Refinement in Carpet and Problems With Visualization

Jonah Miller[1,2]

[1] *Department of Physics, University of Guelph, Guelph, ON, Canada*
[2] *Perimeter Institute for Theoretical Physics, Waterloo, ON, Canada*[*]

In this article, I briefly discuss a particular example refinement situation in the Carpet AMR framework. I also discuss the problems I am having reading this situation into yt.

* jmiller@perimeterinstitute.ca

I. THE TOY PROBLEM

Consider a cubic domain, with

$$\begin{aligned} x_{min} = y_{min} = z_{min} = 0 \\ x_{max} = y_{max} = z_{max} = 240 \\ \Delta x = \Delta y = \Delta z = 8 \end{aligned} \tag{1}$$

for the coarsest grid. There are four refinement levels, under cell-centred refinement,[1] each starting at $x = y = z = 0$ and extending to a radius of half of the radius of the parent grid. We refine by a factor of two. In other words, the first grid level would have

$$\begin{aligned} x_{min} = y_{min} = z_{min} = 0 \\ x_{max} = y_{max} = z_{max} = 120 \\ \Delta x = \Delta y = \Delta z = 4 \end{aligned} \tag{2}$$

given that cell edges must align. Each grid has a layer of ghost cells 3 layers deep around it. Figure 1 shows the refinement levels on the domain. Note that the axes are not set correctly. (Note that figure 1 is a slice plot. This will be relevant later.)

II. THE REFINEMENT SITUATION

Here I describe the refinement situation and define the language used by Carpet. As one might expect, Carpet aligns the start of the grids at $x = y = z = 0$ and extends the ghost zones out into negative x, as shown in figure 2. However, Carpet independently decides the beginning *index* from which to begin counting grid cells, which is refinement-level local and which may change based on refinement level and Carpet's whims.

At each refinement level, Carpet chooses one "fiducial" cell from which to begin counting the cells at that level. On the coarsest level, this is always the first cell. But on the finer levels it may not be. For example, in figure 2, the first ghost cell begins at index 2. The cells at index 0 and 1 do not really exist and so we call them "fake" cells.

To tell us where the fiducial "first cell" resides on refined levels, Carpet provides a parameter

`offset`

which we call OF_i for the i^{th} level. OF_i tells us the displacement (in cell widths at the i^{th} level) between the cell centre of the first cell at level $i-1$ and the cell centre at level i.

To help ourselves use this parameter, we define the parameter

`gridstart`

or G_i for grid i, which defines the distance (in cell widths at level i) between the centre of the fiducial first cell and the left edge of the first cell of the coarsest grid (ghost or not). So grid 0 always has $G_0 = 0.5$.

The `gridstart` parameter for level i obeys the following formula

$$G_i = G_{i-1} * f_i + \mathrm{OF}_i, \tag{3}$$

where f_i is the refinement factor between level i and level $i-1$. For now we assume that

$$f_i = 2 \; \forall \; i.$$

To tell us how many fake cells exist between the fiducial first grid cell at each level and the first ghost cell, Carpet provides the

[1] One can set Carpet to use either cell-centred or vertex-centred refinement. I focus on cell-centred for now.

region

parameter, which we call R_i. The region parameter tells us the index, with respect to the first fiducial cell, that the first cell with allocated memory (ghost or not) begins. For example, in figure 2,

$$\begin{aligned} R_0 &= 0 \\ R_1 &= 2 \\ R_2 &= 7. \end{aligned} \tag{4}$$

Carpet also tells us the index, with respect to the fiducial first cell at level i, that the active region starts. This parameter, perhaps unsurprisingly, is called

active

or A_i. In our case, since the active regions all start at $x = 0$ and we have 3 ghost cells,

