NAG Library Routine Document

E01TKF

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

E01TKF generates a four-dimensional interpolant to a set of scattered data points, using a modified Shepard method.

2 Specification

```
SUBROUTINE E01TKF (M, X, F, NW, NQ, IQ, RQ, IFAIL)
INTEGER             M, NW, NQ, IQ(2*M+1), IFAIL
REAL (KIND=nag_wp) X(4,M), F(M), RQ(15*M+9)
```

3 Description

E01TKF constructs a smooth function $Q(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^4$ 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 E01TKF 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 E01TKF 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 arguments are provided by the routine, and advice on alternatives is given in Section 9.2.

E01TKF is derived from the new implementation of QSHEP3 described by Renka (1988b). It uses the modification for high-dimensional interpolation described by Berry and Minser (1999).

Values of the interpolant $Q(\mathbf{x})$ generated by E01TKF, and its first partial derivatives, can subsequently be evaluated for points in the domain of the data by a call to E01TLF.

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 Arguments

1: M – INTEGER *Input*

On entry: m, the number of data points.

Constraint: $\mathrm{M} \geq 16$.

2: X(4, M) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{X}(1:4, r)$ 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 three-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 $\mathrm{NQ} = \min(38, \mathrm{M} - 1)$ is used instead.

Constraint: $\mathrm{NQ} \leq 0$ or $14 \leq \mathrm{NQ} \leq \min(50, \mathrm{M} - 1)$.

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

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

7: RQ($15 \times \mathrm{M} + 9$) – 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 argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 $= \langle value \rangle$.
Constraint: M ≥ 16.

On entry, NQ $= \langle value \rangle$.
Constraint: NQ ≤ 0 or NQ ≥ 14.

On entry, NQ $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: NQ $\leq \min(50, \mathrm{M} - 1)$.

On entry, NW $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: NW $\leq \min(50, \mathrm{M} - 1)$.

IFAIL $= 2$

There are duplicate nodes in the dataset. $\mathrm{X}(i,k) = \mathrm{X}(j,k)$, for $i = \langle value \rangle$, $j = \langle value \rangle$ and $k = 1, 2, \ldots, 4$. The interpolant cannot be derived.

IFAIL $= 3$

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

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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 arguments NW and NQ as well as the smoothness of the function represented by the input data.

8 Parallelism and Performance

E01TKF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

E01TKF 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 E01TKF 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.

9.2 Choice of N_w and N_q

Default values of the arguments N_w and N_q may be selected by calling E01TKF with NW ≤ 0 and NQ ≤ 0. These default values may well be satisfactory for many applications.

If non-default values are required they must be supplied to E01TKF through positive values of NW and NQ. Increasing these argument values makes the method less local. This may increase the accuracy of the resulting interpolant at the expense of increased computational cost.

10 Example

This program reads in a set of 30 data points and calls E01TKF to construct an interpolating function $Q(\mathbf{x})$. It then calls E01TLF 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 10 in E01TLF.

10.1 Program Text

```
    Program e01tkfe

!     E01TKF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e01tkf, e01tlf, 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,*) 'E01TKF Example Program Results'

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

!     Input the number of nodes.
      Read (nin,*) m
      liq = 2*m + 1
      lrq = 15*m + 9
      Allocate (x(4,m),f(m),iq(liq),rq(lrq))

!     Input the data points X and F.
      Do i = 1, m
        Read (nin,*) x(1:4,i), f(i)
      End Do

!     Generate the interpolant.
      nq = 0
      nw = 0

      ifail = 0
      Call e01tkf(m,x,f,nw,nq,iq,rq,ifail)

!     Input the number of evaluation points.
      Read (nin,*) n
      Allocate (xe(4,n),q(n),qx(4,n))

!     Input the evaluation points.
      Do i = 1, n
        Read (nin,*) xe(1:4,i)
      End Do

!     Evaluate the interpolant using E01TLF.
      ifail = 1
      Call e01tlf(m,x,f,iq,rq,n,xe,q,qx,ifail)
      If (ifail==0 .Or. ifail==3) Then
        If (ifail==3) Then
          Write (nout,*) 'Some values below have been extrapolated'
          Write (nout,*)
        End If
        Write (nout,99998) 'Interpolated Evaluation Points', 'Values'
        Write (nout,99997)
        Write (nout,99996)(i,i=1,4)
        Write (nout,99997)
        Do i = 1, n
          Write (nout,99999) i, xe(1:4,i), q(i)
        End Do
      Else
        Write (nout,99995) 'Evaluation routine returned with IFAIL = ', ifail
      End If

99999 Format (1X,I4,1X,4F10.4,1X,F10.4)
99998 Format (/,4X,'   |',5X,A31,5X,'|',A7)
99997 Format (4X,'---|',41('-'),'+',8('-'))
99996 Format (4X,'I  |',4(2X,'XE(I,',I1,')',1X),1X,'|',2X,'Q(I)')
99995 Format (1X,A,I4)
    End Program e01tkfe
```

