## Subject: Re: IDLy approach to splatting points on a grid?
Posted by Jonathan Dursi on Thu, 23 Nov 2006 12:04:09 GMT

Following up on my own post; very gauche, I know, my apologies.

Another approach which has crossed my fevered mind is to do this as a
sparse array multiplication, which has some very attractive properties
-- but again, I don't see how to construct the sparse array without
looping over the particles (or the grid cells).   Either way, it's not
calculating the final result which is causing me heartache (loopache?)
it's the building of the particle-cell `interaction list' / `sparse
array non-zeros' / or however you prefer to think of it.

  - Jonathan

--
Jonathan Dursi
ljdursi@gmail.com

## Subject: Re: IDLy approach to splatting points on a grid?
Posted by MarioIncandenza on Fri, 24 Nov 2006 18:52:18 GMT

Jonathan,

This problem sounds worth some attention. I'm missing a few details, so
I hope I don't obfuscate the thread. I am going to assume that 1) your
problem actually is 2-D, and 2) you really want square windows (not
radii? really?), and 3) each particle has BOTH a radius of influence
AND a value it contributes. These assumptions are probably wrong,
but...

I generally brainstorm these with the following thought experiment:

"If memory was no limitation, how would I solve the problem?"

In this case, like this:

particle_x; particle x-coords
particle_y; particle y-coords
particle_r; particle size of window (R=2 => 5x5)
particle_v; particle values
n_particles=n_elements(particle_r); number of particles
nx_grid; x-dim of grid
ny_grid; y-dim of grid
;make HUGE arrays
big_x = rebin(lindgen(nx_grid),nx_grid,ny_grid,n_particles)

```
offset_x = temporary(big_x) - $
          rebin(particle_x,nx_grid,ny_grid,n_particles)
big_y =  rebin(transpose(lindgen(ny_grid)),nx_grid,ny_grid,n_particle s)
offset_y = temporary(big_y) - $
          rebin(transpose(particle_y),nx_grid,ny_grid,n_particles)
 big_r=rebin(reform(particle_r,[1,1,n_particles]),nx_grid,ny_ grid,n_particles)
 big_v=rebin(reform(particle_v,[1,1,n_particles]),nx_grid,ny_ grid,n_particles)
; Make a binary array of "influenced' cells;
yesno = (offset_x le big_r) * (offset_y le big_r)
; That's easy to modify to use radii instead
answer= yesno * big_v
```

Obviously, memory limitations present a problem. But if you looped the
problem over values of R or V, you need only 3 giant arrays, and if you
chunked the problem according to what you could fit in memory, I expect
you could get something acceptably fast.

I'd like to see how this gets resolved. We all need more practice with
REBIN/REFORM magic.

Good luck,

Edward H.

---

## Subject: Re: IDLy approach to splatting points on a grid?
Posted by Jonathan Dursi on Fri, 24 Nov 2006 22:02:02 GMT
View Forum Message <> Reply to Message

Hi, Ed:

Thanks for the reply...

Your assumptions were:

> 1) your problem actually is 2-D, and
> 2) you really want square windows (not  radii? really?), and
> 3) each particle has BOTH a radius of influence AND a value it contributes.

#3 is completely right, your hunch in #2 that I really want round
windows is correct (but I'm willing to do the function evaluation and
get zero in the corners to simplify things), and for #1, it will often
be the case that this will be a 2d thing, but the ability to also
handle 3d would certainly be a big benefit.


I like your brainstorming approach; I hadn't `gone there' with the 3
indexed arrays because it's not feasible (in general there's on order a

million or so particles) but I think that's a good start; in
particular, I think that in that approach likely likes the nucleus of
how to do this in with sparse matricies.

It's straightforward to find out within which cell each particle lives,
so that instead of examining the whole grid, one could only consider
the window directly... but since some particles will have huge windows,
and most very small ones, I'm not sure that helps...   Still, the
answer's in there somewhere, the hour is just too late for me to come
up with it.   I'll play with this some more tomorrow.

  Jonathan


--
Jonathan Dursi
ljdursi@cita.utoronto.ca