JOURNAL OF COMPUTATIONAL PHYSICS 115, 339–351 (1994)

# Multigrid Methods for *N*-Body Gravitational Systems

CHRIS JESSOP, MARTIN DUNCAN, AND W. Y. CHAU

*Department of Physics, Queen's University, Kingston, Ontario, Canada K7L 3N6*

Received September 30, 1991; revised October 6, 1993

We present details and some test applications of a modified particle-mesh N-body code that features an automatic, adaptive local mesh enhancement in regions where enhanced resolution is desirable. Improvements in grid resolution can be achieved without significant increases in execution time or computer memory. A single subgrid can more than halve the execution time and use less than 17% of computer memory used by an equivalent single grid of the same resolution as the subgrid. The use of two layers of subgrids (by providing a nine-fold change in resolution) can cover several density cores and reduce the execution time and memory requirements by two orders of magnitude from those of a single grid. Using a relaxation Poisson solver and Neumann boundary values interpolated from parent grids, adaptive subgrids can follow the movement of particle subsystems while allowing particles to move from grid to grid. Execution times are shown to be greatly improved over single grid methods and competitive with tree code simulations. The multigrid method is ideal for the investigation of dynamical friction, mergers, and tidal stripping of multicomponent stellar systems in fixed or interacting backgrounds. 

## 1. INTRODUCTION

We present in this paper the details and some simple applications of a 3D particle-multiple-mesh ($PM^2$) N-body code that features an automatic, adaptive, local mesh enhancement scheme. Tests and summary reports have been reported earlier in various workshops and conferences (see e.g. [7, 8, 16, 17]).

The power and usefulness of N-body simulation in the study of gravitational systems hardly need any emphasis, and the literature abounds with the results of the application of various codes to astrophysical phenomena ranging from the large-scale structure of the Universe to stellar clusters. Each code has its own advantages and drawbacks, and this again has been extensively discussed in the literature (see, for example, [20]). Very briefly, the Aarseth-type particle–particle (PP) scheme offers the most exact, direct method of solution, but suffers from the well-known limit on particle number ($N_p$) due to the execution time dependence on $N_p^2$. The particle–mesh (PM) and the particle–particle–particle–mesh ($P^3M$) methods circumvent this limitation [14], but they are constrained by grid resolution and the difficulty of meshing the small-scale calculations to the large-scale ones [6]. Recent developments introduce significant improvements to the conventional PP, PM, and $P^3M$ schemes. Tree codes [2, 13, 5] recognize that the gravitational influence of distant particles can be closely approximated by multipole moments, and thus while one treats the interaction of a particle with its close neighbours with the direct PP method, one can regard its interaction with distant ones via many-particle nodes and not via individual particles. This scheme avoids the effects of grid resolution and geometry and is gaining rapid popularity. Different versions differ in how the tree of nodes is organised.

Based on the standard PM philosophy, a new hierarchical particle mesh code (H-PM) has recently been reported [21] which makes use of a small, simple tree of PM calculations. Basically, subgrids are built within a main grid and with the allowance for particle mass difference in different grids, both the dynamical range in mass and length can be dramatically increased. We will compare Villumsen's approach with ours in Section 2. Multigridding methods have been used extensively in aerodynamics and smooth-particle-hydrodynamics simulations [12, 3, 18]. These systems often have complicated geometries and great improvements have been made in handling grids that match both the physical geometry of the boundaries and the expected fluid behaviour [19]. Unconnected grids, another form of multigridding, have been used by James and Weeks [15] to study the interaction of separated galaxies. Each galaxy is surrounded by a grid and the interaction is calculated by the use of boundary corrective charges.

Couchman [9] uses multigridding to improve the performance of the $P^3M$ method. Instead of using refined grids to improve the resolution of the grid-derived potentials and forces, he uses refined chaining grids to solve the short-range forces on the particles. Refined potential and force grids are used to calculate the forces on those particles that no longer lie in the same refined chaining mesh cell. Due to the complications of coding that would ensue, Couchman uses this method rather than using refined grids with boundary conditions containing the long-range force information.

Our code does what Couchman's did not do; it uses the boundary conditions of the refined grids to supply the force components from the rest of the simulation system. It is also close in spirit to Villumsen's, since both are based on the standard PM philosophy. Ours, however, allows for automatic, adaptive, hierarchical local mesh enhancement, making it possi-



0021-9991/94 $6.00

ble to attain the desired high resolution in only specific regions of interest, an obvious advantage for such problems as cosmological large-scale clustering and the evolution of galaxies and star clusters. In our scheme, a coarse mesh is employed for the whole computational box, and the potential is solved by standard potential solvers. We implement the alternating direction implicit (ADI) method, thus rendering the code particularly appropriate for, say, the study of isolated systems without periodic boundary conditions, such as galaxies and star clusters. The ADI solver also frees us from the need to have computational cubes that are integral powers of 2 or 3 in size. In the smaller regions with refined meshes, the potential equations are solved with the required boundary conditions interpolated from the covering mesh. Each particle is then moved with the appropriate force over the global time step.

The details of the basic features and technical aspects of our code are presented in Section 2 of this paper. Various tests performed on the code are reported in Section 3, as well as simulations of dynamically evolving systems. Conclusions, summary, and plans for future work are given in Section 4.

## 2. MULTIGRID HIERARCHIES

The use of discrete grids to solve the field equation for the N-body problem has one great limitation. The grid imposes artificial characteristic lengths that restrict the range of sizes of the structures that develop. The structures are confined to the size of the grid, and the mesh spacing imposes a softening length that suppresses the formation of small systems [4, 6]. The simplest solution is to use yet finer meshes to increase the range, at the expense of greater computational time. However, even with the largest and fastest of computers the limit is a computational cube with about 256 grid nodes to a side.

The advantage of grid-based computational methods is that the force evaluation times vary no more strongly than linearly with particle number. However, the time to calculate forces varies as $N_g^3 \log_2 N_g$, where $N_g$ is the number of cells in each dimension. While the density assignment, force interpolation, and pusher times are linear with $N_p$. Ideally $N_g^3$ should be chosen to vary linearly with $N_p$, so that the overall execution time should vary as $N_p \log_2 N_p$. However, clumpy systems lead to great wastage of grid nodes in the low density regions.

By layering a set of subgrids over the main grid, improvements to the resolution can be made without the otherwise attendant $N_g^3 \log_2 N_g$ increase of computational time. However, in many astrophysical problems, the system is not in equilibrium and the regions of interest move. Adaptive grids that change size and move with the density profile result in improved force calculations where needed. The grid positions must be stored and recalculated periodically as the simulation develops. Care must be taken in choosing an algorithm for calculating the new grid node positions to balance the decrease in mesh spacing with the increase of density; otherwise a strong tendency exists for the nodes to over-concentrate in the regions of highest density.

Villumsen [21] has devised a multigrid code in order to handle cosmological simulations. In the nested cubic grids the particle number densities are the same; in each grid there is the same average number of particles per cell. As a particle from a coarse grid passes into a fine grid it is replaced with a swarm of smaller particles. The swarm of particles and the "parent" particle are evolved independently and if the "parent" particle leaves the region of the subgrid then the swarm is removed smoothly. The forces for the subgrid particles are obtained by adding the internal field from the fine grid to the external field from the coarse grid. In order that the masses of the subgrid not be double-counted, the coarse grid potential must be re-solved with zero density in the region of the subgrid. This hierarchy can be extended to as many subgrids as necessary, but each subgrid has the same size and shape as the main grid. One advantage of this approach is that the subgrids can overlap, as they do not interact in any way. A "parent" particle that enters two subgrids creates a swarm in each. The FFT Poisson solver is used throughout and several numerical tricks are used to increase its speed for the non-periodic subgrids. We note, however, that in this scheme all the subgrids are the same size; they are all cubes of, for example, $64^3$ nodes. Hence, an elongated structure must be covered by several subgrids. Also, each grid that has subgrids must have the potential solved again for each subgrid and the solution of the potential fields is the most time expensive step in the scheme.