10.2 Program Data

```
 E01TKF Example Program Data
 30                        M the number of data points
 0.81  0.15  0.44  0.83   6.39  X, F data point definition
 0.91  0.96  0.00  0.09   2.50
 0.13  0.88  0.22  0.21   9.34
 0.91  0.49  0.39  0.79   7.52
 0.63  0.41  0.72  0.68   6.91
```

```
 0.10  0.13  0.77  0.47   4.68
 0.28  0.93  0.24  0.90  45.40
 0.55  0.01  0.04  0.41   5.48
 0.96  0.19  0.95  0.66   2.75
 0.96  0.32  0.53  0.96   7.43
 0.16  0.05  0.16  0.30   6.05
 0.97  0.14  0.36  0.72   5.77
 0.96  0.73  0.28  0.75   8.68
 0.49  0.48  0.58  0.19   2.38
 0.80  0.34  0.64  0.57   3.70
 0.14  0.24  0.12  0.06   1.34
 0.42  0.45  0.03  0.68  15.18
 0.92  0.19  0.48  0.67   4.35
 0.79  0.32  0.15  0.13   1.50
 0.96  0.26  0.93  0.89   3.43
 0.66  0.83  0.41  0.17   3.10
 0.04  0.70  0.40  0.54  14.33
 0.85  0.33  0.15  0.03   0.35
 0.93  0.58  0.88  0.81   4.30
 0.68  0.29  0.88  0.60   3.77
 0.76  0.26  0.09  0.41   4.16
 0.74  0.26  0.33  0.64   6.75
 0.39  0.68  0.69  0.37   5.22
 0.66  0.52  0.17  1.00  16.23
 0.17  0.08  0.35  0.71  10.62  End of data points
9                               N the number of evaluation points
  0.10  0.10  0.10  0.10        XE evaluation point definition
  0.20  0.20  0.20  0.20
  0.30  0.30  0.30  0.30
  0.40  0.40  0.40  0.40
  0.50  0.50  0.50  0.50
  0.60  0.60  0.60  0.60
  0.70  0.70  0.70  0.70
  0.80  0.80  0.80  0.80
  0.90  0.90  0.90  0.90        End of evaluation points
```

10.3 Program Results

```
 E01TKF Example Program Results

      |      Interpolated Evaluation Points     | Values
   ---|-----------------------------------------+--------
   I  |  XE(I,1)   XE(I,2)   XE(I,3)   XE(I,4)  |  Q(I)
   ---|-----------------------------------------+--------
   1     0.1000    0.1000    0.1000    0.1000     2.7209
   2     0.2000    0.2000    0.2000    0.2000     4.3166
   3     0.3000    0.3000    0.3000    0.3000     5.5397
   4     0.4000    0.4000    0.4000    0.4000     6.5347
   5     0.5000    0.5000    0.5000    0.5000     7.5691
   6     0.6000    0.6000    0.6000    0.6000     8.7335
   7     0.7000    0.7000    0.7000    0.7000    10.0524
   8     0.8000    0.8000    0.8000    0.8000    11.6254
   9     0.9000    0.9000    0.9000    0.9000    13.3268
```