Traffic Simulation using Nagel-Schreckenberg Cellular Automaton Model

Phys 210 Term Project

R Rowen Aziz

Overview

- In a stochastic cellular automaton model, there is a grid of cells where the state of each cell changes with time according to some probability distribution. This describes a random dynamical system in discrete time. In such a model, simple rules may lead to complex behaviour.
- In the Nagel-Schreckenberg traffic simulation cellular automaton model, vehicles occupy cells in a grid, and undergo acceleration, slowing down and motion depending on the condition of cells in its neighbourhood. The behaviour of each vehicle is also dependent on randomization.

Project goal

- To write MATLAB code to simulate traffic movement using a cellular automaton model.
- To test the correctness of the model and compare it with known solutions.
- To investigate the model under different initial conditions and boundary conditions.

4-step computational model for all vehicles in grid

- Acceleration: if velocity of vehicle < Vmax, and distance to next car > V+1, increase speed to v+1
- Slowing down (due to other cars): if a vehicle at site I sees the next vehicle at site i+j (with j<=v), it reduces speed to j-1
- Randomization: with probability p, the velocity v goes to v+1
- Car motion: each vehicle moves by v cells equal to its velocity

Transition probability for each vehicle at time t:

$$\frac{dP(\{\sigma_j\})}{dt} = -\sum_i W\left(-\sigma_{i,} - \sigma_{i+1} \,\middle|\, \sigma_{i,}\sigma_{i+1}\right) P\left(\left\{\sigma_{i,}\sigma_{i+1}, t\right\}\right)$$

$$+\sum_i W\left(\sigma_{i,}\sigma_{i+1} \,\middle|\, -\sigma_{i,} - \sigma_{i+1}\right) P\left(\left\{-\sigma_{i,} - \sigma_{i+1}, t\right\}\right)$$

Project Timeline

Dates	Activities
10/ 21 – 10/31	Do basic research, derive equations and begin code design
11/1 – 11/15	Implement code
11/16 – 11/20	Test code
11/21- 11/27	Run numerical experiments, analyze data, begin report
11/28-12/01	Finish report
12/02	Submit project

References

- K. Nagel, M. Schreckenberg. A cellular automaton model for freeway traffic. J. Phys I France 2 (1992): 2221-2229.
- http://laplace.physics.ubc.ca/210/Doc/term-projects/kdv.pdf
- http://en.wikipedia.org/wiki/Cellular_automaton
- http://en.wikipedia.org/wiki/Stochastic_cellular_automaton
- http://en.wikipedia.org/wiki/Dynamical_system

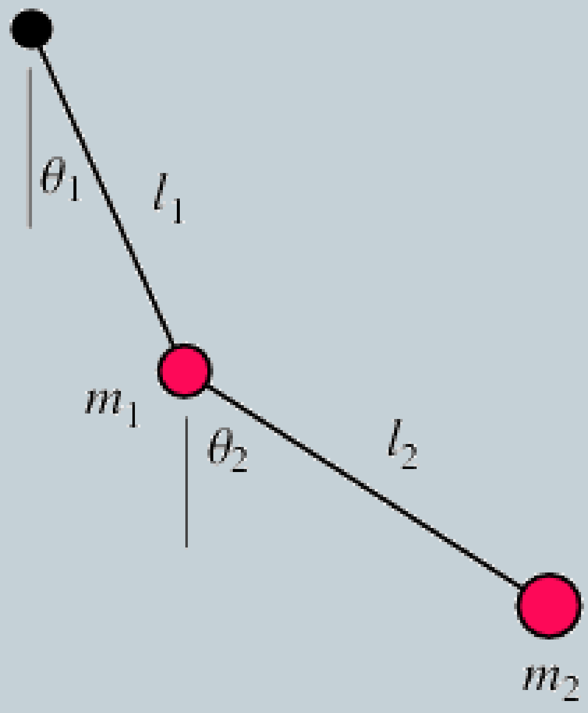
Compound Pendulum

PHYSICS 210 PROJECT PROPOSAL

KABIR CHATTOPADHYAY

Project Overview

- Compound pendulums are pendulums that are connected end to end



Objectives

- Simulate the chaotic motion of a compound pendulum using matlab
- Variables to modify:
  - Number of connectors
  - Length of connector
  - Mass
  - Gravitational force

Equations of motion

$$\theta_1' = \omega_1$$

$$\theta_2' = \omega_2$$

$$\omega_1' = \frac{-g\,(2\,m_1 + m_2)\sin\theta_1 - m_2\,g\sin(\theta_1 - 2\,\theta_2) - 2\sin(\theta_1 - \theta_2)\,m_2\,(\omega_2^{\,2}\,L_2 + \omega_1^{\,2}\,L_1\cos(\theta_1 - \theta_2))}{L_1\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$

$$\omega_2' = \frac{2\sin(\theta_1 - \theta_2)\,(\omega_1^{\,2}\,L_1\,(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \omega_2^{\,2}\,L_2\,m_2\cos(\theta_1 - \theta_2))}{L_2\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$

These are for compound pendulums with 2 masses.

Assumptions

- The connector is assumed to be massless.
- The masses and connectors are able to pass through connectors

Timeline

Date	Goal
22nd Oct – 28th Oct	Finalise theoretical details and design code
29th Oct – 15th Nov	Write and test code
16th Nov – end	Analyse data and prepare paper

Bibliography

- http://scienceworld.wolfram.com/physics/dimg270.gif
- http://www.myphysicslab.com/dbl_pendulum.html

Toomre Model of Galaxy Collisions

Physics 210 Project Proposal

Bryant Cheng
Oct.22/2013

Project Overview

- Galaxy collisions occur due to the gravitational interactions between two galaxies
- Toorme model is a simplified simulation of the process
- In the model, stars and the cores of galaxy are represented as particles with their sizes corresponding to their mass

Objectives

- Write a MATLAB code to depict the collision of two galaxies, using Toorme model
- Use different variable to specify the initial conditions of the two galaxies
- Create a visual representation of the simulation

Assumption

- The mass of individual stars is ignored(only consider the mass in galaxy core)
- Gravitational interactions within the galaxy is ignored, and the collisions of stars inside the galaxy is ignored

Formula

- Force of Gravity $F = G\frac{M+m}{r^2}$
- Centripetal force $F = \frac{mv^2}{r}$

Numerical Approach and Experiments

- Assume the collision happens in a plane, so consider the vectors in 2-Dimesional for all variables
- Try to increase n as much as possible
- Alter the initial conditions of the two galaxies for various results
- Check whether the collisions of two spiral galaxies creates an elliptical galaxy
- Use xvs for interactive analysis and generation of mpeg animations, and MATLAB's plotting facilities for plots to be included in my report

Project Timeline

Dates	Goals
10/23-10/30	Basic Research and design code
10/31-11/15	Implement Code
11/16-11/20	Test Code
11/21-11/26 11/27-11/30 11/31~12/2	Run numerical experiments, analyze data, begin report Finish report Check error and submit project

Reference

- "Toomre Sequence," COSMOS - The SAO Encyclopedia of Astronomy. http://astronomy.swin.edu.au/cosmos/T/Toomre+Sequence. 2013/10/21
- "Alar Toomre," Wikipedia: The Free Encyclopedia. http://en.wikipedia.org/wiki/Alar_Toomre. 2013/10/21

Toomre Model of Galaxy Collisions

Trevor Clarke
35339126

PHYS 210 **University of British Columbia**

Overview

- American astrophysicist Alar Toomre pioneered computer-based models of interactions between galaxies in the 1970's, along with brother Juri Toomre

- Simulations involve simplified gravitational interactions between massive galaxy nuclei treated as points, where stars are "test particles" with $m = 0$
  - Stars are included to aid with visualization, and are only acted upon by gravitational forces – they do not to contribute to gravitational interactions
  - Collisions between stars are ignored – stars appear to "pass through" one another

- Despite the simplified nature of Toomre's simulations, his model can accurately simulate many interesting features and behaviour of observed collisions with modest computing power

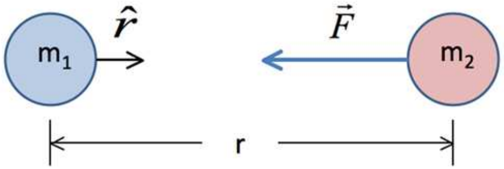
Project Goals

- To write a MATLAB code which simulates the collision of 2 - 3 galaxies (depending on initial levels of success) using the Toomre model
- To explore the behaviour of galaxy interactions over a range of initial conditions, including variation in initial positions, velocities, and masses of colliding galaxies
  - *Question: do interactions between spiral galaxies produce elliptical galaxies?*
- To establish correctness of modeling using the law of conservation of energy
- To generate a visual simulation, as well as plots to be included in final report, using appropriate visualization platforms

Mathematical Formulation

- Toomre model: Newtonian mechanics a sufficient approximation
- Accounts only for gravitational force – all other forces considered negligible

- Newton's law of gravitation:



$$\vec{F} = -\frac{Gm_1m_2}{r^2}\hat{r} \qquad (1)$$

- From above, and using Newton's second law (F = ma):

$$\mathbf{a} = \Sigma\mathbf{F}/m = \Sigma Gm/r^2 \qquad (2)$$

- Kepler's third law:

$$P^2 = 4\pi^2 r^3/Gm \qquad (3)$$

Numerical Approach

- Derive equations of motion from relationship between force and acceleration:
  - $\mathbf{a} = \mathbf{F}/m = \partial \boldsymbol{v}/\partial t = \partial^2 \boldsymbol{r}/\partial t^2$ (4)

- Discretize equations of motion using FDA's
  - $\mathbf{F}/m = \partial \boldsymbol{v}/\partial t \approx (v_{i+1,j} - v_{i-1,j})/(2\Delta t)$ (5)
  - $\mathbf{F}/m = \partial^2 \boldsymbol{r}/\partial t^2 \approx (r_{j+1,j} - 2r_{j,j} + r_{j-1,j})/(\Delta t)^2$ (6)

- Define initial conditions
  - Unique solutions exist for specified initial conditions

- Implement above over finite domains of space and time

Testing and Numerical Experiments

- Test over a broad range of initial conditions, first with two galaxies
  - Attempt to include a third galaxy
- Seek optimal balance between small step size and time for computation
- Check numerical results for conservation of energy
- May seek comparison with established models

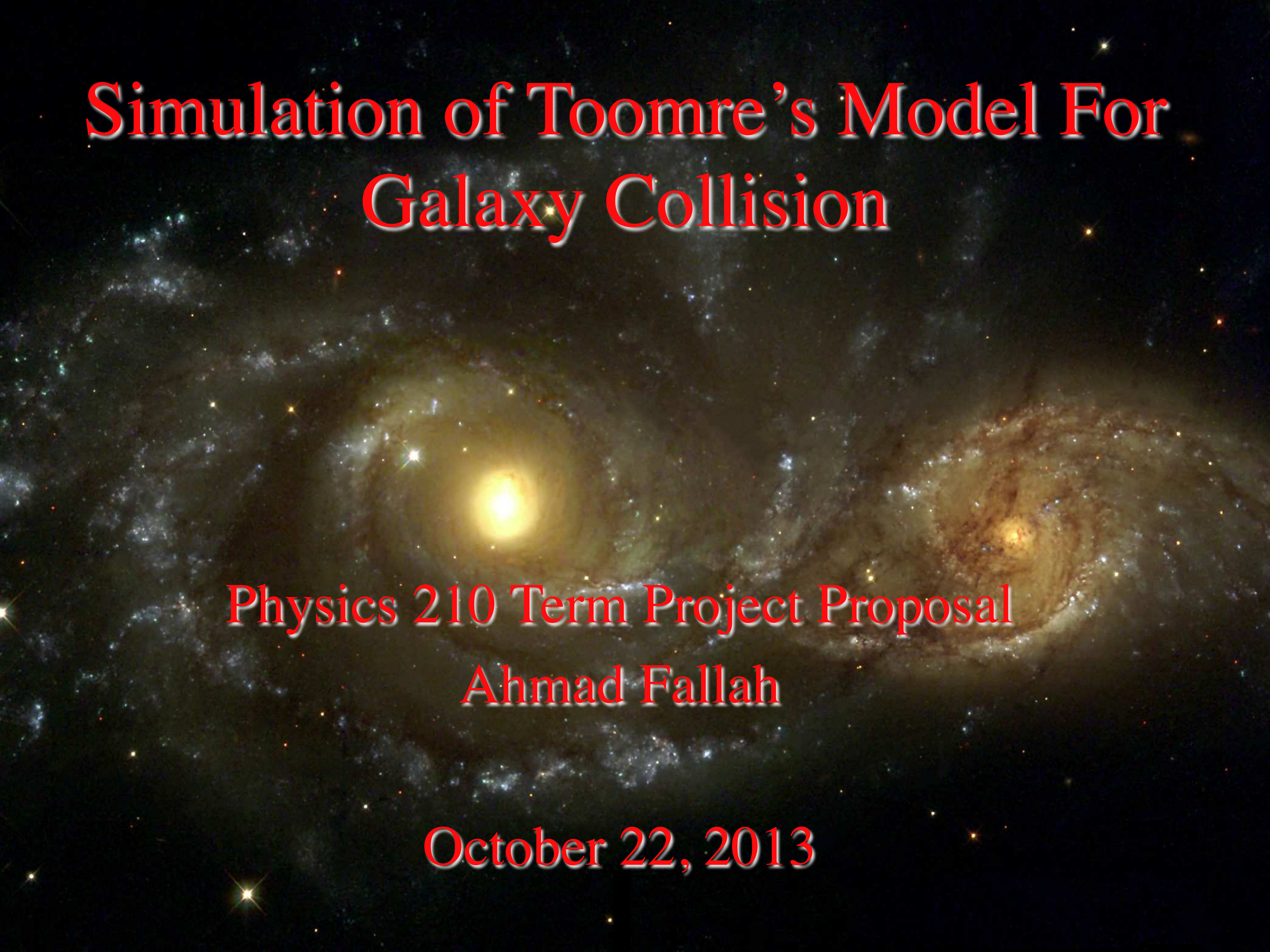
Project Timeline

Dates	Activities
10/19 – 10/26	Do basic research, derive equations & begin code design
10/27 – 11/2	Implement code
11/3 – 11/9	Test code
11/10 – 11/16	Run numerical experiments, analyze data, begin report
11/17 – 11/23	Finish report
11/24 – 11/30	Submit project

References

- http://scitechdaily.com/images/ngc5426-gemini.jpg
- http://sciencenotes.ucsc.edu/9701/full/features/galaxy/Toomre.html
- http://astronomy.swin.edu.au/cosmos/T/Toomre+Sequence
- Choptuik, Matt. “Finite Difference Solution of the Korteweg & de Vries (KdV) Equation – Physics 210 Term Project Proposal”. Available at: http://bh0.phas.ubc.ca/210/Doc/term-projects/kdv.pdf.

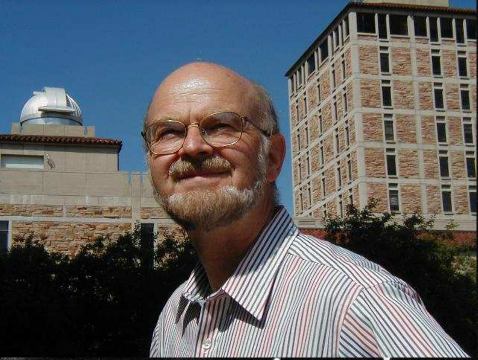
Simulation of Toomre's Model For Galaxy Collision



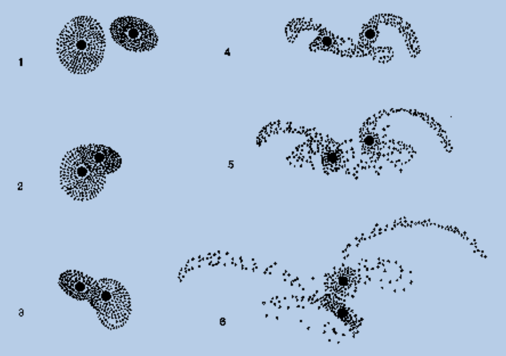
Physics 210 Term Project Proposal

Ahmad Fallah

October 22, 2013



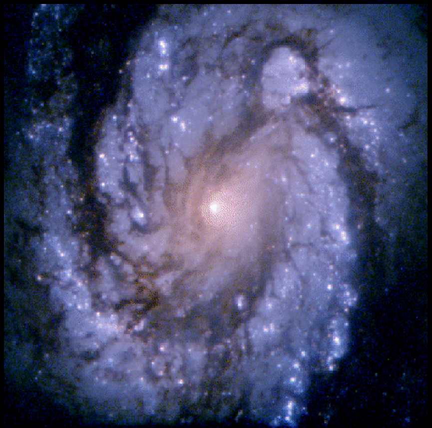
Overview



- Toomre's Model is a model which simulate the collision/interaction of two galaxies.
- In the early 1970s, Alar Toomre with his brother Juri set a collision of two galaxies in motion with limited computing power, kept the number of stars to 1000.
- Toomre's Model uses Newton's gravitational laws to simulate the collision of two stars

Project Goal

- To write a Matlab code which simulate a collision of two galaxies by using Toomre's Model.
- To investigate various initial conditions such as velocity, mass, and positions.
- To find the final type of the galaxy after the collision such as Elliptical, Spiral, S0 and Irregular galaxy
-



Mathematical Formulation

- The attraction force between each particle can be determine by Newton's second law

$$F = m \cdot a$$

$$\frac{\partial^2 x}{\partial t^2} = G \cdot \frac{M}{r^2}$$

$$F = G \cdot \frac{M \cdot m}{r^2}$$

- In N-Body simulations we use a system of N particles, therefore we use second order differential equation of motion:

$$\frac{d^2 \mathbf{r}_i}{dt^2} = -G \sum_{\substack{j=1 \\ i \neq j}}^{N} \frac{m_j}{r_{ij}^3} \mathbf{r}_{ij},$$

- Kepler's Third Law:

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

Numerical Approach

- I will use use finite difference technique to simulate the gravitational attraction between two galaxies along with their stars.

- The initial conditions for velocity and position, and the boundary condition for total mass will be specified.

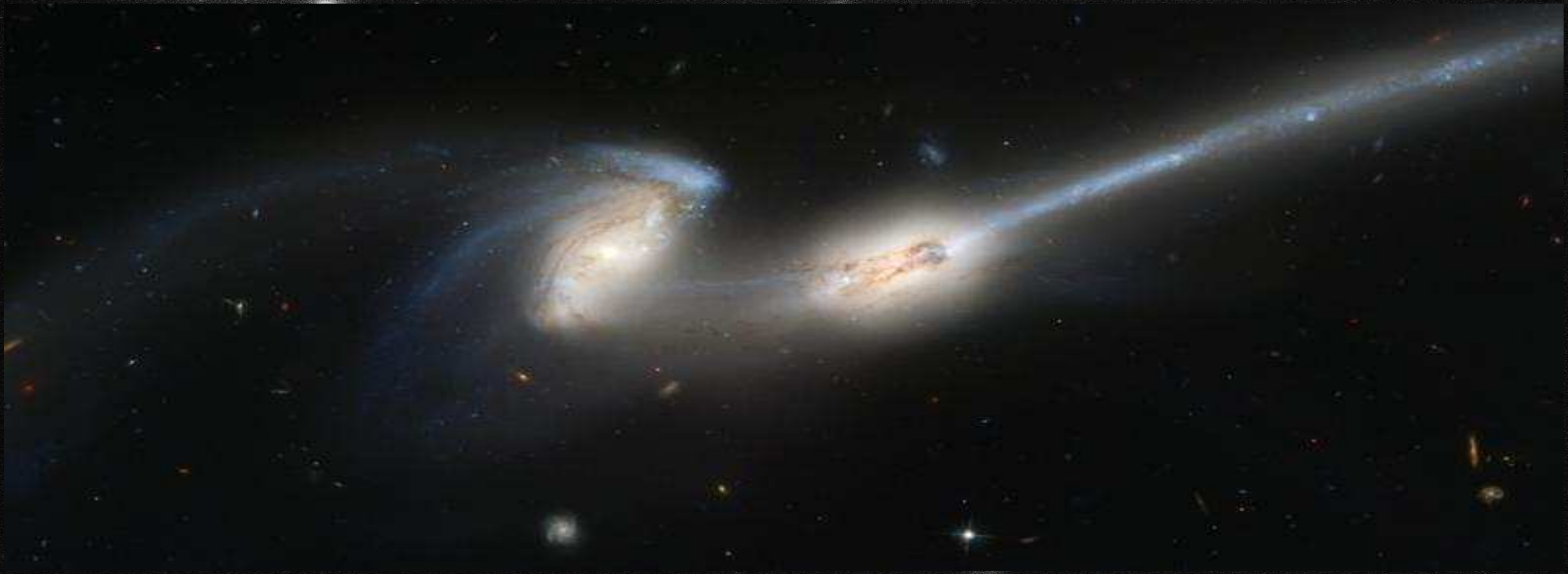
- In the collision of two galaxies the problem arises when two galaxies are too close to each other, this is due to the singularity when $r_{ij} \rightarrow 0$, therefore I am going to use softened potential given by:

$$\Phi_i = -G \sum_{\substack{j=1 \\ i \neq j}}^{N} \frac{m_j}{(r_{ij}^2 + \varepsilon^2)^{1/2}}$$

- I will also limit the number of particles to $N \leq 10^3$ to save calculation time and increase the simulation efficiency.

