Subject: Re: determining if a point is "inside" or "outside" a shape
Posted by Med Bennett on Mon, 18 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I have solved this problem with the included two functions. It works by first creating a point outside the polygon, then creating a line from the outside point
to the test point. Then, the algorithm counts the number of intersections between the test line and the lines making up the polygon. An odd number indicates that the test point is inside the polygon, while an even number indicates a point outside the polygon. This method ensures a correct result even
in the case of polygons with concave sides, etc. The function inpoly requires the
line_int function, and returns an array with the same number of elements as the

number of data points, with a 1 for points inside the polygon and 0 for points outside the polygon. Hope that this helps and that you can decipher my lousy code. There is a problem if two adjacent points have the same X coord (slope of one line segment is infinite) - this needs to be fixed!

```
================================
function inpoly,polyxy,dataxy

; procedure to determine which points in dataxy
; fall inside a given polygon
; polygon is defined by series of vertices given
; in polyxy

polyxy = double(polyxy)
dataxy = double(dataxy)

;determine polygon min and max xand y values

pxmax = max(polyxy(0,*))
pxmin = min(polyxy(0,*))
pymax = max(polyxy(1,*))
pymin = min(polyxy(1,*))

;determine size of input arrays

dsize = size(dataxy)
if dsize(0) eq 1 then dpts = 1 else dpts = dsize(2)
datain = intarr(dpts)

psize = size(polyxy)
ppts = psize(2)
int = intarr(ppts)
```

```
; determine which points lie inside polygon bounding box

w = where((dataxy(0,*) lt pxmax) and (dataxy(0,*) gt pxmin) and $
  (dataxy(1,*) lt pymax) and (dataxy(1,*) gt pymin),c)

for i = 0,c-1 do begin ;loop through data

    testpt = [pxmin-(.1*(pxmax-pxmin)),pymax+(.1*(pymax-pymin))]
    l1 = [[testpt],[dataxy(*,w(i))]] ;line 1 is from testpt to data point

;loop through polygon edge segments - last point in polygon array must be same
as
first!

    for j = 0,ppts-2 do begin
      if ( polyxy(0,j) ne polyxy(0,j+1) or polyxy(1,j) ne polyxy(1,j+1) ) then
begin
        l2 = [[polyxy(*,j)],[polyxy(*,j+1)]] ;second line connects two vertices

        int(j) = line_int(l1,l2) ;calculate line intersection
      endif
    endfor

    nint = total(int) ;total no. of intersections

;even if point is outside polygon; odd if inside
    if (nint/2 eq fix(nint/2)) then datain(w(i)) = 0 $
                              else datain(w(i)) = 1

endfor
print,total(datain),' points inside polygon.'
return,datain

end

==================================
function line_int,l1,l2

; calculate line equations

; print,l1,l2

if ( l1(0,0) ne l1(0,1) ) then begin
  slope1 = (l1(1,1)-l1(1,0))/(l1(0,1)-l1(0,0))
  yint1 = l1(1,0) - slope1*l1(0,0)
;  print,'m1',slope1,'b1',yint1
endif else begin
  if ((l1(0,0) lt max(l2(0,*))) and (l1(0,0) gt min(l2(0,*)))) then return,1 $
```

```
  else return,0
endelse

if ( l2(0,0) ne l2(0,1) ) then begin
  slope2 = (l2(1,1)-l2(1,0))/(l2(0,1)-l2(0,0))
  yint2 = l2(1,0) - slope2*l2(0,0)
;  print,'m2',slope2,'b2',yint2
endif else begin
  if ((l2(0,0) lt max(l1(0,*))) and (l2(0,0) gt min(l1(0,*)))) then return,1 $
  else return,0
endelse

; if lines are parallel, no intersection

if (slope1 ne slope2) then begin

xintersect = (yint2-yint1)/(slope1-slope2)
;print,'xintersect',xintersect
endif else return,0

if ( (xintersect lt max(l1(0,*))) and (xintersect gt min(l1(0,*))) $
 and (xintersect lt max(l2(0,*))) and (xintersect gt min(l2(0,*)))) then
return,1
$
    else return,0


end
==================================
```

"Martin LUETHI GL A8.1 2-4092" wrote:

> Dear all
>
> Is there a simple and fast way to find the indices of array elements, which
> are inside of a boundary (in 2 dimensions). Let's say coord(npoints, 2) is an

> array of coordinates in the plane and bound(nbound, 2) is the array of a
> bounday polygon. The coordintes are not on a grid (otherwise one could use
> polyfillv (PV-Wave).
>
> Thank you for any suggestion!
>
> Martin
>
> --
> ==
> Martin Luethi Tel. +41 1 632 40 92

> Glaciology Section Fax. +41 1 632 11 92
> VAW ETH Zuerich
> CH-8092 Zuerich mail luthi@vaw.baum.ethz.ch
> Switzerland
> ==========

dmarshall@ivory.trentu.ca wrote:

> I have an interesting problem that I'm betting someone has solved already
> in some form or another.
>
> I have a set of x,y coordinates that describe a polygon, 2d in my case.
> How do I tell if another x,y point is inside the boundary of my polygon?
>
> I had an idea to take the center of my polygon and extend a line from it to
> the point and beyond, seeing if I cross the perimeter an odd number of
> times.
>
> Any other suggestions?
>
> Dave.
>
> David Marshall Physics Dept. Trent University
> dmarshall@remove.this.ivory.trentu.ca Peterborough Ontario Canada

---