The ideal multigrid method can handle any number of grids, up to computer limitations, over several moving density peaks of any complicated geometry. However, constraints of code complexity lead to several limitations. For example a large-scale-structure model for the universe that develops voids will have irregular sheets and ropes of high density that cross the simulation space. Such a simulation will require that the refined grids abut each other as they try to follow the density contours. If rotation of the grids is allowed for better coverage, then overlap of fine grids must be considered. The resulting boundary conditions and bookkeeping become very complicated, since the particles cannot be identified as belonging to a single grid. The simplest algorithm, and the one we adopt here, forbids rotation and overlap of grids, leading to a tree hierarchy of grids, as each subgrid is contained totally by its parent grid and only that grid. The particles can be labeled as belonging to one grid and can only pass to parent and child grids and to adjacent siblings.

### 2.1. *Boundary Conditions*

The mathematical analysis of the grid forces within each grid in a multigrid system is no different from that of a single grid (see, for example, [14]). The interaction between the fields on the grids occurs via the boundary potentials of the subgrids. Thus, the subgrids "see" the potential field solutions of their parent grids, but there is no interaction in the opposite sense from subgrid to parent.

To calculate the boundary potentials, an interpolation process is needed to take the coarse grid potential values to the fine grid. The simplest method interpolates the coarse grid potentials to fine grid boundary potentials, the Dirichlet problem. An alternative would include Neumann terms to create a mixed boundary condition, or to eliminate the Dirichlet components entirely. Control and continuity of derivatives parallel to the boundary (the parallel components of the forces) are maintained by Dirichlet boundary conditions while Neumann conditions allow continuity of derivatives across the boundary. Continuity of all derivatives, all forces, along and across the boundary is desirable, but demanding this leads to an overdetermined subsystem that cannot be solved in general.

Using Dirichlet conditions, as we do, the form of the interpolation functions can be set by the geometry of the system and the form of the equations of motion. To establish the interpolating formulae, we have used polynomial basis functions. In the general case, the interpolation function should have no preferred axis, since there is no reason for the axes of the subgrids to be parallel to the parent grid. However, if the grids have consistent orientation and the grid boundaries are constrained to lie on coarse grid planes, no cross-boundary terms are required. We will adopt the constraint of consistent orientation but not require the boundary to coincide with a coarse grid plane. Maintaining the constraint of orientation, our studies have found that linear interpolation perpendicular to the boundary is satisfactory, especially when boundary zones (to be discussed below) are incorporated. The interpolation order on the surface of the boundary can be obtained by considering that the force field is the gradient of the potential field. It is desirable that the interpolation maintain continuity of at least that derivative. This requires a second-order interpolation function, since a first-order function is at best piece-wise continuous in that derivative.

Further specification depends on the force interpolation function and on the finite difference scheme chosen. A common group of interpolation functions, based on Bessel functions has been shown to maintain conservation of momentum [14]. In the second-order method, the CIC (cloud in cell) function with second-order difference operators, the force at a coarse grid point is defined by the potential difference of the neighbouring points. The matching of fine grid potential on the boundary and the linear form of the CIC interpolation leads to a third-order Hermite polynomial.

The main problem with the multigrid system is with this fitting of the potentials across the boundary, as is evidenced by the need to choose the direction of continuity of forces in the boundary. Consider that the parent grid potentials on the coarse grid are an approximation of the correct potential with a resolution scale $H$. This approximation is interpolated onto a system with another resolution scale $h$, $h \leq H/n$ with $n \geq 2$. The result is that the forces near the boundary have two undesirable properties. The first is the aforementioned discontinuity of the force across or within the boundary. Second, there can be strong spurious fluctuations in the fields in the subgrid.

The second problem arises from two causes: the non-ideal interpolation functions, and the presence of particles near the boundary. Interpolation functions are polynomials of finite order but ideally need to be able to model the $1/r$ form of the potential function. Thus, even if the potentials on the coarse grid were the analytically correct potentials, the interpolation function will, in general, give fine grid potential values that are incorrect for the intermediate points. This leads to errors with a characteristic length of about the coarse grid spacing.

More importantly, if an isolated particle wanders near a boundary, it will see an "image particle" on the other side of the boundary since the smoothing length changes discontinuously at the boundary. The potential well around a particle has a rounded cusp and the depth of the well depends on the degree of this rounding. This presents no problem on a single grid since this is the normal softening scale that suppresses particle interactions within the smoothing scale. However, if a particle sits near a boundary, this rounded cusp of the coarse grid will be transferred to the fine grid boundary values. Since the resolution has changed, the scale of the rounding is wrong, and more importantly the potential well is not deep enough. The effect of elevating the potential next to the particle is similar to the introduction of an image charge in a corresponding electrical system.

These problems can be eliminated by using a grid larger than the region of interest. This is necessary in any case, since the force at the edge of a grid is not defined and the width of the interpolation function means that a particle cannot obtain its forces from the subgrid until it moves sufficiently far onto the grid that it does not look to the grid boundary for forces. A widened boundary zone, henceforth called the buffer zone, is created which is not used for the calculation of forces. However, widening the buffer zone exacerbates the problem of the force discontinuity experienced by the particles when they do cross its inner boundary. This inner boundary has no significance to the calculation of force fields, but it is used for labeling the particles so that it need not be an abrupt transition but can be broadened into a continuous change over a transition zone. By creating a linear transition from coarse grid to fine as the particle crosses the transition, the discontinuity of forces can be eliminated.

Most systems do not allow for periodic boundary conditions on the main grid. Thus, Dirichlet conditions are also set for this grid, and these same problems will also exist on the main grid if the boundary conditions are specified algebraically. For example, if the boundary potentials are calculated from a multipole expansion of an isolated system, then if any particle approaches the boundary, it will see a very strong negative mass image. The result can be very strong, with particles being accelerated strongly and leaving the grid on the opposite side at high velocity. This problem can be minimized by creating a tree of multipole expansions from which the boundary poten-

tials are calculated, in the same way that a tree of multipole expansions is used by tree codes.

The fine grid now has three regions. The bulk of the grid is treated as any other grid. The buffer zone is the region in which the short-range force errors due to the interpolation errors and close particles are strongest. In this zone, the particles are pushed by the forces from the coarse grid so that the particle does not see the fine grid forces and errors, but the particles are identified with the fine grid since they are needed to calculate the necessary density field. The third, or transition, zone smooths the transition from the coarse grid to the fine grid. It is essential that the program keep track of which particles are in these zones, since they are needed for the fine grid's density but they obtain their forces partially from the coarse grid.

Our experiments have shown that the width of the buffer zone should be measured, as might be expected, in units of the coarse grid resolution. The erroneous field fluctuations on the boundary die away quickly within the distance of two coarse grid spaces. The transition zone should be wide enough that the change does not cause disruption of subsystems that cross it, but otherwise it should be as small as possible. A total boundary region, buffer, and transition of two coarse grid intervals and a transition zone width of one fine grid interval is necessary for the boundary treatment outlined above. The effects on practical grid sizes are immediately obvious: given that each grid will not be larger than, say, $32^3$, and that four coarse grid zones of the finer grid will not be used, a resolution increase greater than a factor of 8 is not possible. However, two- and threefold increases are practical, and order of magnitude increases can be achieved by nesting grids. In the future, a more sophisticated boundary treatment that uses the Green's form of the grid potential to build the boundary potentials might allow for a reduction of the buffer zone and improve efficiency.

