

**BUILDING A COMPUTER CLUSTER IN
ORDER TO SIMULATE DARK MATTER
INTERACTIONS IN PARALLEL**

By

Colin Lauer

A thesis submitted in partial fulfillment of the requirements for
the degree of

Bachelor of Science

Houghton College

October, 2013

Signature of Author.....

Department of Physics
October , 2003

.....

Dr. Christopher Wells
Research Supervisor

.....

Dr. Mark Yuly
Professor of Physics

**BUILDING A COMPUTER CLUSTER IN ORDER
TO SIMULATE DARK MATTER INTERACTIONS
IN PARALLEL**

By

Colin Lauer

Submitted to the Department of Physics
on October , 2013 in partial fulfillment of the
requirement for the degree of
Bachelor of Science

Abstract

A computer cluster was built for the purpose of performing N-body simulation. The planned simulations are to study Dark Matter (DM) particles which have self-interactions which behave similarly to the Coulomb force.

The cluster was made of two Apple iMacs. The cosmological simulation code GADGET-2 was used to run the simulations and TORQUE Resource Manager was used to manage cluster communication. The cluster and software are not limited to DM simulations, but could be used to simulate other exotic cosmologies.

Thesis Supervisor: Dr. Christopher Wells
Title: Assistant Professor of Physics

TABLE OF CONTENTS

| | |
|--|-----------|
| Chapter 1 Introduction & Background | 5 |
| 1.1 The Existence of Dark Matter | 5 |
| 1.1.1 Rotation Curves..... | 5 |
| 1.1.2 Gravitational Lensing..... | 8 |
| 1.1.3 The Bullet Cluster | 10 |
| 1.2 Motivation | 11 |
| Chapter 2 Theory | 14 |
| 2.1 N-body Simulations | 14 |
| 2.1.1 Force Calculations..... | 15 |
| 2.1.2 Time Integration | 19 |
| 2.2 Parallel Computations | 22 |
| Chapter 3 Experiment | 22 |
| 3.1 Overview | 23 |
| 3.2 GADGET-2 | 23 |
| 3.3 TORQUE | 25 |
| 3.4 The Cluster | 25 |
| 3.5 Conclusions and Suggestions for Future Work | 25 |

TABLE OF FIGURES

| | |
|---|----|
| Figure 1. A graph of a synthetic rotation curve | 7 |
| Figure 2. A diagram showing light deflecting on a lensing object's gravitational field..... | 8 |
| Figure 3. A photograph of a lensing galaxy and four objects being distorted by the gravitational lens.... | 9 |
| Figure 4. An image of the X-ray gas of the Bullet Cluster with mass distribution contours..... | 12 |
| Figure 5. Rotation curves of dwarf and LSB galaxies | 13 |
| Figure 6. A diagram illustrating how the simulation space is divided into nodes in the tree algorithm. | 16 |
| Figure 7. A particle which has been “smeared” into a mass cloud | 17 |
| Figure 8. A graph showing three different solutions to a Kepler orbit for sixteen orbits | 20 |
| Figure 9. Graphs of the energy of a Kepler orbit simulation using different time-stepping methods... | 21 |
| Figure 10. An example of Peano-Hilbert curves in two and three dimensions | 22 |

INTRODUCTION & BACKGROUND

1.1 The Existence of Dark Matter

A number of 20th century observations point to a discrepancy between the visible mass distribution in an astronomical system and the mass distribution implied by the observed state of bodies in the astronomical system. The most successful explanation of this discrepancy is that there is additional matter, called Dark Matter (DM), which does not interact electromagnetically and so cannot be detected directly. There are several pieces of evidence for DM, three of which are described below.

1.1.1 Rotation Curves

The earliest evidence for DM was the rotation curves of galaxies in galaxy clusters and stars in galaxies. Fritz Zwicky—a Swiss Astronomer who, at the time, was a professor and researcher at the California Institute of Technology—was the first person to discover that the observed rotation curves of galaxies orbiting in galaxy clusters do not match the Newtonian predictions based on the distribution of the visible matter. What Zwicky noticed was that the rotation speeds of galaxies in the Coma Cluster were essentially independent of distance from the center of the cluster, despite the fact that the distribution of visible matter in the cluster suggested that the galaxies outside the edge of the cluster should have been moving significantly slower than those at the edges [1].

To understand the Newtonian expectation, consider the gravitational force on a single nebula in a cluster with the visible edge at a distance R from the center of the cluster:

$$\mathbf{F}_g(\mathbf{r}) = \frac{GM(r)m}{r^2}(-\hat{\mathbf{r}}), \tag{1.1}$$

where r is the nebula's orbit radius, G is the gravitational constant, M is the mass of the matter inside

the nebula's orbit radius, and m is the mass of the galaxy. Since the gravitational force acts as a centripetal force on the orbiting nebula,

$$\mathbf{F}_g(r) = \frac{mv^2(r)}{r} (-\hat{\mathbf{r}}), \quad (1.2)$$

where v is the orbital speed of the galaxy as a function of r . Combining (1.1) and (1.2), the orbital speed can be found to be

$$v(r) = \sqrt{\frac{GM(r)}{r}}. \quad (1.3)$$

If the mass inside the orbit has a constant density ρ up to the edge of the cluster, it can be seen that M goes like the volume of the matter for galaxies inside the visible edge ($r < R$). Assuming the visible matter is the only significant contributor to the gravitational force, M is constant for galaxies outside the visible edge ($r > R$). Now, M can be written as a piecewise function:

$$M(r) = \begin{cases} \frac{4\pi}{3} \rho r^3 & r < R \\ \frac{4\pi}{3} \rho R^3 & r > R. \end{cases} \quad (1.4)$$

Substituting (1.4) into (1.3) and looking only at the r dependence yields

$$v(r) \sim \begin{cases} r & r < R \\ r^{-1/2} & r > R \end{cases} \quad (1.5)$$

and it can be seen that v is expected to decrease with r outside of the visible edge of the cluster. What

Zwicky observed, however, was that outside the visible edge of the Coma Cluster, v stayed fairly constant. He proposed that the discrepancy was due to matter which could not be detected visually, which he called “Dark Matter.”

