Research Article Open Access

David H. Marsico* and Samuel N. Stechmann

Expanding Grids for Efficient Cloud Dynamics Simulations Across Scales

https://doi.org/10.1515/mcwf-2020-0101
Received February 22, 2020; accepted June 24, 2020

Abstract: With large eddy simulations (LES) and/or cloud-resolving models (CRMs), it is now possible to simultaneously simulate shallow and deep convection. However, using traditional methods, the computational expense is typically very large, due to the small grid spacings needed to resolve shallow clouds. Here, the main purpose is to present a method that is computationally less expensive by a factor of roughly 10 to 50. Unlike traditional grid stretching of only the vertical z grid spacing, the present method involves expansion of the grid spacing in all coordinate directions (x,y,z) and time t. A fine grid spacing of O(10)-O(100) m can be used near the surface to resolve boundary layer turbulence, and the grid spacing expands to be O(1000) m at higher altitudes, which reduces computational cost while still resolving deep convection. Example simulations are conducted with a simplified LES/CRM in 2D to verify the theoretical cost savings.

Keywords: mesh refinement, cloud resolving model, large eddy simulation, stretched grid, grid nesting

MSC: 76R10

1 Introduction

Clouds and convection occur across a range of scales, and the grid spacing needed in numerical simulations can be different for different cloud phenomena. For instance, a grid spacing of roughly 1 km is needed to simulate deep convection [5, 6], whereas a grid spacing of roughly 10 to 100 m is needed to simulate shallow clouds [25, 27].

To simulate both shallow and deep convection together, the computational expense is very large. Not only is a small grid spacing needed to resolve shallow clouds, but a large domain is also needed to encompass the scales of deep convective clouds. It is only somewhat recently that computational power has increased to a point where shallow and deep convection could be simulated together in large eddy simulations (LES) [8, 13, 18].

Here we present a setup that could bring a significant computational savings. The basic idea is illustrated in Fig. 1, and it involves the use of a fine grid spacing of roughly 10 to 100 m only in the lowest portions of the atmosphere where such high resolution is needed to resolve shallow clouds. At higher altitudes, the grid expands to a coarser grid spacing of roughly 1 km, thereby bringing a large computational savings, while still allowing deep convective clouds to be resolved. In contrast to the common approach of stretching of a grid in only the vertical z direction, the present method involves an expansion of the grid spacing in all coordinate directions x, y, z and time t, which brings additional computational savings.

The setup of the expanding grid could be useful in any setting where shallow clouds and deep convection are simulated together. For instance, while superparameterization techniques for climate models were

***Corresponding Author: David H. Marsico:** Department of Mathematics, University of Wisconsin-Madison, Madison, WI, USA, E-mail: marsico@wisc.edu
Samuel N. Stechmann: Department of Mathematics & Department of Atmospheric and Oceanic Sciences, University of Wisconsin-Madison, Madison, WI, USA, E-mail: stechmann@wisc.edu

