Hierarchical Tree-Structures as Adaptive Meshes[*]

David J. Edelsohn

CRPC-TR91186

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

November 1991

Hierarchical Tree-Structures as Adaptive Meshes

David J. Edelsohn
Northeast Parallel Architectures Center
Syracuse University
Syracuse, NY 13244

Abstract

1 Introduction

Two basic types of simulations exist for modeling systems of many particles: grid-based (point particles indirectly interacting with one another through the potential calculated from equivalent particle densities on a mesh) and particle-based (point particles directly interacting with a one another through potentials at their positions calculated from the other particles in the system). Grid-based solvers traditionally model continuum problems, such as fluid and gas systems, and mixed particle-continuum systems. Particle-based solvers find more use modeling discrete systems such as stars within galaxies or other rarefied gases. Many different physical systems, including electromagnetic interactions, gravitational interactions, and fluid vortex interactions, all are governed by Poisson's Equation:

$$\nabla^2 \phi = -4\pi G \rho, \qquad (1)$$

for the gravitational case. To evolve N particles in time, the exact solution to the problem requires calculating the force contribution to each particle from all other particles at each timestep:

$$\mathbf{F}_i = \sum_{j \neq i}^{N} \frac{G m_i m_j (\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}. \qquad (2)$$

The $O(N^2)$ operation count is prohibitive for simulations of more than a few thousand particles commonly required to represent astrophysical and vortex configurations of interest.

One method of decreasing the operation count utilizes grid-based solvers which translate the particle problem into a continuum problem by interpolating the particles onto a mesh representing density and then solve the discretized equation. Initial implementations were based upon Fast Fourier Transform (FFT) and Cloud-in-Cell (CIC) methods which can calculate the potential of a mass distribution on a three-dimensional grid with axes of length M in $O(M^3 \log M^3)$ operations but at the cost of lower accuracy in the force resolution. All of these algorithms are discussed extensively by Hockney and Eastwood [12].

A newer type of grid-based solver for discretized equations classified as a multilevel or multigrid method has been in development for over a decade [7, 8]. Frequently the algorithm utilizes a hierarchy of rectangular meshes on which a traditional relaxation scheme may be applied, but multiscale methods have expanded beyond any particular type of solver or even grids, per se. Relaxation methods effectively damp oscillatory error modes whose wavenumbers are comparable to the grid size, but most of the iterations are spent propagating smooth, low-

wavenumber corrections throughout the system. Multigrid utilizes this property by resampling the low-wavenumber residuals onto secondary, lower-resolution meshes thereby shifting the error to higher wavenumbers comparable to the grid spacing where relaxation is effective. The corrections computed on the lower-resolution meshes are interpolated back onto the original finer mesh and the combined solutions from the various mesh levels determine the result.

Many grid-based methods for particle problems have incorporated some form of local direct force calculation, such as the particle-particle / particle-mesh (PPPM) method or the Method of Local Corrections (MLC), to correct the force on a local subset of particles. The grid is used to propagate the far-field component of the force while direct force calculations provide the near-field component either completely or as a correction to the "external" potential. The computational cost strongly depends on the criterion used to distinguish near-field objects from far-field objects. Extremely inhomogeneous systems of densely clustered particles can deteriorate to nearly $O(N^2)$ if most of the particles are considered neighbors requiring direct force computation.

A class of alternative techniques which have been implemented with great success utilize methods to efficiently calculate and combine the coefficients of an analytic approximation to the particle forces using spherical harmonic multipole expansions in three dimensions.

$$
\begin{aligned}
\phi_\gamma(\mathbf{r}) &= \int_{V_\gamma} d^3x\, G(\mathbf{r} - \mathbf{x})\rho(\mathbf{x}) \\
&= \sum_{n=0}^{\infty} \phi_{\gamma(n)}(\mathbf{r}) \qquad (3) \\
&= M_{\gamma(0)}G|_{\mathbf{r}-\mathbf{r}_\gamma} - M_{\gamma(1)}^{i}\partial_i G|_{\mathbf{r}-\mathbf{r}_\gamma} \\
&\quad + \frac{1}{2}M_{\gamma(2)}^{ij}\partial_i\partial_j G|_{\mathbf{r}-\mathbf{r}_\gamma}
\end{aligned}
$$

$$
-\frac{1}{6}M_{\gamma(3)}^{ijk}\partial_i\partial_j\partial_k G|_{\mathbf{r}-\mathbf{r}_\gamma}\cdots,
$$

where the multipole moments

$$
\begin{aligned}
M_{\gamma(n)}^{i_1\ldots i_n} &= \int_{V_\gamma} d^3x\, (\mathbf{x}-\mathbf{r}_\gamma)^{i_1}(\mathbf{x}-\mathbf{r}_\gamma)^{i_2} \\
&\quad \cdots (\mathbf{x}-\mathbf{r}_\gamma)^{i_n}\rho(\mathbf{x}), \qquad (4)
\end{aligned}
$$

