
Subject: Re: contour and points

Posted by [TonyL](#) on Fri, 01 Jul 2011 01:11:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Jun 28, 9:13 pm, Wox <s...@nomail.com> wrote:

> On Tue, 28 Jun 2011 03:10:55 -0700 (PDT), Gray

>

> <grayliketheco...@gmail.com> wrote:

>> Hi all,

>

>> What's the easiest/best way to determine which of a set of points is

>> inside a contour (created with cgcontour)? Thanks!

>

>> --Gray

>

> Maybe I'm missing something, but what do you mean by "inside a

> contour"?

>

> You can check whether the point's elevation is greater than the

> contour level or less than the contour level. What's inside or outside

> depends on your definition but I suppose "inside" = "greater than" for

> topographic data.

Try this function inside.pro

```
; docformat = 'rst'
```

```
;+
```

```
; Determines if a point is inside a polygon.
```

```
;
```

```
; :Returns:
```

```
; 1 if the point is inside the polygon, 0 if outside the polygon
```

```
;
```

```
; :Params:
```

```
; x : in, required, type=float
```

```
; x coordinate of the point
```

```
; y : in, required, type=float
```

```
; y coordinate of the point
```

```
; px : in, required, type=fltarr(n)
```

```
; x coordinates of the polygon
```

```
; py : in, required, type=fltarr(n)
```

```
; y coordinates of the polygon
```

```
;-
```

```
FUNCTION Inside, x, y, px, py
```

```
; x - The x coordinate of the point.
```

```

; y - The y coordinate of the point.
; px - The x coordinates of the polygon.
; py - The y coordinates of the polygon.
;
; The return value of the function is 1 if the point is inside the
; polygon and 0 if it is outside the polygon.

    sx = Size(px)
    sy = Size(py)
    IF (sx[0] EQ 1) THEN NX=sx[1] ELSE RETURN, -1 ; Error if px not
a vector
    IF (sy[0] EQ 1) THEN NY=sy[1] ELSE RETURN, -1 ; Error if py not
a vector
    IF (NX EQ NY) THEN N = NX ELSE RETURN, -1 ; Incompatible
dimensions

    tmp_px = [px, px[0]] ; Close Polygon
in x
    tmp_py = [py, py[0]] ; Close Polygon
in y

    i = indgen(N) ; Counter
(0:NX-1)
    ip = indgen(N)+1 ; Counter (1:nx)

    X1 = tmp_px(i) - x
    Y1 = tmp_py(i) - y
    X2 = tmp_px(ip) - x
    Y2 = tmp_py(ip) - y

    dp = X1*X2 + Y1*Y2 ; Dot-product
    cp = X1*Y2 - Y1*X2 ; Cross-product
    theta = Atan(cp,dp)

    IF (Abs(Total(theta)) GT !PI) THEN RETURN, 1 ELSE RETURN, 0
END

```
