1 Running the Code

The code is compiled with the command "make smr_test," which produces the executable file "smr_test." The code can be run with the command "./smr_test." ("smr" stands for "static mesh refinement".)

2 Namelist Variables

N_grids: The number of grid sections.

N_vert: The number of grid cells on the underlying coarse grid. This value must be equal to the first value in "ref_levels."

ref_levels: A vector whose entries define the expanding grid configuration. The first entry must be equal to "N_vert" and the last entry must be equal to zero. The length of this vector is equal to N_grids+1. See section 3 for a description.

N_scalars: The number of prognostic variables: horizontal velocity, vertical velocity, equivalent potential temperature, and total water mixing ratio.

N_x: The number of horizontal grid cells on the coarsest grid section.

h_x/h_z: The horizontal and vertical grid spacing of the coarsest grid section.

rat: The refinement ratio. The default value is two, and this can't change.

Tend: The ending time in seconds.

dt: The time step in seconds used on the coarsest grid section.

write_freq: The frequency at which data is written in seconds.

file_name: The name of the NetCDF data file.

3 Grid Setup

Each grid section is discretized using a staggered grid. Horizontal velocity is stored at midpoints of vertical cell edges, vertical velocity is stored at midpoints of horizontal cell edges, and temperature and total water are stored at cell centers.

The namelist variable *ref_levels* determines the configuration of the expanding grid. The length of this vector is N_grids+1, where N_grids is the number of grid sections. Furthermore, the number of vertical grid cells of grid section i is 2^{i-1}(ref_levels(i)-ref_levels(i+1)).

To understand how to construct an expanding grid, we start with an underlying coarse grid. This grid covers the entire domain with N_vert vertical grid cells and N_x horizontal grid cells of length h_z and h_x, respectively. An example of the underlying coarse grid is illustrated on the left hand side of Figure 1, where we have ref_levels=(8,0). Now suppose we wanted to insert another grid section to this underlying coarse grid of vertical grid spacing h_z/2 and horizontal grid spacing h_x/2 at a height ih_z above the lower boundary, where i is an integer with $1 < i <$N_vert. To do this, we would set ref_levels=(8,i,0). If we let $i = 4$, we'd obtain the middle grid in Figure 1. This process could then be continued to insert any number of finer grid sections. In general, consider the case of M grid sections, labelled $G_1, \ldots, G_M$. Let N be the number of vertical grid cells of G_M, and let j be an integer such that $1 < j < N$. Inserting j between the M and $M+1$ entries of ref_levels results in an expanding grid whose lowest grid section extends from the bottom of the domain to a height jh_z/2^{M-1}. Note that we have chosen the index at which to insert the additional grid relative to the last, or finest, grid section. For example, the grid on the right of Figure 1 has ref_levels=(8,4,4,0).

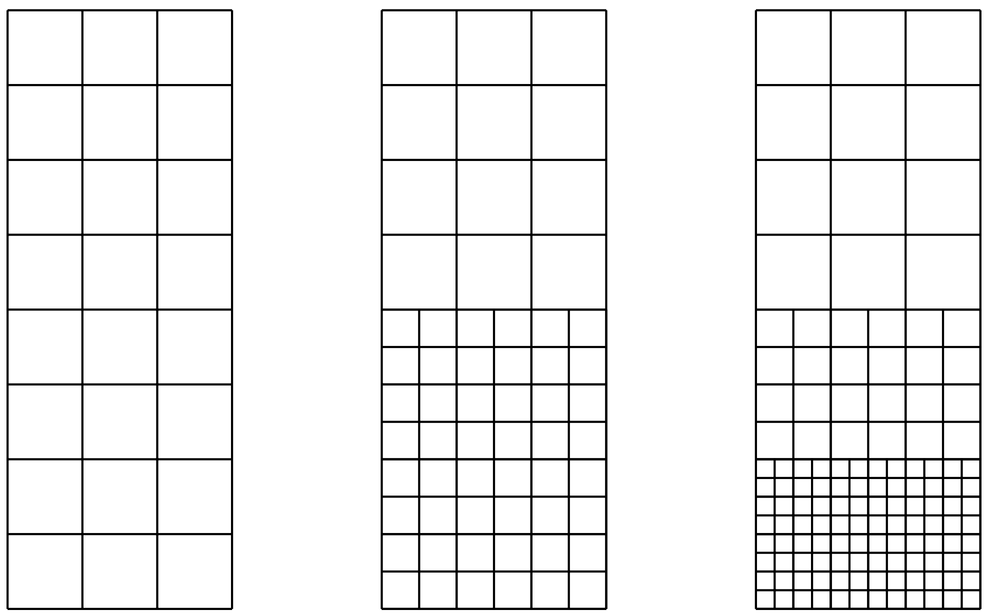

Figure 1: The underlying coarse grid (left), and expanding grids with two, and three grid sections (left, and right, respectively).

4 Output

NetCDF is used to store output from the code in a single file. Each grid section has its own set of variables. For instance, if we had four grid sections, we would have four horizontal velocity variables, one for each section, which are labelled u_1, u_2, u_3, and u_4.