Comparison between Black Scholes And Monte Carlo Simulation method: Through use of different applications

Physics 210 Term Project Proposal

Hassan Bhatti

October 23, 2012

Introduction:

Black Scholes Model:

The BSM model is an example of a closed-form model, which is described as a “valuation model that uses an equation to produce an estimated fair value. The model assumes that the price of heavily traded assets follow a geometric Brownian motion with constant drift and volatility.

Monte Carlo Simulation:

Monte Carlo Option Price is a method often used in Mathematical nance to calculate the value of an option with multiple sources of uncertain-ties and random features, such as changing interest rates, stock prices or exchange rates, etc. This method is called Monte Carlo simulation, naming after the city of Monte Carlo, which is noted for its casinos.

Project Goals:

- Lay forward the fundamental concepts of the above techniques and how they differ.
- To discuss and apply some of the financial methods such as Monte Carlo simulation method and Black Scholes model.
- Explore the application of the techniques to stock volatility and to test the accuracy of both techniques to each other.
- That will be done through applying the concepts on different examples such as different financial options, employee stock option or an American/European call.

Black- Scholes Numerical Methods used:

The following formulae will be used in Black-Scholes calculation to find the accurate result with Black Scholes

Call price *c* or put price *p* are:

$$c = s\Phi(d_1) - xe^{-rt}\Phi(d_2)$$

$$p = xe^{-rt}\Phi(-d_2) - s\Phi(-d_1)$$

$$d_1 = \frac{log(s/x) + (r + \sigma^2/2)t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

Here, log denotes the natural logarithm, and:
s = the price of theunderlying stock
x = the strike price
r = the continuously compounded risk free interest rate
t = the time in years until the expiration of the option
σ = the implied volatility for the underlying stock
Φ = the standard normal cumulative distribution function.

Monte Carlo Simulation:

I will use simulation software either MATLAB or Mathematica to create a sequence of random data indicating the stock prices and the uncertainties. I will define a discrete model according to the applications I will try to apply the simulation to.

Results:

I expect to see that Black-Scholes fail in medium to long term conditions and Monte Carlo to be a better suited system instead of Black-Scholes. The reason being Black-Scholes assumptions cannot take into account the volatility in the market in long run.
I also expect to see that the Monte Carlo simulation to showcase better accuracy as I increase the size of sample.

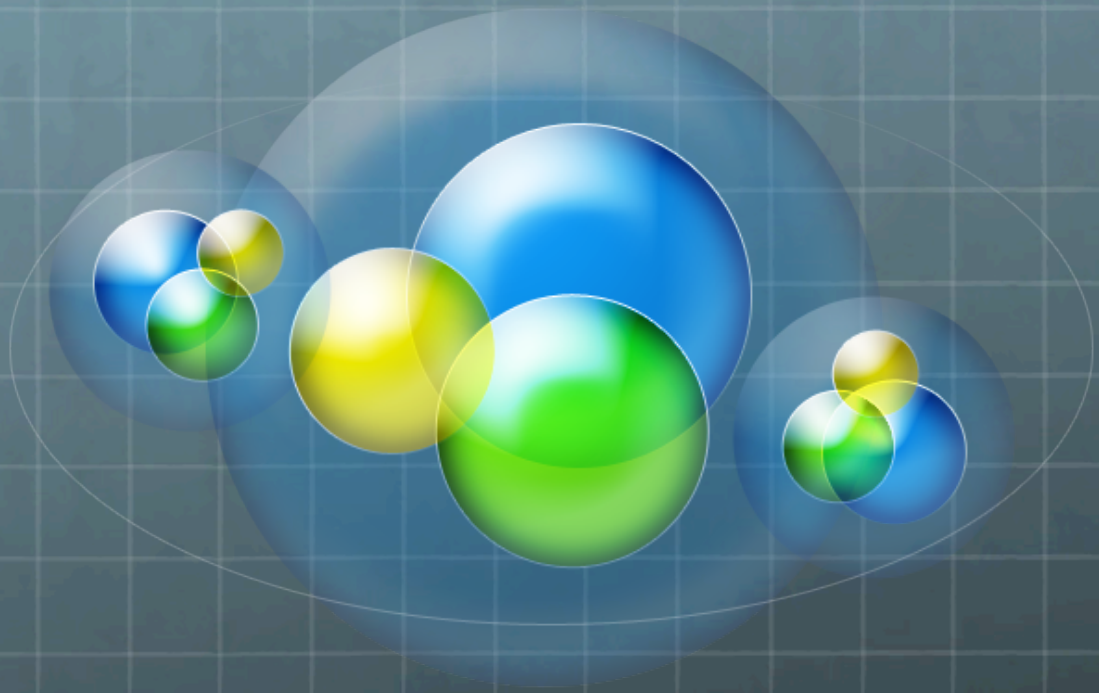
Project Timeline:

Dates	Activities
10/23- 11/02	Finalize the applications and numerical approaches
11/03- 11/15	Test the methods in matlab and numerically
11/16 – 11/23	Compare the results and complete analysis
11/24-11/31	Make presentation and report draft
12/01	Give presentation
12/02	Finish report

References:

- http://www.maxi-pedia.com/Black+Scholes+model
- http://users.aims.ac.za/~bundi/Thesis.pdf
- http://www.scribd.com/doc/88095838/Introduction-to-Stochastic-Calculus-Applied-to-Finance-Stochastic-Modeling
- http://www.cs.princeton.edu/courses/archive/fall09/cos323/papers/black_scholes73.pdf
- http://www.bbc.co.uk/news/magazine-17866646

THANK YOU!



N-body Simulation

Interacting Galaxies

Gelareh Farahi

October 2012

Overview

- Gravitational N-body simulations, that is numerical solutions of the equations of motions for N particles interacting gravitationally have applications from few body or solar system like systems all the way up to galactic and cosmological scales.

Project Goals

- To solve the n body problem using MATLAB and the related equations
- To simulate particle's motion in 2D or even 3D space
- To test the code and verify its validity by applying different initial values and Law of Conservation of Energy using FDA method
- To use a software like xfpp3d to visualize either colliding or interacting galaxies

Mathematical Formulation

- 1. If not electrically charged, the only force acting on particles is the gravitational force and the force resulting from the external potential field which can be calculated by Newton's Law of Gravity.

$$\sum_{i \neq j} G \frac{m_i m_j (\vec{r}_i - \vec{r}_j)}{\left|\vec{r}_i - \vec{r}_j\right|^3} - \vec{\nabla}.\phi_{ext}(\vec{r}_i - \vec{r}_j)$$

eq.1

- However, when the distance between two objects approaches 0, equation 1 presents a singularity.
- In order to avoid singularity, a softening length may be introduced. Therefore equation 1 changes into :

$$\sum_{i \neq j} G \frac{m_i m_j (\vec{r}_i - \vec{r}_j)}{((\vec{r}_i - \vec{r}_j)^2 + \varepsilon^2)^{3/2}}$$

eq.2

- where $\epsilon > 0$ is the softening length, that is a typical distance below which the gravitational interaction is suppressed.

- 2.N-body systems such as galaxies have a relaxation time much longer than the life of the Universe and are thus considered collisionless systems. Therefore, collisionless Boltzmann equation may apply :

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \vec{\upsilon}.\frac{\partial f}{\partial \vec{x}} - \frac{\partial \phi_T}{\partial \vec{x}}.\frac{\partial f}{\partial \vec{\upsilon}} = 0$$

eq.3

where f is a single particle distribution function, υ is the velocity of the particle.

Stars are more or less conserved as they orbit around a galaxy, and their velocities change in a continuous manner,dictated by accelaration due to gravity.

- 3.The total potential field is the sum of an external potential plus the self-consistent field defined from the distribution function itself through the solution of the Poisson equation :

$$\nabla^2\phi(\vec{x},t) = 4\pi G\rho(\vec{r},t)$$

$$\rho(\vec{r},t) = \int f(\vec{x},\vec{\upsilon},t)d^3\upsilon$$

eq.4

Numerical Approach And Testing

- There are two fundamental relations to check the accuracy of the solution :

1. The Law of Conservation of Energy : The total energy of the uncharged particles consists of the kinetic and potential energy :

$$E = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 - \sum_{i=1}^{N}\sum_{j\neq i}^{N} G\frac{m_i m_j}{\left|\vec{r}_i - \vec{r}_j\right|}$$

2. The Conservation of Total Angular Momentum :

$$J = \sum_{i=1}^{N} \vec{r}_i \times m_i \vec{v}_i$$

- One of the methods for solving the equations is the finite difference approximations discussed in lectures (FDA)
- Therefore, we suppose that each particle is a spherical object with mass m, and particular initial velocities in 3 dimensions, a particular position and check if the energy and momentum are conserved using the solutions to the problem

Project Timeline

Dates	
Oct 28-Nov 4	Continue research, Derive equations, Design code
Nov 4- Nov 11	Design code,Implement code
Nov 11 – Nov 15	Test code, run numerical experiments
Nov 15 – Nov 22	Analyze Data,Begin working on presentation and report
Nov 22 – Nov 26	Finish the presentation
Nov 27	Give presentation
Nov 30	Submit code and report

References

- Galactic dynamics / James Binney and Scott Tremaine, Princeton : Princeton University Press
- http://www.scholarpedia.org/article/N-body_simulations_%28gravitational%29#Astrophysical_domains_and_timescales
- Restless Universe: applications of garvitational n-body dynamics to planetary, steller and galactic systems: Blair Atholl 2000
- Gravitational N-body Simulations : Aarseth, Sverre 2003

Physics 210 – Introduction to Computational Physics

TERM PROJECT PROPOSAL PRESENTATION

NEURAL NETWORKS

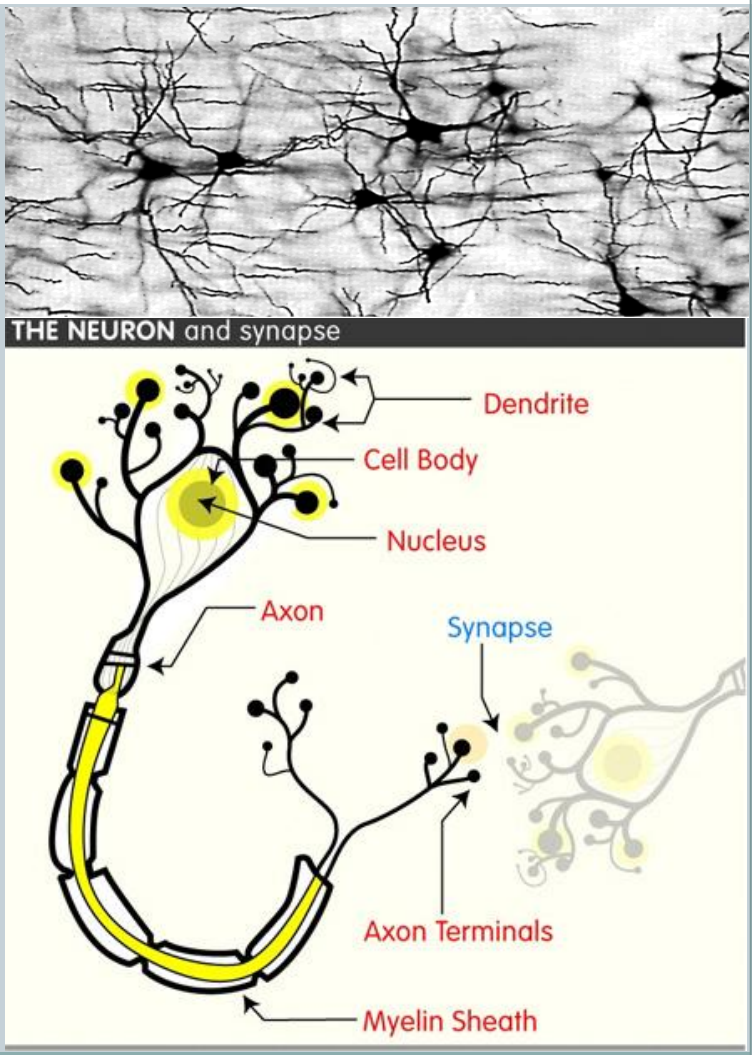
By: William Wu Huang

Theory

A neural network can be found in almost all multicellular organisms.

It is made up of two major cells; axons (presynaptic neuron) and dendrites (postsynaptic neuron). Its cell bodies are known as somas.

Every neuron starts at a stage known as the resting potential. This is roughly -70 mV due to the negatively charged proteins inside the cell and an excess amount of positively charged Sodium (Na^+) and Potassium (K^+) ions (since, by convention, we take the difference in potential between the interior and exterior of the cell).

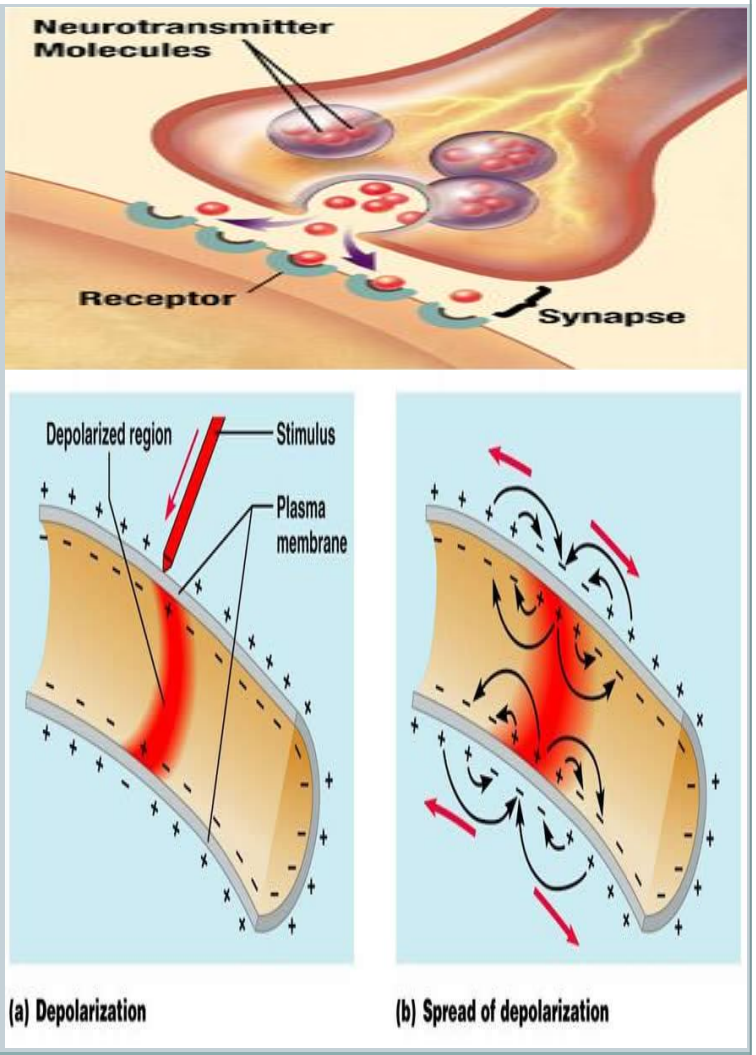


Theory

Stimuli send signals from the sensory receptors (in the peripheral nervous system) to the brain, brainstem, and the spinal cord to process the information given by the stimuli.

This is translated as a command to be relayed to the somatic nervous system (skeletal tissue) and the autonomic nervous system (smooth, cardiac, and adipose tissue and glands).

Each individual stimulus causes a graded potential; a change in the electrical potential by the opening of the Sodium gates to let Sodium ions in (depolarization) or the opening of the Potassium gates (hyperpolarisation).

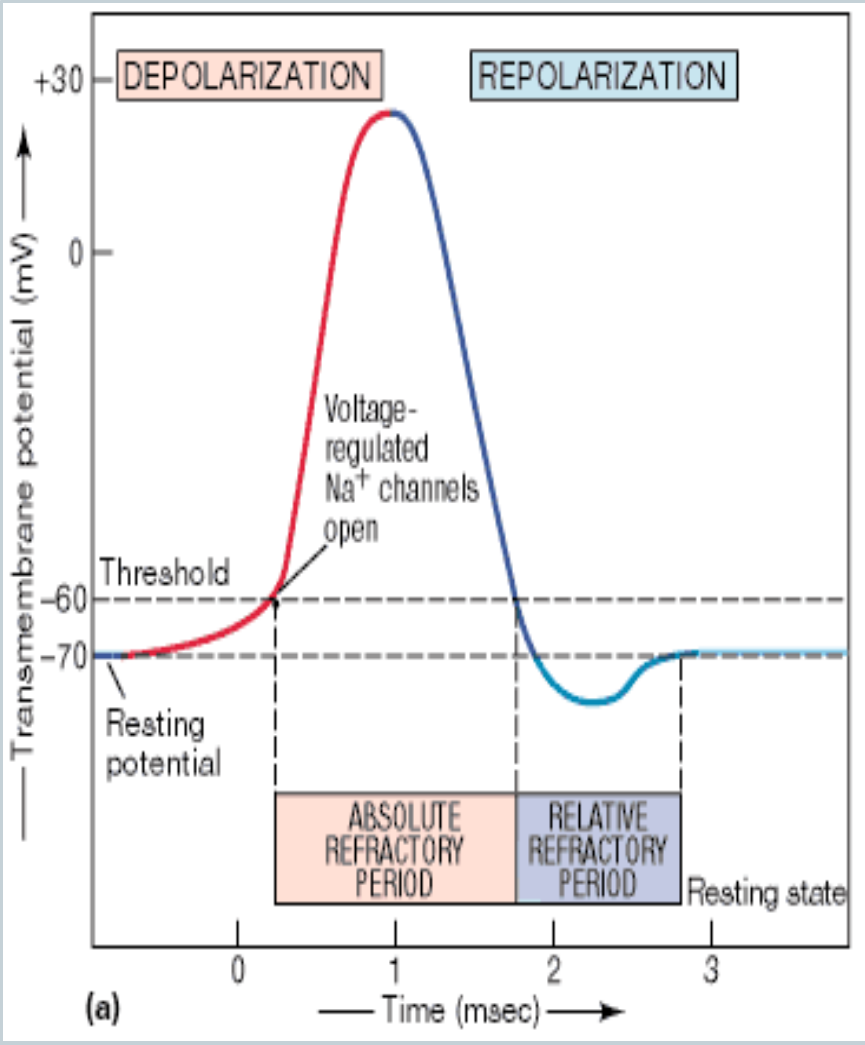


Theory

The principle of superposition applies to stimuli, which means that they can add up together in intensity. If the sum of the stimuli exceeds the critical threshold, then an action potential is generated on the dendrite and is sent down the axon.

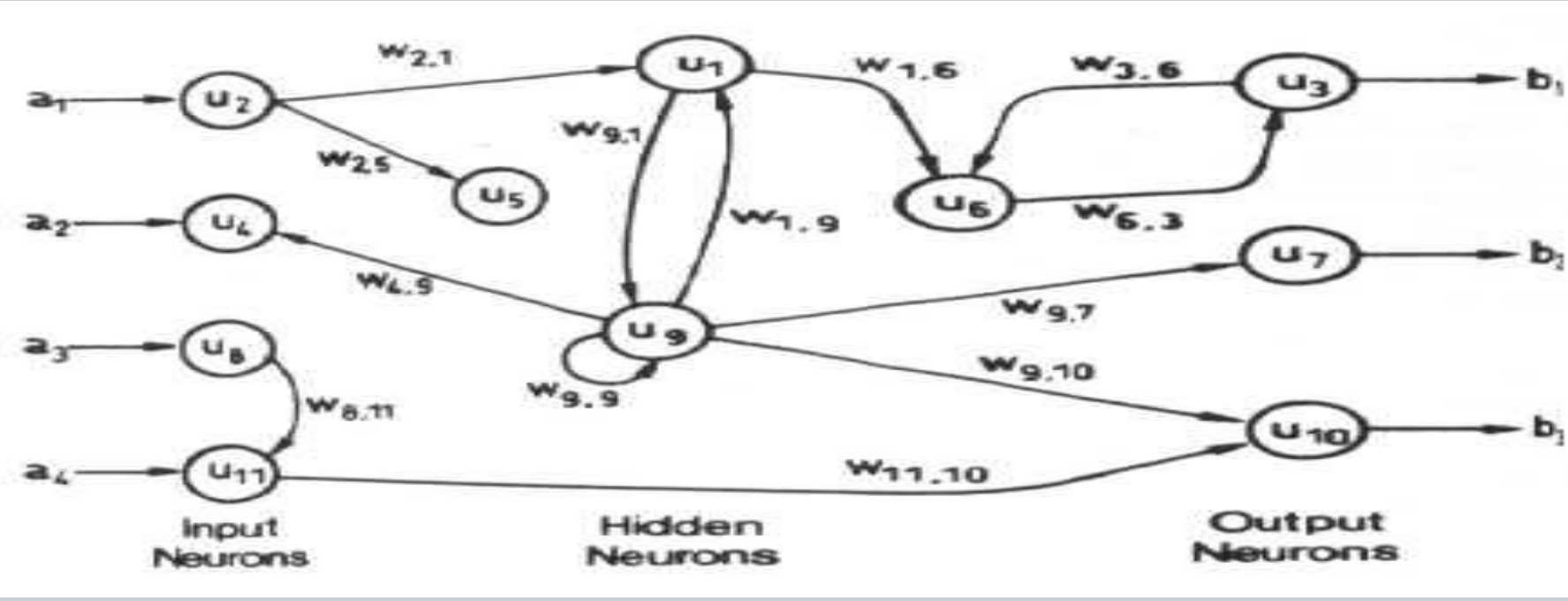
This is what causes an explicit command to be generated by the central nervous system to be delivered to the appropriate organs.

The action potential doesn't necessarily need to be generated in extreme situations, but occurs as an everyday event generated by ordinary stimuli.



Term Project Objectives

I will be using Octave to write a code that models the Ising energy function. A memory will be stimulated on a matrix grid of spins (neurons) to form a virtual neural network system. I will then test the limits of memory-based human recognition over different numbers of neurons and memories while using the Monté Carlo algorithm to ensure the minimum energy of this system overtime.



Mathematical Formulae

The total effective energy of a neural network:

$$\mathbf{E = - \Sigma_{i,j}\, J_{i,j} S_i S_j} \qquad \mathbf{(1)}$$

Here $J_{i,j}$ corresponds to the strengths or interactive energies of these synaptic connections with S_i and S_j representing the pairs of spins in this network.

The energy required to memorize one object or pattern:

$$\mathbf{J_{i,j} = S_i(m) S_j(m)} \qquad \mathbf{(2)}$$

Here m represents the given memory.

For many objects and patterns:

$$\mathbf{J_{i,j} = M^{-1} * \Sigma_m\, S_i(m) S_j(m)} \qquad \mathbf{(3)}$$

Here M represents the sum of m memories.

Mathematical Formulae

To add a new memory to the existing system, *ceteris paribus*:

$$\mathbf{J_{i,j}' = \beta J_{i,j} + \alpha S_i(p) S_j(p)} \qquad \mathbf{(4)}$$

Here $S_i(p)$ and $S_j(p)$ represent the new object or pattern, α determines the rate of learning this, β determines the rate of losing old memories for the original interaction energy, $J_{i,j}$, and $J_{i,j}'$ is the new interaction energy.

To add a new neuron, I'll simply use **(1)** since, for this project, the total energy and the total amount of neurons may be treated as synonymous.

The relationship between the number of neurons and the maximum number of patterns the system can memorize is experimentally estimated to be:

$$\mathbf{M \sim 0.13N} \qquad \mathbf{(5)}$$

Here M represents the maximum memory capacity and N is the number of neurons in the system in which M and N must be integers (where M must be rounded down). **(5)** tells us that in our total memory, roughly 87% of our information is redundant so only 13% is distinct.

Practical Approach

The key point here is that given a distorted figure, the system will use the Monté Carlo algorithm repeatedly until it arrives at the final state, which is what the figure is supposed to be. For example, given a mess , will the neural network be able to realize that this is just the letter A within a finite number of Monté Carlo iterations?

I will model the neural network in terms of Ising, using **(1)**. A pattern will be created using a 1000 x 1000 matrix grid and for this pattern to be recalled, the spin directions must change in such a way that they'll achieve their minimal state, energy-wise.

When $\Delta E_{flip} < 0$, this indicates that the spin can lower its energy by reversing its direction.

Also, I assume that the ambient temperature is 0°C and will use the Monté Carlo algorithm to ensure that the system always tends towards stability in its minimal energy state.

Experimental Design

Testing

I will establish a neural network on the grid with a complex pattern and then set a percentage of the interaction energies $J_{i,j}$ to zero which corresponds to missing or misrepresented components of this pattern. This will be repeated for different percentages over different patterns and different numbers of neurons.

Numerical Analysis

I will use gnuplot to see if **(5)** holds for different cases, using the same patterns for different numbers of neurons, and perhaps determine a better approximation. I will also examine how this relationship changes when I use patterns far more complex and henceforth different than an ordinary letter such as A.

Project Timeline

Time Period	Task
Oct. 22nd – Oct. 29th	Research, Formulation, and Code Design
Oct. 30th – Nov. 4th	Code Deployment
Nov. 5th – Nov. 11th	Code Testing
Nov. 12th – Nov. 18th	Experiment, Final Presentation, and Report
Nov. 19th – Nov. 25th	Numerical Analysis, Final Presentation, and Report
Nov. 26th – Nov. 28th	Final Presentation and Report
Nov. 29th	Launch Presentation
Nov. 30th	Polish and submit Report

Discussion

Thank you for listening. Do you have any questions, comments, and/or concerns?

References

- **Images**

- http://www.alanturing.net/turing_archive/pages/reference%20articles/connectionism/Turing's%20neural%20networks.html
- http://sparkcharts.sparknotes.com/psychology/psychology/section2.php
- http://secondfresh.com/?tag=brain
- http://dundeemedstudentnotes.wordpress.com/2012/04/06/graded-potentials/
- http://cwx.prenhall.com/bookbind/pubbooks/morris5/chapter2/custom1/deluxe-content.html
- http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

- **Information**

- *"Visual anatomy & physiology"* by Martini, Frederic H., and Ober, William C.
- *"Neural Networks and the Brain"*, Section 12.3, by authors unknown.

The End

The Gravitational Interaction of *N* Bodies

Phys 210 Term Project Proposal

By Reneil Pascua

Overview

- One can easily solve a 2-body gravitational interaction problem
  - but what about a 3-body? 4-body? 69 BODY?!

- Solution: Use a computer

Project Goals

- Write a MATLAB (octave) code which satisfies the following:
  - Can take various initial conditions and predict the motion of each body
  - Can be used to simulate particle motion in 2D

The Physics of it

Newton's universal law of gravitation says:

$$F = G\frac{m_1 m_2}{r2}$$

The Physics of it (cont.)

It can be turned into vector form:

$$\mathbf{F}_{12} = -G\frac{m_1 m_2}{\left|\mathbf{r}_{12}\right|^2}\,\hat{\mathbf{r}}_{12}$$

r_i is the position vector of the i-th particle
r_{12} is the distance between the two masses
$= r_2 - r_1$

r_{12} with the ^ is the unit vector in the direction of the distance between m1 and m2

In vector notation, $F_{12} = -F_{21}$

The Physics of it (cont.)

- The Force vector experienced by the i-th particle in the system is:

(r̲ is r with a ^ on top)

$$\mathbf{F}_i = Gm_i * \sum_{i=/=j} (m_j / |\mathbf{r}_{ij}|^2) * \underline{\mathbf{r}}_{ij}$$

  - This will be govern how each particle moves

Testing

- To debug the code, the program will be tested with simple situations (low *n* values such as 1,2,3…, same masses)

- After, it will be tested with arbitrary initial conditions including the number of particles and their masses

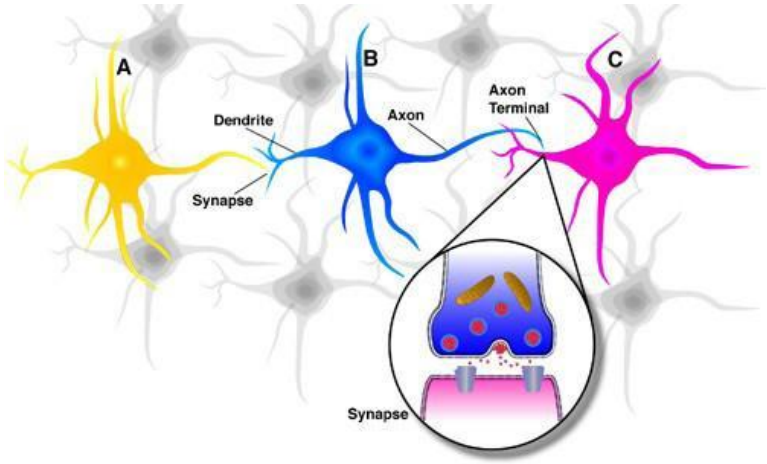
This is the plan:

Week #	What do?
1	Research
2	Design code
3	Write code
4	Debug code
5	Test with varying initial conditions, collect data and graphical simulations
6	Write report
7	Presentation → Finalize report and hand in

References

- http://en.wikipedia.org/wiki/Newton's_law_of_universal_gravitation


Neural Networks
Physics 210
Stephen Wellsby



OVERVIEW:

-The brain consists of approximately $\mathbf{10^{11}}$ neurons, each consisting of the soma (body), dendrites (input lines), axons (output lines), and synapses (connections).

-Each of our hundred billion neurons has on average 7000 connections to other neurons, with estimates of about $\mathbf{10^{14-15}}$ synapses total in the brain. (100 trillion to one quadrillion.)

-The electrical output from a neuron is directly related to the input to the neuron. This can be either excitory or inhibitory, i.e. sometimes a high input forces a high output, sometimes it forces a low output.

-This input/output relationship is very non-linear as can be seen in this diagram.

FIGURE 1:

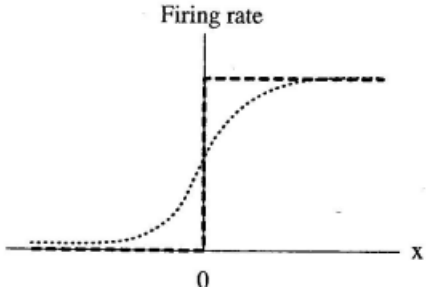
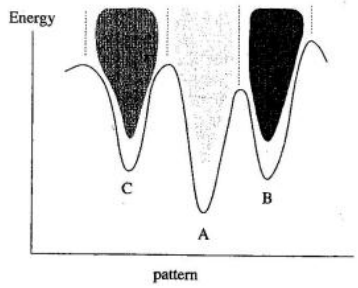


FIGURE 2:



-Because of this non-linear relationship, neurons are often modelled as on/off switches, i.e. firing or non-firing, with the output listed as $s = \pm 1$.

-Different connections have different strengths or weights to them.

-Essentially we move forward in time increments where at each increment and for each neuron, if the neuron has a critical input value then the neuron changes its output value.

-If the system works correctly, it should reach a local energy minimum and display the correct pattern.

PROJECT GOALS:

-Design a neural network simulation based on a 12 x 12 matrix of "neurons."

-Design an Octave or Matlab code which calculates the total energy of the system at those energy minimums.

-It will be able to calculate the strengths of the reactions (interaction energies).

-It will be able to "step" through time increments to reach one of the desired energy minimums corresponding to one of the patterns programmed into it.

-Finally it will test the failure rate of the system, that is, how many of the connections can be destroyed before the system stops functions correctly.

MATHEMATICAL FORMULATIONS:

Total energy of the system:

$$E = -\sum_{i,j} J_{i,j} * s_i * s_j$$

$J_{i,j}$ being the connection strength, s_i & s_j being the outputs (± 1).

Therefore, the sum of the synaptic inputs to neuron 'i' would be:

$$\sum_j J_{i,j} * s_j$$

Once the pattern has been created, the interaction energies are then determined by:

$$J_{i,j} = s_i(m) * s_j(m)$$

Or in the case of multiple patterns, by:

$$J_{i,j} = \frac{1}{M} \sum_m s_i(m) * s_j(m)$$

NUMERICAL APPROACH:

-Create a 12 x 12 matrix of neurons.

-Using a time step function, each of the ‘neurons’ will be given a chance to flip.

-Essentially if the change in energy for a neuron is negative, that means that it would reach a lower energy level and therefore the neuron would flip.

-The program will record each steps progress and compare it to the previous iteration. If no change occurs, the pattern is in the lowest energy state available and the iterations will cease.

-For each case, it will record the number of time steps required to reach the minimum.

TESTING AND NUMERICAL EXPERIMENTS:

-Investigate amount of randomness that can be inserted into the initial pattern for it to fall into one of the set patterns. I will initially try with just one pattern programmed in, and then with two and three patterns programmed in and compare the different scenarios. Perhaps also with four but will depend on time required/available.

-Investigate amount of error, or amount of ignored connections that my system can tolerate before it ceases to function correctly. Again, this will be tried with one, two and three (and perhaps four) different patterns programmed into the matrix.

PROJECT TIMELINE:

October 24-31	Research, derive equation, design code
November 1-5	Implement code
November 6-9	Test code
November 10-20	Run program determining failure points of system
November 21-25	Analyze Data, write report.
November 26	Submit presentation and final report
November 27	Class presentation

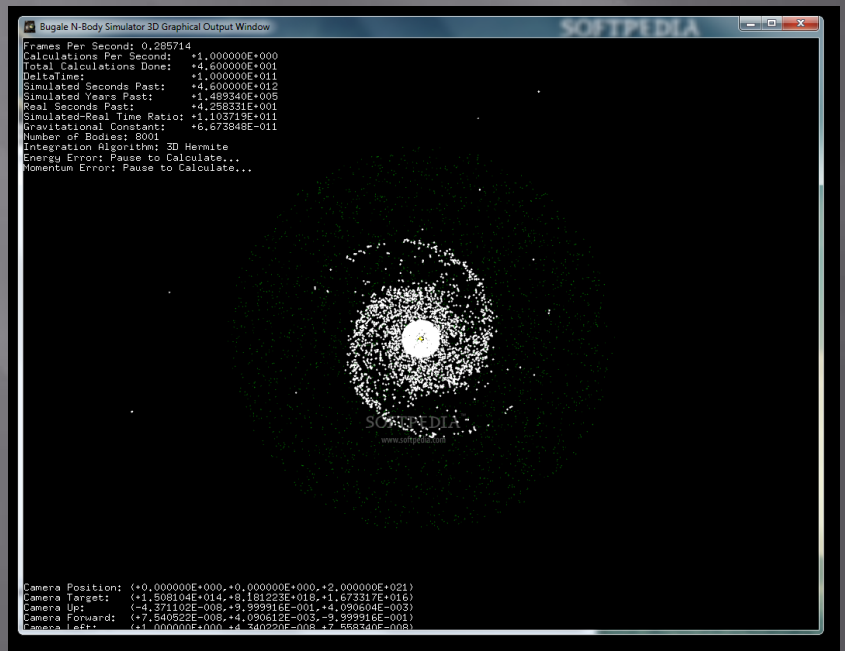
REFERENCES:

-N.J Giordano and H. Nakanishi, *Computational Physics, 2nd Edition,* Prentice Hall, West Lafayette, 2005.

-Drachman, D (2005). “Do we have brain to spare?” *Neurology* 64 (12): 2004-5

ELECTROSTATIC N-BODY

Tanner Braithwaite



Governing equations

- ((K*q1*q2)/R^2)*r^

Basic equation for electrostatic force on a partical, total force is the sum of all the particles in the simulation acting on a particle, in there respective direction

- P=F Δ T

An equation that will be used to determine the change in momentum of the particle in 3 dimensions

Equations continued

- T=t+△ x
- D=d+v*t

Using these equation I will update the time step over a certain interval of delta x and update the current position D using the last recorded distance, and the velocity *time of the partical after the new momentum is calculated

General format

- Given the initial parameters that the program starts with the system will update every delta x time interval
- The first thing that must be updated is the force on the particles this will then update the velocity of the particles and the velocity will update the position….and repeat
- Given N amount of particles this will show how they interact based on charge distance etc.

Things to watch out for

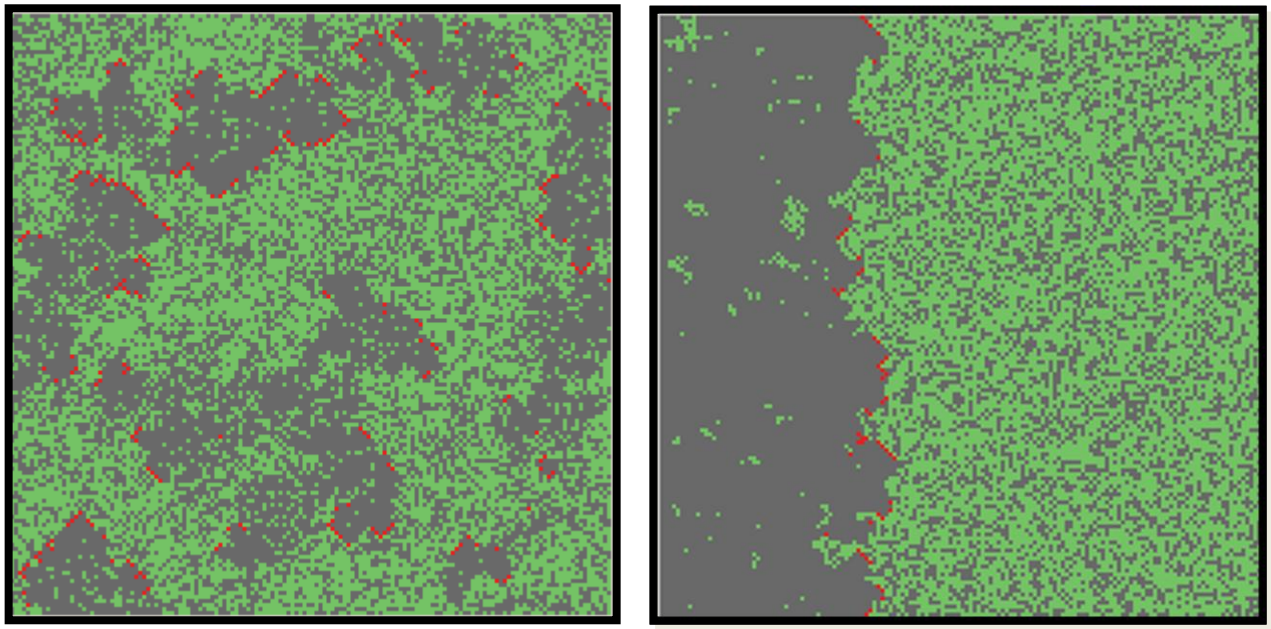
- Collapse/repulsion: no one wants to watch something collapse in on itself or distance itself so much nothing happens…boring
- Must work with any given amount of particles and particle spacing must be placed randomly
- I intend to try to make this simulation in 3D and try to show the effects from all angles

schedule

week1	week2	week3	week4	week 5
research	code/program	code/program	testing	finish write up
being coding	testing	testing	debugging	IT'S DONE
see examples			presentation	
			being write up	


NbBody : 10000
Eps : 0.40000
Dt : 0.03125
Time : 337

Simulating Forest Fires with Finite Cellular Automata



University of British Columbia

PHYS 210 Term Project Proposal

Eric Furugori -- October 21 2012

Overview

- The general forest-fire model is a finite cellular automaton who's cells have 4 possible states (some consider burned as empty):
  - Empty, Burnable, Burning, Burned
- The states of the cells update based on algorithms modelling the probabilistic behaviours of how fires spread based on initial probability conditions.

Project Goals

- Write a python script which performs the simulation with a graphical output, and that can perform statistical analysis of the results.
- Compare the statistical results to those of other simulations.
- Conduct a thorough investigation on the dependencies of the initial conditions, and of the evolution of the system over time.
- Investigate the problem in 1d, 2d, and 3d.
- Compare simulation with square grids and hexagonal grids, and with von Neumann and Moore neighbourhoods.
- Demonstrate self-organized criticality in all models.

Mathematical Formulation

- Start with an $N \times N$ dimensional lattice where each cell is defined to be a set of integer coordinates (i, j) such that:
  - $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, N$
- We then consider states of each cell at iteration n:
  - $S^n_{(\mathrm{i,\,j})} \in \{Empty, Burnable, Burning, Burned\}$
- Each cell has neighbours for each burnable $S^n_{(\mathrm{n,m})}$; $n, m \in N$ (for $Manhattan\ Distance = 1$):
  - Von Neumann neighbours: $N^n_{(i,\,j)} \subseteq \left\{S^n_{(\mathrm{i-1,\,j})}, S^n_{(\mathrm{i,\,j-1})}, S^n_{(\mathrm{i+1,\,j})}, S^n_{(\mathrm{i,\,j+1})}\right\}$
  - Moore neighbours: $M^n_{(i,j)} \subseteq N^n_{(i,\,j)} \cup \{S^n_{(\mathrm{i-1,\,j-1}),} S^n_{(\mathrm{i+1,\,j-1})}, S^n_{(\mathrm{i-1,\,j+1})}, S^n_{(\mathrm{i+1,\,j+1})}$
- We have a transition function that determines the next state of a cell:
  - $f: S^n_{(i,j)} \times S^n_{N(i,j)} \rightarrow S^{n+1}_{(i,j)}$
- The transition function is dependent on probability values:
  - P_D: probability that a cell will initially being burnable, and $1 - P_D$ will be empty.
  - P_B: probability of a burning cell becoming a burnt cell from state n to $n + 1$
  - P_I: probability that a burning cell will burn one of its neighbours
- Then we have the evolving lightning-strike model where burned trees are considered empty:
  - P_S: probability that an empty cell will ignite spontaneously
  - P_G: probability that an empty cell will grow a new tree
- This model has the same transition function.

Numerical Approach (The basic model)

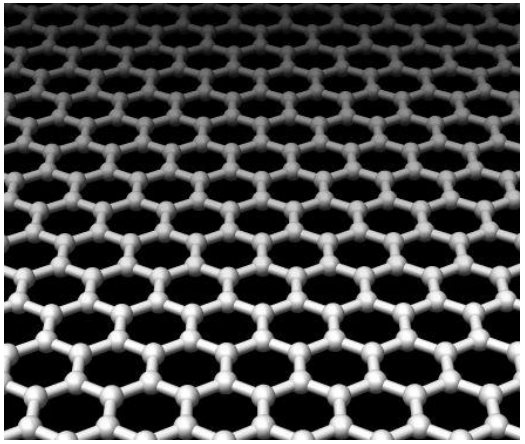
- The basic transition function $f: S^n_{(i,j)} \times S^n_{N(i,j)} \to S^{n+1}_{(i,j)}$ of the first model is a transformation defined by (where $rand \in \mathbb{R},\ 0 \leq rand \leq 1$):
  - If $S^n_{(i,j)} = Empty$ then $S^{n+1}_{(i,j)} = Empty$
  - If $S^n_{(i,j)} = Burnable$ then for each $S^n_{N(i,j)} = Burning$:
    - If $rand < P_I$ then $S^{n+1}_{(i,j)} = Burning$, else $S^{n+1}_{(i,j)} = Burnable$
  - If $S^n_{(i,j)} = Burning$ then:
    - If $rand < P_B$ then $S^{n+1}_{(i,j)} = Burned$, else $S^{n+1}_{(i,j)} = Burning$
  - If $S^n_{(i,j)} = Burned$ then $S^{n+1}_{(i,j)} = Burned$
- The second lightning-strike model evolves with the following maps:
  - If $S^n_{(i,j)} = Empty$ then:
    - If $rand < P_G$ then $S^{n+1}_{(i,j)} = Burnable$, else $S^{n+1}_{(i,j)} = Empty$
  - If $S^n_{(i,j)} = Burnable$ then for each $S^n_{N(i,j)} = Burning$:
    - If $rand < P_S$ then $S^{n+1}_{(i,j)} = Burning, else$:
      - If $rand < P_I$ then $S^{n+1}_{(i,j)} = Burning$, else $S^{n+1}_{(i,j)} = Burnable$
  - If $S^n_{(i,j)} = Burning$ then $S^{n+1}_{(i,j)} = Empty$

Numerical Approach (Real life simulation)

- We can extend our stochastic model to have more parameters. Let w parameterize weather conditions and g parameterize geological conditions. We can then write the transition function to be (idea from *A Stochastic Forest Fire Growth Model by Boychuk et al.*):
  - $f\big((i,j), N_{(i,j)}(t); w(t), g(t)\big) = \sum_{(k,l)\in N_{(i,j)}(t)} \lambda\Big((i,j),(k,l), S_{(k,l)}(t), w(t), g(t)\Big)$
  - $\lambda\Big((i,j),(k,l), S_{(k,l)}, w, g\Big) = f_0\Big((k,l)-(i,j), S_{(k,l)}, w_{(k,l)}, g\Big)$
- Note that I have replaced iteration n with t to denote that this is a function of time, hence at any f at time t we get the next state of the world over a discretized timestep Δt.
- λ denotes a rate, notably, a translation of f_0 centered at (i,j)
- The weather parameter w will carry information such as wind speed and direction, temperature, and fuel moisture content.
- The geographical parameter g with contain terrestrial information such as topology.

Numerical Approach (Further analysis)

- Next a hexagonal grid will be examined. This is best explained visually.
  - Behold the molecular structure of graphene (from Wikipedia Commons)
  - Each hexagonal cell now has 6 first order neighbours, 12 second order neighbours, and $6 \cdot (n\ orders)$ neighbours.
  - An ideal discretization to this problem are points with an infinitely large number of neighbours. This model is closer to that ideal than a square grid.



- The transformations described earlier still hold.

Visualization and Plotting Tools

- I will be using the `pygame` library to render the simulations in real time, and to add interactivity to the simulation.
- The most likely candidate for a plotting library would be `matplotlib`

Testing and Numerical Experiments

Testing:

- The time evolutions should approach a critical point as the system's attractor. We must test if the system truly reaches a stationary state:
  - $k = f(p+1)/p$ then using Taylor Series on k:
  - $P_D = \frac{z+1+k \pm \sqrt{(z-1)^2+2(z+1)k+k^2}}{2z} = \frac{1}{z} - \frac{1}{z^2-z}k + O(k^2)$
- Where z is a number (from 0 to z) of trees one burning tree can ignite
- As for a method to determine other properties… I'll have to do more research.

Testing and Numerical Experiments (Con't)

Numerical Experiments:

- In lightning-strike model, determine the significance of the ratio of P_G/P_S.
- In spread model, determine the significance of P_D.
- Determine the effects of a phase change.
- Compare square grids with hexagonal grids with the same parameters.

Project Timeline

Dates	Activities
10/22-10/24	Do research on self-organizing finite cellular automata and design code
10/25-10/28	Have a working implementation of both models for d=1 and d=2
10/29-11/04	Analyze statistical results that arise from these models
11/05-11/09	Design and implement hexagonal model from d=1 and d=2
11/10-11/18	Implement time-dependent model with weather and geographical algorithms
11/19-11/26	Analyze results, begin report
11/27-11/29	Present results
11/30	Complete and submit final draft of report

References

Boychuk, D. et al, *A Stochastic Forest Fire Growth Model* Springer Science+Business Media, LLC 2008

Christensen, K. and H. Flyvbjerg and Z. Olami, *Self-Organized Critical Forest-Fire Model: Mean-Field Theory and Simulation Results in 1 to 6 Dimensions*, Physical Review Letters, 71:17(2737), 25 Oct. 1993

Drossel, B. and F. Schwabl, *Self-Organized Critical Forest-Fire Model*, Physical Review Letters, 69:11(1629), 14 Sept. 1992

QUESTIONS?

COME AT ME

The N-Body Problem: A finite difference approximation

By Felipe Iriondo

Oct 22nd 2012

Brief Introduction and Summary

-What is the N-body problem?

-What would a solution look like?

Goals

- Write, test, and debug a working octave(MATLAB) code to solve the N-body problem.

-Use the code to simulate interactions between particles and display them through visual software

Equations of Motion

Make liberal use of:

Newton's second law:

$$\mathbf{F} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = m\mathbf{a},$$

Newton's law of gravitation:

$$\mathbf{F}_{12} = -G\frac{m_1 m_2}{|\mathbf{r}_{12}|^2}\hat{\mathbf{r}}_{12}$$

Particles in system have initial velocity(V0) and position (X0)

Approach to solving the problem

The problem will be solved by using finite difference approximations to calculate forces and acceleration based on the particle's previous position and velocity

$$\frac{\vec{F}}{m} = f'(\vec{v}_0) = \frac{f(\vec{v}_0 + \Delta t) - f(\vec{v}_0)}{\Delta t}$$

Testing

-Check the simulation's data for velocities and energy. They should agree to a certain margin of error for each time as they should always be conserved

Numerical experiments

- Attempt trials with different initial masses, velocities and positions.

- Attempt trials with symmetrical conditions and verify if solutions are symmetric

Timeline

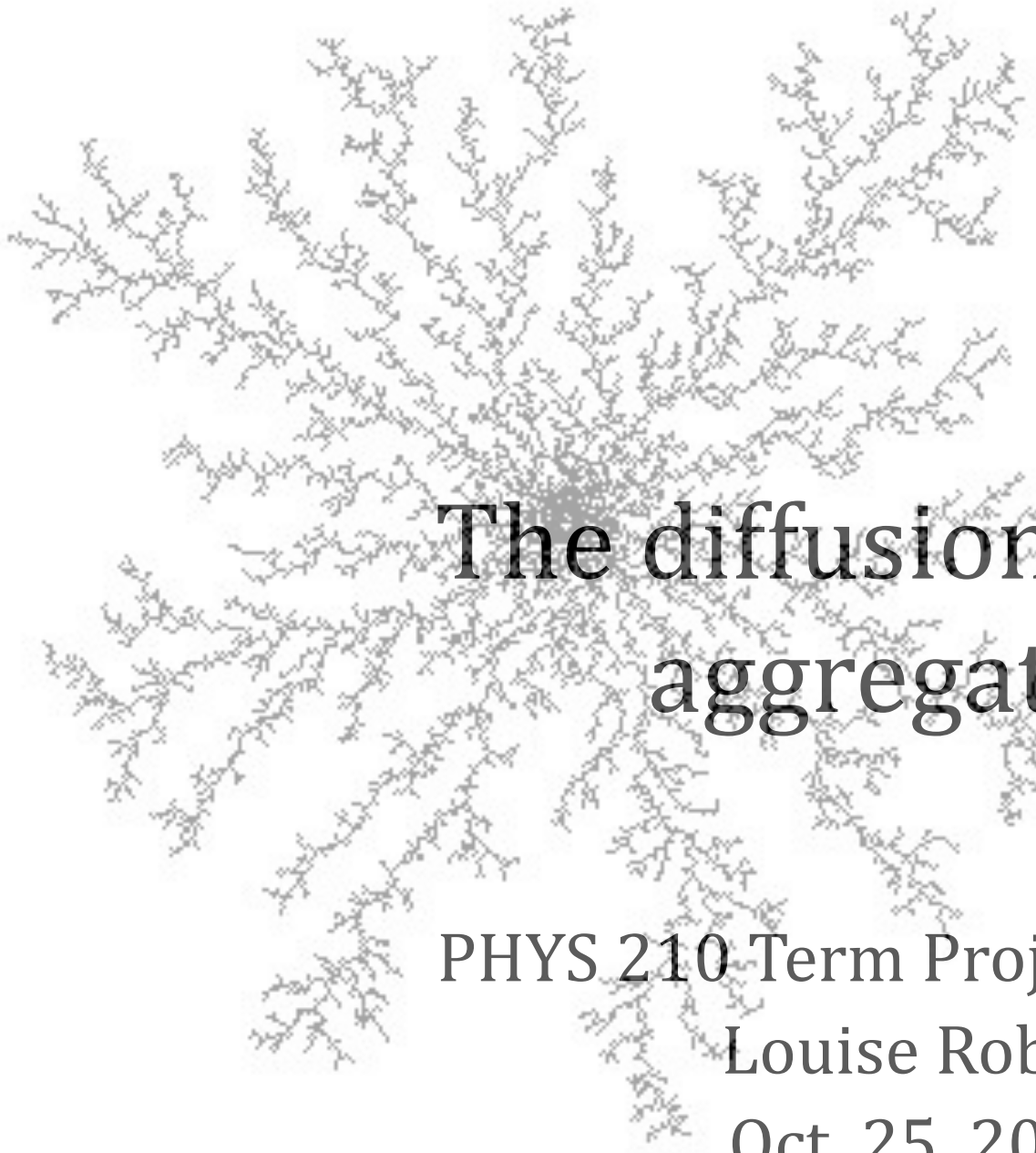
Day	Progress
Oct23-30th	Research relationships and understand problem parameters
Oct30th-Nov6th	Design and write code
Nov6th-13th	Test,adjust and debug code
Nov13th-20th	Begin experiments, Take a start at report and presentation
Nov20th-27th	Work on final copy of report
December 1st	Give oral presentation
December1st-3d	Scramble to finish report
December 4th	Submit finished copy of report

References

-Wikipedia

-http://www.amara.com/papers/nbody.html

-http://www.scholarpedia.org/article/N-body_simulations



The diffusion limited aggregation

PHYS 210 Term Project Proposal

Louise Roblin

Oct. 25, 2012

Introduction

- DLA: the process of irreversible and stochastic aggregation of small particles to form clusters
- A model of fractal growth, invented by T.A. Witten and L.M. Sander , in 1981
- An n-body gravitational particle simulation in 2D

The model

- We start with one immobile particle at the origin of a lattice.
- Another particle comes from far away and arrives randomly in one of the adjacent sites of the first particle, and is immobilized instantly.
- A third particles comes randomly in a site adjacent to the two first ones.
- Etc.

