E01TGF – 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

E01TGF generates a three-dimensional interpolant to a set of scattered data points, using a modified Shepard method.

2 Specification

```
SUBROUTINE E01TGF(M, X, Y, Z, F, NW, NQ, IQ, LIQ, RQ, LRQ, IFAIL)
real            X(M), Y(M), Z(M), F(M), RQ(LRQ)
INTEGER         M, NW, NQ, IQ(LIQ), LIQ, LRQ, IFAIL
```

3 Description

This routine constructs a smooth function $Q(x, y, z)$ which interpolates a set of m scattered data points (x_r, y_r, z_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 method, which is a generalization of the two-dimensional method described in [5], interpolates the input data with the weighted mean

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

where

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

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 [2], [3]), whereby the method becomes local by adjusting each $w_r(x, y, z)$ to be zero outside a sphere with centre (x_r, y_r, z_r) and some radius R_w. Also, to improve the performance of the basic method, each q_r above is replaced by a function $q_r(x, y, z)$, which is a quadratic fitted by weighted least-squares to data local to (x_r, y_r, z_r) and forced to interpolate (x_r, y_r, z_r, f_r). In this context, a point (x, y, z) 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 [1].

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 8.2.

This routine is derived from the routine QSHEP3 described by Renka [4].

Values of the interpolant $Q(x, y, z)$ 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 E01THF.

4 References

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

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

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

[4] Renka R J (1988) Algorithm 661: QSHEP3D: Quadratic Shepard method for trivariate interpolation of scattered data *ACM Trans. Math. Software* **14** 151–152

[5] 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 $\geq$ 10.

2: X(M) — ***real*** array   *Input*
3: Y(M) — ***real*** array   *Input*
4: Z(M) — ***real*** array   *Input*

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

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

5: F(M) — ***real*** array   *Input*

On entry: the data values f_r, for $r = 1, 2, \ldots, m$.

6: 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 NW $\leq$ 0 the default value NW = min(32,M$-$1) is used instead.

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

7: 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 $q_r(x, y, z)$ (see Section 3). If NQ $\leq$ 0 the default value NQ = min(17,M$-$1) is used instead.

Constraint: NQ $\leq$ 0 or 9 $\leq$ NQ $\leq$ min(40,M$-$1).

8: IQ(LIQ) — INTEGER array   *Output*

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

9: LIQ — INTEGER   *Input*

On entry: the dimension of the array IQ as declared in the (sub)program from which E01TGF is called.

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

10: RQ(LRQ) — ***real*** array   *Output*

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

11: LRQ — INTEGER   *Input*

On entry: the dimension of the array RQ as declared in the (sub)program from which E01TGF is called.

Constraint: LRQ $\geq$ 10 $\times$ M + 7.

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 Errors 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 < 10,

     or 0 < NQ < 9,

     or NQ > min(40,M-1),

     or NW > min(40,M-1),

     or LIQ < 2 × M + 1,

     or LRQ < 10 × M + 7.

IFAIL = 2

On entry, $(X(i),Y(i),Z(i)) = (X(j),Y(j),Z(j))$ for some $i \neq j$.

IFAIL = 3

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

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 E01TGF will depend in general on the distribution of the data points. If X, Y and Z are uniformly randomly distributed, then the time taken should be O(M). At worst O(M^2) time will be required.

8.2 Choice of N_w and N_q

Default values of the parameters N_w and N_q may be selected by calling E01TGF with NW $\leq$ 0 and NQ $\leq$ 0. These default values may well be satisfactory for many applications.

If non-default values are required they must be supplied to E01TGF 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 NW = min(32,M-1) and NQ = min(17,M-1) have been chosen on the basis of experimental results reported in [3]. 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 [3].