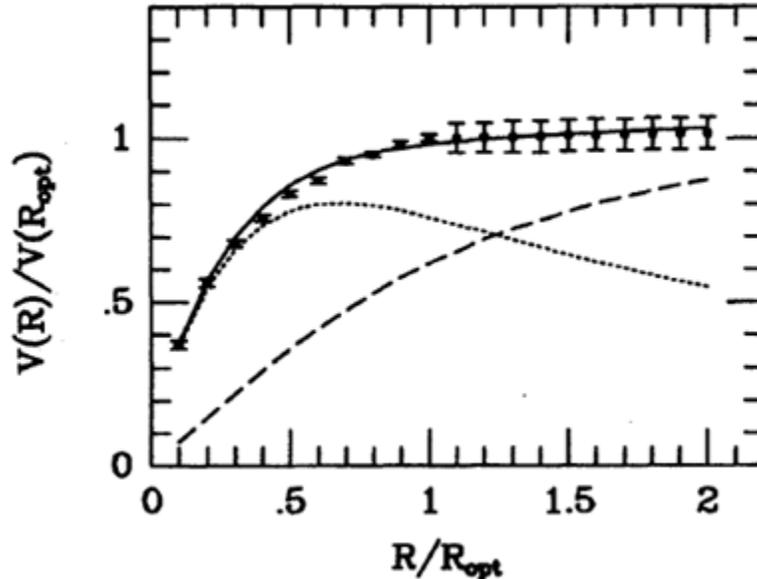


Figure 1. A graph of a synthetic rotation curve made by averaging 1,790 velocities from 70 galaxies. The solid line is a best-fit to the data, the dotted line is the contribution by the visible matter, and the dashed line is the contribution by the DM. R_{opt} is a scaling factor. Figure taken from Ref [4].

Zwicky realized that not only do rotation curves offer good evidence for the existence of DM, but they can also give important information about the structure of nebulae and other cosmological structures. By studying a nebula’s—or a galaxy’s—rotation curve, the radial mass distribution of the cluster can be inferred, even though most of the mass cannot be visually detected.

Despite Zwicky’s evidence of DM, the rest of the scientific community was not interested in the problem. Most people figured it was just a problem of uncertainty in measurements and would go away once more sophisticated instruments were developed [2]. However, by the end of the 1960’s,

others [3] also had noticed a discrepancy between observed rotation curves and what was expected from the visible light, in this case, the rotation curves of stars within galaxies, rather than galaxy clusters.

After studying many galaxies, it became apparent that this missing mass problem was common to galaxies and galaxy clusters alike. Figure 1 shows a rotation curve made by averaging 1,790 velocities from 70 galaxies [4]. This rotation curve is a subset of a compilation of about 1,100 galaxies and is a group of galaxies which have a similar brightness. The result of this subset is characteristic of the other rotation curves in the compilation.

1.1.2 Gravitational Lensing

Another piece of evidence for DM is gravitational lensing, a phenomenon of General Relativity.

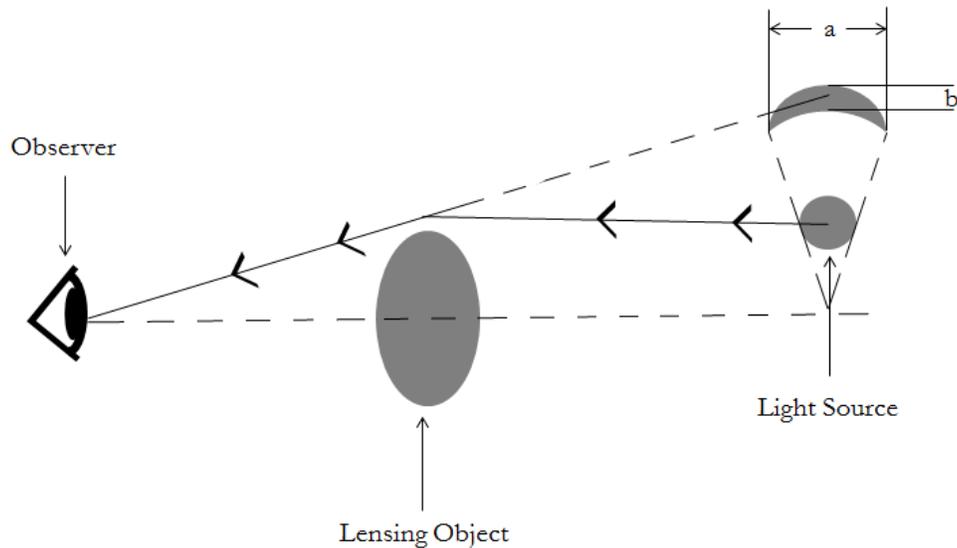


Figure 2. A diagram showing how light from a source is deflected by the lensing object's gravitational field. The altered location of the observed image is shown by the dashed line and the skewed image is represented by the crescent shape. The values a and b are the moments of distortion and can be used to determine the mass distribution of the lensing object.

According to General Relativity, mass curves space-time and as a result light is deflected by massive objects, as seen in Figure 2. As a result of the lensing, the light source appears to have originated from a different location than its actual position and the image is distorted and its shape appears to be smeared.

Gravitational lensing offers another piece of evidence for the existence of DM, as well as a method of determining the mass distribution of a lensing object, since the amount that the light is bent depends on how massive the lensing object is. This method is better than that of rotation curves [5] because it does not have to be assumed that the objects inside a galaxy or galaxy cluster are gravitationally bound to that larger cosmological object or that the system is, on average, in equilibrium. In addition, gravitational lensing can determine the mass and mass distribution of objects which are made up completely of DM and thus visually undetectable.

