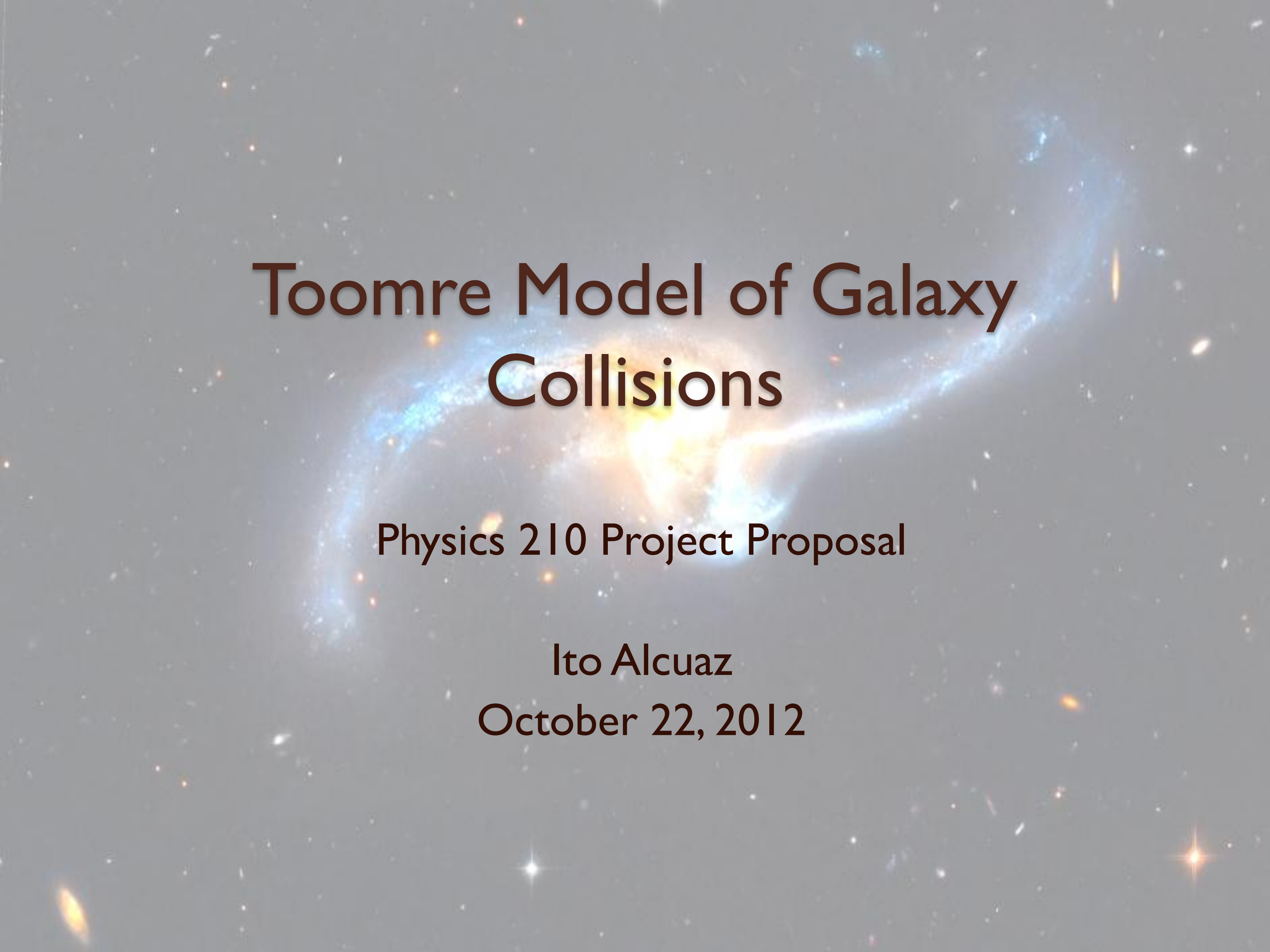


Toomre Model of Galaxy Collisions

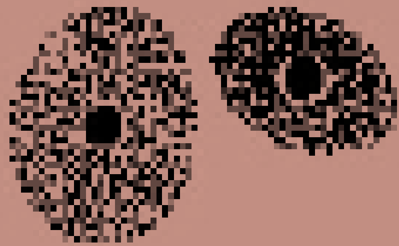
Physics 210 Project Proposal

Ito Alcuaz

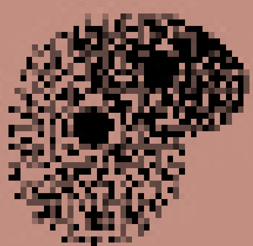
October 22, 2012

Project Overview

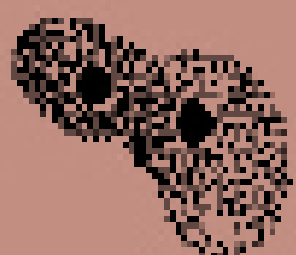
- Galactic collisions occur as a result of the gravitational interactions between two neighbouring galaxies — this alters the configuration of each galaxy.
- The Toomre Model oversimplifies this process — interstellar medium (ISM) and dark matter are ignored.
- Celestial objects are represented as particles — size corresponds to relative mass.
- Only stars and galactic cores are represented in this model.



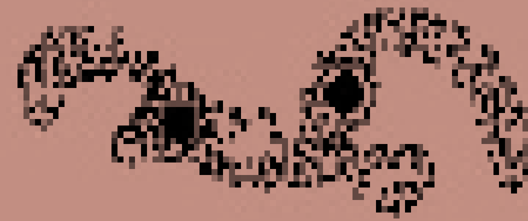
1



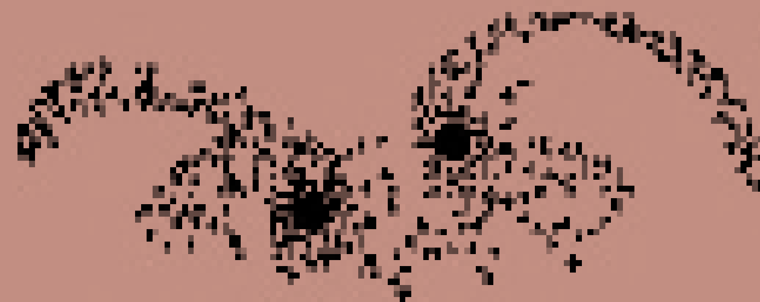
2



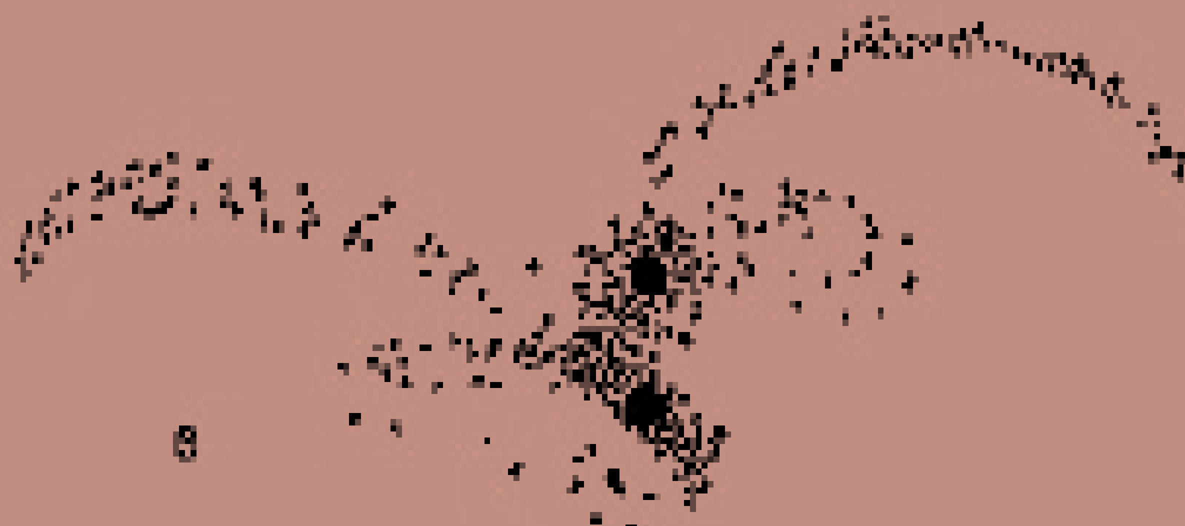
3



4



5



6

Objectives

- To write a Matlab (Octave) code which depicts the collision between two galaxies, using the criterions as defined by the Toomre Model.
- To investigate the case where the mass of one galaxy is much smaller than the other — "galactic cannibalism".
- To create a visual representation from the compiled results — a .mpeg file if possible, otherwise images, graphs, etc.


Milky Way Doomed?

Mathematical Formulations

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

$$F = ma_c = \frac{mv^2}{r}$$

$$P^2 = \frac{4\pi^2}{G(M_1 + M_2)}a^3$$

$$\left(\frac{P}{2\pi}\right)^2 = \frac{a^3}{G(M + m)}$$

(circular orbits)

Assumptions

- ISM (gases and cosmic rays) and black holes ignored.
- M=0 for stars (consider the dominant gravitational force to come from the galaxy core).
- Gravitational interactions between stars are ignored.
- Collisions between stars negated (i.e. they pass through each other).
- Collision between galactic cores negated.

Numerical Approach, Experiments & Testing

- Consider 3-dimensional vectors for all variables.
- Vary initial conditions for each galactic core — in terms of position, size, velocity, angle, shape — similarly for stars.
- Gradually increase n to as large as possible.
- Spiral galaxy + spiral galaxy = elliptical galaxy?
- Attempt to model the (hypothetical) Andromeda-Milky Way collision.

Project Timeline

- 10/22-10/29 — Research & design code
- 10/30-11/05 — Implement code
- 11/06-11/12 — Test code
- 11/13-11/30 — Run numerical experiments, work on presentation & report
- 12/01 — Final Presentation
- 12/04 — Submit report

References

- 1.“Alar Toomre”. Wikipedia: The Free Encyclopedia. http://en.wikipedia.org/wiki/Alar_Toomre. Retrieved 2012-10-22.
- 2. "Toomre Sequence". Cosmos: The Swinburne Astronomy Online Encyclopedia.http://cosmos.swin.edu.au/entries/toomresequence/toomresequence.html?e=1. Retrieved 2012-10-22.
- 3. “Spiral and Elliptical Galaxies”. CambridgeCosmology. http://www.damtp.cam.ac.uk/research/gr/public/gal_class.html. Retrieved 2012-10-22.

October 22^{nd} 2012

Project Proposal

Gravitational problem of a uniform disk with 100 discontinuities

Pritish Amit Bhoonah

PHYS 210

Overview

- Project Description

  - Variant of the Gravitational N Body Problem

  - Initially intended a (very) simple model for star forming clouds but reduced it to a 2 dimensional problem in polar coordinates

  - Goal is to use Matlab (Octave) to solve this variant of the N Body problem and to test the model

  - Backup plain is the regular Gravitational N body problem

Theoretical Aspects

-
- Problem consists of a disk, radius R = 110 parsecs of uniform mass density except at 100 locations chosen arbitrarily (using Polar Coordinates)

$$\boldsymbol{R_n(r,\emptyset) = (\,n, \frac{2\pi n}{100}\,)}, \;\; 1 \leq n \leq 100$$

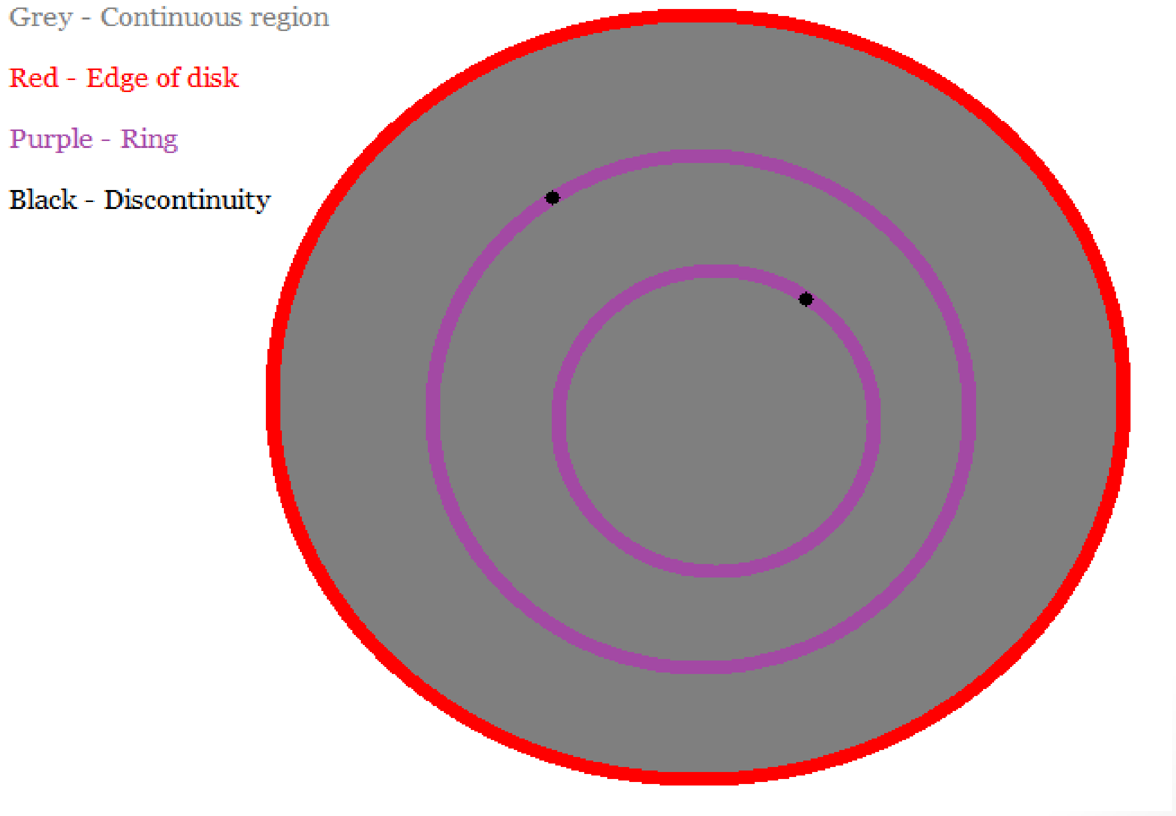
- Mass is given by

$$\boldsymbol{M(r,\emptyset)} = \;\boldsymbol{\rho}, (r,\emptyset) \neq R_n(r,\emptyset)$$

- Value of ρ and masses of the discontinuities to be determined later to match a typical Hydrogen Molecular Cloud mass (100 - 1000000 solar masses)

Theoretical Aspects

- Problem will be separates into one dimensional and two dimensional rings, as illustrated below for a disk of 2 discontinuities



Computational Aspects

- Euler-Lagrange Equations solved on the domain

$$0 \leq r \leq 110,\ 0 \leq \emptyset \leq 2\pi,\ \ 0 \leq t \leq T\ ,$$

with T and initial conditions to be determined later.

- Euler-Lagrange equations will be discretized using the appropriate Finite Difference technique and solved using the Newton-Raphson method

- Results will be plotted and tested; techniques to be determined later

Timeline

Date	Activity
Oct 22 – Oct 29	Finish theoretical details and decide if project will be pursued or not; begin code design
Oct 30 – Nov 15	Implement and test code; run numerical experiments
Nov 15 till end	Analyse data and work on presentation and report

Questions
&
Comments

N Gravitational Body Simulation

Physics 210 Term Project Proposal

Bowen Chan

Oct 23, 2012

Overview

Simulation of future states of
n gravitationally interacting particles
given initial conditions

Project Goals

Solve the n-body problem in 3D numerically with Octave code

Mathematical Formulation

$$F_{g,i,j} = -\frac{Gm_i m_j}{|r_i - r_j|^3}(r_i - r_j)$$

Force on the i^{th} body due to the j^{th} body

Numerical Approach

Use FDAs

$$v_{i,k+1} = v_{i,k\text{-}1} + 2\,a_{i,k}\Delta t,\; v_1 = v_0 + a_0\Delta t$$

$$r_{i,k+1} = r_{i,k\text{-}1} + 2\,v_{i,k}\Delta t$$

$$r_{i,k+1} = 2\,r_{i,k} - r_{i,k\text{-}1} + a_{i,k}\Delta t^2$$

$$r_{i,k+1} = r_{i,k} + v_{i,k}\Delta t + \frac{1}{2}a_{i,k}\Delta t^2$$

$$a_{i,k} = \sum_{j}^{N\text{-}1} F_{g,i,j,k}$$

Project Timeline

Week	Activities
Week 1 Oct 28 – Nov 3	Research problem and mathematical formulations
Week 2 Nov 4 – Nov 10	Write approximations for equations in Octave
Week 3 Nov 11 – Nov 17	Implement and test code
Week 4 Nov 18 – Nov 24	Analyze data and complete report and presentation
Week 5 Nov 25 – Nov 30	Present and submit report

Comments? Questions?

The Spreading of Forest Fires Using Cellular Automata

PHYS210 Term Project

Carly Christy

October 23rd, 2012

Overview

- I will be simulating fire growth across a landscape grid of cells.
  - Cellular automata are systems operating over a fixed space and time, on a lattice. Each cell has a starting state and rules governing their transitions to other states
- For simplicity, the area considered will be a flat plane and wind speed will be neglected.

Project Goals

- To simulate the spread of forest fires by writing and implementing a MATLAB code
- Further my knowledge in the usage of Cellular Automata.
- Test my model to insure it agrees with forest fire behavior.

Project Goals

- To simulate the spread of forest fires by writing and implementing a MATLAB code
- Further my knowledge in the usage of Cellular Automata.
- Test my model to insure it agrees with forest fire behavior.

Cellular Automata Transition Rules

$$S_c^{t+1} = f\left(S_c^t, S_{n(c)}^t\right)$$

This transition rule will take the into account the previous state of the cell,   as well as the states of the neighboring cells. With this input, the transition rule will return a new state for the cell after one time step.

- The Rules
  - The Rules of each cell are as follows:
    - A cell (Tree) may only catch fire in the next step if it is in an ignitable state and [at least] one of its neighboring cells are burning. It will remain in its natural state otherwise.
    - In the next step, a burning cell becomes a burnt cell.
    - A burnt cell will become a growing cell in the next step.
    - A growing cell will become an ignitable cell in the next step.

Each state will be given a numerical value thus allowing a numerical transition in the code.

The state of the cell <i,j> at time t is given by:

$$s(t)_{ij} = \frac{\text{burned our area of } \langle i,j \rangle}{\text{total area of } \langle i,j \rangle}$$

-The state of the cell <i,j> at time t+1 is given by:

$$s(t+1)_{ij} = g\left(s(t)_{i+\alpha,j+\beta} + \sum_{\alpha,\beta \in N_a} \left[\left(\mu_{\alpha\beta}(i,j)\right)\left(s(t)_{i+\alpha,j+\beta}\right) \right] + \sum_{\alpha,\beta \in N_d} \left[\left(\mu_{\alpha\beta}(i,j)\right)\left(s(t)_{i+\alpha,j+\beta}\right) \right] \right)$$

-g is a discretization function

- $\mu_{\alpha\beta}(i,j)$ is a function of wind $\left(\omega_{\alpha\beta}(i,j)\right)$, topography $\left(h_{\alpha\beta}(i,j)\right)$ and fire spread rate $\left(r_{\alpha\beta}(i,j)\right)$.

- $$\mu_{\alpha\beta}(i,j) = \left(\omega_{\alpha\beta}(i,j)\right)\left(h_{\alpha\beta}(i,j)\right)\left(r_{\alpha\beta}(i,j)\right)$$

Testing and Numerical Experiments

- I will investigate different initial conditions, i.e starting position of fire and how much of the forest is already burning.
- I intend to vary the spread rate as well as the spread tendency (how likely the neighboring cell is to ignite)
- To test my model, I will plot graphs and compare with forest fire data.

Project Timeline

Dates	Planned Work
October 22nd-October 25th	Basic research and Mathematical Formulation
October 26th-November 1st	Design Code
November 2nd-November 12th	Implement and Test Code
November 13th-November 19th	Run Experiments and begin report
November 20th- November 25th	Finish Report
November 26th	Edit and Practice Presentation
November 27th	Present project
November 30th	Hand in report

References

- Hernandezencinas, A., L. Hernandezencinas, S. Hoyawhite, A. Martindelrey, and G. Rodriguezsanchez. "Simulation of Forest Fire Fronts Using Cellular Automata." *Advances in Engineering Software 38.6 (2007): 372-78. Print.*

- Bodrozic, Ljiljana, Darko Stipaniccv, and Marijo Scric. "Forest Fires Spread Modelling Using Cellular Automata." *Department for Modelling and Intelligent Systems: University of Split, Croatia (n.d.): n. pag. Web. 2006.*

Equilibrium configuration of N identical charges on the surface of a sphere

Claire Curran

OVERVIEW

- **Simulating what will happen when you put N charges on a sphere**
- **Charges will be of same charge making them repel each other**
- **They will find a new equilibrium each time another one is added.**

GOALS

- Learn how to do a simulation in MATLAB using FDA's and through coding
- Complete my first term project

How?

- Use FDA's and physics laws

$$F = k\frac{q_a q_b}{r^2}$$

Dates	What I SHOULD be doing
October 29 – November 4	Designing Code
November 5 – November 11	Testing/Debugging
November 12 – November 18	Analyzing Code/ Create Outline for Report
November 19 – November 25	Write Report and Create Presentation
November 26 – November 30	Present and BE DONE ☺

Simulation of the Motion of N Interacting Particles in Three Dimensions

PHYSICS 210 PROJECT PROPOSAL

TERESA FABBRICINO

Overview

- The N-body problem involves predicting the motion of N particles due to gravitational interactions.
- Given any initial velocities and positions, after any time t, the velocities and positions can be determined

Project Goals

- Create a MATLAB code to describe the interaction of N particles due to gravitational forces
- Visually simulate these interactions
- Test several initial conditions and values for n
- Analyze energy and momentum conservation

Equations of Motion

Gravitational interaction between 2 particles

$$F = -G\frac{mM(r - R)}{|r - R|^3}$$

Newton's Second Law of Motion

$$F_i = m_i\left(\frac{d^2\vec{x}}{dt^2}\right)$$

Numerical Approach

- Using finite difference approximations

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

With Δt small and $\vec{v}_o$ being the velocity from the previous step

If particles are within a certain range (radius) of each other, gravitational force will grow arbitrarily large. Therefore a softening factor is needed to decrease error and more accurately interpret results.

Testing and Numerical Experiments

Testing

- Calculate energy and momentum of system in each successive loop for each particle

Numerical Experiments

- Investigate several initial positions and velocities, hence varying energy of the system and exploring the different interactions depending on initial conditions
- Explore symmetry of system

Project Timeline

Dates	Activities
October 21-27	Research Mathematical Form and Design Code
October 27-November 4	Design and Implement Code
November 4-10	Test Code
November 10-15	Run Numerical Experiments/Begin Presentation and Report
November 15- 20	Analyze Data, Continue Report
November 20-24	Finish Presentation and complete Report
November 27	Present Project!
November 30	Submit Report!

References

- http://en.wikipedia.org/wiki/N-body_problem
- http://www.math.ubc.ca/~feldman/demos/demo1.html
- http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-
- 97to98/bryan/page1.html
- http://musr.physics.ubc.ca/p210/projects/2006/Lougheed-3body.pdf

Artificial Neural Network

A Phys210 Term Project Proposal

Philip Henderson

Oct.23/2012

Project Goals:

Writing MATLAB (octave) code that simulates a neural network

By using an Energy Function we can use an Ising Model to represent the neural network

Use/calculate the interaction energies between neurons to determine the most stable state. In our case this will be a recognizable letter.

To employ the Monte Carlo Procedure to go through all of the inputs on the lattice (either + or -) , and for each one, determine if flipping it will bring the system closer to the most stable state.

Example of an ideal stable state on a 10 by 10 grid:

0 0 0 0 **1 1** 0 0 0 0

0 0 0 **1 1 1 1** 0 0 0

0 0 **1 1** 0 0 **1 1** 0 0

0 **1 1** 0 0 0 0 **1 1** 0

0 **1 1 1 1 1 1 1 1** 0

1 1 1 1 1 1 1 1 1 1

1 1 1 0 0 0 0 **1 1 1**

1 1 0 0 0 0 0 0 **1 1**

1 1 0 0 0 0 0 0 **1 1**

1 1 0 0 0 0 0 0 **1 1**

Proposed Schedule:

Dates	Goals
Oct 23	Proposal Presentation
Oct 25 - Nov 1	Research & Design Code
Nov 1 - Nov 5	Implement Code
Nov 5 - Nov 10	Test Code
Nov 10 - Nov 20	Run Numerical Experiments
Nov 20 - 26	Analyze Data
Nov 20 - 26	Summarize Project and Finish Presentation
Nov 27 - 29	Presentation of Completed Project
Nov 30	Written Report and Source Code ready to submit

Mathematical Approach:

Effective Energy of our Spin System:

$E = -\sum_{i,j} J_{i,j} s_i s_j$; Where the sum of synaptic inputs to neuron *i* is $\sum_j J_{i,j}\, s_j$ and $J_{i,j}$ is the interaction energy of neurons.

The Distance Between Two Patterns:

$\Delta_{m,n} = 1/N \sum_I [s_i(m) - s_i(n)]^2$; where m and n are two different patterns (For example Q and O are different patterns but look similar), $s_i(m)$ is the configuration of spin i in the pattern m, and N is the total number of spins.

Energy Minima for our Spin Configuration (The most stable state will be the ideal pattern of the character/letter/number. It tells us how to store a single pattern):

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

Energy to store many patterns:

$J_{i,j} = 1/M * \sum_m s_i(m)s_j(m)$; Where M is the total number of patterns, and the summation index m refers to the stored patterns.

Testing and Numerical Approach:

Use an Ising Model, where the states of each pixel can be either +1 or -1, with 100 by 100 grid resolution

The Monte Carlo Procedure brings our system closer to the most stable state with each iteration and get it to recall the correct pattern

Try input patterns with a high amount of randomness and see if it can still recall the correct pattern. Essentially, test the limits of the memory and ability to recall stored patterns

Investigate further applications of this model based off of these initial conclusions

References:

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

-http://en.wikipedia.org/wiki/Monte_Carlo_Methods

-http://en.wikipedia.org/wiki/Ising_model

GRAVITATIONAL INTERACTION OF N-CELESTIAL BODIES USING FINITE DIFFERENCE APPROXIMATION

PHYS 210 TERM PROJECT PROPOSAL

TAM HO

OCTOBER 23, 2012

1. Overview:
   _Using the relationship between defined properties of a celestial body, we can calculate the gravitational force exerted by two particles on each other.

   _All bodies need to have positive mass for the equation to work.

   _We can then do the same thing for N number of particles, which will be specified by the user at the beginning of each simulation.

   _Will write an octave code describing all the processes needed for the simulation.

2. Equation:

$$F_{gi} = \frac{G * m_i * m_j}{R^2}$$

   _Equation for gravitation force between two particles.

   _G is gravitational constant, not changed for all particles where

$$G = 6.67300 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$$

   _For F of i, there will be two particles with mass m of i and m of j interacting with each other.

   _For n particles, there will be n*(n-1) forces, so with F, i goes from 1 to n*(n-1).

   _R is the distance between the center of two bodies.

_Force lead to acceletation, from then will determine position of each bodies after a certain time.

3. Finite Differencial Approximation:

   _Simulation is done based on gridline model, like pixels.

   _Each calculations result in a different positions value of each bodies over time.

   _Delta X, the distance between gridlines, will the precision of the stimulation, decreases as Delta t, the distance between time values, decreases.

   _Delta X decreases as R distance value decreases.

