Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

GAMMS: a Multigrid–AMR code for computing gravitational fields

Claudio Gheller

CINECA, via Magnanelli 6/3 - 40033 Casalecchio di Reno BO (Italy)

Flavio Sartoretto, Michele Guidolin

Università di Venezia, Dipartimento di Informatica, Via Torino 155, Mestre VE (Italy)

Abstract.
This paper describes our GAMMS (Gravitational Adaptive Mesh Multigrid Solver) code, which is a numerical gravitational solver, based upon a multigrid (MG) algorithm and a local refining mesh strategy (AMR, Adaptive Mesh Refinement). MG allows for an efficient ad accurate evaluation of the gravitational field, while AMR provides high resolution computations exactly where needed. GAMMS is an open source code which takes advantage of the open source, based upon MPI, DAGH parallel library. Our code can be easily integrated with both N-body and fluid-dynamics code, in order to perform high resolution astrophysical simulations.

1. Introduction

Many astrophysical problems deal with the computation of gravitational forces. Usually, this task can be performed only by computer simulations. A large computational domain is to be treated, and good spatial resolution is required. Cosmological problems are distinguished examples. A cosmological structure formation process involves an extremely large dynamical range. As large as the universe volumes are featured (10^3 Mpc), yet one must solve the problem of resolving star formation regions in galaxies, which are typically at the 1 pc scale. The spatial dynamical range per dimension required is as large as 10^9, well beyond the maximum value, 10^3, that nowadays can be managed on supercomputers.

A solution to this range problem relies upon combining the Multigrid (MG) method (Brandt 1977, Suisalu & Saar 1995) with local refinement techniques. Multigrid solves differential equations using a set of multilevel grids, for an *optimal* $O(N)$ computational cost. Moreover, MG allows for efficiently tracking errors, and it can be effectively combined with local refinement techniques. We exploited Adaptive Mesh Refinement (AMR), which allows for dynamically defining and managing the computational meshes; they can be either refined, where high resolution is required, or coarsened, where low resolution is enough. Hence, the treatment of any interesting physical process is performed at its

© Copyright 2004 Astronomical Society of the Pacific. All rights reserved.

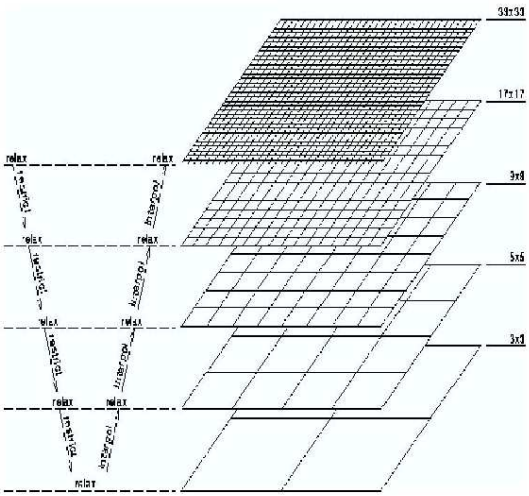

Figure 1. Sketch of an MG V-cycle on a grid hierarchy.

proper spatial resolution, whereas it would be too expensive by uniform grid treatment.

The ensuing code, GAMMS (Gravitational Adaptive Mesh Multigrid Solver) is described in the sequel.

2. Multigrid Algorithm

Let us consider Poisson's equation

$$\Delta u(\mathbf{x}) \;=\; \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbf{\Omega},$$

where $u(\mathbf{x})$ is the gravitational potential, $F(\mathbf{x})$ is the mass density field, Δ is the Laplacean operator and Ω is the problem domain. Our Multigrid technique estimates the solution by a Finite Difference (FD) scheme on a uniform, square grid (base grid) on Ω, thus yielding a linear system of equations. The problem solution is approximated by a Grid Function (GF), i.e. a value on each mesh node.

The linear system is solved using a relaxation method, which computes u on each node by considering only the adjacent ones, hence they efficiently reduce the error only at high frequency modes, while performing bad on smooth error components, which live at a *whole domain* scale. Slow convergence would emerge. On the other hand, large scale smooth errors can be read as local errors on coarser grids, where they can be smoothed out using the same relaxation technique. Hence, several discretization levels of the same continuous problem can be used to smooth out errors at different scales. This is the basic idea of MG, which enrolls a hierarchy of coarser and coarser grids to reduce the error at larger and larger scales, thus providing fast convergence. Traversing the hierarchy is called performing a *V-cycle*. Figure 1 sketches a sample V-cycle.