$$A_i = R_i + 3 \; \forall \; i,$$

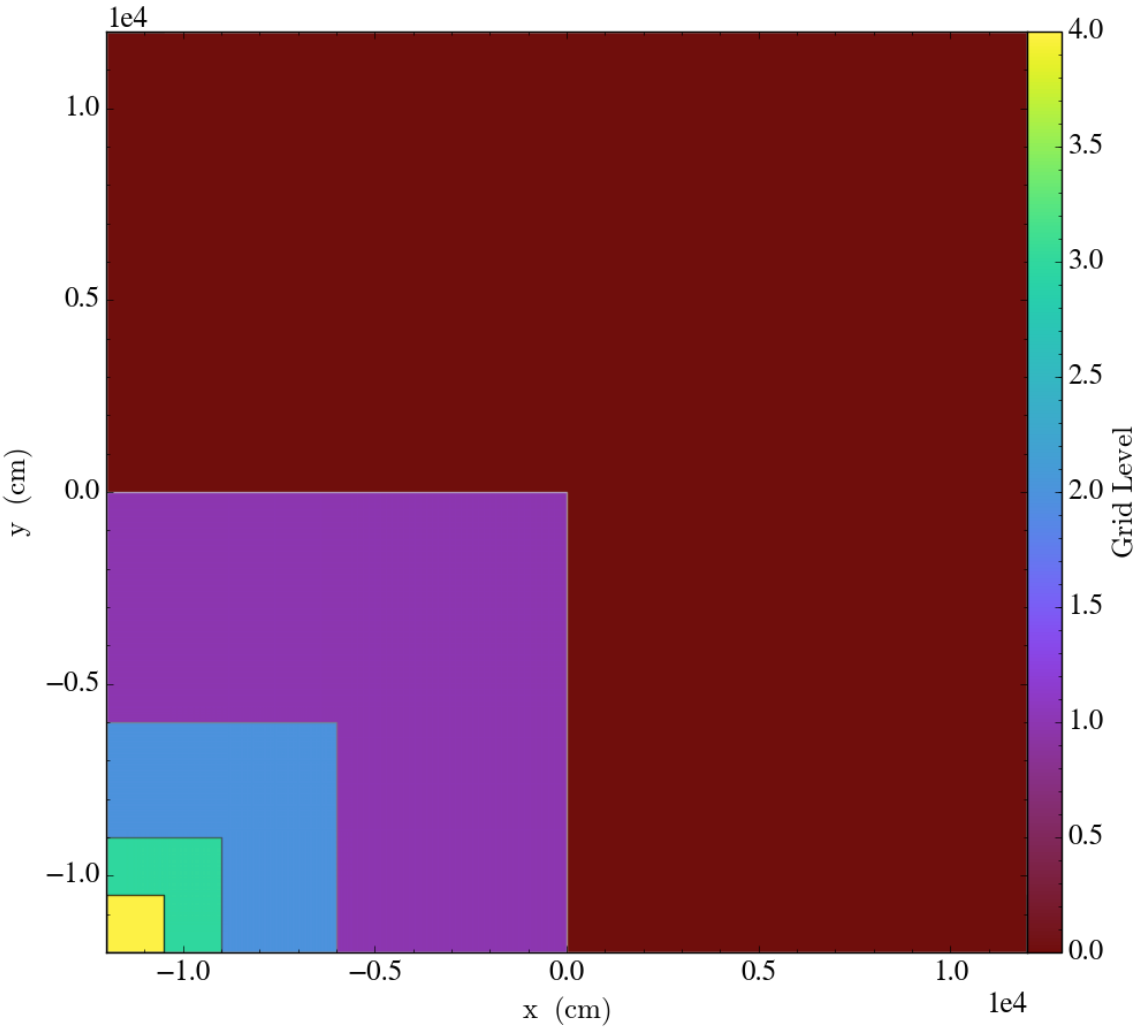

FIG. 1. The nested grids in our toy refinement problem. There are five grid levels, each nested within the parent. This is a sliceplot.