Visualization

Create an mpeg simulation file by using MATLAB software.



Testing & Numerical Experiments

- Investigating the final galaxy type after the collision between the two galaxy.
- Varying the initial conditions and boundary conditions, such as mass, velocity, positions and angles of the collision to investigate different results.
- Compare the simulation with other existing simulation from the internet.

Project Timeline

Dates	Activity
10/22-10/29	Do Basic research, derive equations & design code
10/30-11/05	Coming up with an algorithm and design for coding
11/06-11/12	Implement code
11/13-11/20	Test Code
11/21-11/27	Run numerical experiment and fill-up the gap in code
11/27-11/29	Finish Report
12/01	Submit Term Project

References

http://www.ias.ac.in/jarch/jaa/8/17-31.pdf

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

http://hyperphysics.phy-astr.gsu.edu/hbase/kepler.html

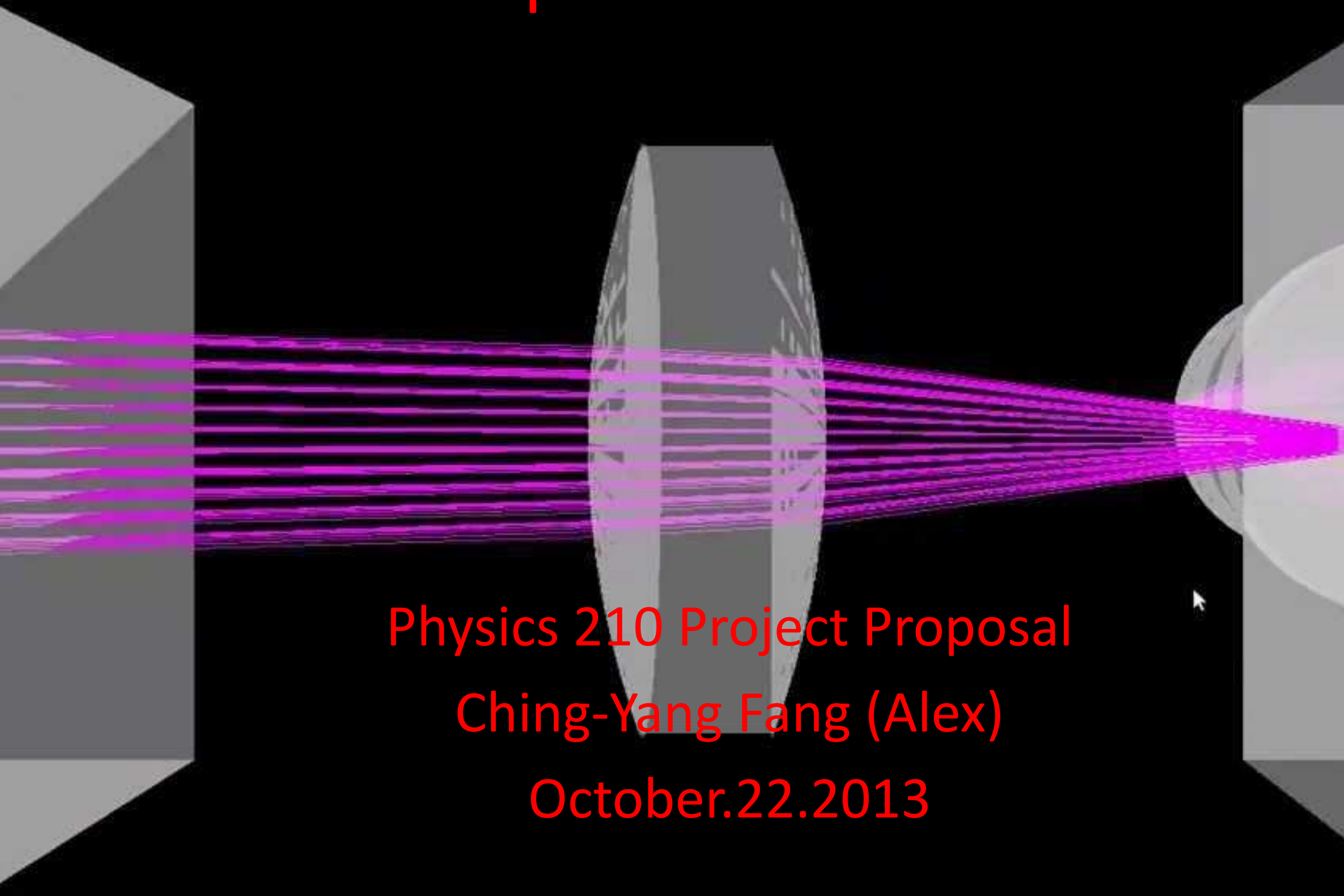
http://lcd-www.colorado.edu/jtoomre/

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

Colliding Galaxies NGC 4038 and NGC 4039
Hubble Space Telescope • Wide Field Planetary Camera 2

PRC97-34a • ST ScI OPO • October 21, 1997 • B. Whitmore (ST ScI) and NASA

Optic Simulation



Physics 210 Project Proposal

Ching-Yang Fang (Alex)

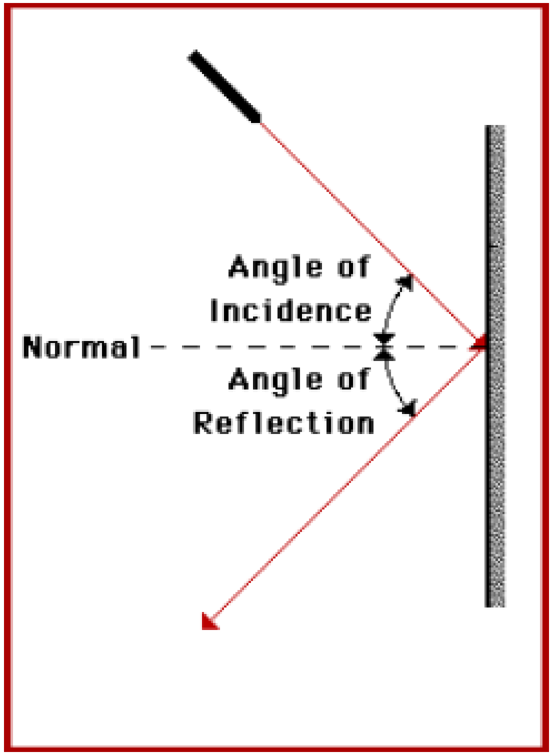
October.22.2013

- Project Overview:
- A simulation of light rays through different lens, prisms, and mirrors resulting in refraction and reflection.

- Project Goal:
- To write an MATLAB code that accounts the reflection and refraction of light rays through different objects using first order finite difference technique.
- To create a visual representation from the compiled result.

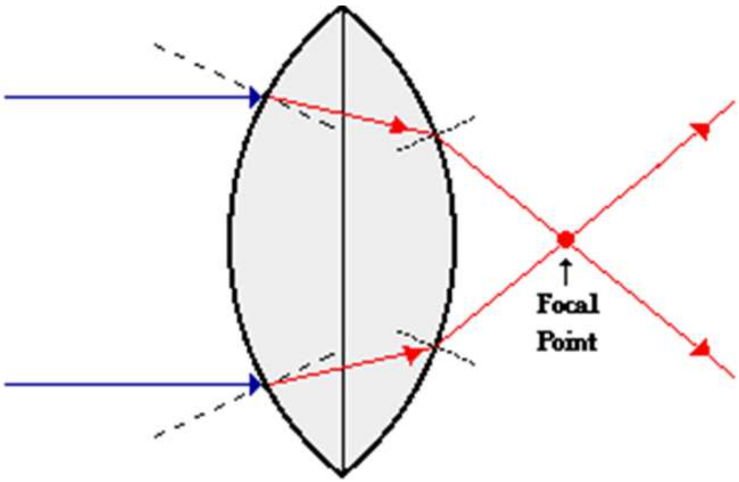
- Mathematical Formulations
- Angle of incidence= angle of reflection
- Snell's Law: $\sin(\theta 2) = (n_1/n_2)\sin(\theta 1)$
- n_1= refraction index of air=1.000293
- n_2= refraction index of crown glass= 1.52

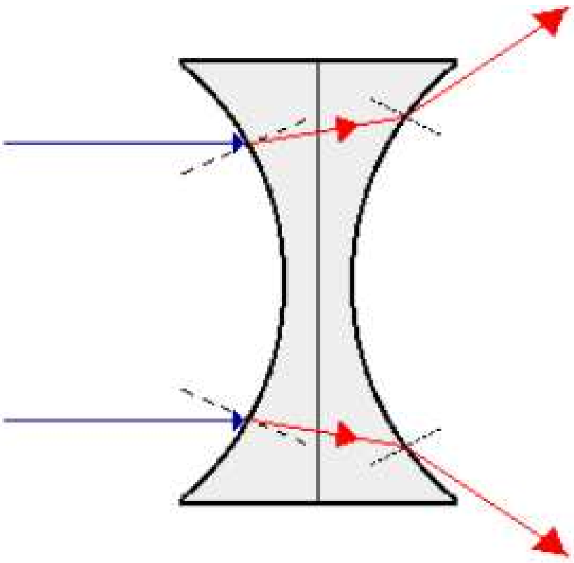
- Approach:
- For Mirror:
  - calculate the angle of incidence(θ)
  - use matrix to represent the components of light then compute the rotation of its components separately by 2θ then multiply the resulting direction matrix by -1 (because the direction is now opposite. When it is REFLECTED)



- For Lens and Prisms:

  -calculate the θ between the light and the normal vector at the point of contact
  -apply Snell's Law to find th refracted angle.





- Visualization and Plotting Tools
- I will use xvs for interactive analysis and generation of mpeg animations, and
- MATLAB's plotting facilities for plots to be included in my report
- Testing & Numerical Experiments
- Error Convergent Test:
  - fixing initial data and compute the solution/simulation with discretization scales h, h/2, h/4, h/8 …… and ensure the error term $O(h^2)$ is decreasing.

- Numerical Experiment:
- Investigate the interaction between light and different lens and mirrors with different incoming angles.
- Investigate how refraction indexes determine the refraction angles.

- Project Timeline:

Dates	Activities
10/22 – 10/26	Basic research, derive calculation
10/27 – 11/16	Implementing code and building program structure
11/16 – 11/18	Test code
11/19 – 11/25	Running numerical experiments, analyze data and being report
11/26 – 11/30	Finish report
11/30	Submit (deadline 12/02)

Reference:

- http://www.physicsclassroom.com/class/refrn/u14l5b.cfm
- http://en.wikipedia.org/wiki/Snell's_law
- http://www.baylee-online.net/Projects/Raytracing/Algorithms/Laws-Of-Optic

Gravitational N-Body Problem

Erik Frieling

Phys 210 Term Project Proposal

Overview

- Simulation of N interacting particles given initial conditions
- Implement code in Matlab
- Work in 3D, maybe switch to 2D
- Present graphically

Mathematical Formulation

$$F = \frac{Gm_1m_2}{|r|^2}\hat{r}$$

$$F = ma = m\frac{dv}{dt}$$

Initial conditions

- For each particle:
  - Velocity
  - Position
  - Mass

Note

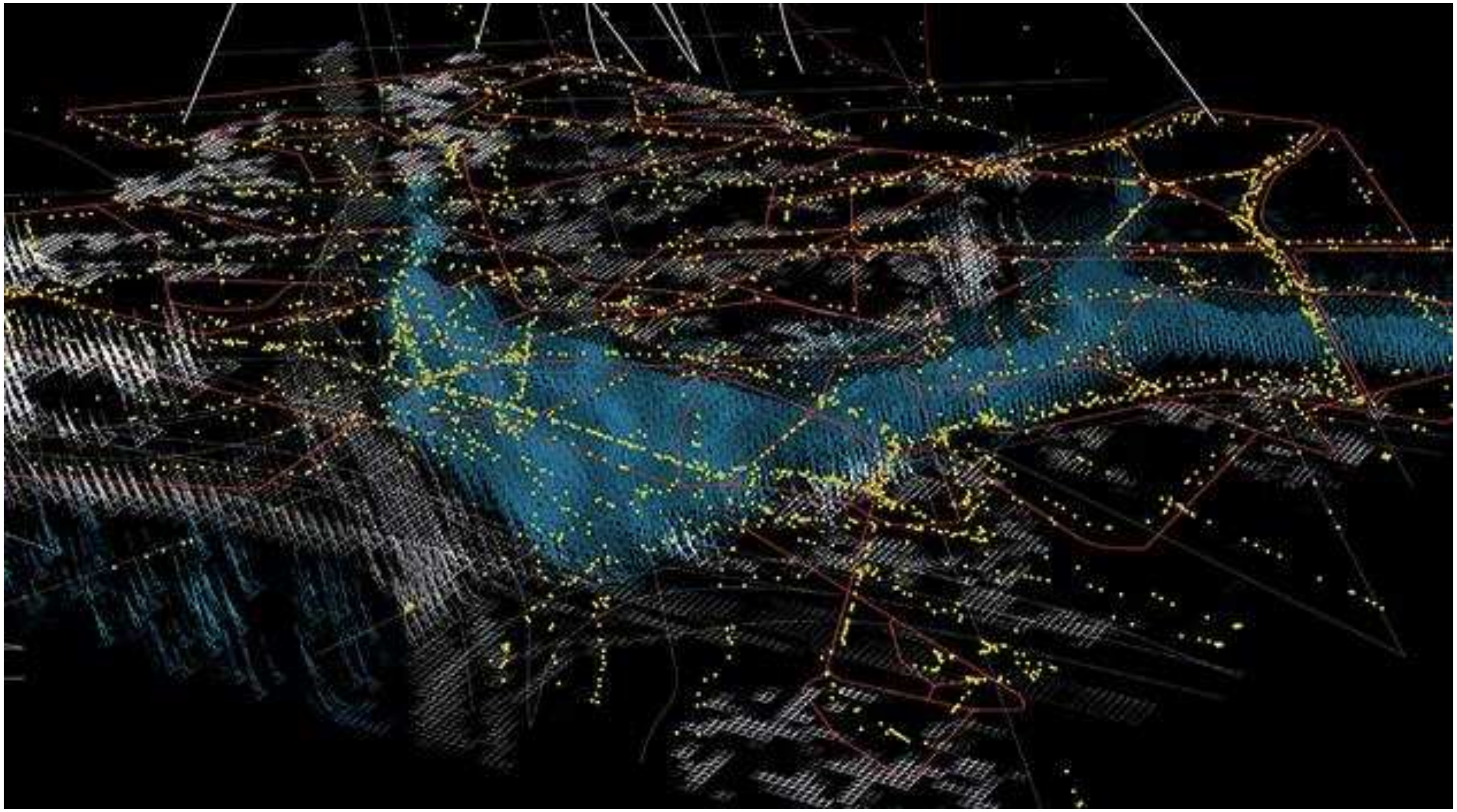
- At r=0, F will become infinite
- Thus, combine particles that come into a certain critical radius

Project Timeline

By October 31st	Research code, finish design
By November 9th	Fully implement the code
By November 18th	Test and refine code
By November 25th	Analyze solutions and complete report
By December 2nd	Handin

Traffic simulations by cellular automata

Rick Gao
October 21, 2013



Ways to simulate

- *Real Transportation Systems*
- *Cellular Automata*
- *Car-following models*
- *Numerical PDE methods*
- *Microscopic traffic flow models*

Overview

A cellular automata is a discrete model which consists of a regular grid of cells, and each cell has a finite number of states.

For traffic simulation:

Grid - grids make up the road

State – car exist (1) / no car(0)

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

Project Goals

- *To write a MATLAB code to simulate traffic by using cellular automata.*
- *To investigate how the speed and acceleration of cars related to the cells.*
- *To study lane changing behavior for cars*
- *To analyze what causes traffic congestion*

Mathematical Formulation

- *Cell stat (t) = f (cell neighborhood at (t-1))*
- *The Nagel-Schreckenberg model*
- *1. Acceleration: $V(n) \rightarrow \min(V(n) + 1, V_{max})$*
- *2. Deceleration: $V(n) \rightarrow \min(V(n), D(n) - 1)$*
- *3. Randomization: $V(n) \rightarrow \max(V(n) - 1, 0)$ with probability p*
- *4. Movement: $X(n) \rightarrow X(n) + V(n)$*

–
  Overview
- *Project Goals*
- *Mathematical Formulation*
- *Numerical Approach (don't worry if you're unsure about this: for many projects, the computational techniques will be covered in future lectures and labs)*
- *Visualization and Plotting Tools (above comment applies here)*
- *Testing and Numerical Experiments*
- *Project Timeline*
- *References*

Mathematical Fomulation

- *The average density per lane is*

  <p>L=N/(L)

- *Lane change model*

  -Gap(i)<L

  -Gap0(i)>L(0)

  -Gap(0,back(i))>L(0),back

  -rand()<Pchange

Numerical Approach

- *Plot*
  - *position of cars vs time*
  - *density per lane vs time*
  - *Lane change vs density*
- *Using traffic signal to control car in intersect*
- *Apply lane change model and Nagel-Schreckenberg model for cars*

Visualization and Plotting Tools

- *use MATLAB plotting facilities*

Testing

- *I will test lane change model, traffic signal, and the Nagel-Schreckenberg model separately.*

  - Change the parameter and check if the graph is reasonable.

Project timeline

Date	Activity
Oct 23 – Oct 27	*Research*
Oct 28 – Oct 29	*Write code for lane change*
Oct 29 - Nov 2	*Write code for traffic signal*
Nov 2 – Nov 4	*Write code for the rest*
Nov 5 – Nov 11	*Test code and analyze*
Nov 12 – Nov 25	*Write lab report*
Nov 25 – Dec 2	*Finish up and submit*

Referance

- *M. Rickert, K. Nagel, M. Schreckenberg, A. Latour, Two lane traffic simulations using cellular automata, Physica A: Statistical Mechanics and its Applications, Volume 231, Issue 4, 1 October 1996, Pages 534-550, ISSN 0378-4371,*
- *Dietrich E. Wolf, Cellular automata for traffic simulations, Physica A: Statistical Mechanics and its Applications, Volume 263, Issues 1–4, 1 February 1999,*
- *http://en.wikipedia.org/wiki/Cellular_automaton*

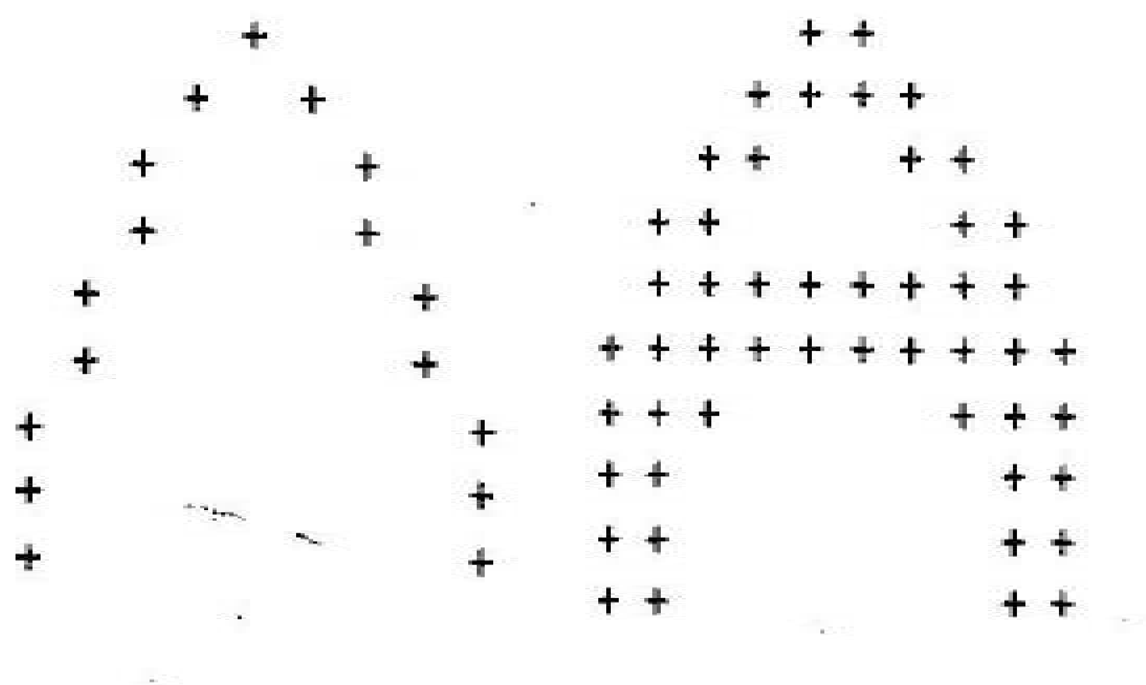
Simulation of a Simple Neural Network

Jaspreet Garcha

What am I talking about?