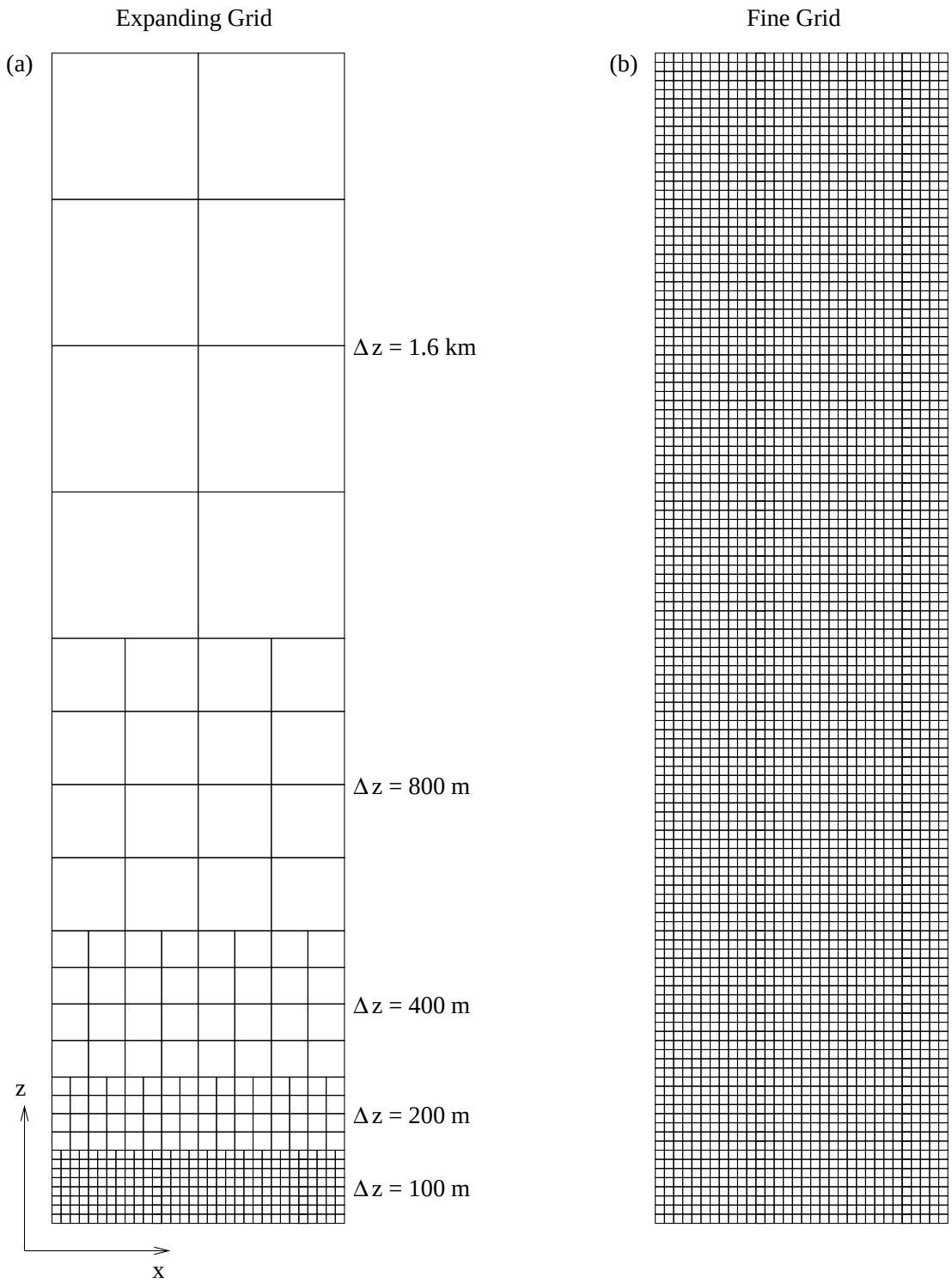

Figure 1: Schematic diagram of (a) the expanding grid and (b) the fine uniform grid. The expanding grid shown is a schematic diagram and has 5 grid sections, and each section has a height of 4 grid cells (except for the lowest grid section, which has 8 grid cells). In the numerical experiments, the expanding grid uses 6 grid sections, and each section has a height of 10 grid cells (except for the lowest grid section, which has 20 grid cells); and the finest and coarsest grid sections have grid spacings of 50 and 1600 meters, respectively.

originally targeted at resolving deep convection without resolving shallow clouds [4, 21], new configurations are currently being explored to resolve shallow clouds as well [3, 11, 19, 24]. Also, while it is perhaps too computationally expensive at the present time, in the future it is feasible to consider the possibility of resolving shallow clouds in weather prediction and global CRMs [22, 23, 32]. At the present time, for LES of shallow and deep convection over limited areas [8, 13, 18], grid nesting is often used to combine horizontal regions of different horizontal grid spacing; the expanding grid could possibly allow for additional computational savings that could be reinvested elsewhere, such as in cloud microphysics and subgrid-scale processes [7, 26, 30], or in larger domain sizes to include additional mesoscale or synoptic-scale variability, or in larger ensembles of simulations.

This paper is organized as follows. In section 2, we describe the expanding grid and the setup of the numerical simulations. In sections 3, 4, and 5, we compare numerical simulations of the expanding grid versus uniform grids with fine or coarse grid spacings. A setup with 2D simulations is used to allow comparison with fine-grid-spacing simulations at reasonably achievable computational cost. The numerical simulations also provide a verification of the theoretical estimates of computational cost savings, which are described in detail in the appendix.

2 Numerical Setup

The expanding grid is set up in the following way. The domain is decomposed into M grid sections, which we denote as G_i, for $1 \le i \le M$. See Fig. 1 for a schematic illustration of the setup (but note that the actual grid setup differs from Fig. 1 in some details, due to space limitations for a single page of the journal; these differences are described in the caption of Fig. 1). Each grid section has uniform grid spacing of h_i in all spatial directions (so $\Delta x_i = \Delta y_i = \Delta z_i = h_i$), with time step of Δt_i. The grids are refined in the sense that $h_i = h_{i-1}/2$, and $\Delta t_i = \Delta t_{i-1}/2$. For the specific tests of the present paper, we use 6 grid sections, where grid section G_1 extends from an altitude of 16 kilometers to 32 kilometers with a grid spacing of 1600 meters and a time step of 8 seconds. The thickness of grid G_i is then half the thickness of G_{i-1} for $i > 1$, so that G_6 uses $h_6 = 50$ m and covers the lower most 1 kilometer in the vertical direction. Other configurations are also possible by, e.g., changing the number of grid sections M, allowing different numbers of vertical grid cells per grid section, etc. We choose the present setup because it can be described easily in terms of powers of 2 and because it meets the basic requirements on grid spacings for simulating both shallow and deep convection (see section 1). For additional tests, we have also used setups with 5 and 7 grid sections, so that the cost of inserting or removing a fine grid adjacent to the bottom of the domain can be determined.

To implement an expanding grid setup, two options are as follows. First, one could use existing adaptive mesh refinement (AMR) software in a static configuration that remains the same at each time step, which would essentially amount to a nested grid setup [2, 29, 31]. One advantage of this is its flexibility, as AMR techniques can generally refine any region of the domain and can even use dynamic/adaptive mesh refinement criteria. A potential disadvantage would be the difficulty in incorporating AMR software into existing codes, so this approach may be best implemented in new codes that are designed from the start with AMR capabilities in mind, such as those used in [14–16, 28], and [10], where the focus has traditionally been on horizontal rather than vertical refinement. In AMR, each grid cell is often represented as the node of a tree, the depth of which is equal to the number of levels of refinement at a given point in space. So to incorporate AMR into an existing code, one would need to change the array data structures into these tree-based ones. Besides this additional difficulty, one would still expect an AMR code with the expanding grid defined in a statically refined configuration would offer the same speedup as the second approach.

As a second approach, which is taken in the present paper, modifications could be made to an existing code. In making the modifications, the basic idea is to leverage the fact that an existing code can solve the dynamics over an individual grid section, which is arranged as a regular array. What is needed in addition, then, is a wrapper code that calls the existing code to update each grid section, as needed. This calling procedure is then implemented as in AMR and grid nesting methods to move forward all grid sections one full

time step [1, 20]. For example, since grid section G_1 uses $\Delta t_1 = 8$ s and grid section G_2 uses $\Delta t_2 = 4$ s, grid section G_2 must advance two Δt_2 time steps for every one Δt_1 time step.

