Subject: Re: Number of points within a rectangular region
Posted by wmconnolley on Sat, 01 Nov 2003 14:58:25 GMT
View Forum Message <> Reply to Message

Ben Tupper <btupper@bigelow.org> wrote:
> Isa Usman wrote:
>> I'm trying to get the indices to the points that fall within a rectangular
>> region. I'm aware of David Fanning's INSIDE routine that does this but
>> because I have got 100,000 points its going to involve a lot of looping. So
>> I decided to use the HIST_ND with reverse indices but its not giving me the
>> right results. What am I missing?

This is (I hope) a vectorised version of INSIDE.

-W.

```
FUNCTION inside, x, y, px, py, index=index

; Purpose: see if point is inside polygon
; Category: maths
; Input: x, y - [vector of] points
;        px,py - points defining polygon (will be closed automatically)
; Output: vector of 1's and 0's
;         OR
;         indicies of points inside (if /index is set)
; Author: "Bï¿½rd Krane" <bard.krane@fys.uio.no>
; Mods: wmc - make it work with x, y as vectors
; More-info: posted to comp.lang.idl-pvwave on Wed, 01 Apr 1998 12:26:38 +0200
; See-also: http://www.ecse.rpi.edu/Homepages/wrf/geom/pnpoly.html for another method,
possibly better
; This is better than my routine "is_inside"
; Note: reduce test from 1e-8 to 1e-4, since we are usually in single precision. In fact, "0.1"
;       would do as well.

@comm_error

    sx = size(px)
    sy = size(py)
    IF (sx[0] EQ 1) THEN NX=sx[1] ELSE message,'px not a vector'
    IF (sy[0] EQ 1) THEN NY=sy[1] ELSE message,'py not a vector'
    IF (NX EQ NY) THEN N = NX ELSE message,'Incompatible dimensions'

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

    i  = indgen(N,/long)                    ; indices 0...N-1
    ip = indgen(N,/long) + 1                   ; indices 1...N
```

```
    nn=n_elements(x)
    X1 = tmp_px(i)#replicate(1,nn)  - replicate(1,n)#reform([x],nn)
    Y1 = tmp_py(i)#replicate(1,nn)  - replicate(1,n)#reform([y],nn)
    X2 = tmp_px(ip)#replicate(1,nn) - replicate(1,n)#reform([x],nn)
    Y2 = tmp_py(ip)#replicate(1,nn) - replicate(1,n)#reform([y],nn)

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

    ret=replicate(0l,n_elements(x))
    i=where(abs(total(theta,1)) GT 0.01,count)
    if (count gt 0) then ret(i)=1
    if (n_elements(ret) eq 1) then ret=ret[0]

    if (keyword_set(index)) then ret=(indgen(/long,n_elements(x)))(where(ret eq 1))

    return,ret

END
```

-W.

--
William M Connolley | wmc@bas.ac.uk | http://www.antarctica.ac.uk/met/wmc/
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself