Simulation of the Motion of *N* Gravitationally Interacting Particles Using Euler's Method

Physics 210 Term Project Proposal

Masoud Rafiei

October 20, 2009

The University of British Columbia

Overview

- The n-body problem is solved as the Initial Value Problem using Euler's approximation

Project Goals

- To approach the n-body problem numerically; writing a MATLAB code in 3D space which satisfies the initial condition and predicts the future
- To simulate particles' motion in two dimensional Cartesian coordinate system
- To investigate the system's behavior when various symmetrical initial conditions are applied
- To verify the accuracy of the method, using The Law of Conservation of Energy.

Mathematical Formulation

- In classical mechanics, Newton's Law of Universal Gravitation can be written as

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

Which derives the net force acting on the particle i

In addition,

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

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

Mathematical Formulation (continued)

By expanding coordinates of each vector, we obtain

$$\begin{bmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} \end{bmatrix} = G \cdot \sum_{i \neq j} m_j \frac{\left(\begin{bmatrix} r_{x_j} \\ r_{y_j} \\ r_{z_j} \end{bmatrix} - \begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} \right)}{((r_{x_j} - r_{x_i})^2 + (r_{y_j} - r_{y_i})^2 + (r_{z_j} - r_{z_i})^2)^{3/2}} \qquad i = 1,2,3,\ldots,n$$

$$\begin{bmatrix} v_{x_i} \\ v_{y_i} \\ v_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix} + \begin{bmatrix} v_{\circ x_i} \\ v_{\circ y_i} \\ v_{\circ z_i} \end{bmatrix}$$

$$\begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix} + \begin{bmatrix} r_{\circ x_i} \\ r_{\circ y_i} \\ r_{\circ z_i} \end{bmatrix}$$

Numerical Approach

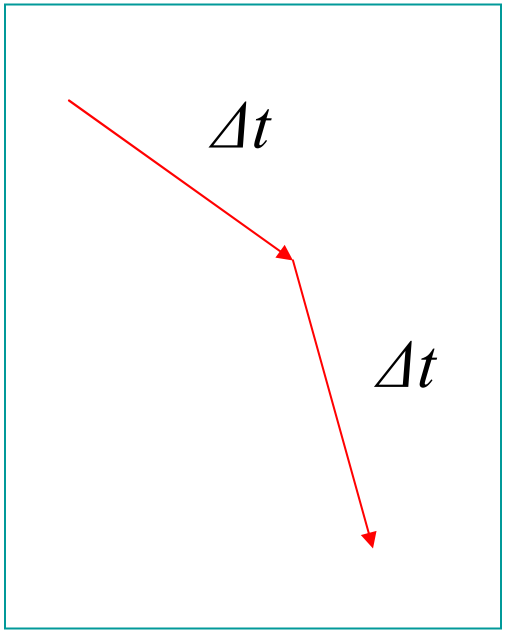
- EQ.2 & EQ.3 represent Euler's method of approximation in which the acceleration (in EQ.2) and the velocity (in EQ.3) are assumed to be constant during Δt (which can be chosen small enough to satisfy the desired accuracy)

$$\begin{bmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} \end{bmatrix} = G \cdot \sum_{i \neq j} m_j \frac{\left(\begin{bmatrix} r_{x_j} \\ r_{y_j} \\ r_{z_j} \end{bmatrix} - \begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} \right)}{((r_{x_j} - r_{x_i})^2 + (r_{y_j} - r_{y_i})^2 + (r_{z_j} - r_{z_i})^2)^{3/2}} \qquad \text{EQ.1}$$

$i = 1,2,3,\dots,n$

$$\begin{bmatrix} v_{x_i} \\ v_{y_i} \\ v_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix} + \begin{bmatrix} v_{\circ x_i} \\ v_{\circ y_i} \\ v_{\circ z_i} \end{bmatrix} \qquad \text{EQ.2}$$

$$\begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix} + \begin{bmatrix} r_{\circ x_i} \\ r_{\circ y_i} \\ r_{\circ z_i} \end{bmatrix} \qquad \text{EQ.3}$$



Numerical Approach (continued)

- Given initial velocity and initial position (which can be generated randomly) of all particles as the Initial Conditions, the motion can be predicted by desired number of successive loops

