Shun Wang

Approximate Inverse Preconditioners for Simulations with Adaptive Mesh Refinement

4333 Siebel Center
201 N. Goodwin
Urbana,IL,61801-2302
shunwang@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.

PARAMESH represents the computational grid by a list of many small grid blocks distributed for load balancing and minimizing communication. It carries out grid refinements and unrefinements at every time step. So, the system of equations changes (possibly) every time step. This makes certain preconditioners, like ILU, hard to realize in AMR. In addition, every preconditioner that requires a significant amount of preprocessing might become very expensive. Moreover, PARAMESH redistributes the grid blocks among the processors, possibly at every time step. The user does not have much control over this distribution. This makes preconditioners, like domain decomposition type preconditioners, less suitable for PARAMESH. A good preconditioner for AMR discretizations should be easy to compute and apply for a dynamically (re)distributed collection of grid blocks, and should easily accommodate frequent changes in the grids. Therefore, it should be easy to update when the mesh is changed. For all these reasons, explicit approximate inverse preconditioners turn out to be a good candidate for AMR based iterative solvers. In addition, explicit approximate inverse preconditioners are easy to update to accommodate small changes of the matrix due to adaptive time steps and nonlinear PDEs.

As PARAMESH applies refinements and unrefinements to the grid the matrix changes. The matrix may also change due to nonlinearities in the PDEs and possibly changes in time step. Frequently regenerating the approximate inverse from scratch would greatly increase the cost of computation. Therefore, we consider a number of techniques to limit these costs, mainly techniques to adapt an existing preconditioner for small changes in the matrix. The first one is to generate the approximate inverse only for newly refined blocks in every time step and keep the approximate inverse on the corresponding coarse blocks to be used after possible future unrefinements. The second one is to construct the approximate inverse for newly refined blocks cheaply from the existing coarse grid approximate inverse rather than generating it from scratch using, for example, the standard Frobenius norm minimization. We will discuss several other techniques for cheap adaptation.

Another important issue is the effectiveness in reducing the number of iterations. To be effective, the approximate inverse preconditioner needs to incorporate global information. We will use multigrid type techniques to incorporate global information in the preconditioner in a more efficient way than using Frobenius norm minimization. Though a multigrid solver itself may not work well for certain problems, see the presentation by Zhen Cheng (UIUC), multigrid-like techniques are well-suited for the PARAMESH data structure.

There is a trade-off between effectiveness and efficiency of approximate inverse preconditioners. The more accurate the approximate inverse is, the more effective the preconditioner will be, but also the more it will cost to compute the approximate inverse and apply it every iteration. There will be an optimal point. We will analyze this trade-off with experimental results.

At this point, we do not have the real model yet. So, our experiments are based on convection-diffusion type model problems. Our results indicate that approximate inverse preconditioners generally work well compared with other preconditioners built on PARAMESH, especially when the system is ill-conditioned. To provide a good benchmark for convergence, we will also compare the convergence of explicit approximate inverse preconditioners with other standard preconditioners, including some that are not easy to use in PARAMESH.

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