### 2.2. *Grid Selection*

The choice of subgrids is governed by the structures in the simulation. The need for a finer mesh arises when the current mesh is not accurate enough to follow the greater fluctuations in the force field generated by small structures of enhanced density. The simplest approach uses the density distribution to generate a criterion field indicating the need for grid refinement. The subgrid structure is rearranged wherever the current grid resolution is less or greater than warranted, subject to the restrictions on grid orientation and overlap.

The specifics of the operation that generates the subgrids depend on the type of system under study. For example, a self-similar criterion would require that the average particle number density per grid element be roughly constant. In practice, some threshold density was specified. If the density exceeded that value in a region, a subgrid was created around that region. However, any threshold level that is compared to local density has the limitation that a density distribution can always be created that results in an infinite progression of successively finer grids. Moreover, the chosen criterion function must be sensitive to any real density fluctuations but not so sensitive that it reacts to the random sampling fluctuations.

The self-similar criterion is very sensitive to this kind of fluctuation and is not suitable for many systems. For example, for a constant density spherical system, the $\sqrt{n}$ standard deviation in the numbers of particles per cell is greater than 0.5 of the average for any reasonable main grid since there are few particles per cell. The result is that to eliminate such a criterion from falsely detecting $1\sigma$ variations, an excess 69,000 particles are needed on a $64^3$ grid. It was found that to eliminate unwanted spurious subgrids, the threshold had to be set above $10\sigma$, more than 8 times the average density. For grid-filling systems, such as cosmological simulations, the requirements are not as limiting since the local density is closer to the average, but care is needed if strong density peaks develop. For the simulation program, we typically used a threshold of 16 times the average density. In future versions an average over a volume containing several grid cells will be used to reduce the effects of counting noise.

Any centrally concentrated system that has a power-law density profile needs more complicated criterion functions, since they tend to produce an excess of grids. The simplest solution sets a limit on the number of levels of subgrids, which we usually set at three, and the required density threshold can be made to depend on the grid level. Systems like this also require a higher threshold than others, since otherwise the subgrid boundaries often lie within the buffer zone of the parent grid.

### 2.3. *Energy Conservation*

Conservation of the total energy of the particle system is a useful tool for checking the reliability of the computer simulation. However, care must be taken in defining the potential energy, since the particles have accelerations that are based on a smoothed potential. This is not a problem when only one grid is used. However, if particles are moving about a set of many grids, each with a different resolution scale, then the softening length is not well defined. Indeed, we find in our simulations that energy non-conservation of the order of 10% is not unusual.

In our calculation the potential energy is calculated by

$$U = -\sum_{i \neq j} \frac{Gm_i m_j}{(r_{ij}^2 + 2 \times d_{ij}^2)^{3/2}},$$

where $d_{ij}$ is the softening length of the particle on the grid with the larger resolution, noting that no overlap of subgrids is permitted. However, using this equation leads to a computation time proportional to $N_p^2$, and even for 5000 particles the time taken to evaluate this equation once every 20 steps exceeds the force calculation times. The computational time can be reduced by randomly selecting a representative subgroup of particles and calculating its configuration energy. Then the total potential energy is

$$U(N_p) = U(N < N_p) \times 2^{(N_p/N)}$$

to within statistical fluctuations. By selecting $N$ small enough, the time saving can be used to calculate $U(N < N_p)$ a number of times in order that a standard deviation be calculated and that the central limit theorem be applied.

It should be noted that this method of calculating $U$ employs a "direct" potential that is not the same as the grid potential actually used to calculate the forces. The two are not really coupled and tests have shown that this leads to discrepancies on the order of 4%.

Also, complications exist when the grid structure itself is changing with time. In practice the fluctuations due to the changing grid are on the order of 5%, except for cold collapse simulations where the extreme changes in scale can cause correspondingly larger changes in the potential energy. However, all but 1% of this fluctuation is due to this change in $d_{ij}$ in the calculation of the energy. For a set of fixed subgrids and an equilibrium system the energy error is typically 1% over an entire simulation when 20 random sets of 500 particles are selected.

### 2.4. *The Program*

The first program written using adaptive multigridding needed to be flexible enough to handle all the different tests and yet not so complicated that alterations required large changes to the program structure. Thus, the restrictions that allowed a simple tree of subgrids were imposed and only the simplest boundary conditions and regridding algorithms were employed, as described above.

For simplicity it was decided that subgrid boundaries would be constrained to lie parallel to the main grid and that no overlapping of subgrids would be permitted. The selection process for regridding is based solely on the density contrast with a limit of three levels of nesting. An adaptive time-step leap-frog pusher based on the central density and velocities was used for the time integration of collapse models. Given that the program recalculates the positions of the subgrids at periodic intervals, it is also a simple matter to check the size of the particle system and to adjust the size of the main grid to follow global expansions and contractions.

At present a single global time-step that is allowed to vary with time has been used. The current multigridding scheme allows the spatial resolution of the force fields to vary with position, but it does not allow the subgrids to be solved at different times; the boundary conditions necessary for the potential calculations for the subgrids require that the potential from the entire course grid be solved each time. Given this requirement, the extra time used to push the particles on the course grid is not significant. It may be possible in future versions to decouple the coarse and fine grid potential calculations and implement a time-step that differs from grid to grid.

A 3D ADI solver was selected to invert Poisson's equation. Second-order difference equations are used, including the nine-point Laplacian star and a two-point difference for derivatives. Although CIC is the optimal scheme [11] it was simple to keep the options of the NGP and TSC schemes. Shaping operators for CIC and TSC were included [10, 17]. For the subgrids, the necessary Dirichlet boundary conditions are calculated by a third-order Hermite interpolation. The particle integrator, the pusher, used was the standard leap-frog. Particle masses are stored so that different species of particles can be included.

The main grid boundary conditions can be handled by direct summation of the algebraic potential due to each particle. This is only practical when the number of particles is small, but it is necessary for the tests using only a few particles. Normally, the main grid boundary potentials are calculated from a multipole expansion, up to and including quadrupole terms, of each grid and subgrid. Particles that have left the main grid are assigned accelerations based on the forces from that multipole expansion.

When the simulation consists of a system with a single density peak, the number of escapees can be controlled by maintaining a constant ratio of main grid size to system size. It was found that, like the subgridding criterion, the most appropriate definition of the system size was based on the density profile. For spherical systems, the radius that contained the density contour of $\frac{1}{8}$ of the average density within the half-mass radius satisfied the need to contain the whole system without becoming excessively large as individual particles escaped. Since the initial ratio of system size to grid size is maintained, control on the actual size of the main grid is made in the specification of the initial grid size.

### 2.5. *Implementation*

The program was developed on a $\mu$VAX II using VAX Fortran, but it has been modified for use on an IBM mainframe and UNIX-based machines for execution. While Fortran has no recursive abilities nor pointers, the ability to pass array dimensions simplified the procedure and function definitions. The lack of recursion was handled by writing a function that tracked the progress of a pointer (and the pointer to the last considered node) through the tree of grids so that iteration could be employed. The indexing of grids and particles uses integers.

Lists of particles are used in order to reduce the need to check all the particles against all the grid boundaries each time a density field is calculated, or forces are interpolated to particles from a grid. The particle list must be scanned three times at the beginning of each time step in order to allow for all possible movements of particles from grid to grid.

Since standard Fortran has no capability for dynamic allocation and pointers, integers are used to index arrays that must be defined to be large enough to handle all expected conditions. Also, Fortran has no structured variables, so the grid values are stored on different arrays but grouped in a common block. In order to simplify programming, the grid tree and all linked lists start with dummy nodes.