- Ising Model of network
- Monte Carlo Method of evaluation

(image from Giordano 12.3-4)



Yeah, but what about....

- Firing Rate? Neurons either firing (+1) or not (-1)
- Transit Time? Signals transmit instantaneously (VERY GOOD APPROX.)
- Connection Pattern? Connections are symmetric/all-encompassing

Maths and Such

$$E = -\sum_{ij} (J_{ij} * s_i * s_j)$$

where $J_{ij} = (1/M) * \sum_m (s_i(m) * s_j(m))$

for $M < \sim 0.13 * (\#\text{ of cells})$

What am I doing?

- Exploring the relationship between damage to the neural network (set value to 0) and the number of memories that can be successfully stored/recalled
- Compare to theoretical value (~0.13N)

Timeline

Do project stuff	Now-Dec.1
Chill	Dec. 2

Citations and Such

- “Giordano, 12.3-4 Neural Networks” Phys 210 Homepage http://laplace.physics.ubc.ca/210/Doc/term/Giordano-12.3-4-Neural-Networks.pdf
- http://en.wikipedia.org/wiki/Ising_model
- http://en.wikipedia.org/wiki/Monte_Carlo_method

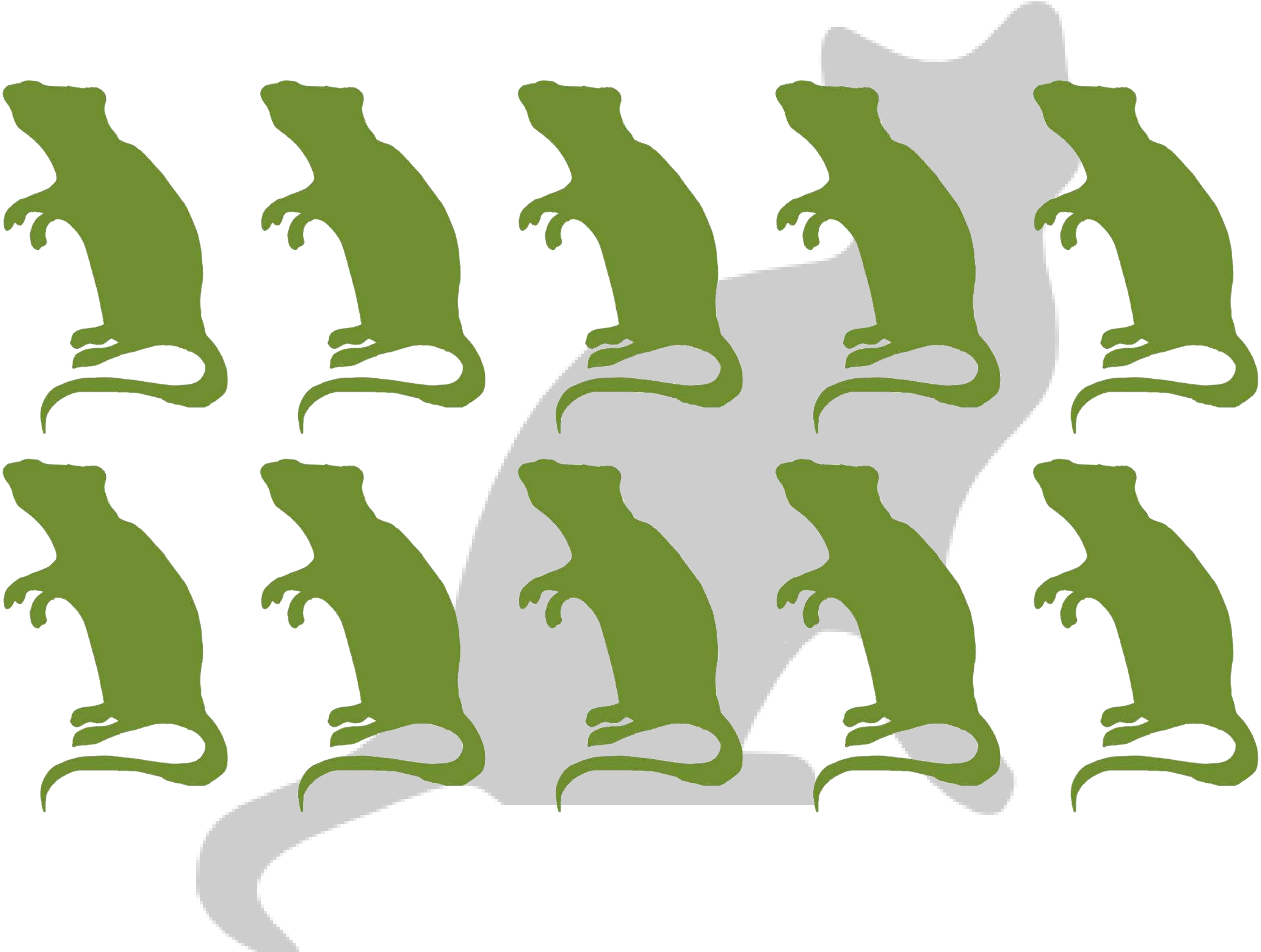
Questions and such?

NEURAL NETWORKS

OCT 22 2013 - AMON GE

INTRO

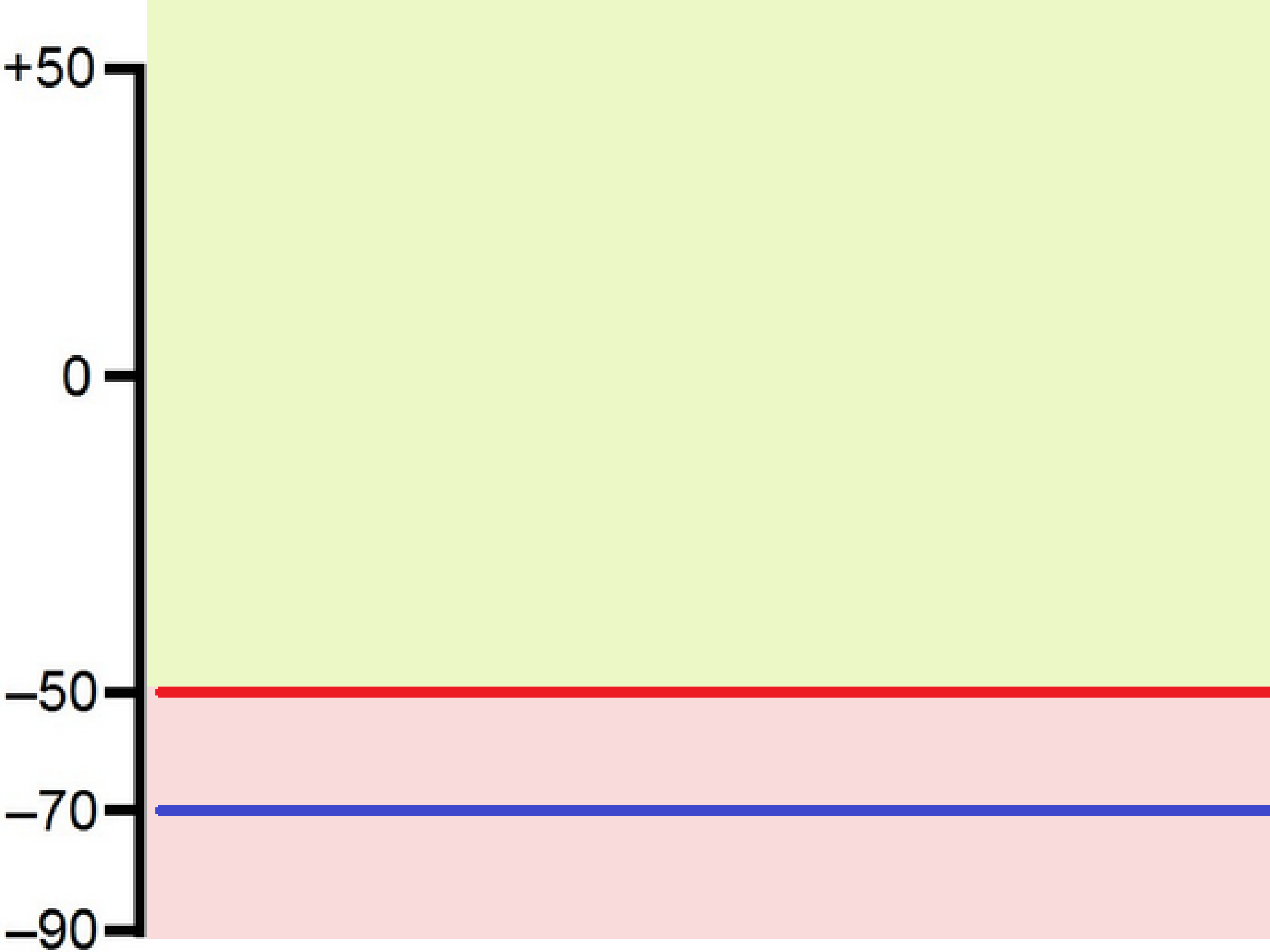
10^{12}



A LOT.

2

INTER-
CONNECTED



+50
0
−50
−70
−90

ALL OR NONE

MATH

OF A MEMORY SIMULATION

ALL OR NONE

+1 **OR**

-1

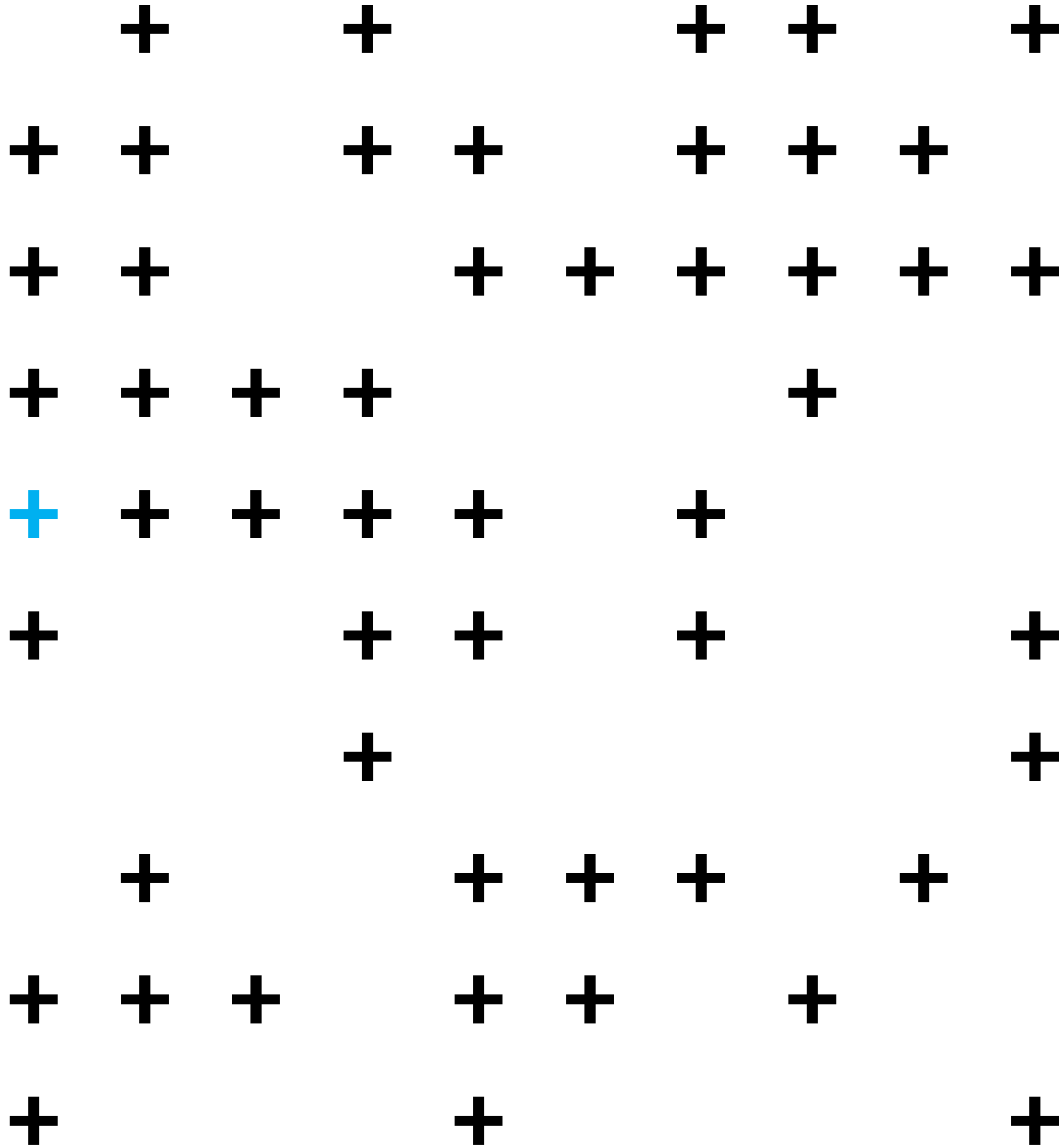
−	+	−	+	−	−	+	+	−	+
+	+	−	+	+	−	+	+	+	−
+	+	−	−	+	+	+	+	+	+
+	+	+	+	−	−	−	+	−	−
+	+	+	+	+	−	+	−	−	−
+	−	−	+	+	−	+	−	−	+
−	−	−	+	−	−	−	−	−	+
−	+	−	−	+	+	+	−	+	−
+	+	+	−	+	+	−	+	−	−
+	−	−	−	+	−	−	−	−	+

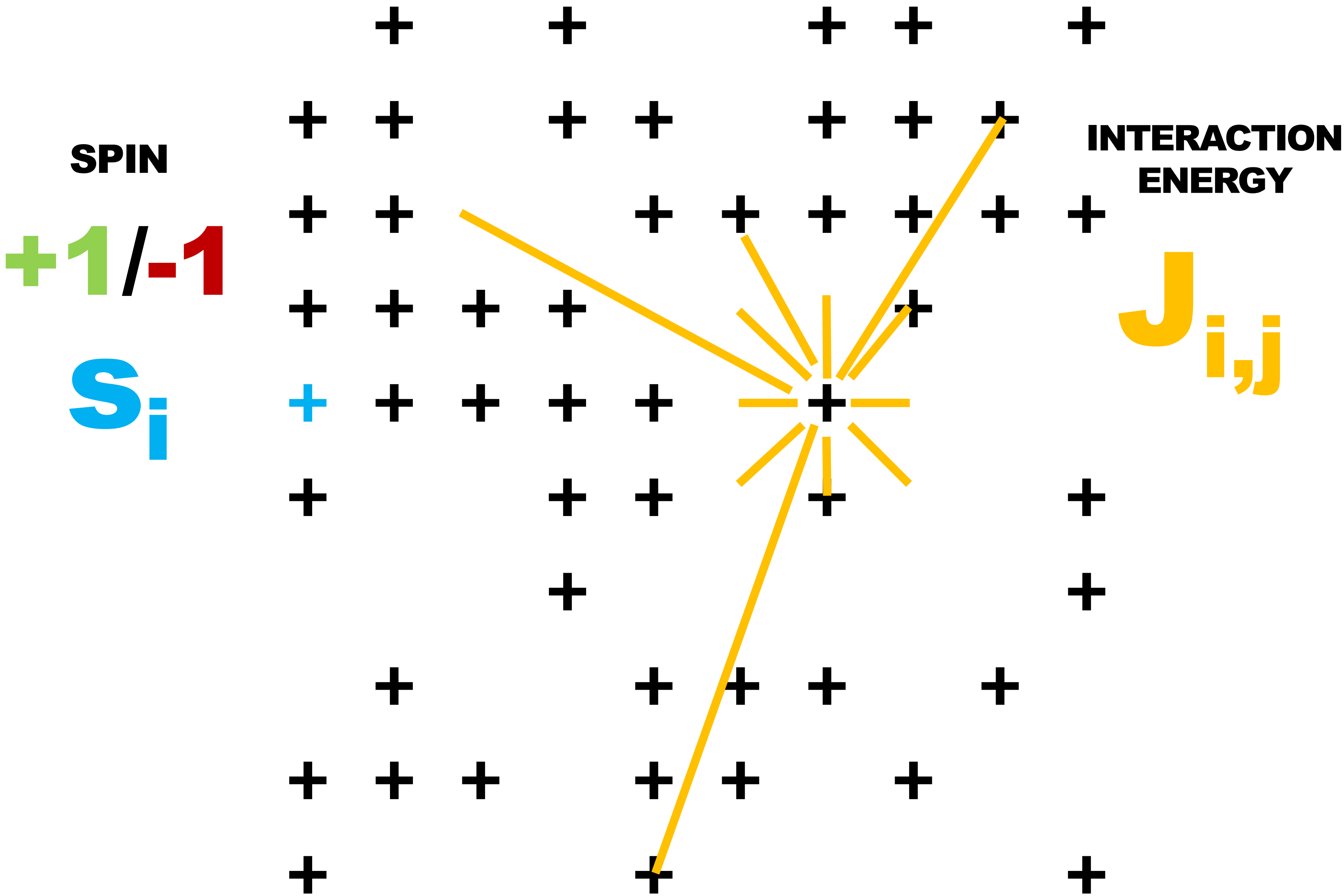
	+		+			+	+		+
+	+		+	+		+	+	+	
+	+			+	+	+	+	+	+
+	+	+	+				+		
+	+	+	+	+		+			
+			+	+		+			+
			+						+
	+			+	+	+		+	
+	+	+		+	+		+		
+				+					+

SPIN

+1/-1

s_i




SPIN
+1/-1
s_i
INTERACTION ENERGY
$J_{i,j}$

PATTERN

m

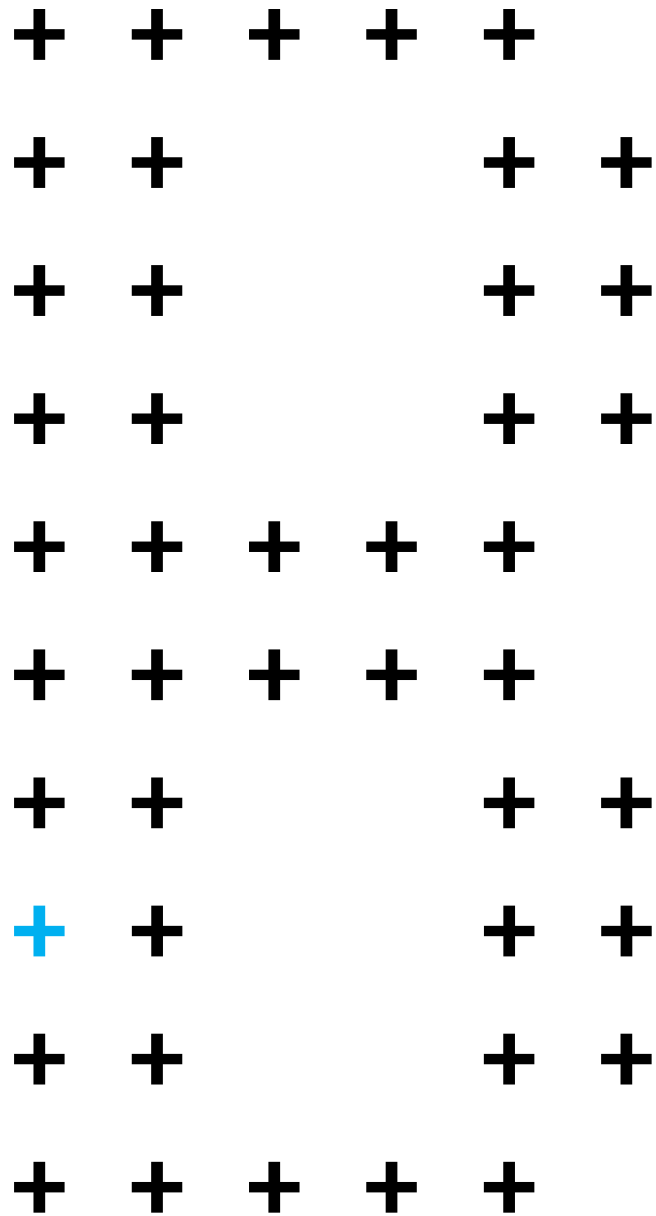
```
+ + + + +
+ +     + +
+ +     + +
+ +     + +
+ + + + +
+ + + + +
+ +     + +
+ +     + +
+ +     + +
+ + + + +
```