V_γ are the disjoint spatial regions, and $G(r)$ is the Green's function. Instead of integrating G over the volume V_γ, one may compute the potential (and, in a similar manner, the gradient) at any position by calculating the multipole moments which characterize the density distribution in each region, evaluating G and its derivatives at $\mathbf{r}-\mathbf{r}_\gamma$, and summing over indices.

Not only does spatially sorting the particles into a tree-type data structure provide an efficient database for individual and collective particle information [18], but the various algorithms require and utilize the hierarchical grouping of particles and combined information to calculate the force on each particle from the multipole moments in $O(N\log N)$ operations or less.

Implementations for three dimensional problems frequently use an oct-tree — a cube divided into eight octants of equal spatial volume which contain sub-cubes similarly divided. The cubes continue to nest until, depending on the algorithm, the cube contains either zero or one particles or a few particles of equal number to the other "terminal" cells. Binary trees which subdivide the volume with planes chosen to evenly divide the number of particles instead of the space also have been used [3]; a single bifurcation separates two particles spaced arbitrarily close together while the oct-tree would require arbitrarily many sub-cubes refining one particular region. This approach may produce fewer artifacts by not imposing an arbitrary rectangular structure onto the simulation, but construction is more difficult and information about

2

each cut must be stored and used throughout the computation.

Initial implementations for both grid-based and multipole techniques normally span the entire volume with a uniform resolution net in which to catch the result. While this is adequate for homogeneous problems, it either wastes computational effort and storage or sacrifices accuracy for problems which exhibit clustering and structure. Many of the algorithms described above provide enough flexibility to allow adaptive implementations which can conform to complicated particle distribution or accuracy constraints.

2 Adaptive Structures

Mesh-based algorithms have started to incorporate adaptive mesh refinement to decrease storage and wasted computational effort. Instead of solving the entire system with a fixed resolution grid designed to represent the finest structures, local regions may be refined adaptively depending on accuracy requirements such as the density of particles. Unlike finite-element and finite-volume algorithms which deform a single grid by shifting or adding vertices, adaptive mesh refinement (AMR) algorithms simply overlay regions of interest with increasingly fine rectangular meshes. Berger, Colella, and Oliger have pioneered application of this method to hyperbolic partial differential equations [5, 6]. Almgren recently has extended AMR for multigrid to an MLC implementation [1].

Adaptive mesh refinement traditionally has been limited to rectangular regions. McCormick and Quinlan have extended their very robust, inherently conservative adaptive mesh multilevel algorithm called Asynchronous Fast Adaptive Composite (AFAC) [14] to relax non-rectangular sub-regions directly between two grid levels. The algorithm is a true mul-tiscale solver not limited to relaxation-type solvers. AFAC provides special benefits for parallel implementations because the various levels in a single multigrid cycle may be scheduled in any convenient order and combined at the end of the cycle instead of the traditional, sequentially-ordered V-cycle.

In the particle-based solver regime, the Barnes-Hut [4] method utilizes an adaptive tree to store information about one particle or the collective information about particles in the sub-cubes. Each particle calculates the force on itself from all of the other particles in the simulation by querying the hierarchical database, descending each branch of the tree until a user-specified accuracy criterion has been met. The accuracy is determined by the solid angle subtended by the cluster of particles within the cube from the vantage point of the particle calculating the force. If the cube contains a single particle or all of the particles in the cube can be approximated by the center of mass, the force is computed using a multipole expansion; otherwise, each of the eight sub-cubes is examined in turn using the same criterion. By utilizing combined information instead of the individual data at the terminal node of each branch, the algorithm requires $O(N \log N)$ operations.

The Fast Multipole Method (FMM) developed by Greengard and Rokhlin [11] utilizes new techniques to quickly compute and combine the multipole approximations in $O(N)$ operations. Initial implementations sorted the particles into groups on a fixed level of the tree with the hierarchical pyramid structure providing the communication network used to combine and re-propagate the multipole-calculated potential. Recent enhancements include adaptive refinement of the hierarchy creating structures similar to a Barnes-Hut tree [10].