9 Example

This program reads in a set of 30 data points and calls E01TGF to construct an interpolating function $Q(x, y, z)$. It then calls E01THF 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

```
*     E01TGF Example Program Text
*     Mark 18 Release. NAG Copyright 1997.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LIQ, LRQ
      PARAMETER        (MMAX=100,NMAX=100,LIQ=2*MMAX+1,LRQ=10*MMAX+7)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, M, N, NQ, NW
*     .. Local Arrays ..
      real             F(MMAX), Q(NMAX), QX(NMAX), QY(NMAX), QZ(NMAX),
     +                 RQ(LRQ), U(NMAX), V(NMAX), W(NMAX), X(MMAX),
     +                 Y(MMAX), Z(MMAX)
      INTEGER          IQ(LIQ)
*     .. External Subroutines ..
      EXTERNAL         E01TGF, E01THF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01TGF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
*
*     Input the number of nodes.
*
      READ (NIN,*) M
      IF (M.GT.0 .AND. M.LE.MMAX) THEN
*
*         Input the data points X,Y,Z and F.
*
          DO 20 I = 1, M
             READ (NIN,*) X(I), Y(I), Z(I), F(I)
   20     CONTINUE
*
*         Generate the interpolant.
*
          NQ = 0
          NW = 0
          IFAIL = 0
          CALL E01TGF(M,X,Y,Z,F,NW,NQ,IQ,LIQ,RQ,LRQ,IFAIL)
*
*         Input the number of evaluation points.
*
          READ (NIN,*) N
*
*         Input the evaluation points.
*
          DO 40 I = 1, N
             READ (NIN,*) U(I), V(I), W(I)
   40     CONTINUE
```

```
      *
      *        Evaluate the interpolant using E01THF.
      *
               IFAIL = -1
               CALL E01THF(M,X,Y,Z,F,IQ,LIQ,RQ,LRQ,N,U,V,W,Q,QX,QY,QZ,IFAIL)
      *
               WRITE (NOUT,*) '    I     U(I)      V(I)      W(I)       Q(I)'
               DO 60 I = 1, N
                   WRITE (NOUT,99999) I, U(I), V(I), W(I), Q(I)
        60     CONTINUE
      *
            END IF
            STOP
      *
      99999 FORMAT (1X,I6,4F10.4)
            END
```

9.2   Program Data

```
E01TGF Example Program Data
 30                         M, the number of data points
  0.80  0.23  0.37  0.51  X, Y, Z, F data point definition
  0.23  0.88  0.05  1.80
  0.18  0.43  0.04  0.11
  0.58  0.95  0.62  2.65
  0.64  0.69  0.20  0.93
  0.88  0.35  0.49  0.72
  0.30  0.10  0.78 -0.11
  0.87  0.09  0.05  0.67
  0.04  0.02  0.40  0.00
  0.62  0.90  0.43  2.20
  0.87  0.96  0.24  3.17
  0.62  0.64  0.45  0.74
  0.86  0.13  0.47  0.64
  0.87  0.60  0.46  1.07
  0.49  0.43  0.13  0.22
  0.12  0.61  0.00  0.41
  0.02  0.71  0.82  0.58
  0.62  0.93  0.44  2.48
  0.49  0.54  0.04  0.37
  0.36  0.56  0.39  0.35
  0.62  0.42  0.97 -0.20
  0.01  0.72  0.45  0.78
  0.41  0.36  0.52  0.11
  0.17  0.99  0.65  2.82
  0.51  0.29  0.59  0.14
  0.85  0.05  0.04  0.61
  0.20  0.20  0.87 -0.25
  0.04  0.67  0.04  0.59
  0.31  0.63  0.18  0.50
  0.88  0.27  0.07  0.71  End of data points
 6                         N, the number of evaluation points
  0.10  0.10  0.10         U, V, W evaluation point definition
  0.20  0.20  0.20
  0.30  0.30  0.30
  0.40  0.40  0.40
  0.50  0.50  0.50
  0.60  0.60  0.60         End of evaluation points
```

9.3 Program Results

```
E01TGF Example Program Results

      I     U(I)     V(I)     W(I)     Q(I)
      1    0.1000   0.1000   0.1000   0.2630
      2    0.2000   0.2000   0.2000   0.1182
      3    0.3000   0.3000   0.3000   0.0811
      4    0.4000   0.4000   0.4000   0.1552
      5    0.5000   0.5000   0.5000   0.3019
      6    0.6000   0.6000   0.6000   0.5712
```