

NAG Library Routine Document

E01SHF

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

3 Description

E01SHF takes as input the interpolant $Q(x, y)$ of a set of scattered data points (x_r, y_r, f_r) , for $r = 1, 2, \dots, m$, as computed by E01SGF, and evaluates the interpolant and its first partial derivatives at the set of points (u_i, v_i) , for $i = 1, 2, \dots, n$.

E01SHF must only be called after a call to E01SGF.

This routine is derived from the routine QS2GRD described by Renka (1988).

4 References

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

5 Arguments

- | | | |
|----|---------------------------------|--------------|
| 1: | M – INTEGER | <i>Input</i> |
| 2: | X(M) – REAL (KIND=nag_wp) array | <i>Input</i> |
| 3: | Y(M) – REAL (KIND=nag_wp) array | <i>Input</i> |
| 4: | F(M) – REAL (KIND=nag_wp) array | <i>Input</i> |

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 | <i>Input</i> |
|----|-------------------------|--------------|

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

- | | | |
|----|---------------|--------------|
| 6: | LIQ – INTEGER | <i>Input</i> |
|----|---------------|--------------|

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 (KIND=nag_wp) array | <i>Input</i> |
|----|------------------------------------|--------------|

On entry: 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: $N \geq 1$.
- 10: U(N) – REAL (KIND=nag_wp) array *Input*
 11: V(N) – REAL (KIND=nag_wp) array *Input*
On entry: the evaluation points (u_i, v_i) , for $i = 1, 2, \dots, n$.
- 12: Q(N) – REAL (KIND=nag_wp) array *Output*
On exit: the values of the interpolant at (u_i, v_i) , for $i = 1, 2, \dots, n$. 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 (KIND=nag_wp) array *Output*
 14: QY(N) – REAL (KIND=nag_wp) array *Output*
On exit: the values of the partial derivatives of the interpolant $Q(x, y)$ at (u_i, v_i) , for $i = 1, 2, \dots, n$. 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. 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, $M < 6$,
 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).

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

Computational errors should be negligible in most practical situations.

8 Parallelism and Performance

E01SHF 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 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.

10 Example

See Section 10 in E01SGF.
