

CS267 Homework 0

Lauren Goodfriend

February 7, 2013

I am a 4th year PhD student in environmental engineering. My research interests are boundary layer turbulence, large-eddy simulation, numerical methods for Navier-Stokes, and adaptive grids. After this class, I hope to be able to improve my research code.

I study large-eddy simulation of the atmospheric boundary layer on block structured non-uniform grids. This type of grid, which includes nested grids, is very popular in atmospheric science as a way of concentrating computational power. Large-eddy simulation is a widely used method of approximately solving incompressible Navier-Stokes in turbulent regimes, including the atmospheric boundary layer. These two techniques are well understood separately, but generate special errors at grid refinement interfaces when used together. I am trying to better understand and mitigate those errors.

My current work is studying simple test cases. I compare my results on a non-uniform grid to my results on uniform grids to test how the grid interface changed the results. For example, I have tested explicit filtering of the advective term of Navier-Stokes combined with a turbulence closure called the reconstruction method (Figure 1). When no explicit filtering is used, there is an unphysical peak in the vorticity magnitude at the grid interface. Explicit filtering reduces the size of this peak, and additional levels of reconstruction further reduce it. These results suggest that explicit filtering and reconstruction improve the grid transition.

Numerically and mathematically, I need to step forward an elliptic/hyperbolic PDE in time and solve a Poisson equation. I work on Fortran 90 research code parallelized with domain decomposition and MPI. The grid, including domain decomposition, is managed by the Paramesh package¹. I have been working on my advisor's 32 node Linux cluster, but I am moving to work primarily on a NCAR supercomputer, Yellowstone, which is #13 of the top 500. Both of these machines are composed of shared memory nodes connected into a larger distributed memory machine.

This application has been successful in its scientific objectives, but its use of parallelism could be greatly improved. Performance is problematic, particularly in the ghost cell filling subroutines and in the multigrid Poisson equation solver, which dominate the timing. Theoretically, wall time should decrease with increased numbers of processors, up to giving one block of the decomposed domain to each processor, since the blocks are fairly large. However, because of the nuances of the ghost cell filling and Poisson solving subroutines, the code achieves peak performance using 1/4 of this theoretical maximum number of useful processors. This performance suggests a bug in the MPI implementation.

The ghost cell filling bottleneck is unavoidable, because neighboring blocks must stop computation to exchange information several times within every timestep, but it may be possible to improve its performance. More information than necessary may be passed when filling ghost cells, and this waste of memory access could slow the code. The Poisson equation solver performance is limited both by ghost cell filling steps and the direct solve at the core of the multigrid algorithm. The direct solve is currently a FFT, which could definitely be improved by better use of FFTW, and may be a substandard parallel FFT algorithm.

I hope to improve the performance of my research code by streamlining the ghost cell filling subroutines and improving the direct solve used in multigrid. There is no theoretical reason why this code, which is a standard domain decomposition of a PDE, should not scale well. I am optimistic that the bugs currently preventing scalability can be fixed and allow this code to run large problems on large numbers of processors. Further improvement may be possible by using OpenMP to take advantage of shared memory within a single node.

¹http://www.physics.drexel.edu/~olson/paramesh-doc/Users_manual/amr.html

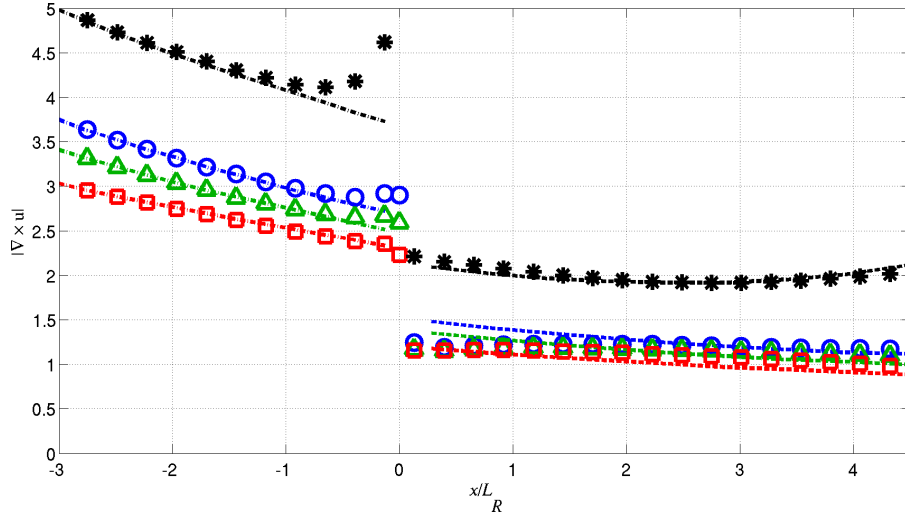


Figure 1: Results using a Gaussian filter, fine to coarse transition. The simulation was performed on a 96^3 grid adjoining a 48^3 grid; the grid interface is at $x = 0$. The plots are averages of 154 snapshots in time collected every 300 timesteps. The magnitude of the vorticity is plotted as averages in the yz -plane, varying with the direction of flow along the x -axis: *, no explicit filtering; \circ , zero-level reconstruction; \triangle , one-level reconstruction; \square , five-level reconstruction. Dashed lines represent uniform grid solutions.