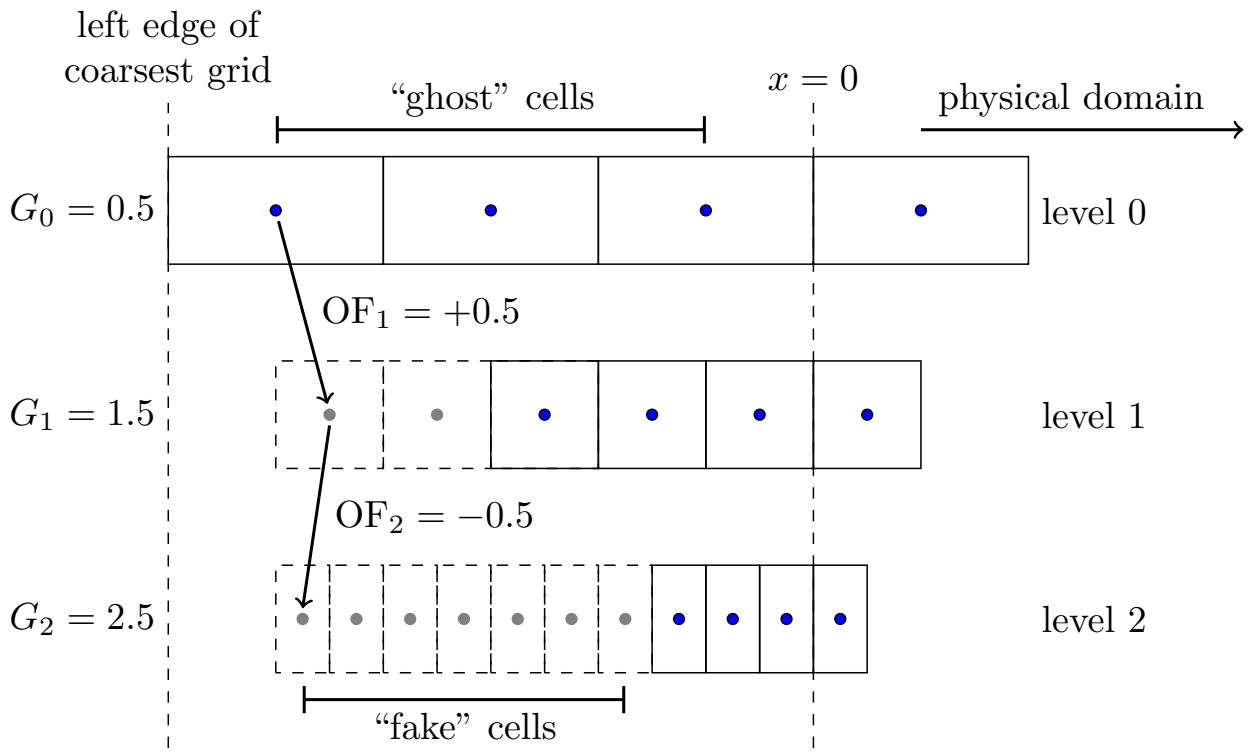

FIG. 2. The cell structure near refinement boundaries in Carpet, shown for the first three levels in our toy problem. Each level has three ghost cells. One physical cell is shown but many more exist. Fake cells are shown in grey.

but this isn't necessarily so.

Putting this all together, we have that the start index of the active region of each grid, with respect to the left edge of the coarsest grid, is

$$S_i := G_i + A_i - 0.5, \tag{5}$$

where the factor of 0.5 is the move from cell-centres to cell edges.

III. THE PROBLEM

I am having difficulty figuring out how to pass all this information to yt. I am currently setting the start index of each grid to S_i. I am (naively) setting all cells at level 0 to be the parents of all cells at level 1, etc. But for this example, that shouldn't matter.

I can safely perform slice plots, and the grids appear in the proper place. However, when I perform projection plots, my grids appear offset, and perhaps shrunken, as shown in figure 3.