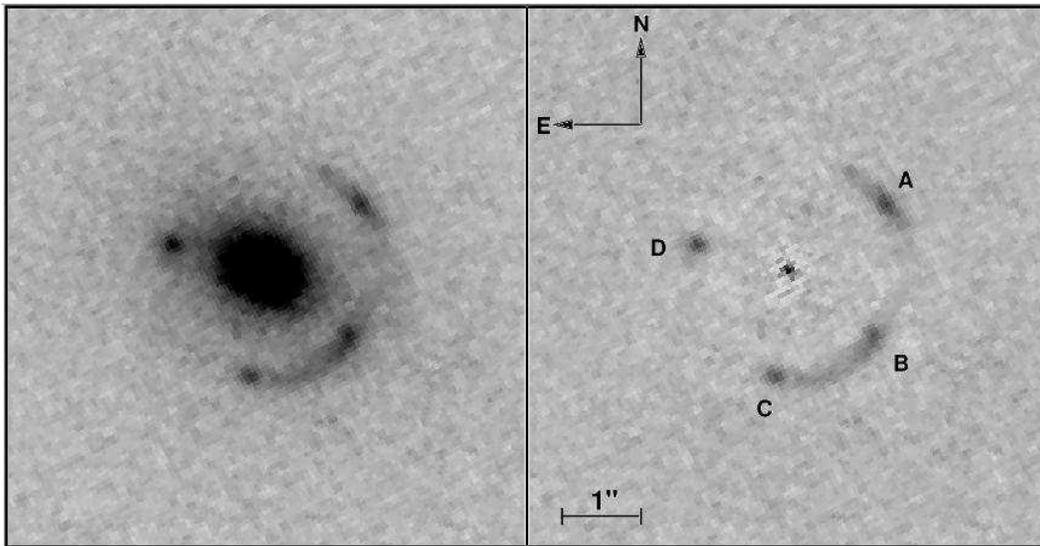


Figure 3. A photograph of a lensing galaxy—the large spot in the middle of the left image—and four objects being distorted by the gravitational lens—the spots labeled A-D in right image. The lensing galaxy has been removed in the right image by using different light filters. Figure taken from Ref [7].

The mass and mass distribution of the lensing object can be determined by measuring the moments of

distortion—the values a and b in Figure 2—of background images at different locations around the lensing object in pictures taken by telescopes (e.g. Figure 3). This can be done because there is a sufficiently high density of background galaxies anywhere in the sky [6]. These galaxies are useful in the case of gravitational lenses because they are behind most galaxies and galaxy clusters, providing a good background to be distorted by the lenses, and because they all share a blue color which is easily distinguishable from the red of most galaxies and galaxy clusters being studied. The mass of objects such as galaxies and galaxy clusters which have been measured using gravitational lensing is in good agreement with the mass which have been determined by rotation curves [5].

1.1.3 *The Bullet Cluster*

Though DM is one explanation for the discrepancy between visible mass in a cosmological system and its gravitational effects on other objects in that system, there are other solutions to the problem which have been proposed. One other solution is Modified Newtonian Dynamics (MOND) [8]. MOND is the hypothesis that, in the limit of small accelerations, acceleration is not directly proportional to the gravitational force, as is stated by Newton’s Second Law, but is, rather, proportional to the unmodified acceleration according to the function

$$\frac{a^2}{a_0} = g, \tag{1.6}$$

where a is the modified acceleration, a_0 is a scaling acceleration constant, and g is the unmodified acceleration. This modification reproduces a constant speed, which can be seen by solving for a , setting it equal to the centripetal acceleration, and using (1.1) for g :

$$v = \sqrt[4]{a_0 GM}. \tag{1.7}$$

This result matches the observed rotation curves without the need for undetectable matter. However,

the galaxy cluster 1E0657-558, commonly called the Bullet Cluster, provides evidence for the existence of a weakly interacting particle which results in undetectable mass in cosmological systems [9].

The Bullet Cluster is made up of two sub-clusters, one larger and one smaller, which have collided with one another. The bullet-like shape of the smaller sub-cluster indicates a collision and gives the cluster its name. Using gravitational lensing, the mass distribution of the Bullet Cluster has been determined and is shown in Figure 4. As can be seen, the highest mass concentration is at a different location than the visible X-ray gas of the cluster. This may be due to the gas in the two sub-clusters colliding and slowing down more than the DM, which is apparently collisionless. The study of the Bullet Cluster supports the particle nature of DM as opposed to a modified gravitational explanation of previous observations.

1.2 Motivation

The simplest model of DM is cold dark matter (CDM), which only interacts via the gravitational and weak forces. A CDM model which includes a cosmological constant does well at explaining large-scale ($\gg 1$ Mpc) cosmological observations [11], such as the rate of expansion and topology of the Universe and fluctuations of the cosmic microwave background. However, with higher resolution telescopes and computer simulations, there is evidence that suggests that the CDM predictions do not agree with observational evidence on the galactic scale (\leq a few Mpc) [12]. One such piece of evidence is that the radial dependency of galaxy mass densities predicted by CDM simulations is steeper than what is observed from rotation curves of galaxies. This discrepancy between the dependencies can be seen in the rotation curves in Figure 5. The observational data, shown by circles, are from dwarf and low surface brightness (LSB) galaxies and have been taken from Ref [13]. They have been normalized to a characteristic scale length r_s , which fits that data from the different galaxies to a single shape of a rotation curve and can be done since rotation curves of DM dominated galaxies tend to have the same shape. The rotation curve from the CDM simulation has been scaled two different ways, one so that it fits to the asymptote of the data, and the other so that it fits better at small orbital radii [14]. It can be seen that, no matter how the theoretical predictions are normalized, the rotation curves increase too quickly to fit the observed data.

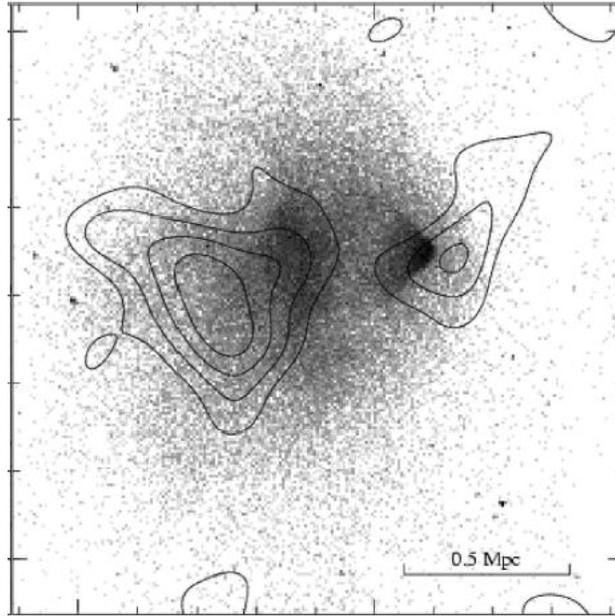


Figure 4. An image of the X-ray gas of the Bullet Cluster with mass distribution contours overlaid on top. The X-ray gas image is from Ref [10], and the contours are from Ref [9]. Figure taken from Ref [9].

