Subject: Re: locate particles on a grid
Posted by aman on Tue, 22 May 2012 01:46:36 GMT
View Forum Message <> Reply to Message

On Monday, May 14, 2012 7:30:30 PM UTC-4, Craig Markwardt wrote:
> On Monday, May 14, 2012 4:35:29 PM UTC-4, aman wrote:
>> I wrote a code in IDL to locate 10,000 particles over a grid of
>> size : 512 x 512 using Kernel statistics, in particular Gaussian. This
>> program has 3 FOR loops with an exponential calculation. It is really
>> taking a long time to get me an output. Two days have already been
>> over and this code is still running. I am not sure if IDL is good for
>> this array size and 512x512 iterations.
>>
>> My question is whether it is IDL problem or my code problem. I
>> discussed it with other students and they suggested me to do this work
>> in FORTRAN. But I am still not very sure about this, as many
>> astronomers use IDL for their research.
>
> Next time, put some PRINT statements so that you have some progress information.
>
> I imagine you are looping over X and Y. Don't do that. Precompute X and Y as 2D arrays, and only loop over the 10000 count. IDL just as happily evaluates EXP(-X^2-Y^2) where X and Y are 2D arrays as when X and Y are scalars.
>
> This goes with the philosophy of doing a lot of operations per FOR loop iteration.
>
> Craig

Thanks for the reply. My code is "kind of" fixed, but it stills takes almost an hour to finish 512x512 grid. I would like to switch to 2K x 2K grid , but did not give it a try yet. Could you please explain the idea behind "loop over the count". I am getting used to IDL slowly, but it is still new to me. Here is my code:

****
part=20000 ; particle number
gsize=512. ; Number of grids
h=0.05 ; grid size (center to one corner)

p=fltarr(part,2)
gx=fltarr(gsize)
gy=fltarr(gsize)
dx=1.5/gsize
dy=0.8/gsize

for i=0,gsize-1 do begin
gx[i]=gx[i-1]+dx

```
gy[i]=gy[i-1]+dy
endfor
printf,2, gx,gy

grid=fltarr(gsize,2)

for k=0, gsize-1 do begin
for j=0, gsize-1 do begin
d=sqrt((p(*,0)-grid(j,0))^2+(p(*,1)-grid(j,1))^2)
printf,3,d
a=where(d le h, count)
if count ne 0 then begin
final(j,k)=final(j,k)+(15*(1-(d(a)/h)^2)^2/(16*h)) ;biweight statistics(kernel)
endif
printf,4,grid(j,0),grid(k,1),final(j,k)/part
endfor
printf,4, " "
endfor
end
```

****

I deleted opening/closing file lines and other comments to make it look simple.

19,5 Top

-Aman
