NAG Library Function Document

nag_3d_shep_eval (e01thc)

1 Purpose

nag_3d_shep_eval (e01thc) evaluates the three-dimensional interpolating function generated by nag_3d_shep_interp (e01tgc) and its first partial derivatives.

2 Specification

```
#include <nag.h>
#include <nage01.h>
void nag_3d_shep_eval (Integer m, const double x[], const double y[],
     const double z[], const double f[], const Integer iq[],
     const double rq[], Integer n, const double u[], const double v[],
     const double w[], double q[], double qx[], double qy[], double qz[],
     NagError *fail)
```

3 Description

nag_3d_shep_eval (e01thc) 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 nag_3d_shep_interp (e01tgc), and evaluates the interpolant and its first partial derivatives at the set of points (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$.

nag_3d_shep_eval (e01thc) must only be called after a call to nag_3d_shep_interp (e01tgc).

This function is derived from the function 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 Arguments

1: **m** – Integer *Input*

2: **x**[**m**] – const double *Input*

3: **y**[**m**] – const double *Input*

4: **z**[**m**] – const double *Input*

5: **f**[**m**] – const double *Input*

On entry: **m**, **x**, **y**, **z** and **f** must be the same values as were supplied in the preceding call to nag_3d_shep_interp (e01tgc).

6: **iq**[$(\mathbf{2} \times \mathbf{m} + \mathbf{1})$] – const Integer *Input*

On entry: must be unchanged from the value returned from a previous call to nag_3d_shep_interp (e01tgc).

7: **rq**[$(\mathbf{10} \times \mathbf{m} + \mathbf{7})$] – const double *Input*

On entry: must be unchanged from the value returned from a previous call to nag_3d_shep_interp (e01tgc).

8: **n** – Integer *Input*

On entry: n, the number of evaluation points.

Constraint: **n** ≥ 1.

9: **u**[**n**] – const double *Input*
10: **v**[**n**] – const double *Input*
11: **w**[**n**] – const double *Input*

On entry: **u**$[i-1]$, **v**$[i-1]$, **w**$[i-1]$ must be set to the evaluation point (u_i, v_i, w_i), for $i = 1, 2, \ldots, n$.

12: **q**[**n**] – double *Output*

On exit: **q**$[i-1]$ 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 nag_real_largest_number (X02ALC)), and nag_3d_shep_eval (e01thc) returns with **fail**.**code** = NE_BAD_POINT.

13: **qx**[**n**] – double *Output*
14: **qy**[**n**] – double *Output*
15: **qz**[**n**] – double *Output*

On exit: **qx**$[i-1]$, **qy**$[i-1]$, **qz**$[i-1]$ 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 nag_real_largest_number (X02ALC)), and nag_3d_shep_eval (e01thc) returns with **fail**.**code** = NE_BAD_POINT.

16: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_BAD_POINT

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**, **qy** and **qz** have been set to nag_real_largest_number: nag_real_largest_number = ⟨*value*⟩.

NE_INT

On entry, **m** = ⟨*value*⟩.
Constraint: **m** ≥ 10.

On entry, **n** = ⟨*value*⟩.
Constraint: **n** ≥ 1.

NE_INT_ARRAY

On entry, values in **iq** appear to be invalid. Check that **iq** has not been corrupted between calls to nag_3d_shep_interp (e01tgc) and nag_3d_shep_eval (e01thc).

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 How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

NE_REAL_ARRAY

On entry, values in **rq** appear to be invalid. Check that **rq** has not been corrupted between calls to nag_3d_shep_interp (e01tgc) and nag_3d_shep_eval (e01thc).

7 Accuracy

Computational errors should be negligible in most practical situations.

8 Parallelism and Performance

nag_3d_shep_eval (e01thc) 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' Notefor your implementation for any additional implementation-specific information.

9 Further Comments

The time taken for a call to nag_3d_shep_eval (e01thc) 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 $O(\mathbf{n})$. At worst $O(\mathbf{mn})$ time will be required.

10 Example

See Section 10 in nag_3d_shep_interp (e01tgc).