NAG Library Routine Document

E01THF

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

E01THF evaluates the three-dimensional interpolating function generated by E01TGF and its first partial derivatives.

2 Specification

```
SUBROUTINE E01THF (M, X, Y, Z, F, IQ, LIQ, RQ, LRQ, N, U, V, W, Q, QX,     &
                   QY, QZ, IFAIL)
INTEGER            M, IQ(LIQ), LIQ, LRQ, N, IFAIL
REAL (KIND=nag_wp) X(M), Y(M), Z(M), F(M), RQ(LRQ), U(N), V(N), W(N),     &
                   Q(N), QX(N), QY(N), QZ(N)
```

3 Description

E01THF takes as input the interpolant $Q(x, y, z)$ of a set of scattered data points (x_r, y_r, z_r, f_r), for $r = 1, 2, \ldots, m$, as computed by E01TGF, and evaluates the interpolant and its first partial derivatives at the set of points (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$.

E01THF must only be called after a call to E01TGF.

This routine is derived from the routine QS3GRD described by Renka (1988).

4 References

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

5 Parameters

1: M – INTEGER *Input*

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

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

4: Z(M) – REAL (KIND=nag_wp) array *Input*

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

On entry: M, X, Y, Z and F must be the same values as were supplied in the preceding call to E01TGF.

6: IQ(LIQ) – INTEGER array *Input*

On entry: must be unchanged from the value returned from a previous call to E01TGF.

7: LIQ – INTEGER *Input*

On entry: the dimension of the array IQ as declared in the (sub)program from which E01THF is called.

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

8: RQ(LRQ) – REAL (KIND=nag_wp) array *Input*

On entry: must be unchanged from the value returned from a previous call to E01TGF.

9: LRQ – INTEGER *Input*

On entry: the dimension of the array RQ as declared in the (sub)program from which E01THF is called.

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

10: N – INTEGER *Input*

On entry: n, the number of evaluation points.

Constraint: N ≥ 1.

11: U(N) – REAL (KIND=nag_wp) array *Input*
12: V(N) – REAL (KIND=nag_wp) array *Input*
13: W(N) – REAL (KIND=nag_wp) array *Input*

On entry: U(i), V(i), W(i) must be set to the evaluation point (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$.

14: Q(N) – REAL (KIND=nag_wp) array *Output*

On exit: Q(i) contains the value of the interpolant, at (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$. If any of these evaluation points lie outside the region of definition of the interpolant the corresponding entries in Q are set to the largest machine representable number (see X02ALF), and E01THF returns with IFAIL $= 3$.

15: QX(N) – REAL (KIND=nag_wp) array *Output*
16: QY(N) – REAL (KIND=nag_wp) array *Output*
17: QZ(N) – REAL (KIND=nag_wp) array *Output*

On exit: QX(i), QY(i), QZ(i) contains the value of the partial derivatives of the interpolant $Q(x, y, z)$ at (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$. If any of these evaluation points lie outside the region of definition of the interpolant, the corresponding entries in QX, QY and QZ are set to the largest machine representable number (see X02ALF), and E01THF returns with IFAIL $= 3$.

18: 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 < 10,
or LIQ $< 2 \times \mathrm{M} + 1$,

or LRQ $< 10 \times \mathrm{M} + 7$,
or $\mathrm{N} < 1$.

IFAIL $= 2$

Values supplied in IQ or RQ appear to be invalid. Check that these arrays have not been corrupted between the calls to E01TGF and E01THF.

IFAIL $= 3$

At least one evaluation point lies outside the region of definition of the interpolant. At all such points the corresponding values in Q, QX, QY and QZ have been set to the largest machine representable number (see X02ALF).

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Parallelism and Performance

E01THF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

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

The time taken for a call to E01THF will depend in general on the distribution of the data points. If X, Y and Z are approximately uniformly distributed, then the time taken should be only $O(\mathrm{N})$. At worst $O(\mathrm{MN})$ time will be required.

10 Example

See Section 10 in E01TGF.