A thin disk galaxy simulator for astrophysics pedagogy

J.S. Olsson and G.F. Vrijsen
Department of Physics, University of Illinois at Urbana-Champaign
1110 West Green Street, Urbana, IL 61801-3080, USA

Abstract

The Digital Demo Room (ddr) is a project to create an online collection of astrophysics simulators for pedagogical purposes, including ddr-galaxy. In this paper, the workings of ddr-galaxy, a 2-d disc galaxy simulator, are explained. ddr-galaxy's primary motivation is to assist students in learning about galactic dynamics, by providing a easy to use galaxy simulator which will eventually include a simple, graphical, web interface. The major sections of the simulator are discussed and explained in turn.

1 Preliminary Overview

A complete n-body code contains many carefully constructed subroutines. Much of this code, including rendering interfaces, extraneous functions for development purposes, etc. is of little interest – both from the perspective of physics and of pedagogy. The bulk of this paper will deal, therefore, with the subsections of ddr-galaxy that are of general interest to the design of galactic n-body codes. Namely, this includes background potentials, potential solvers, trajectory integrators, and initial conditions routines (other considerations, more specific to ddr-galaxy, are briefly outlined below). Finally, a brief example of data from a ddr-galaxy sample run is considered.

1.1 Specific Considerations

ddr-galaxy is written in ANSI C, and was primarily developed on Alpha EV56 workstations. Both the GNU C Compiler (gcc) and Compaq's Alpha compiler (ccc) were utilized, for debugging and optimization purposes respectively.

Screen rendering is implemented using the OpenGL graphics API, both to on-screen and off-screen contexts (the latter for MPEG creation). A freely available MPEG library, libfame, is used to generate movie files, which can then be accessed via a WWW interface.

2 Simulator Components

2.1 Analytic Background Potentials

ddr-galaxy is capable of evolving particle trajectories with or without the influence of interparticle gravitation. In the absence of such self-gravity, the ability to impose an analytic background potential is clearly important– by using analytic functions which approximate the potential field of a galaxy, the solutions to particle trajectories may be viewed with minimal computational cost (trajectory integration is O(N), while the potential is differenced using the O(1) analytic background potential). However, under self-gravity, the imposition of an analytic background potential can also be extremely useful– to model the effect of a bulge or halo potential near the center of the disk, for example.

2.1.1 Power Law Potential

A simple background potential is of the form

$$\phi(r) = \frac{1}{(r^2 + \epsilon^2)^{n/2}}$$

where ϵ is a small softening parameter[1]. Typically, n may be set to unity for a Keplerian potential. For small N, the trajectories of non-self-gravitating particles can be evolved as if on orbits about a fixed, central, massive object. With self-gravity, this field is an appropriate choice of background potential for large N systems with bar instabilities, effectively "pinning down" the center of mass (which is particularly important when mass may leave the system, i.e.: mass leaves the grid on which the potential is solved). This potential field is identical to the Hernquist[2] potential for $n = 1$.

For a general potential $\psi(r, \theta)$, the Lagrangian is $L = \frac{1}{2}m(\dot{r} + r\dot{\theta})^2 - \psi$. Solving the Euler-Lagrange equation for the general variable r, we have $\ddot{r} = r\dot{\theta}^2 - \frac{\partial \psi}{\partial r}$. For cold, circular orbits ($\ddot{r} = 0$) this reduces to the useful relation

$$\dot{\theta} = \sqrt{\frac{1}{r}\frac{\partial \psi}{\partial r}} \qquad (1).$$

The orbital period of a particle on such a trajectory is then simply

$$T = \frac{2\pi r}{v_c} \qquad (2).$$

From this equation, it is clear that the circular velocity of a point mass in such a Keplerian potential is the familiar

$$v_c = \sqrt{\frac{GM}{r}}$$

with orbital period

$$T = \frac{2\pi r^{3/2}}{\sqrt{GM}}.$$

2.1.2 Logarithmic Potential, Time Dependant/Independent Bar

The general form of the logarithmic potential is as follows,

$$\phi(r, \theta) = \frac{1}{2}\log(1 + r^2(1 + \epsilon \cos^2(\theta - \Omega t)))$$

where ϵ is a parameter which scales the effect of the bar, while Ω is its angular frequency. This background potential is not in itself useful for self-gravitating simulations, but is a useful analytic approximation of a galaxies bar potential.

