Subject: Re: data inside a circle
Posted by btt on Tue, 21 Jun 2005 20:09:57 GMT
View Forum Message <> Reply to Message

kuyper@wizard.net wrote:
> Ben Tupper wrote:
> ...
>
>> I guess I am a bit late with this but the following is from the online
>> help to IDL 6.1
>>
>> " The IDLanROI::ContainsPoints function method determines whether the
>> given data coordinates are contained within the closed polygon region."
>>
>> So, you could define the circle boundary as the ROI and pass the points
>> to the object method.
>
>
> A circle is not a closed polygon. It can be approximated with arbitrary
> accuracy by a closed polygon with a sufficiently large number of sides.
> However, as long as you use a finite number of sides, there will always
> be a certain amount of inaccuracy in that approximation. The more sides
> you use, the slower the comparison; at some desired level of accuracy,
> it's quicker to perform the correct test for being inside a circle,
> than it is to test for being inside a polygon approximation to a
> circle.
>
> In any event, the key problem in this particular problem is not the
> test for being inside a single cirle; the problem is that the test is
> against a very large number of circles. Doing that efficiently in IDL
> is tricky; and it's not clear to me that IDLanROI::ContainsPoints helps
> address that problem.
>

Yes and yes. I get it now.

I have cobbled together a test procedure that shows, as you hint, that IDLanROI::ContainsPoints is not a light-footed solution.

Cheers,
Ben

```
*****START
PRO testHoop, $
   nIter = nIter, $  ;number of iterations
   nP = nP, $        ;number of scatter points
   nC = nC           ;number of points that make
```

```
                    ;up the circular polygon

if n_elements(nIter) EQ 0 then nIter = 2
if n_elements(nP) EQ 0 Then nP = 10000
if n_elements(nC) EQ 0 then nC = 1000
x = RANDOMU(s, nP[0])
y = RANDOMU(s, nP[0])
roi = OBJ_NEW('IDLanROI' )

   ;make the points on the circle
;(offset = [0,0] and radius = 1 to start)
points = ((2.0 * !Pi )/(nC[0]-1.0) ) * FINDGEN(nC[0])
xp = COS(points )
yp = SIN(points)

   ;the elapsed time over all iterations
dt = 0.0d

For i = 0, nIter[0]-1 do Begin

   start = systime(/sec)

   rxy = RANDOMU(s,3)

   cx = rxy[1] + rxy[0] * xP
   cy = rxy[2] + rxy[0] * yP
   roi->setproperty, data = transpose([[cx],[cy]])

   ok = roi->ContainsPoints(x,y)

   fini = systime(/sec)
   dt  += (fini-start)
   print, 'iter, radius, x0, y0 ', i,  rxy
EndFor

print, 'elapsed time (s)  = ', dt
print, 'time per iter (s) = ', dt/niter

OBJ_DESTROY, roi
end

*****END
```