The data structures for an expanding grid require some special treatment, but can be accomplished in a simple way using native Fortran capabilities. In particular, the numerically expanding grid is not arranged in space as a regularly spaced array. Nevertheless, each of the individual grid sections *is* a regularly spaced array. Therefore, the many regularly spaced arrays can be collected together and stored using a Fortran derived data type, and individual grid sections can be accessed efficiently using pointers. Note that this setup could possibly be extended so that grid sections could be inserted or removed based on dynamic criteria by using Fortran's dynamic memory allocation, although the simpler static configuration is used for the present initial study.

As the standard case, the domain size is 32 kilometers in the vertical and 16 kilometers in the horizontal direction. To provide some additional tests, we also consider cases where the horizontal length of the domain is increased to be 24 km or 32 km. The simulations presented here are all 2D in order to allow for comparison with simulations with the fine uniform grid at a reasonable computational cost. Theoretical cost estimates will be presented below for both the 2D and 3D cases, in order to provide both a comparison with the 2D numerical simulations here and an estimate for expectations for 3D simulations. As comparison cases, in addition to the expanding grid, we also consider a uniform fine grid with a grid spacing of 50 meters and a time step of 0.25 seconds, and a uniform coarse grid with a grid spacing of 1600 meters and a time step of 8 seconds.

To test a model's ability to simulate both shallow and deep convection, numerical experiments are set up to simulate the transition from shallow to deep convection, similar to, e.g., [12]. We solve the equations for moist, non-precipitating atmospheric dynamics with phase changes of water between vapor and cloud water, with some details described in the appendix. To remain as simple as possible for these tests, a Boussinesq rather than anelastic setup is used [17]. The simple setup allows us to use a simple code, and the code is available on Zenodo with the DOI 10.5281/zenodo.3857736 to provide an illustration of how to modify an existing code in order to allow the capability of the expanding grid. To initialize the simulations, small amplitude random temperature perturbations are inserted near the surface. Prescribed large scale moistening and cooling are applied as forcing terms to the prognostic equations for the thermodynamic variables [5, 9], and a sponge layer is added to all variables in the upper part of the domain.

3 Shallow and deep convection together: What is cost of fine vs. expanding grid?

Fig. 2 shows a comparison of numerical simulations for the expanding grid and the uniform fine grid. Specifically, the horizontally-averaged liquid water mixing ratio is plotted, as a function of height z and time t. Both the expanding grid and fine grid show a gradual deepening in a transition from shallow convection to deep convection. This suggests that the expanding grid can simulate the same types of convective features as the fine grid.

While it is clear from Fig. 1 that the expanding grid is less expensive than the uniform fine grid, it is useful to have quantitative estimates of the cost savings. As theoretical estimates of cost savings, we find

$$\frac{N_f}{N_{ex}} \approx \frac{T_f}{T_{ex}} \approx 2^{M-1}, \tag{1}$$

where N_f and N_{ex} are the total number of grid points in the fine grid and expanding grid, respectively, and T_f and T_{ex} are the total time to carry out the simulations for the fine grid and expanding grid, respectively. Details of the estimates are shown in the appendix. The approximate estimate of 2^{M-1} holds for both the 2D and 3D cases. Note that the cost savings depends only on M, which is the number of grid sections (i.e., the number of times the grid is refined, plus one), and does not depend on the number of vertical grid cells per grid section. For the standard setup here with $M = 6$, the theoretical estimate of cost savings is then a factor

of 32 (or, based on a more refined estimate in the appendix, a factor of 27). What we find numerically is that it is 43 times faster than the fine grid, which suggests that the cost savings is even greater in practice than in the crude theoretical estimates. To provide some additional tests, we also compared the sensitivity of the time savings to different domains by performing additional simulations on domains with horizontal extents of 24 and 32 kilometers; we observe that the expanding grid is 41 and 56 times faster than the fine grid, respectively.

Based on the considerations above, we estimate that the expanding grid is 10 to 50 times faster than traditional fine grids. The lower estimate represents an expanding grid with $M = 6$ grid sections, as in the standard case considered here, with a grid spacing that expands from $h_6 = 50$ m to $h_1 = 1600$ m. While such a setup was seen here to offer a speedup of roughly 45 times, that comparison was relative to a uniform grid, so we decrease the estimate by a factor of 4 to offer a comparison with a vertically stretched grid. The upper estimate of 50 times, on the other hand, represents an expanding grid with $M = 8$ sections, which could be set up with a grid spacing that expands from $h_8 = 12.5$ m to $h_1 = 1600$ m. While the increase from $M = 6$ to $M = 8$ should come with an increased speedup from roughly 45 times to roughly 180 times, we again decrease the estimate (now to 50 times) to remain conservative and to offer a comparison to a stretched grid rather than a uniform grid. The choice of the number of grid sections may be somewhat empirical, and problem dependent. For instance, a domain with a large vertical extent, and therefore a wide range of scales, would benefit from a large number of grid sections, perhaps as many as 8. However, for domains with relatively small vertical extents, 5 or 6 grid sections may be sufficient. In any case, the expanding grid offers a substantial cost savings.

4 Refining a deep-convection grid to resolve shallow convection: How much added expense?

Another question of interest is, How much extra expense is needed to refine a deep-convection grid in order to resolve shallow convection? Theoretically, we expect the expanding grid to come with additional expense, beyond the coarse grid expense, by a factor of about 700. What we observe in numerical simulations is somewhat close to this: the expanding grid requires more computation time than the coarse grid by a factor of $O(1000)$ for the three different domain sizes. However, this factor of 700–1000 is small in a relative sense; if one instead refines the uniform coarse grid to the uniform fine grid, the additional expense is theoretically a factor of roughly 33,000.

