time taken for a call of E01SAF is approximately proportional to the number of data points, m. The routine is more efficient if, before entry, the values in X, Y, F are arranged so that the X array is in ascending order.

9 Example

This program reads in a set of 30 data points and calls E01SAF to construct an interpolating surface. It then calls E01SBF 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.

```
*     E01SAF 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             XHI, XLO, YHI, YLO
      INTEGER          I, IFAIL, J, M, NX, NY
*     .. Local Arrays ..
      real             F(MMAX), GRADS(2,MMAX), PF(NMAX), PX(NMAX),
     +                 PY(NMAX), X(MMAX), Y(MMAX)
      INTEGER          TRIANG(7*MMAX)
*     .. External Subroutines ..
      EXTERNAL         E01SAF, E01SBF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01SAF 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
*        Generate the triangulation and gradients.
         IFAIL = 0
*
         CALL E01SAF(M,X,Y,F,TRIANG,GRADS,IFAIL)
*
*        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,*)
            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 E01SBF(M,X,Y,F,TRIANG,GRADS,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)
      END
```

9.2 Program Data

```
E01SAF 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 the data points
   7    3.0   21.0            Grid definition, X axis
   6    2.0   17.0            Grid definition, Y axis
```

9.3 Program Results

```
 E01SAF Example Program Results

          X     3.00    6.00    9.00   12.00   15.00   18.00   21.00
     Y
   17.00       41.25   27.62   18.03   12.29   11.68    9.09    5.37
   14.00       47.61   36.66   22.87   14.02   13.44   11.20    6.46
```

```
   11.00      38.55   25.25   16.72   13.83   13.08   10.71    6.88
    8.00      37.90   23.97   16.79   16.43   15.46   13.02    9.30
    5.00      40.49   29.26   22.51   20.72   19.30   16.72   12.87
    2.00      43.52   33.91   26.59   22.23   21.15   18.67   14.88
```