The ability of Fortran to redimension arrays as they are passed into subroutines alleviates the worst of the problems that result from the lack of dynamic allocation. The resolution and relative dimensions of the grids change as the simulation proceeds, so the memory space required for each grid is not known beforehand. It would be prohibitive to assume some maximum array size, since this would be at least $65^3$ nodes. More flexible and efficient use of memory is achieved by using a very primitive stack that allocates space from a single large linear array.

In testing the program it was found that the truncation and interpolation errors were large enough that double precision real variables were not warranted. The increased computation speed and halving of memory requirements that this meant also permits larger grids and smaller time-steps.

Subroutines that tracked the use of CPU time by each task were written into the program and at the end of execution, the program stores the accumulated times into a file for study. As expected, the execution times of the pushers, density assignment, force interpolation, linked list overhead, and output routine times all vary linearly with particle number. Similarly, the boundary value, Poisson solvers, and linked list routines depend linearly with the number of grids, while the density assignment and force interpolation routines vary with the number of grids to some power less than one. Typical values for each operation are shown in Table I. The totals show a typical run time of 47 s/time-step for 1000 particles on the $\mu$VAX for a multigrid system while the comparable single grid took almost three times longer at 137 s/time-step. The MIPS 120 reduces these times by an order of magnitude. The 2000 particle system of two interacting Plummer spheres with adaptive time-step and regridding upon a main grid of $41 \times 25 \times 25$ took only 20.6 s/time-step on that machine. Increasing the particle number to 10,000 added only 2.0 s/time-step.

**TABLE I**

Program Run Times on a $\mu$VAX II

| Operation | Action | Dependences[a] | Times[b] Single | Multi |
|---|---|---|---|---|
| PUSH | Calculate the next set of particle vectors | Np | 0.11 | 0.12 |
| DENSITY | Get the grid densities | $Np \times G^{a,c}$ | 0.61 | 1.1 |
| BOUND | Calculate boundary potentials | $Ng^2 \times G$ | 2.1 | 2.5 |
| FIELDS | Solve Poisson equation | $G \times Ng^3 \log_2 Ng$ | 120 | 30. |
| FORCES | Interpolate forces | $Np^b \times G^c$ | 1.4 | 2.3 |
| LISTS | Update linked lists | $Np \times G$ | 0.10 | 0.15 |
| OVERHEAD | Misc. | | 0.006 | 0.006 |
| SIMULAT | Simulation dependent routines inc. regridding adaptive time-step | Np | 0.60 | 3.1 |
| CONTROL | Calculate energies, momenta, and moments | $c^{c,d}$ | 12 | 8.3 |
| OUTPUT | Write particle data and control quantities | $Np * d^d$ | 0.10 | 0.12 |
| Total | | | 137 | 47 |

[a] Np—no. of particles. Ng—no. of grid nodes across a grid, or the main grid. G—no. of grids.

[b] Average CPU time per time-step on a $\mu$VAX II for a Plummer model system of 1000 particles with adaptive gridding, variable time-step, and output every 10 cycles, on a single $33^3$ grid, or a multigrid system with a $17^3$ main grid (regridding creating a second grid) measured in seconds $\pm 5\%$.

[c] The values a, b, c, d depend on the simulation: a and b depend on the distribution of particles among the grids and boundary zones; c and d depend on the frequency with which output is performed.

[d] This operation should be strictly linear with the particle number, except for the potential energy which is quadratic. In order to eliminate the quadratic effect, an approximation based on a sample of 500 particles is taken. Since this calculation still dominates the time, the overall time required is almost independent of the number of particles, and it is controlled by the frequency of output.

## 3. CODE TESTING

### 3.1. *General*

The validity of the code depends on the correctness of the basic method and the accuracy of the various key components, and in the final analysis it is confirmed only when simulation of the test systems yields results agreeing with those available from other methods. We have not carried out a rigorous error analysis of all aspects of the code. However, the very good qualitative agreement in our various tests gives us confidence in the basic soundness of our multigrid scheme. We present in this section a concise summary of these tests, while deferring to later work for more rigorous quantitative analysis.

The calculation of the various fields and the particle forces in our code is no different from that of any other PM scheme as long as the region of interest is more than about two cells away from the grid boundary. This is illustrated in Fig. 1, where the potential of a single particle is plotted as a function of the distance in grid cell units. In Section 2 we have already presented special ways of handling regions within two cell units of the boundary, specifically the introduction of the buffer and transition zones. With these provisions, results from various tests agree in general with what is to be expected.

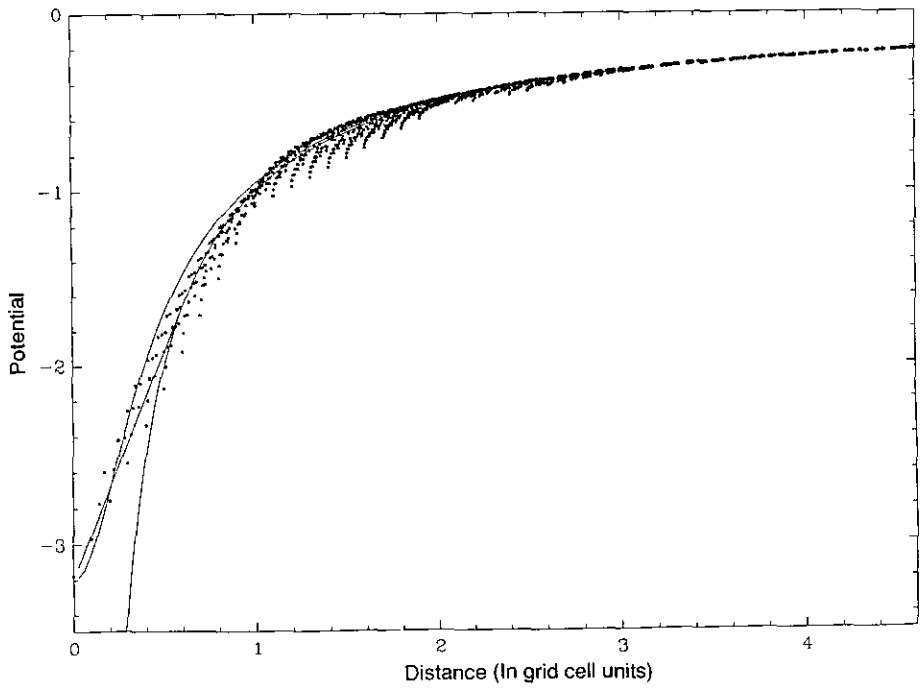


**FIG. 1.** Potential field about a unit charge on a unit grid. The points are the grid potentials, while the three lines are the analytic curves $-r^{-1}$, $-(r^2 + a^2)^{-1/2}$, and the piecewise $-r^{-1}$ if $r > c$; $-2/c + r/c^2$ if $r < c$. Matching the depth of the well requires that $a = 0.31$ and $c = 0.63$. The piecewise function is a better fit. Also, most of the spread due to angular variation dies away within two grid cells of the particle.

### 3.2. *Tests of Components in the Code*

Results of preliminary testing had been reported elsewhere [7], but some will be mentioned here again for sake of completeness. The effects of subgrid boundary conditions on single and two particle systems were studied. It was these tests that revealed the need for the buffer and transition zones. Despite the fact that cubic interpolation in the surface of the boundary would be generally more accurate, it was found that the time expense of cubic interpolation normal to the surface could not be justified by any increase in accuracy, so we used linear interpolation.