2.1.3 Harmonic Potential

A particle in a harmonic potential undergoes simple harmonic oscillations. In two dimensions, this potential function is

$$\phi = Ax^2 + By^2$$

where A and B are parameters which determine the relative strength of the field in both x and y respectively. Many interesting trajectories can be plotted for this potential, i.e.: various Lissajous curves and "box" orbits. Figure 1 shows one such curve below.

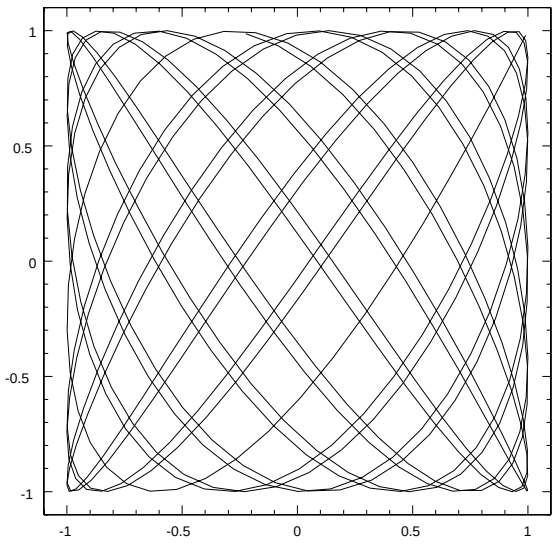

Fig. 1 The trajectory of a particle in a harmonic potential of the form $\phi = x^2 + 2y^2$.

2.1.4 Isochrone Potential

A reasonable approximation for a disc galaxy might assume roughly constant density near the center, while the density goes to zero at larger radii. Such a potential is the Isochrone potential, specified as

$$\phi(r, \theta) = \frac{GM}{b + \sqrt{b^2 + r^2}}$$

M is the total mass of the system, while b is a constant which determines the systems "linearity". In other words, as b grows, the potential becomes more linear in r. A plot of the Isochrone potential vs. radius was generated for several values of b, and is included as Figure 2 below.

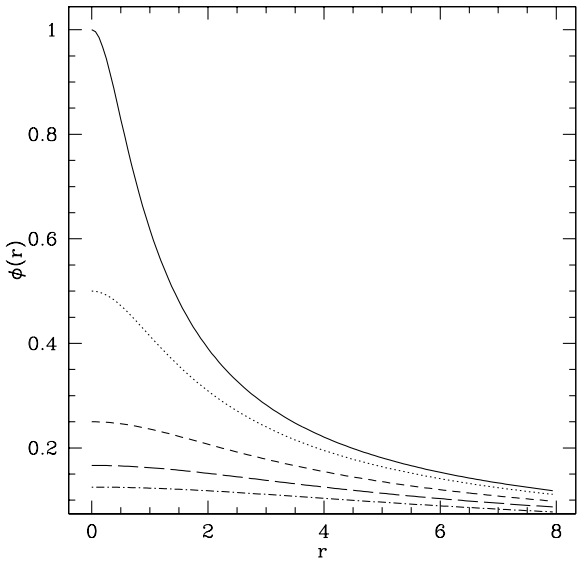

Fig. 2 Potential vs. Radius for several values of the parameter b in the isochrone potential. From top to bottom, b=.5,1,2,3,4.

By equation (1) we have for the circular speed at radius r

$$v_c = \sqrt{\frac{GMr^2}{(b+a)^2 a}},$$

where

$$a \equiv \sqrt{b^2 + r^2}.$$

2.1.5 Homogeneous Sphere Potential

The potential due to a homogeneous sphere, on a plane which intersects the sphere through its origin takes the form

$$\phi(r,\theta) = \begin{cases} -2\pi G\rho(a^2 - \frac{1}{3}r^2), & r < a \\ -\frac{4\pi G\rho a^3}{3r}, & r > a, \end{cases}$$

where a is the radius of the sphere. This potential is similar to the potential due to a halo near the center of an otherwise thin disc galaxy, and accordingly, is of practical use as a background potential for a self-gravitating system.

