1 Introduction

I am a first year PhD student in Applied Science and Technology and my research advisor is Phillip Colella. My research interest is in developing numerical methods for PDEs, specifically for fluid dynamics and plasma physics. Since much of the current technologies in numerics are parallelized, from this class I wish to learn about the tools used to write good parallel code.

2 Application: Adaptative Mesh Refinement

Adaptative mesh refinement (AMR) is a method for solving PDEs for problems that exhibit multiscale effects by using multiple meshes. Locally refined meshes are used and evolved in regions identified with the need for higher resolution (e.g., a shock in a fluid or boundary layer problems) so that the effects of the small scale features can be captured while coarser meshes are used in regions that do not exhibit the feature of interest. This allows for an effective use of computational resources. A key challenge in parallelizing AMR is effectively mapping data from the physical domain to the processors to minimize the communication between processors as the calculation proceeds (since neighboring meshes inevitably need to share some information with each other).

A parallel implementation of AMR is a software framework called Chombo. Chombo is a distributed memory code developed at LBL written in C++ running with fortran subroutines. Chombo has ran benchmark tests on Jaguar (rank 3 in 2011, upgraded to Titan in 2012; now ranked 2) in 2009 and achieved weak scaling.

References

[1] https://commons.lbl.gov/display/chombo