The value of the expanding grid is in its additional realism, compared with the coarse grid, and in achieving that realism for the least added expense possible. In particular, in the uniform coarse grid, cloud formation does not occur until about 10 hours, and there is less of a gradual transition from shallow clouds to deep. See Figs. 2 and 3. Instead, there is a rapid build-up of liquid water starting at approximately 5 kilometers in altitude, associated with a maximum vertical velocity of 50 m/s. The cloud top then increases up until 12 hours, when it reaches its greatest height of 15 km. Beyond this time, the solution remains nearly constant in that the maximum vertical velocity is nearly zero and the clouds cover a constant 35% of the domain, with the maximum horizontally averaged liquid water staying at about 8.5 g/kg. The coarse grid is evidently very different than the expanding and fine grids, as the coarse grid does not resolve shallow clouds.

5 Which grid gives the most realistic simulation, given available resources?

In this section, we assume that there is a given amount of resources, and we ask for the grid that gives the most realistic simulations. To that end, we compare the expanding grid to the “equivalent uniform grid," or just the “equivalent grid” for short, which is defined as the uniform grid with the same number of grid cells as

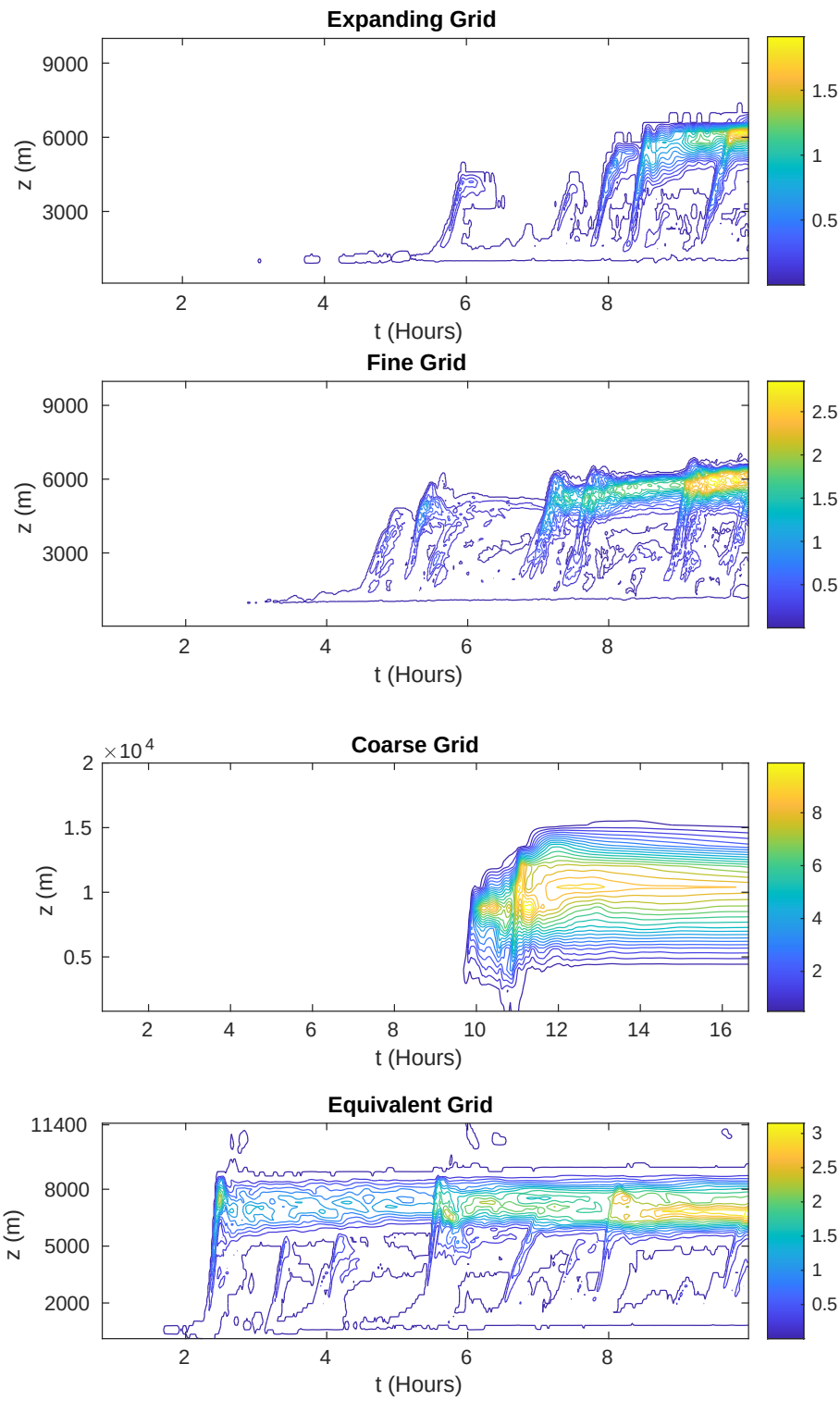

Figure 2: Comparison of simulations using four different grids. The horizontally averaged liquid water (g/kg) is shown for all four grids. Note that the coarse grid simulation lasts for approximately 17 hours, since it has a late initiation of clouds; the other cases are shown for 10 hours. The “equivalent grid" is defined in section 5.