Both Katzenelson and Anderson have noted the applicability of a variety of "tree algo-

rithms" to the N-body problem. Katzenelson utilizes the common structure of the Barnes-Hut and FMM algorithms to study how this problem can be mapped to a variety of parallel computer designs [13]. Anderson utilizes the multigrid framework as a basis for communication in his FMM implementation which substitutes Poisson integrals for spherical harmonic multipole expansions [2].

3 Tree as Grid

We propose that the exact same hierarchical structure used by particle-based methods now may be effectively utilized in adaptive mesh refinement implementation. The spatially structured cubic volumes into which the masspoints are sorted are inherently situated, sized, and ordered as an efficient adaptive mesh representing the system of interest. Instead of interpreting the hierarchy as a graphical representation of the tree-shaped database, it can function as the physical mesh which links the grid resolution with the particle density. Figures 1 and 2 represent a two-dimensional tree-structure from a particle simulation. Figures 3 and 4 show the configuration in figure 2 represented by a composite grid. The similarity between figures 2 and 3 demonstrates the convergence of these two different approaches. Tree levels and cells may not directly correspond with grid levels and zones, i.e. multiple particles (and cells) from multiple levels would be collected to form a single grid level of appropriate resolution aligned with the tree cells.

This relationship stems from the grid-based algorithms reliance on the locality of the discrete operator and the particle-based schemes similar utilization of locality to efficiently collect, combine, and redistribute the multipole moments. In the Poisson case, the locality stems from the regularity of harmonic functions which allow accurate approximation of
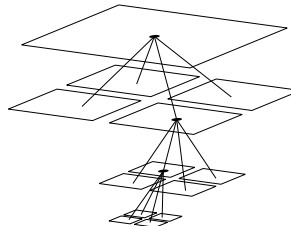

Figure 1: A perspective view of a 4-level 2-D tree including lines and dots representing parent-child relationships.

the smooth, far-field solution by low-order representations [1]. Barnes-Hut requires the locality of the tree not just as a framework for the algorithm but to provide the ability to selectively descend into sub-cubes as needed during the computation allowing Salmon to create "locally essential" datasets per processor [15]. Locality is common to and useful for many loosely synchronous parallel algorithms [9].

This union of hierarchies provides opportunities beyond similar programming structure [2, 13]: it allows easier synthesis of combined particle and mesh algorithms and allows hierarchy-building developments to benefit both simulation methods. An additional advantage of the oct-tree over the binary tree (recursive bisection) for dividing space is evident when combining particle and mesh algorithms: the spatially divided oct-tree allows for easy alignment with a mesh while the the binary tree does not easily overlay a mesh or another tree [17]. The parallel implementation of the Barnes-Hut code by Salmon [15], including domain decomposition and tree construc-
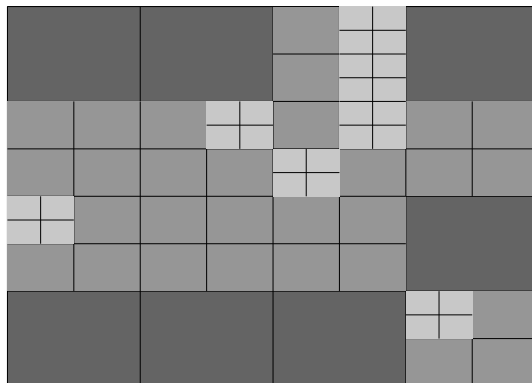
4

Figure 3: The flattened tree in figure 2 interpreted as a composite grid.

Figure 2: A collapsed representation of a small, two-dimensional Barnes-Hut tree containing 32 particles.

tion, provides insights applicable to adaptive mesh refinement on massively-parallel multiple instruction multiple data (MIMD) computers. The locality of the algorithms precisely provide the structure necessary for efficient parallel domain decomposition and ordered, hypercube-like communication on MIMD architectures.

An astrophysical model combining a smooth fluid for gas dynamics with discrete particles representing massive objects can occur entirely on a mesh or using a mixed simulation. The block structures available in the AFAC algorithm allow arbitrarily-shaped, nested regions of rectangular meshes to be used as the relaxation grid for a multilevel algorithm; these regions can directly represent the partially complete sub-cubes present in oct-tree data structures frequently used in three dimensional particle simulations. When combining both methods, the density of mass points is no longer sufficient as an estimate for necessary grid resolution, so additional criteria based upon acceptable error in other aspects of the simulation, e.g. accurately reproducing shocks, will affect the construction of the mesh. But the grid can adapt to these constraints and the hierarchy still provides the multipole information at points of interest.