In addition to the mass distribution of DM dominated galaxies being inconsistent with CDM predictions, there is also a discrepancy between observations and CDM predictions of how mass density depends on total mass. CDM predicts that the central density of haloes should vary with mass, with more massive haloes being less concentrated than less massive ones [15]. However, observational data of galaxy central densities does not seem to show any dependence on the mass of the galaxy, but rather that most galaxies appear to have a central density of $\approx 0.02 M_{\odot}/\text{pc}^3$, regardless of total mass [16], where M_{\odot} is a solar mass. Another piece of evidence against CDM is the scarcity of dwarf galaxies—groups of matter which are smaller than and located inside and around galaxies—which are observed. Kauffman, White, and Guiderdoni [17], using a semi-analytic CDM model of the Local Group, found that the number of dwarf galaxies which are predicted by CDM is larger than the number observed in the Local Group by a factor of 5-10. In the model, there were hundreds of objects whereas there are only twenty-eight known members of the Local Group.

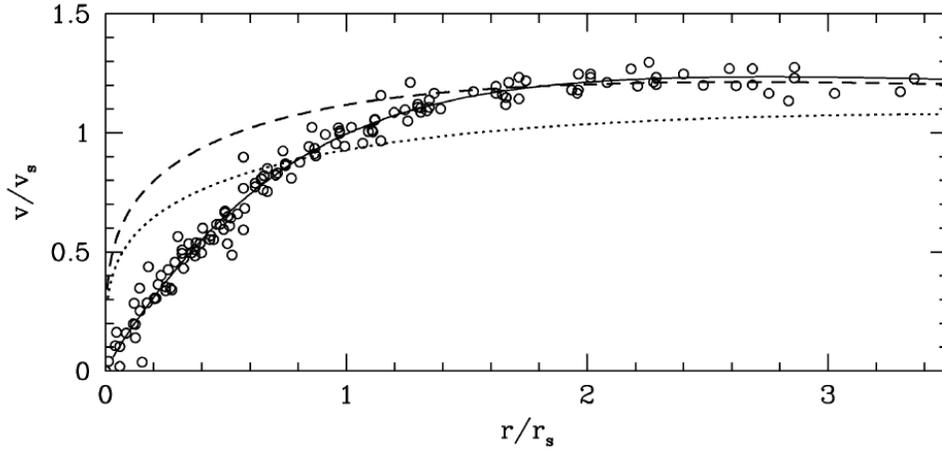


Figure 5. Rotation curves of dwarf and LSB galaxies. The circles are the data from Ref [13]. The values r_s and v_s are scaling factors. The solid line is a fit to this data and the dashed and dotted lines are the rotation curves of the simulations from Ref [14], normalized using two different methods, one to fit the data at the asymptote, the other to fit the data at small orbital radii. Figure taken from Ref [14].

One solution to these problems could be self-interacting DM (SIDM), first proposed by Spergel and Steinhardt [12]. Unlike CDM, SIDM particles can collide with other SIDM particles, like billiard balls hitting one another. SIDM theoretically predicts lower central densities independent of mass and a shallower radial dependence on density than CDM because, at high central densities, collisions are so common that the number of particles escaping from the center of the halo is the same as the number of particles going in, thus maintaining a stable core. In addition, SIDM predicts fewer dwarf galaxies, for two reasons. The first is because SIDM haloes are more likely to be affected by tidal forces, due to their lower densities and larger core radii. The second is because the DM in a dwarf galaxy's halo is stripped when it collides with a larger halo, since the DM in the larger halo has larger velocities. The DM in larger haloes has a higher speed because of the halo's stronger gravitational potential which, when the halo is at equilibrium, results in a higher kinetic energy. Since collisions are unlikely at low densities, SIDM does not predict anything different than CDM at large scales where CDM has been

successful. SIDM has had some success at reproducing, in computer simulations, the observed inner properties of DM [18].

A cluster was built in order to simulate DM which is similar to SIDM but has, rather than billiard-ball-like collisions, a Coulomb-like self-interaction. This theoretical DM is atomic in nature, similar to visible matter, but whose Coulomb-like interaction is governed by different a gauge boson [19]. The interaction is said to be Coulomb-like because it is a long-range radial force whose magnitude goes like r^{-2} and particles have a positive, negative, or neutral “charge.” Like the Coulomb force, like charges repel one another, opposite charges attract one another, and neutral charges have no interaction other than gravity.

Chapter 2

THEORY

2.1 N-body Simulations

Typically, the number of bodies in a cosmological system—commonly labeled N —is very large (e.g., the number of stars in a galaxy). Since the bodies experience gravitational forces from each other, the number of forces acting on a body depends on $N - 1$. Each body has this many forces acting upon it, so the number of calculations needed to evaluate the gravitational forces exactly for all the particle goes like $N(N - 1)$.

There are too many particles to obtain the positions and velocities by analytically solving Newton’s equations of motion:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \tag{2.1}$$

$$m \frac{d\mathbf{v}}{dt} = \mathbf{f}, \tag{2.2}$$

where \mathbf{x} , \mathbf{v} , and m are the position, velocity, and mass of the particle, respectively, and \mathbf{f} is the force acting on the particle. Because of these complexities, N-body systems must be solved numerically.

This numerical solution can be done on a computer by simulating particles in a virtual space. These simulations are known as N-body simulations. The positions and velocities of these particles after a time step are calculated based on the forces acting on each one. For a DM simulation, these forces are gravity and any self-interactions being studied. At each time step, a “snapshot” of the simulation is taken by the computer. This is a record of the position and velocity of each particle and is recorded so that it can be seen how the distribution of particles changes as time progresses.