One might expect that the coarse grid potential field approximation would be accurate enough that the potential field on the fine grid would fit smoothly on the subgrid boundary. However, early results with collapsing spherical systems revealed a discontinuity in the forces normal to the boundaries. The Dirichlet boundary conditions ensured the continuity of the components along the surface but not of the normal component. The result was a separation of the particle system into a cubical core and an envelope with a cubical inner boundary. If the volume of significant density came close, within two coarse grid zones, to the boundary, the system evolved into a cubical structure, even when the bulk of a spherical model resided on the inner grid. The addition of a buffer zone and a transition zone (discussed above) solved these problems at the expense of a larger grid, and errors introduced by the multigridding were reduced to the same order as those due to the truncation errors of a single grid.

### 3.3. *Steady State and Equilibrium Tests*

The easily implemented isolated binary system is no easier for grid methods to evolve than any large $N_p$ system and it has the advantages of having an analytic solution and many conserved quantities for comparison. The details have been reported [7], and we will simply mention here that a comparison between a single $17^3$ grid, a $9^3$ grid, and a $13^3 \subset 9^3$ ($13^3$ within a $9^3$) grid system showed stability for as long as the minimum particle separation was greater than four grid spacings. Furthermore, while momenta are well conserved in all cases ($\sim$1 part in a million), energy conservation and shape of the orbit requires good resolution. In this context, the $13^3 \subset 9^3$ calculations yield results comparable to those obtained with the single $17^3$ while requiring about half the time.

Stability of equilibrium models was tested using Plummer spheres, a well-studied stable configuration with a density profile given by

$$\rho(r) = \left(\frac{3}{4\pi}\right)\frac{M}{R^3}\frac{1}{[1 + (r/R)^2]^{5/2}},$$

where $M$ is the cluster mass and $R$ is the Plummer radius; $5 \times$

$10^3$ particles were used, with $M = 1$ and $R = 0.2$. The evolution was followed on a single $33^3$ grid, an adaptive system on a $17^3$ main grid, and an adaptive system on a $33^3$ main grid. The subgrids doubled the grid resolutions for these simulations. These runs demonstrate clearly the importance of grid resolution and suitable choice of time steps in producing reliable results. The high density core would tend to expand unless the discretization scale or the softening length (i.e., the grid resolution) is much less than the characteristic scale of the system. It was found that the high density core must be covered by at least four grid meshes. An inappropriate choice of time-step, on the other hand, gave rise to unrealistic expansion that actually looks like an explosion.

Overall, the final system evolving from the Plummer models varies little among all three simulations. Good momentum conservation was observed in all cases, staying on the same order as the original non-zero momentum due to the random process of particle placement. The $33^3$ multigrid run conserved energy to 0.6%, compared with 3.1% for the single $33^3$ grid, after taking the statistical sampling fluctuations into account. As shown in Table I, the $33^3$ single grid system required 137 s for each time-step while the $17^3$ system was almost 3 times faster, requiring only 47.

### 3.4. *Dynamical Tests and Relaxation to Equilibrium*

Dynamical tests were performed by following the collapse of a perfectly cold (zero initial velocity dispersion) spherical system. This tested all aspects of the code, including in particular the handling of particles moving between grids. Furthermore, the obvious disparity between the symmetry of the physical system (spherical) and that of the code (cubic) would expose any undesirable, spurious grid-induced effects.

First the cold collapse of a spherical homogeneous system was modelled; $4 \times 10^4$ particles of equal mass and zero initial velocity were randomly distributed within a sphere. The evolution was followed on a single $33^3$ grid, as well as $17^3 \subset 17^3$ grid. Over a time interval of $10^3$ time steps, covering one complete collapse and a subsequent re-expansion, there are very good qualitative agreements between the two simulations. However, the double grid simulation ran 5 times faster. Simulation runs without the use of adaptive time-steps and grid sizes led to chaotic systems in which energy conservation is grossly violated and grid geometry was imprinted on the particle distribution.

A more realistic system than the spherical homogeneous case is one in which the initial density falls off as $r^{-1}$. In this case, $2 \times 10^3$ particles were used. The collapse was marked by the formation of radial clustering which developed in the postcollapse system as a prolate ellipsoidal bar with axis ratios of 1.0 : 1.1 : 1.6 (Figs. 2 and 3) in reasonable agreement with the results of Aguilar, Merritt, and Duncan [1] and those obtained from a direct PP calculation. As the system evolved, the code automatically responded by placing a second subgrid layer. Moreover, the subgrid resolution was improved so that at the time of greatest contraction the central resolution was 9 times that of the main grid, and 4.5 times that of the initial subgrid. As shown in Fig. 2, the energy error after 3.6 time units was 12% while the maximum excursion was 32%. While these errors seem large, and are larger than usual, most of this is accounted for by the changing subgrid resolution for reasons described previously. Figure 3 shows both the initial system and three axial projections of the final particle and grid system.

A comparison test was performed on a direct summation $N^2$ code. For 2000 particles the execution times were very similar. The evolution of the system was very similar to that of the multigrid simulation. The final system was an ellipsoid with axial ratios of 1.0 : 1.5 : 2.3 (somewhat larger than the multigrid run), but it was aligned in the same direction as the bar of the multigrid simulation.

Even though the run-times for this 2000 particle simulation using both the multigrid method and the direct PP were similar, we estimate that for a system of $10^5$ particles, it would take 45 s/time-step on a MIPS-120 for the multigrid code, 4 h/time-step for a single grid with the same resolution and 15 h/ time-step for the PP. Furthermore, comparison runs using 20,000 particles $33^3 \subset 33^3$ grids, the multigrid method was 5.7 times faster than a treecode algorithm with the corresponding softening lengths. These timing comparisons are, of course, highly system-dependent and the above numbers apply only to the cold collapse system.

### 3.5. *Simulations with Moving Density Peaks*

Figure 4 shows the results of the collision of two Plummer spheres of 5000 particles each, with $R = 0.2$ and unit mass each. Both the initial impact parameter and trajectory energy were set to zero. The initial cluster positions and grid arrangement are shown in the $t = 0$ scatter plot while the other three again show the resulting system. The adaptive multigrid routines immediately identified the two clusters for separate subgrids that eventually were replaced by the elongated grid seen in the second time-frame. After the collision, the system settles into a long lived prolate system with axial ratios of 1.0 : 1.1 : 1.4. This is typical of the results of similar merger simulations of White [22].

Another astrophysical effect of interest is the tidal stripping of clusters. This type of system provides another good test of the multigridding process since the density peak moves over large distances during the simulation. It is a simple matter to add the background tidal forces explicitly to the time-step pusher (note that momentum is no longer conserved).

