NAG Library Function Document
nag_2d_shep_eval (e01shc)

1 Purpose
nag_2d_shep_eval (e01shc) evaluates the two-dimensional interpolating function generated by
nag_2d_shep_interp (e01sgc) and its first partial derivatives.

2 Specification
#include <nag.h>
#include <nage01.h>

void nag_2d_shep_eval (Integer m, const double x[], const double y[],
const double f[], const Integer iq[], const double rq[], Integer n,
const double u[], const double v[], double q[], double qx[],
double qy[], NagError *fail)

3 Description
nag_2d_shep_eval (e01shc) 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 nag_2d_shep_interp (e01sgc), and evaluates the
interpolant and its first partial derivatives at the set of points \((u_i, v_i)\), for \( i = 1, 2, \ldots, n \).

nag_2d_shep_eval (e01shc) must only be called after a call to nag_2d_shep_interp (e01sgc).

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

4 References
scattered data ACM Trans. Math. Software 14 149–150

5 Arguments
1: \( m \) – Integer \hspace{1cm} \text{Input} \\
2: \( x[m] \) – const double \hspace{1cm} \text{Input} \\
3: \( y[m] \) – const double \hspace{1cm} \text{Input} \\
4: \( f[m] \) – const double \hspace{1cm} \text{Input} \\

On entry: \( m, x, y \) and \( f \) must be the same values as were supplied in the preceding call to
nag_2d_shep_interp (e01sgc).

5: \( iq[(2 \times m + 1)] \) – const Integer \hspace{1cm} \text{Input} \\

On entry: must be unchanged from the value returned from a previous call to nag_2d_shep_interp
(e01sgc).

6: \( rq[(6 \times m + 5)] \) – const double \hspace{1cm} \text{Input} \\

On entry: must be unchanged from the value returned from a previous call to nag_2d_shep_interp
(e01sgc).

7: \( n \) – Integer \hspace{1cm} \text{Input} \\

On entry: \( n \), the number of evaluation points. \\
Constraint: \( n \geq 1 \).
8: \( u[n] \) – const double

9: \( v[n] \) – const double

*Input*

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

10: \( q[n] \) – double

*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 nag_real_largest_number (X02ALC)), and nag_2d_shep_eval (e01shc) returns with fail.code = NE_BAD_INTERPOLANT.

11: \( qx[n] \) – double

*Output*

12: \( qy[n] \) – double

*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 nag_real_largest_number (X02ALC)), and nag_2d_shep_eval (e01shc) returns with fail.code = NE_BAD_INTERPOLANT.

13: \( \text{fail} \) – NagError *

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

### 6 Error Indicators and Warnings

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_INTERPOLANT**

On entry, 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 nag_real_largest_number = \(\langle value\rangle\).

**NE_BAD_PARAM**

On entry, argument \(\langle value\rangle\) had an illegal value.

**NE_INT**

On entry, \(m = \langle value\rangle\).

Constraint: \(m \geq 6\).

On entry, \(n = \langle value\rangle\).

Constraint: \(n \geq 1\).

**NE_INTERNAL_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

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

See Section 3.6.6 in the Essential Introduction for further information.

**NE_INVALID_ARRAY**

On entry, values in \(iq\) appear to be invalid. Check that \(iq\) has not been corrupted between calls to nag_2d_shep_interp (e01sgc) and nag_2d_shep_eval (e01shc).
On entry, values in $\textit{rq}$ appear to be invalid. Check that $\textit{rq}$ has not been corrupted between calls to nag_2d_shep_interp (e01sgc) and nag_2d_shep_eval (e01shc).

**NE_NO_LICENCE**

Your licence key may have expired or may not have been installed correctly. See Section 3.6.5 in the Essential Introduction for further information.

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Parallelism and Performance

nag_2d_shep_eval (e01shc) 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 function. 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 nag_2d_shep_eval (e01shc) 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 nag_2d_shep_interp (e01sgc).