If the Method of Local Corrections is incorporated to provide greater accuracy for local interactions, the neighboring regions requiring correction can utilize the Barnes-Hut test of opening-angle or the Salmon test of cumulative error contribution [16] instead of a direct proximity calculation. The correction can be calculated using a multipole expansion instead of the direct particle-particle interaction which improves efficiency for the worst-case scenario of dense clusters. While the same machinery
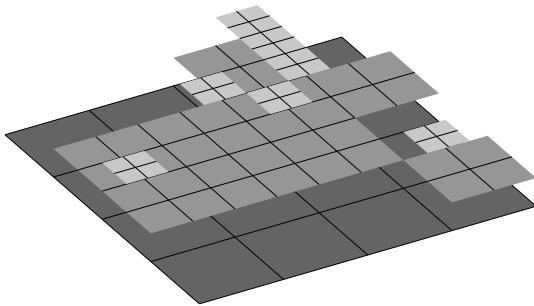
5

Figure 4: Another view of the composite grid in figure 3 showing the individual grid levels from which it is constituted.

can be used to solve the entire particle problem with a multipole method, some boundary conditions may be much harder to implement, necessitating the use of a local correction grid method.

4   Conclusion

Grid-based particle simulation algorithms continue to provide an effective technique for studying systems of point-like particles in addition to continuum systems. These methods are a useful alternative to grid-less simulations which cannot incorporate fluid interactions or complicated boundary conditions as easily or effectively. While the approach is quite different, the tree-structure and enhanced accuracy criterion which are the bases of multipole methods are equally applicable as the fundamental structure of an adaptive refinement mesh algorithm. The two techniques complement each other well and can provide a useful environment both for studying mixed particle-continuum systems and for comparing results

even when a mesh is not necessitated by the physically interesting aspects of the modeled system. The hierarchical structure naturally occurs in problems which demonstrate locality such as systems governed by the Poisson Equation.

Implementations for parallel, distributed-memory computers gain direct benefit from the locality. Because both the grid-based and particle-based methods form the same hierarchical structure, common data partitioning can be employed. A hybrid simulation using both techniques implicitly has the information for both components — particle and fluid — at hand on the local processor node, simplifying the software development and increasing the efficiency of computing such systems.

Considerations such as the efficiency of a deep, grid-based hierarchy with few or even one particle per grid cell need to be explored. Current particle-based algorithm research comparing computational accuracy against grid resolution, i.e. one can utilize lower computational accuracy with a finer grid or less refinement with higher computational accuracy, will strongly influence this result. Also, the error created by interpolating the particles onto a grid and then solving the discrete equation must be addressed when comparing grid-less and grid-based methods.

Acknowledgments

6

References

[1] A. Almgren. A fast adaptive vortex method using local correction. Technical Report 527, Center for Pure and Applied Mathematics, U.C. Berkeley, June 1991. Ph.D. Thesis.

[2] C.R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comput.*, 13, 1992.

[3] A. Appel. An efficient program for many-body simulation. *SIAM J. Sci. Stat. Comput.*, 6, 1985.

[4] J. Barnes and P. Hut. A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324, 1986.

[5] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82, 1989.

[6] M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 54, 1984.

[7] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation*, 31, 1977.

[8] W. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, 1987.

[9] G.C. Fox et al. *Solving Problems on Concurrent Processors*. Prentice Hall, New Jersey, 1988.

[10] L. Greengard. Private communications, 1991.

[11] L. Greengard and V.I. Rokhlin. A fast algorithm for particle simulations. *Journal of Compuational Physics*, 87, 1987.

[12] R.W. Hockney and J.W. Eastwood. *Computer Simulations Using Particles*. Adam Hilger, New York, 1981.

[13] J. Katzenelson. Computational structure of the N-body problem. *SIAM J. Sci. Stat. Comput.*, 10, 1989.

[14] S. McCormick and D. Quinlan. Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors. *Parallel Computing*, 12, 1989.

[15] J. Salmon. Parallel hierarchical N-body methods. Technical Report 90-14, Center for Research on Parallel Computation, Caltech, December 1990.

[16] J. Salmon and M. Warren. Skeletons from the treecode closet. *Journal of Computational Physics*, 1993. submitted.

[17] H. Samet. An overview of hierarchical spatial data structures. In *Proceedings of the Fifth Israeli Symposium on Artificial Intelligence, Vision, and Pattern Recognition*, pages 331–351, Tel Aviv, Israel, December 1988.

[18] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Massachusetts, 1990.