The first test was the somewhat artificial system of a King model in circular orbit about a large fixed central softened mass, such that the tidal radius of the cluster distribution is the tidal radius set by the central background potential. Again a comparison simulation was performed using direct summation methods. Both simulations started with 2000 particles orbiting

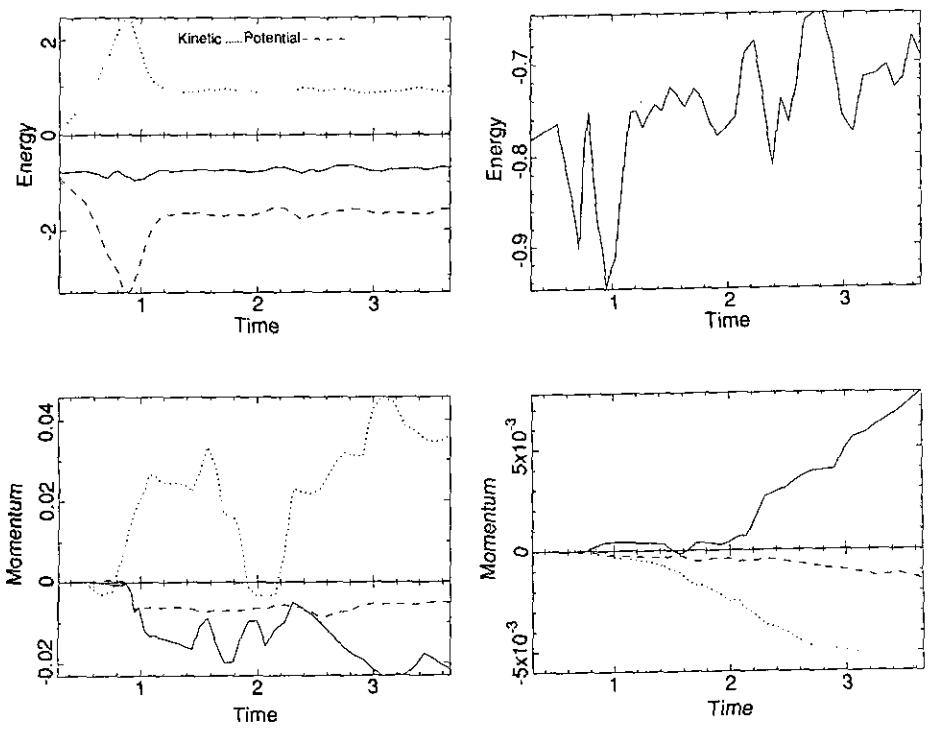


**FIG. 2.** Cold collapse of $1/r$ density sphere: diagnostics (a), energies (b), expansion of total energy (c), linear momentum (d); angular momentum for parts (c) and (d): ——— $x$ component; · · · $y$ component; – – – $z$ component.

about a central object with mass 30 times that of the cluster and a softening radius equal to the orbital radius of 6.8. During the evolution, shown in Fig. 5, the subgrids tracked the motion of the King model. Sitting on a $33*33*17$ main grid, the subgrids provided a six-fold increase in resolution. As the simulation progressed, particles were stripped from the outer layers of the cluster. After one and a quarter orbits, 363 particles had been stripped from the cluster in the multigrid simulation while 334 had been stripped in the PP direct summation simulation. Moreover, the particle distribution in the tails was visually similar. The remaining cluster had slightly larger size in the multigrid simulation but the difference was within 7% (at the half-mass radius).

Comparison simulations of mergers in tidal fields were also made. Such mergers are probably common in groups of galaxies. The simulations involved the merger of two stellar systems in the central tidal field. The initial system was like that of the previous except that the second of the identical galaxies was placed on an orbit designed to cause the two to merge. The impact parameter of the collision was not quite zero, being approximately the half-mass radius of the galaxy.

During the collision the core of the outer galaxy swung almost a complete rotation around the other until finally coalescing into one central peak. Over this time almost half of the outer galaxy was thrown off. Again, the two simulation methods produced similar systems. The multigrid simulation allowed 471 particles to escape while 451 particles escaped in the direct summation simulation. Despite the frequent regridding needed in the multigrid simulation, energy was conserved to 4.2%. The most noticeable difference between the two simulations was that the resulting core was a little larger for the multigrid simulation: The half-mass radius of the merger remnant in the multigrid simulation was 20% larger than that of the PP simulation. The results of these simulations were sufficiently intriguing that we intend to pursue a systematic study of the merger of galaxies in tidal fields further in the near future.

## 4. FURTHER DEVELOPMENTS AND CONCLUSIONS

### 4.1. *Further Developments*

The current simulation program can easily handle simulations of systems with 100,000 particles in multiple subgrids. Simulations of 1000 time-steps take 11 CPU hours on the MIPS 120 (which is roughly a 2 Mflop machine). The method is ideal for systems containing several galaxies and/or stellar clusters and a background of distributed particles.

The current implementation is suitable for simulations of systems where there is at most one preferred set of axes. It is also ideal for systems where it is necessary to have high resolution for a single core. Even so, the practical resolution increase from the finest grid to the coarsest is limited to less than two orders of magnitude (corresponding to six orders of magnitude

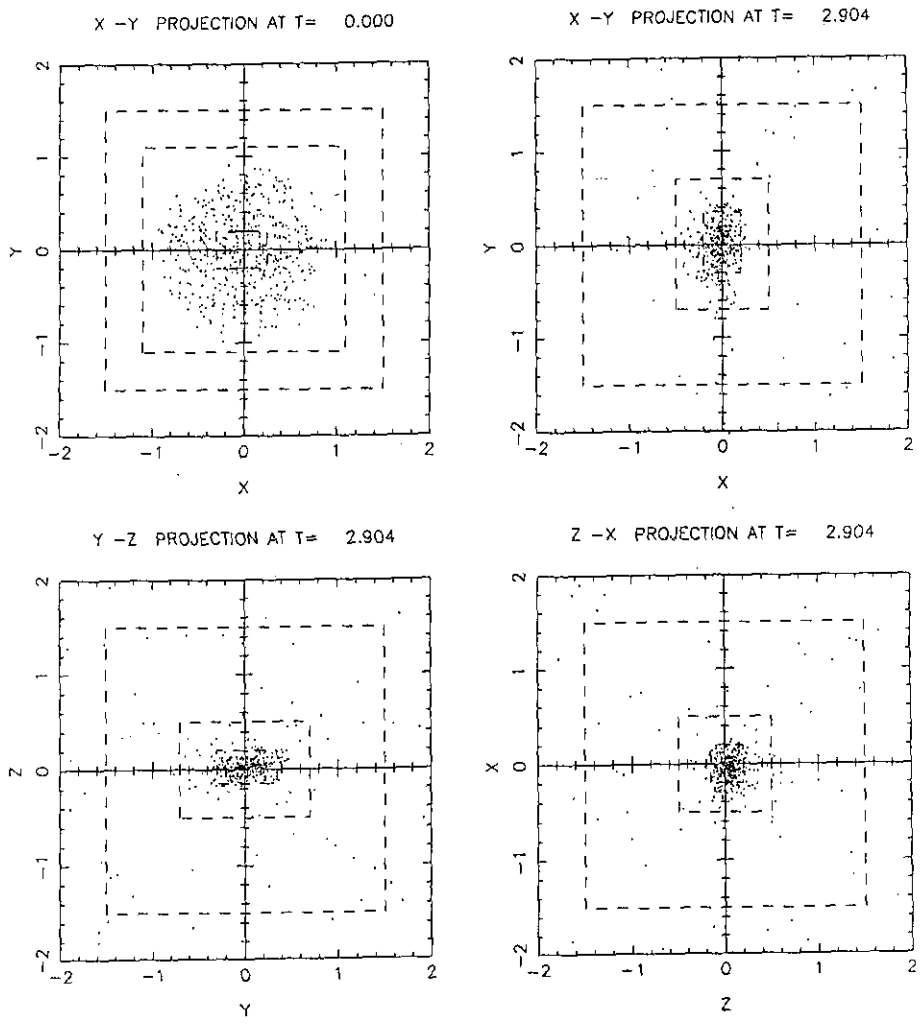


**FIG. 3.** Cold collapse of $1/r$ density sphere showing initial and final system configurations. The dashed lines show the grid borders. For clarity, only a few hundred of the 2000 particles are shown. The final system is a prolate ellipsoid with axial ratios of 1.0 : 1.1 : 1.6.

in density contrast), due to the cumbersome boundary zones. These boundary zones preclude gains greater than 3 or 4 between a grid and its next lowest subgrid. Even doubling the resolution on a 33 subgrid leads to boundary zones that occupy about half of the subgrid. Thus, the most important limitation of the current program is the wastage of grid space in the boundary zones. The emphasis of further development must be to improve the boundary conditions so that the boundary zones can be made narrower. Correction terms due to the fine-grid density distribution near the boundary can be added to the boundary potentials. These terms could be applied to the boundary near any particles in the buffer zone. These terms would depend on the ratio of resolution of the grid to its parent, their relative orientation, and, to a lesser degree, the relative offset. This should reduce the image mass problem to a large extent. While it does nothing for the continuity problem, the limitation is due to the width of the buffer zone rather than the transition zone. Work has started along these lines.