2.1.1 Force Calculations

The gravitational forces acting on a particle in a cosmological N-body simulation are calculated from the positions of the other particles in the simulation. A difficulty arises in accurately and efficiently assigning positions to each particle. There are several solutions to this problem, two of which are of interest for this project: the tree algorithm and the Particle Mesh (PM) methods.

In the tree algorithm method, the space of the simulation is divided into smaller and smaller sections, called nodes, illustrated in Figure 6. Rather than calculating the force from each particle exactly, the tree algorithm uses the multipole of the particles in the undivided space, called the root node, to calculate the force on a particle. Using the multipole to calculate the force reduce the number of calculations for a single particle from $N - 1$ to the order of $\log N$ [20]. However, it also introduces some error into the computation. This error can be reduced by using the multipole of the smaller nodes, called opening a node. The computer tests the accuracy of the force calculation in some way, such as comparing it with the force calculated in the previous time step, and, if the calculation is

determined to not be acceptably accurate, the node is opened. This process is repeated until the specified level of accuracy is achieved.

In the Particle Mesh method, the force on one particle is calculated by assigning the positions of the other particles to vertices on a grid or mesh. The crudest way to do this, called the nearest grid point (NGP) method, is to treat each particle as a point mass and assign its position to its nearest vertex. A more sophisticated way of doing this, called clouds-in-cells (CIC) method [21], is to “smear” the particles into cubes, called clouds, and assigning them to several vertices based on the mass distribution of the cloud, as illustrated in Figure 7. In the diagram, the mass in the bottom left section of the cloud is assigned to the vertex at position (i, j) , the mass in the bottom right section of the cloud is assigned to the vertex at position $(i+1, j)$, the mass in the top left section of the cloud is assigned to

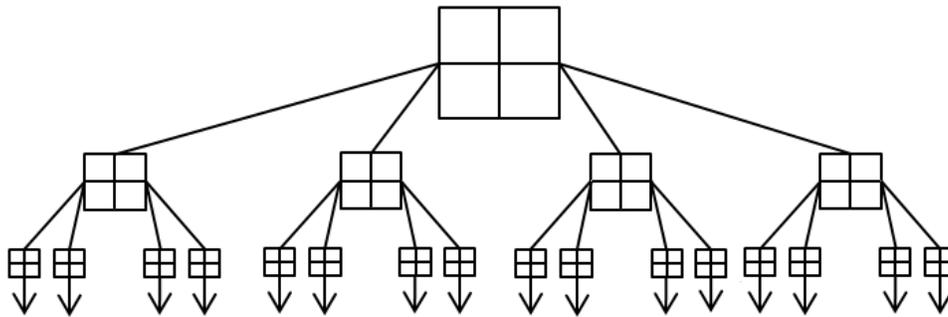


Figure 6. A diagram illustrating how the simulation space is divided into nodes in the tree algorithm.

the vertex at position $(i, j+1)$, and the mass in the top right section of the cloud is assigned to the vertex at position $(i+1, j+1)$. The CIC method is a better approximation of the position of the particle than the NGP method and introduces less error into the force calculation; however, it also requires more processing [22].

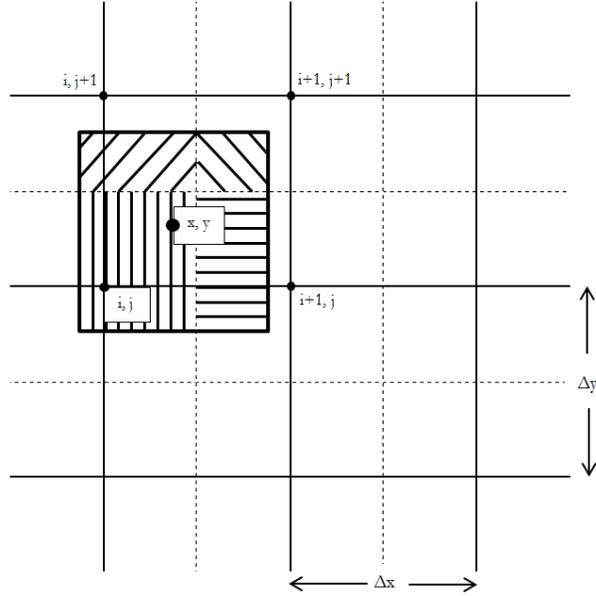


Figure 7. A particle at position (x, y) which has been “smeared” into a mass cloud. Shading is given to show the four different sections of the cloud which are assigned to their respective vertices.

Once the particles have been laid out in the grid, the mass density at position (i, j) is found by summing over the clouds as

$$\rho(i, j) = \sum_{clouds} a_{ij} \rho_c(x, y) \quad (2.3)$$

where ρ_c is the density of the cloud at position (x, y) and a_{ij} is the area of the cloud in the cell centered at position (i, j) divided by the area of the cell. The total density can then be determined from the density at each vertex. This total density is then transformed into Fourier space, where the calculation of the potential from Poisson’s equation becomes simpler and less processor intensive [20]. Poisson’s equation for gravity comes from substituting

$$\mathbf{f}(\mathbf{x}) = -\nabla \Phi(\mathbf{x}), \tag{2.4}$$

which can be written since the gravitational force is conservative, into Gauss' Law of gravity:

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = -4\pi G\rho(\mathbf{x}). \tag{2.5}$$

Written out, Poisson's equation for gravity is

$$\nabla^2 \Phi(\mathbf{x}) = 4\pi G\rho(\mathbf{x}), \tag{2.6}$$

where $\mathbf{f}(\mathbf{x})$ is the gravitational force and ρ is the mass density which was determined from the mesh. Poisson's equation can be written in real-space as a convolution of the mass density with a Green's function:

$$\Phi(\mathbf{x}) = \int g(\mathbf{x} - \mathbf{x}')\rho(\mathbf{x}') d^3 \mathbf{x}', \tag{2.7}$$

where $g(\mathbf{x}-\mathbf{x}')$ is the Green's function. In Fourier-space, this equation is no longer an integral, but simply a multiplication:

$$\hat{\Phi} = \hat{g}(\mathbf{k}) \cdot \rho(\mathbf{k}), \tag{2.8}$$

where the "hats" indicate that the functions are now in Fourier-space and \mathbf{k} is a wave-number vector. Once the gravitational potential is calculated, it is transformed back into real-space and, hence, the gravitational force is calculated.

