E01SHF – 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

E01SHF evaluates the two-dimensional interpolating function generated by E01SGF and its first partial derivatives.

2 Specification

```
      SUBROUTINE E01SHF(M, X, Y, F, IQ, LIQ, RQ, LRQ, N, U, V, Q, QX,
     1                  QY, IFAIL)
      real              X(M), Y(M), F(M), RQ(LRQ), U(N), V(N), Q(N),
     1                  QX(N), QY(N)
      INTEGER           M, IQ(LIQ), LIQ, LRQ, N, IFAIL
```

3 Description

This routine takes as input the interpolant $Q(x,y)$ of a set of scattered data points (x_r, y_r, f_r), for $r = 1, 2, \ldots, m$, as computed by E01SGF, and evaluates the interpolant and its first partial derivatives at the set of points (u_i, v_i), for $i = 1, 2, \ldots, n$.

E01SHF must only be called after a call to E01SGF.

This routine is derived from the routine QS2GRD described by Renka [1].

4 References

[1] Renka R J (1988) Algorithm 660: QSHEP2D: Quadratic Shepard method for bivariate interpolation of scattered data *ACM Trans. Math. Software* **14** 149–150

5 Parameters

1: M — INTEGER *Input*

2: X(M) — ***real*** array *Input*

3: Y(M) — ***real*** array *Input*

4: F(M) — ***real*** array *Input*

On entry: M, X, Y and F must be the same values as were supplied in the preceding call to E01SGF.

5: IQ(LIQ) — INTEGER array *Input*

On entry: IQ must be unchanged from the value returned from a previous call to E01SGF.

6: LIQ — INTEGER *Input*

On entry: the dimension of the array IQ as declared in the (sub)program from which E01SHF is called.

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

7: RQ(LRQ) — ***real*** array *Input*

On entry: RQ must be unchanged from the value returned from a previous call to E01SGF.

8: LRQ — INTEGER *Input*

On entry: the dimension of the array RQ as declared in the (sub)program from which E01SHF is called.

Constraint: LRQ $\geq 6 \times$ M + 5.

9: N — INTEGER *Input*

On entry: n, the number of evaluation points.

Constraint: N ≥ 1.

10: U(N) — ***real*** array *Input*

11: V(N) — ***real*** array *Input*

On entry: the evaluation points (u_i, v_i), for $i = 1, 2, \ldots, n$.

12: Q(N) — ***real*** array *Output*

On exit: the values of the interpolant at (u_i, v_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 E01SHF returns with IFAIL = 3.

13: QX(N) — ***real*** array *Output*

14: QY(N) — ***real*** array *Output*

On exit: the values of the partial derivatives of the interpolant $Q(x, y)$ at (u_i, v_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 and QY are set to the largest machine representable number (see X02ALF), and E01SHF returns with IFAIL = 3.

15: 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 < 6,
or LIQ $< 2 \times$ M + 1,
or LRQ $< 6 \times$ M + 5,
or 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 E01SGF and E01SHF.

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 and QY have been set to the largest machine representable number (see X02ALF).

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Further Comments

The time taken for a call to E01SHF will depend in general on the distribution of the data points. If X and Y are approximately uniformly distributed, then the time taken should be only O(N). At worst O(MN) time will be required.

9 Example

See Section 9 of the document for E01SGF.