

# NAG Library Routine Document

## E01EBF

**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

E01EBF 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 E01EAF.

### 2 Specification

```
SUBROUTINE E01EBF (M, N, X, Y, F, TRIANG, PX, PY, PF, IFAIL)
  INTEGER          M, N, TRIANG(7*N), IFAIL
  REAL (KIND=nag_wp) X(N), Y(N), F(N), PX(M), PY(M), PF(M)
```

### 3 Description

E01EBF takes as input a set of scattered data points  $(x_r, y_r, f_r)$ , for  $r = 1, 2, \dots, n$ , and a Thiessen triangulation of the  $(x_r, y_r)$  computed by E01EAF, and interpolates at a set of points  $(px_i, py_i)$ , for  $i = 1, 2, \dots, m$ .

If the  $i$ th 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  $(x_s, y_s)$ .

E01EBF must only be called after a call to E01EAF.

### 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 Arguments

1: M – INTEGER *Input*

*On entry:*  $m$ , the number of points to interpolate.

*Constraint:*  $M \geq 1$ .

2: N – INTEGER *Input*

*On entry:*  $n$ , the number of data points. N must be unchanged from the previous call of E01EAF.

*Constraint:*  $N \geq 3$ .

- 3: X(N) – REAL (KIND=nag\_wp) array Input  
 4: Y(N) – REAL (KIND=nag\_wp) array Input  
*On entry:* the coordinates of the  $r$ th data point,  $(x_r, y_r)$ , for  $r = 1, 2, \dots, n$ . X and Y must be unchanged from the previous call of E01EAF.
- 5: F(N) – REAL (KIND=nag\_wp) array Input  
*On entry:* the function values  $f_r$  at  $(x_r, y_r)$ , for  $r = 1, 2, \dots, n$ .
- 6: TRIANG( $7 \times N$ ) – INTEGER array Input  
*On entry:* the triangulation computed by the previous call of E01EAF. See Section 9 in E01EAF for details of how the triangulation used is encoded in TRIANG.
- 7: PX(M) – REAL (KIND=nag\_wp) array Input  
 8: PY(M) – REAL (KIND=nag\_wp) array Input  
*On entry:* the coordinates  $(px_i, py_i)$ , for  $i = 1, 2, \dots, m$ , at which interpolated function values are sought.
- 9: PF(M) – REAL (KIND=nag\_wp) array Output  
*On exit:* the interpolated values  $F(px_i, py_i)$ , for  $i = 1, 2, \dots, m$ .
- 10: 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,  $N = \langle value \rangle$ .  
 Constraint:  $N \geq 3$ .

IFAIL = 2

On entry,  $M = \langle value \rangle$ .  
 Constraint:  $M \geq 1$ .

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 E01EAF.

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.

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**

Not applicable.

## **8 Parallelism and Performance**

E01EBF is not threaded in any implementation.

## **9 Further Comments**

The time taken for a call of E01EBF is approximately proportional to the number of interpolation points,  $m$ .

## **10 Example**

See Section 10 in E01EAF.

---