PATTERN

m

SPIN

s(m)



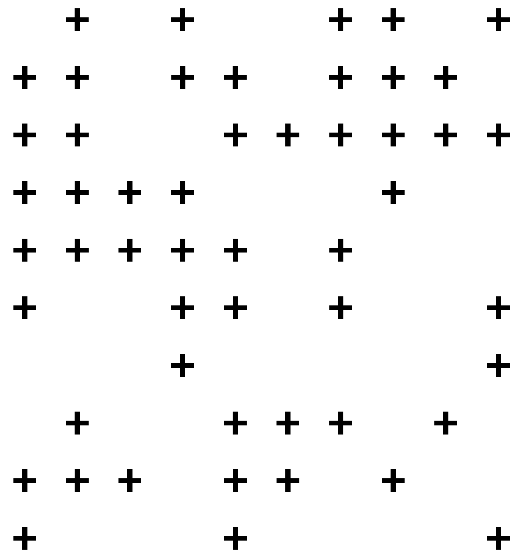
```
        + +             + + + + +         + + + + + +
      + + + +           + +     + +     + + + + + + +
    + +     + +         + +     + +     + + +
  + +         + +       + +     + +     + +
  + + + + + + + +       + + + + +       + +
+ + + + + + + + + +     + + + + +       + +
+ + +         + + +     + +     + +     + +
+ +             + +     + +     + +     + + +
+ +             + +     + +     + +     + + + + + + +
+ +             + +     + + + + +         + + + + + +
```

PATTERNS

= M

INTERACTION ENERGY

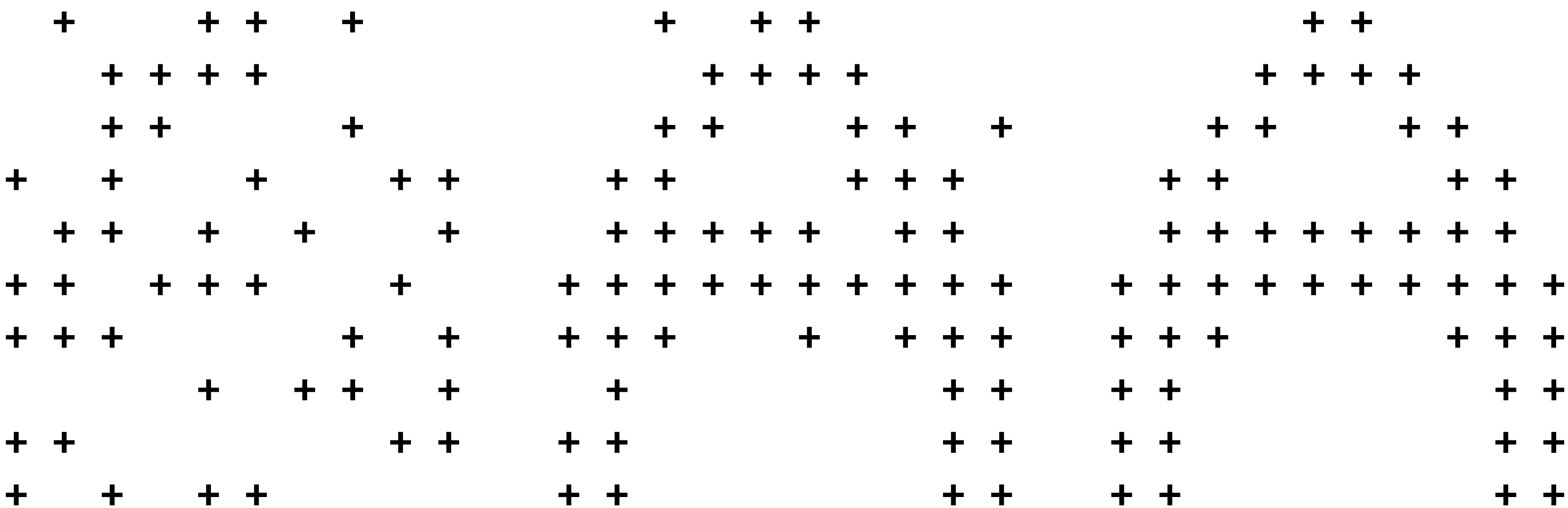
$$J_{i,j} = \frac{1}{M}\sum_{m} s_i(m)\cdot s_j(m)$$



EFFECTIVE ENERGY

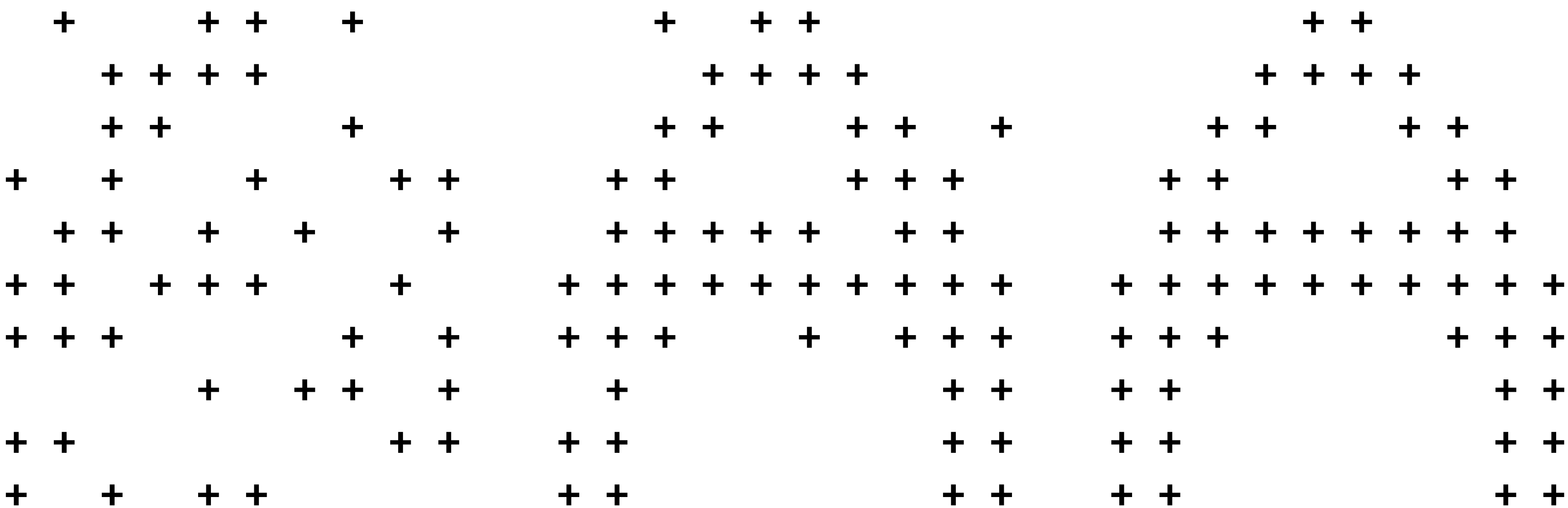
$$E = -\sum_{i,j} J_{i,j} \cdot s_i \cdot s_j$$

MONTE CARLO



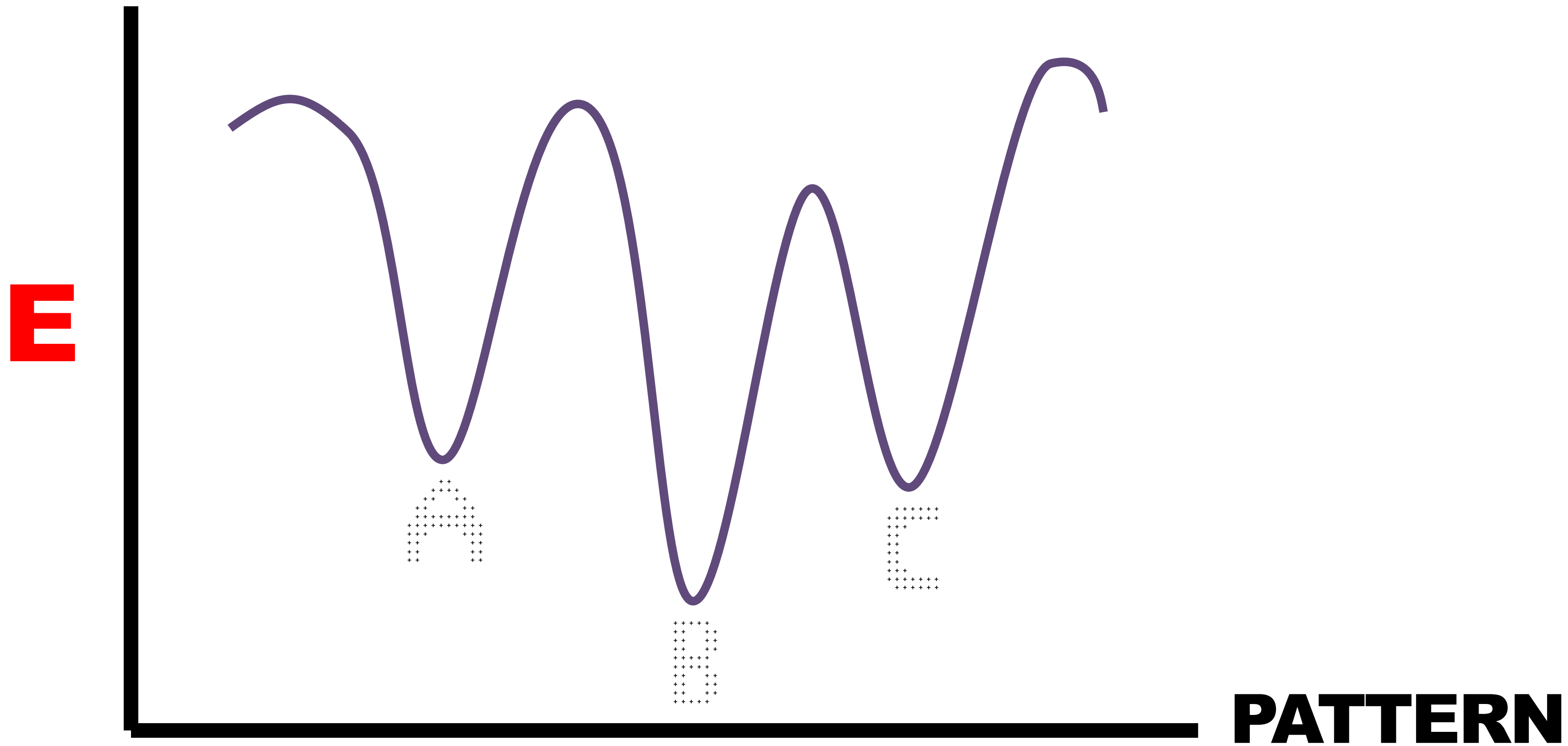
$$\Delta E_{FLIP} < 0 : \text{FLIP}$$

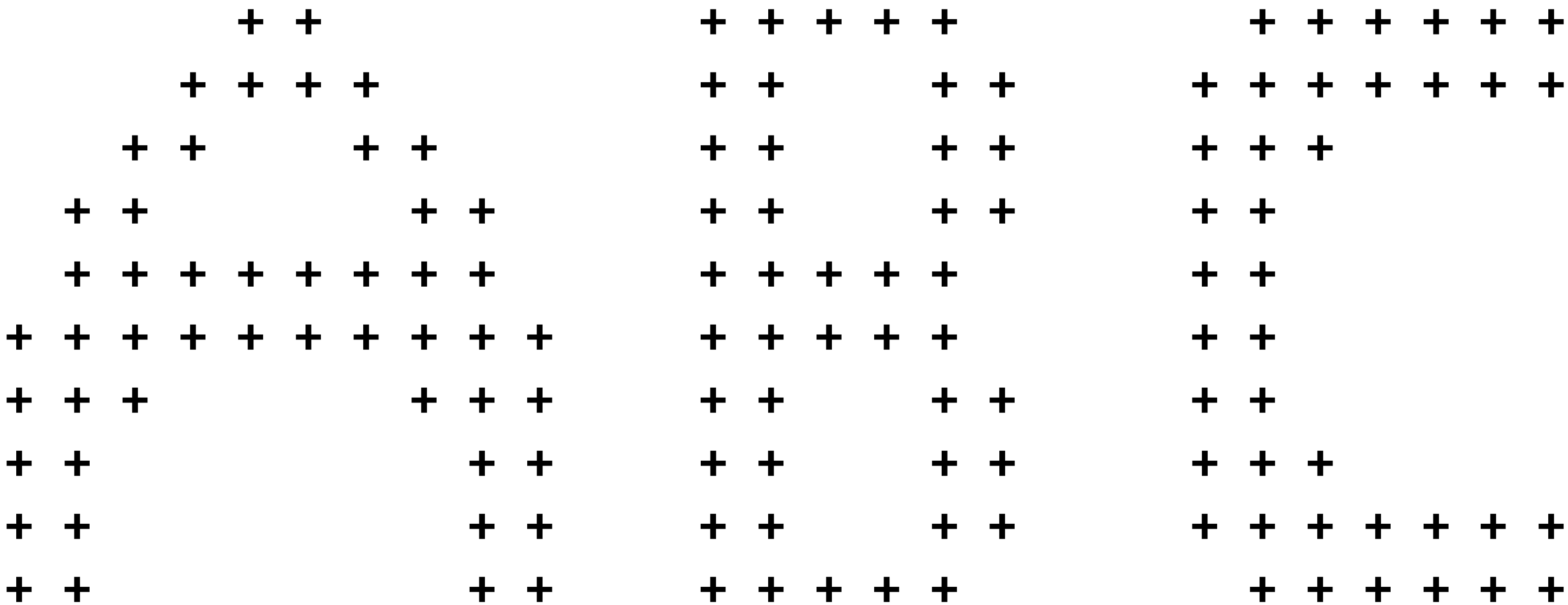
$$\Delta E_{FLIP} \geq 0 : \text{NO FLIP}$$



$$\Delta E_{FLIP} < 0 : \text{FLIP}$$

$$\Delta E_{FLIP} \geq 0 : \text{NO FLIP}$$





MAX PATTERNS

$M \approx 0.13\,N$

(SPINS)

FURTHER THOUGHTS

DAMAGE : p_{damage}

probability to set

$J_{i,j} = 0$

DAMAGE : p_{damage}

probability to set

$J_{i,j} = 0$

LEARNING : $J_{i,j}$ **(new)**

$= \alpha\, J_{i,j}\,(\text{old}) + \beta\, s_i(m_{new})\, s_j(m_{new})$

GOALS

SIMULATE A NETWORK

SIMULATE A NETWORK

TEACH IT TO REMEMBER

SIMULATE A NETWORK

TEACH IT TO REMEMBER

TEACH IT TO LEARN

OCT
22
23
24
25
26
27
28
29
30
31

NOV	
1	17
2	18
3	19
4	20
5	21
6	22
7	23
8	24
9	25
10	26
11	27
12	28
13	29
14	30
15	
16	

DEC
1
2
3
4

RESEARCH

DEPLOYMENT

EXPERIMENT

REPORT

PRESENTATION

QUESTIONS?

REFERENCES

GIORDANO, N. J., & NAKANISHI, H. (1997). NEURAL NETWORKS AND THE BRAIN. *COMPUTATIONAL PHYSICS* (PP. 418-436). UPPER SADDLE RIVER, N.J.: PRENTICE HALL.

IMAGES

HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/6/68/BLANKMAP-USA-STATES-CANADA-PROVINCES.PNG

HTTP://WWW.CLKER.COM/CLIPARTS/2/D/5/F/11954407231266496009LIFTARN_CAT_SILHOUETTE.SVG.MED.PNG

HTTP://WWW.HYATTS.COM/ECOM/IMAGES/R/RAT2.JPG

HTTP://SCIENTOPIA.ORG/BLOGS/SCICURIOUS/FILES/2011/05/NEURONS4.JPG

HTTP://WWW.PHYSIOLOGYWEB.COM/LECTURE_NOTES/NEURONAL_ACTION_POTENTIAL/FIGS/NEURONAL_ACTION_POTENTIAL_PHASES.JPG

Modelling Diffusion Limited Aggregation

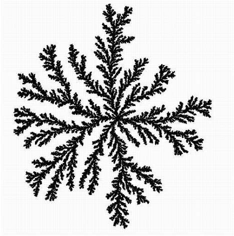
& Possible Applications to Snowflake Formation.

Heather Guy

October 21, 2013

Overview

- Diffusion Limited aggregation (DLA): A basic model where particles undergoing 'random walks' due to brownian motion collide and join to form aggregates.
- Applies to systems where diffusion is the primary means of transport.

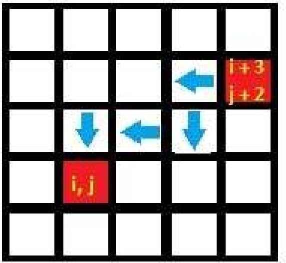


Project Goals

- To write a MATLAB code which creates a simulation of diffusion limited aggregation which can be adapted to include different numbers of particles and initial conditions.
- To establish correctness of the implementation of the code through comparison with known solutions.
- To adapt the model to show anisotropic aggregation in order to model the first stage of snowflake formation and compare with known solutions. *(time dependent)*

Basic model

- A seed particle is fixed in the center of finite-difference lattice with grid spacing h.
- A second particle enters the lattice at a random location and undergoes a 'random walk' until it leaves the lattice or collides and 'sticks' to the first particle.
- The process is repeated for a large number of particles.



Visualization and Plotting Tools

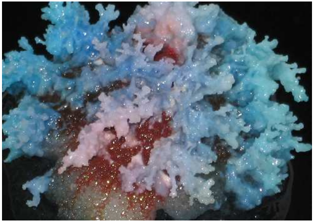
- I will use MATLAB's plotting facilities for plots to be included in my report.

Testing and Numerical Experiments

- Comparison with known solutions.
- Tests for statistical self-similarity (fractal dimensions)
- Investigate the effect of different total numbers of particles.
- Investigate the effect of multiple particles diffusion through the lattice at the same time.

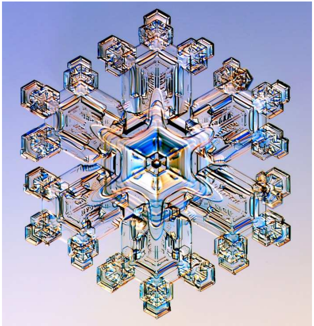
Possible Applications

- Dielectric breakdown
  - Lightning
- Crystal growth
  - Snowflake formation
- Coral growth
- Coalescing of dust or smoke particles



Possible Adaptations for Snowflake Formation

- Anisotropic
- Preferential crystal growth in 6 directions, hexagonal lattice?



Project Timeline

Dates	Activities
10/15–10/26	Do basic research, derive equations & begin code design
10/27–11/15	Implement code
11/16–11/19	Test code
11/20–11/26	Run numerical experiments, analyze data, begin report
11/27–11/29	Finish report
11/29	Submit project! (absolute deadline is 12/02)

References

http://people.umass.edu/machta/introduction.html
http://pauli.uni-muenster.de/tp/fileadmin/lehre/NumMethoden/WS0910/ScriptPDE/Heat.pdf
http://en.wikipedia.org/wiki/Diffusion_equation
http://paulbourke.net/fractals/dla/
http://ritagibbs108.wordpress.com/2011/02/17/exactitude-calvino/
http://photography.nationalgeographic.com/photography/photo-of-the-day/lightning-arizona/
http://www.its.caltech.edu/~atomic/snowcrystals/photos/w031230b033.jpg

Forest Fire Modeling using Cellular Automata

Physics 210 Term Project Proposal

M. Braden Holt

Overview

- The simulation will consist of a grid of hexagonal cells, with each cell's behaviour depending on that of its 6 neighbours.
- Each cell will be occupied by a tree, burning, or empty.

Project Goals

- Improve my knowledge of programing and cellular automata.
- Write a Matlab (Octave) procedure simulating the spread of forest fires.
- Adjust the simulation to reflect the behaviour of real forest fires.

