Zhen Cheng

Multigrid-Type Solvers for Simulations with Adaptive Mesh Refinement

4333 Siebel Center
201 N. Goodwin
Urbana,IL,61801-2302
zcheng@uiuc.edu
Eric de Sturler

The Interoperability Based Environment for Adaptive Meshes (IBEAM) is a NASA funded object-oriented framework for astrophysical simulations on high-performance, distributed memory, parallel computing platforms. In this project, we aim to solve radiation-hydrodynamic models of Gamma-Ray bursts. As such models require a high variation in the resolution of the computational grid, we use the PARAMESH package to support adaptive mesh refinement (AMR) on parallel machines. The PARAMESH package is developed by the Computational Technologies Team of NASA Goddard Space Flight Center.

PARAMESH was originally designed for explicit finite difference methods. A drawback for our problems is that excessively small time steps may be required for stability. To avoid this problem, we are implementing implicit methods on the PARAMESH package. These methods demand efficient solvers for large sparse linear systems. However, in principle PARAMESH refines and unrefines the grids at every time step. Moreover, PARAMESH stores the grids as a large collection of relatively small grid blocks (with pointers for neighbors, children and parent blocks), which are redistributed frequently over the processors for load balancing. The user has no control over the distribution of grid blocks over the processors. Finally, the matrix is typically not computed explicitly. These features make many preconditioners typically used with Krylov subspace methods difficult to implement and/or expensive to use, such as domain decomposition type preconditioners and ILU type preconditioners.

Multigrid is in principle well-suited for such an environment. However, typical problems may involve strong convection, jumps in coefficients, and strong anisotropy. Such features typically require robust versions of the multigrid algorithm. However, these features,too, may be difficult to implement and/or expensive. Therefore, we study a number of combinations of preconditioned Krylov subspace methods and multigrid methods. An important feature of our problems (and many other dynamic AMR discretizations) is that most refinements and unrefinements occur above a certain level (which may vary over the computational domain). Hence, we can often identify a (relatively high) level where changes in the grid are relatively rare. More expensive computations, such as computing a very good preconditioner, at this level can be amortized over many time steps.

The multigrid method consists of two main components: smoothing, to reduce high-frequency error, and coarse grid correction, to reduce low-frequency error. For hard problems such as diffusion-convection problems, problems with jumps in the coefficients, and problems with strong anisotropy, robust smoothers are needed. In the PARAMESH environment, line, and plane smoothers are hard to implement, relatively expensive, and (parallel) direct solvers for such subproblems may need to be recomputed at every time step. Therefore, we experiment with other block smoothers. For the coarse grid correction, the full approximate scheme (FAS) is used instead of the usual residual-error correction. This is required on AMR type meshes, because some parts of the domain are not covered by finer grids, so that we have to solve for the solution itself. In order to improve robustness in the coarse grid solves we experiment with a relatively high level 'direct' solver using a preconditioned Krylov method at levels where changes in the grid(s) are relatively infrequent. As preconditioner we consider explicit sparse approximate inverse preconditioners. These preconditioners are fairly insensitive to the redistribution of blocks over the processors, and can be updated for changes in the grid. Techniques for doing such updates will be presented by Shun Wang (UIUC) in a separate presentation. Sparse approximate inverses can also be used effectively in block smoothers.

Based on the designs discussed above, we have implemented several variations of multigrid/Krylov subspace solvers on PARAMESH, including multigrid with a direct solver on the coarsest grid level, multigrid with a Krylov subspace iterative solver on a specified grid level, and multigrid with a preconditioned Krylov iterative solver on a specified grid level, all with several types of block smoothers. Some of these variations are built on top of Krylov subspace methods. More specifically, rather than carrying out a V-cycle (or other scheme) to the coarsest grid level and then using a direct solver, we carry out the V-cycle (or other scheme) to some relatively finer grid level, and then use a Krylov subspace solver as a replacement for a direct solver on that level. We use an iterative solver instead of a direct solver here since the matrix is not stored explicitly, which makes a direct solver difficult without assembling the large sparse matrix. Furthermore, it would be hard to deal with the redistribution of grid blocks and the grid adaptations in a direct solver. Moreover, preconditioned iterative methods will typically achieve better parallel efficiency. Sparse Approximate inverse preconditioners are efficient for the same reasons as we mentioned above. Notice that these methods can also be considered as Krylov subspace methods at the more static grid levels enhanced with multigrid techniques to deal with the more dynamic grid levels.

Detailed experiment results will be presented and discussed in the talk.

The IBEAM project is sponsored under a Round III Grand Challenge Cooperative Agreement with NASA's Computational Technolgies Project.