Using equation (1) again, the circular velocity for a particle in this potential field will be

$$v_c = \sqrt{\frac{4\pi G\rho}{3}} r.$$

Equation (2) yields

$$T = \sqrt{\frac{3\pi}{G\rho}}$$

for the orbital period.

2.1.6 Superposition of Potentials

In addition to imposing lone background potentials, it is often useful to impose a superposition of several different potentials. At the conclusion of this paper, a brief summary of a run involving such a superposition is included. In that particular run, a Keplerian, homogeneous sphere, and logarithmic potential were used. This combination aimed to model a large disc galaxy, with a halo and a central compact object.

2.2 Potential Solvers

2.2.1 Direct Summation

One obvious (and terribly inefficient) method for solving particle accelerations is the direct summation method. The algorithm simply iterates through each particle and sums the forces due to all other particles in the system. In x and y, these equations take the form

$$g_{x_{i,j}} = Gm \sum_{i \neq j} \frac{x_i - x_k}{r_d^3}$$

$$g_{y_{i,j}} = Gm \sum_{i \neq j} \frac{y_i - y_k}{r_d^3}$$

where $r_d = r + \epsilon$ and $r = \sqrt{(x_i - x_k)^2 + (y_i - i_k)^2}$.

Accordingly, while this allows for as much precision as the machine architecture permits, the operation scales as $O(N^2)$. For large N, this compares rather unfavorably to more advanced particle-mesh algorithms. However, it is interesting for pedagogical purposes (the primary motivation of ddr-galaxy), and has therefore been included as an option. Figure 3 below demonstrates how this method compares computationally to the Fourier potential solver.

2.2.2 Fourier Method

A considerably more efficient method of calculating the potential on the plane involves Fourier methods. Some accuracy is lost as the system becomes "discretized", but such algorithms are overwhelmingly more efficient as seen in Figure 3.

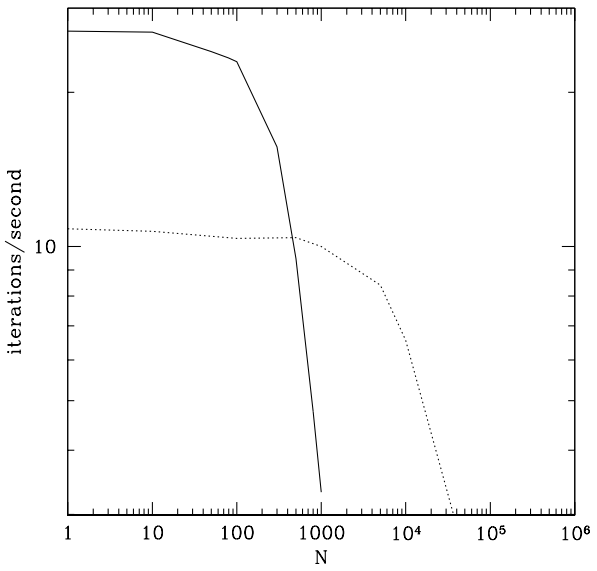

Fig. 3 Plotted here is the number of iterations per second vs. number of particles for both the direct summation (dotted line) and Fourier method (solid line) of potential solvers. It is clear that as N grows large, the computational cost of the direct summation method quickly becomes prohibitive. Note that this is not a fair comparison per se, since the number of bodies *and* the gridsize (here 256x256) effect the efficiency of the Fourier method. However, for any gridsize, N may be chosen sufficiently large so that the Fourier method is much more efficient than direct summation (as shown here).

To begin, the plane is divided into a grid of $N_x \times N_y$ cells. The potential at the α^{th} cell is then

$$\Phi_\alpha = \sum_{\beta}^{N_x N_y} G_{\alpha\beta} M_\beta$$

where M_β is the mass of the particles in the β^{th} cell and $G_{\alpha\beta}$ is the potential at the α^{th} cell due to the mass at the β^{th} cell. Then, clearly

$$G_{\alpha\beta} = -\frac{G}{\sqrt{\epsilon^2 + |\vec{x}_\alpha - \vec{x}_\beta|^2}}$$

where $\vec{x}_\alpha$ and $\vec{x}_\beta$ are the position vectors of the pertinent cells.