The regridding algorithm could do with further development. The simple process of laying grids whenever density peaks grow and removing grids whenever they diminish past a threshold level frequently leads to continual changing of grid sizes and resolutions when the system densities fluctuate across the threshold levels. A smoothing process would reduce the sensitivity to small random temporal fluctuations.

The generalization of the code to allow various grid orientations is fairly simple provided they do not overlap. If grids are allowed to overlap, the simple structures of a grid tree and a linked list of masses for each grid break down. It would no longer be possible to identify a single grid to which a particle belongs and in some cases a subgrid may be covered by two or more parent grids. In the latter case, the boundary conditions of the subgrid become particularly awkward. So long as no grid is allowed to lie over the boundary of a coarser grid, a grid tree structure can be maintained and the boundary conditions are no more of a problem than previously.

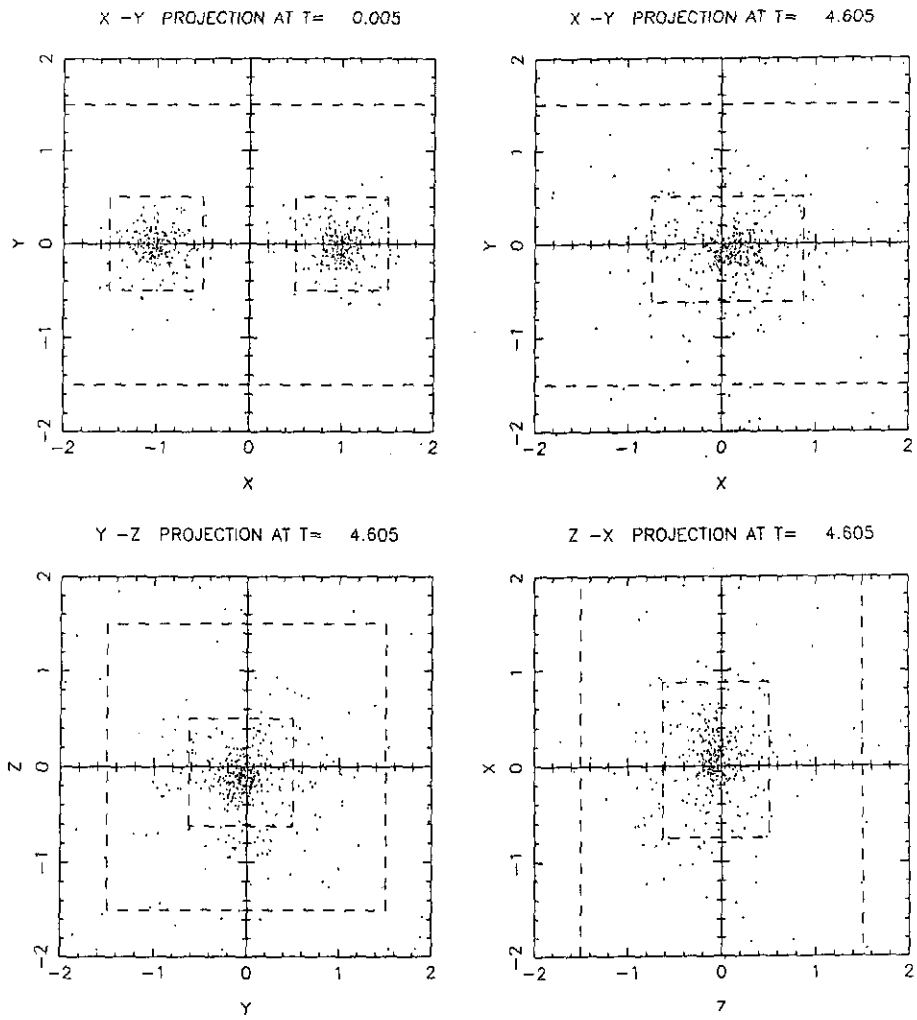


**FIG. 4.** Collision and merger of two Plummer models starting from a head-on parabolic trajectory. Each Plummer sphere has 5000 particles although, for clarity, only a few hundred are shown here: (a) original system; (b), (c), and (d) are *X-Y*, *Y-Z*, and *Z-X* projections of the final system. The dashed lines show the grid borders. During the simulation the two subgrids around the original spheres coalesce into one rectangular grid.

The problem of multiple particle lists can be solved if true dynamic allocation is available since that would permit lists of lists, allowing a particle to belong to multiple grids. Fortran does not have this capability. Further development of this kind in this language would require that this capability be simulated by dynamic subarray allocation in the form currently used for the grid meshes but with proper "garbage collection" and free list handling.

However, it might not be necessary to solve these complications if a selective Brandt multigrid method were incorporated. The normal Brandt multigrid solver refines the solving mesh over the whole solution space. By selectively refining regions of high density, the Brandt multigrid process could be incorporated with our multigrid method. The coupling of the solutions of the potential fields on each of the grids might reduce the discontinuity in the forces at the grid boundaries. If this approach were to eliminate the need for boundary zones, then the presence of many abutting subgrids would not be a difficulty.

As demonstrated by Table I, the time taken to calculate the forces dominates until the number of particles approaches 10 per grid cell, even for very small grids. Thus, little computational time is wasted if all the particles on a grid, indeed on all grids, are moved with a common time-step. However, for systems with strong density peaks, accuracy within these peaks demands a much smaller time-step than for the general field. However, execution times could be reduced even further if each grid had a separate time-step, so long as the boundary conditions could be extrapolated from previous coarse grid frames. The density peaks of a system could be evolved with the necessary time-step for accuracy without wasting computational time by over-resolving the evolution on the main grid.

