NAG Toolbox

nag_interp_2d_triang_bary_eval (e01eb)

1 Purpose

nag_interp_2d_triang_bary_eval (e01eb) performs barycentric interpolation, at a given set of points, using a set of function values on a scattered grid and a triangulation of that grid computed by nag_interp_2d_triangulate (e01ea).

2 Syntax

```
[pf, ifail] = nag_interp_2d_triang_bary_eval(x, y, f, triang, px, py, 'm', m, 'n', n)
[pf, ifail] = e01eb(x, y, f, triang, px, py, 'm', m, 'n', n)
```

3 Description

nag_interp_2d_triang_bary_eval (e01eb) takes as input a set of scattered data points (x_r, y_r, f_r), for $r = 1, 2, \ldots, n$, and a Thiessen triangulation of the (x_r, y_r) computed by nag_interp_2d_triangulate (e01ea), and interpolates at a set of points (px_i, py_i), for $i = 1, 2, \ldots, m$.

If the ith interpolation point (px_i, py_i) is equal to (x_r, y_r) for some value of r, the returned value will be equal to f_r; otherwise a barycentric transformation will be used to calculate the interpolant.

For each point (px_i, py_i), a triangle is sought which contains the point; the vertices of the triangle and f_r values at the vertices are then used to compute the value $F(px_i, py_i)$.

If any interpolation point lies outside the triangulation defined by the input arguments, the returned value is the value provided, f_s, at the closest node (x_s, y_s).

nag_interp_2d_triang_bary_eval (e01eb) must only be called after a call to nag_interp_2d_triangulate (e01ea).

4 References

Cline A K and Renka R L (1984) A storage-efficient method for construction of a Thiessen triangulation *Rocky Mountain J. Math.* **14** 119–139

Lawson C L (1977) Software for C^1 surface interpolation *Mathematical Software III* (ed J R Rice) 161–194 Academic Press

Renka R L (1984) Algorithm 624: triangulation and interpolation of arbitrarily distributed points in the plane *ACM Trans. Math. Software* **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

5 Parameters

5.1 Compulsory Input Parameters

1: **x**(**n**) – REAL (KIND=nag_wp) array

2: **y**(**n**) – REAL (KIND=nag_wp) array

The coordinates of the rth data point, (x_r, y_r), for $r = 1, 2, \ldots, n$. **x** and **y** must be unchanged from the previous call of nag_interp_2d_triangulate (e01ea).

3: **f**(**n**) – REAL (KIND=nag_wp) array

The function values f_r at (x_r, y_r), for $r = 1, 2, \ldots, n$.

4: **triang**(7 × **n**) – INTEGER array

The triangulation computed by the previous call of nag_interp_2d_triangulate (e01ea). See Section 9 in nag_interp_2d_triangulate (e01ea) for details of how the triangulation used is encoded in **triang**.

5: **px**(**m**) – REAL (KIND=nag_wp) array

6: **py**(**m**) – REAL (KIND=nag_wp) array

The coordinates (px_i, py_i), for $i = 1, 2, \ldots, m$, at which interpolated function values are sought.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the dimension of the arrays **px**, **py**. (An error is raised if these dimensions are not equal.)

m, the number of points to interpolate.

Constraint: **m** ≥ 1.

2: **n** – INTEGER

Default: the dimension of the arrays **x**, **y**, **f**. (An error is raised if these dimensions are not equal.)

n, the number of data points. **n** must be unchanged from the previous call of nag_interp_2d_triangulate (e01ea).

Constraint: **n** ≥ 3.

5.3 Output Parameters

1: **pf**(**m**) – REAL (KIND=nag_wp) array

The interpolated values $F(px_i, py_i)$, for $i = 1, 2, \ldots, m$.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: **n** ≥ 3.

ifail $= 2$

Constraint: **m** ≥ 1.

ifail $= 3$

On entry, the triangulation information held in the array **triang** does not specify a valid triangulation of the data points. **triang** has been corrupted since the call to nag_interp_2d_triangulate (e01ea).

ifail $= 4$

At least one evaluation point lies outside the nodal triangulation. For each such point the value returned in **pf** is that corresponding to a node on the closest boundary line segment.

ifail $= -99$

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

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

The time taken for a call of nag_interp_2d_triang_bary_eval (e01eb) is approximately proportional to the number of interpolation points, m.

9 Example

See Section 10 in nag_interp_2d_triangulate (e01ea).

9.1 Program Text

```
function e01eb_example

fprintf('e01eb example results\n\n');

% Scattered Grid Data
x = [11.16; 12.85; 19.85; 19.72; 15.91;  0.00; 20.87;  3.45; 14.26; ...
     17.43; 22.80;  7.58; 25.00;  0.00;  9.66;  5.22; 17.25; 25.00; ...
     12.13; 22.23; 11.52; 15.20;  7.54; 17.32;  2.14;  0.51; 22.69; ...
      5.47; 21.67;  3.31];
y = [ 1.24;  3.06; 10.72;  1.39;  7.74; 20.00; 20.00; 12.78; 17.87; ...
      3.46; 12.39;  1.98; 11.87;  0.00; 20.00; 14.66; 19.57;  3.87; ...
     10.79;  6.21;  8.53;  0.00; 10.69; 13.78; 15.03;  8.37; 19.63; ...
     17.13; 14.36; 0.33];
f = [22.15; 22.11;  7.97; 16.83; 15.30; 34.60;  5.74; 41.24; 10.74; ...
     18.60;  5.47; 29.87;  4.40; 58.20;  4.73; 40.36;  6.43;  8.74; ...
     13.71; 10.25; 15.74; 21.60; 19.31; 12.11; 53.10; 49.43;  3.25; ...
     28.63;  5.52; 44.08];
% Triangulate on (x,y)
[triang,ifail] = e01ea(x,y);
% Perform barycentric interpolation at (3.0,17.0)
px = 3;
py = 17;
[pf, ifail] = e01eb(x, y, f, triang, px, py);

fprintf('Interpolated value for f at (%4.1f,%4.1f) = %7.2f\n',px,py,pf);
```

9.2 Program Results

```
e01eb example results

Interpolated value for f at ( 3.0,17.0) =   39.05
```