FIG. 1. (a) Aggregate of 3000 particles on a square lat-

Project goals

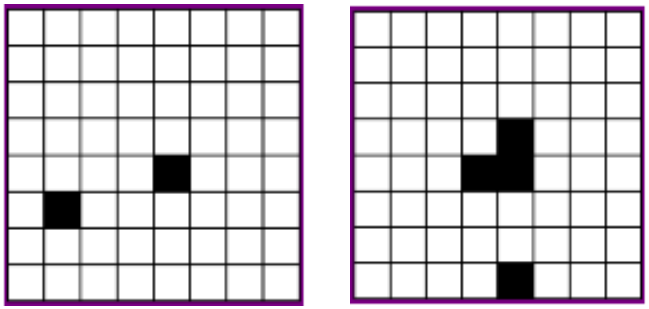
- Write a MATLAB (octave) code which calculates numerically the motions of gravitational particles
- Create a simulation of 2D diffusion-limited aggregations, with varying initial conditions

Mathematical approach

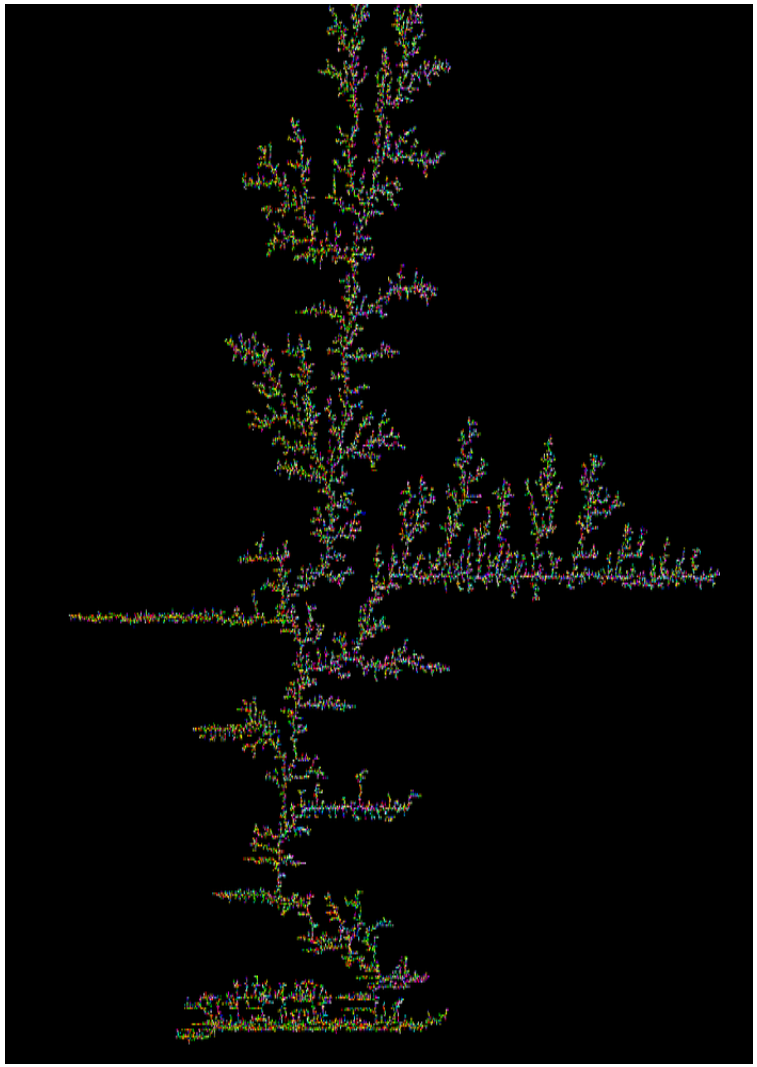
- Gauge field Ψ: $\Psi \leq 0$ inside, $\Psi \geq 0$ outside, $\Psi = 0$ on the surface
- Equation of motion: $\frac{\partial \Psi}{\partial t} + (\vec{V} \cdot \vec{\nabla})\Psi = 0$
- Diffusion: $\frac{\partial n}{\partial t} = D\nabla^2 n - q.$ ➔ $\frac{\partial n}{\partial t} = D\nabla^2 n - q - \lim_{\epsilon \to 0^+} \Phi(\vec{r}, t)\delta(\Psi - \epsilon)|\vec{\nabla}\Psi|$
- $$\frac{\partial \Psi}{\partial t} - \lim_{\epsilon \to 0^+} D \int dt' d\vec{r}' \vec{\nabla} G(\vec{r} - \vec{r}', t - t') \cdot \vec{\nabla}\Psi(\vec{r}, t) \frac{\partial \Psi}{\partial t'}(\vec{r}', t')\delta[\Psi(\vec{r}', t') - \epsilon] = -\frac{1}{n_0} D\vec{\nabla} n^s \cdot \vec{\nabla}\Psi$$
- The solution of this equation gives the description of surface growth.

Numerical approach

-Simulation on a lattice



Visualization and plotting tools in Matlab



Testing and numerical experiments

- **Testing**
  - Start with a simple 2 or 3 particles interaction, to make sure they merge properly
- **Numerical experiments**
  - Investigating a variety of initial positions, which would result in varying the aggregate formations. These different configurations will be analyzed in addition to any other phenomena that are observed during testing.

Timeline

Week	Activities
Week 1	Basic research + design code
Week 2	Implement code
Week 3	Test code
Week 4	Numerical experiments
Week 5	Begin report
Week 6	Final draft of report
Week 7	Presentation + submit report

Source: « Diffusion-limited aggregation », T.A. Witten & L.M. Sander, in *Physical review B*, volume 27, no. 9

Simulation of N gravitationally interacting objects in 3D

Physics 210 Term Project Proposal

Yuan Yao (Andrew)

Overview

- A group of N objects interact with one another under the influence of gravitational forces.
- This problem has been motivated by the need to understand the motion of the Sun, planets and stars in the universe.
- Its first complete mathematical formulation appeared in Newton's *Principia*.

Project Goals

- Approach the N-body problem numerically: write a MATLAB/octave code to predict the future condition of objects based on the given initial condition in 3D
- Investigate the behavior of the system by applying various initial conditions, including special symmetrical conditions
- Verify the code using energy conservation

Mathematical Formulation

1. Newton's Law of Universal Gravitation:

$$\mathbf{F}_{12} = -G\frac{m_1 m_2}{|\mathbf{r}_{12}|^2}\hat{\mathbf{r}}_{12} \quad \text{where } G = 6.67 \times 10^{-11}\ \mathrm{N \cdot m^2 / kg^2}$$

2. Superposition Principle

3. Kinematics:

$$\vec{v}_i = \Delta t \cdot \vec{a}_i + \vec{v}_{\circ_i}$$

$$\vec{r}_i = \Delta t \cdot \vec{v}_i + \vec{r}_{\circ_i}$$

Component-wise…..

$$F_x = \frac{GMm\Delta x}{r^3}$$

$$F_y = \frac{GMm\Delta y}{r^3}$$

$$F_z = \frac{GMm\Delta z}{r^3}$$

where $r = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$

Also,

$$\begin{cases} v_{xf} = \Delta t \cdot a_{xi} + v_{xi} \\ v_{yf} = \Delta t \cdot a_{yi} + v_{yi} \\ v_{zf} = \Delta t \cdot a_{zi} + v_{zi} \end{cases} \qquad \begin{cases} r_{xf} = \Delta t \cdot v_{xi} + r_{xi} \\ r_{yf} = \Delta t \cdot v_{yi} + r_{yi} \\ r_{zf} = \Delta t \cdot v_{zi} + r_{zi} \end{cases}$$

Therefore, given initial velocity + position of particles as the initial condition , we can predict their motion by repeatedly using the above formulas.

We can also increase the accuracy of the simulation by decreasing the time step Δt .

Testing and Experiments

- Based on energy considerations, we can test the simulation by calculating the total kinetic energy of the particles in each step. (It should be constant!)
- In addition, test/experiment using special cases, including symmetrical cases, 2D cases, and 2-body systems.

Timeline

Dates	Activities
10.23-10.28	Research Mathematical formulation, design codes
10.29-11.04	Design, implement codes
11.05-11.11	Test codes, begin numerical experiments
11.12-11.18	Continue with experiments; begin presentation & report
11.19-11.25	Polish up presentation and report
11.27	Project Presentation~
11.30	Submit project!

References

- http://en.wikipedia.org/wiki/N-body_problem
- http://en.wikipedia.org/wiki/Superposition_principle
- http://www.physicsclassroom.com/class/1dkin/u1l6a.cfm

Questions? Comments?

TOOMRE'S MODEL OF GALACTIC MERGERS

PHYS 210: Computational Physics
Term Project Proposal

Alden Campbell

October 23, 2012

OVERVIEW

° Alar Toomre was an Estonian born mathematician who devised an instability criterion for differentially rotating disks. This is now known as the Toomre's stability criterion.

°The Toomre's Model is a model that shows the prediction of interactions between two galaxies when they are close to each other.

°Working with his brother, Juri, they were able to predict galaxy collisions using only Newton's law of gravity. The results they got very closely matched observable galaxy collisions. An good example of this is that of the "Antenna Galaxies".

°The beauty of Toomre's model is that we only need to consider the gravitational effect of the more massive objects of the systems. This means that we only need to consider the effect of the galactic cores (since Mgc >>> Ms) and we can treat the less massive objects (stars) as tracers so that we can see the overall effects.

PROJECT GOALS

- Write an OCTAVE/MATLAB code to simulate Toomre's Model for galactic collision interactions
- Depending on the initial conditions of each galaxy the end result of the collision will be different so I would like to investigate the different results by starting with different initial conditions
- Ultimately, I would like to simulate a galaxy collision that we have observed in the sky to test the model more thoroughly

MATHEMATICAL FORMULATION

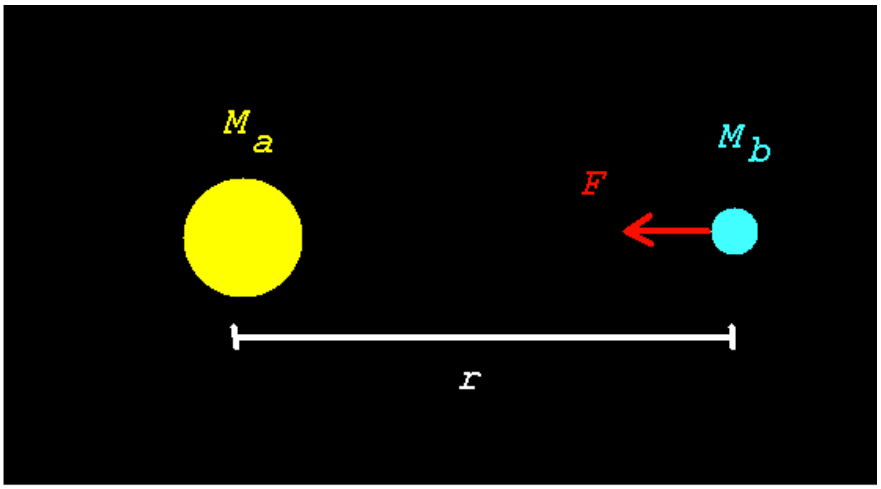
‣ Staying true to Toomre's model the only required equation to govern the simulation is simply Newton's law of gravitation as given by:

$$F = \frac{G \times m_1 \times m_2}{r^2}$$

‣ Since we also know that F=m x a we can rearrange Newton's law of gravitation as follows:

$$a = \frac{G \times m_1}{r^2}$$

where m1 is the mass of the galactic center and m2 is the mass of a star and using the fact that m1>>m2.



$$F = G \frac{M_a M_b}{r^2}$$

NUMERICAL APPROACH

- Since the Barnes-Hut algorithm works well to simulate N-body problems I will attempt to use this algorithm

- As a back up I will use a finite difference approach to simulate the N-body interactions

- If it is not too complex I will attempt to make this simulation three dimensional so as to better visualize what is happening with the interacting galaxies. Although, this may be too large an undertaking to successfully complete the project in the time frame I have

VISUALIZATION

•At this point I am assuming that the built in MATLAB visualization software is capable of producing the desired simulation. If it isn't then I will track down some better visualization software. Hopefully I won't have to do this.

TESTING AND NUMERICAL EXPERIMENTS

‣Once completed I will compare the results of my simulation with accurate galaxy collision simulations (available to view online mostly)

‣Compare the results of a variety of different initial conditions for the two galaxies. Initial conditions include galaxy core mass and radius and possibly their initial radial approach speeds as well

‣If there is time I will also investigate what happens to a globular cluster when placed an arbitrary distance from the galaxy core and see if this distance increases or decreases the globular cluster evaporation time

‣To check for consistency I will do multiple tests of the same initial conditions in order to verify that what I am simulating actually holds true to what we observe in the night sky

‣I will keep the number of stars in each galaxy relatively low so that our computers can actually run the simulations

‣For simplicity's sake I will not consider any 'particle' collisions and assume that all objects pass 'through' each other (which is actually what happens!)

PROJECT TIMELINE

DATE	GOAL
Oct. 29	Basic Research, Equations, Begin Coding
Oct. 30	Basic Research, Equations, Begin Coding
Oct. 31	Work on code
Nov. 1-3	Work on code
Nov. 5	Implement code
Nov. 10	Test Code
Nov. 15	Have a working simulation/ begin writing report
Nov. 20	Data Analysis/ work out any kinks in presentation
Nov. 27	Presentation / have a draft of report
Nov. 30	Submit final report hard copy

REFERENCES

http://en.wikipedia.org/wiki/Barnes–Hut_simulation
http://en.wikipedia.org/wiki/Alar_Toomre
http://faculty.etsu.edu/smithbj/collisions/collisions.html
http://cas.sdss.org/dr3/en/proj/basic/galaxies/collisions.asp

QUESTIONS???


Simulation of N-Body Gravitational Interactions

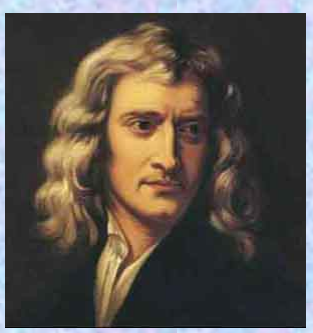
Physics 210 Proposal
Shayan Gheidi
October, 2012
Professor Choptuik


Overview

- This project attempts to create a simulation which illustrates the various interactions between “N-Bodies” bound by Newton’s Second Law.

Project Goal

- To write Matlab (octave) code that simulates motion and gravitational interaction between “N-Bodies”
- Simulates the above under various initial conditions



Physical Representation

- The world simulated will be governed by this formula:
  - **F=(G*m1*m2)/r^2**
  - F=ma → a= (G*m1*m2/r^2)/m

Physical Representation, n=2

- M1 and M2 represent the masses of objects 1 and 2
- G represents Newton's Constant, 6.67384*10^-11m^3/kgs^2
- r, represents the distance between the two bodies.
- The Force is a vector.

Physical Representations, n>2

- Fnet= **ΣF(g)**

- $$U = \sum_{i=1}^{N} \frac{GmM_i}{|\mathbf{r} - \mathbf{R_i}|}$$ For n particles

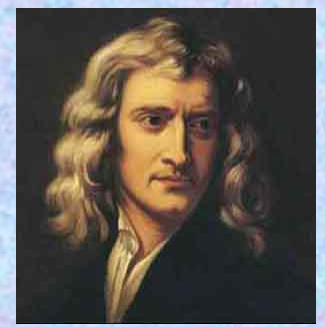
- a1= **ΣF/m**

Numerical Approach

- Finite Difference Approximation

Testing

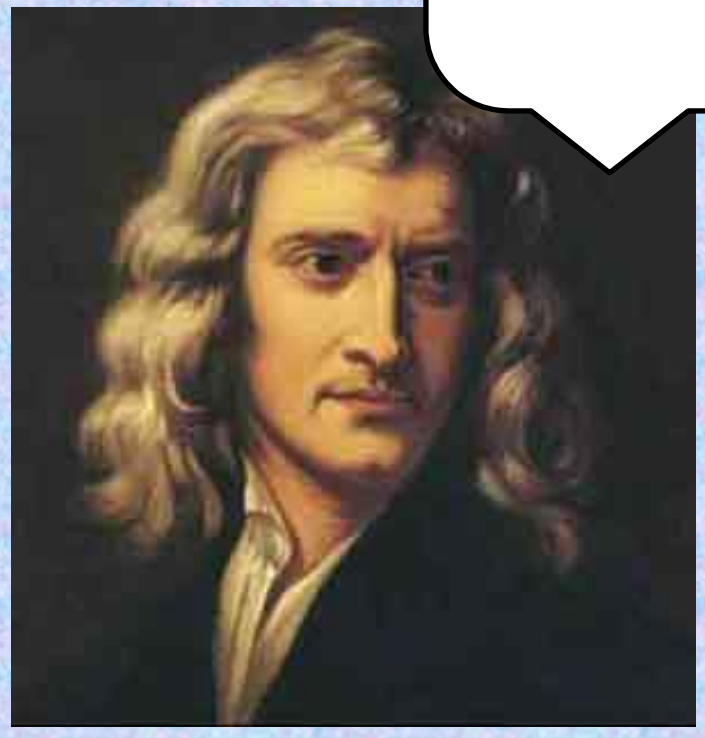
- The project will first be tested with a low n in order to debug and observe general behaviour
- Will then be tested with different values of n and different initial conditions
- Test different sizes of particles and their behaviour.



Timeline

Date	Plan
Week1, Oct22	General Research, gather information
Week2, Oct29	Begin writing code
Week3, Nov5	Still writing code
Week4, Nov12	Finish coding, test and debug
Week5, Nov19	Begin analysing data, test different cases and different initial conditions
Week6,Nov26	Finish writing report, analysis, further studies
??	Presentation
??	Hand in Report

Thank you!