The PM method is a faster calculation of the gravitational force than the tree algorithm, since the complicated computations of the potential are made easier by doing them in Fourier-space. Another advantage of the PM method over the tree algorithm is that there is less error in assigning clouds to a grid than from the multipole approximations of the tree algorithm [20]. However, the resolution of a simulation is limited by the mesh size.

The limitations of each method can be overcome by using a hybrid method called the TreePM method [23]. In the TreePM method, the short-range forces are calculated using the tree algorithm and the long-range forces using the PM method. Using the tree algorithm only at short ranges, which is usually set to be slightly larger than the mesh size, maintains a high resolution and limits the number of force computations done using the tree algorithm which lowers error and processing time. Using the PM method at long-ranges allows for the speed and accuracy of the PM method without sacrificing resolution, since the particles are farther away.

2.1.2 Time Integration

In an N-body simulation, rather than solving (2.1) and (2.3) with a continuous function for the force, the positions and velocities are solved numerically by calculating the forces on the particles for each momentary time step. This force is then used to update the positions and velocities for the next time step.

It is not necessarily required that the positions and velocities be updated all at once as with the Euler method of integration in which the positions and velocities are updated as follows:

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \frac{\mathbf{f}(t_i)}{m} \Delta t \tag{2.9}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i \Delta t, \tag{2.10}$$

where \mathbf{v}_i is the velocity of the particle at time-step t_i , \mathbf{v}_{i+1} is the velocity at the next time-step, \mathbf{x}_i and \mathbf{x}_{i+1}

are the positions of the particle at each respective time-step, $\mathbf{f}(t_i)$ is the force acting on the particle at t_i , and m is the mass of the particle. In fact, it turns out that this type of time integration results in large disagreements between the numerical and exact solutions (see Figure 8) as well as a loss of total energy in the simulation (see Figure 9), meaning energy is not conserved.

These errors are due to the fact that the above equations are not symplectic. A time integration scheme is symplectic if its operators come from an expansion of the equation of motions from a canonical transformation using the Hamiltonian of the system. N-body simulations are more accurate when using symplectic integrators because they are Hamiltonian systems and so the symplectic integrator does not change the phase space volume of the simulation [20].

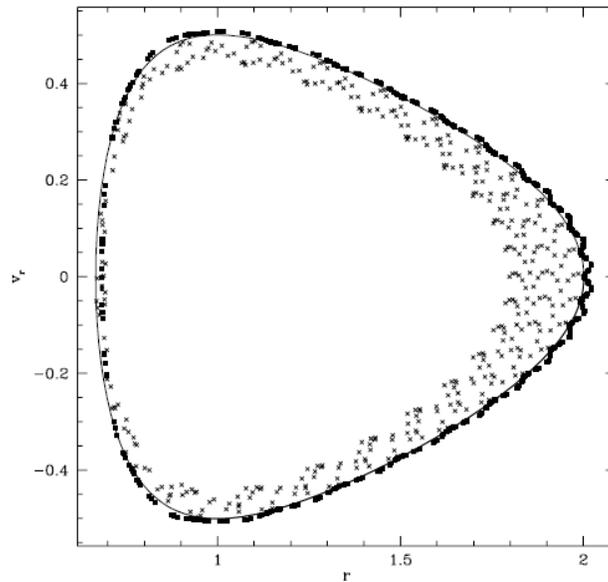


Figure 8. A graph showing three different solutions to a Kepler orbit for sixteen orbits. The squares are the solution using a symplectic integrator, the crosses are the solution using a non-symplectic integrator, and the solid line is the exact solution. Note the solution using a symplectic integrator oscillates around the exact solution whereas the discrepancy between the solution using the non-symplectic integrator and the exact solution grows which each orbit. Figure taken from Ref [24].

One method of time integration which is symplectic is the leapfrog method [24]. In this method, there are two different steps, the kick and the drift. In the kick step, the velocity is updated and in the drift step, the position is updated. These updates are a second order truncation of the canonical transformation equation, however, unlike other second order integration schemes, they only require one force computation per time step. Leapfrog integration is done by alternating the kick and drift

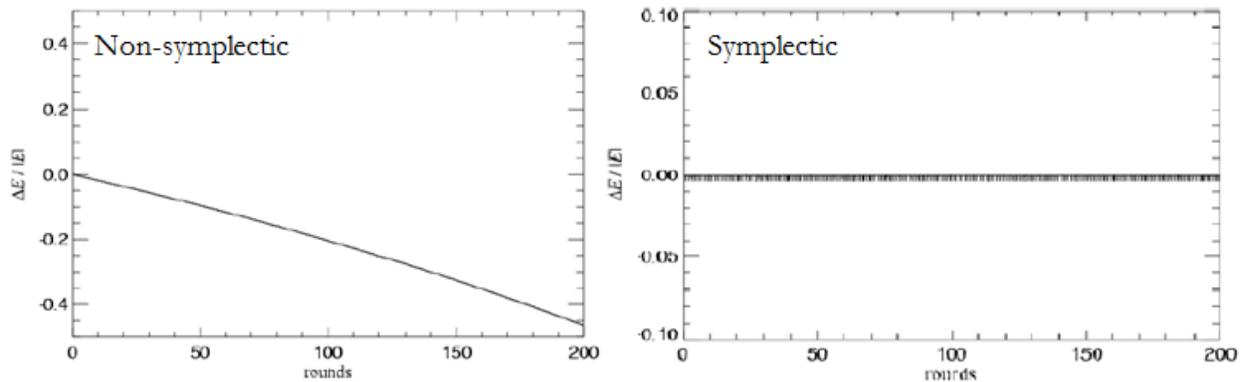


Figure 9. Two graphs of the total energy of a simulation of a Kepler orbit using two different time integration techniques. The left graph used a non-symplectic time integrator and the right used a symplectic integrator. The x axis is the number of time-steps taken and the y axis is the change in total energy of the system normalized to the total energy. Not that the change in total energy of the non-symplectic integrator simulation grows increasingly negative, whereas the change in total energy of the symplectic integrator simulation oscillates around zero. Ideally, the change in total energy will be zero, i.e. energy is conserved in the simulation. Figure taken from Ref [25].