IV. HELP?

If you have any suggestions, I would appreciate any help you can offer. My preliminary attempt at a frontend can be found here:
https://bitbucket.org/yt_analysis/yt/pull-requests/2121/wip-simulationio-frontend
and I have prepared test data here:
https://bitbucket.org/Yurlungur/simulationio-yt-tests/overview

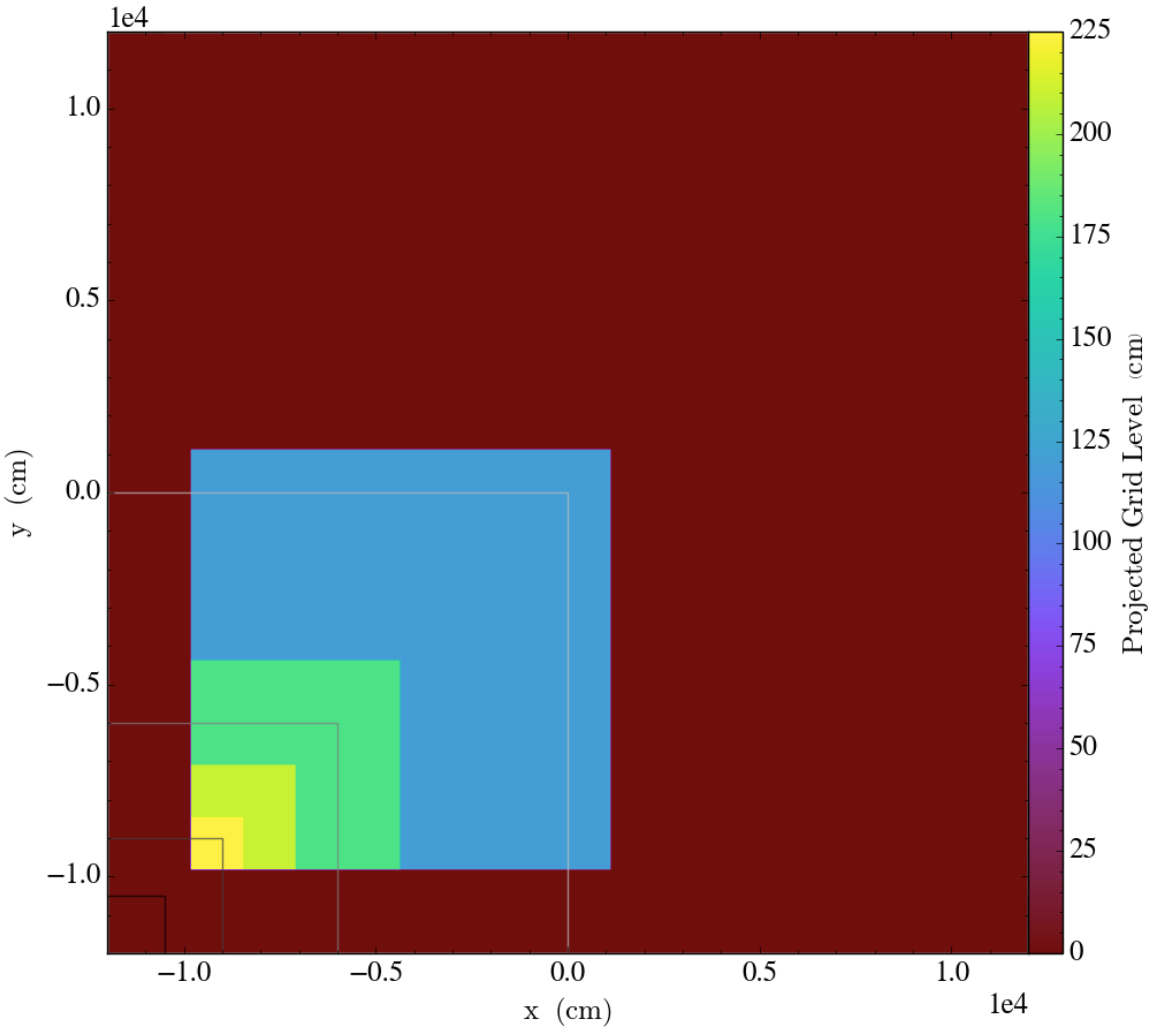

FIG. 3. A projection plot of the grid levels in yt using the grid structure described. There is an offset in the grid bounds, which I do not understand. The x and y axes are wrong, but this is a separate issue.