Forest Fire Simulation

with

Cellular Automaton

PHYS 210 Term Project Proposal

By Dakota Jang

October 22, 2012

Overview

- Cellular automata is a model that involves group of cells changing/evolving by interacting with neighboring cells
- Forest fire spreading depends on the intensity of the fire on the adjacent burning tree
- Assuming that wind can affect the intensity of the flame, I will add a some randomness to the intensity (might ignore for simplicity)

Project Goals

- Write a Matlab(octave) or Java code that simulates a realistic forest fire simulation
- Examine how the intensity of the flame effect the fire spread in terms of
  - The spreading rate
  - The area of burnt cells over certain time

Mathematical Formula

- **Frandsen's (1971) equation for Quasi-steady rate of spread**

$$R = \frac{I_{xig} + \int_{-\infty}^{0} \left(\frac{\partial Zc}{\partial Z} \right)_{z_c} dx}{\rho_b \varepsilon Q_{ig}}$$

- **The Rothermel (1972) spread equation**

$$R = \frac{I_R \xi \left(1 + \phi_w + \phi_S\right)}{\rho_b \varepsilon Q_{ig}}$$

$$I_R = \Gamma' w_n h n_M n_s$$

Numerical Approach

- This simulation will be working in certain space defined in a form of a matrix and each cell would be affected with certain function with respect to the condition of the neighboring cell(s)

$$S_{i,j}^{t+1} = F(S_{i-1,j-1}^{t}, S_{i-1,j}^{t}, S_{i-1,j+1}^{t} S_{i,j-1}^{t} S_{i,j}^{t} S_{i,j+1}^{t} S_{i+1,j-1}^{t} S_{i+1,j}^{t} S_{i+1,j+1}^{t})$$

- The F in the above function will involve
  - how fast the cell will be burn as time goes
  - change in intensity of the flame
  - when the neighboring cell will ignite

$$S_{i,j}^{t} = \frac{A_{burnt}}{A_{total}} \propto I_R$$

Testing & Numerical Experiments

- Simulate a forest fire on a 2 dimensional space
- All initial condition of the cell will be homogeneous (except the initial burning cell)
- Create a initial forest terrain (with/without obstacles)
- Create a initial burning point where the fire will start to spread
- Observe the time and the area of the forest fire simulation

Project Timeline

Dates	Activities
10/22 – 10/28	Do basic research, derive equations & design code
10/29 – 11/04	Implement code
11/05 – 11/11	Test code
11/12 – 11/18	Run numerical experiments, begin presentation & report
11/19 – 11/25	Analyze data, continue work on presentation & report
11/25 – 11/29	Polish presentation and work on final draft of report
11/30	Give presentation!
11/30-11/04	Finish final draft of report
12/04 (11:59 PM)	Submit report!

References

- N. G. Sugihara, Fire in California's Ecosystems, University of California Press, U.S. (2006)
- R. C. Rothermel, A Mathematical model for predicting fire spread in wildland fuels, U.S. Department of Agriculture (1972)
  - http://www.fs.fed.us/rm/pubs_int/int_rp115.pdf
  - http://www.fs.fed.us/rm/pubs_int/int_rn292.pdf
- I. Karafyllidis & A. Thanailakis, *A model for predicting forest fire spreading using cellular automatai,* Democritus University of Thrace, Greece, (1997)

SIMULATION OF *N* GRAVITATING BODIES

Taylor Rowe

Overview

- Differences between $N \leq 2$ and $N > 2$ bodies
- Numerical method
- Computation
- Simulation
- Error checking

Why N≥3?

- In 2 bodies there exists an analytic solution

- For N≥3 no analytic solution exists
  - We must approximate the motion numerically
- 4^{th} Order Runge Kutta Method in MATLAB
  - Numerical method
  - Efficient and stable until large time steps

4th Order Runge-Kutta Method

- In general this method applies to IVP's of form

$$y' = f(t,y) \qquad y(t_0) = y_0$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + \frac{h}{2}, y_n + k_1\frac{h}{2})$$
$$k_3 = f(t_n + \frac{h}{2}, y_n + k_2\frac{h}{2})$$
$$k_4 = f(t_n + \frac{h}{2}, y_n + k_3 h)$$

- We are interested in Newtonian particles

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \vec{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \quad \vec{a} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} \longrightarrow \quad \dot{\vec{r}} = \vec{v} \qquad \dot{\vec{v}} = \vec{a}$$

- We can apply our method to both of these eqns

$$\dot{\vec{r}} = \vec{v}$$

- Write out *k's* in terms of known variables and solve

$$\vec{k}_{1_{v_{i+1}}} = \vec{a}(\vec{r}_i)$$

$$\vec{k}_{2_{v_{i+1}}} = \vec{a}(\vec{r}_i + \frac{h}{2}\vec{k}_{1_{r_{i+1}}})$$

$$\vec{k}_{3_{v_{i+1}}} = \vec{a}(\vec{r}_i + \frac{h}{2}\vec{k}_{2_{r_{i+1}}})$$

$$\vec{k}_{4_{v_{i+1}}} = \vec{a}(\vec{r}_i + h\vec{k}_{3_{r_{i+1}}})$$

$$\vec{v}_{i+1} = \vec{v}_i + \frac{h}{6}(\vec{k}_{1_{v_{i+1}}} + 2\vec{k}_{2_{v_{i+1}}} + 2\vec{k}_{3_{v_{i+1}}} + \vec{k}_{4_{v_{i+1}}})$$

Recursion for *v* based on initial v_i

$$\dot{\vec{v}} = \vec{a}$$

$$\vec{k}_{1_{r_{i+1}}} = \vec{v}_i$$

$$\vec{k}_{2_{r_{i+1}}} = \vec{v}_i(\frac{h}{2}\vec{k}_{1_{v_i+1}})$$

$$\vec{k}_{3_{r_{i+1}}} = \vec{v}_i(\frac{h}{2}\vec{k}_{2_{v_i+1}})$$

$$\vec{k}_{4_{r_{i+1}}} = \vec{v}_i(h\vec{k}_{3_{v_i+1}})$$

$$\vec{r}_{i+1} = \vec{r}_i + \frac{h}{6}(\vec{k}_{1_{r_{i+1}}} + 2\vec{k}_{2_{r_{i+1}}} + 2\vec{k}_{3_{r_{i+1}}} + \vec{k}_{4_{r_{i+1}}})$$

Recursion for *r* based on initial r_i

Simulation

- Using the Runge Kutta method we can generate two matrices:

X : nx3 matrix

Body 1	$r1_1$	$r1_2$	$r1_3$	…
Body 2	$r2_1$	$r2_2$	$r2_3$	…
Body 3	$r3_1$	$r3_2$	$r3_3$	…

T : nx1 vector

Time	t_1	t_2	t_3	…

T is the matrix holding all the times corresponding to each new position

- From X and T matrices we can create a slideshow of the various time steps t_1, t_2 …
  - With small time steps the slideshow is smooth and resembles a simulation

Error in simulation

- An easy test to measure the accuracy of the simulation is to compare the gravitational potential and kinetic energy of each particle at each time step.
  - We know that the total energy must remain constant
- Plotting this sum will show us deviations from ideal Newtonian mechanics
- The other thing we can do is to provide a well known initial condition and observe whether or not the motion resembles what we see in nature (i.e. the moon, sun, and earth)

Thank You

References:

spiff.rit.edu/richmond/nbody/OrbitRungeKutta4.pdf

Betounes, D. (2010). Differential Equations: Theory and Applications. New York: Springer Science.

Satellite Effects on the Rings of a Planet

PHYS 210 Term Project Proposal

Kanil Youngman

October 20th/12

The Overview

- The planet Saturn is famous for its amazing rings around it. I would like to try and model Saturn with 4-5 of its largest moons and to observe the effects that it could have on the rings (could dislodge some, potential collisions with satellites etc).

- Collisions will not be considered between ring particles to keep simplicity. If a collision does occur with another satellite, code will make a note of it.

- Modeled using a Toomre model.

The Goals

- To write a program with Matlab that is able to demonstrate the evolution of N particles over small time intervals
- To be able to use a Toomre model to show the effects that satellites will have on the ring.
- To do a model of Earth and examine if it would be “practical” to jazz up Earth with a ring.

The Math

- I will be using Newton's law for gravitation.

$$F=GMm/r^2$$

Where r= a 3d vector

$$r^2= (r_2-r_1)/|r_2-r_1|^3$$

- Don't forget F=ma and basic kinematics

The Approach

- For a 3-d model, I will be using a vector approach:
- Final acceleration: $a=\langle ax, ay, az \rangle$
- Initial acceleration: $a_o=\langle a_{xo}, a_{yo}, a_{zo} \rangle$
- Etc.
- It will be plugged all into the Gravitation formula and will spit out coordinates, times, and speeds

Still Approaching

- Collisions will not be considered to keep things simple but if locations or two objects do intersect I will know about it and record it down as a collisions.
- The N bodies in the rings will not interact with each other. Only against the planet and any moons around it.

The Schedule

Due Date	Activities
Oct 30	Research and code design
Nov 6	Implement code and debug
Nov 13	Run tests with real initial conditions Saturn/Earth
Nov 20	Begin presentation and final report
Nov 27	Present findings
Nov 30	Hand in final draft

The References

- http://en.wikipedia.org/wiki/N-body_problem
- http://en.wikipedia.org/wiki/Saturn
- http://laplace.physics.ubc.ca/210/


Compliments? Complaints?
© 2009 Jerry Lodriguss / Astropix

Optics - Ray tracing through series of lenses, prisms, mirrors etc

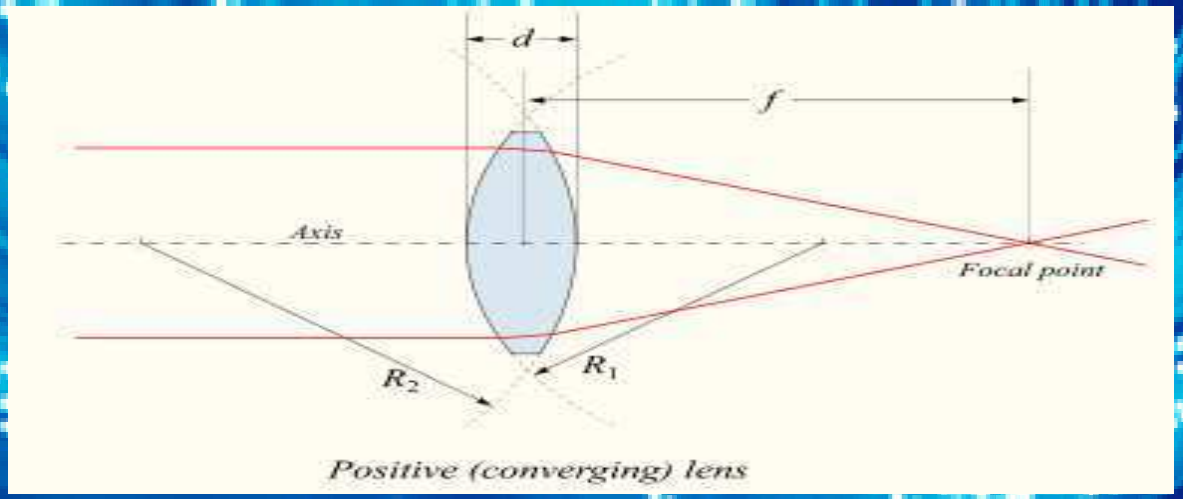
PHYS 210 Term Project Proposal

Chao Wei (David) Chen

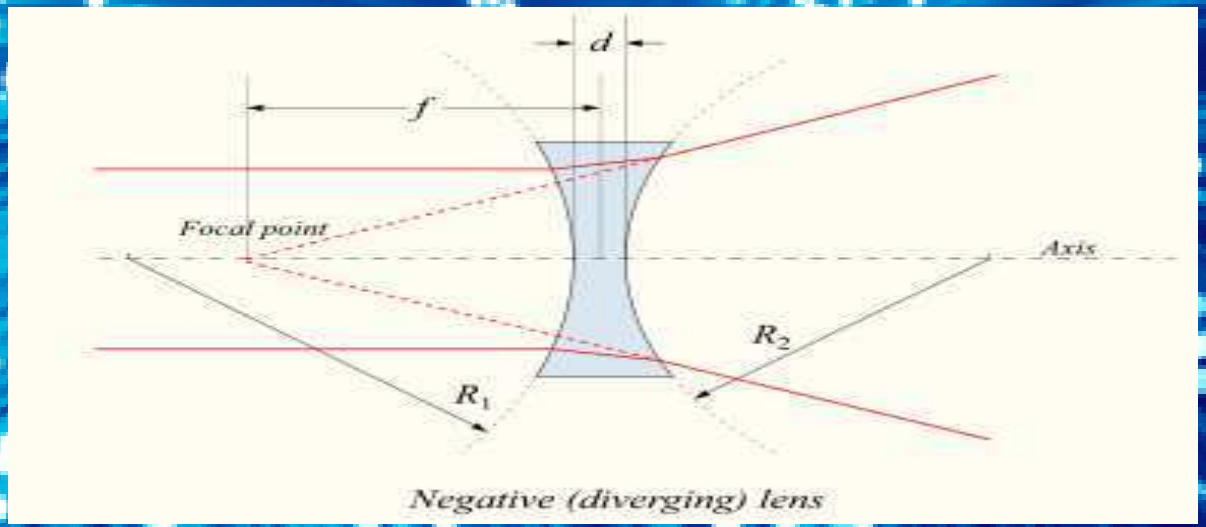
Oct. 22nd, 2012


• Overview
- Light is an electromagnetic wave.
- Light can cause reflection, refraction when intersect with a lens material.
• Project Goals
- To make laser applications and/or use lasers in many ways, such as spectroscopy, microscopy, laser cutting, eye surgery, etc.
- To design and/or upgrade electronic devices.

- Mathematical formulation



Positive (converging) lens



Negative (diverging) lens

- **Mathematical formulation (continued)**

  - Lensmaker's equation

$$P = \frac{1}{f} = (n-1)\left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2}\right],$$

  where

  -P is the principal plane offsets

  -f is the focal length of the lens

  -n is the refractive index of the lens material

  -R_1 is the radius of curvature of the lens surface closest to the light source

  -R_2 is the radius of curvature of the lens surface farthest from the light source

  -d is the thickness of the lens

- Testing

  - Use different radii of the lens surface

  - Test refractive indexes of the lens material

Project Timeline

Dates	Activities
10/19	Do basic research, designing
10/20-21	Test code
10/21-22	Implement code, test code, and prepare numerical experiment
10/23	Begin presentation

References

- http://www.newport.com/Technical-Note-Optics-Formulas/144956/1033/content.aspx
- http://en.wikipedia.org/wiki/Lens_(optics)
- https://www.google.com/search?hl=en&newwindow=1&q=optics&bav=on.2,or.r_gc.r_pw.r_cp.r_qf.&bpcl=35466521&ion=1&biw=1920&bih=889&um=1&ie=UTF-8&tbm=isch&source=og&sa=N&tab=wi&ei=MqGGUMSKAenriwL2pIGYDQ

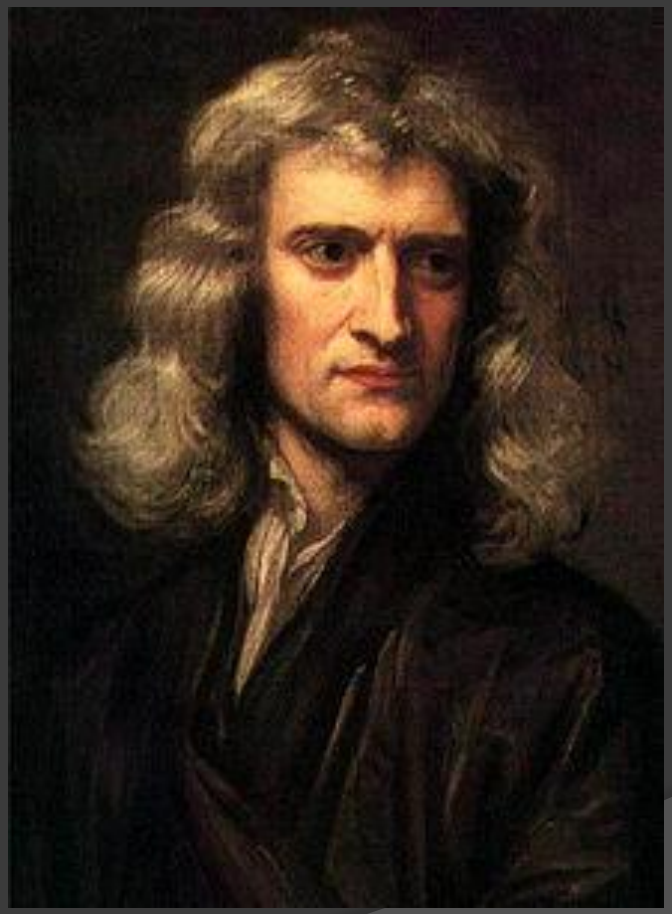
A Simulation of *N* Particles Interacting Gravitationally by Means of Finite Difference Approximation:

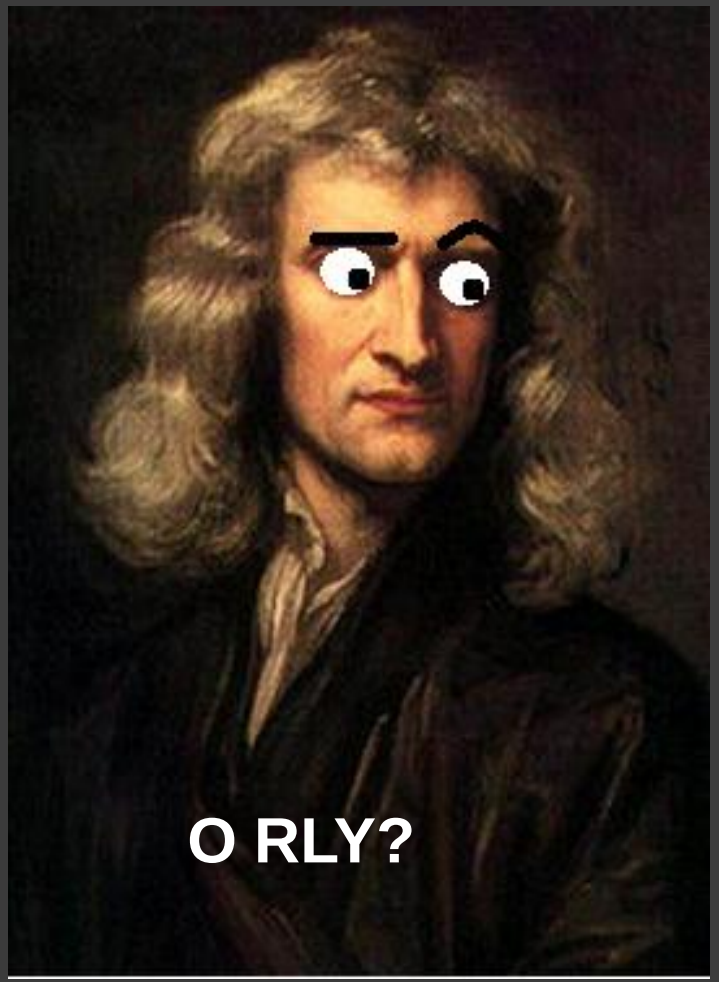
THE TOOMRE MODEL OF GALAXY COLLISIONS

- A Term Project Proposal Presentation by Sebastián Gil 44578110
- PHYS 210, October 2012

Overview:

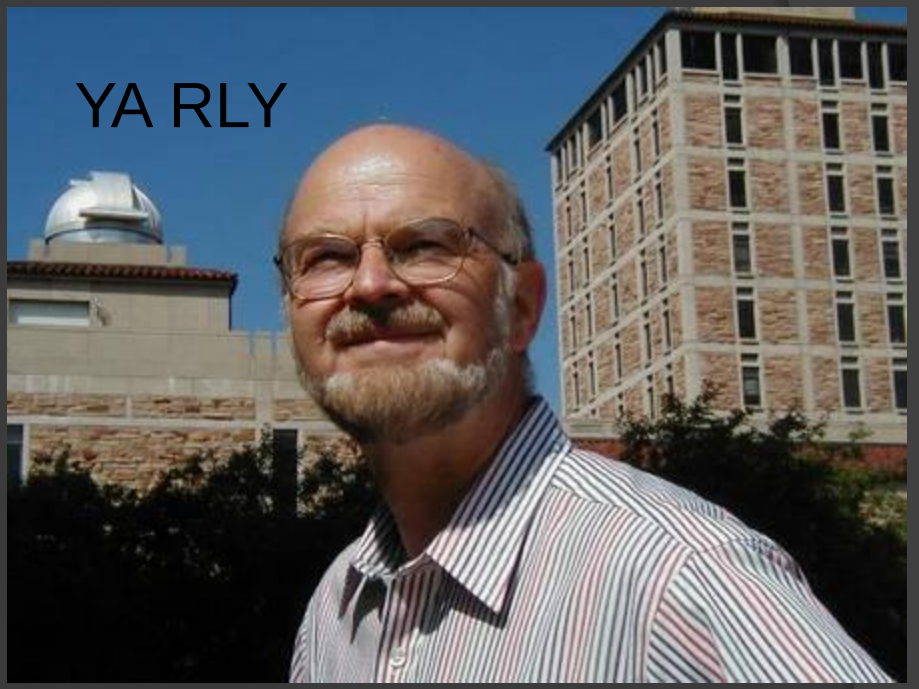
- The N-body Problem: Classically Convoluted
  - Based on the current positions and velocities of celestial objects, is it possible to predict their motions for the future and to deduce their motions in the past?



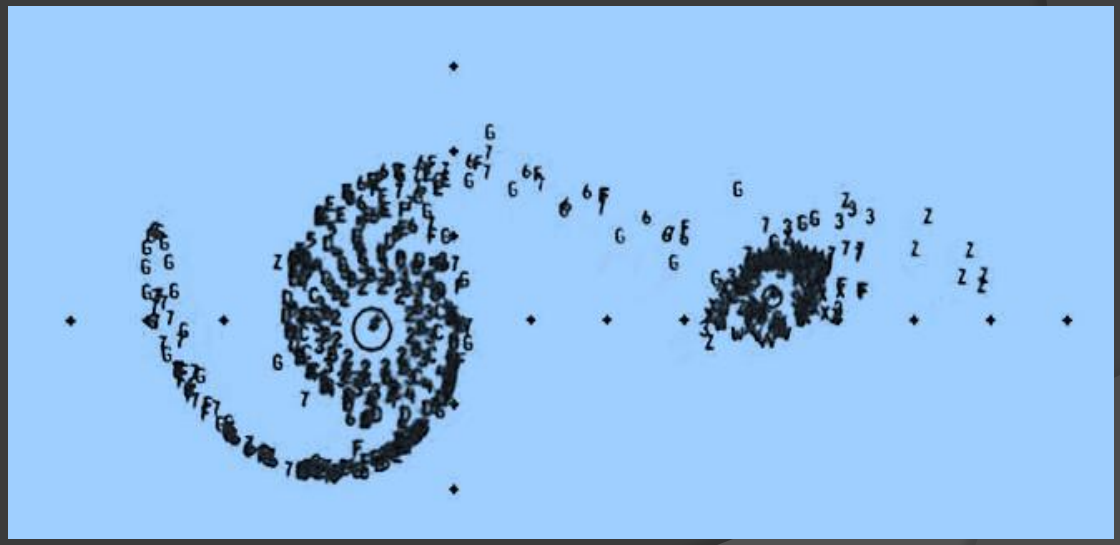


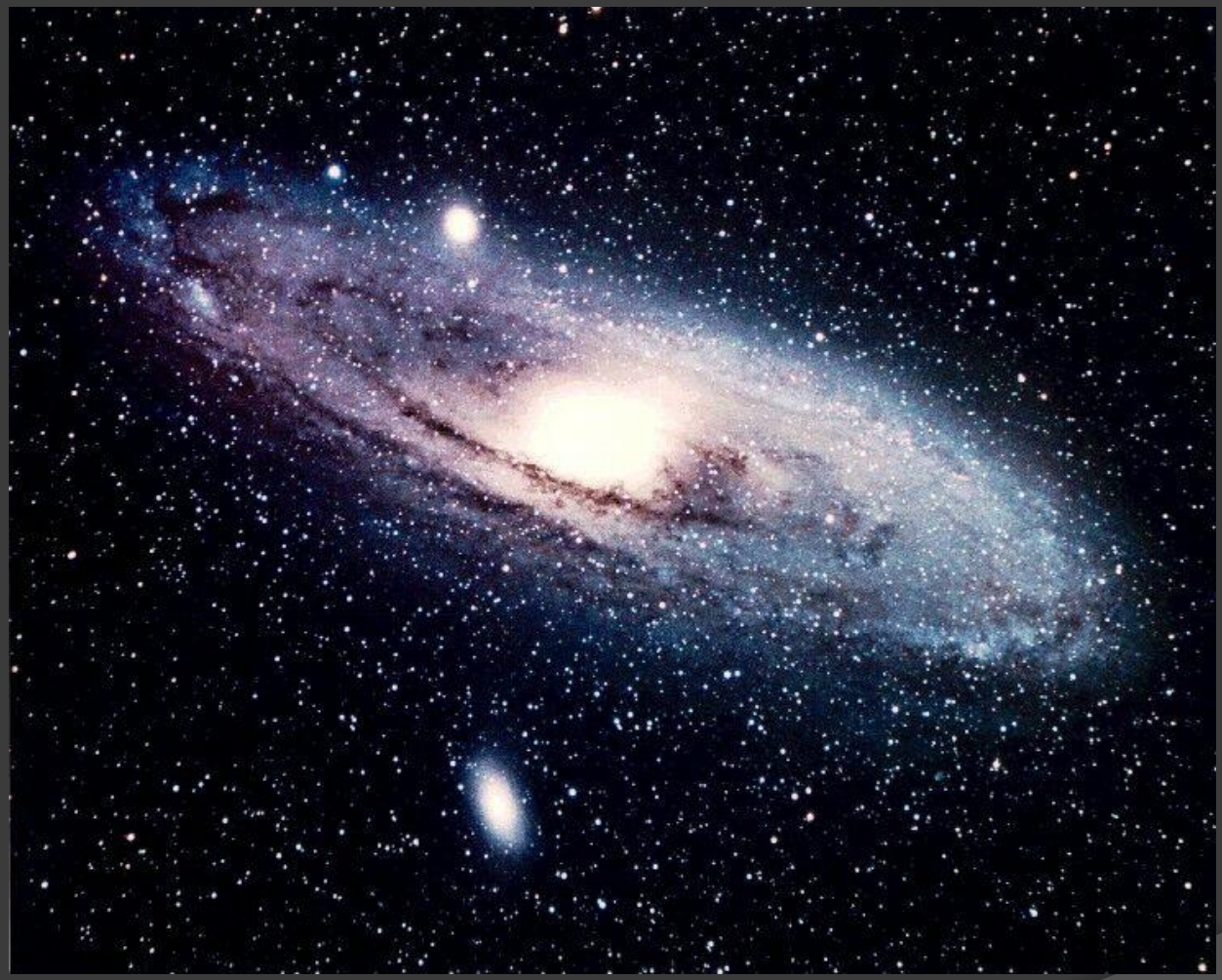
- Yes!
- The N-body problem cannot be solved analytically for $n \geq 3$, but numerical approximation methods *can* be used.

- In the 1970s, Alar Toomre (along with his brother Jüri) were the first to conduct computer simulations to model galaxy mergers.
- Reproduced the Antennae galaxies and their tidal tails




NO WAI!!!





Project Goals

- To write a MATLAB (Octave) code that models numerically the N-body problem in 3D space
- To test this model with different initial parameters of velocity, position, mass, and approach angles
- To reproduce the Toomres' result for the Antennae Galaxies making the simplifying assumption that the mass of the stars in a galaxy are significantly less than that of the galactic core
- To predict the outcome of the merger between the Milky Way and the Andromeda Galaxy
- To verify the validity of the model by means of the law of Conservation of Energy

Mathematical Formulation

- The bulk of the calculations for the gravitational interaction of the objects is based on Sir Isaac Newton's Second Law of Motion and his Law of Universal Gravitation:

$$\vec{F}_{(2,1)i} = G\sum_{i=1}^{n} \frac{m_1 m_2}{\vec{r}_{(2,1)i}^{\,2}}$$

$$\vec{v}_i = \frac{d\vec{r}_i}{dt}$$

$$\vec{a}_i = \frac{d^2\vec{r}_i}{dt^2} m$$

Numerical Approach

- The differential equations for velocity and acceleration are parameterized and re-expressed as three dimensional vector-valued functions
- The simplifying assumption that the galactic core is significantly more massive than the individual stars is made
  - So only the galactic cores will affect each other and their respective stars gravitationally—the gravitational contribution of the stars is deemed as negligible
- A reasonable time-step per iteration has to be determined

- The key step for this model is to express the differential equations as finite difference approximations (FDAs) which can then be manipulated numerically in octave.
- The 1st order differential equation for velocity and the 2nd order differential equation for acceleration are to be re-expressed as approximations using the following definitions:

$$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$$

$$f''(x) \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2}$$

In this approximation, the error can be obtained by means of Taylor Series expansions.

Testing and Numerical Experiments

1. Time-step interval decided based on minimizing error while still saving computational time
2. Seek methods to reduce computational time and increase number of bodies per simulation
3. Test different independent variables: mass, initial velocity and position, approach angles, etc.

4. Reproduce Toomre Model results for merger of Antennae Galaxies, compare with current observed state
5. Use results and Conservation of Energy to determine accuracy of model
6. After improving model, use it to reproduce merger of Milky Way and Andromeda Galaxy, comparing results with other predictions.
7. ????
8. PROFIT!!!

Project Timeline (Tentative)

Date	Task
October 25 – 31	Research necessary physics and mathematical procedures
November 1 – 12	Design code, implement computation saving methods
November 13 – 15	Test code
November 16 – 22	Debug and improve code
November 22 – 24	Test code, optimize model
November 25 – 30	Write report and work on final presentation
November 29	Completed Project Presentation
November 30	Submit completed report and code

References

- http://en.wikipedia.org/wiki/Alar_Toomre
- http://en.wikipedia.org/wiki/Galaxy_merger
- http://en.wikipedia.org/wiki/N-body_problem
- http://kinotonik.net/mindcine/toomre/
- Dubinski, The Great Milky Way – Andromeda Collision
- Past PHYS 210 Term Project Presentations (2009)

Diffusion Limited Aggregation

Phys 210 Term Project Proposal

Kais Jooya

Oct . 25 2012

Overview

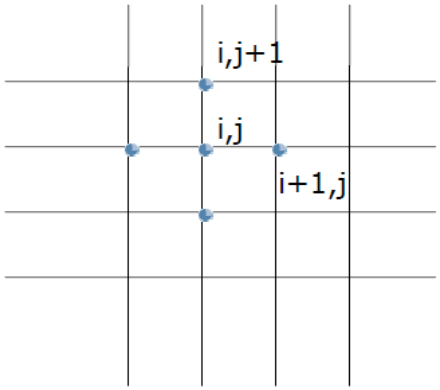
- Diffusion Limited Aggregation is the study of particles that undergo random walks and cluster to form aggregates.

Project Goals

- To write Octave code to model this phenomenon.
- Expand this idea for different initial condition and different influences.

Numerical Approach

Using linear systems we can model this.



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

We can use Markov chains to a weighted transition matrix.

Thank you
for your time.

Diffusion Limited Aggregation

Physics 210 Term Project Proposial
October 25, 2012
Jonathan Ruhnke

http://markjstock.org/photo/2003/2003-07-06/dscn1144_med.jpg
http://markjstock.org/photo/2003/2003-10-26/dscn2322.jpg

Overview and Goals

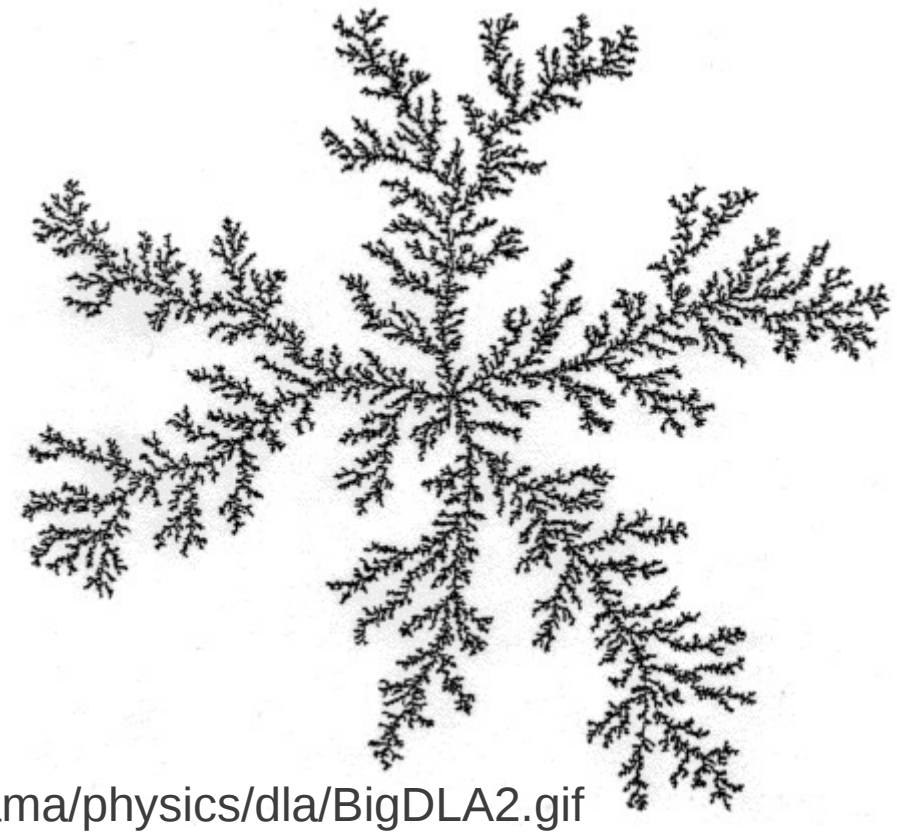
- Create a script in octave that uses diffusion limited aggregation to model one or more physical systems. (dielectric breakdown, crystal formation, etc.)
- Study model and investigate the effects of changing perameters of the system

http://upload.wikimedia.org/wikipedia/commons/thumb/3/3c/Square1.jpg/250px-Square1.jpg

p://upload.wikimedia.org/wikipedia/commons/thumb/b/b8/DLA_Cluster.JPG/250px-DLA_Cluster.JPG

