Subject: Re: accelerate processing time
Posted by Jean[1] on Mon, 10 Jul 2006 17:29:54 GMT
View Forum Message <> Reply to Message

Hi,

I would try something like this, to reduce the load:
1) get your central point coordinates
x_coord = 12.2
y_coord = 5.4

2) find all the points from the x and y arrays that MAY be within your
"radio" distance

possiblesPoints = where(x eq x_coord + radio or x eq x_coord - radio or
y eq y_coord + radio or y eq y_coord - radio)

3) from this set of points, compute your distance in one step.

sqDistance = (x_coord - x[possiblesPoints])^2 + (y_coord -
y[possiblesPoints])^2

4) select the proper distance

goodDistance = where(sqDistance LE radio^2)

5) retreive the coordinates

neighborsX = x[possiblesPoints[goodDistance]]
neighborsY = y[possiblesPoints[goodDistance]]

Jean H.

m.goullant@gmail.com wrote:
> Hi all,
> I'm a newbie in IDL and in programming. I have the following question:
>
> I have an irregular point cloud (X,Y,Z), and i must calculate the
> euclidean distances several times because I need to do some iterations
> to get the final results:
>
> radio=mask /2
>
> FOR i=0L,N_ELEMENTS(z)-1 DO BEGIN
>
> distances=sqrt((x-x[i])^2+(y-y[i])^2) ; Euclidean distances
> neighbors=WHERE(distances LE radio)

>
> ENDFOR
>
> If I have 2 million floating points, with several iterations, the
> computing time will be very tedious.
>
> Can anyone advice me in a solution for this!
>
> Thank you! Best regards,
> Marie
>