Mathematical Formulation

$$S_{ab}(t+1) = g * \Big(S_{a+\alpha, b+\beta}(t) + \sum_{(\alpha,\beta) \in V_n} [\mu_{\alpha\beta}(a, b) * S_{a+\alpha,b+\beta}(t)] + \sum_{(\alpha,\beta) \in V_d} [\mu_{\alpha\beta}(a,b) * S_{a+\alpha, b+\beta}(t)] \Big)$$

Where:

- S_{ab} is the state of a cell at (a, b).
- g is the discretization function
- $\mu_{\alpha\beta}(a,b) = \omega_{\alpha\beta}(a,b) * h_{\alpha\beta}(a,b) * r_{\alpha\beta}(a,b)$
- $\mu_{\alpha\beta}(a,b)$ is a function of:
  - wind: $\omega_{\alpha\beta}(a,b)$,
  - height/topography: $h_{\alpha\beta}(a,b)$
  - fire spread rate: $r_{\alpha\beta}(a,b)$.

Numerical Approach

- Additional variables such as the frequency of new trees, and frequency of new fires will have to be included and tested.
- Adjusting the values in the equation mentioned previously will determine the behaviour/speed of the fire.
- Graphs to show numerical data visually:
  - Propagation of fire front over time

Visualizing and Plotting Tools

- Use Matlab plotting function for:
  - Graphs mentioned previously
  - Cellular automata

Testing and Numerical Experiments

- Tested Variables:
  - Frequency that new trees appear
  - Frequency that new fires start
  - Probability of a burning tree setting a neighbour on fire (determined by S(a,b) equation)
- These variables will be varied in an attempt to make the simulation behave as a real fire.
- The length of time it takes for a tree to burn out will be equal to one tick of time in the simulation, so the rate of fire is not a tested variable

Project Timeline

Date	Action
Oct 22 – Oct 31	Research, Derive Equations, Begin Code
Nov 1 – Nov 15	Write and Implement Code
Nov 16 – Nov 20	Text, Fix, and Improve Code
Nov 21 – Nov 26	Numerical Experiments, Begin Report
Nov 27 – Dec 1	Finish Report
Dec 1 – Dec 3	Revise and Hand In Report

References

– http://laplace.physics.ubc.ca/210/Proposals-2009/02-ALL.pdf
– "Modelling forest fire spread using hexagonal cellular automata" Authors: L. Hernández Encinas, S. Hoya White, A. Martín del Rey, G. Rodríguez Sánchez ScienceDirect Applied Mathematical Modeling 31 (2007) 1213-1227

N Body Problem

Simulation of the movement of n gravitationally interacting particles

By Kamaria Kuling

Physics 210 Project
Term Proposal

Overview

Given n bodies with various initial conditions, simulate their movement due to the gravitational forces they experience.

Project Goals

Using Matlab (Octave), simulate the n-body problem

Mathematical Interpretation

Force of Gravity: $f = G\frac{m_1 m_2}{r^2}$ (1)

$$\mathbf{F}_g = -\sum_{k=1}^{N-1} \frac{G m_i m_k}{r_{ki}^3}\mathbf{r}_{ki} \quad (2)$$

Using Newton's Second Law, F=ma

$$m_i \cdot \frac{d\vec{v}_i}{dt} = \vec{F}_i, \quad (3)$$

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i \quad (4)$$

$$\frac{d\vec{v}_i}{dt} = G \cdot \sum_{\substack{j=1 \\ j \neq i}} \frac{m_j}{r_{ij}^3} \cdot \vec{r}_{ij} \quad (5)$$

$$\dot{\mathbf{v}}_i = \mathbf{a}_i = \frac{\mathbf{F}_i}{m_i} = \sum_{j \neq i} G m_j \frac{(x_j - x_i)\hat{\mathbf{i}} + (y_j - y_i)\hat{\mathbf{j}} + (z_j - z_i)\hat{\mathbf{k}}}{[(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2]^{\frac{3}{2}}}. \quad (6)$$

Numerical Approach

Using finite differencing approximations (FDAs)

f'(x)=(f(x+Δt)-f(x))/Δt

Choosing a small Δt

Testing and Numerical Experiments

- Test with various initial conditions
- Determine the accuracy of the method using laws of conservation of energy and momentum

Plotting/Visualization Tools

Matlab, for plotting, and others for the animation.

Testing and Numerical Experiments

- Test with various initial conditions
- Determine the accuracy of the method using laws of conservation of energy and momentum

Project Timeline

Dates	Activities
October 23 - November 7	Basic research, derive equations, begin code design.
November 7 - November 19	Design and implement code
November 19 - November 22	Test code
November 22 - December 2nd	Run numerical experiments, analyze data, write report
December 2nd	Finish and submit report

References

- http://www.kof.zcu.cz/st/dis/schwarzmeier/gravitational_simulation.html
- http://www.arachnoid.com/gravitation_equations/
- http://en.wikibooks.org/wiki/Astrodynamics/N-Body_Problem
- http://www.cs.utoronto.ca/~wayne/research/thesis/msc/node24.html

Questions?

Comments?

Suggestions?

Thanks!

Simulation of Toomre's Model of Galaxy Formation

PHYS 210 – TERM PROJECT PROPOSAL

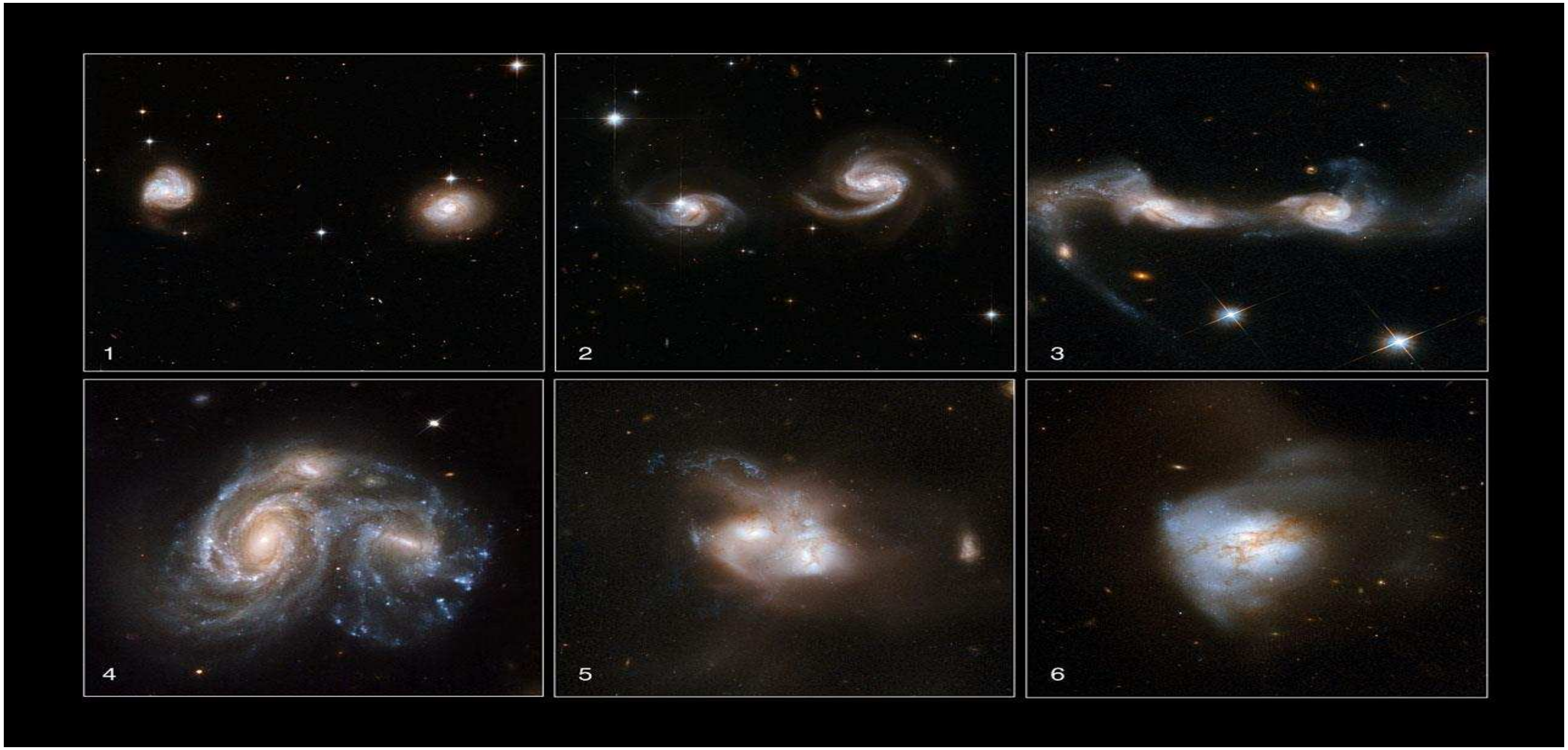
Morgan J. Maher
Oct 21st, 2013

Overview

- Will present the interaction between two galaxies by using simplified formulas to calculate the individual motion of stars around two interactive cores.
- Will make several assumptions in doing so :
  - Galaxies are spherically symmetrical configuration of mass points
  - The individual mass points (i.e. the stars) will not be altered during the collision
  - Internal energies and angular momentum of colliding galaxies will remain unchanged throughout the collision
  - The only forces acting on the system, are the gravitational interactions between individual mass points that are part of the system. In other words, the force implemented by galaxies outside of this system will not be considered
- Essentially shows the way that the gravitational attraction between two galaxies will affect their final formation and equilibrium point

Objectives

- To use Matlab to create an accurate model of galaxy collision, as was depicted by Toomre.
- To visually represent the resultant data using built-in Matlab software to create an mpeg file
- To alter the various parameters and observe how each one affects the final formation of the galaxies.



Equations of Motion

$$F_c = ma_c = \frac{mv^2}{r} = \frac{4\pi^2 rm}{P^2}$$ Centripetal force

$$F_g = \frac{GMm}{r^2}$$ Gravitational force

$$F_c = F_g \therefore \frac{4\pi^2 rm}{P^2} = \frac{GMm}{r^2} \therefore P^2 = \frac{4\pi^2 r^3}{GM}$$

Since we will essentially be looking at a binary system between the two galaxies, we get

$$P^2 = \frac{4\pi^2 r^3}{G(M_1+M_2)}$$ Thus indicating the period of motion around the center of mass of the system

$$\vec{L} = m\vec{r} \times \vec{v}$$ Orbital angular momentum

$$K = \frac{1}{2}mv^2,\ U = -\frac{GMm}{r}$$ Kinetic and Potential energy

$$\partial A = \frac{1}{2}r(r\partial\theta) = \frac{1}{2}r^2\frac{\partial\theta}{\partial t}dt$$ The area ∂A swept out by the radius vector from one mass to another in an orbital system over an infinitesimal time ∂t (derived from Kepler's Second law)

Testing & Numerical approach

- I will vary initial conditions for either one or both galaxies, including velocities, relative size, angle, shape, and position.
- I will increase the number of stars, in one or both galaxies, and observe the effects of having more mass points, as well as whether or not this change allows for a more realistic representation of the concept.
- I will attempt to plot or graph various aspects of the data in order to view different trends, some of which I may be familiar, and other that I am not
- If I have extra time, I will try to recreate the initial conditions of two galaxies which are in the process of colliding at this moment. This would provide an interesting and realistic application of the simulation, as well as allowing me to compare my results with those that have already been done

Rough Timeline

Date	Objectives
Oct 22/24	Project Proposal Presentations
Oct 29/31	Basic research, equations, and programming
Nov 5/7	Have the majority of the coding complete
Nov 12/14	Improve on different aspects of simulation
Nov 19/21	Test different parameters, and analyze the results
Nov 26/28	Complete draft, and ensure that nothing was overlooked
Dec 2	Submit final project

References

- “Non-Axisymmetric Responses of Differentially Rotating Disks of Stars” - Julian, W. H. & Toomre, A. http://articles.adsabs.harvard.edu//full/1965ApJ...141..768A/0000769.000.html
- Galaxy Crash http://burro.cwru.edu/JavaLab/GalCrashWeb/backgrnd.html
- Collisions and Encounters of Stellar Systems http://astro.berkeley.edu/~echiang/classmech/gd2_chapter8.pdf
- Astr 200, UBC – Paul Hickson, lecture notes http://www.phas.ubc.ca/~hickson/astr200/

Project Proposal for the Simulation of the Motion of N Gravitationally Interacting Particles

Physics 210
Amraaz Mangat
37747128

Project Overview

-To write a Matlab code to simulate the gravitational interactions between N particles.

Project Goals

-Using Finite Difference Approximation to create a visual representation of the interacting particles.
-Solve the N-body problem using Matlab codes.

Mathematical Formulas

- The main equation we use is by combining the equations for Newton's second law and his law of gravitation to get the equation of motion in vector form.

$$m_i \mathbf{a}_i = G \sum_{j=1, j\neq i}^{N} \frac{m_i m_j}{r_{ij}^2} \hat{\mathbf{r}}_{ij}$$

-This equation can be used to show the gravitational interaction between N particles.

Testing and Numerical Experiments

-To test if the simulation is accurate I can check if the experiment conserves energy by plotting the kinetic and potnetial energy vs. time.

-To analyze and compare data, I can simply input many different initial conditions and see if the simulation is consistent with how it should be.

Project Timeline:

Dates	Tasks
Oct. 15th - Oct. 26th	Do basic research, derive equations and begin designing code.
Oct. 27th - Nov. 15th	Implement code.
Nov. 16th - Nov. 19th	Test code.
Nov. 20th - Nov. 26th	Run numerical experiments, analyze data and begin report.
Nov. 27th - Nov. 29th	Finish project and submit

References

- http://laplace.physics.ubc.ca/210/Doc/fd/nbody.pdf
- http://en.wikipedia.org/wiki/N-body_simulation
- http://www.kof.zcu.cz/st/dis/schwarzmeier/gravitational_simulation.html
- The project proposals of the the past years

The Electrostatic interactions of N-bodies

Physics 210 Term project proposal

Fall 2013

Courtney Markin

Overview

- Coulombs law describes how two charged particles interact with one another
- However the computations become unrealistic to by hand for more than two particles
- Develop a MATLAB code which predicts how N particles of the same charge will distribute themselves in an equilibrium on the surface of a sphere

Project Goals

- To write a MATLAB code which solves the electrostatic force equation using finite differencing approximations (FDA) for N particles of the same charge on a sphere
- To establish a correct implementation of the code through convergence tests and comparison with known solutions

Mathematical formulation

$$\boldsymbol{F_{21}} = \frac{q_1 q_2}{4\pi\varepsilon_0}\frac{\widehat{\boldsymbol{R_{21}}}}{|\boldsymbol{R_{21}}|^2}$$

Electrostatic Force

$$\boldsymbol{F_{21}} = -\nabla U_{21}$$

Relationship to potential energy

$$U_{21} = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{|\boldsymbol{R_{21}}|}$$

Potential Energy

Numerical Approach

- The electrostatic force equation will be discretized using first order FDA's
  - Form a lattice with respect to x,y,z and t

Testing and Numerical Experiments

- Test by comparing to conservation of energy laws
- Compare using the Convergence test
- Test with different initial conditions

Project Timeline and References

Dates	Activities
Oct 21-27	Research and design code
Oct 27-Nov4	Design and Implement Code
Nov 4-13	Test Code
Nov 13-18	Run Numerical Experiments/ Begin Report
Nov 18-25	Analyze Data, Continue Report
Nov 25-30	Finish Report
Dec 1st	Free Day
Dec 2nd	Hand in Report

Thank you.

Any questions, comments or concerns?

Term Project Proposal: Modelling 1-D Traffic Flow with Cellular Automata

Kevin Martin

University of British Columbia

kevpmart@gmail.com

October 21, 2013

Overview

History of Cellular Automata

- The idea of Cellular Automata was first by both John von Neumann and Stanislaw Ulam (independently) in the 1940's [4] [3].
- Stephen Wolfram would later conduct detailed research on 1-D cellular automata in the 1980's leading to the now standard description of elementary cellular automata [5].

What are Cellular Automata?

Cellular automata are an example of a discrete dynamical system (in all of space, time and the cellular states). A lattice of individual cells, each having any one of finitely many sates, form the system. A series of local rules determines the time evolution of each cell; The next state of a given cell is only dependant on its own state and that of it's neighbours one time step previous [1]

Goals for this Project

- Write MATLAB code that models linear one lane, traffic flow (a 1-D lattice) with periodic boundary conditions.
- Modify above code to model a two lane traffic flow model (a 2-D lattice), again with periodic BCs.
- Investigate various starting configurations and traffic densities.
- Compare the these models with data from traffic conditions on (hopefully) a local roadway to establish the validity of this model.

Local Rules for 1-D Model

In order for this model to be physical four basic rules must be established. If there are n cells in our model the position of a vehicle is given by

$$x_i(t+1) = x_i(t) + v_i(t) \mod n, \qquad \forall\, i \in \{1, \ldots, n\}.$$

Here the mod n is a result of the periodic BCs $x = n + 1 \equiv 1$. Assuming that the "roadway" has a maximum velocity v_{max} the rules are defined:

These rules are defined [2]:

Acceleration: if the velocity v_i of a "vehicle" has $v_i < v_{\max}$ and there is more that $v_i + 1$ spaces between said vehicle and the nearest in front of it then it speeds up one unit $[v_i(t) \rightarrow v_i(t+1) = v_i(t) + 1]$.

Following Distance: If the distance between a vehicle at cell i and nearest vehicle (at cell j) in front of it is less than its velocity v_i then it "slows down" to $v_i = i - j - 1$.

Randomization: To account for the human nature or drivers, at some probability p the velocity of each vehicle will decrease by one unit $[\mathcal{P}(v \rightarrow v-1) = p]$.

Vehicle Motion: Each vehicle is advanced v, its velocity, cells.

The Simulation Algorithm

After all of the initial conditions are entered (road "length" n, maximum speed $v_{\max}$, driver probability p and initial distribution $\vec{x}$) the iteration steps involved in the simulation will be as follows:

1. Measure follow distance: $d_i = x_i - x_{i+1}$.
2. Determine acceleration: $v_i = \min\{v_{\max} v_i + 1\}$.
3. Determine follow distance: $v_i = \min\{d_i, v_i\}$.
4. Factor in driver randomization: $v_i = \max\{0, v_i - 1\}$ with p probability.
5. Movement: $x_i(t + 1) = x_i(t) + v_i(t)$.
6. Update visual output.

Numerical Experiments

- Run the simulations for various numbers and densities of vehicles in the system.
- Use various starting configurations, e.g. evenly spaced, large cluster, ect.
- Determine the maximum density where the traffic flow stabilizes.
- Compare against real life traffic data.

Timeline

Date	Activity
Oct 14 - 20	Do basic research, write proposal
Oct 21 - 27	Learn underlying theory, begin program design
Oct 27 - 31	Write initial Code
Nov 1-8	Run simulations with various realistic conditions.
Nov 9 - 15	Analyze data, attempt at 2-D model
Nov 15- 21	Begin final report
Nov 21 - 27	Finish up final report, get it proofread
Nov 31	Submit final report (due Dec 2)

[1] Hurd, Lyman, *Generalities: CA FAQs* Online. URL: http://cafaq.com/general/index.php Oct 16, 2013.

[2] Nagel, K. Schreckenberg, M. *A cellular automaton model for freeway traffic*. Journal de Physique I 2 (12) (1992), pp. 2221

[3] Pickover, Clifford A, *The Math Book: From Pythagoras to the 57th Dimension, 250 Milestones in the History of Mathematics*. Sterling Publishing Company, Inc. 2009, p. 406.

[4] von Neumann, John, *The general and logical theory of automata*. Cerebral Mechanisms in Behavior, John Wiley & Sons, New York, 1951, pp. 1-31.

[5] Wolfram, Stephen, *Statistical Mechanics of Cellular Automata*. Reviews of Modern Physics 55 (3), 1983, pp. 601644.

Questions?
Comments?