steps. What makes leapfrog integration different than Euler integration is that rather than updating the position and velocity all the way through the time step at once, in the case of the Kick-Drift-Kick method, the velocity is updated halfway through the time step, then the position is updated to the next time step using the velocity halfway through the time step calculated in the previous step, and finally the velocity halfway through the time step is updated to the next time step using the same force used in the first kick step. This can be written mathematically as

$$\mathbf{v}_{i+1/2} = \mathbf{v}_i + \frac{\mathbf{f}(\mathbf{x}_i)}{m} \Delta t, \tag{2.11}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_{i+1/2} \Delta t, \tag{2.12}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_{i+1/2} + \frac{\mathbf{f}(\mathbf{x}_{i+1})}{m} \Delta t. \tag{2.13}$$

In addition, the error in the Leapfrog method goes like Δt^2 , rather than Δt , like the Euler method [20].

2.2 Parallel Computations

Since the force on a particle depends only on the positions of the other particles, the force acting on one particle can be calculated separately from the forces acting on the other particles and so these calculations are well suited for parallel computations. This means that the calculations can be divided up and done by different processors. Much time can be saved by dividing the work up like this, since multiple calculations can be done at once.

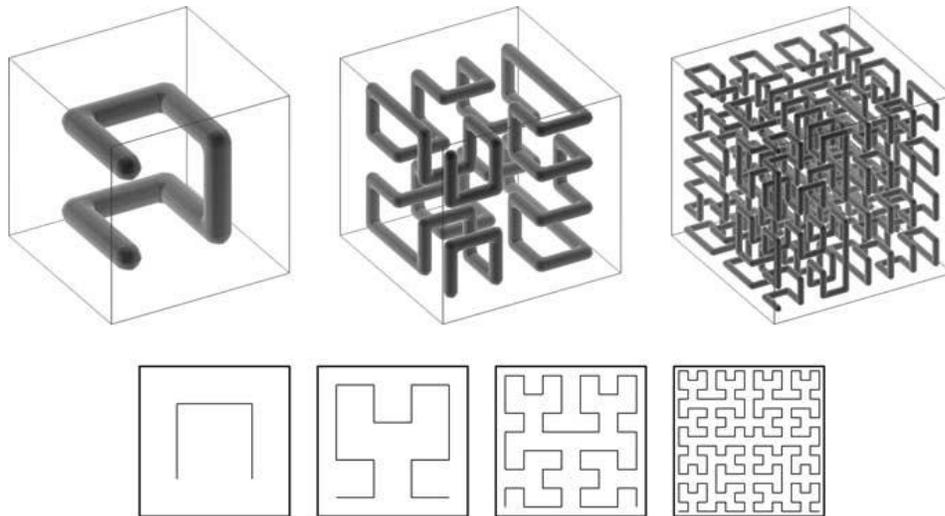


Figure 10. An example of Peano-Hilbert curves in two and three dimensions. Figure taken from Ref [20].

When computations are done in parallel, there is a main processor, called the controller, on which the job is started. The controller then sends computations to be done on other processors in the cluster, called nodes. These nodes then do the calculations they have been assigned and send the results back to the controller which compiles and stores the data.

The method used in this project for determining which nodes get which jobs is by splitting the simulation space into different domains using a Peano-Hilbert curve [20]. A Peano-Hilbert curve is a type of space-filling fractal that can be used to describe a three-dimensional space as a one-dimensional like this is so that the approximation errors do not depend on the number of processors. This is because of the similarities between splitting space up using the Tree algorithm and a Peano-Hilbert curve.

Chapter 3

EXPERIMENT

3.1 Overview

In order to run N-body simulations which test the distribution of DM with Coulomb-like interactions, a cluster of two computers and six processors was built. The initial cluster was small so that building it was simple and the concepts of setting up a cluster could be learned easily. Once the small test cluster was built, the server could be expanded with more computers without any additional knowledge or software. As of the time of this paper's writing, no steps have been taken to expand the cluster.

3.2 GADGET-2

GADGET-2 [20], a cosmological simulation code written by Volker Springel in ANSI C, was the software which was used to run the N-body simulations. It can run simulations in parallel, distributing jobs to nodes using Peano-Hilbert curves as discussed above, and uses the TreePM algorithm and KDK leapfrog technique to calculate the gravitational forces acting on the particles and the time integration, respectively. The options in GADGET-2 are varied and possible simulations range from

Newtonian Mechanics simulations to purely DM simulations to complex simulations of the formation of the Universe.

GADGET-2 uses three open-source software libraries to run its simulations: the Message Passing Interface (MPI) [26], the GNU science library (GSL) [27], and the Fastest Fourier Transform in the West (FFTW) [28]. MPI is what gives instructions for the parallel computations. GSL provides a library of mathematical routines, such as random number generation and matrix operations. FFTW is used to perform the Fourier transforms of the PM force calculations. These software libraries must be installed on all of the computers in the cluster in order to run simulations.

There were three files in the GADGET-2 folder which needed to be edited in order to run a simulation: the Makefile, the parameter file, and the initial conditions file. The Makefile makes an executable which is run by MPI to start the simulation. Inside the Makefile, there are settings which can be changed to set the path to where the libraries for GSL, FFTW, and MPI are stored on the computer. The parameter file sets the options for the simulation depending on what type of simulation is desired. For example, this is the file which is edited to set the particles to be purely DM particles. The parameter file also directs the simulation program to the initial conditions file and sets where the snapshots are stored. The initial conditions file holds the information of the positions and velocities of the particles at the first time-step; it is essentially the starting snapshot of the simulation. It is written using the GNU Data Language (GDL), an open source numerical data analysis and imaging tool. The initial conditions are read by the parameter folder once the executable which was made by the Makefile has been run.