This periodic interaction potential G is written on the mesh for a unit source at the origin. The Fourier transform of this Green's function, $\hat{G}$ is found and stored. Each time the potential solver is called (it is called both in the main simulation loop and in the initial conditions code), a source distribution ρ is placed on a second mesh, M. In the main loop, ρ is found by smearing the mass using Cloud-In-Cell interpolation (CIC), while the initial conditions code uses an analytic density distribution function. In both cases, the Fourier transform $\hat{\rho}$ of ρ is found. By the convolution theorem, $\hat{\Phi} = \hat{G}\hat{\rho}$. This $\hat{\Phi}$ may in turn be easily inverse Fourier transformed to reveal the desired discretized potential. A simple finite differencing scheme, along with bilinear interpolation for off-grid particles, is then used to tabulate particle accelerations along x and y.

The above Green's function, while quite simple, is not ideal. A more appropriate choice of interaction potential allows the convolution method to find the potential for an isolated distribution of mass (one in which periodic boundary conditions do not contribute to the potential on the grid). By isolated, it is meant that the potential vanishes as $r \to \infty$ (i.e., $\phi \propto r^{-1}$).

The method for such an isolated system, as outlined by Hockney and Eastwood[3], calls for the use of only one-quarter of the available mesh points for the interaction potential. Any quarter will suffice, but ddr-galaxy has chosen the bottom left. Elsewhere, the density distribution is made to be zero. Using this

method, a correct r^{-1} potential is obtained within the used quarter of the grid, while it is non-physical elsewhere (which is of little consequence, as elsewhere the potential is never used). Additionally, the potential is that of an isolated system, i.e., no contribution is made to the potential due to non-physical boundary conditions (periodicity).

2.3 Integration Methods

The integrator used by ddr-galaxy is able to advance particles in a specified potential field Φ. This potential can be the result of a combination of a self-gravitating system and an analytic background potential. Several schemes can be employed, ranging from minimal computational cost and poor accuracy to high computational cost and high accuracy. Each particle has a specified position (x, y) and velocity (v_x, v_y), which the integrator uses to advance the particles. The acceleration of each particle is the normal Newtonian acceleration:

$$a_x = -\frac{\partial \Phi}{\partial x}$$

2.3.1 Euler

The Euler integration scheme, being first order, is the most basic of the integration methods employed by ddr-galaxy. It has a very low computational cost, but its error term is $O(\Delta t^2)$, where Δt is the stepsize[4]. The general form for the Euler method is

$$\begin{aligned} v_{x,n+1} &= v_{x,n} + \Delta t\ a_x(x_n, y_n, t) \\ x_{n+1} &= x_n + \Delta t\ v_{x,n} \end{aligned}$$

which advances a solution from t to $t + \Delta t$. Although cheap to compute, this method is not very desirable for practical purposes, due to its inaccuracy. All formula's shown are for the x-dimension only.

2.3.2 Leapfrog

The Leapfrog scheme is much more practical for a variety of reasons. Its computational cost is only slightly higher than that of the Euler method, but it is much more accurate. The Leapfrog formulas are

$$\begin{aligned} v_{x,n+\frac{1}{2}} &= v_{x,n-\frac{1}{2}} + \Delta t\ a_x(x_n, y_n, t) \\ x_{n+1} &= x_n + \Delta t\ v_{n+\frac{1}{2}} \end{aligned}$$

which has an error term of $O(\Delta t^3)$ because it is time-centered, i.e. it uses the derivatives at the midpoint of each step. To prepare our system for the Leapfrog scheme, one can see that the velocity must be moved half a timestep ahead of the position. Therefore a higher order half-step must be taken by the velocity, before anything else gets computed. This step must be of equal or higher accuracy than $O(\Delta t^3)$ so that the overall accuracy of the scheme is conserved. The extra step taken by the velocity is

$$v_{x,\frac{1}{2}} = v_0 + \frac{\Delta t}{2}\ a_x(x_0, y_0, 0) + \left(\frac{\Delta t}{2}\right)^2 \frac{\partial a_x}{\partial x}|_{(x_0,y_0,0)}$$

after which the standard Leapfrog formulae apply.