   _However, as precision value increases exceed certain point, such as the accuracy and uncertainty values of the constant G, and the mass and distance measurements, the accuracy decreases significantly, so need to beware.

4. Code testing:

   _Try out different values of each variables to make sure theres no bug that would terminate the simulation abruptly.

   _Look at general projections and movements of the particles, make sure there is no sudden 'jump' in positions out of uncertainty scales.

5. Project timeline:

23/10 Presentation of Project Proposals

24/10 – 30/10 Addition Researching

30/10-10/11 Code Writeup, Discussions, Questions

10/11-20/11 Finishup code, Graphical representation

20/11 – 25/11 Finishup project, Testing, Debugging, Discussions

25/11-30/11 Complete project, formatting, discussions

30/11 Handin project

30/11 VACATIONS!!!

ANY QUESTIONS?

COMMENTS?

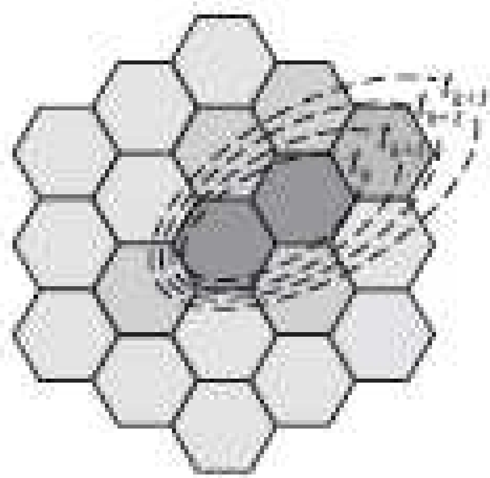
COMPLAINTS?

THANK YOU


Forest Fire
By: Jim Hou

Overview

- -Create a 2D cellular automata with hexagonal cells.
- -Forests have a flat surface thus, slope is not a factor.
- Winds



Project goals

- -To write codes/procedures in Matlab to simulate the spread of fires using 2D cellular automata.
- -To study the spread along with the wind factor.
- -To study different spread rates given different initial conditions.

Mathematical Formula

- $\phi = \phi_w$
- $\phi_w = A(v\ a_w)^B(\beta/\beta_{cp})^{-C}$
- $R_{max}=R_o(1+\phi)$
- $R_0=I_R\ \xi\ /(\rho_b h Q_{ig})$

Numerical Approach

- -The position of the fire will be set specifically or randomly in the forest.

- -Using gnuplot to plot graphs and provide diagrams to show surface fires.

- - Determine the number of time steps to burn a given factor or forest areas.

Testing and Numerical Experiments

- -Examine the spread of forest fire and compare with real surface fire.
- -Examine the spread given different initial conditions
- - The time it takes for fires to stop burning given certain spread rates.
- Use graphs to explain

Timeline

Dates	Activities
Oct 22 - Oct27	Research more about forest fires (different types of fires). Find equations.
Oct 28- Nov10	Implement and test codes
Nov11- Nov17	Run experiments, start on the report.
Nov-18- Nov24	Analyze the data, continue to work on report.
Nov-25- Nov26	Extract important info from the report and make a condensed presentation.
Nov. 27 - Nov29	Give presentation, and finish lab report
Nov.30	Submit Lab report

References

- http://en.wikipedia.org/wiki/Wildfire
- http://www.iemss.org/iemss2006/papers/s11/315_Spataro_1.pdf
- http://www.forestencyclopedia.net/p/p478
- http://rosettacode.org/wiki/Forest_fire#Text
- http://www.sciencedirect.com/science/article/pii/S0307904X06000916

N-Body Simulation

PHYS 210 Term Project Proposal

Richard Hsia

October 23, 2012

N-Body Problem

- *N*-body problems involve the calculations of predicting the motions *n* objects
- These objects may interact with each other gravitationally, due to electric force, or some other mysterious phenomenon

N-Body Problem cont'd

- According to Wikipedia, 2-body problems have been completely solved, and 3-body solutions exist for special cases
- Thus it seems that calculations involving more bodies are too approximate to find accurate solutions

Project Goals

- Write a code in Octave which effectively solves the 3-body (or maybe more) problem and perhaps present the simulation using a short animation
- Use the same code and input different parameters to also obtain valid results

Personal Learning Goals

- Gain a deeper understanding regarding *n*-body problems
- Successfully write code in Octave and sufficiently test the code to make sure there will be no bugs
- Perhaps also learn how to use my results to make a short animation

Approach 1

- There is a theoretical expression to solve the *n*-body problem called **“The n-body problem by Taylor series”**. It is an implementation of the power series. It looks very complicated, but it seems to be the easiest approach, so I will try this approach first.

Approach 2

- A second approach is called the **“Particle-Mesh Method”**. This approach is used for computing gravitational potential. There is also a similar method called the **“Particle-Particle Method”**.

Project Timeline

Dates	Activities
10/21-10/27	Term project proposal and basic research
10/28-11/03	Design and implement code
11/04-11/10	Test and re-design code
11/11-11/17	Run experiments, gather and analyze data
11/18-11/24	Write report
11/24-11/30	11/27 Presentation; 11/28-11/29 Polish report; 11/30 Hand in project

- References
  - http://en.wikipedia.org/wiki/N-body_problem
  - http://www.amara.com/papers/nbody.html#pm
  - http://people.ee.duke.edu/~ayt/ewaldpaper/node15.html

The N-Body Problem

Ling Wai Ada Ip

Term Project Proposal

Project Overview:

- Particle interaction with other particles and the prediction of the motion of these objects which are undergoing mutual gravitation (Coulomb forces)

Goals ☺ :

- To write a program using MATLAB (octave) that models N particles at a random initial position and coming together under gravitational forces
- Setting different number N of particles
- Test the results

Mathematical Interpretation/ Analysis

- Gravitational force derived by Newton:

$$F_{grav} = \frac{Gm_1 m_2}{r^2} \qquad \vec{F}_1 = \frac{Gm_1 m_2}{r^2}\hat{r}$$

where m_1 & m_2 = two objects, r= distance between them, G= gravitational constant = 6.67300 × 10^{-11} m^3 kg^{-1} s^{-2}, unit vector r

- Knowing Newton's second law F=ma,

$$\vec{a} = \frac{Gm_2}{r^2}\hat{r} \qquad \frac{d^2\vec{x}}{dt^2} = \frac{Gm_2}{r^2}\hat{r}$$

Now in terms of the position of a particle at certain time t

$$\frac{d^2\vec{x}_i}{dt^2} = \sum_{j=1}^{n} \frac{Gm_j}{r^2}\hat{r}$$

$$= \sum_{j=1}^{n} \frac{Gm_j}{||\vec{x}_i - \vec{x}_j||^3}(\vec{x}_i - \vec{x}_j)$$

Numerical analysis

- Taylor Series will be implemented to solve these differential equations (position and velocity at time t)

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}h^2 y''(t_0) + O(h^3).$$

where h is the size of each step

- Taking the differential (in order to find state at a later time t):

$$\frac{d\vec{x}_i}{dt}(t+a) = \frac{d\vec{x}_i}{dt}(t) + a * \frac{d^2\vec{x}_i}{dt^2}(t)$$

- Combining these two equations, we achieve:

$$\vec{x}_i(t+h) = \vec{x}_i(t) + h * \frac{d\vec{x}_i}{dt}(t) + \frac{h^2}{2} * \frac{d^2\vec{x}_i}{dt^2}(t)$$

- There is an error with this numerical analysis

Testing:

- Test results by comparing the solution to the N=1 and N=2 body problems

Numerical Experiment:

- Investigate the system using different initial states
- Investigate the system using different number N of particles

Project Timetable:

Date:	What should be done:
Oct 18- 25	Research and design codes
Oct 26- Nov 2	Implement codes
Nov 3- 10	Test codes and start on final report and presentation
Nov 11- 18	Analyze data and continue to work on report and presentation
Nov 19- 27	Be ready for presentation and continue working on report
Nov 27/29	Presentation of project
Nov 28-29	Complete final draft of report
Nov 30	Hand in completed and final draft of project

References:

- http://www.cs.hut.fi/~ctl/NBody.pdf
- http://mit.edu/pkrein/Public/Final%20Paper%20UW324.pdf
- http://en.wikipedia.org/wiki/Euler_method

PHYS 210 – Term Project Proposal

Toomre Model of Collisions of Galaxies

Jordon Johnson
October 23, 2012

Overview

The Toomre model simplifies galactic collisions using the assumptions that the mass of individual stars is negligible, and that Newtonian mechanics are adequate

This approach reduces the number of interactions calculated while still giving a good basic approximation of structures formed during galactic collisions

Overview

The Toomre simulations were able to produce tails such as those in the Antennae galaxies and The Mice

Project Goals

Create a set of MATLAB (octave) scripts that use the Toomre assumptions to simulate the collision of two spiral galaxies

Simulate for several initial conditions to see which, if any, is more likely to result in the formation of an elliptical galaxy

If time allows, improve the model

Equations to be Used

Newton

$$\vec{a} = \sum \frac{Gm_i}{\left|\vec{r}_i\right|^2} \hat{r}_i$$

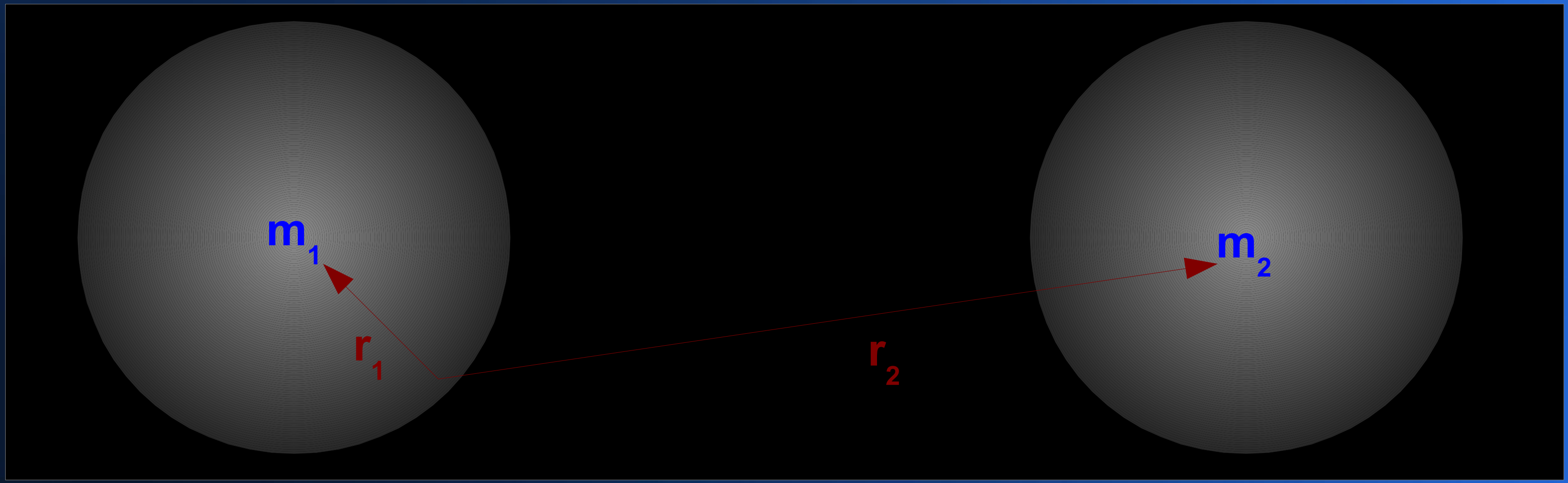
Kepler

$$P^2 = \frac{4\pi^2}{Gm_i} a^3$$

Numerical Approach

Finite Difference Approximation

As Toomre, assume each galaxy consists of a point containing the total mass of the galaxy, with the stars having negligible mass



Numerical Approach

Initially, assume a flat disk and no dark matter

If time allows, “thicken” the disk, and account (roughly) for the presence of dark matter in the galactic disks; see how it changes the results

Timeline

October 22 – 27	Complete research; begin writing pseudocode for modelling and testing scripts
October 29 – November 10	Complete pseudocode; code scripts and test
November 12 – 17	Run simulations; begin analysis
November 19 – 24	Begin writing report and creating presentation
November 27th	Present findings
November 30th	Submit project for evaluation

References

Carroll & Ostlie. An Introduction to Modern Astrophysics. New York: Addison-Wesley, 1996.

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

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

http://asd.gsfc.nasa.gov/archive/sm3b/art/image_gal/stills/post_miss/ast_mice.jpg

Questions?

RAY TRACING THROUGH A SERIES OF MIRRORS, LENSES, AND PRISMS

2012 TERM PROJECT

JACKIE KORN

OCT. 2012

OVERVIEW

- As light travels, it interacts with each medium it travels through, and each surface it hits.
- For objects like mirrors, and mediums such as prisms and lenses, each single interaction can be predicted with simple formulas.
- However, for a complicated series of these optical components, the calculations would be quite extensive.

PROJECT GOALS

- To have a visual representation in Matlab (Octave) of the the results of the calculations for light rays over time.
- To have initial conditions (mirror, lens, and prism configuration) that can be changed, in order to test many situations for the path of light over time.
- To observe the change in direction, focus, and colour for light rays as they interact with optical components.

MATHEMATICAL FORMULATION

- For any mirror, the angle that a light ray hits it (with respect to the normal) will be equal to the angle that the ray reflects. $\theta i = \theta r$
- When light travels through a medium other than a vacuum, its velocity changes (slows down). This change in velocity affects its direction. This can be shown with: $n1 sin(\theta 1) = n2 sin(\theta 2)$ *(Law of Refraction)*
  - Where n1 and n2 are the indices of refraction.
- For lenses and mirrors, the resulting placement and formation of the light rays can be found using variations of the simple formulae:

  $1/do+1/di=1/f$ *and* $M = hi/ho = -di/do$
  - This is assuming a thin lens, where the focal length is much greater than the thickness of the lens

APPROACH

- For each optical component, I will have Matlab calculate the angle and position of the resulting ray, and input that data into the calculation of the next component.
  - Using the formula for refraction of light, image distance, and image size, the initial conditions for the next component can be found.
- For prisms, I will use the law of refraction to find the resulting colours of the light ray.
  - Each wavelength of light is changed by the index of refraction differently ($\lambda o = \lambda / n$), and therefore will emerge travelling at a different angle.
  - The light will be split up into separate colours, whose paths can be calculated and followed as they continue through the series.

TIMELINE

DATE	TO DO
Oct 22 – Oct 28	Research on writing the code
Oct 29 – Nov 4	Begin writing source code
Nov 5 – Nov 11	Finish code and begin testing
Nov 12 – Nov 18	Test code and start written report
Nov 19 – Nov 30	Finalize project and report; test and fix any issues.

References: http://www.wikipedia.org/
Past presentations.
Prior physics courses.

THE END

The Toomre Model for Galaxy Collisions

Physics 210 Term Project Proposal

Michelle Kunimoto

October 23, 2012

Image: http://comosr.spps.org/Astronomy_Meteorology.html

Project Overview

- As galaxies approach each other, the gravitational attractions amongst them may become strong enough to cause “collisions”
- Results of these collisions depend on initial velocities, positions, and masses of the interacting galaxies
- Toomre Model refers to the work by Alar and Jüri Toomre in the 1970’s, who created the first computer simulations of galaxy collisions

Project Goals

- To write a MATLAB (Octave) program that recreates the Toomre Model simulations
- To use various initial velocities, masses, star counts, and other conditions to cover a range of different collisions
- To test the code by comparing results with products of actual galaxy interactions

Relevant Equations

- Newton's Law of Gravitation:

$$\vec{F} = \frac{GMm}{r^3}\hat{r}$$

- Acceleration due to gravitational force:

$$\vec{a} = \frac{GM}{r^3}\hat{r}$$

- Kepler's Third Law where m<<M:

$$p = \frac{4\pi R^2}{GM}r^3$$

Numerical Approach

- One galaxy (the "target galaxy") will be motionless and start at the "origin", and the other (the "intruder galaxy") will start a distance away with a given velocity
- Program will start by specifying initial conditions, such as:
  - Number of stars per galaxy
  - Galactic masses
  - Initial intruder location with respect to target
  - Initial intruder velocity
  - Number of time steps
  - More

Numerical Approach (Cont.)

- Stars will have gravitational attraction to both galactic nuclei
  - Since masses of stars are much smaller than masses of galactic nuclei, I will assume the stars have negligible gravitational attraction to other stars
- Both galactic nuclei will have mutual gravitational attraction to each other
- Knowing initial positions of objects, I will use finite difference approximation to iteratively calculate…
  - Accelerations from the net gravitational forces
  - New velocities from accelerations
  - New positions from velocities
  - New forces from positions
  - New accelerations from forces
  - Redo over and over, for all x-,y-, and z-components, and for both stars and nuclei

Numerical Approach (Cont.)

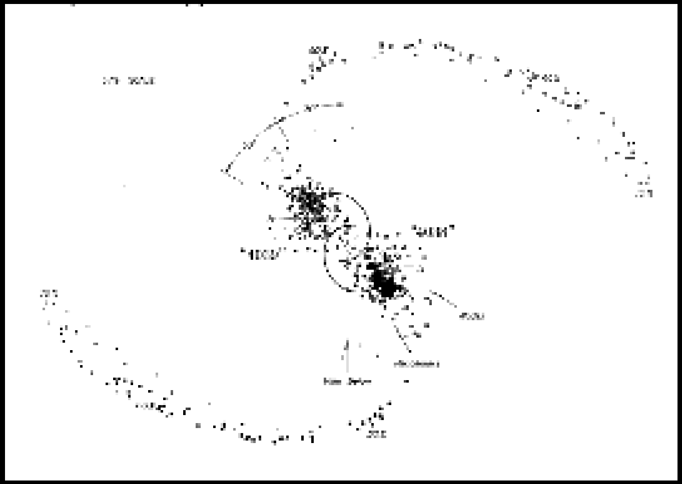
- Still very much a work in progress!
- Lots to work out…

Visualization

- As it stands: I will do everything with Octave
- If this is not viable or if better options are available, I will try other programs

Testing and Numerical Experiments

- I will compare my results with the Toomre Model of the collision of the Antennae galaxies



- Using the same various initial conditions for both, I will compare the results from my program with the results of the galaxy colliding program available on Astronomy lab computers

Images: http://faculty.etsu.edu/smithbj/collisions/collisions.html

Project Timeline

Dates	Activities
10/23-10/29	Do basic research, derive equations, and design code
10/30-11/03	Implement code
11/04-11/09	Test code
11/10-11/15	Run numerical experiments, begin presentation and report
11/16-11/22	Analyze data, continue work on presentation and report
11/23-11/26	Polish presentation and work on final draft of report
11/27	Give presentation
11/27-11/30	Finish final draft of report
11/30	Submit report

References

- Carroll, Bradley and Ostlie, Dale. *An Introduction to Modern Astrophysics, 2nd Edition*. Pearson Education, Inc. (2007)

- Choptuik, Matt. “Finite Difference Solution of the Korteweg & de Vries (KdV) Equation – Physics 210 Term Project Proposal”. <http://bh0.phas.ubc.ca/People/matt/Teaching/09Fall/PHYS210/Doc/term-projects/kdv.pdf>

- <http://en.wikipedia.org/wiki/Alar_Toomre>

- <http://en.wikipedia.org/wiki/Antennae_Galaxies>


Questions? Comments? Suggestions?
Image: http://science.nationalgeographic.com/wallpaper/science/photos/galaxies-gallery/antennae-galaxy/

Toomre Model of Galactic Collisions

Nick Leach

Overview

- Simulation of the collision of two galaxies
- N-Body simulation in three dimensions
- Represented as massive cores with stars in orbit
- No direct stellar collisions
- Implement using FDAs and leapfrog integration

Goals

- Use Octave to create accurate simulation
- Be able to recreate distinctive galaxy pairs
  - E.g. Antennae Galaxies, Mice Galaxies
- Check correctness in two ways
  - Recreate visible galaxies
  - Conserve energy: kinetic and gravitational

Assumptions

- Simplifying assumptions must be made:
- Ignore stellar collisions
- Ignore frictional forces of interstellar dust
- Assume interstellar gravitational forces are negligible
- Ignore general relativistic effects
- All stars are initially in circular orbits

The Maths

- Newton's Laws:

$$F_g = G\frac{m_1 + m_2}{r^2}$$

$$F = m * a$$

- Acceleration of each star:

$$a_i = G\sum_{i=1}^{n}\frac{m_j}{\left|r_i - r_j\right|^2}\hat{r}$$

The Maths (cont'd)

- Each star begins in a stable, circular orbit around its galactic core
  - Kepler's Law:

$$P^2 = \frac{a^3 * 4\pi^2}{G(m_1 * m_2)}$$

Numerical Approach

- Discretize equations of motion using second-order FDA's
  - Form a lattice with respect to x,y,z and t
- Approximate the derivative of

$$a_i = G \sum_{i=1}^{n} \frac{m_j}{\left| r_i - r_j \right|^2} \hat{r}$$

- Leapfrog Integration
  - Alternatively calculate velocities and positions
- Softening parameter to avoid infinite forces

Test and Numerical Experiments

- Conservation of energy
- Replicate actual galaxies
  - Irregular galaxies and notable pairs
- Create stable merges

References

- http://www.cs.hut.fi/~ctl/NBody.pdf
- http://www.browndeertechnology.com/docs/BDT_OpenCL_Tutorial_NBody-rev3.html
- http://arxiv.org/pdf/astro-ph/0011568v1.pdf
- http://www.cs.princeton.edu/courses/archive/fall09/cos126/assignments/nbody.html
- http://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion#Second_law
- http://en.wikipedia.org/wiki/N-body_problem#Mathematical_formulation_of_the_n-body_problem
- http://en.wikipedia.org/wiki/Euler_method
- http://en.wikipedia.org/wiki/Leapfrog_integration
- http://en.wikipedia.org/wiki/Symplectic_integrator
- http://en.wikipedia.org/wiki/Hamilton%27s_equations
- http://en.wikipedia.org/wiki/N-body_simulation

Simulation of Gravitational Interactions of N Particles

Physics 210 Term Project Proposal

Ben Leung

Overview

- The N-body problem predicts the motion of particles interacting with each other due to gravitational forces

Project Goals

- To write a MATLAB code which simulates particles' movement in 2-dimensional space
- To investigate the behavior of the system given various initial conditions
- To test the accuracy of the code by comparing results to known solutions

Mathematical Formulae

Newton's Law of Universal Gravitation:

$$m_j \ddot{\mathbf{q}}_j = G \sum_{k \neq j} \frac{m_j m_k (\mathbf{q}_k - \mathbf{q}_j)}{|\mathbf{q}_k - \mathbf{q}_j|^3}, j = 1, \ldots, n$$

Shows the gravitational attraction between n particles with masses $m_{1..n}$, and positions $q_{1..n}$

Numerical Approach

- By using the acceleration found using Newton's Law of Universal Gravitation, the motion of the particle can be predicted using Euler's Method:

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

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

Given a set of initial conditions, v_o and r_o, and using Δt as the step size.

Testing and Numerical Experiments

Testing:

- Test the method using various step sizes and initial conditions

Numerical Experiments

- Simulate a various of systems with different n sizes and varying masses

Project Organization

- Basic research, derive equations, design code
- Implement code
- Test code
- Run numerical experiments, begin presentation and report
- Analyze data, work on presentation and report
- Polish presentation and work on final draft of report
- Give presentation
- Final draft of report
- Submit report

References

- http://bh0.phas.ubc.ca/People/matt/Teaching/09Fall/PHYS210/Doc/term-projects/kdv.pd
- http://laplace.physics.ubc.ca/210/Proposals-2009/01-ALL.pdf
- http://en.wikipedia.org/wiki/N-body_problem
- http://www.stkate.edu/physics/phys111/EulerMethod.html

Time-dependent Schrödinger equation *(general)*

$$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger's Wave Equation In 1D

Marisa Maharaj

+ Schrodinger's Equation

- Quantum mechanics
- Describes the motion of a particle such as an electron in time. Shows particle/wave nature of matter and light.
- Given in the form of the heat equation (but will have imaginary coefficients)

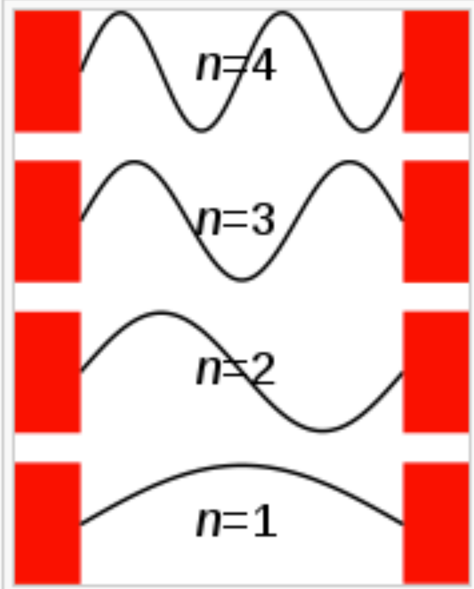
$$\frac{\partial u}{\partial t} = \gamma \, \frac{\partial^2 u}{\partial x^2} \, , \qquad 0 < x < \ell, \qquad t \geq 0,$$

- Will solve the 1D case in an infinite square well/

particle in a box

Time-dependent Schrödinger equation *(single non-relativistic particle)*

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$



Initial wavefunctions for the first four states in a one-dimensional particle in a box

Numerical Methods

- Express derivatives as an expansion (as in forward difference or central difference methods)

$$\frac{\partial u}{\partial t}(t_i, x_j) \approx \frac{u(t_{i+1}, x_j) - u(t_i, x_j)}{\Delta t} + O(\Delta t) \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta t} + O(\Delta t).$$

$$\frac{\partial^2 u}{\partial x^2}(t_i, x_j) \approx \frac{u(t_i, x_{j+1}) - 2\,u(t_i, x_j) + u(t_i, x_{j-1})}{(\Delta x)^2} + O\big((\Delta x)^2\big)$$
$$\approx \frac{u_{i,j+1} - 2\,u_{i,j} + u_{i,j-1}}{(\Delta x)^2} + O\big((\Delta x)^2\big),$$

- Set up equations with derivatives

- Use data at prior time to solve for state of system at a later time. Make $u_{i+1,j}$ subject of the equation

$$u_{i+1,j} = \mu\, u_{i,j+1} + (1 - 2\mu) u_{i,j} + \mu\, u_{i,j-1}, \qquad \begin{array}{l} i = 0, 1, 2, \ldots\,, \\ j = 1, \ldots, n-1. \end{array}$$

Step sizes subject to von Neumann stability condition

$$\mu = \frac{\gamma\,\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

Solve matrix equation

- Using boundary and initial conditions (which will result in vectors) we will get a matrix equation which can be diagonialized to find the solution.

$$\mathbf{u}^{(i+1)} = A\,\mathbf{u}^{(i)} + \mathbf{b}^{(i)}$$

$$A = \begin{pmatrix} 1-2\mu & \mu & & & & \\ \mu & 1-2\mu & \mu & & & \\ & \mu & 1-2\mu & \mu & & \\ & & \mu & \ddots & \ddots & \\ & & & \ddots & \ddots & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}, \qquad \mathbf{b}^{(i)} = \begin{pmatrix} \mu\,\alpha_i \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mu\,\beta_i \end{pmatrix}$$

Will use Crank-Nicholson method to solve

- I have not checked how using imaginary numbers will work in this, I will have to make sure they work here.

- Thank you!

N-Body Simulation of Gravitationally interacting Particles

PHYS 210 Term Project Proposal

Shuchita Mahey
October 23, 2012

Overview

- The N-Body simulation is a model used to predict the motion of N particles in space which are interacting with each other through a force, such as the Gravitational force
- Given the initial positions and velocities of the particles, how the entire system will evolve with time can be predicted

Project Goals

- Write a MATLAB code to simulate the N-Body problem and display the results visually
- Test and debug as necessary

Mathematical Model

- The only force acting on each particle is Gravitational force, which is calculated as follows:

$$F = \frac{GM_1M_2}{r^2}$$

- The net force acting on one particle is the sum of all the gravitational forces exerted on it by the other particles
- Each particle will have an initial velocity and position, which will be repeated updated as the particles move around
- This will be done using Finite Difference Approximations

Project Timeline

- Summarize the special features and advantages of the product being introduced

Oct 22 – Oct 28	Basic Research and Design Code
Oct 29 – Nov 11	Write Code
Nov 11 – Nov 18	Run Experiments, Analyze Data, Debug
Nov 18 – Nov 29	Work on Report, Continue analyzing, prepare Presentation
Nov 29	Presentation
Nov 30	Submit Report

References

- http://laplace.physics.ubc.ca/210/Proposals-2009/02-ALL.pdf
- http://en.wikipedia.org/wiki/N-body_problem

Gravitational N-body

PHYS 210 Term Project Proposal
Jakob Manning

Overview

Problem to be solved: How can the gravitational interactions of many particles be modeled on a two dimensional Cartesian coordinate system?

Program used: Maple (Octave)

Known: Initial positions, masses, velocities.

Simplifications: Model each component as a point mass moving through 2 dimensions.

The Math

The force and resulting acceleration of a particle can be expressed with these two fundamental equations.

$$\vec{F} = \frac{Gm_1m_2}{r^3}\vec{r}$$

F = ma

Example of Calculations for

The acceleration experienced by Object 'i' can be written in the form

$$\frac{d\vec{v}_i}{dt} = G\sum\frac{m_j}{{r_{ij}}^3}\cdot\vec{r}_{ij}$$

Numerical Approach

Use finite difference approximations

Repeatedly evaluate how gravitational forces changes the velocity (and position) of each point mass, assuming that the acceleration and velocity are constant over small spans of time.

Project Timeline

Due Date	Work
October 28	Research, formulate equations, design code
November 4	Implement code
November 11	Test code
November 16	Run experiments, begin presentation and report
November 22	Analyze results, continue presentation and report
November 26	Final draft of presentation
November 27	Give presentation
November 29	Final draft of report
November 30	Submit report

Numerical Solution to Gravitational N-Body Problem

Ashkan Memarian

Oct 25th 2012

Overview

- Overview
  - The gravitational interaction of N massive objects is governed by Newton's second law with the Newton's gravitational force acting between any two objects.
  - A two body problem can be solved analytically but for $N \geq 3$ this problem has no closed form solution.
- Project Goals
  - To write an Octave code which solves the gravitational N-body problem for any number of objects using second order finite difference techniques.
  - To establish correctness of the code via verification with the closed form solutions and other special case approximate solutions.
  - To investigate a variety of initial conditions.

Mathematical Formulation

- The force acting on the jth object is:

$$\vec{F}_{ji}(t) = \sum_{\substack{i=1 \\ j \neq i}}^{N} \frac{GM_j M_i}{r_{ji}^2(t)} \hat{r}_{ji}(t) \tag{1}$$

$$\hat{r}_{ji} = \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|} \tag{2}$$

- The problem will be formulated as an initial boundary problem on the domain:

$$\vec{r}_{min} \leq \vec{r}_j \leq \vec{r}_{max} \qquad t_{min} \leq t \leq t_{max} \tag{3}$$

with initial conditions:

$$\vec{r}_j(t=0) = \vec{r}_j^{\ 0} \tag{4}$$

$$\frac{\mathrm{d}\vec{r}_j(t)}{\mathrm{d}t}|_{t=0} = \dot{\vec{r}}_j^{\ 0} \tag{5}$$

Testing & Numerical Experiments

- Testing
  - The code will be used to generate solution to the N-body problem with fixed initial data and several increasing levels and convergence of the solution will be unsured.
  - Compare the generated solutions with closed form solutions (in the cases where they exist) and ensure the residues are of $O(h^2)$.
- Numerical Experiments
  - Solutions for several various initial and boundary conditions will be investigated.

Project Timeline

Dates	Activities
Oct 28th to Nov 3rd	Do basic research, derive equations and design code
Nov 4th to Nov 10th	Implement code
Nov 11th to Nov 17th	Test code, run numerical experiments, begin presentation and report
Nov 18th to Nov 24th	Analyze data, continue work on presentation and report
Nov 25nd to No 26th	Polish presentation and work on final draft of report
Nov 27th	Give presentation
Nov 28th to Nov 29th	Finish final draft of report
Nov 30th	Submit report

References

- H. Goldstein et al, Classical Mechanics, Addison Wesley, (2002)

Traffic Simulation through Cellular Automata

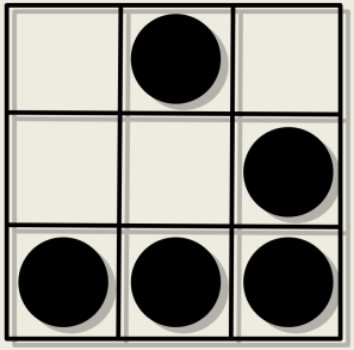
Gavin Noble

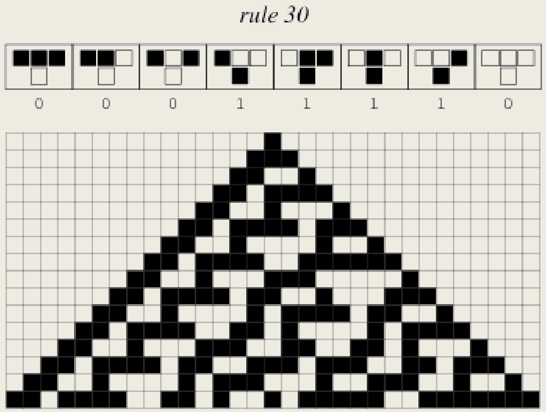
25/10/12

Overview

Cellular automaton?

- collection/grid of discrete “cells” that interact and evolve with one another by deterministic/probabilistic means
- famous examples: Conway’s Life, Wolfram’s 1-d Rules





Traffic simulation?

- cells can be ‘full’ (contain a car) or ‘empty’ (not contain a car)
- full cells inspect their neighboring cells and act accordingly

Goals

- Replicate real-world traffic flow patterns
- Investigate the causes and catalysts of freeway traffic jams (traffic density, speed limits, etc.)

Numeric Approach

The Nagel-Schreckenberg model:

- one freeway lane $\leftrightarrow$ one-dimensional array
- each array element may be empty or non-empty
- all non-empty elements governed by the same 4 rules:
  1) acceleration; if there is room and $v_i < v_{max}$, $v_i \to v_i+1$
  2) deceleration; if there is no room and $v_i > 0$, $v_i \to v_i-1$
  3) randomization; with some probability, $v_i \to v_i \pm 1$
  4) advance vehicle amount proportional to v_i
- manipulate initial density, averaged over the array
- monitor time-averaged site densities and flow-rates

Numeric Approach

The Nagel-Schreckenberg model:

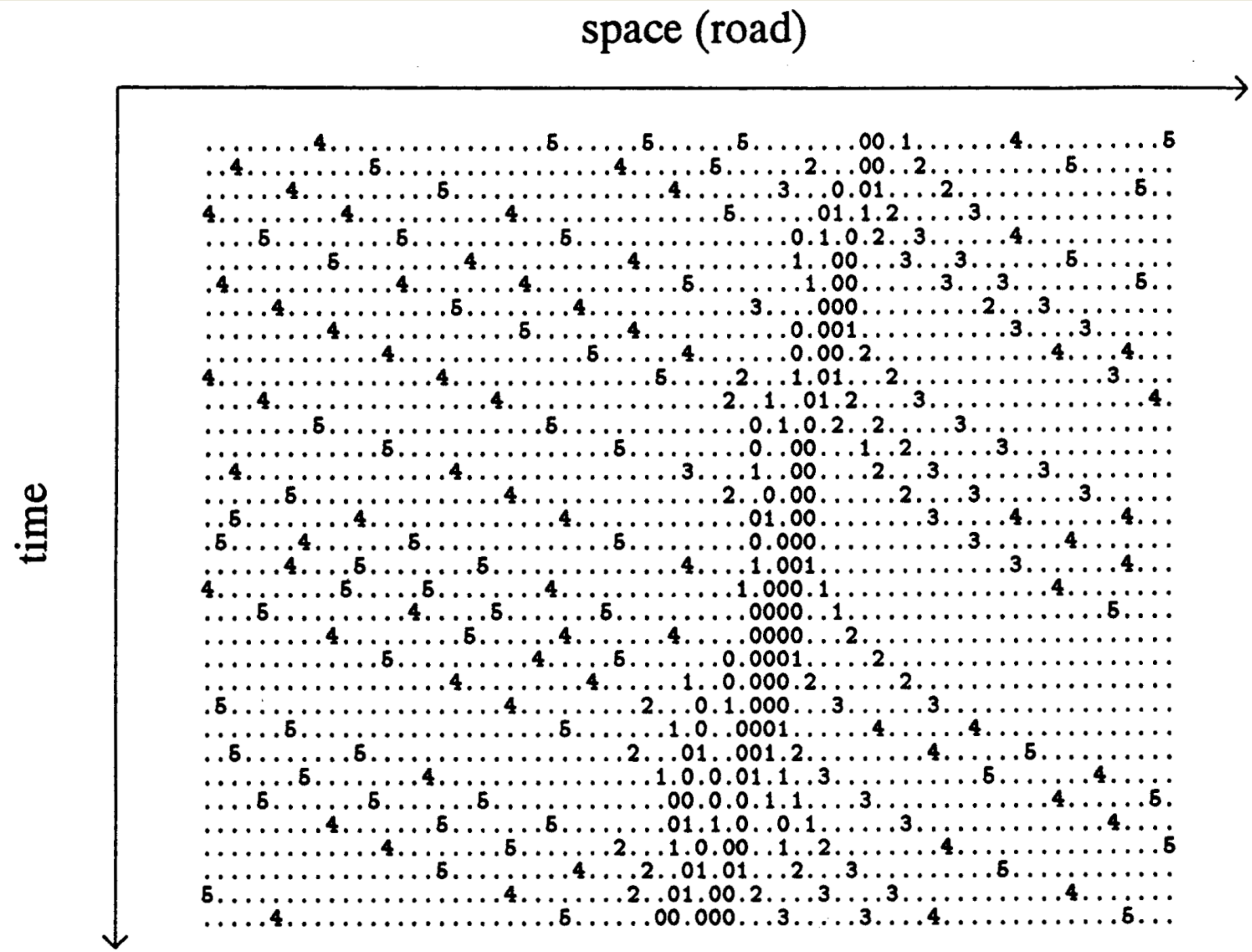
$$\bar{\rho}^T = \frac{1}{T}\sum_{t=t_0+1}^{t_0+T} n_i(t)$$

Where $n_i(t)=1$ if the ith site is occupied, 0 otherwise.

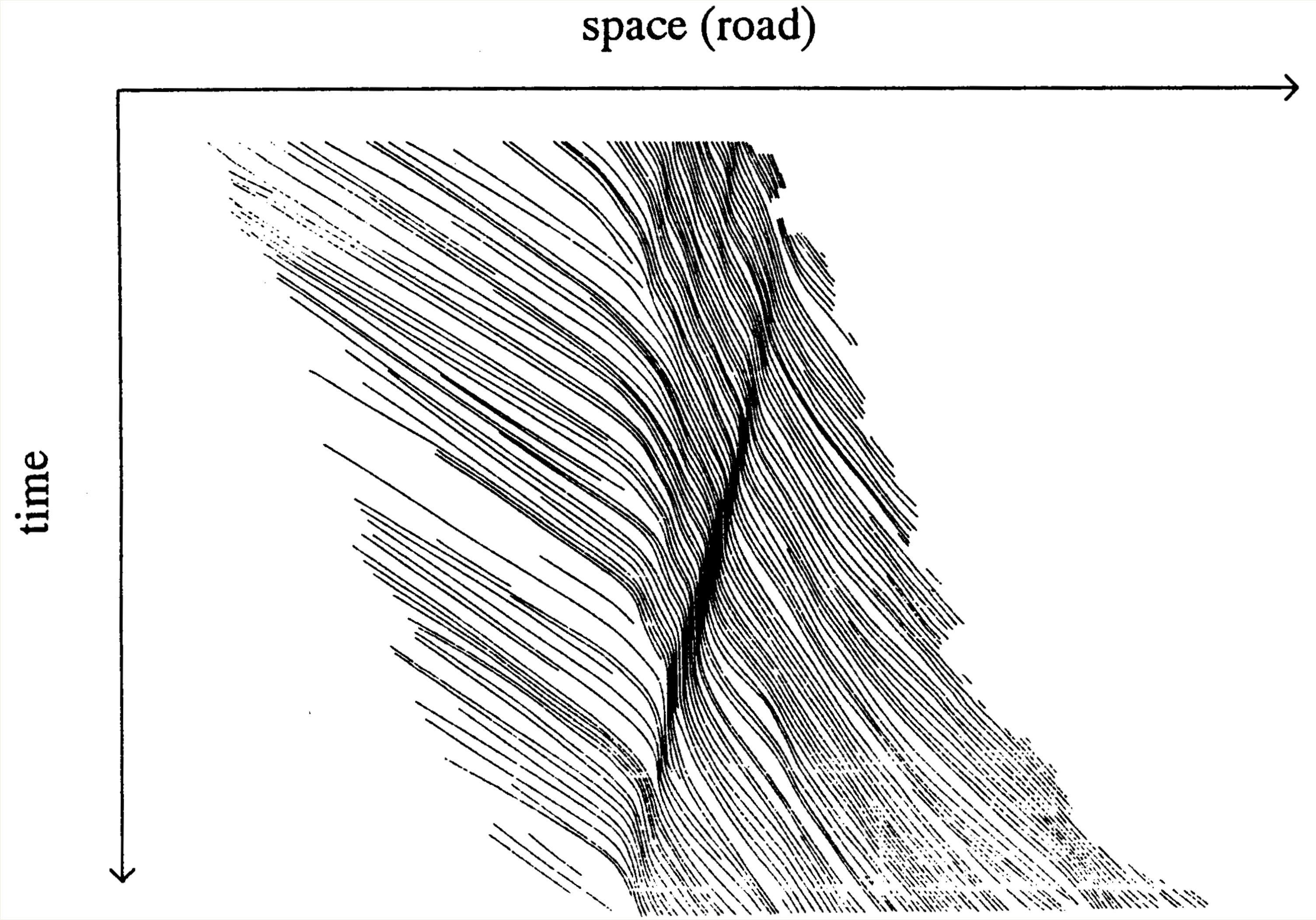
$$\bar{q}^T = \frac{1}{T}\sum_{t=t_0+1}^{t_0+T} n_{i,i+1}(t)$$

Where $n_{i,i+1}(t)=1$ if there is movement between sites i and i+1, 0 otherwise

Experimentation



Experimentation



Experimentation

Extend the system described by Nagel and Schreckenberg to the general case:

- multi-lane roads
- intersecting/forked roads

Quantitative comparison:

- ideal speeds?
- ideal densities?

