nag_2d_scat_eval (e01sbc)

1. Purpose

nag_2d_scat_eval (e01sbc) evaluates at given points the two-dimensional interpolant function computed by nag_2d_scat_interpolant (e01sac).

2. Specification

```
#include <nag.h>
#include <nage01.h>

void nag_2d_scat_eval(Nag_Scat_Struct *comm, Integer n, double px[],
            double py[], double pf[], NagError *fail)
```

3. Description

This function takes as input the parameters defining the interpolant $F(x, y)$ of a set of scattered data points (x_r, y_r, f_r), for $r = 1, 2, \ldots, m$, as computed by nag_2d_scat_interpolant (e01sac), and evaluates the interpolant at each of the points (px_k, py_k), for $k = 1, 2, \ldots, n$.

When **method** = **Nag_RC**, the derivatives stored in **comm** will be used to compute the interpolant if necessary. A triangle is sought which contains the point (px_k, py_k), and the vertices of the triangle along with the partial derivatives and f_r values at the vertices are used to compute the value $F(px_k, py_k)$. If the point (px_k, py_k) lies outside the triangulation defined by the input parameters, the returned value is obtained by extrapolation. In this case, the interpolating function F is extended linearly beyond the triangulation boundary. The method is described in more detail in Renka and Cline (1984) and the code is derived from Renka (1984).

Alternatively, if **method** = **Nag_Shep**, then all points that are within distance of (px_k, py_k), along with the corresponding nodal functions stored in **comm**, will be used to compute a value of the interpolant, if necessary.

nag_2d_scat_eval must only be called after a call to nag_2d_scat_interpolant (e01sac).

4. Parameters

comm
Pointer to a communication structure of type Nag_Scat_Struct which must be unchanged from the previous call of nag_2d_scat_interpolant (e01sac).

n
Input: the number of points at which the evaluation of the interpolant is required.
Constraint: $\mathbf{n} \geq 1$.

px[n]
py[n]
Input: the x- and y-coordinates of the kth point (px_k, py_k), for $k = 1, 2, \ldots, \mathbf{n}$, at which the interpolant is to be evaluated.

pf[n]
Output: the values of the interpolant evaluated at the points (px_k, py_k), for $k = 1, 2, \ldots, \mathbf{n}$.

fail
The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_NO_SETUP
The setup function nag_2d_scat_interpolant (e01sac) has not been called.

NE_SETUP_ERROR
The call to setup function nag_2d_scat_interpolant (e01sac) produced an error.

NE_STRUCT_CORRUPT
The structure ⟨*value*⟩ has been corrupted since the previous call to ⟨*value*⟩.

NE_INT_ARG_LT
On entry, **n** must not be less than 1: **n** = ⟨*value*⟩

NW_VALUE_EXTRAPOLATED
The evaluation point (⟨*value*⟩, ⟨*value*⟩) of (px, py) lies outside the triangulation boundary. The returned value, ⟨*value*⟩, of **pf** was computed by extrapolation.

NE_BAD_INTERPOLANT
On entry, the interpolant cannot be evaluted because the evaluation point (px, py) of (⟨*value*⟩, ⟨*value*⟩) is outside the support region of the input data points defined by **optional.rnw**= ⟨*value*⟩ as set in nag_2d_scat_interpolant (e01sac).

6. Further Comments

The time taken for a call of nag_2d_scat_eval is approximately proportional to the number of data points, m, used by nag_2d_scat_interpolant (e01sac).
The results returned by this function are particularly suitable for applications such as graph plotting, producing a smooth surface from a number of scattered points.

6.1. Accuracy

Computational errors should be negligible in most practical situations.

6.2. References

Franke R and Nielson G (1980) Smooth Interpolation of Large Sets of Scattered Data *Internat. J. Num. Methods Engrg.* **15** 1691–1704.
Renka R L (1984) Algorithm 624: Triangulation and Interpolation of Arbitrarily Distributed Points in the Plane *ACM Trans. Math. Softw.* **10** 440–442.
Renka R L and Cline A K (1984) A Triangle-based C^1 Interpolation Method *Rocky Mountain J. Math.* **14** 223–237.
Shepard D (1968) A Two-dimensional Interpolation Function for Irregularly Spaced Data *Proc. 23rd Nat. Conf. ACM* Brandon/Systems Press Inc, Princeton pp 517–523.

7. See Also

nag_2d_scat_interpolant (e01sac)

8. Example

See the example program for nag_2d_scat_interpolant (e01sac).