2.3.3 Midpoint

The Midpoint method is somewhat related to the Euler method, in that the midpoint method starts out by taking an Euler "trial step". Then it takes a real step using the value obtained from the trial step as follows:

$$\begin{aligned} dx1 &= \Delta t\ v_{x,n} \\ dv_x1 &= \Delta t\ a_x(x_n, y_n, t) \\ x_{n+1} &= x_n + \Delta t\ (v_{x,n} + \frac{dv_x1}{2}) \\ v_{x,n+1} &= v_{x,n} + \Delta t\ a_x(x_n + \frac{dx1}{2}, y_n + \frac{dy1}{2}, t + \frac{\Delta t}{2}) \end{aligned}$$

where $dy1$ can be obtained in exactly the same way as $dx1$, but using $v_{y,n}$ instead of $v_{x,n}$. This method has an error term of $O(\Delta t^3)$, like the Leapfrog method, but is somewhat more expensive to compute. However the Midpoint method is important because it is easily generalized into much more accurate methods, namely the Runge-Kutta and Cash-Karp schemes.

2.3.4 Runge-Kutta

One of the most often used integration schemes is the fourth order Runge-Kutta method, which is a

weighted average of four trial steps. Essentially, this method looks like an expanded Midpoint formula:

$$\begin{aligned}
dx1 &= \Delta t\, v_{x,n} \\
dv_x1 &= \Delta t\, a_x(x_n, y_n, t) \\
dx2 &= \Delta t\, (v_{x,n} + \frac{dv_x1}{2}) \\
dv_x2 &= \Delta t\, a_x(x_n + \frac{dx1}{2}, y_n + \frac{dy1}{2}, t + \frac{\Delta t}{2}) \\
dx3 &= \Delta t\, (v_{x,n} + \frac{dv_x2}{2}) \\
dv_x3 &= \Delta t\, a_x(x_n + \frac{dx2}{2}, y_n + \frac{dy2}{2}, t + \frac{\Delta t}{2}) \\
dx4 &= \Delta t\, (v_{x,n} + dv_x3) \\
dv_x4 &= \Delta t\, a_x(x_n + dx3, y_n + dy3, t + \Delta t) \\
x_{n+1} &= y_n + \frac{dx1}{6} + \frac{dx2}{3} + \frac{dx3}{3} + \frac{dx4}{6} \\
v_{x,n+1} &= v_{x,n} + \frac{dv_x1}{6} + \frac{dv_x2}{3} + \frac{dv_x3}{3} + \frac{dv_x4}{4}
\end{aligned}$$

This scheme has a high accuracy ($O(\Delta t^5)$), but is also relatively heavy on computational cost, because one computes four trial steps per particle, as opposed to Euler's one step per particle.

2.3.5 Cash-Karp

The Cash-Karp method is essentially the same as a fifth order Runge-Kutta method, with the various constants being those found by Cash and Karp (Numerical Recipes, 2nd ed.). To save space, we will give the general form of the formulas, with the exact values for the constants given in appendix A.

$$\begin{aligned}
dx1 &= \Delta t\, v_{x,n} \\
dv_x1 &= \Delta t\, a_x(x_n, y_n, t) \\
dx2 &= \Delta t\, (v_{x,n} + b_{21} dv_x1) \\
dv_x2 &= \Delta t\, a_x(x_n + b_{21} dx1, y_n + b_{21} dy1, t + a_2 \Delta t) \\
&\cdots \\
dx6 &= \Delta t\, (v_{x,n} + b_{61} dv_x1 + \cdots + b_{65} dv_x5) \\
dv_x6 &= \Delta t\, a_x(x_n + b_{61} dx1 + \cdots + b_{65} dx5, y_n \\
&\quad + b_{61} dy1 + \cdots + b_{65} dy5, t + a_6 \Delta t) \\
x_{n+1} &= y_n + c_1 dx1 + c_2 dx2 + c_3 dx3 + c_4 dx4 \\
&\quad + c_5 dx5 + c_6 dx6 \\
v_{x,n+1} &= v_{x,n} + c_1 dv_x1 + c_2 dv_x2 + c_3 dv_x3 + c_4 dv_x4 \\
&\quad + c_5 dv_x5 + c_6 dv_x6
\end{aligned}$$

This method is the most accurate one ddr-galaxy has to offer, its error being $O(\Delta t^6)$. With this accuracy comes a very high computational cost, which makes this method less practical than its accuracy suggests.

