NAG Library Routine Document

E01TMF

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

E01TMF generates a five-dimensional interpolant to a set of scattered data points, using a modified Shepard method.

2 Specification

```
SUBROUTINE E01TMF (M, X, F, NW, NQ, IQ, RQ, IFAIL)

INTEGER             M, NW, NQ, IQ(2*M+1), IFAIL
REAL (KIND=nag_wp) X(5,M), F(M), RQ(21*M+11)
```

3 Description

E01TMF constructs a smooth function $Q(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^5$ which interpolates a set of m scattered data points $(\mathbf{x}_r, f_r)$, 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 Shepard (1968), interpolates the input data with the weighted mean

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

where $q_r = f_r$, $w_r(\mathbf{x}) = \dfrac{1}{d_r^2}$ and $d_r^2 = \|\mathbf{x} - \mathbf{x}_r\|_2{}^2$.

The basic method is global in that the interpolated value at any point depends on all the data, but E01TMF uses a modification (see Franke and Nielson (1980) and Renka (1988a)), whereby the method becomes local by adjusting each $w_r(\mathbf{x})$ to be zero outside a hypersphere with centre $\mathbf{x}_r$ and some radius R_w. Also, to improve the performance of the basic method, each q_r above is replaced by a function $q_r(\mathbf{x})$, which is a quadratic fitted by weighted least squares to data local to $\mathbf{x}_r$ and forced to interpolate $(\mathbf{x}_r, f_r)$. In this context, a point $\mathbf{x}$ is defined to be local to another point if it lies within some distance R_q of it.

The efficiency of E01TMF is enhanced by using a cell method for nearest neighbour searching due to Bentley and Friedman (1979) with a cell density of 3.

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, and advice on alternatives is given in Section 8.2.

E01TMF is derived from the new implementation of QSHEP3 described by Renka (1988b). It uses the modification for five-dimensional interpolation described by Berry and Minser (1999).

Values of the interpolant $Q(\mathbf{x})$ generated by E01TMF, and its first partial derivatives, can subsequently be evaluated for points in the domain of the data by a call to E01TNF.

4 References

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

Berry M W, Minser K S (1999) Algorithm 798: high-dimensional interpolation using the modified Shepard method *ACM Trans. Math. Software* **25** 353–366

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 661: QSHEP3D: Quadratic Shepard method for trivariate interpolation of scattered data *ACM Trans. Math. Software* **14** 151–152

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.

Note: on the basis of experimental results reported in Berry and Minser (1999), it is recommended to use $\mathrm{M} \geq 4000$.

Constraint: $\mathrm{M} \geq 23$.

2: X(5,M) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{X}(1:5,r)$ must be set to the Cartesian coordinates of the data point $\mathbf{x}_r$, for $r = 1, 2, \ldots, m$.

Constraint: these coordinates must be distinct, and must not all lie on the same four-dimensional hypersurface.

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

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

4: 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 $\mathrm{NW} = \min(32, \mathrm{M} - 1)$ is used instead.

Constraint: $\mathrm{NW} \leq \min(50, \mathrm{M} - 1)$.

5: NQ – INTEGER *Input*

On entry: the number N_q of data points to be used in the least squares fit for coefficients defining the quadratic functions $q_r(\mathbf{x})$ (see Section 3). If $\mathrm{NQ} \leq 0$ the default value $\mathrm{NQ} = \min(50, \mathrm{M} - 1)$ is used instead.

Constraint: $\mathrm{NQ} \leq 0$ or $20 \leq \mathrm{NQ} \leq \min(70, \mathrm{M} - 1)$.

6: IQ($2 \times \mathrm{M} + 1$) – INTEGER array *Output*

On exit: integer data defining the interpolant $Q(\mathbf{x})$.

7: RQ($21 \times \mathrm{M} + 11$) – REAL (KIND=nag_wp) array *Output*

On exit: real data defining the interpolant $Q(\mathbf{x})$.

8: 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, M < 23,
or NQ < 20,
or NQ $> \min(70, \mathrm{M} - 1)$,
or NW $> \min(50, \mathrm{M} - 1)$.

IFAIL $= 2$

On entry, $\mathrm{X}(1:5, i) = \mathrm{X}(1:5, j)$ for some $i \neq j$. The interpolant cannot be derived.

IFAIL $= 3$

On entry, all the data points lie on the same four-dimensional hypersurface. No unique solution exists.

7 Accuracy

On successful exit, the routine generated interpolates the input data exactly and has quadratic precision. Overall accuracy of the interpolant is affected by the choice of parameters NW and NQ as well as the smoothness of the routine represented by the input data. Berry and Minser (1999) report on the results obtained for a set of test routines.

8 Further Comments

8.1 Timing

The time taken for a call to E01TMF will depend in general on the distribution of the data points and on the choice of N_w and N_q parameters. If the data points are uniformly randomly distributed, then the time taken should be $O(m)$. At worst $O\left(m^2\right)$ 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 E01TMF with NW ≤ 0 and NQ ≤ 0. These default values may well be satisfactory for many applications.

If nondefault values are required they must be supplied to E01TMF through positive values of NW and NQ. Increasing these parameter values 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 $= \min(32, \mathrm{M} - 1)$ and NQ $= \min(50, \mathrm{M} - 1)$ have been chosen on the basis of experimental results reported in Berry and Minser (1999). In these experiments the error norm was found to increase with the

decrease of N_q, but to be little affected by the choice of N_w. The choice of both, directly affected the time taken by the routine. For further advice on the choice of these parameters see Berry and Minser (1999).