The End

ELECTROSTATIC INTERACTIONS OF N CHARGES ON A SPHERE MODELED USING FINITE DIFFERENCE APPROXIMATIONS

Physics 210 Project Proposal

Kendall McIntyre

Overview

- When bound to a spherical surface, N charges will seek to minimize their electric potential energy by maximizing their distance from each other. This relationship is given by Coulomb’s Law

Project Goals

- Write a Matlab code that correctly predicts the electrostatic interactions between N charges on a sphere and subsequently visually portrays those interactions on a 3D model
- To compare the results of the equilibrium structures against known results and structures
- To investigate the effect of various different initial conditions on the eventual result

Mathematical Equations

- Coulomb's Force

$$f(r) = q\sum_{i=1}^{N} \frac{q_i}{4\pi\varepsilon_0} \frac{r - r_i}{\left|r - r_i\right|^3}$$

  - Where f(r) is the electric force acting on charge q exerted by N particles taken

- Electric Potential Energy

$$U = q\sum_{i=1}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_1}{\left|r - r_i\right|}$$

Numerical Approach

- Using finite differencing approximations, the previously mentioned derivatives will be solved for to determine the force acting on each charge such that electric potential energy is minimized.
- All charges will have a value of either + or – 1

Visualization and Plotting Tools

- Analyze the lattices that form and how they vary with increasing values of charged particles
- A visual representation of the data will be created using techniques that shall be learned in the future

Project Timeline

Dates	To Do
October 15th – October 26th	Research and work out equations. Begin code design
October 27th -November 15th	Construct Code
November 16th – November 20th	Test Code
November 21st – November 30th	Analyze Results and Begin Report
December 1st – 2nd	Finishing Touches
December 2nd	Submit Project

References

- http://teacher.nsrl.rochester.edu/phy122/Lecture_Notes/Chapter26/Chapter26.html
- http://farside.ph.utexas.edu/teaching/em/lectures/node28.html
- http://en.wikipedia.org/wiki/Electrostatics

Forest Fire Spread Simulation Using Cellular Automata

Physics 210 Term Project Proposal

By Cameron Metcalfe

Overview

- The movement of forest fires can be predicted
- Cellular Automation is a useful tool

Project Goals

- Create a MATLAB program to simulate the spread of a forest fire
- Add additional variables to create a more realistic event simulation
- Test a variety of initial conditions
- Observe and evaluate the affected terrain

Mathematical Formulation

- The cells will be stored in the elements of a matrix

$$C^{(k)} = \begin{pmatrix} a_{0,0}^{(k)} & \cdots & a_{0,z-1}^{(k)} \\ \vdots & \ddots & \vdots \\ a_{r-1,0}^{(k)} & \cdots & a_{r-1,z-1}^{(k)} \end{pmatrix}$$

- The probability of a cell being burnt depends on the 8 surrounding cells
- Using the Moore neighborhood model

Indexing

(i-1,j-1)	(i,j-1)	(i+1,j-1)
(i-1,j)	(i,j)	(i+1,j)
(i-1,j+1)	(i,j+1)	(i+1,j+1)

Mathematic Formulation

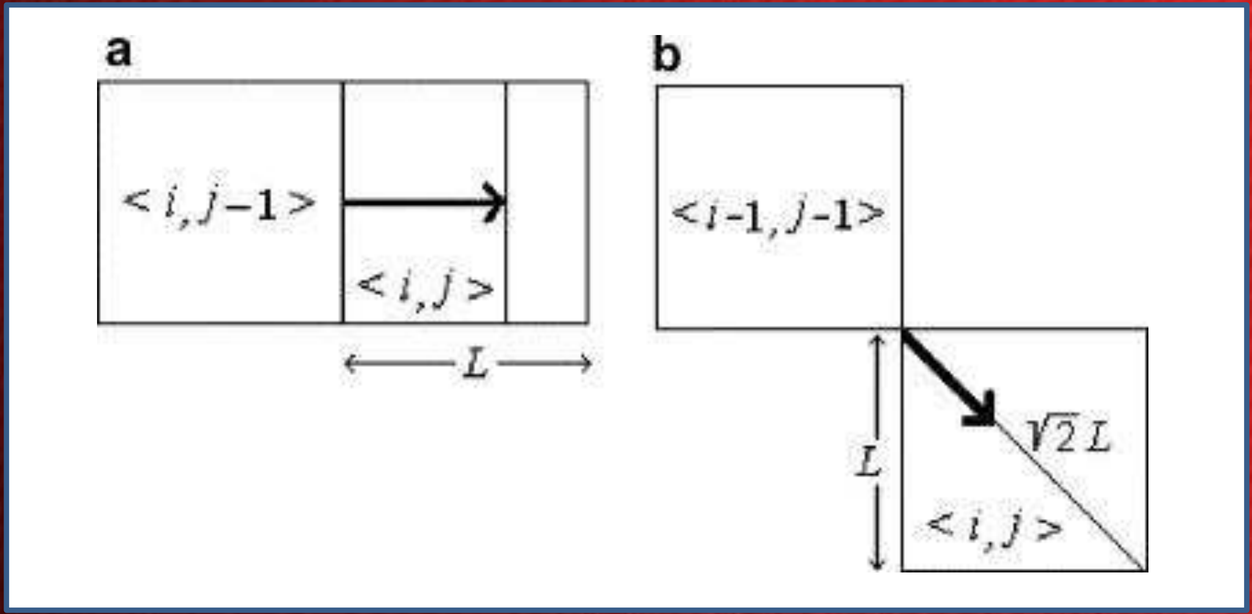
- The state of a cell at time t + 1 is determined by a function of the surrounding cells

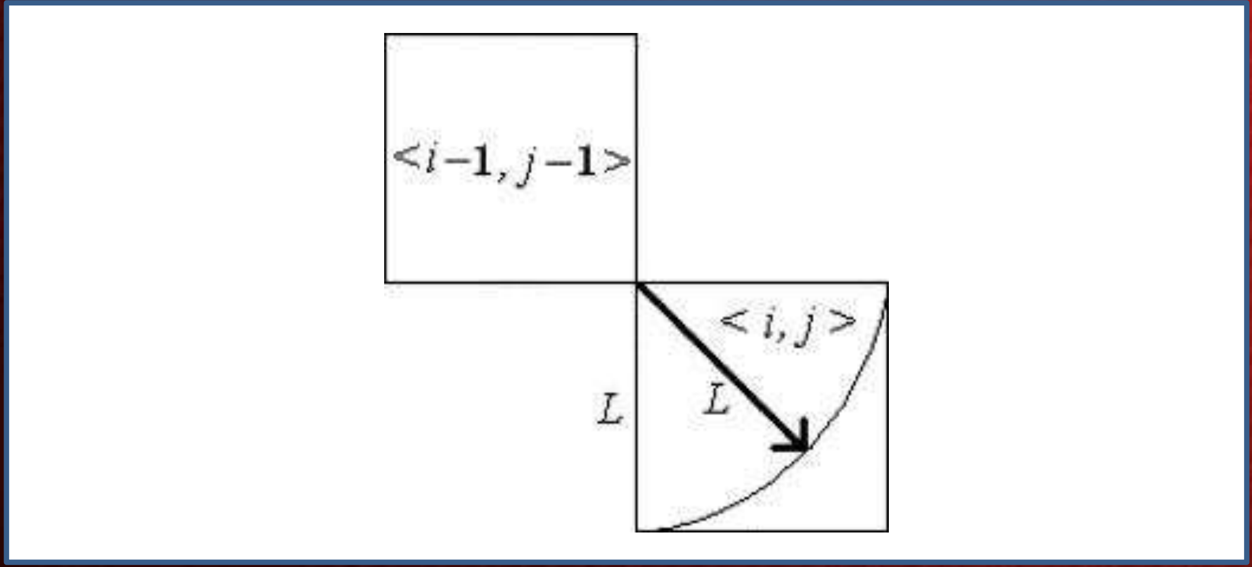
$$a_{ij}^{(t+1)} = f\left(a_{i+\alpha_1, j+\beta_1}^{(t)}, \ldots, a_{i+\alpha_8, j+\beta_8}^{(t)}\right),$$

$$a_{ij}^{(t+1)} = \sum_{(\alpha,\beta)\in V_M} \mu_{\alpha\beta}\, a_{i+\alpha, j+\beta}^{(t)},$$

$$a_{ij}^{(t)} = \frac{\text{burned out area of } (i, j)}{\text{total area of } (i, j)},$$

- Basic vs. Circular diagonal cell influence





Numerical Approach

- The addition of variables such as wind and height affect the parameters of the model
- Linear systems form the basis of cellular automata

Visualization and Plotting Tools

- Use of MATLAB plotting
- Use of xvs for simulation purposes

Testing and Numerical Experiments

- Start by simulating fire with one point of origin on a uniform terrain
- Move on to multiple origins as well as possible weather variables such as wind
- Compare the results of my simulation to the spread of actual historical forest fires

Project Timeline

Dates	Goals
October 25 – 31	Additional research on mathematical methods
November 1 – 4	Begin coding in MATLAB
November 5 – 10	Test code thoroughly
November 11 – 16	Add additional variables such as wind and test again
November 17 – 29	Work on report including analysis of data
November ~	Present analysis
November 30	Submit Report

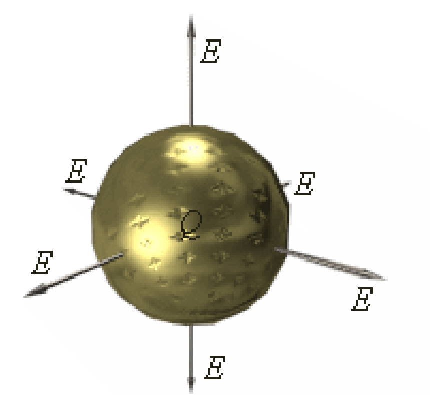
References

- Hernandez Encinas et al, Simulations of forest fire fronts using cellular automata, Elsivier, (2006)
- http://www.sciencedirect.com/science/article/pii/S0965997806001293
- http://people.bath.ac.uk/jpc25/M126website/planning.html

Simulation of the electrostatic interaction of N-particles in 3D using Finite Difference Approximation

PHYS 210 Term Project Proposal

Yousef Mirza



Overview

- Particles with like charges on a sphere are initially placed at arbitrary positions with initial velocity = 0. Then, overtime they disperse to their equilibrium position

- This equilibrium position is determined by the Coulomb inverse-square law.

- Will need to account for a dissipation in the system since the charges come to a stop.

Goals

- Write MATLAB codes and use FDA's to solve for the differential equations that describe this interaction.
- Use MATLAB to simulate the behaviour of n-charges in 3D, so Cartesian components will be needed.

Mathematical Functions

- The coulomb's law can be written in the form

$$m_i\,\mathbf{a}_i = m_i \frac{d^2\mathbf{r_i}}{dt^2} = -k_e \sum_{j=1,\,j\neq i}^{N} \frac{q_i\,q_j}{r_{ij}^2}\,\hat{\mathbf{r}}_{ij} \qquad 0 \leq t \leq t_{\max}$$

where i= 1, 2, …. N.

Note: K is negative since it's a repulsive force.

- Since the choice of R(radius) is arbitrary and won't effect the equilibrium position of the N charges, we can simply set R=1, so that

$$r_i \equiv |\mathbf{r}_i| \equiv \sqrt{x_i^2 + y_i^2 + z_i^2} = 1$$

Mathematical Functions (continued)

- Since I will be using equal mass and equal charge, it will be convenient to non-dimensionalize, so then

$$m_i = 1 \ , \ i = 1, 2, \dots N$$

$$q_i = 1 \ , \ i = 1, 2, \dots N$$

$$k_e = 1$$

- I will also need to add some friction to the system, so the charges settle, and this can be done by using a parameter which is proportional to velocity.
- These two things would therefore simplify the coulomb Equation to

$$\mathbf{a}_i = -\sum \frac{\hat{\mathbf{r}}_{ij}}{r_{ij}^2} - \gamma \mathbf{v}_i$$

Numerical Approach.

- The simplified coulomb equation, then can also be written in the form

$$\frac{d^2 x_i(t)}{dt^2} = -\sum_j \frac{(x_j - x_i)}{r_{ij}^3} - \gamma \frac{dx_i}{dt}$$

Where the y and z components have the same form.

- This equation can then be discretized using the finite difference technique. We can use the second-order centred formula and the centred approximation for the first derivate which then gives the discretized equation.

$$\frac{x_i^{n+1} - 2x_i^n + x_i^{n-1}}{\Delta t^2} = -\sum_j \frac{\left(x_j^n - x_i^n\right)}{\left(r_{ij}^n\right)^3} - \gamma \frac{x_i^{n+1} - x_i^{n-1}}{2\Delta t}$$

- To solve this system of equation, I will use multi-dimensional arrays to store discrete positions.

Visualization, Testing and Numerical experiments.

- Visualization tool will be MATLAB, and I will need to increase the number of particles to get more complex structures
- Since I know where the charges will be on the surface of the sphere, I can check if I am computing the equilibrium positions correctly.
- Will need to experiment with the adjustable parameter of friction, **γ**, when I implement my codes.

Project Timeline Reference

Dates	Activities
Oct.17 to oct.27	Grasp the whole idea, derive equations and begin code design
Oct.28 to Nov.15	Implement codes.
Nov.15 to Nov.20	Test codes
Nov.20 to Nov.26	Run numerical experiments, analyze data and start the report.
Nov.26 to Nov.29	Finish Report.
Nov.29	Submit the project.

References

http://www.math.umn.edu/~olver/pd_/nfd.pdf

http://laplace.physics.ubc.ca/210/Doc/fd/nbody.pdf

Questions?
Comments?
Suggestions?

Tracing rays using mathematical operations with arrays.

PHYS 210 TERM PROJECT PROPOSAL
ARMAN NOOR

● Overview

- Mathematical operations with arrays allows the user to perform complex linear algebra calculations.

-These calculations include dot product and cross product of two vectors in 3-D space which becomes useful to find the angle between two rays, distance between two points, and by knowing the speed of light, we can calculate the time and therefore, speculate the path of the rays.

● Project Goals

- To write a MATLAB code using mathematical operations with arrays to trace rays in 3-D space.

- To essentially determine a pattern for the movement of rays through different mediums at different angles.

- To predict the movement of the ray knowing a theoretical location of its starting point.

- To investigate any unexpected behaviour in the movement path of the rays (preferably, a ray moving between two infinite parallel plates with different initial starting position of the ray)

- Mathematical formulation

- The most frequent equations that I will be dealing with will be:

-Dot product or cross product to find angles between two lines:

dot product:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta,$$

cross product:

$$A = \|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta.$$

Calculating distance between two points A to B, where A and B are 3-D vectors
:

||B-A||

Using the formula d=vt (and knowing the speed of light), we can calculate the time it takes for the light to move from one point to the other.

- Numerical approach

- Through my investigation with the light rays and the path it takes, I will hopefully be able to create and equation which predicts the path of the rays motion. Using this equation, I can determine different characteristics of the ray at any given point in its movement path. These characteristics include time of movement from initial point, direction and distance travelled. I can also if desired, decide a stopping point for the ray to then determine when the light ray will reach its final destination from the given point. I will attempt to find this equation for the pattern of the ray through its movement and especially through its behaviour when reflected off of the walls of the medium.

Note: The medium in this case allows the ray to make perfect reflections on its walls without affecting the speed of the ray.

- Visualization

Using matlab, I can create the path that the ray will take graphically on a diagram as i have the direction of the ray and the path that it takes (also I am hoping to graph the final equation and get the path of the ray)

Testing and numerical experiments

- Testing

-After having found my equation for the predicted path of the ray, I will use a series of different points that the ray passes through to and use it in my equation to see if the characteristics of those points are the same as the characteristics of those points as calculated using step-by-step calculations (these calculations will involve finding distance between points and the time between points)

- Numerical Experiments

- I will experiment with rays moving with different initial starting positions and different initial movement directions to see how it will affect the progressive path of the ray. This will allow me to understand the rays movement behaviour more thoroughly and thus, make a more suitable equation for the predicted ray path. Hypothetically speaking however, I believe that the equation for the predicted ray motion will be a little different depending on the initial direction and starting motion of the ray. I might also changed the distance between the two plates that I am tracing this ray through to see how it will affect my results.

- # Project Timeline

Date	Activities
22/10/2013-26/10/2013	Basic research, overall understanding, and finalized plan for the program coding
27/10/2013-15/11/2013	Implementation of code
16/11/2013-19/11/2013	Code testing
20/11/2013-26/11/2013	Run numerical experiments, analyze data, begin report
27/11/2013-29/11/2013	Final editing
30/11/2013	Hand in Project

References:

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

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

Questions?

Comments?

Suggestions?

Thank for listening!

THE EXTREMELY ORIGINAL "N-BODY PROBLEM" GRAVITATIONAL INTERACTIONS USING FINITE DIFFERENCE APPROXIMATIONS

PHYS 210 Term Project Proposal

Taryn Nowak-Stoppel

Overview

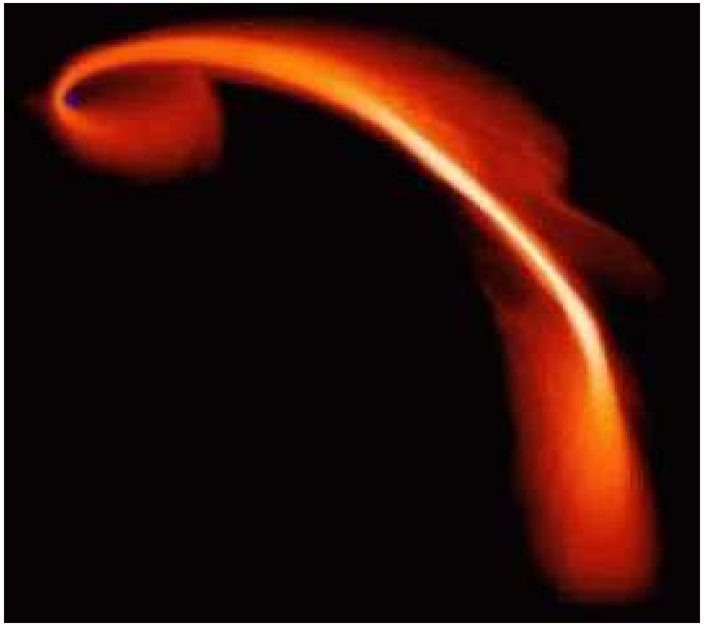
- Simulation of test particles approaching a black hole, or other interstellar object with a high gravitational field
- n test particles will end differently depending on it's initial trajectory; knowing initial velocities and positions, final velocities and positions can be found after time t

1.

2.

3.

4.



Project Goals

- Solve the n-body problem in MATLAB in 3 dimensions, using finite difference approximations
- Test different initial conditions for n

Math Math Math Math Math

$$F = \frac{\left(-GmM(r-R)\right)}{|r-R|^3}$$

Gravitational Interaction between 2 particles

Numerical Approach

- FDA:

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

Testing and Numerical Experiments

Testing

- Try simulation out to confirm there are no bugs
- Examine simulation and compare to other recent models

Numerical Experiments

- Investigate interactions using varying initial conditions

Project Timeline

- 24/10/13: present project proposal
- 1-8/11/13: write and test code
- 9-12/11/13: run experiments and collect data
- 13-28/11/13: analyze data, write out final report
- 29/11/12: submit project

References

- http://bh0.phas.ubc.ca/210/Doc/term-projects/kdv.pdf
- http://laplace.physics.ubc.ca/210/Proposals-2012/L1A-All-Proposals.pdf
- Matthew Choptuik's teachings via PHYS 210 lectures and labs
- http://en.wikipedia.org/wiki/N-body_simulation

Traffic Simulation Using Cellular Automata

PHYS 210 Term Project Proposal

Sarah Parry

Overview

- A simulation of the movement of single lane traffic using cellular automata.
- A cellular automatan is model in which cells in the grid interact with each other in a finite number of ways.

Project Goals

- To write a MATLAB code which simulates the movement of single lane traffic.
- To observe the behaviour of traffic in the simulation.
- To observe the formation of traffic jams in the simulation.

Approach

- A car on the grid can behave in 3 different ways:
  - Acceleration
  - Maintaining speed
  - Braking
- These behaviours are determined by several factors including:
  - Whether the space directly in front of the car is empty or filled.
  - How many spaces in front of the car are empty.
  - Probability.
  - Randomness

Testing

- Vary initial conditions (max speed, density, etc…) to observe effect on simulation.
- Determine ideal conditions for avoiding traffic jams.