2.4 Integrator Comparison

To ensure that the integrators ran properly, a convergence test was run using several of the integrators. We placed 2000 bodies on a Keplerian background potential of the form

$$\Psi(r) = \frac{1}{r^2 + .01^2}$$

and ran this simulation over 10 units of time for various different timestep sizes. Figure 4 below shows the average energy loss per unit time as a function of timestep for several integrators.

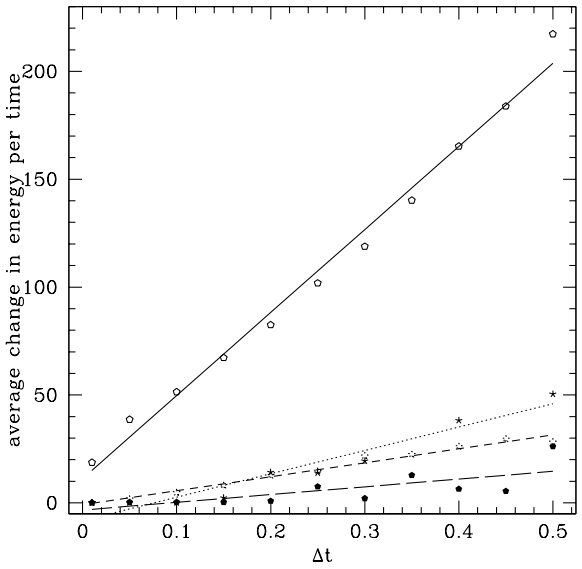

Fig. 4 Convergence test of several integration schemes. Plotted is the average change in energy per time vs. timestep size. From top to bottom- Euler, Leapfrog, Midpoint, Cash-Karp.

Euler diverges very rapidly as the timestep is increased, while the other three are much more efficient

at conserving energy. The error goes to zero for all integrators as $\Delta t \to 0$, indicating that every integrator converges.

2.5 Initial Conditions

In order to model a galaxy, one needs to set up the particles in such a way as to most resemble a configuration in a real disk galaxy. This means that particles need to be started on a circular orbit, with some comparatively small radial velocity distributions. The two main parts of ddr-galaxy —background potential and self-gravity— require distinct initial conditions setup.

2.5.1 Initial Conditions for Background Potentials

Every particle needs to be assigned a position in the galaxy. The initial conditions are set up in (r, θ) because the equations are simpler, and then they are converted into (x, y) using

$$\begin{aligned} x &= r\cos(\theta) \\ y &= r\sin(\theta) \end{aligned}$$

The initial angle (θ) is randomly determined between 0 and 2π, with equal probability for every angle. The initial r-coordinate is randomly determined according to an analytic radial surface density. In ddr-galaxy a Gaussian surface density of the form

$$\Sigma(r) = \frac{1}{2\pi\sigma^2} e^{\frac{-r^2}{2\sigma^2}}$$

is used, where σ is user determined.

The velocities in ddr-galaxy are determined as (v_r, v_θ) and then converted to (v_x, v_y) using

$$\begin{aligned} v_x &= v_r\cos(\theta) - r\ v_\theta \sin(\theta) \\ v_y &= v_r\sin(\theta) + r\ v_\theta \cos(\theta) \end{aligned}$$

To obtain v_θ such that the particle is placed on a circular orbit, one needs to solve the Lagrange equation for the generalized coordinate r. Lagrange's equation says

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{r}}\right] = \frac{\partial L}{\partial r}$$

For a Lagrangian of the form $L = \frac{m}{2}(\dot{r}^2 + r^2\dot{\theta}^2) - \Psi$ —and the added circular orbit condition that $\ddot{r} = 0$— this yields

$$v_\theta = \dot{\theta} = \sqrt{\frac{1}{r}\frac{\partial \Psi}{\partial r}}$$

Generally, a star is not on a perfectly circular orbit around the center of a galaxy. Therefore a small velocity —compared to v_θ— needs to be applied in the radial direction. A small v_r, distributed Gaussian-like, is therefore applied, leaving us with a perfectly functional background potential galaxy.