Variables / Equations	First Input	Second Input	Final Output
EQ.1	$\vec{r}_{\circ i}$	$\vec{r}_{\circ j}$	$\vec{a}_i$
EQ.2	$\vec{v}_{\circ i}$	$\vec{a}_i$	$\vec{v}_i$
EQ.2	$\vec{r}_{\circ i}$	$\vec{v}_i$	$\vec{r}_i$

$$\begin{bmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} \end{bmatrix} = G \cdot \sum_{i \neq j} m_j \frac{\left(\begin{bmatrix} r_{x_j} \\ r_{y_j} \\ r_{z_j} \end{bmatrix} - \begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} \right)}{((r_{x_j} - r_{x_i})^2 + (r_{y_j} - r_{y_i})^2 + (r_{z_j} - r_{z_i})^2)^{3/2}} \qquad \text{EQ.1}$$

$i = 1,2,3,\ldots,n$

$$\begin{bmatrix} v_{x_i} \\ v_{y_i} \\ v_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix} + \begin{bmatrix} v_{\circ x_i} \\ v_{\circ y_i} \\ v_{\circ z_i} \end{bmatrix} \qquad \text{EQ.2}$$

$$\begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} = \Delta t \cdot \begin{bmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix} + \begin{bmatrix} r_{\circ x_i} \\ r_{\circ y_i} \\ r_{\circ z_i} \end{bmatrix} \qquad \text{EQ.3}$$

Test & Numerical Experiments

Testing

- Accuracy Testing: Calculate the total amount of energy (Kinetic + Gravitational Potential) in each successive loop for each particle. Then, plot the values as a function of time.

Numerical Experiments

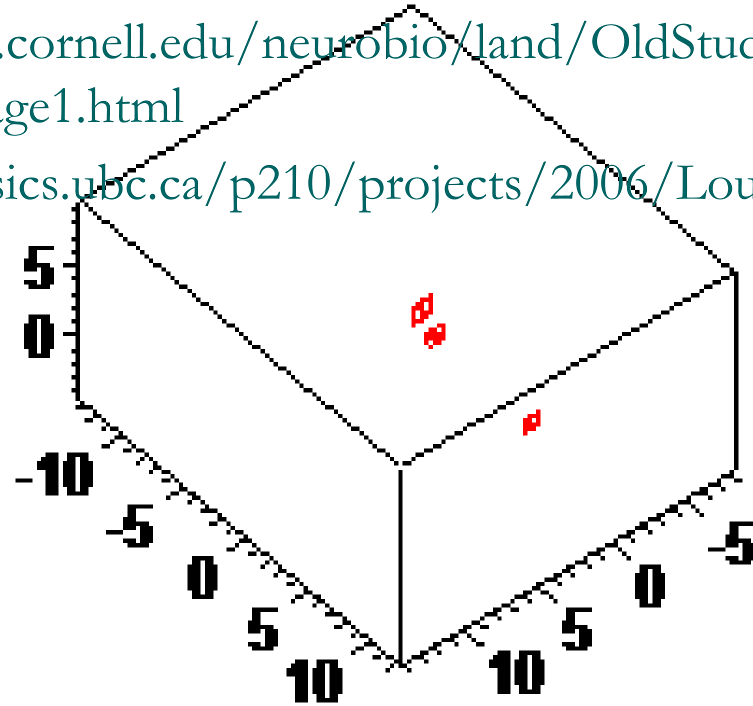
- Simulate the motion of a large number of particles in two dimensional coordinate system (i.e. restricted to a flat surface)
- Apply various (geometrical) symmetrical initial conditions and investigate the behavior of the system – Does the system retain its symmetry?

Project Timeline

Dates	Activities
10/23-10/29	Research on mathematical formulation & design code
10/30-11/05	Design code & implement code
11/6-11/12	Test code
11/13-11/19	Apply numerical experiments, begin presentation & report
11/20-11/26	Analyze Data, continue work on presentation & report
11/27-11/30	Polish Presentation & work on final draft of report
12/01	Present your project!
12/01-12/04	Finish & submit the report

References

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




APPLAUSE !!

Toomre Model for Galaxy Collisions

Phys 210 – Term Project Proposal

Eric Walker

October 20^{th}, 09

- Overview
  - Simplified of the toomre equations where the stars have a mass << galaxy core.
  - Using more basic formulas to calculate the motion of the individuals stars around the two interacting cores
  - Eventually settle into equilibrium

- Project Goals
  - Use MATLAB(octave) to work through the equations and create an accurate system
  - Have functional code to resolve the forces over time on each of the N suns
  - Use parameters (initial velocities, orientation of galaxy planes, and impact parameters) to see how the collisions can vary.