Timeline

Week 1	Research and Begin Code
Week 2	Implement Code
Week 3	Test Code and Improve Code
Week 4	Run Experiments
Week 5	Analyze Data and Start Report
Week 6	Finish Report

References

http://www.stephenwolfram.com/publications/academic/cellular-automata.pdf

http://sjsu.rudyrucker.com/~han.jiang/paper/

http://www.academia.edu/877411/Real-Time_Traffic_Simulation_Using_Cellular_Automata

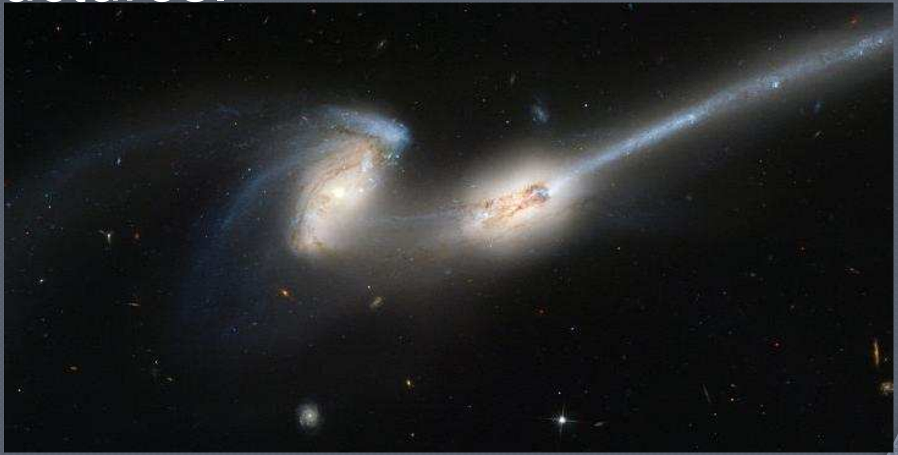
Toomre model of galaxy collisions

Phas210 Introduction to Computational Physics (Fall 2013)

Jolanta Peplinska
24th October 2013

Overview

- **Alar Toomre** – Focused his research on the dynamics of galaxies, and he was the first to conduct computer simulations of them merging.
- There are different types of galaxies, based on shape, however some did not fit the criteria, they were found peculiar, and Toomre's simulations managed to reproduce some of these structures.
- Toomre's model is a simplification as it ignores the interstellar medium and dark matter.



- Only the cores and stars surrounding them are considered in Toomre’s model.
- Gravitational fields of the galaxies result in them disturbing one another. They will pull in the stars, from the discs surrounding the cores, to form broad fans.
- It is now believed that galaxies are constantly interacting and the bigger ones engulf the smaller ones and get even bigger.

Project goals

- To write a MATLAB code which simmulates galaxy collisions, using the Toomre model.
- To look at how changing different variables will effect the collision e.g.: the galaxies passing each other at different distances, different masses of the galaxies, different approach of the galaxies.
- To see how well the Toomre model fits, actual results (e.g. compare with more complex simulations)
- Create snapshots, of the different stages of the collisions, and see if the shape of the galaxies is simillar to some of the peculiar ones.
- Try and simulate the Andromeda-Milky Way coliision.

Mathematical Formulation

- In this simplified model, newton's law of gravitation will be used:

$$F = G\frac{m_1 m_2}{r^2} \qquad F = ma = \frac{mv^2}{r} \qquad a = \frac{Gm_1}{r^2}$$

- Two galactic nuclei will be moving under their mutal grvitational attraction, using Newton's law of gravity.
- The nuclei will be surrounded by point-like stars, each with a mass, m, which will only experience the gravitational attraction from the galactic nuclei.
- We can also use Kepler's 3rd law for the stars

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

Testing and Numerical experiments

- The positions, masses, shapes, velocities and angles will be varied for the galaxies.
- And the results compared.
- I will repeat the simulations many times, to see if the results are consistent.
- I will compare my simulations with those I can find online.
- I will look at the structures formed by the two galaxies and see if they look similar to the peculiar shapes.
- I will try and keep the amount of stars surrounding the galaxy cores to a maximum, so that the simulation runs smoothly.

Numerical approach

- The aim will be to calculate the positions of the nuclei of the two galaxies, and the positions of the stars, during a series of steps, which will be seperated by an interval Δt.
- The positions will be determined by the forces acting on the bodies.
- I will also try and use the Barnes-Hut algorithm – it puts particles, which are sufficiently close enough to each other, into groups.
- The finite difference approach could possibly also be used.
- Visualization
- I will attempt to create mpeg files using the built in MATLAB visualization software.

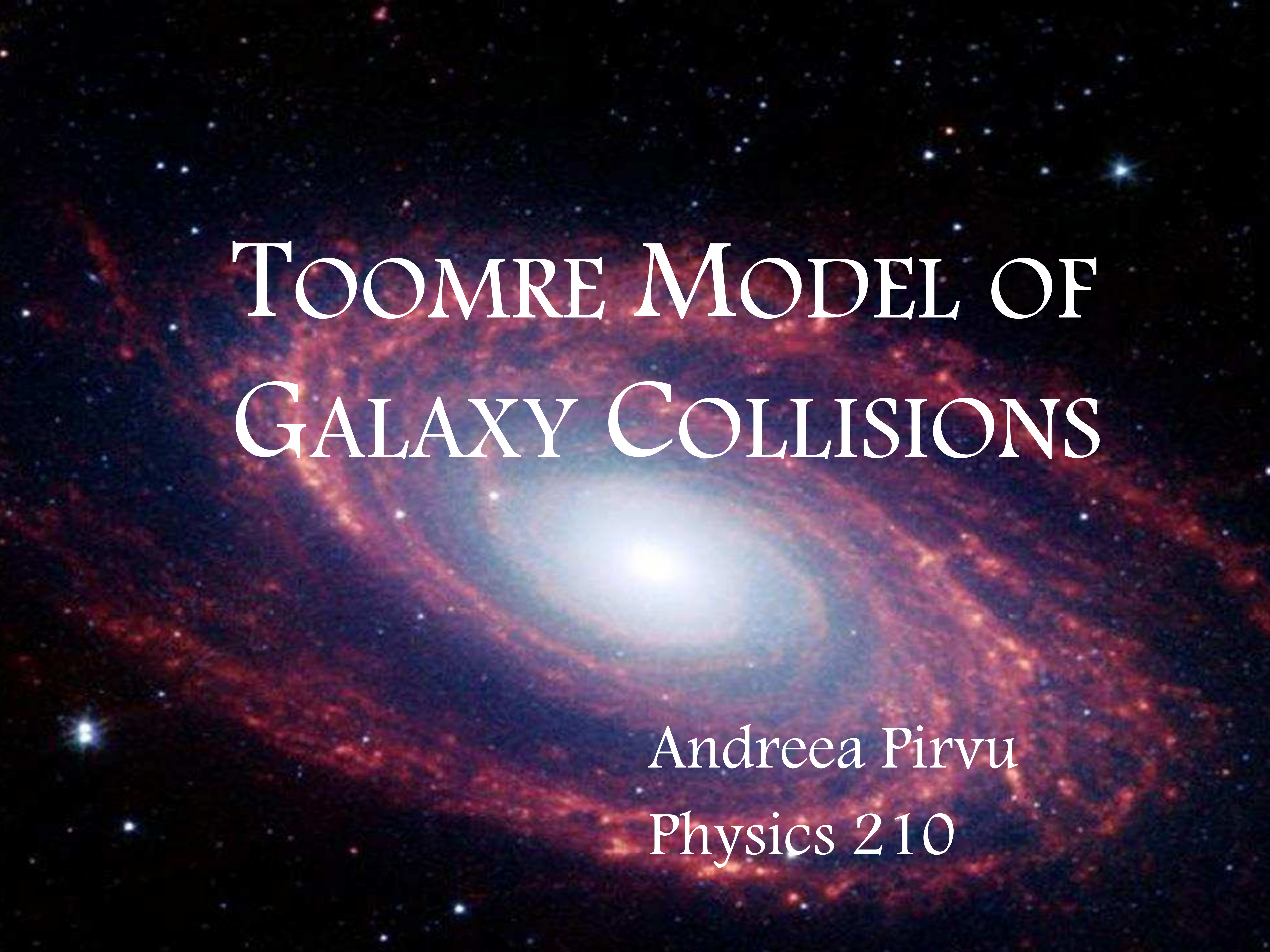
Project Timeline

I will be writing the report alongside the different activities

10/21-10/28	Research the topic, determine all the equations needed, begin code design
10/29-11/15	Implement code
11/16-11/20	Test code
11/21-11/24	Numerical experiments, analyze data
11/25-12/01	Finish report and submit

References

- http://en.wikipedia.org/wiki/File:NGC4676.jpg
- http://faculty.etsu.edu/smithbj/collisions/collisions.html
- http://users.monash.edu.au/~adonea/GALAXIES_ASP2062/LAB_MERGERS/lab.pdf
- http://arborjs.org/docs/barnes-hut



Toomre Model of Galaxy Collisions

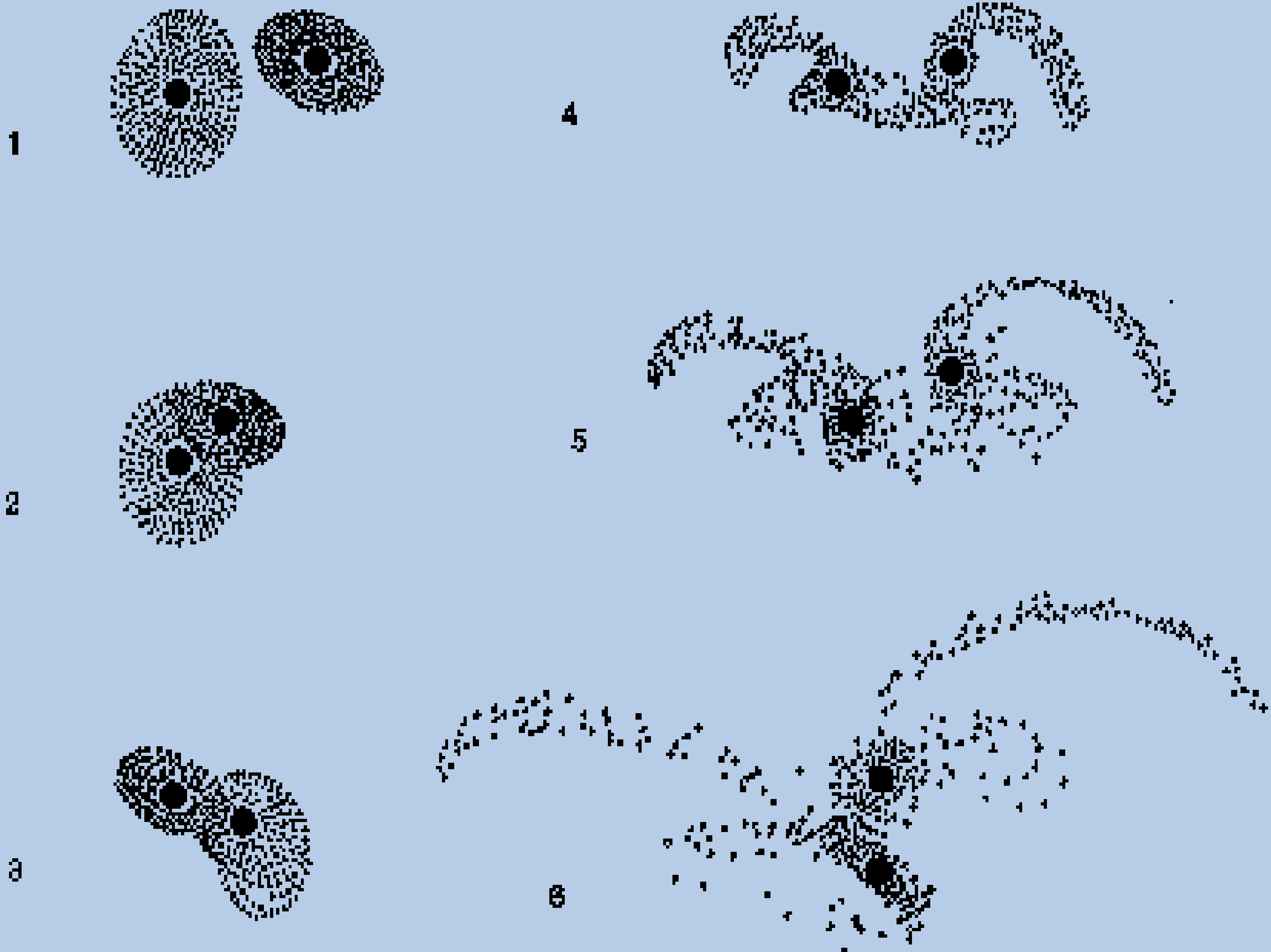
Andreea Pirvu
Physics 210

Overview

- Galactic collisions are very common in the evolution of galaxies.
- Colliding galaxies are galaxies whose gravitational fields result in the disturbance of one another.
- Do these galaxies literally "collide"? No!
- When these celestial bodies hit, they merge!

Overview continued

- In the 1970's Alar and Juri Toomre were able to illustrate the collision of two galaxies.
- Their model was very crude but accurate
  - Due to limited computing power only 1000 stars were used, while galaxies have billions
  - Interstellar gas and dark matter were ignored
- They observed large tidal tails; long steaks of stars spun off by gravity


1
2
3
4
5
6

Goals

- To write a matlab/octave code that depicts the collision of two galaxies using Toomre assumptions.
- To investigate different initial conditions such as mass of stars, angle, velocity etc.
- To try and produce effects similar to that of an actual galaxy interaction
- Maybe depict the interaction between the Milky Way and Andromeda.

Mathematical Formulation

- Newton's law of graviation:

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

  - m1 being the mass of the galactic center and m2 being the mass of a star.
- Given F=ma, we get:

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

- Kepler's law

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

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

Numerical Approach

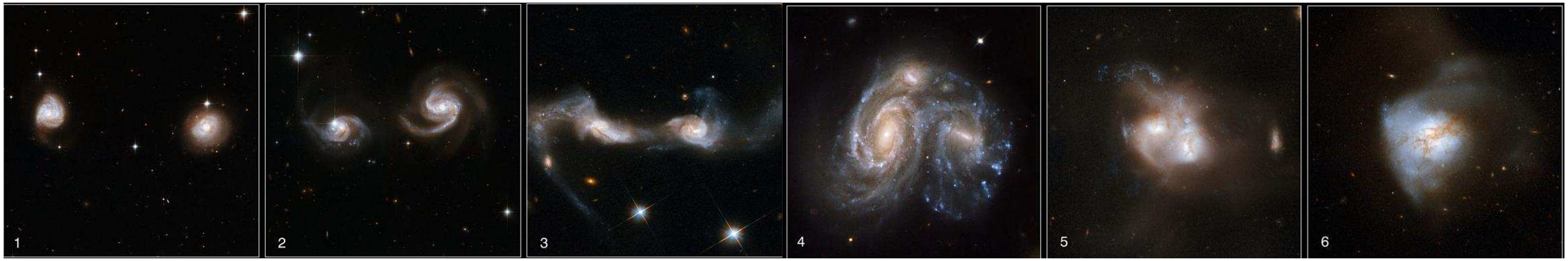
- Dark matter, interstellar medium and black holes are ignored.
- The mass of stars will all be the same.
- Gravitational forces between stars are negated
- Consider vectors in 3D. The final and initial velocities and positions will be in three dimensions:

$$a = (a_x, a_y, a_z), d = (d_x, d_y, d_z)$$

- Use of finite difference approximations
- This is still a work in progress! Finding the exact formulae is my job for this week.

Testing and numerical experiments

- Once the code is working, I hope to experiment with different initial conditions.
  - I will alter angles, velocity, distance, angles and mass of the galaxies
  - Attempt to replicate the (hypothetical) Milky Way-Andromeda collision.



Visualization

- Use Matlab for plotting and animations
- Use XVS for mpeg animations if needed

Timeline

Dates	Activities
10/21-10/28	Research and derive equations
10/29-11/15	Design and implement code
11/16-11/19	Test code
11/20-11/26	Run experiments, analyze data and begin report
11/27-11/29	Finish report
11/30	Hand in!

References

- http://astro.berkeley.edu/~echiang/classmech/gd2_chapter8.pdf
- http://sciencenotes.ucsc.edu/9701/full/features/galaxy/Toomre.html
- http://curious.astro.cornell.edu/question.php?number=351
- http://astronomy.swin.edu.au/cosmos/T/Toomre+Sequence

Equilibrium of N charged partials on a sphere

Sam Ramsey – October 24, 2013

Overview

- An arbitrary number of like charged partials on the surface of a sphere will repel each other by Columbus law.
- An equilibrium will be achieved when the lowest potential energy of the system is obtained.

Project goals

- To write a program in MATLAB(octave) that calculates the equilibrium position of N charged particles on a sphere.
- To test the simulation by comparison to known equilibrium positions of initial conditions.
- To study the equilibrium positions that result from varying the initial conditions.

Mathematical Formulation

- The vector form of coulombs equation for N charged particles,

$$F = \frac{q}{4\pi\epsilon_0}\sum_{i=1}^{N}\frac{q_i(r - r_i)}{|r - r_i|^3}$$

- The force due to friction will be given by,

$$F_f = -\gamma v$$

- The acceleration of the particle due to these forces is given by Newton's second law,

$$F = m\frac{dv}{dt}$$

Numerical Approach

- To simplify the problem the sphere will have a unitary radius.
- A finite difference approximation will be use for the forces acting on each particle.
- Initial conditions will be specified or generated at random.

Testing and Numerical Experiments

- Initial conditions will be varied to see how this will effect the equilibrium positions of the particles.
- The program will be tested by comparing the results to known equilibrium positions of N particles.

Project Timeline

Date	
Oct 15 – 26	Research and design code
Oct 27 – Nov 15	Write code
Nov 16 – 19	Test and debug code
Nov 20 – 26	Run experiments, analyse data, start report
Nov 27 – 30	Finish and submit report

TRAFFIC SIMULATIONS USING STOCHASTIC CELLULAR AUTOMATA

Yuliang (Kevin) Shi

OVERVIEW

- Cellular Automaton analysis is operates on a regular grid (or array) of **cells**, with each cell taking on one of a finite number of states.
- In this case each cell will represent a short section of pavement enough to fit a generic automobile, a list of such cells will represent a length of road. A cell can have one of two states, occupied or unoccupied.
- For each cell, the cells in its immediate surroundings are defined as its neighborhood

OVERVIEW

- Following some initial conditions, cellular automata work according to a given set of rules that dictate the new state of each cell based on its current state and its neighbourhood of cells.
- These rules are applied iteratively over a some number of time-steps, it is key to make at least one of such rules probabilistic (e.g. cars randomly changing speed) so that results are not deterministic.
- In our case, there would be some initial distribution of occupied/unoccupied cells corresponding to cars on a street and spaces between them.
- The time steps will progress and rule will dictate the movement of cars in this virtual universe.

PROJECT GOALS

- Analyse characteristics of traffic flows in various situations
  - Traffic Circle
  - Bottleneck
  - More complex cases
  - I really haven’t decided yet

MATHEMATICAL FORMULATION

- In any possible case, one of the key pieces of information we care about is the flow rate of traffic as function of traffic density
- One way to measure this in our simulation is:
- For some cell i:
  - $density = \frac{1}{T}\sum_{t_0}^{T} d_i(t)$
  - Where T is some overall time period, t is some timestep and t_0 is the initial timestep, and $d_i(t)$=1 if the cell is occupied at time t and 0 if unoccupied

MATHEMATICAL FORMULATION

- For some cell i:
  - $time - average\ flow = \frac{1}{T}\sum_{t_0}^{T} f(t)$
  - Where T is some overall time period, t is some timestep and t_0 is the initial timestep, and f(t)=1if car motion is detected over this cell at some time t.
  - Numerical approach, Visualization and Plotting yet to decide

TESTING AND NUMERICAL EXPERIMENTS

- The rules we set up in our automaton and different traffic densities, reflect real-world traffic regulations and peak/off peak road conditions, respectively. Therefore, by manipulating these variables we can for various traffic situations we can:
  - Gain insight into optimal traffic densities to achieve peak flow-rate
  - What the optimal traffic regulations are (speed limit etc) to obtain the best flow-rate for some traffic density
  - etc

TIMELINE

- 10/20-10/31 Do basic research, derive equations & begin code design
- 10/31-11/15 Implement code
- 11/16-11/19 Test code
- 11/20-11/26 Run numerical experiments, analyze data, begin report
- 11/27-11/29 Finish report
- 11/29 Submit project!
- seems familiar? I think so too

REFERENCES

- http://en.wikipedia.org/wiki/Cellular_automaton
- http://laplace.physics.ubc.ca/210/Doc/term/nagel-schreckenberg-ca-traffic-j-physique-1992.pdf

N BODY GRAVITATIONAL INTERACTION SIMULATION USING THE PARTICLE MESH METHOD

PHYS 210 PROJECT PROPOSAL

SUNDEEP SINGH (35372119)
PROF. MATTHEW CHOPTUIK

OCT 24, 2013

Overview

Many physical systems are comprised of a system of particles interacting with each other through their fundamental forces. As such, it is of consequence to simulate this large scale interaction using approximations of the basic laws of physics. And this is the purpose of this project.