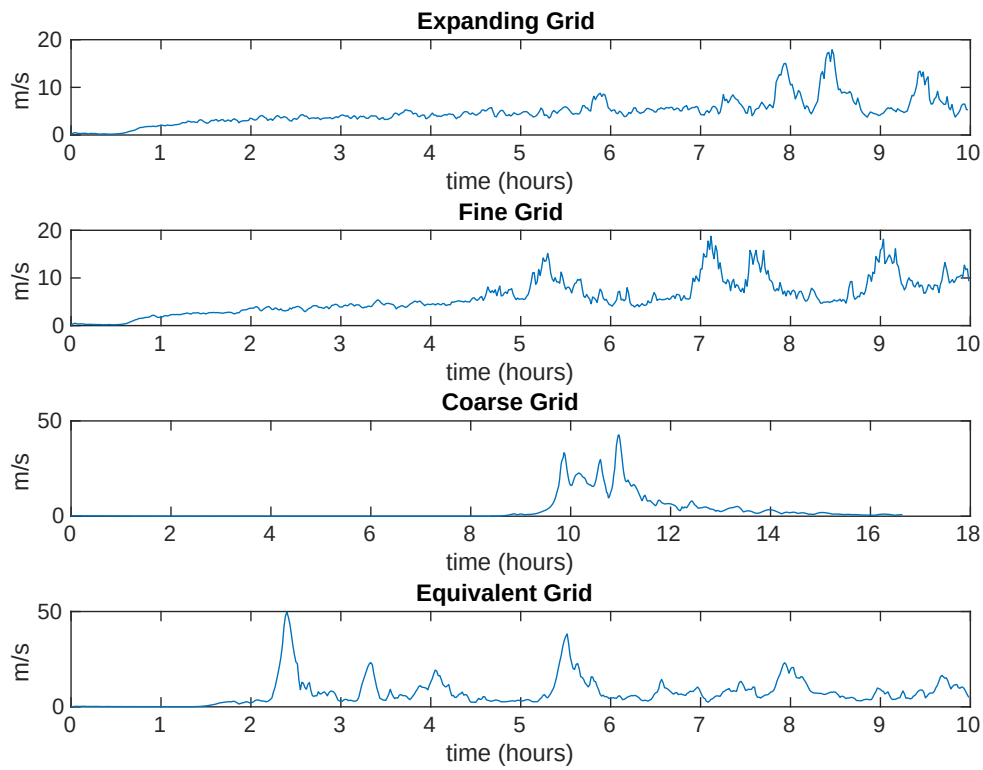

Figure 3: Same as Fig. 2, except for the maximum vertical velocity (m/s). Note that the coarse grid simulation lasts for approximately 17 hours, since it has a late initiation of clouds; the other cases are shown for 10 hours.

the expanding grid. This results in a uniform grid spacing of 232 meters. As the resolution is approximately four times as coarse as the fine grid, a time step of 1 second is used. Through this comparison, we test whether the grid arrangement of the expanding grid offers more realism in comparison to a uniformly arranged grid.

The horizontally averaged liquid water for the equivalent grid is shown in figure 2. Broadly speaking, the equivalent grid lacks the gradual transition from shallow to deep convection that was seen in the fine grid and expanding grid. Instead, in the equivalent grid, the formation of deep convection occurs somewhat abruptly. Cloud formation first occurs at a height of approximately 1 km at 2.5 hours, and the transition to deep convection occurs shortly after, with the cloud top at a height of about 9 km. Compared to the coarse grid, the equivalent grid features the same abrupt transition to deep convection, with an extremely strong updraft.

6 Conclusions

An expanding grid setup has been presented as a way of allowing efficient simulation of both shallow and deep convection. In particular, the expanding grid was estimated to be 10 to 50 times faster than traditional fine grids.

Both theoretical and numerical estimates were presented for the cost savings, and details were described in section 3 and the appendix. In comparing the expanding grid and a fine uniform grid, a rough theoretical estimate of the cost savings is a factor of 2^{M-1}, where M is the number of grid sections in the expanding grid (i.e., the number of levels of refinement, plus 1). In the standard configuration tested here with $M = 6$ grid

sections, the cost savings is then roughly estimated to be a factor of 32. The rough theoretical estimates of cost savings are the same for either 2D or 3D and for either number of grid points or total computation time. In practice, numerical examples showed an even greater speedup of a factor of 40 to 50.

Implementation of the expanding grid could be done in a number of ways. For instance, one could use AMR in either a static or dynamic/adaptive configuration. Here, as another option, we described how one can modify an existing code without needing full AMR capabilities. The expanding grid can be implemented in a straightforward way using a Fortran derived data type to store the expanding grid as a collection of regular arrays.

Some first comparisons were shown between the expanding grid and a uniform fine grid. Due to the large cost of the uniform fine grid, a simplified LES/CRM in 2D was explored as a feasible setup. The comparisons show similar gradual deepening from shallow convection to deep convection in both the expanding grid and the uniform fine grid, and the maximum vertical velocity in the two cases shows similar magnitudes and variability. It would be interesting in the future to make further detailed comparisons and in 3D to better understand the similarities and differences in variability on the expanding grid versus a uniform fine grid. It would also be interesting to consider other configurations of the expanding grid, by trying, e.g., different refinement altitudes or different numbers of vertical grid cells for each grid section.

A Appendix

In this section, we derive theoretical estimates for the computational speedup offered by the expanding grid. The derivation is split into two parts: first the number of grid cells is calculated, and second the computation time is calculated by also estimating the number of time steps.

First we derive an expression for the total number of grid cells in the expanding grid. It will be a function of the total number of grid sections, M, and the number of grid cells on the coarsest section in the vertical and horizontal direction, which we denote N_{ex}^z, and N_{ex}^x. The 2D case will be considered first for simplicity, and the 3D case is presented later below. Here, we assume that in the expanding grid, the vertical extent of each grid section is half that of the previous grid section, except the height of the lowest grid section is the same as that of the previous grid section. Grid section i then has $2^{i-1}N_{ex}^x N_{ex}^z$ grid cells for $i < M$, and grid section M has $2^M N_{ex}^x N_{ex}^z$ cells. Summing over the total number of grid sections yields the expression

$$N_{ex} = N_{ex}^x N_{ex}^z (2^M + 2^{M-1} - 1), \tag{2}$$

where N_{ex} is the total number of grid cells of the expanding grid.