Timeline

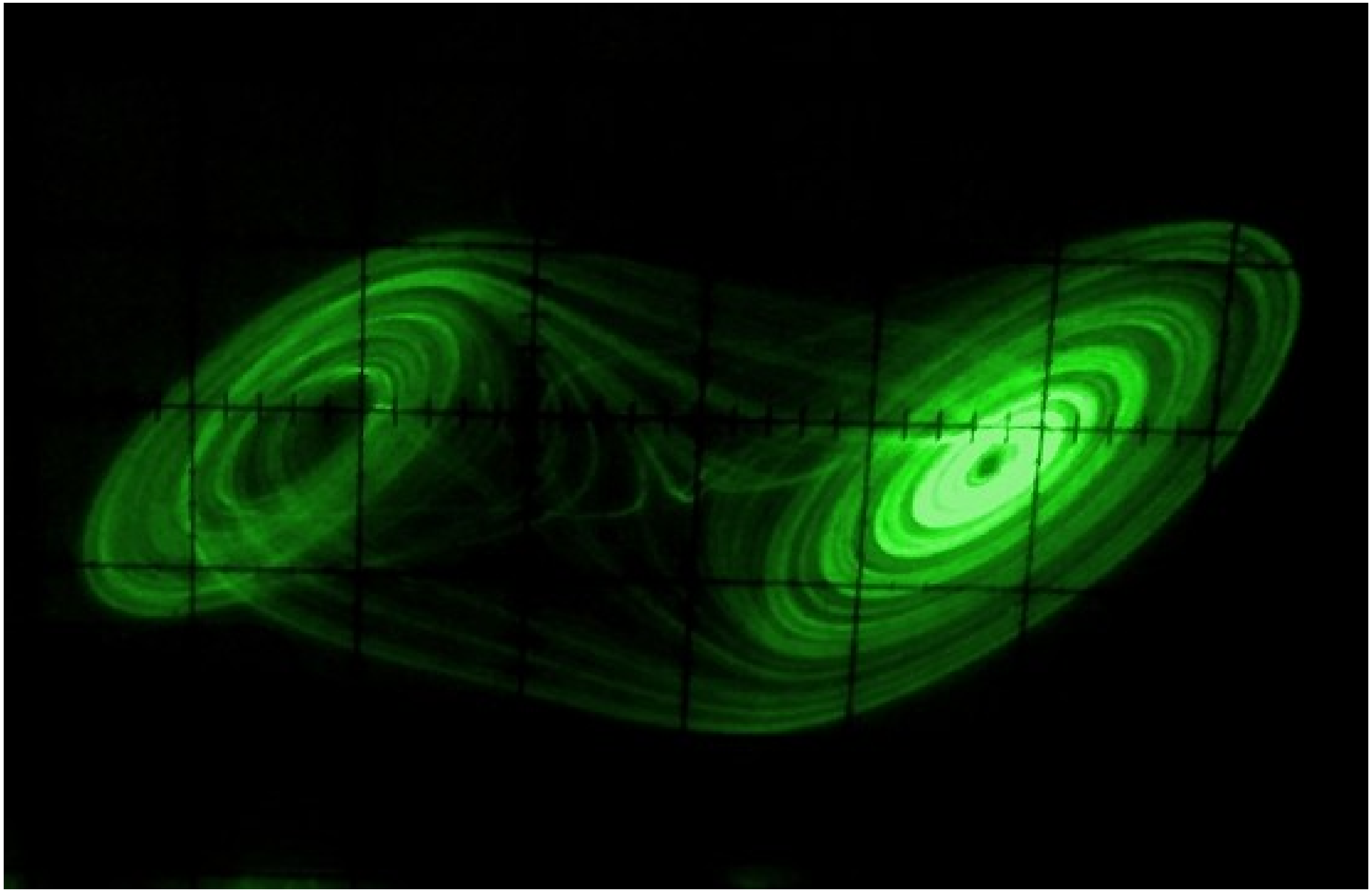
Dates:	Do:
Week 1, Oct 28	Research; Design Code; Preliminary Implementation
Week 2, Nov 04	Test Code; Ensure Basic Functionality; Experimentation
Week 3, Nov 11	Extend Code; More Experimentation; Outline Report
Week 4, Nov 19	Prepare Presentation; Write Report
Week 5, Nov 26	Give Presentation; Polish & Submit Report

References:

- Nagel and Schreckenberg, "A cellular automaton model for freeway traffic" J Phys I France 2 (1002) 2221-2229
- *http://mathworld.wolfram.com/CellularAutomaton.html*

Simulation of the chaotic behavior of Chua's Circuit.

Phys 210 Term Project
Brian Pamandanan
Oct 25, 2012



Project Overview

-Chaotic systems occur thoughout nature in such as phenomena like weather systems, N-body(N > 2) interactions, and population dynamics.

-However, until the invention of Chua's curcuit, there were no known reproducible functional chaotic circuits.

-Chua's circuit provides an easily built and easily simulated model of chaos that can be tweaked to have different chaotic behaviors

Goals of Project

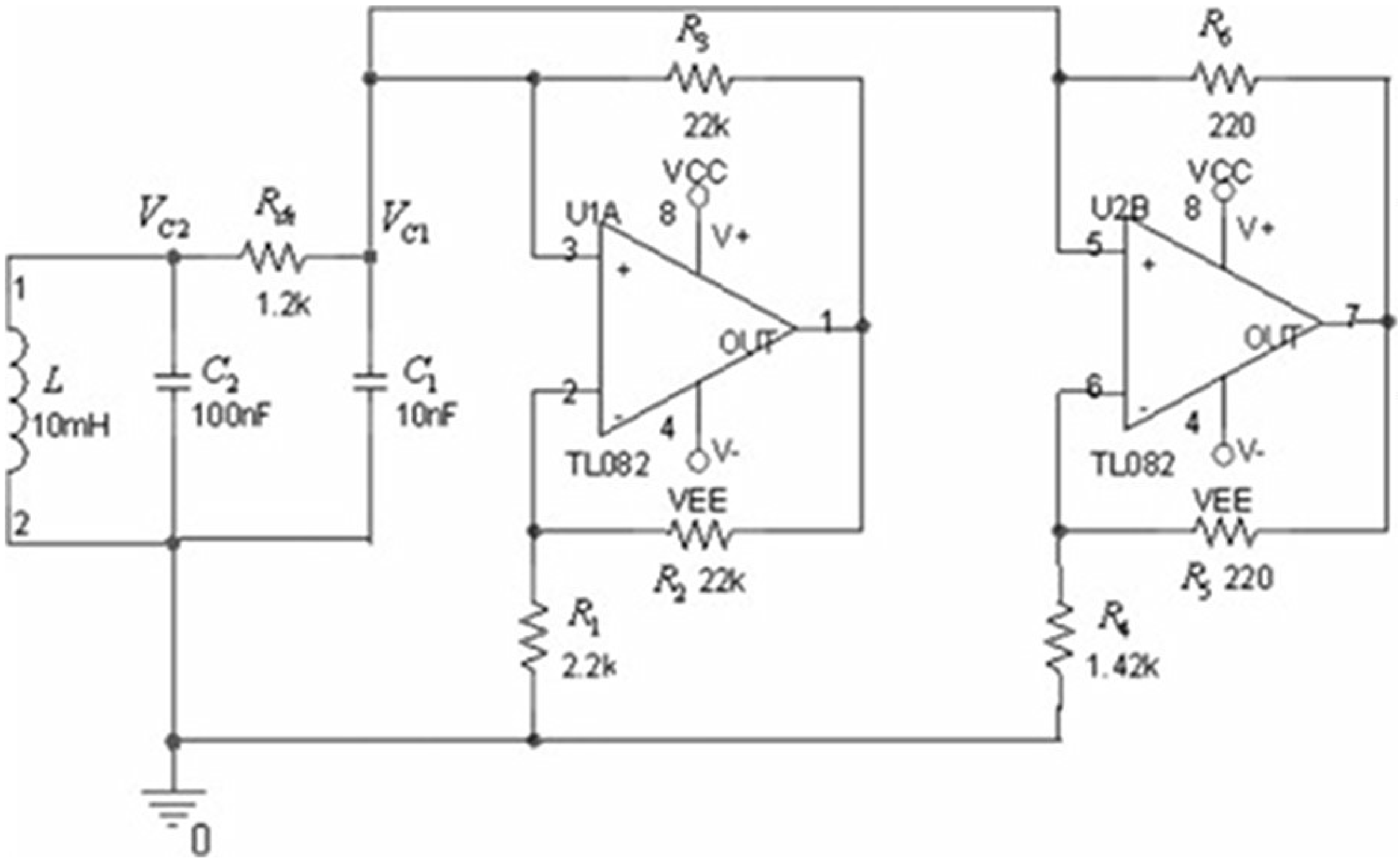
-To create a MATLAB(octave) program that solves numerically the nonlinear ODEs that characterize the original Chua circuit.

-To use this simulation to demonstrate features of chaos including period doubling and sensitivity to initial conditions.

Physical Background and Mathematical Formulation

- The original Chua circuit contains 3 passive linear energy-storage components (2 capacitors and 1 inductor). The voltages across the capacitors and the current across the inductor are the variables that show the behaviour of the circuit.

- It also contains a non-linear, locally active(active over some voltage range), component. This component provides the non-linearity required for chaos and the power for the circuit



R3
22k
R6
220
VCC
8
V+
U1A
3
+
2
-
OUT
1
4
V-
TL082
VEE
R2 22k
R1
2.2k
U2B
5
6
7
R5 220
R4
1.42k
VC2
R4
1.2k
VC1
L
10mH
C2
100nF
C1
10nF
0

- When we apply circuit theory to characterize the behaviour of the voltage through C_1 and C_2 and the current though L, we can get the following (non-dimensional) equations:

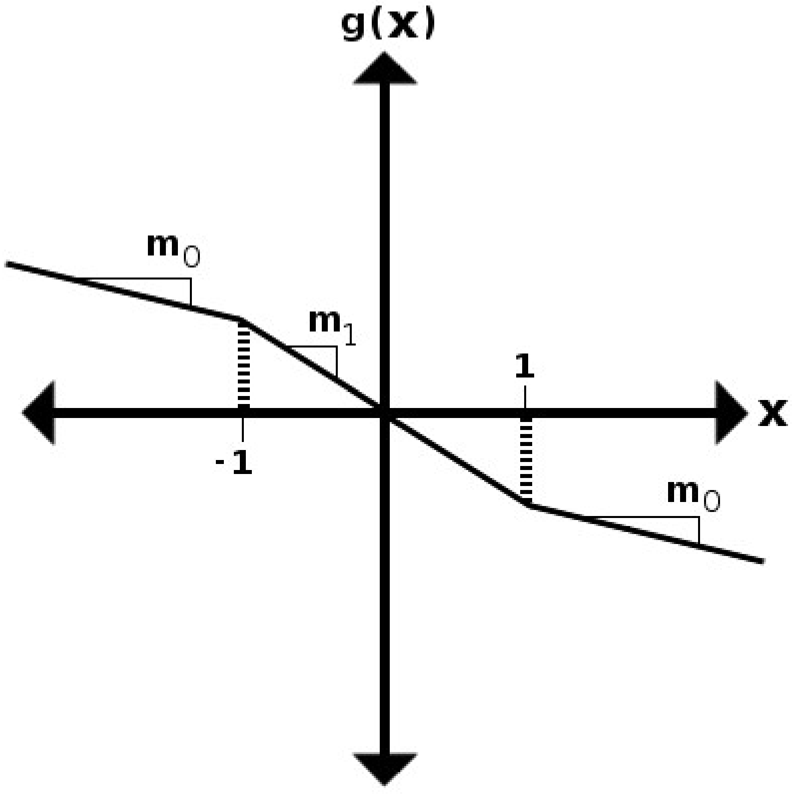
$$\frac{dx}{dt} = \alpha[\, y - x - g(x)\,]$$

$$\frac{dy}{dt} = x - y + z$$

$$\frac{dz}{dt} = -\beta y$$

- The function g(x) encodes the response of nonlinear active component.

$$g(x) = m_0 x + \frac{1}{2}(m_1 - m_0)(|x+1| - |x-1|)$$



- For chaos to occur, m_1 and m_0 must be negative.

Testing & Numerical Approach

- Still unsure of which approximation method to use
- Whatever method I choose, I will test to find a step size that balances speed and accuracy.
- I plan making sure simulation actually works(i.e. Actually displays chaotic behaviour, doesn't crash, etc.)
- I will vary the parameters α, β, $m_{0,}$ and m_1 to find 'regions' of interest. (period doubling, actual chaos)

Tentative Timeline

- Oct 25 – Oct 31: Design and begin implementing code.
- Nov 1 – Nov 9: Implement code and test it.
- Nov 10 – Nov 16: Run experiments and begin writing report.
- Nov 17 – Nov 22: Analyze results and continue report.
- Nov 23 – Nov 26: Prepare presentation and finish report.
- Nov 27: Presentation
- Nov 27 – Nov 30: Polish up report and code.