Mathematical Formulation

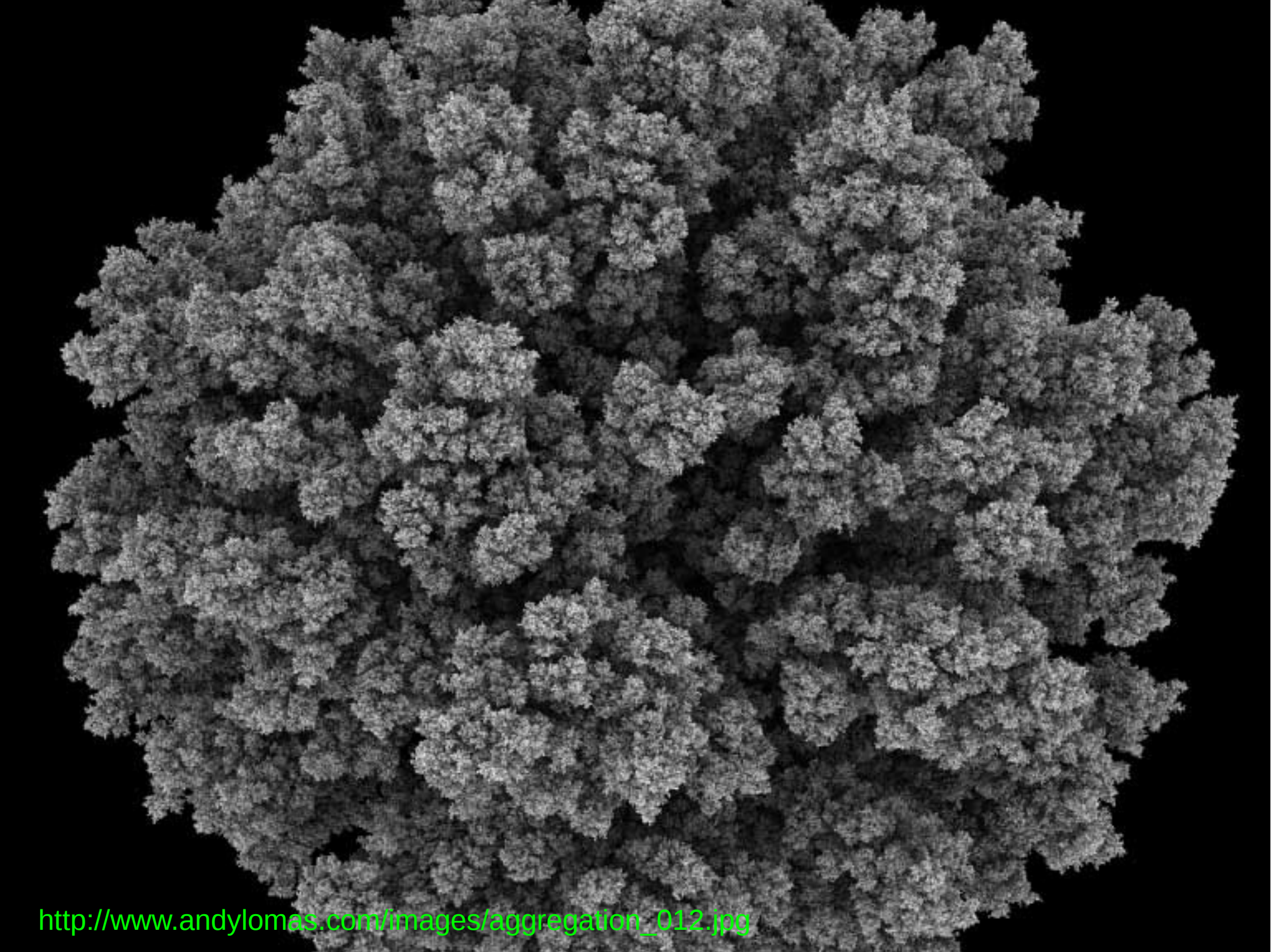
- Start with a seed at some position
- Set a particle at some other postion and allow it to randomly walk until it hits the seed or a previous path
- Repeat



http://classes.yale.edu/fractals/panorama/physics/dla/BigDLA2.gif

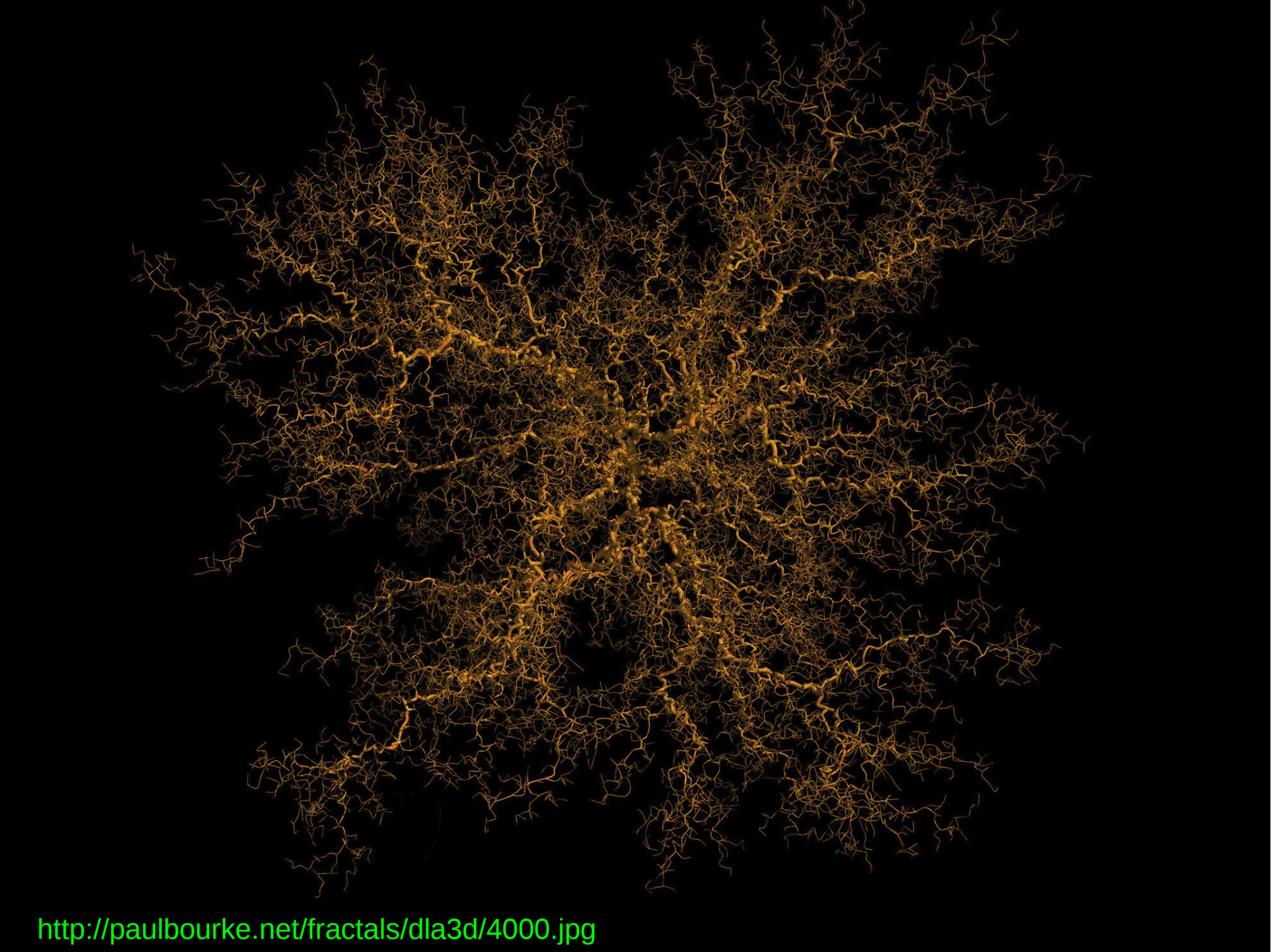
Timeline

Dates	Goals
10/22-10/27	Research and design code
10/28-11/10	Implement and test code
11/11/-11/17	Run experiments, begin presentation and report
11/18-11/24	Analyze data, continue on presentation and report
11/25 12/02	Finish presentation and report
12/03	Final presentation
12/04	Hand in report
12/05	Sleep



http://www.andylomas.com/images/aggregation_012.jpg

http://toxiclibs.org/wp-content/uploads/2010/02/dlatest_spiral2-680x382.png



http://paulbourke.net/fractals/dla3d/4000.jpg

N-BODY SIMULATION OF GRAVITATIONAL INTERACTION

PHYS 210 Term Project Proposal

NATHAN CLINE

OCTOBER 23RD, 2012

- # OVERVIEW

  - The N-Body problem involves being able to predict the future and past motions of all bodies in a system given their present positions and velocities
  - Given any initial velocity and position, the state of the system can be determined at an arbitrary time t

- # GOALS

  - Design a functioning Octave code in 2D space which accurately simulates N-bodies under the effect of gravity, given initial conditions
  - Verify the Octave code by implementing the Law of Conservation of Energy to ensure the outcome is what is desired

CODING APPROACH

- Define a scope for the model (i.e. initial viewing boundaries)
- Define a “body” with parameters size and mass
- Define functions for initial position and velocity (in both x and y)
- Test that the bodies move in linear motion in the absence of gravity, given initial conditions
- Implement FDA for gravitational effect and apply to each body
- Check, exhaustively, at points and different initial conditions to ensure that the Law of Conservation of Energy holds

PROJECT TIMELINE

Dates	Activities
10/23 – 10/29	Derive FDA for Gravitational Effect, Design Scope
10/30 – 11/05	Implement and test Linear Motion, Implement FDA for Gravity
11/06 – 11/12	Test Code for errors with varying # of bodies and initial conditions
11/13 – 11/19	Run specific mathematical experiments on all aspects of the simulation, Begin report & presentation
11/20 – 11/26	Continue verifying simulation mathematically, Finalize report

References

http://www.math.uvic.ca/faculty/diacu/diacuNbody.pdf

N-Body Problem

Physics 210 Term Project Proposal

Connor Greig

Term 1 2012

- Overview
  - The N-body problem is the problem of describing the motion of N bodies under their mutual influence
  - We need to perform computer simulations to describe the motion reliably – done by numerical integrations
  - I will focus on the common fourth-order Runge-Kutta method
  - The N-body problem can be solve as an IVP using this method

- Project Goals
  - To write a code in Matlab (taking in to account initial values), approaching the N-body problem numerically
  - To test this code using different parameters

- Mathematical Formulation
  - Specify IVP as: $\dot{y} = f(t, y), \quad y(t_0) = y_0.$
  - The Runge-Kutta method for an IVP is given by:

    $y_{n+1}=y_n+h/6(k_1+2k_2+2k_3+k_4)$
  - and

    $k_1=f(t_n, y_n)$

    $k_2=f(t_n+h/2, y_n+k_1h/2)$

    $k_2=f(t_n+h/2, y_n+k_2h/2)$

    $k_2=f(t_n+h/2, y_n+k_3h/2)$

    Which defines k_1 through k_4.

- Mathematical Formulation
  - Our problem involves a 3D second order differential equation, need to modify previous equation
  - Express gravitational potential U of object in coordinates x,y,z – we can compute accelerations in all directions from:

$$x'' = -\,dU/dx$$

$$y'' = -\,dU/dy$$

$$z'' = -\,dU/dz$$

Mathematical Formulation

- We define position vector r, velocity vector v and acceleration vector a as the following:

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \vec{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \quad \vec{a} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix}$$

- Which means

$$\dot{\vec{r}} = \vec{v}$$

$$\dot{\vec{v}} = \vec{a}$$

Mathematical Formulation

- Definition for k coefficients (r' = v):

$$\begin{aligned}
\vec{k}_{1_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i\right) \\
\vec{k}_{2_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + \vec{k}_{1_{r_{i+1}}}\frac{h}{2}\right) \\
\vec{k}_{3_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + \vec{k}_{2_{r_{i+1}}}\frac{h}{2}\right) \\
\vec{k}_{4_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + \vec{k}_{3_{r_{i+1}}}h\right)
\end{aligned}$$

- Definition for k coefficients (v' = a):

$$\begin{aligned}
\vec{k}_{1_{r_{i+1}}} &= \vec{v}_i \\
\vec{k}_{2_{r_{i+1}}} &= \vec{v}_i\vec{k}_{1_{v_{i+1}}}\frac{h}{2} \\
\vec{k}_{3_{r_{i+1}}} &= \vec{v}_i\vec{k}_{2_{v_{i+1}}}\frac{h}{2} \\
\vec{k}_{4_{r_{i+1}}} &= \vec{v}_i\vec{k}_{3_{v_{i+1}}}h
\end{aligned}$$

- Coefficients need to be computed alternately for velocity and position
- Lastly can determine velocity anfd position vector at next time step by:

$$\vec{v}_{i+1} = \vec{v}_i + \frac{h}{6}\left(\vec{k}_{1_{v_{i+1}}} + 2\vec{k}_{2_{v_{i+1}}} + 2\vec{k}_{3_{v_{i+1}}} + \vec{k}_{4_{v_{i+1}}}\right)$$

$$\vec{r}_{i+1} = \vec{r}_i + \frac{h}{6}\left(\vec{k}_{1_{r_{i+1}}} + 2\vec{k}_{2_{r_{i+1}}} + 2\vec{k}_{3_{r_{i+1}}} + \vec{k}_{4_{r_{i+1}}}\right)$$

- # Project Timeline

Dates	Activities
Oct. 22-28	Begin research on mathematical formulation and design code
Oct. 29-Nov. 4	Finish designing code, implement code
Nov. 5-Nov. 11	Test code
Nov. 12-Nov. 18	Run numerical experiments, begin presentation and report
Nov. 19-Nov. 25	Analyze data, continue to work on presentation and report
Nov. 26	Polish presentation and work on final draft of report
Nov. 27	Give presentation
Nov. 28-Nov. 30	Review final draft and finish report
Nov. 30 (11:59pm)	Submit report

N-BODY SIMULATION PROJECT

By Mike Koonts

Physics 210

Fall 2012

The Problem

- Two – Body problem
  - Completely solved
  - Newton published a solution in his Principia Mathematica
- Three – Body problem
  - There is no solution
  - All solutions are for special cases, or Taylor series approxamations

Goals

- Use a combination of C++ and Octave to create a general n-body approximation simulator
- Use that simulator to view the out come of different sets of initial conditions
- Simulate the gravitational interaction of a hyper velocity star passing near, or through, a star system

Schedule

Oct 21 - 27	Gather Ideas and Plan
Oct 28 – Nov 3	Write Pseudo Code
Nov 4 – 10	Code/Test
Nov 11 – 17	Code/Test
Nov 18 – 24	Polish
Nov 25 – 28	Finalize

Simulation of a N-Charge distribution on a sphere

Physics 210 Project Proposal

Alexandre Silva - 22 October 2012

References

- http://en.wikipedia.org/wiki/Coulomb's_law

Project Overview

- When n-like charges are placed on a sphere, they spread as to maximize distance from one another
- The potential is also minimized when distances are at maximum and this where the system will want to go
- Coulomb's Law presents the electrostatic force that is responsible for the behaviour of charges on the surface

Project Goals

- Simulate the behaviour of n-charges, that can be initiated with random positions or with specific geometric initial positions
- Understand what are the formed shapes and lattices that correspond to equilibrium states
- Use this approach to extend simulation for other shapes and surfaces

Mathematical Formulation: Coulomb's Law

- For a given particle j, the total Coulomb force can be written as the addition of all Coulomb forces that are being exerted on this particle:

$$\mathbf{F}_j = \frac{q}{4\pi\epsilon_0} \sum_{i=0}^{N} \frac{q_i(\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3}, i \neq j \qquad (1)$$

- The movement of charges is to be noted as that includes the dissipation of energy due to friction to achieve equilibrium. This force is proportional to the speed of the particle and can be formulated like:

$$F_f = -\gamma v \qquad (2)$$

-

Numerical Approach

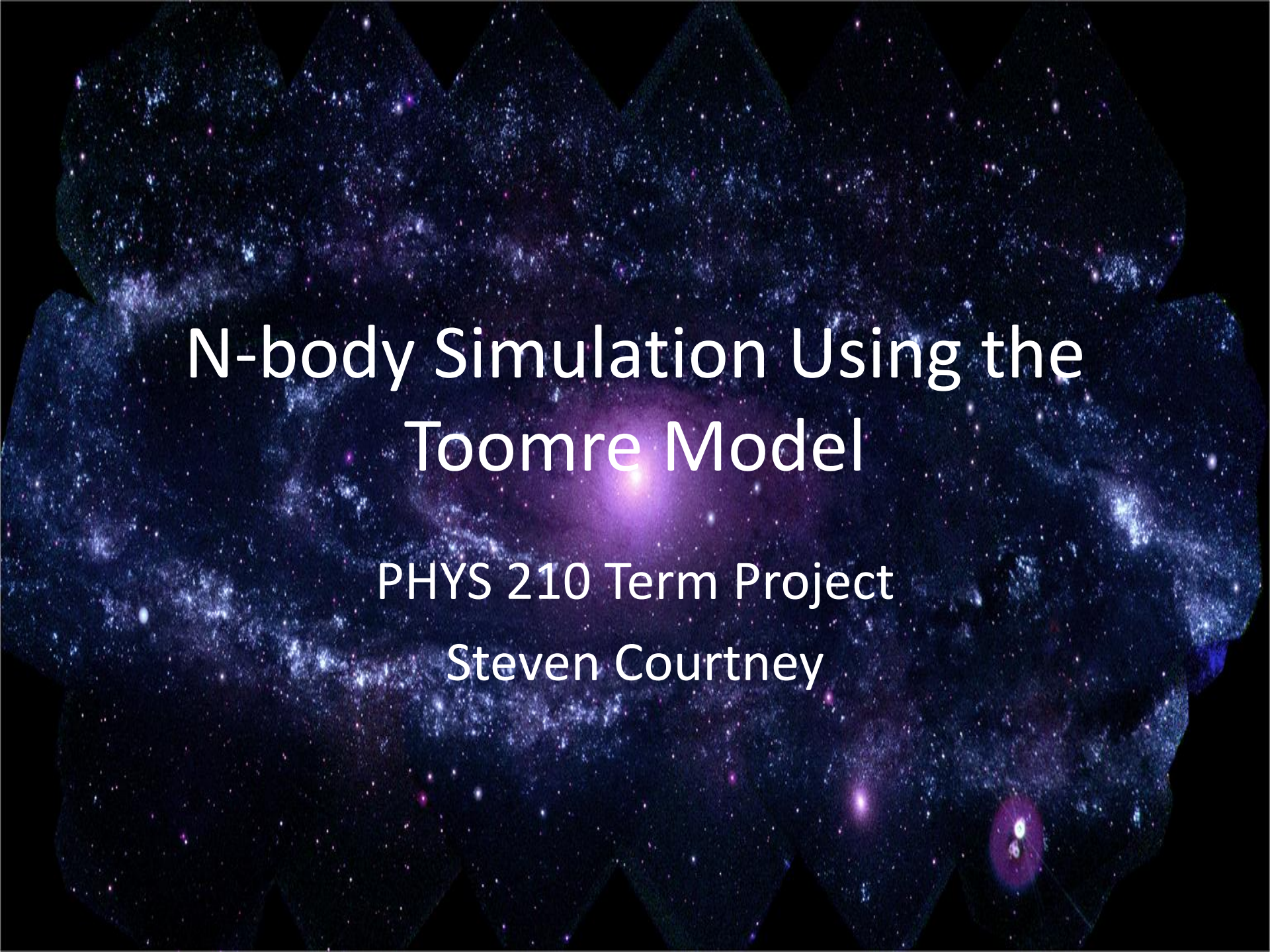
- FDA's will be used to find solutions for the differential equations that describe the system
- The initial positions of charges can either be determined randomly or can be determined by using specific coordinates
- For simplicity, particles will always start at rest and the sphere will have an unitary radius

Visualization, Testing and Numerical Experiments

- Check the quantity of time steps required to achieve equilibrium by increasing the amount of particles
- Do the same as previous while varying the initial conditions/positions
- Visualize the lattices formed at equilibrium and increase quantity of particles to achieve more complex structures
- Visualization and Plotting Tools:
- - Matlab will be used for the animations

Project Timeline

Dates	Activity
Now - Nov. 6	research, derive eqns
Nov. 7 - Nov. 20	implement and test code
Nov. 21 - Nov. 25	run numerical experiments and write report
Nov. 26 - Nov. 28	analyze data, continue presentation and report
Nov. 29	polish final presentation
Dec. 1	give presentation
Dec. 1 - Dec. 4	finish and submit report



N-body Simulation Using the Toomre Model

PHYS 210 Term Project

Steven Courtney

Overview and Goals

- Program an N-body simulation in octave showing the interaction of two galaxies in close contact with one and other.