The number of grid cells in the uniform coarse grid, N_c, and uniform fine grid, N_f, can be determined in terms of N_{ex}^x. If we denote the width of the domain by L_x, and the height by L_z, and assume that $\Delta x = \Delta z$, then the area of a single coarse grid cell is $(L_x/N_{ex}^x)^2$, and the area of a single fine grid cell is $[L_x/(2^{M-1}N_{ex}^x)]^2$. Dividing the total area of the domain by the area of a single grid cell yields

$$N_c = \frac{L_z (N_{ex}^x)^2}{L_x}, \tag{3}$$

$$N_f = \frac{L_z (2^{M-1} N_{ex}^x)^2}{L_x}. \tag{4}$$

Now we determine estimates for the amount of time each grid will take for a single time step. We assume that the coarse grid has a time step of Δt, and the fine grid has a time step of $\Delta t/2^{M-1}$. For a regularly spaced grid, we compute time estimates according to the formula

$$\text{Time} = T_1 \times (\text{number of grid cells}) \times (\text{number of time steps per coarse time step}), \tag{5}$$

where T_1 is a dimensional constant with units of time. (In a more sophisticated calculation, we could replace T_1 with the function $T(i)$, where i is the grid section number, and $T(i)$ is the time complexity for grid section

i. So theoretically, we could have different results if there is significant variation in $T(i)$ for different grid sections.) With equation 5, we find that

$$T_c = T_1 N_c, \tag{6}$$

$$T_f = 2^{M-1} T_1 N_f, \tag{7}$$

where N_c and N_f are given by equations (3), and (4), respectively.

For the expanding grid, we apply equation (5) to each grid section individually and then sum over the number of grid sections. Note that grid section i takes 2^{i-1} time steps per coarse time step. With this in mind, we have

$$T_{ex} = 2^{M-1}(2^{M-1} N_{ex}^x N_{ex}^z) T_1 + \sum_{i=1}^{M} 2^{i-1}(2^{i-1} N_{ex}^x N_{ex}^z) T_1. \tag{8}$$

This simplifies to the expression

$$T_{ex} = \frac{N_{ex}^x N_{ex}^z}{12} \left(7 \cdot 4^M - 4\right) T_1. \tag{9}$$

In comparisons of computational cost, it is ratios that are needed. Based on the formulas above, one can determine the ratios

$$\frac{N_f}{N_{ex}} = \frac{2^{2M-1}}{3 \cdot 2^{M-1} - 1} \approx \frac{1}{3} 2^M \tag{10}$$

and

$$\frac{T_f}{T_{ex}} = \frac{3 \cdot 2^{3M}}{7 \cdot 4^M - 4} \approx \frac{3}{7} 2^M. \tag{11}$$

In arriving at these ratio formulas, one must use the fact that $\Delta x = \Delta z$ within each grid section and that $(1/2)(L_z/N_{ex}^z) = L_x/N_{ex}^x$. (It is $(1/2)L_z$ that arises, not L_z, since the height of the coarsest grid section is one-half the height of the entire domain.) Two noteworthy items are the following. First, notice that the cost speedup is roughly the same ($\approx 2^{M-1}$) for both the ratio of the number of grid cells, N_f/N_{ex}, and the ratio of the computation times, T_f/T_{ex}. Second, notice that these ratios depend only on the number of grid sections, M, and not on the number of vertical grid cells per grid section.

The choice $\Delta x = \Delta z$ was made for simplicity, but our results are actually the same for anisotropic grids with the same assumptions. We illustrate this with an example by calculating the number of grid cells on expanding and fine grids. The grid spacings for each grid section on a two-dimensional expanding grid can be written as $L_x/2^{M-1}N_{ex}^x$ and $L_z/2^{M-1}N_{ex}^z$, and the total number of grid cells is still given by equation (2). The total number of grid points on the fine grid is then $2^{2M-2}N_{ex}^x N_{ex}^z$. Comparing this with equation (2) indicates that cost savings in the anisotropic case only depends on the number of grid sections.

We can easily generalize to three dimensions. In this case, grid section i has $4^{i-1}N_{ex}^x N_{ex}^y N_{ex}^z$ grid boxes for $i < M$, and grid section M has $4^{M-\frac{1}{2}}N_{ex}^x N_{ex}^y N_{ex}^z$, where N_{ex}^y is the number of grid boxes in the y direction on the coarsest grid section. One can show that

$$N_{ex}^{3d} = \frac{1}{3} N_{ex}^x N_{ex}^y N_{ex}^z (4^M + 3 \cdot 4^{M-1} - 1), \tag{12}$$

$$N_c^{3d} = \frac{L_y L_z (N_{ex}^x)^3}{(L_x)^2}, \tag{13}$$

$$N_f^{3d} = \frac{L_y L_z (2^{M-1} N_{ex}^x)^3}{(L_x)^2}. \tag{14}$$

The time estimates are similar, and we arrive at the equations

$$T_{ex}^{3d} = \frac{1}{7} N_{ex}^x N_{ex}^y N_{ex}^z \left(8^M + 7 \cdot 8^{M-1} - 1\right) T_1, \tag{15}$$

$$T_c^{3d} = T_1 N_c^{3d}, \tag{16}$$

$$T_f^{3d} = 2^{M-1} T_1 N_f^{3d}. \tag{17}$$

Based on the formulas above, one can determine the 3D ratios

$$\frac{N_f^{3d}}{N_{ex}^{3d}} = \frac{3 \cdot 2^{3M-2}}{7 \cdot 4^{M-1} - 1} \approx \frac{3}{7} 2^M \tag{18}$$

and

$$\frac{T_f^{3d}}{T_{ex}^{3d}} = \frac{7 \cdot 2^{4M-3}}{15 \cdot 8^{M-1} - 1} \approx \frac{7}{15} 2^M. \tag{19}$$

The derivation is similar to the 2D case. Notice that the cost speedups are roughly the same ($\approx 2^{M-1}$) and roughly the same as in the 2D case. In fact, the 2D ratio T_f/T_{ex} is exactly the same as the 3D ratio N_f^{3d}/N_{ex}^{3d}, which is reasonable because they both represent three-dimensional grids: (x, z, t) in the former case and (x, y, z) in the latter.