References

- Chua's circuit, Scholarpedia
- Chua's circuit, Wikipedia
- http://www.chuacircuits.com
- Chua, L. O. (1992) The Genesis of Chua’s Circuit. Archiv für Elektronik und Ubertragung-stechnik, 46 : 250-257.
- Zhiguo Shi, Shaohua Hong, Kangsheng Chen, Experimental study on tracking the state of analog Chua's circuit with particle filter for chaos synchronization, Physics Letters A, Volume 372, Issue 34, 18 August 2008, Pages 5575-5580

The Dispersion of Charges over a Sphere

Timothy Roosma

Goals

- To investigate the dispersion of charges on a sphere beginning from random positions
- To build an Octave code to simulate this process and to visualize it
  - Use finite difference approximation to calculate what will happen
  - Visualize if using MATLAB's built in program

Method

- Use Coulomb's equations for electric force:

$$\boldsymbol{F} = k_e \frac{q_1 q_2 \hat{\mathbf{r}}_{21}}{r_{21}^2}$$

- Drag Force

$$\mathbf{F}_d = -b\mathbf{v}$$

- Use Newton's second law to find the motion of the charges caused by the forces

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

More Equations

- Coulomb's equation now generalized for a system of n charges

$$\boldsymbol{F}(\boldsymbol{r}) = \frac{q}{4\pi\varepsilon_0}\sum_{i=1}^{N} q_i \frac{\boldsymbol{r}-\boldsymbol{r_i}}{|\boldsymbol{r}-\boldsymbol{r_i}|^3} = \frac{q}{4\pi\varepsilon_0}\sum_{i=1}^{N} q_i \frac{\widehat{\boldsymbol{R_i}}}{|\boldsymbol{R_i}|^2},$$

Additional Information

- Charges will have to be restricted to the surface of the sphere
- Different surfaces may be attempted once the code for the dispersion over a sphere is complete and working
- All charges will be identical

Timeline

Dates	Goals
Present – October 31	Further research and designing the basic code
Nov 1 – Nov 6	Testing and debugging
Nov 7 – Nov 8	Testing other surfaces (possibly)
Nov 9 – Nov 16	Take data from simulations and begin working on final report
Nov 17 – Nov 23	Analyze data and continue/finish final report
Nov 24 – Nov 27	Prepare final presentation
Nov 29	Give final Presentation
Nov 30 11:58.59 PM	Submit final report

Thank you

- Questions?
- Suggestions?

Simulation of a Neural Network

Alice Shang

October 25, 2012

Overview

- A neural network, such as a brain, is composed of individual units (neurons) which interact with each other via electrical signals
- The output rate of a neuron is a function of the sum of its inputs (excitatory minus inhibitory)
- Since there is a threshold level for whether a neuron will fire or not, these interactions can be modelled as an on/off system
- The neuron can thus be modelled as an Ising spin, which has two states: either firing (1) or not firing (-1)

Project Goals

- To write a MATLAB code which will:
  - Model a neural network as a lattice of Ising spins
  - Calculate the overall energy of the Ising spin system
  - Include an algorithm to examine the spin of each neuron, and alter it depending on certain conditions (determined by Monte Carlo method)
  - Recognize and recall preset patterns in the lattice when given a 'vaguely similar' pattern
- To use this code to visually simulate a neural network

Visualization

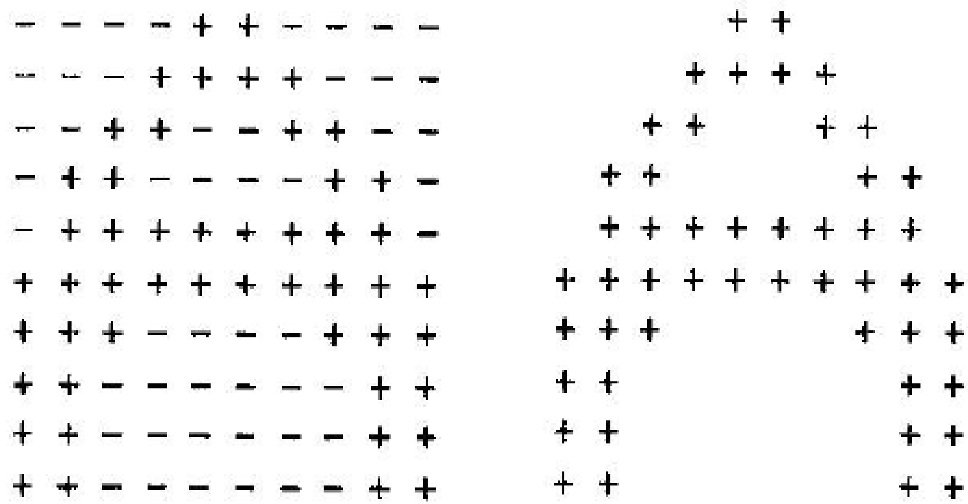


FIGURE 12.24: Left: a 10×10 lattice of spins with a particular configuration of $+$ and $-$ spins; right: the same lattice but with the spins for which $s = -1$ replaced by blanks. This network holds (i.e., displays) the letter A.

Mathematical Formulation

- Energy function

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

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

- $s_i(m)$ and $s_j(m)$ represent the states of neurons (-1 or 1) for a certain pattern (m).
- s_i and s_j are the states of neurons in the present configuration of the lattice

Numerical Approach

- Putting the previous equations together,

$$E(m) = -\sum_{i,j} s_j(m)\, s_i(m)\, s_i\, s_j$$

  - Note that this is for a single pattern; the 1/M term in front of the sum is for M patterns.
- When s_i and s_j are equal to $s_i(m)$ and $s_j(m)$, the lattice is in the correct configuration, and the value of E will be at a minimum (a very negative number)

Numerical Approach

- The Monte Carlo procedure is an algorithm which looks at the value of a spin, then compares it to the calculated value of E.
  - If flipping the spin would make E more negative, then it will be flipped
  - Otherwise, leave the spin alone
  - This procedure goes through as many iterations as it takes to reach a stable state (i.e. all spins are left alone)
- Since the stable state is determined by arbitrary values (a preset pattern), this effectively results in the lattice 'recalling' that pattern

Testing and Numerical Experiments

- Changing a certain percentage of the lattice to have a 'random' spin, then seeing how well the program can retrieve the preset pattern
- Testing the number of patterns that can be stored:
  - The theoretical limit for the number of patterns that can be stored is 0.13*N, where N is the total number of spins
  - If two patterns are very similar, the program is unlikely to be able to distinguish them from one another

Testing and Numerical Experiments

- Simulating a damaged network:
  - Add a chance (a certain probability) for the value of $J_{i,j}$ to be 0 regardless of the spin states, in addition to original alteration of the lattice
  - $J_{i,j}$ is a value which is related to how two spins are connected, so this procedure is analogous to severing the connection between two neurons
- Simulating learning:
  - Add constants and additional parameters to the formula for $J_{i,j}$ to alter the desired stable state with each iteration

Visualization

- Example of recalling the letter 'A' in a 10x10 grid:

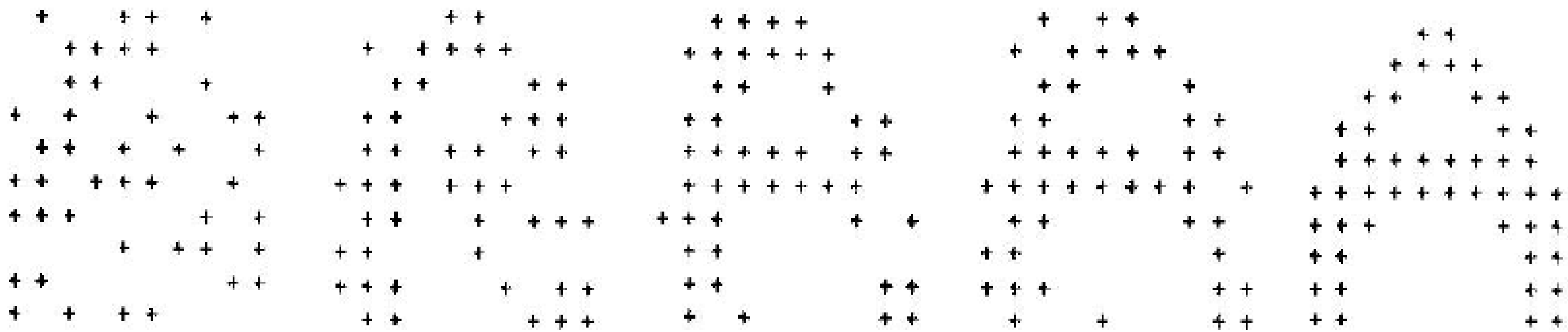


FIGURE 12.32: Operation with 90% of the $J_{i,j}$ set to zero. Left to right: initial pattern and spin configurations after one, two, five, and nine Monte Carlo sweeps through the lattice.

Project Timeline

Dates	Objectives
Oct 25 – Oct 31	Preliminary research, design code
Nov 1 – Nov 5	Implement and test code
Nov 6 – Nov 11	Run experiments and collect data
Nov 12 – Nov 18	Begin presentation preparation, begin report
Nov 18 – Nov 26	Polish presentation, finish report
Nov 27 – Nov 29	Presentation
Nov 30	Hand in report, pat self on back

Questions? Comments? Suggestions?

Motion simulation of N particles with Gravitational interaction

PHYSICS 210 Term project proposal

Garrett Stewart

Summary

Using Euler's approximation to solve the initial value problem of n gravitationally interacting particles.

Project goals

-To design and program a simulation in 2D space with Cartesian co ordinates that predicts any particles motion and position at any point in time
-To numerically approach the n body problem by writing a code for MATLAB that satisfies the initial condition in 3D space.
-To investigate the simulations response as different initial conditions are implemented
-To verify accuracy using conservation of energy

Mathematical Formula

Using Newton's Law for Gravitational motion I find that

$$\overrightarrow{F_{g_i}} = G\sum \frac{m_i m_j}{(r_j - r_i)^{\frac{3}{2}}}(\vec{r_j} - \vec{r_i})$$

This formula will allow the program to calculate the net force on a particle.

The particles position and velocity can be described by the equations

$$\vec{v_i} = \Delta t \cdot \vec{a_i} + \overrightarrow{v_{0_i}}$$

$$\vec{r}_i = \Delta t \cdot \vec{v_i} + \overrightarrow{r_{0}}_i$$

Numerical approach continued

Using Euler's method of approximation we can assume that if I take small enough values for the time interval Δt, that the velocity and acceleration of the particle remain constant for that interval; thus, the equation for motion and position.

Using this information and my randomly generated initial values the simulation can predict the motion and position of a particle after a number of successive loops.

$$\begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix} = G \sum \frac{\begin{bmatrix} r_{xj} \\ r_{yj} \\ r_{zj} \end{bmatrix} - \begin{bmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{bmatrix}}{((r_{xj} - r_{xi})^2 + (r_{yj} - r_{yi})^2 + (r_{zj} - r_{zi})^2)^{3/2}}$$

Testing and numerical experiments

To test the accuracy of this simulation I can calculate the total kinetic and potential energy at any given time and plot that vs time, which will allow me to have a comparison point for the raw data collected.

For the Numerical experiments I will simulate the motion of n number of particles (where n>3) on a 2D surface (ie restricted to a plane).
After this initial experiment I will apply various symmetrical initial conditions to the particles to see if they retain their symmetry, if so the simulation will be successful.

Date	
10/23-10/29	Research on mathematical formula and design code
10/30-11/05	Design Code and implement code
11/6- 11/12	Test code
11/13-11/19	Apply numerical experiments, begin report
11/20-11/26	Analyze data continue work on presentation and report
11/27-11/30	Polish presentation and work on final draft of report
12/01	Present project
12/01-12/04	Present project
10/23-12/04	Drink away the pain of being a physics major

References

-http://www.cs.ubc.ca/~nando/nipsfast/slides/gray.pdf

-http://laplace.phas.ubc.ca/210/Proposals-2009/01-ALL.pdf

-http://en.wikipedia.org/wiki/N-body_simulation

Galaxy Interactions and Collisions

Em Storey

Phys 210 Term Project Proposal

Project Overview

- Galaxies, like all other celestial objects, are affected by the pull of gravity
- It takes an extremely massive object to have sufficient gravity to affect an entire galaxy
- Galaxy “collision” models are a method of modeling and observing the mutual effects that the gravities of two galaxies have on each other.
- Galaxies will primarily be modeled as galactic nuclei in the simulation.

Project Goals

- To write script which simulates the interaction/collision of two galaxies and is able to model the trajectories of two galactic nuclei.
- The script should also be able model the positions and trajectories of stars orbiting one of the galaxies. Because the mass of the galaxies is so much greater than the masses of the stars, the stars modeled are not to affect the galactic nuclei. The stars are added as a method of visualizing the collision.

Mathematical Formulation

- Each particle experiences force which can be described with Newton's Law of Universal Gravitation:

$$F = G\frac{M_1 M_2}{(r_{12})^2} \qquad F = ma \qquad m_1 a_1 = G\frac{m_1 m_2}{(r_2 - r_1)^2}$$

- Components of each particle's motion given by:

$$\vec{v}_f = \vec{v}_i + \vec{a}\Delta t \qquad (x_f, y_f, z_f) = (x_i, y_i, z_i) + \vec{v}\Delta t$$

- Both galactic nuclei will be set with initial positions (x_0, y_0, z_0) and velocities $\langle v_x, v_y, v_z \rangle$ prior to the commencement of the simulation.

Mathematical Formulation

- For simplicity in calculation:
- Stars, when they are present, will all have the same mass, and not be influenced by each other's gravity. They will all have circular orbits. They are subjected to the same forces as the galactic nuclei. Their period and radii of orbit will be represented by Kepler's Law:

$$P^2 = \frac{4\pi^2 a^3}{G(M_1+M_2)}$$

when $M_1 \gg M_2$: $$\frac{P^2}{a^3} = \frac{4\pi^2}{GM_1}$$

Numerical Approach

- Total force felt by a particle is the sum of all forces from both galactic nuclei. Total forces will be calculated using derivatives. These derivatives will then be replaced with FDAs.