Project Goals

1) To research and understand (an) algorithm(s) that effectively and efficiently model basic particle interaction
2) To choose of the multiple algorithms and implement using MATLAB
3) To place conditions on the simulation and inspect the different outcomes
4) To test the system using basic principles of physics such as Conservation of Energy and Conservation of Momentum in a closed system (non-relativistic physics, v << c)

Mathematical Formulation

Mathematically, the n-body problems idea can be formulated as follows:

(1) $U(x_0) = \sum_{i}^{n} \boldsymbol{F}(\boldsymbol{x_0}, \boldsymbol{x_i})$ Where U is the quantity that determines the motion at x_0 and F are the pairwise interactions spanning all the particles in the system with the particle at x_o

F represents the force caused by some particle at the location x_i. The gravitational force felt by a particle x of mass m due to other particles x_i with mass m_i can be expressed as:

(2) $F(x) = \sum_{i=1}^{n} Gmm_i \frac{x - x_i}{|x - x_i|^3}$ $\frac{r}{|r|^3}$ used instead of $\frac{1}{r^2}$ because the former accounts for the sign of the force

(3) $F(x) = (ma)_x$

Mathematical Formulation Cont.

F can be used to find acceleration of a particle according to (3). This can be used to find the velocity of a particle and position (and thus the complete state) by the following:

(4) $v_f = v_i + a_i \cdot \Delta t$

(5) $x = x_0 + v_i \cdot \Delta t$

v_i is used in the calculation of x as opposed to v_f because (assuming discretized time and position) the state of the particle (position and velocity) are determined by the previous state and any relevant changes. The same argument applies for why a_i is used.

As constants don't affect summations, the constant m in (2) can be factored out; so by dividing both sides by m in the resultant equation and in (3) and equating them, the following is obtained:

(6) $$a_x = G \cdot \sum_{i=1}^{n} m_i \frac{x - x_i}{|x - x_i|^3}$$

Type Check: The factors G, m, and $\frac{1}{|x-x_i|^3}$ are all scalar, leaving $x - x_i$ the only vector component. And since a_x is a vector and the sum of two vectors is a vector and any scalar multiple of a vector is a vector, the types match

Mathematical Formulation Cont.

Using (4), (5), and (6) in conjunction and splitting the individual vectors into their components yields the following:

(7) $$\{a_x, a_y, a_z\} = G \cdot \sum_{i=1}^{n} m_i \frac{\{s_x - s_{ix}, s_y - s_{iy}, s_z - s_{iz}\}}{\sqrt{(s_x - s_{ix})^2 + (s_y - s_{iy})^2 + (s_z - s_{iz})^2}^{\,3}}$$

x in (6) has been replaced by s denoting spacial coordinate to avoid confusion

(8) $$\{v_{fx}, v_{fy}, v_{fz}\} = \{v_{ix}, v_{iy}, v_{iz}\} + \{a_{ix}, a_{iy}, a_{iz}\} \cdot \Delta t$$

(9) $$\{s_{fx}, s_{fy}, s_{fz}\} = \{s_{ix}, s_{iy}, s_{iz}\} + \{v_{ix}, v_{iy}, v_{iz}\} \cdot \Delta t$$

If all the particles are set as coplanar in the xy plane, then all z components in (7), (8), and (9) would reduce to 0.

Numerical Approach

Calculating (1) by using the subsequent equations (2) through (9) is very inefficient and results in an algorithm of $O(n^2)$. U in (1) can be thought of as the potential of the particle. Let us express U as follows:

(10) $U = \nabla\varphi, \nabla U = c\rho$ ρ is the mass density (mass/area unit) of the computational area, φ is a value that we hypothesize/assign as being related U as such.

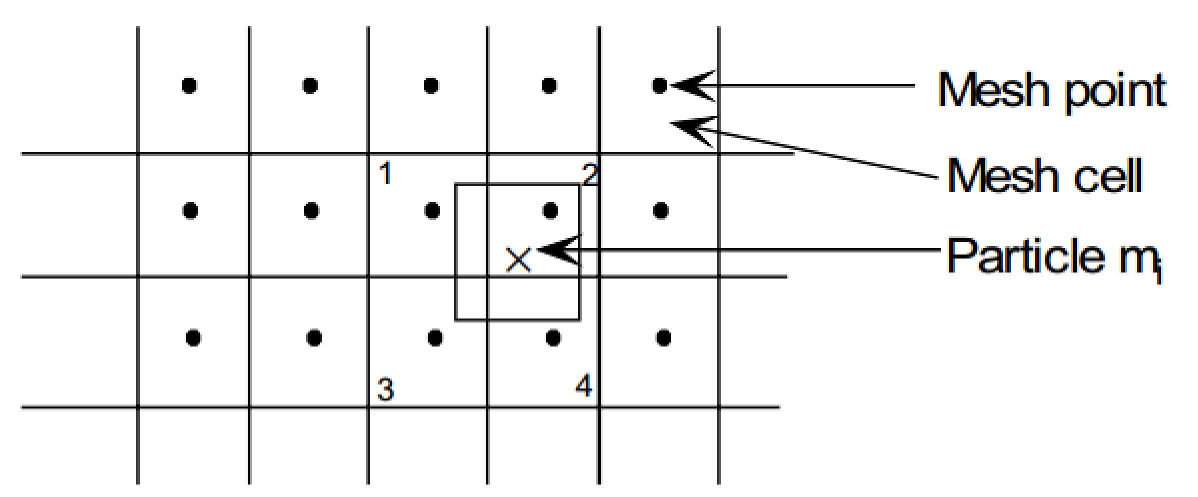
Simple substitution yields:

(11) $\nabla^2\varphi = c\rho$ This is called the Poisson equation (Tancred Lindholm)

Before anything meaningful can be done with this equation, the mesh itself must be shown

Numerical Approach cont.

The PM method overlays the computational area with a grid and grid points centered in each cell. The particle can be anywhere on the grid. This is as follows:



(Tancred Lindholm)

The easiest method to approximate ρ quickly is Nearest Gridpoint (NGP) (Tancred Lindholm). In this method the mass of each particle is simply assigned to its nearest mesh point (in the diagram, m_i would be assigned to mesh point 2). The subsequent calculations simply use the sum of all masses in the mesh cell concentrated at the mesh point in calculations, and as such as an approximation that is more accurate the smaller the mesh cell.

Numerical Approach cont.

Once ρ has been assigned values, the potential can be solved for by using Fast Fourier Transform (the exact implementation is unknown to me at this point). φ would then result in the potential at each mesh point. All particles in the corresponding mesh cell would then feel that potential (approximation), and then (1) can be used to calculate the exact motion of said particles. There is limited spatial resolution, but this is balanced by the fact that this particular algorithm can handle a large number of particles accurately and with order $O(n)$, the slowest steps being the FFT which is $O(Glog(G))$, where G is the number of grid points (Tancred Lindholm), which is a drastic improvement over simply computing (1) for every particle. The PM method can be made more or less accurate depending on the size of Δx and Δt.

Testing and Numerical Experiments

Testing

The two test that can be done are conservation of energy and conservation of momentum. Find and sum all the energies of each particle at each discrete step of the simulation, and plot vs time. Ideally, the resulting graph should have a flat line at some value, but within error the graph should stay at around the same value (the simulation is an approximation to the ideal). The same can be done for conservation of momentum, as momentum cannot change despite the collisions between particles.

Numerical Experiments

Randomly distributed particles with no initial velocity should converge to the center of mass of the entire computational area. Collisions can be turned off or made inelastic, and the initial center of mass can be labeled to see if the particles really do converge to that point. Particles given a velocity perpendicular to the vector between the particle and the center of mass of the group should ideally orbit, escape (only to hit the edge of the computational area and bounce back), or spiral into the center. This can also be modelled and checked.

Project Timeline

Dates	Tasks
Oct 28 – Nov 3	Basic Research/Start Code
Nov 4 – Nov 18	Implement Code
Nov 18 – Nov 20	Test Code
Nov 20 – Nov 24	Run Experiments/Start Report
Nov 24 – Nov 27	Finish & Submit Report
Nov 27 – Dec 2	Extra days just in case

References

Lindholm, T. (1999). N-Body Algorithms. Retrieved from http://www.cs.hut.fi/~ctl/NBody.pdf

Wikipedia. (2013, October 19). *Newton's law of universal gravitation*.
Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Newton's_law_of_universal_gravitation

N-body Simulation

Nutifafa Sumah

Overview

- The simulation aims to predict the motion and behaviour of astronomical bodies in a system based on their initial position and velocity.
- This can be solved directly for a system with only 2 bodies, however for more bodies this can only be solved approximately.

Project Goals

- Write a MATLAB/octave code to determine the behaviour of a group of objects based on specific initial conditions.
- Simulate this behaviour under different initial conditions.

Scientific Approach

- Newton's 2nd Law
- Newton's Law of Universal Gravitation
- Principle of Superposition
- Kinematics

Coding Approach

- Finite Difference Approximation

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

Testing

- Analyze conservation of energy and momentum.
- Analyze the effect of different masses, initial velocities and position.
- Analyze the effect of different time-step intervals.

Timeline

- Research
- Start Coding
- Debug
- Testing
- Begin Report
- Finalize and Submit

Feedback?

N-Body Problem and the Simulation of

Kevin Sun

October 24th 2013

Project Introduction

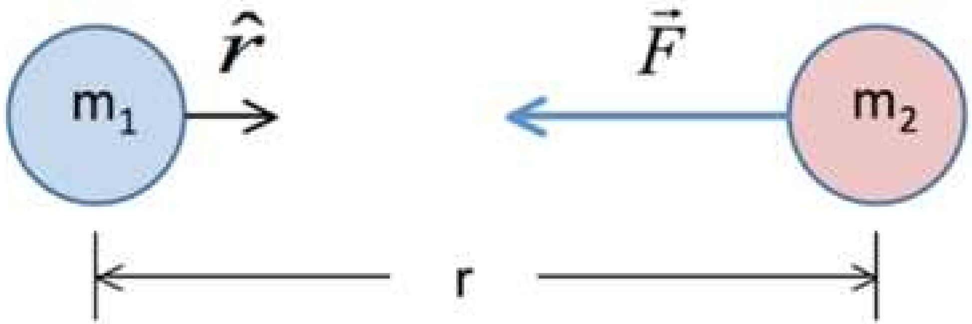
- Galaxies can be modeled to act as numerous N-bodies interacting through forces like gravity
- The collisions between two masses can be simulated with variable initial conditions (mass, velocity, position)
- Toomre Sequences depicts the events of two spiral galaxies (tidal tails and merger remnants)

Project Goals

- To write a MATLAB program that reproduces the Toomre Sequence using simplified physics, finite difference approximations and discretized particle grids
- Vary initial conditions to model scenarios with changes in mass, velocity, and position
- Attempt to add a changing point mass in a stationary location to simulate the effects of a black hole on galaxies

Mathematical Equations

- Newton's law of universal gravitation, where G is the gravitational constant: 6.67×10^{-11} N·(m/kg)^2



$$\vec{F} = -\frac{Gm_1m_2}{r^2}\hat{r}$$

Continued…

- Newton's Second Law:

$$F = ma = \frac{Gm_1m_2}{r^2}$$

- Arriving at the acceleration of a particle due to gravity:

$$a = \frac{Gm}{r^2}$$

Approach

- Galaxies will be treated as clusters of massless point particles that represent stars
- All of the mass of the galaxies will be at their respective centers, used to trace motion.
- Define initial conditions that can be changed (position, initial velocity, acceleration, mass)
- Will use FDA and a specified time step to trace the motion of the particles across a discretized particle grid

Tentative Timeline

- Proposal, research, derive equations, formulate code Oct 22- Oct 29
- Implement and test code Oct 24-Nov 7
- Numerical tests and experimentation, begin report and final presentation Nov 7 - Nov 14
- Analyze data and continue report and final presentation Nov 14 - Nov 21
- Refine presentation and begin final draft of report Nov 21 - 28
- Submit report by November 30 (final date is December 2nd)

References

- http://www.newscientist.com/article/dn13635#.UmSf4vmsiSo
- http://astronomy.swin.edu.au/cosmos/T/Toomre+Sequence

Optics and Ray Tracing

PHYS 210 Term Project Proposal

Ben Vause

October 19th 2013

Overview

Light interacts with different surfaces in a variety of ways:

- Light travelling through a prism will experience refraction and will disperse according to wavelength
- Light which strikes a mirror will reflect at an angle dependent on the angle with which it hits the mirror
- Light passing through a lens will be refracted at an angle dependent on where the ray passes through the lens

Project Goals

- To write a MATLAB code which traces the path of one or more rays of light as they are effected by various optical objects such as lenses, prisms, and mirrors
- To investigate the changes which occur as the initial conditions are varied such as the properties of the light ray(s)
- To produce results which would agree with an experiment of the same set-up

Mathematical Formulation

- When referring to mirrors:
  - The angle of incidence is equivalent to the angle of reflection
    - $\Theta i = \theta r$
- When referring to prisms and determining refraction angle:
  - Snell's Law
    - $n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$
- When referring to lenses and determining refraction angle and focal length:
  - Snell's Law (again)
    - $n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$
  - Thin lens approximation
    - $1/s + 1/s' = 1/f$

Numerical Approach

- Boundary conditions: Ray may be thought of as inside a box with mirrors on internal walls
- Properties of light ray:
  - Total velocity will be the speed of light
  - White light (visible light)
  - Position at any given time = x_n, y_n
  - Next position → $x_{n+1} = x_n + v_x \Delta t$, $y_{n+1} = y_n + v_Y \Delta t$

Testing & Numerical Experiments

- Test different wavelengths of light and see if they follow the path that calculations have proven
- Check that all points of interaction (where the ray and an object meet) are points of intersection on the 2D plane
- Rays diverge or converge at the correct focal point depending on the lens
- The angle of incidence and reflection are the same for all interactions with mirrors

Project Timeline

Date	Activity
10/15 - 10/26	Start and finish basic research, experiment with equations & begin code design
10/27 - 11/15	Implement code
11/16 - 11/19	Test code
11/20 - 11/26	Run numerical experiments, analyze data, begin report
11/27 - 11/29	Complete report
12/01	Submit finished term project

References

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

http://laplace.phas.ubc.ca/210/

http://hyperphysics.phy-astr.gsu.edu/hbase/ligcon.html

Simulation of the Motion of a Compound Pendulum Using Ordinary Differential Equations

PHYS 210 Term Project Proposal

Lincoln Wu

- Overview

- My initial idea for the project is to construct a simple model that simulate the motion of a folding polypeptide into a protein.
- A system of compound pendula is a good choice to exemplify the idea of a chain of polymer and computationally more appropriate for a programming novice like me.
- A compound pendulum consists of an arbitrary number of pendula joined end to end with one end of the entire chain fixed to a pivot point. The pendula themselves have no mass while the end of each pendulum has a particle permanently attached. The system is frictionless.

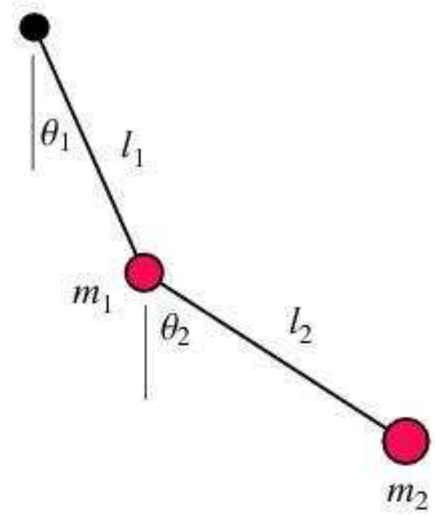
- Project Goals

- To write an MATLAB (octave) code which solves the equation of motion of the system of compound pendulum in a 2D plane under the influence of gravity, using an ordinary differential equation approach.
- To investigate if the simulation could include the function to allow the program user to change the initial conditions, including pendulum starting angle, pendulum length, mass of the particles, and the number of pendulums.
- To establish correctness of the implementation of the code through convergence tests and comparison with known solutions.

- Mathematical formulation (Equation of Motion)

- To keep the presentation simple, a double pendulum is used for demonstration.
- Double pendula are an example of a simple physical system which can exhibit chaotic behavior. Consider a double bob pendulum with masses m1 and m2 attached by rigid massless wires of lengths l1 and l2. Further, let the angles the two wires make with the vertical be denoted θ1 and θ2, as illustrated below. Finally, let gravity be given by *g*. Then, in a 2D cartesian coordinate of x-axis and y-axis, the positions of the red bobs are given by:

$$
\begin{aligned}
x_1 &= l_1 \sin\theta_1 \\
y_1 &= -l_1 \cos\theta_1 \\
x_2 &= l_1 \sin\theta_1 + l_2 \sin\theta_2 \\
y_2 &= -l_1 \cos\theta_1 - l_2 \cos\theta_2.
\end{aligned}
$$



The potential energy of the system is then given by:

$$\begin{aligned} V &= m_1 g y_1 + m_2 g y_2 \\ &= -(m_1 + m_2) g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2 \end{aligned}$$

The kinetic energy of the system is then given by:

$$\begin{aligned} T &= \tfrac{1}{2} m_1 v_1^2 + \tfrac{1}{2} m_2 v_2^2 \\ &= \tfrac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} m_2 [l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)] \end{aligned}$$

The Lagrangian is:

$$L \equiv T - V$$

$$= \tfrac{1}{2}(m_1 + m_2) l_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

If the Lagrangian of a system is known, then the equations of motion of the system may be obtained by a direct substitution of the expression for the Lagrangian into the Euler–Lagrange equation:

For θ1, the equation of motion is:

$$(m_1 + m_2)l_1\ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + g(m_1 + m_2)\sin\theta_1 = 0.$$

For θ2, the equation of motion is:

$$m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 g \sin\theta_2 = 0$$

By coupling the above two second-order ordinary differential equations, we can solve numerically for θ1(t) and θ2(t), for any particular choice of parameters and initial conditions.

By the power of induction, I can devise the system of equation of motion for an arbitrary number of pendula with varying initial conditions.

- Visualization and Plotting

With the equation of motion solved and $\theta n(t)$ handy, I can simulate the motion of the compound pendulum using tools like xvs to generate mpeg animations. MATLAB's plotting function could also be used for plots to be included in my report.

- Testing and Numerical Experiment

- Testing
  - Check the numerical results against the known solutions for simple compound pendulum systems. (ie. Double pendulum)
- Numerical Experiments
  - Investigate the motion of the pendulum in a 3-dimensional space.

- Project Timeline

Dates	Activities
10/15–10/26	Do basic research, derive equations & begin code design
10/27–11/15	Implement code
11/16–11/19	Test code
11/20–11/26	Run numerical experiments, analyze data, begin report
11/27–11/29	Finish report
11/29	Submit project! (absolute deadline is 12/02)

- References

- http://bh0.phas.ubc.ca/210/Doc/term-projects/kdv.pdf
- http://scienceworld.wolfram.com/physics/DoublePendulum.html
- http://en.wikipedia.org/wiki/Lagrangian

- Questions?
- Comments?
- Suggestions?

THE GRAVITATIONAL N-BODY SIMULATION

PHYS 210 Project Presentation

Zhixuan Xu

Overview

- The gravitational n-body simulation is a simulation of n interacting particles or masses which only under the influence of gravity.
- The n-body simulation is able to predict the positions and velocities of particles after a time.

Project Goals

- Create a MATLAB (or Octave) code to simulate the gravitational-only interaction between n particles.
- Visualize the motion of the n particles in 2D(or 3D if possible).
- Try to get various simulations depending on number of particles.

Mathematical Formula

- The Newton's Second Law:

$$m\bar{a} = \frac{d\bar{F}}{dt}$$

$$\bar{a} = \frac{d\bar{v}}{dt}$$

$$\bar{v} = \frac{d\bar{x}}{dt}$$

- $\bar{a}$ is the acceleration of the particle.
- $\bar{v}$ is the velocity of the particle.
- $\bar{x}$ is the displacement of the particle.

Mathematical Formula (Continued)

- The Newton's Second Law and the law of gravitation:

$$m_i \overline{a_i} = G \sum_{j=1, j \neq i}^{n} \frac{m_i m_j}{\overline{r_{ij}}^2} \widehat{r_{ij}}, i = 1,2 \ldots N$$

- $\overline{a_i} = a(t)$ is the acceleration of the ith particle.
- m_i, m_j are the mass of the ith and jth particle.
- $\overline{r_{ij}}$ is the vector from the ith particle to the jth particle.
- $\widehat{r_{ij}}$ is the unit vector along $\overline{r_{ij}}$.

Project Timeline

Dates	Activities
10/15-10/26	Research, derive equations
10/27-11/15	Design Code and implement code
11/16-11/19	Test code
11/20-11/26	Run numerical experiments, analyze data, begin report
11/27-11/29	Finish report
11/29	Submit