To go up and down along the grid hierarchy, two problem–dependent operators must be devised: a *restriction operator*, which allows to map onto a coarser

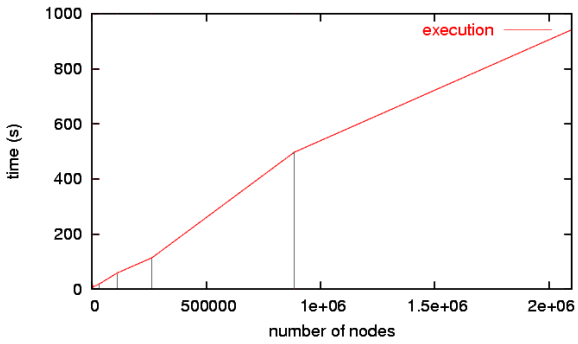

Figure 2. CPU seconds spent by MG vs number of grid points.

grid any GF given on a fine grid; a *prolongation operator*, in order to map any GF given on a coarse grid, onto a one–level higher, finer grid.

The problem is eventually solved only at a coarse grid level, the upper levels being resolved by exploiting the prolongation operator. Since coarsening produce smaller algebraic systems, the computational cost of solving a linear system on a coarse grid is low. One can prove that MG has an $O(N)$ computational cost, N being the number of grid nodes. MG can also easily manage locally refined grids.

3. AMR Technique

The AMR technique (Berger & Oliger 1984) allows for refining any region in the computational volume, by building a nonuniform grid, which is the union of uniform sub-grids G^0,...,G^M, each one featuring its mesh size $d_k = h^{k+1} = h^k/2$, $k = 1, \ldots, M$; d_0 being a given value.

Inside our GAMMS code, each adaptive mesh is built and controlled using the DAGH (Distributed Adaptive Grid Hierarchy) library (see `http://www.-caip.rutgers.edu/~parashar/DAGH`). DAGH provides data abstractions and all the tools required in order to define and manage grid values on a nonuniform, dynamically evolving mesh. DAGH is an open source package and features a parallel implementation, based upon a standard MPI library.

4. Results

At the present stage, MG and AMR are not fully combined. We tested them separately.

Tests on our MG implementation proved that the algorithm converges in few relaxation sweeps, when a moderate number of V-cycles is performed. Figure 2 shows that the computational cost of our MG code increases linearly with the number of grid points, as theory predicts.

Tests on our AMR code show that it can efficiently handle an adaptive mesh. Figure 3 shows a three–level local refinement driven by a Gaussian function, which was set initially on the coarsest, uniform grid. The grid was locally

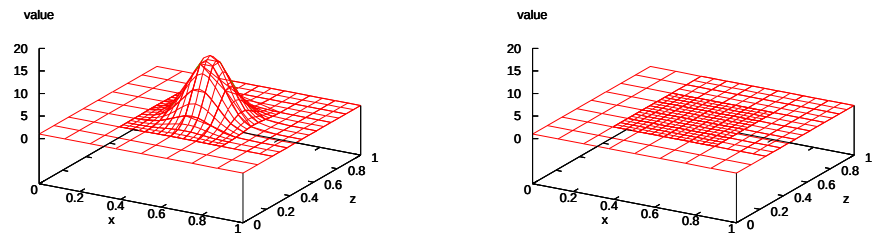

Figure 3. Hierarchical grid structure adapted to a Gaussian function. Left: the Gaussian function. Right: the final, locally refined mesh.

refined according to the magnitude of the derivative: larger magnitude requires finer resolution.

A detailed description of the codes and tests can be found in the Degree Thesis of R. Peruzzo (2003) and U. Zuccon (2003).

References

Brandt A., Math. Comp., 1977, 31, 333

Suisalu I., Saar E., 1995, MNRAS, 274, 287S

Berger M. J., Oliger J., 1984, J. Comp. Phys., 484, 54

Peruzzo R."Implementazione Object Orented Implementation of the AMR technique to calculate the gravitational field", Degree Thesis, 2003, University of Venice

Zuccon U. "Implementation of a 3D algorithm to calculate the gravitational field" Degree Thesis, 2003, University of Venice