Further development could be made of direct summation

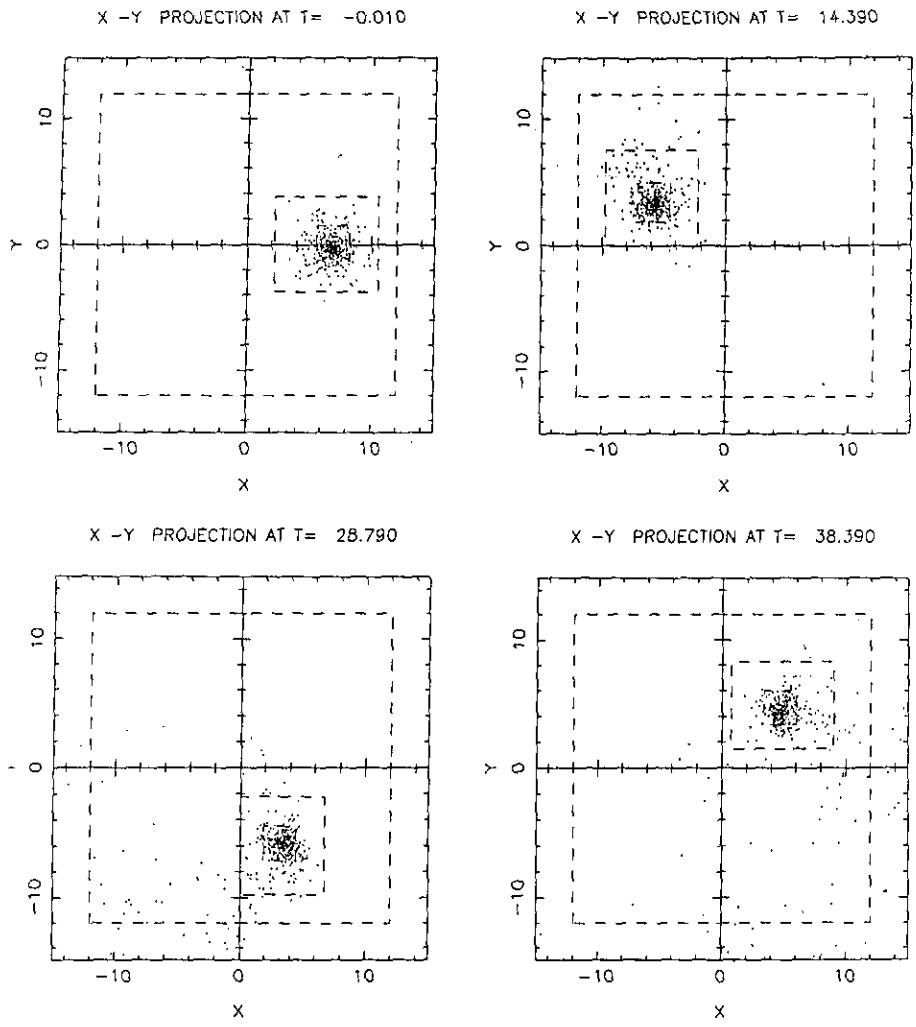


**FIG. 5.** Tidal stripping in a central force field. A King model with the tidal radius set by the background potential is in a circular orbit. The system has 2000 particles of which 363 are stripped over 1.3 orbits. Shown are snapshots with the viewing axis perpendicular to the plane of the motion. Note that the movement of the subgrids follows the moving King system: (a) is the original system; (b), (c), and (d) are later times.

routines. For example, globular cluster simulations require that the core be treated in a manner that does not eliminate the collisional process. Thus, a system that uses a grid for the envelope and PP summation for the core would be ideal for globular cluster evolution studies. If the time-step were allowed to vary from grid to grid and if two and three body regularization were included for the core regions, core collapse and the formation of binary pairs and the subsequent cluster heating could be followed.

### 4.2. *Conclusions*

In the previous sections it has been shown that the use of selective refinement of grids can be used to improve the resolution of forces and hence the accuracy of results in $N$-body systems. The method is most useful for the simulation of collisionless systems in which the density is such that a number of distinct subsystems exist. The separation of the solving space into subregions also allows for parts of the system to be handled in a different manner than the field.

The method is not designed for collisional systems, since the necessary accuracy with faster run-times can be achieved by tree codes and $P^3M$ methods. However, the multigrid method can be adapted to allow direct summation methods on leaf grids, which would make the code good at handling globular cluster evolution, which most grid systems cannot handle.

The method has been compared to a direct summation method with good agreement. For 100,000 particles the multigrid method reduces the computational time over direct methods by three orders of magnitude while one to two orders of magnitude are saved over a single grid system with the same resolution of the finest subgrid. Tree codes are the most competitive but

still the multigrid method is significantly faster for the particular systems studied.

There still needs to be further development on the details and coding as discussed earlier. The boundary conditions and zones pose the greatest difficulties and any further development must address this problem. Incorporating a selective Brandt multigrid solver into the code may solve this dilemma. If not, further effort should also be made for solving the problem of laying overlapping grids. Also, more study needs to be done to create a more sophisticated regridding algorithm.

Currently the program can be used for simulations of galaxy–galaxy interactions, gas-free galaxy collisions. It is ideal for studying the merger of galaxies in tidal fields and the evolution of groups and clusters of galaxies. The improvements listed above, if successful, would allow the code to follow the structures that develop in elongated or flattened systems, including cosmological simulations. If a pressure component were added to the force calculations, as with fluid dynamics codes, the structures in dissipational disk galaxies could also be modeled.

## ACKNOWLEDGMENTS

The authors thank Dr. K. L. Chan for providing the ADI Poisson solver, and Queen's University, NSERC, and a Reinhardt Scholarship for financial support.

## REFERENCES

1. L. Aguilar, D. Merrit, and M. Duncan, "Cold Collapse as a Way of Making Elliptical Galaxies," in *Structure and Dynamics of Elliptical Galaxies, IAU Symposium 127, 1987,* p. 519; *Bull. Am. Astron. Soc.* **19** (1987).
2. J. Barnes and P. Hut, *Nature* **324,** 446 (1986).
3. M. J. Berger and A. Jameson, *AIAA J.* **23,** 561 (1985).
4. F. R. Bouchet, J.-C. Adam, and R. Pellat, *Astron. Astrophys.* **114,** 144 (1984).
5. F. R. Bouchet and L. Hernquist, "Implementation of a Tree Code for Cosmology," in *Large Scale Structures of the Universe, IAU Sympos. 130, 1988,* p. 563.
6. F. R. Bouchet and H. E. Kandrup, *Ap. J.* **299,** 1 (1985).
7. K. L. Chan, W. Y. Chau, C. Jessop, and M. Jorgenson, "Multiple-Mesh-Particle Scheme for N-Body Simulations," in *The Use of Supercomputers in Stellar Dynamics, Proceedings, Workshop in Princeton, 1986,* edited by P. Hut and S. L. McMillan, Lecture Notes in Physics, Vol. 267 (Springer-Verlag, New York/Berlin, 1986), p. 146.
8. W. Y. Chau, C. Jessop, M. Jorgenson, and K. L. Chan, Paper No. 23.10, *168th Meeting of the AAS, 1986;* abstract in *Bull. Am. Astron. Soc.* **18,** (1986).
9. H. M. P. Couchman, *Ap. J.* **368,** L23 (1991).
10. J. W. Eastwood and R. W. Hockney, *J. Comput. Phys.* **16,** 342 (1974).
11. J. W. Eastwood, *J. Comput. Phys.* **18,** 1 (1975).
12. W. D. Gropp, Yale Research Report YALEU/DCS/RR-278, 1983 (unpublished).
13. L. Hernquist, *Comput. Phys. Commun.* **48,** 107 (1988).
14. R. W. Hockney and J. W. Eastwood, *Computer Simulations Using Particles* (McGraw–Hill, New York, 1981).
15. R. A. James and T. Weeks, "Multiple Mesh Techniques for Modelling Interacting Galaxies," in *The Use of Supercomputers in Stellar Dynamics,* edited by P. Hut and S. L. McMillan, *Proceedings, Workshop in Princeton,* Lecture Notes in Physics, Vol. 267 (Springer-Verlag, New York/Berlin, 1986), p. 125.
16. C. Jessop, M.Sc. thesis, Queen's University, 1989 (unpublished).
17. C. Jessop, W. Y. Chau, and K. L. Chan, Paper No. 17.01, *172nd Meeting of the AAS 1988;* abstract in *Bull. Am. Astron. Soc.* **20** (1988).
18. P. Kutler, "A Perspective of Theoretical and Applied Computational Fluid Dynamics," in *21st Aerospace Sciences Meeting, AIAA, 1983.*
19. T. H. Pulliam and J. L. Steger, "Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," in *23rd Aerospace Sciences Meeting, AIAA, 1985.*
20. J. A. Sellwood, *Annu. Rev. Astron. Astrophys.* **25,** 151 (1987).
21. J. Villumsen, *Ap. J. Suppl.* **71,** 407 (1989).
22. S. D. M. White, *Mon. Not. R. Astron. Soc.* **189,** 831 (1979).