2.5.2 Initial Conditions for Self-Gravitating Systems

The initial conditions for a self-gravitating system are not as easy to set up, since stable configurations are rare. The (r, θ) coordinates are set up using a procedure analogous to that explained for background potentials, with one modification. Since ddr-galaxy can model the bulge at the center of a galaxy by a spherical background potential, it is unnecessary to have many particles close to the center of the system. Therefore a surface density of the form

$$\Sigma(r) = r^2 e^{-r}$$

is used for its property that it approaches 0 rapidly towards the center, and it exhibits exponential decay at larger radii.

For the circular velocities, the same principle holds as for background potentials. Circular orbits are still desired, but now the potential is composed of an analytic background potential and a discrete potential due to the bodies. Therefore, if the potential is defined as $\Psi = \Psi_{bg} + \Psi_{sg}$, where bg and sg stand for background and self-gravity, respectively, we can still use the relation

$$\dot{\theta} = \sqrt{\frac{1}{r}\frac{\partial \Psi}{\partial r}}$$

to determine v_θ. However, it has been shown[5] that a self-gravitating system is unstable unless a certain radial velocity dispersion is included. This radial velocity dispersion is of the form

$$\frac{1}{2\pi\sigma_r^2} e^{\frac{-r^2}{2\sigma_r^2}}$$

where σ_r is the minimal velocity dispersion required for stability. It is given by

$$\sigma_r = \frac{3.36\, G\Sigma(r)}{\kappa}$$

where κ is the frequency of small oscillations about the circular radius

$$\kappa = \sqrt{\frac{\partial^2 \Psi}{\partial r^2}|_{(r,\theta,t)} + 3\frac{L^2}{r^4}};$$

here, L is the angular momentum of the particle.

A model set up in this manner should prove to be stable, especially if appropriate damping has been introduced to cool down the system.

3 Sample Run

A sample run was conducted with 20,000 bodies self-gravitating in a superposition of Keplerian, Logarithmic, and a homogeneous sphere potential. The intent was to simulate the potential due to a large halo, centered upon a massive compact object. The homogeneous sphere had a uniform density of

$$\rho = \frac{3M}{8\pi(\frac{d}{2})^3}$$

where M is the mass of the disc (here unity) and d is the disc's diameter.

The initial conditions were chosen such that each particle began on circular orbits, with a small radial velocity dispersion– as outlined in section 2.5.2. The density distribution in r took the form,

$$\Sigma(r) = \frac{1}{12\pi} r^2 e^{-r},$$

again, as outlined previously.

The accompanying plots were produced at several time-slices throughout the simulation. As can be clearly seen, the particles begin with the specified density distribution and then quickly begin to form spiral arms as the simulation continues

4 Conclusions

Since galaxies evolve very slowly, it is hard to see any changes in a galaxy during one lifetime. This simulator is designed to be a useful tool for students seeking to learn more about galactic evolution, building galaxy models, or who just want to play with a working galaxy model. It is a model where many of the parameters are input by the user, letting them experiment with many different initial conditions and potential types. It allows students to actually see a model galaxy evolve with their own eyes, something that isn't possible without computer simulations.

We would like to thank Dr. Charles Gammie for supervising us during this project. We are also grateful to the NSF for funding the UIUC REU, and to Nicole Drummer for coordinating the program under which this research was conducted.

Appendix A - Cash-Karp Parameters

i	a_i	b_{ij}					c_i	c_i^*
1							$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		0	$\frac{277}{14446}$
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
$j =$		1	2	3	4	5		

References

[1] J. Binney and S. Tremaine, *Galactic Dynamics* , 1994

[2] L. Hernquist, The Astrophysical Journal, 356:359-364, 1990 June 20

[3] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles* , 1999

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C* , Second Edition, 1997

[5] A. Toomre, Ap.J. 139:1217, 1964