We end by noting that, thus far, we have compared the expanding grid to uniform grids. While some LES and CRM setups do use uniform grid spacings [18, 25], it is also common to use a vertically stretched grid, where the horizontal grid spacings remain fixed but the vertical grid spacing is stretched (e.g., it may be 50 m near the surface and stretched to be 500 m near the tropopause). So we remark briefly about the savings we could expect relative to a vertically stretched grid. We compare to the vertical structure of the grid in [13], where 256 vertical levels are used, and the smallest vertical grid spacing is 50 meters. Assuming that the stretched grid has a uniform horizontal resolution of 50 meters, we find that the three dimensional expanding grid is 12 times faster, which is a reduction in the speedup by a factor of roughly 3 relative to the estimate of $2^{M-1} = 32$ that arises when comparing against a uniform grid.

B Appendix

In this section, we describe our model in more detail, and provide a discussion of some aspects of its implementation. We solve the moist, non-precipitating Boussinesq equations, which are

$$\frac{D\mathbf{u}}{Dt} = -\nabla\phi + b\hat{\mathbf{z}} \tag{20a}$$

$$\frac{D\theta_e}{Dt} + \frac{d\tilde{\theta}_e}{dz} w = 0 \tag{20b}$$

$$\frac{Dq_t}{Dt} + \frac{d\tilde{q}_t}{dz} w = 0 \tag{20c}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{20d}$$

where p' is the pressure, ρ_0 is a constant background density, $\phi = p'/\rho_0$, $\mathbf{u}(\mathbf{x}, t)$ is the velocity vector, θ_e is the equivalent potential temperature anomaly, and q_t is the anomalous total water mixing ratio. The buoyancy, b, is defined as

$$b = g\left(\frac{\theta}{\theta_0} + R_{vd} q_v - q_l\right), \tag{21}$$

where $\theta_0 = 300$ K is a constant background potential temperature, $g = 9.8$ m s^{-2} is the acceleration due to gravity and $R_{vd} = \left(R_v/R_d\right) - 1 = 0.61$, where R_d is the gas constant for dry air and R_v is the gas constant for water vapor.

Simple parameterizations were chosen for the initial tests shown here. For example, subgrid-scale turbulence was parameterized using eddy viscosity and eddy diffusion in the vertical and hyperviscosity and hyperdiffusion in the horizontal. Note that the parameterizations should be adapted to the grid spacing. For example, if a different subgrid-scale turbulence parameterization were used, such as a Smagorinsky or turbulent kinetic energy closure, the mixing length scale depends on the grid spacing, so one would need make

sure the appropriate length scales are used for each grid section. One set of physical processes that was not included is rainfall, ice, and associated cloud microphysics; instead, only phase changes between water vapor and cloud liquid water were included. Another physical process that was not included was radiative transfer. We point out, however, that a variety of approaches could potentially be used, depending on how much detail is desired for the radiation model. For instance, one could use all available grid information with a radiative transfer code that can be used with mesh refinement. As a less expensive alternative, one could average over the fine grid data and use a conventional radiative transfer model on coarsened columns. Our reason for using such a simplified setup was to demonstrate that basic features of multiscale convection could be reproduced in a more computationally efficient manner. It will be interesting in the future to investigate the effects of additional physical complexity.

For the numerical methods, we used a third order upwind discretization of the advection terms, and the pressure gradient and incompessibility constraint are treated using a projection method. The Poisson equation for the pressure was solved using an iterative method (conjugate gradient method), although one could also use Fourier transforms in the horizontal direction to yield a one-dimensional differential equation in the vertical direction, as is often done.

For passing information between different grid sections, the array for each variable has extra rows and columns to store ghost cells values to be used at physical boundaries and boundaries between grid sections. This is needed, for instance, for the advection terms. At grid section boundaries, the coarse grid variables are interpolated to provide these ghost cell values for the adjacent fine grid. For the Poisson equation for the pressure, rather than solving the Poisson equation over each grid section individually, we solve it over the desired grid section and all lower-altitude grid sections (albeit using a uniform grid spacing across all of those grid sections) in order to avoid the need for a pressure boundary condition at the grid section's lower boundary.

To store the data output, NetCDF or similar formats can be used, even for the expanding grid, since the expanding grid can be treated as a collection of regular arrays, and regular arrays can be stored efficiently and easily. In the data file, each variable is labelled with the grid section to which it corresponds. For example, in the expanding grid used in the numerical simulation in the paper, we have six horizontal velocity variables, each corresponding to a different grid section. These variables are denoted by "ui, " where i is the grid section number.

References

[1] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53(3):484–512, 1984.

[2] E. Goodfriend, F. K. Chow, M. Vanella, and E. Balaras. Improving large-eddy simulation of neutral boundary layer flow across grid interfaces. *Mon. Wea. Rev.*, 143(8):3310–3326, 2015.

[3] W. W. Grabowski. Towards global large eddy simulation: Super-parameterization revisited. *J. Meteor. Soc. Japan*, 94(4):327–344, 2016.

[4] W. W. Grabowski and P. K. Smolarkiewicz. CRCP: a Cloud Resolving Convection Parameterization for modeling the tropical convecting atmosphere. *Physica D: Nonlinear Phenomena*, 133:171–178, 1999.

[5] W. W. Grabowski, X. Wu, and M. W. Moncrieff. Cloud-resolving modeling of tropical cloud systems during Phase III of GATE. Part I: Two-dimensional experiments. *J. Atmos. Sci.*, 53:3684–3709, December 1996.

[6] W. W. Grabowski, X. Wu, M. W. Moncrieff, and W. D. Hall. Cloud-resolving modeling of cloud systems during Phase III of GATE. Part II: Effects of resolution and the third spatial dimension. *J. Atmos. Sci.*, 55(21):3264–3282, 1998.

[7] Wojciech W Grabowski, Hugh Morrison, Shin-Ichiro Shima, Gustavo C Abade, Piotr Dziekan, and Hanna Pawlowska. Modeling of cloud microphysics: Can we do better? *Bulletin of the American Meteorological Society*, 100(4):655–672, 2019.