GADGET-2 includes files of code which contain all of the instructions for the computer to run simulations. These files hold the information for running force computations and they are what need to be modified so that the simulations include the Coulomb-like self-interactions.

Once a simulation has been run, the GDL can be used to image and compile the snapshots into a video of the simulation. The snapshots can also be used to measure characteristics of the simulation system, such as halo densities and number of dwarf galaxies.

3.3 TORQUE

TORQUE was the Resource Manager used for the cluster. It was the software that controlled the communication between computers. Unlike MPI, which gave the instructions for passing information, TORQUE was the program which controlled which nodes were given which jobs and did the actual information passing.

To setup the server, TORQUE was first installed on to the controller. The server must then be configured, meaning the controller must be given the name and number of nodes. Then, TORQUE must be installed to all of the nodes, which is done by making install files on the controller which are sent to the nodes and installed. Once TORQUE is installed on all the nodes, they must be configured by creating a file containing the controller name on each one. Finally, the computers must be authorized to communicate with each other.

3.4 The Cluster

The cluster was made out of two Apple iMacs, one had two processor cores (model ID iMac11,2) and the other had four (model ID iMac12,1). The computers were connected by Ethernet cables so that there would be less of a delay in the data passing compared to that of a wireless connection. An example simulation which is included with GADGET-2 of a collision between two galaxies was run to confirm that the cluster was working properly.

3.5 Conclusions and Suggestions for Future Work

Now that a working test cluster has been built, more computers can be added to grow the cluster. This will improve processing power and decrease the amount of time it takes to complete a simulation. Another way to decrease processing time would be to use a more sophisticated scheduler. TORQUE comes with its own scheduler, which determines which nodes to send computations to and when, but it is simple and rudimentary. By having a more advanced scheduler, not only could processing time be decreased, but options could be available to not use a node if it is in use by another program, allowing for individual computers in the cluster to be used for other purposes even during a simulation.

Since the cluster has been built to test a Coulomb-like DM self-interaction, the GADGET-2 code must be modified to include this force. Since GADGET-2 uses the TreePM method to calculate forces, care must be taken so that this new interaction is include in both the short and long range computations of the force. A charge characteristic must be added to the DM particles so that the DM can be positively, negatively, or neutrally charged and the force can be attractive or repulsive. This characteristic would have to be recorded by the snapshot files and set in the initial conditions.

Once these modifications have been made, DM simulations could be run. From the simulation results, the characteristics of the system relevant to the CDM problem discussed above could be determined. Results could be compared between simulations including CDM, Coulomb-like self-interacting DM, and a mix of the two.

References

- [1] F. Zwicky, *Astrophys. J.* **86**, 217 (1937).
- [2] M. Persic, P. Salucci and F. Stel, *M.N.R.A.S.* **281**, 27 (1996).
- [3] B. Parker, *Invisible Matter and the Fate of the Universe* (Plenum Press, New York and London, 1989).
- [4] V. C. Rubin, *Sci. Am.* **248**, 96 (1983).
- [5] A. Tyson, *Phys. Today* **45**, 24 (1992).
- [6] L. V. E. Koopmans and T. Treu *Astrophys. J.* **583**, 606 (2003).
- [7] J. A. Tyson, F. Valdes and R. A. Wenk, *Astrophys. J.* **349**, L1 (1990).
- [8] M. Milgrom, *Astrophys. J.* **270**, 365 (1983).
- [9] D. Clowe, A. Gonzalez and M. Markevitch, *Astrophys. J.* **604**, 596 (2004).
- [10] N. A. Bahcall, J. P. Ostriker, S. Permuter and P. J. Steinhardt, *Science* **284**, 1481 (1999).
- [11] D. N. Spergel and P. J. Steinhardt, *Phys. Rev. Lett.* **84**, 3760 (2000).
- [12] A. V. Kravstov, A. A. Klypin, J. S. Bullock and J. R. Primack *Astrophys. J.* **502**, 48 (1998).
- [13] M. Moore, T. Quinn, F. Governato, J. Stadel and G. Lake, *M.N.R.A.S.* **310**, 1147 (1999).
- [14] M. Markevitch, A. H. Gonzalez, L. David, A. Vikhlinin, S. Murray, W. Forman, C. Jones and W. Tucker, *Astrophys. J.* **567**, L27 (2002).
- [15] J. F. Navarro, C. S. Frenk and S. D. M. White, *Astrophys. J.* **462**, 563 (1996).
- [16] C. Firmani, E. D'Onghia, V. Avila-Resse, G. Chincarini and X. Hernández, *M.N.R.A.S.* **315**, L29 (2000).
- [17] G. Kauffman, S. D. M. White and B. Guiderdoni, *M.N.R.A.S.* **264**, 201 (1993).
- [18] R. Davé, D. N. Spergel, P. J. Steinhardt and B. D. Wandelt, *Astrophys. J.* **547**, 574 (2001).
- [19] D. E. Kaplan, G. Z. Krnjaic, K. R. Rehermann and C. M. Wells, *J.C.A.P.* **1005**, 021 (2010).
- [20] V. Springel, *M.N.R.A.S.* **364**, 1105 (2005).
- [21] C. K. Birdsall and D. Fuss, *J. Comput. Phys.* **135**, 141 (1969).
- [22] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, 1981).

- [23] J. S. Bagla and S. Ray, *New Astron.* **8**, 665 (2003).
- [24] T. Quinn, N. Katz, J. Stadel and G. Lake, (unpublished)1997.
- [25] V. Springel, Summer School on Cosmological Numerical Simulations, 3rd Week – Wednesday (Lecture), Potsdam, 2006.
- [26] P. Balaji, W. Bland, J. Dinan, D. Goodell, W. Gropp, R. Latham, A. Peña and R. Thakur, (Manual) 2013.
- [27] The GSL Team, (webpage) 2013
http://www.gnu.org/software/gsl/manual/html_node/. [Accessed 27 September 2013].
- [28] M. Frigo and S. Johnson, *Proceedings of the IEEE* **93**, 216 (2005).