- Finite Difference Approximations will model the resulting motions over a grid/lattice.

Testing and Numerical Experiments

- The inputs that can be manipulated will be:
  - Initial Velocity
  - Initial Position
  - Number of Stars
- Different configurations of the above criteria will be tested and observed. Any odd behaviour observed will be researched, and re-tested. Code will be modified as needed if any faults are found.
- Any results that seem odd may have their conditions tested on an existing galaxy collision simulator (compared to a “known solution”)

Timeline

Date	Activities
Oct 25 – Nov 1	Research Equations, Design Code
Nov 1 – Nov 8	Implement Code
Nov 8 – Nov 15	Run Numerical Experiments Begin Presentation & Report
Nov 15 – Nov 22	Data Analysis Polish Presentation & Report
Nov 27	Present Final Result
Nov 30	Submit Term Project

REFERENCES

- **Introduction** to **modern astrophysics** / Bradley W. Carroll, Dale A. Ostlie.
- http://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion

PHYS 210 Term Project Proposal

Will Stratton

Simulation of N Interacting Particles in Three Dimensions

Equilibrium Configuration of N Identical Charges on the Surface of a Sphere

Thomson Problem

- Identical charges on a sphere repel each other until reaching an equilibrium state when the total sum of repulsive (electrostatic potential) forces is minimized

$$f(N) = \sum_{i=1}^{N} \frac{1}{|x_i, x_j|}$$

- x_i is the position of the i^{th} electron

Mathematical Formulation

- Electrostatic Potential Energy

$$U_E = \frac{1}{2}\sum_{i=1}^{N} q_i \Phi(x_i)$$

- $\Phi(x_i)$ is the electrostatic potential due to all other point charges

Goals

- Write an octave code which minimizes the Thomson Problem
- Simulate the interaction of N identical charges on a sphere until equilibrium is reached
- Investigate a variety of N bodies

Timeline

- Week 1: Research, design code
- Week 2: Implement and test code
- Week 3: Experiment, begin presentation & report
- Week 4: Analyze, finish presentation & continue report
- Week 5: Present, finish & submit report

End

Simulation of 1 Lane Highway Traffic flow using the Nagel-Schreckenberg Model

Physics 210 – Term Project Proposal

Rylan Tailfeathers

October 22, 2012

Overview

- To simulate realistic traffic flow on a highway using a model based on cellular automata.
- The Nagel-Schreckenberg model gives a set of rules similar to cellular automata which each 'cell' will have to use to determine the next state.
- The 4 rules as described in the original model include:
- 1. Acceleration
- 2. Braking
- 3. Randomization
- 4. Car motion
- These rules create a realistic model of highway traffic including the randomness of drivers in actual situations.

Project Goals

- To write an octave code using cellular automata and the NS model to simulation highway traffic flow.
- To adjust initial conditions (car density) and maximum velocities and observe changes to behaviors

The Rules of the Model

- The model is on a discrete 100 cell length with each cell being a set length of any value. A velocity of 1 means the car will move 1 cell.
- Acceleration – If a car is not at the max velocity specified then it will accelerate by one unit. This assumes each car wants to go the fastest possible
- Braking – If the car moves faster than the amount of empty cells in front then the car will slow down to the amount of empty cells available. This is the interaction between cars to avoid accidents
- Randomization – There is a probability p, that after braking it will be reduced by 1 more unit. This simulates drivers that overreact and brake too hard
- Car Motion – After applying the 3 rules above, the car moves based on its velocity

Timeline

October 30-November 4 – Research and design structure of code

November 5-13 – Implement and Test code

November 14-20 – Run simulation and begin work on presentation and report

November 21-24 – Analyze data and finish presentation, continue work on report

November 25-27 – Practise presentation and finish report

November 27-29- Present project

November 30 – Submit report

References

- http://en.wikipedia.org/wiki/Nagel-Schreckenberg_model
- A. Schadschneider. The European Physical Journal. June 1999. Volume 10, Issue 3, pp 573-582

The Gravitational N-Body Problem

Physics 210 Term Proposal

Yi-Wen Eve Tsou

- Overview

  - Predicts the motion of a group of N particles which interact with each other gravitationally
  - If initial boundaries for position and velocities are specified at time (t_o), we can determine their positions at a future and past point in time

- Project Goals

  - To design a MATLAB (Octave) code to simulate the N-body problem and to visualize the movement of particles and their forces of interactions between each other
  - To test the code written
  - To investigate initial conditions and see how this affects particle interactions

Mathematical Formulae

- Using Law of Universal Gravitation:

$$\vec{F}_{g_i} = m_i \vec{a}_i = m_i \frac{d^2 \vec{r}_i}{dt^2} = G \cdot \sum_{i \neq j} \frac{m_i m_j}{(r_j - r_i)^{3/2}} (\vec{r}_j - \vec{r}_i) \qquad i = 1,2,3,\ldots,n$$

  - Where G is a gravitational constant
  - r is the distance between object j and object i
- The total force is given by:

$$F_T = F_{gi} + F_o$$

- By combining Newton's second law and Law of Universal Gravitation:

$$m_i a_i = F_T$$

$${}^{d}/_{dt}\, m_i v_i = F_T$$

$$m_i {}^{d}/_{dt} v_i + v_i {}^{d}/_{dt} m_j = F_{Ti}$$

$$m_i r_i'' + m_i' r_i' = F_T$$

$$r_i'' = - \sum_{k=1}^{N-1} G m_k r_{ki} / r^3_{ki}$$

Mathematical Formulae cont'd…

- If $F_T = F_G$ then:

$$r_i'' = -\sum_{k=1}^{N-1} Gm_k r_{ki} / r^3_{ki}$$

- If $m_1 >> m_2$ then:

$$G(m_1 + m_2) = Gm_1$$

- We'll denote as m_1=M and as m_2=m
- The Gravitational Parameter is denoted as

$$\mu = GM$$

where μ = 398, 601 km^3/s^2

- Equation of Motion:

$$r'' + {}^{\mu}/_{r^3} = 0$$

- Thus:

$$r_i'' + \sum_{k = 1}^{N-1} Gm_k r_{ki} / r^3_{ki} = 0$$

Numerical Approach

- The Law of Gravitational Force is then set to a non-linear second order ordinary differential equation:

$$\partial^2 \vec{r}_i / \partial t^2 = \vec{F}_i / m_i$$

- Then specify an initial set of conditions for r_i, m and $v_i = \partial r_i / \partial t$
    - A unique solution should exist
- Using a Backward Euler Taylor Series:

$$t_k = t_0 + h \cdot k \qquad k \in \{1, 2, \ldots\}$$

$$\dot{y} = f(t, y(t))$$

$$y(t-h) = y(t) - h\dot{y}(t) + \frac{h^2}{2}\ddot{y}(t) + \cdots$$

$$y_k \leftarrow y_{k-1} + h \cdot f(t_k, y_k)$$

- Take the derivative of the Taylor series and combine it with the original Taylor series to obtain your final testing equation

$$\vec{x}_i(t+h) = \vec{x}_i(t) + h * \frac{d\vec{x}_i}{dt}(t + h/2)$$

$$\frac{d\vec{x}_i}{dt}(t+a) = \frac{d\vec{x}_i}{dt}(t) + a * \frac{d^2\vec{x}_i}{dt^2}(t)$$

$$\vec{x}_i(t+h) = \vec{x}_i(t) + h * \frac{d\vec{x}_i}{dt}(t) + \frac{h^2}{2} * \frac{d^2\vec{x}_i}{dt^2}(t)$$

- Testing
  - Make sure that energy and momentum is conserved due to the laws of conservation
- Numerical Experiments
  - Using different initial inputs for both position (r_i) and velocities, I will analyze the different results that may occur
  - Create a simulation which mimics the behaviour of the results

Project Timeline

Date:	Activities:
10/22 – 10/29	Research and Design Code
10/30 – 11/7	Implement and Test Code
11/8 – 11/14	Run numerical experiments, begin presentation and report
11/14 – 11/20	Analyze data, continue working on presentation and report
11/20 – 11/26	Finish presentation and work on final draft and report
11/28	Finish final report
11/30	Submit report

References

- http://en.wikibooks.org/wiki/Astrodynamics/N-Body_Problem
- http://en.wikipedia.org/wiki/N-body_problem
- http://www.scholarpedia.org/article/N-body_simulations_(gravitational)
- http://vuduc.org/teaching/cse8803-pna-sp08/slides/cse8803-pna-sp08-20.pdf

Cellular automata simulation of forest fires

Phys 210 Term Project Proposal

Jared Wong

October 25, 2012

Overview

- Cellular automata can simulate a complex system where many models fail, such as fire propagation

- For simplicity, I will model a flat forest with no wind

- I will use a lattice of square cells, where the next state of a cell will depend on the state of it's 8 neighbours

Project Goals

- To write code in MATLAB simulating the 2-D spread of forest fires with cellular automata
- To ensure the behaviour of the model is comparable to the known behaviour of forest fires
- To determine the amount of forest burnt, and how long it takes under a variety of initial conditions

Model Transition Rules

- A cell can have one of 4 states: empty, vegetation, burning, and burned
- The next state of a cell is determined based on it's current state and that of it's neighbours
- An empty cell remains empty, a burned cell remains burned

Model Transition Rules Cont.

- A burning cell becomes a burned cell with probability B
- A vegetation cell remains vegetation if no neighbours are burning
- A vegetation cell becomes a burning cell with probability I(N), with N being the number of burning neighbours

Testing and Numerical Experiments

- Investigate the effect varying I(N) has on the spread of fire
- May include a probability for a cell to spontaneously ignite (for example, from a lightning strike or wind)
- Determine the relation between I(N) and the amount of forest consumed, as well as how long burning lasts

Project Timeline

Dates	Activities
10/26-11/01	Research, derive equations, design code
11/02-11/08	Implement code, test code
11/09-11/15	Run numerical experiment, begin report and presentation
11/16-11/22	Analyse results, continue report and presentation
11/23-11/26	Finalize report and presentation
11/27-11/29	Present term project
11/30	Submit term project

References

- “Forest fires spread modeling using cellular automata approach”, L. Bodroži et al.
- “Stochastic cellular automata model for wildland fire spread dynamics” R.M Almeida and E.E.N Macau
- “A Cellular Automata Model for Fire Spreading Prediction” J. Quartieri et al.

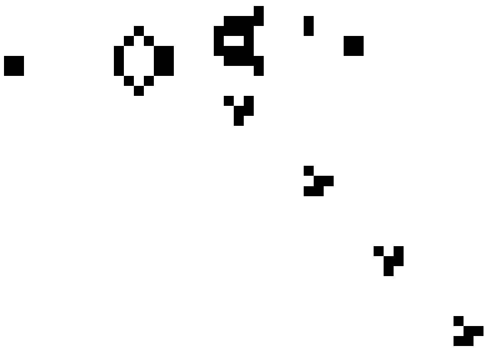
Questions?

Cellular Automata: Forest Fires

Yihui Zhang

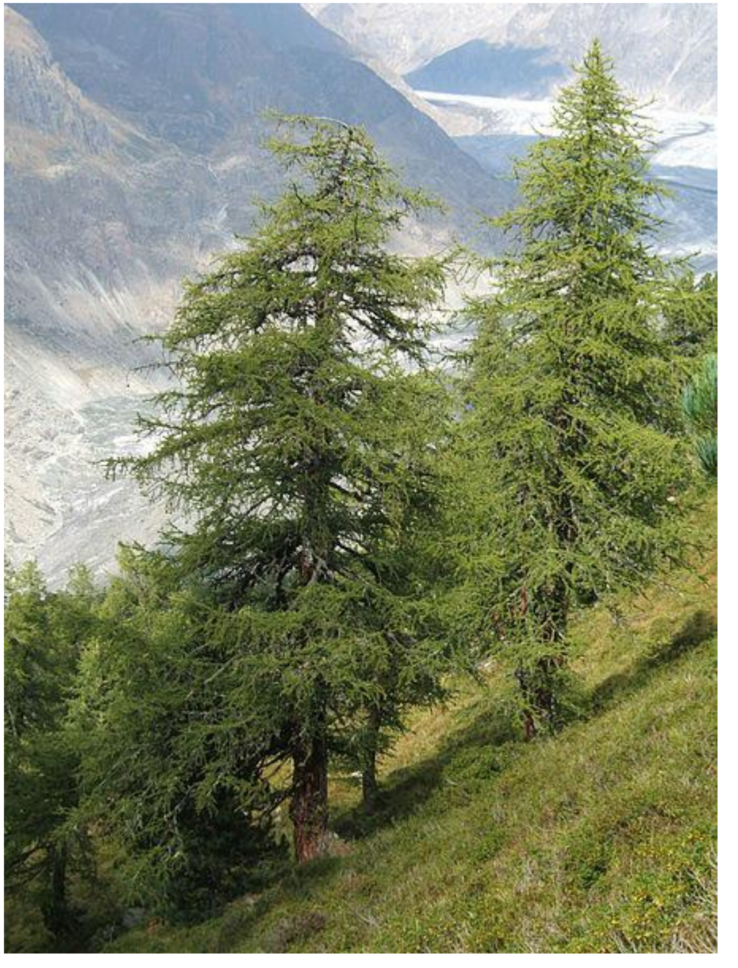
Cellular Automata

- Consists of a grid of cells, each of which has a state (e.g. on or off).
- An initial state for each cell is chosen.
- During each generation, the state of a cell is altered according to some rule.
- This rule consists of determining a cell’s state based on the state of cells its neighbourhood.
- Example: Game of Life



Forest Fires

- Whether a tree starts burning depends on the state of its neighbours.
- Thus, cellular automata can model forest fires.
- I will try hexagonal and square grids.
- I will also try a variety of different rules so the model agrees with actual forest fires.



Schedule

- Weeks 1&2: Research/Learning
- Week 3: Write code
- Week 4: Test code/Running experiments
- Week 5: Write report
- Week 6: Present/Polish Report