- Use the assumption that only the heavy bodies effects are considered on both the light and heavy bodies.

Overview and Goals

- Be able to see the effects of varying initial conditions on the system.

- Make comparison to other simulations of the collisions of galaxies.

Mathematical Model

- Newton's law of gravitation:
- $F_{net} = \sum F_i = m_i a_i$
- $F = ma = m\frac{d^2y}{dx^2} = \frac{Gm_1m_2}{r^2} \cdot \hat{r}$
- Use finite difference approximations for first order partials to solve.

Timeline

DATE (Week of)	PLAN OF ACTION
OCT 21st	Finalize mathematical formulation
OCT 28th	Start writing code
NOV 4th	Finish writing and implementing code
NOV 11th	Test code and start report
NOV 18th	Finalize report
NOV 25th	Presentation and report submission!


Questions?

Boundaries of Overlapping Gravitational Fields

Physics 210 Term Project Proposal

Zhao Yi (Johnny) Han

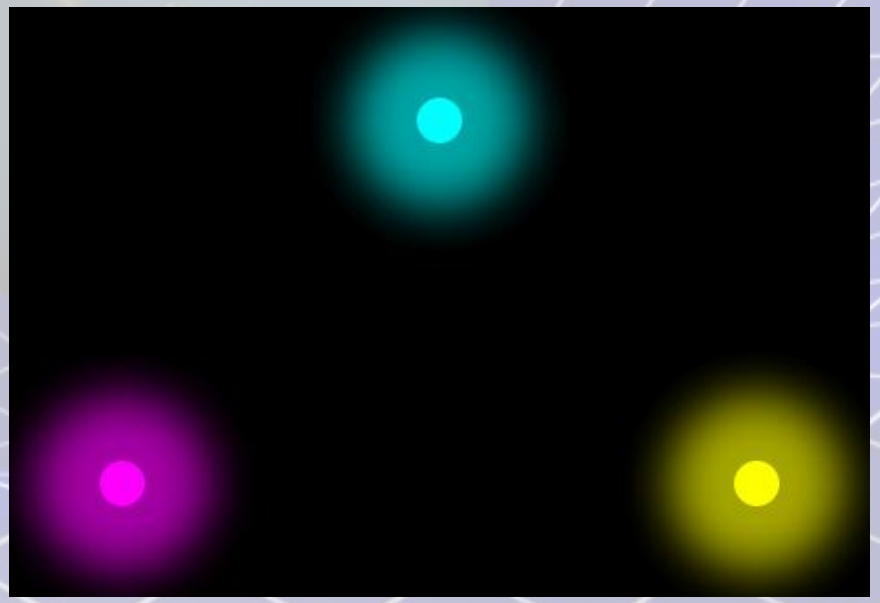
October 25, 2012

Overview

- Gravitational attraction occurs between any two massive bodies.
- Magnitude of gravitational force obeys inverse square law as distance changes.
- Multiple forces linearly superimpose.
- Object's acceleration proportional to force and inversely proportional to mass.

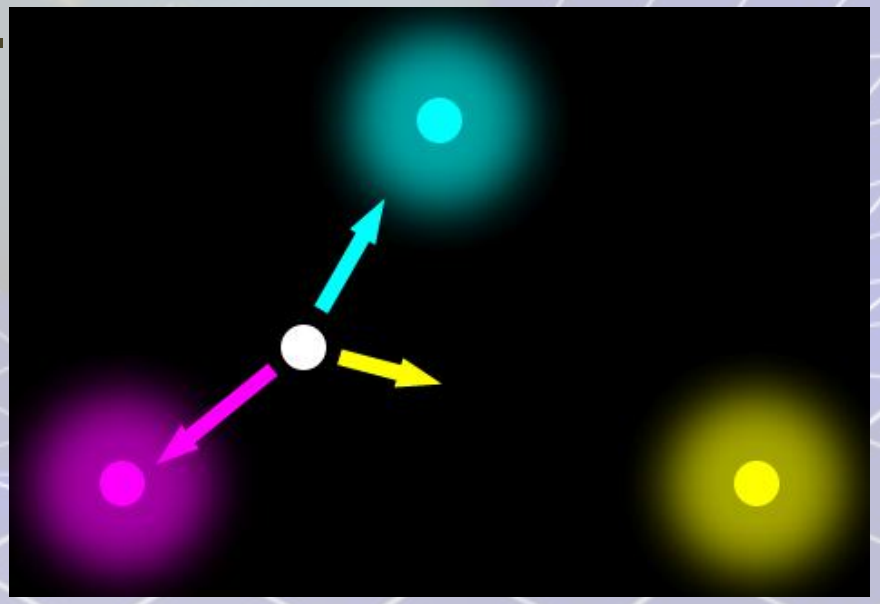
Project Goals

- This will take some explaining so bear with me...
- First, place a few sources of gravitational force on the 2D plane, and assign each a color.
- The gravity sources have mass but are assumed to be immovable.



Project Goals

- Place a test mass at some point in space. This test mass experiences gravitational forces from all three gravity sources.
- The test mass is initially motionless. Calculate its eventual destination.
- If it arrives at within a certain radius of one of the sources, color the test mass's starting point the color assigned to the mass it arrives at.



Project Goals

- Do this for every point in the 2D plane at some finite resolution, to obtain an image of the presumably fairly complicated boundaries of the gravitational fields of the gravity sources.
- Boundary may be a fractal but I am not sure.
- Note that while it is calculated, the test mass's trajectory isn't visibly shown in this simulation.

Approach

- The gravitational attraction between two masses is

$$\vec{F} = \frac{GMm}{|\vec{r}|^2}\left(\frac{\vec{r}}{|\vec{r}|}\right)$$

  where $\vec{r} = (\Delta x, \Delta y)$ is the distance vector between the masses, M and m are the masses, and G is the gravitational constant.
- The acceleration of a particle obeys

$$\vec{F} = m\vec{a}.$$

Approach

- This simplifies to

$$\vec{a} = \frac{GM}{|\vec{r}|^2}\left(\frac{\vec{r}}{|\vec{r}|}\right)$$

- A finite difference approximation will be used to approximate the test mass's motion.

$$\vec{r}_{j+1} = \vec{r}_j + \Delta t\left(\vec{v}_j\right)$$

$$\vec{v}_{j+1} = \vec{v}_j + \Delta t\left(\vec{a}_j\right)$$

- A program will be written in MATLAB (octave) to recursively calculate the particle's trajectory

Testing

- In 1D, using only one gravity source, see if the simulation reasonably approximates classical kinematics and dynamics.
- In 2D, using only one gravity source, by setting the initial velocity to some nonzero value, see if circular, elliptic, parabolic, and hyperbolic trajectories can be produced.

Experiments

- Various numbers of gravitational sources will be placed in various locations.
- Some gravity sources may have different strengths and/or different radii. Radius may or may not depend on strength.
- Test masses may be given nonzero initial velocities.
- Modifications to the inverse square law may be attempted.

References

- http://bh0.phas.ubc.ca/People/matt/Teaching/09Fall/PHYS210/Doc/term-projects/kdv.pdf
- Wikipedia
- Newton's laws of motion and universal gravitation as remembered from high school

N-Body Gravity Simulation

Phys 210 Term Project Proposal

Gilbert Lee
October 25, 2012

Overview

- The N-**body problem** is the problem of predicting the motion of particles that interact with each other gravitationally.
- Given initial position and velocity, we can solve the N-body problem
- It is therefore possible to predict the position and time as well as velocity of the particles.
- Solving this problem can help us better understand the motion of larger objects such as the sun or other orbital planets.

Project Goals

- Write Matlab (Octave) codes to solve the N-body problem
- Perform a simulation of the results and demonstrate using visualization tools
- To test the codes for correctness using solutions of 2-body problems (N=2)

Mathematical Formulation

- Newton's Second Law F=ma can be written as:

$$\mathbf{F}_i \equiv m_i \mathbf{a}_i = m_i G \sum_{j=1, j \neq i}^{N} m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3}.$$

where i= 1,2,..N F is the force on the ith particle

G is gravitational constant r is the variable function for time

The right hand side of the equation sums up the forces on the particle

This Equation can solve the problem for n particles in 3 dimensions

Numerical Approach

- I will use a finite difference approximation or Euler's Method to solve the problem (depending on which way is more applicable and appropriate)
- Initial positions and velocities will be specified to generate reasonable results

Visalization and Plotting Tools

- I will use the Matlab software to create a mpeg file
- Or maybe use sm or gnuplot to generate plots

Testing and Numerical Experiments

- Once the codes are working, I will test using solutions for N=2 body systems

Project Timeline

Dates	Activities
10/25-10/28	Do basic research, derive equation and design code
10/29-11/3	Implement code
11/4-11/9	Test code
11/10-11/15	Run numerical experiments
11/16-11/21	Continue work on project
11/22-11/26	Tidy up remaining parts of project
11/27-11/29	Presentation
11/29-11/30	Final amendments
11/30	Submit report

- Reference:
- http://en.wikipedia.org/wiki/N-body_problem

Any Questions??????????????

Most Importantly

THANK YOU very much for your kind attention

Modelling Snowflakes with Diffusion Limited Aggregation

University of British Columbia

PHYS 210 Term Project Proposal

Scott Veale

October 23rd 2012

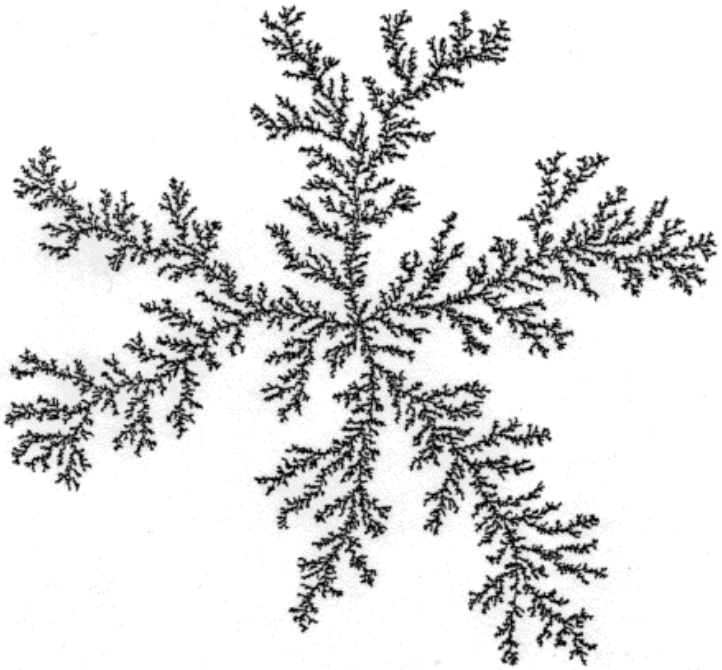
Project Overview

- A snowflake begins to form when water molecules latch on to a piece of dust in cold, humid conditions.
- This intital crystal structure allows the snowflake to show hexagonal symmetry
- The fine details of the snowflake are determined by surrounding temperature and humidity. Unique surroundings lead to the unique structures found in nature

Project Goals

- Write code in Octave to generate a plot that qualitatively resembles a snowflake
- Vary the parameters of the simulation to adjust symmetry and fractal dimension
- Compare fractal dimension to snowflakes found in nature, and propose natural analogs to simulation parameters.

Diffusion Limited Aggregation



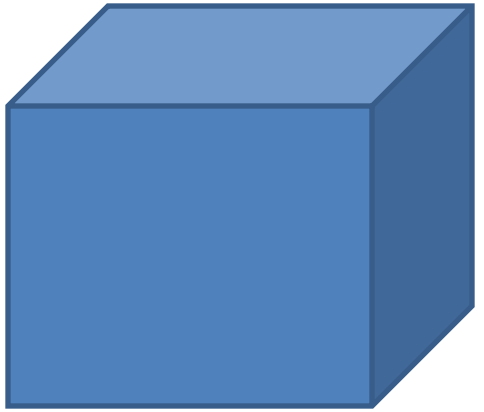
- DLA usually begins with one particle, and adds new particles one by one. Each new particle is initialized in a random location and moves randomly until it encounters the aggregate. The object created exhibits fractal geometry.

- To model a snowflake, I will have to modify the basic DLA premise.
- I can add new particles, augmenting the random additions, to achieve hexagonal symmetry.
- I can tweak the ‘stickiness’ - the probability that a particle will stay where it is when it hits the aggregate - in order to manipulate the structure of the aggregate and its fractal dimension

Fractal Dimension

- For a line, D = 1

- For a square, D = 2



- For a cube, D = 3

- If each particle is the same and so is the distance between them, the number of particles N in an object of length r follows

$$N = r^D$$

where D is the Dimension of the object.

- For a line, D = 1

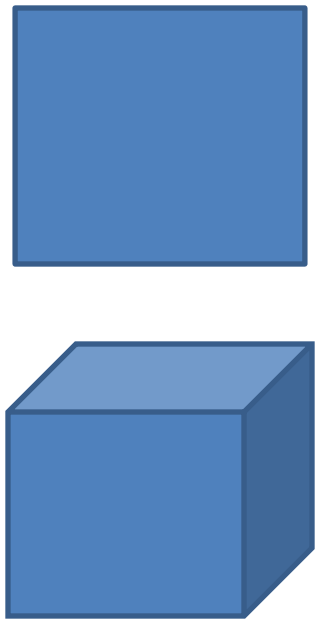
$$N = r^1$$

- For a square, D = 2

$$N = r^2$$

- For a cube, D = 3

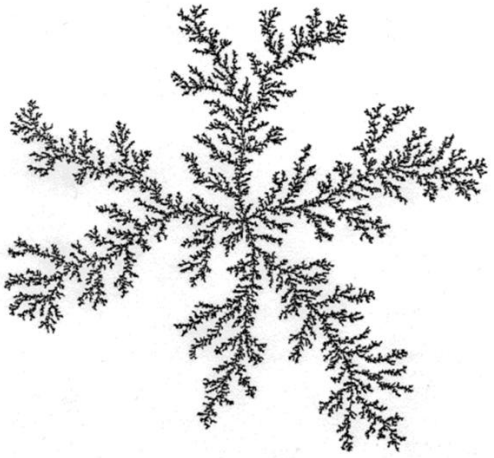
$$N = r^3$$



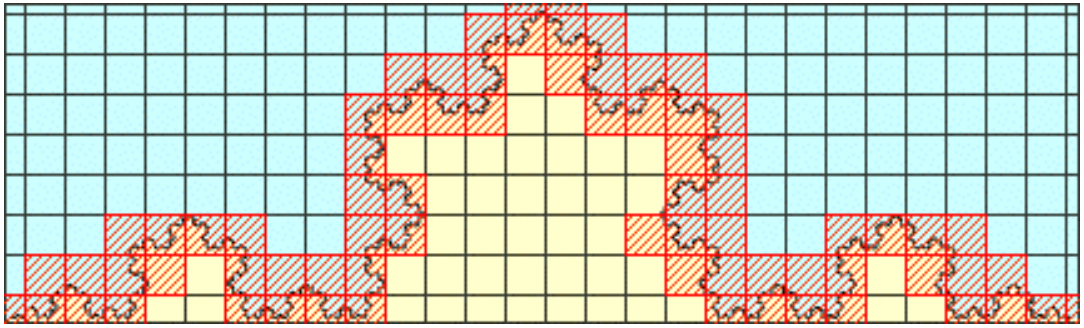
- For normal DLA constructs,

$$D \approx 1.76$$

$$N = r^{1.76}$$



Alternative: Box Counting Method



Another approach is to count the number (N) of boxes of length and height (h) that it takes to cover the object. As h is varied, dimension is given by

$$D = \lim_{h \to 0} \frac{\log(N)}{\log(\frac{1}{h})}$$

Project Timeline

Date	Activity
10/23 – 10/29	Research and design code
10/30 – 11/05	Write code
11/06 – 11/12	Continue writing, test code
11/13 – 11/19	Run experiments, start work on presentation
11/20 – 11/28	Finish work on presentation, start report
11/29	Give Presentation
11/30	Submit Report

References

-“Snowflake” http://en.wikipedia.org/wiki/Snowflake

-“Diffusion-Limited Aggregation”

http://classes.yale.edu/fractals/panorama/physics/dla/snow/snow.html

- “Box counting”

http://en.wikipedia.org/wiki/Box_counting