[8] R. Heinze, A. Dipankar, C. C. Henken, C. Moseley, et al. Large-eddy simulations over Germany using icon: a comprehensive evaluation. *Q. J. Roy. Met. Soc.*, 143(702):69–100, 2017.

[9] G. Hernandez-Duenas, A. J. Majda, L. M. Smith, and S. N. Stechmann. Minimal models for precipitating turbulent convection. *J. Fluid Mech.*, 717:576–611, 2013.

[10] C. Jablonowski, M. Herzog, J. E. Penner, R. C. Oehmke, Q. F. Stout, B. van Leer, and K. G. Powell. Block-structured adaptive grids on the sphere: Advection experiments. *Mon. Wea. Rev.*, 134(12):3691–3713, 2006.

[11] F. Jansson, G. van den Oord, I. Pelupessy, J. H. Grönqvist, A. P. Siebesma, and D. Crommelin. Regional superparameterization in a global circulation model using large eddy simulations. *J. Adv. Model. Earth Syst.*, page in press, 2019.
[12] M. Khairoutdinov and D. Randall. High-resolution simulation of shallow-to-deep convection transition over land. *J. Atmos. Sci.*, 63(12):3421–3436, 2006.
[13] M. F. Khairoutdinov, S. K. Krueger, C.-H. Moeng, P. A. Bogenschutz, and D. A. Randall. Large-eddy simulation of maritime deep tropical convection. *J. Adv. Model. Earth Syst.*, 1(4), 2009.
[14] M. A. Kopera and F. X. Giraldo. Analysis of adaptive mesh refinement for imex discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *J. Comput. Phys.*, 275:92–117, 2014.
[15] M. A. Kopera and F. X. Giraldo. Mass conservation of the unified continuous and discontinuous element-based galerkin methods on dynamically adaptive grids with application to atmospheric simulations. *J. Comput. Phys.*, 297:90–103, 2015.
[16] S. Marras, M. A. Kopera, and F. X. Giraldo. Simulation of shallow-water jets with a unified element-based continuous/discontinuous Galerkin model with grid flexibility on the sphere. *Q. J. Roy. Met. Soc.*, 141(690):1727–1739, 2015.
[17] D. H. Marsico, L. M. Smith, and S. N. Stechmann. Energy decompositions for moist Boussinesq and anelastic equations with phase changes. *Journal of the Atmospheric Sciences*, page in press, 2019.
[18] L. Orf, R. Wilhelmson, B. Lee, C. Finley, and A. Houston. Evolution of a long-track violent tornado within a simulated supercell. *Bull. Amer. Meteor. Soc.*, 98(1):45–68, 2017.
[19] H. Parishani, M. S. Pritchard, C. S. Bretherton, M. C. Wyant, and M. Khairoutdinov. Toward low-cloud-permitting cloud superparameterization with explicit boundary layer turbulence. *J. Adv. Model. Earth Syst.*, 9(3):1542–1571, 2017.
[20] N. A. Phillips and J. Shukla. On the strategy of combining coarse and fine grid meshes in numerical weather prediction. *J. Atmos. Sci.*, 12(5):763–770, 1973.
[21] D. Randall, M. Khairoutdinov, A. Arakawa, and W. Grabowski. Breaking the cloud parameterization deadlock. *Bull. Amer. Meteor. Soc.*, 84:1547–1564, November 2003.
[22] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno, and S.-I. Iga. Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations. *J. Comput. Phys.*, 227(7):3486–3514, 2008.
[23] M. Satoh, B. Stevens, F. Judt, M. Khairoutdinov, S.-J. Lin, W. M. Putman, and P. Düben. Global cloud-resolving models. *Curr. Clim. Change. Rep.*, 5(3):172–184, 2019.
[24] T. Schneider, S. Lan, A. Stuart, and J. Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Lett.*, 44(24), 2017.
[25] Axel Seifert, Thijs Heus, Robert Pincus, and Bjorn Stevens. Large-eddy simulation of the transient and near-equilibrium behavior of precipitating shallow convection. *Journal of Advances in Modeling Earth Systems*, 7(4):1918–1937, 2015.
[26] Shin-ichiro Shima, Kanya Kusano, Akio Kawano, Tooru Sugiyama, and Shintaro Kawahara. The super-droplet method for the numerical simulation of clouds and precipitation: A particle-based and probabilistic microphysics model coupled with a non-hydrostatic model. *Quarterly Journal of the Royal Meteorological Society*, 135(642):1307–1320, 2009.
[27] A.P. Siebesma, C.S. Bretherton, A. Brown, A. Chlond, J. Cuxart, P.G. Duynkerke, H. Jiang, M. Khairoutdinov, D. Lewellen, C.H. Moeng, et al. A large eddy simulation intercomparison study of shallow cumulus convection. *J. Atmos. Sci.*, 60(10):1201–1219, 2003.
[28] W. C Skamarock and J. B. Klemp. Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.*, 121(3):788–804, 1993.
[29] A. Staniforth. Regional modeling: A theoretical discussion. *Meteorol. Atmos. Phys.*, 6(1):15–29, 1997.
[30] S. N. Stechmann. Multiscale eddy simulation for moist atmospheric convection: Preliminary investigation. *J. Comput. Phys.*, 271:99–117, 2014.
[31] P. P. Sullivan, J. C. McWilliams, and C.-H. Moeng. A grid nesting method for large-eddy simulation of planetary boundary-layer flows. *Boundary-Layer Meteorology*, 80(1):167–202, 1996.
[32] H. Tomita, H. Miura, S. Iga, T. Nasuno, and M. Satoh. A global cloud-resolving simulation: Preliminary results from an aqua planet experiment. *Geophys. Res. Lett*, 32(8), 2005.