9 Example

This program reads in a set of 30 data points and calls E01TMF to construct an interpolating function $Q(\mathbf{x})$. It then calls E01TNF 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.

See also Section 9 in E01TNF.

9.1 Program Text

```
    Program e01tmfe

!     E01TMF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: e01tmf, e01tnf, 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(:,:), rq(:), x(:,:),  &
                                          xe(:,:)
      Integer, Allocatable             :: iq(:)
!     .. Executable Statements ..
      Write (nout,*) 'E01TMF Example Program Results'

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

!     Input the number of nodes.
      Read (nin,*) m
      liq = 2*m + 1
      lrq = 21*m + 11
      Allocate (x(5,m),f(m),iq(liq),rq(lrq))

!     Input the data points X and F.
      Do i = 1, m
        Read (nin,*) x(1:5,i), f(i)
      End Do

!     Generate the interpolant.
      nq = 0
      nw = 0

      ifail = 0
      Call e01tmf(m,x,f,nw,nq,iq,rq,ifail)

!     Input the number of evaluation points.
      Read (nin,*) n
      Allocate (xe(5,n),q(n),qx(5,n))

!     Input the evaluation points.
      Do i = 1, n
        Read (nin,*) xe(1:5,i)
      End Do

!     Evaluate the interpolant using E01TNF.
      ifail = 0
```

```
      Call e01tnf(m,x,f,iq,rq,n,xe,q,qx,ifail)

      Write (nout,99998) 'Interpolated Evaluation Points', 'Values'
      Write (nout,99997)
      Write (nout,99996)(i,i=1,5)
      Write (nout,99997)
      Write (nout,99999)(i,xe(1:5,i),q(i),i=1,n)

99999 Format (1X,I4,1X,6F10.4)
99998 Format (/4X,'   |',10X,A31,10X,'|',A7)
99997 Format (4X,'---|',51('-'),'+',7('-'))
99996 Format (4X,'I  |',5(2X,'XE(',I1,',I)',1X),1X,'|',3X,'Q(I)')
    End Program e01tmfe
```

9.2 Program Data

```
 E01TMF Example Program Data
 30                                M the number of data points
 0.81  0.15  0.44  0.83  0.21  6.39  X, F data point definition
 0.91  0.96  0.00  0.09  0.98  2.50
 0.13  0.88  0.22  0.21  0.73  9.34
 0.91  0.49  0.39  0.79  0.47  7.52
 0.63  0.41  0.72  0.68  0.65  6.91
 0.10  0.13  0.77  0.47  0.22  4.68
 0.28  0.93  0.24  0.90  0.96 45.40
 0.55  0.01  0.04  0.41  0.26  5.48
 0.96  0.19  0.95  0.66  0.99  2.75
 0.96  0.32  0.53  0.96  0.84  7.43
 0.16  0.05  0.16  0.30  0.58  6.05
 0.97  0.14  0.36  0.72  0.78  5.77
 0.96  0.73  0.28  0.75  0.28  8.68
 0.49  0.48  0.58  0.19  0.25  2.38
 0.80  0.34  0.64  0.57  0.08  3.70
 0.14  0.24  0.12  0.06  0.63  1.34
 0.42  0.45  0.03  0.68  0.66 15.18
 0.92  0.19  0.48  0.67  0.28  4.35
 0.79  0.32  0.15  0.13  0.40  1.50
 0.96  0.26  0.93  0.89  0.61  3.43
 0.66  0.83  0.41  0.17  0.09  3.10
 0.04  0.70  0.40  0.54  0.37 14.33
 0.85  0.33  0.15  0.03  0.36  0.35
 0.93  0.58  0.88  0.81  0.40  4.30
 0.68  0.29  0.88  0.60  0.47  3.77
 0.76  0.26  0.09  0.41  0.14  4.16
 0.74  0.26  0.33  0.64  0.36  6.75
 0.39  0.68  0.69  0.37  0.12  5.22
 0.66  0.52  0.17  1.00  0.43 16.23
 0.17  0.08  0.35  0.71  0.17 10.62  End of data points
6                                    N the number of evaluation points
 0.10  0.10  0.10  0.10  0.10        XE evaluation point definition
 0.20  0.20  0.20  0.20  0.20
 0.30  0.30  0.30  0.30  0.30
 0.40  0.40  0.40  0.40  0.40
 0.50  0.50  0.50  0.50  0.50
 0.60  0.60  0.60  0.60  0.60       End of evaluation points
```

9.3 Program Results

```
 E01TMF Example Program Results

      |           Interpolated Evaluation Points          | Values
   ---|---------------------------------------------------+-------
   I  |  XE(1,I)   XE(2,I)   XE(3,I)   XE(4,I)   XE(5,I)  |   Q(I)
   ---|---------------------------------------------------+-------
   1     0.1000    0.1000    0.1000    0.1000    0.1000    3.2313
```

2	0.2000	0.2000	0.2000	0.2000	0.2000	4.2476
3	0.3000	0.3000	0.3000	0.3000	0.3000	5.2695
4	0.4000	0.4000	0.4000	0.4000	0.4000	6.3838
5	0.5000	0.5000	0.5000	0.5000	0.5000	7.6837
6	0.6000	0.6000	0.6000	0.6000	0.6000	9.3885