Complex - Equations of Motion

- Poisson Equation – gravitational potential

$$\nabla^2 \Phi(x,t) = 4\pi G \rho(x,t)$$

- Continuity equation – statement of mass conservation

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_g v) = 0.$$

- Euler Equation – time variation of velocity

$$\frac{\partial v}{\partial t} + (v \cdot \nabla) v = -\frac{1}{\rho_g} \nabla \mathcal{P} - \nabla \Phi.$$

- Dissipation of energy in a collisional fluid

$$\rho_g \frac{\partial u}{\partial t} + \rho_g (v \cdot \nabla) u + \mathcal{P} (\nabla \cdot v) = -\mathcal{L}$$

- Ideal gas and isothermic equations of state

$$\begin{aligned} \mathcal{P} &= (\gamma - 1) \rho_g u \\ \mathcal{P} &= \rho_g v_s^2 \end{aligned}$$

- Collision-less Boltzmann Equation

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f - \nabla \Phi \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0.$$

- Newton's Second Laws as a two first order differential equations

$$\begin{aligned} \frac{d\boldsymbol{x}_i}{dt} &= \boldsymbol{v}_i \\ \frac{d\boldsymbol{v}_i}{dt} &= -\nabla \Phi \end{aligned}$$

- # Simplified Version!

- Each particle has an initial period, distance, orbit(circular) defined by:

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

- Since mass of each star is significantly less then mass of galaxy m = 0.

- Calculate gravity through newtons law:

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

- Numerical Approach
  - Barnes-Hut algorithm will be attempted... If success I will be able to create a much larger N (possibly thousands)
  - If not then a finite difference approach will be used (number N is an exponential value of difficulty rather than linear)

- Visualization
  - Build in software with MATLAB to create mpeg file

- Numerical Experiment
  - Simply calculate the force acting on each body and let the physics take over.
  - This is rather taxing for the CPU so N will be limited to a few hundred stars in each galaxy
  - Collision-less approach to keep it reasonable
  - Experiments initial parameters can create numerous experiments once the formulas are functional

- Project Timeline

Dates	Goal
Nov . 1st	Basic Research, Equations, Start coding
Nov. 7th	Implement code
Nov. 14th	Code Testing
Nov. 21st	Have an operational experiment and start report/presentation
Nov. 28th	Data analysis/presentationcompletion
Dec. 1st	Present Data/Have a draft completed
Dec. 4th	Submit final Paper

- References

  – A Gentle Introduction to Numerical Simulations of Galaxy Mergers, Christopher Q. Trinh(2006)
  – Collision-induced Galaxy Formation, THE ASTROPHYSICAL JOURNAL, 497:541554, 1998

Questions??????


APPLAUSE !!

Finite Difference Solution of the Reaction-Diffusion Equation in One Dimension

PHYS 210 Term Project Proposal

Michael Hall

October 20, 2009

- Overview
  - The reaction diffusion equation is a semi linear heat equation in one space variable and time which admits interesting “travelling wave” solutions
  - The reaction-diffusion equation solutions propagate at a constant speed dependant on the rate of diffusion / annihilation of the reactants

- Project Goals
  - To write an MATLAB (octave) or an Java code / applet which solves the reaction diffusion equation numerically, using finite difference techniques
  - To establish correctness of the implementation of the code through appropriate tests and comparison with known solutions
  - To investigate a variety of parameters for the equation, including those describing travelling wave solutions

- Mathematical Formulation (Equations of Motion)
  - The reaction diffusion equation can be written in the form

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + R(u)$$

    where $u \equiv u(x,t)$

  - The equation will be solved with initial conditions which will ensure an eye-catching, mind-boggling display.

- Numerical Approach
    - Finite difference techniques. Enough said.

- Testing and Numerical Experiments
  - Use a variety of parameters. Analyze results. Repeat ad nauseam.

- Project Timeline

Dates	Activities
10/23 – 10/29	Do basic research, derive equations and design code
10/30 – 11/05	Implement code
11/06—11/12	Test code
11/13—11/19	Run numerical experiments, begin presentation & report
11/20—11/26	Analyze data, continue work on presentation & report
11/27—11/30	Polish presentation and work on final draft of report
12/01	Give presentation!
12/01—12/03	Finish final draft of report
12/04	Submit report!

- References

  - Zhou et al, Traveling Wave Speed And Solution In Reaction-Diffusion Equation In One Dimension, Applied Mathematics and Mechanics, Vol 22, No 6, Jun 2001.
  - http://www.youtube.com/
  - http://www.wikipedia.org/
  - http://laplace.physics.ubc.ca/210/Doc/term-projects/kdv.pdf


APPLAUSE !!

Analysis of Simulated Protein Folding Data Using Free Energy Potentials

PHYS 210 Term Project Proposal

Gene Polovy

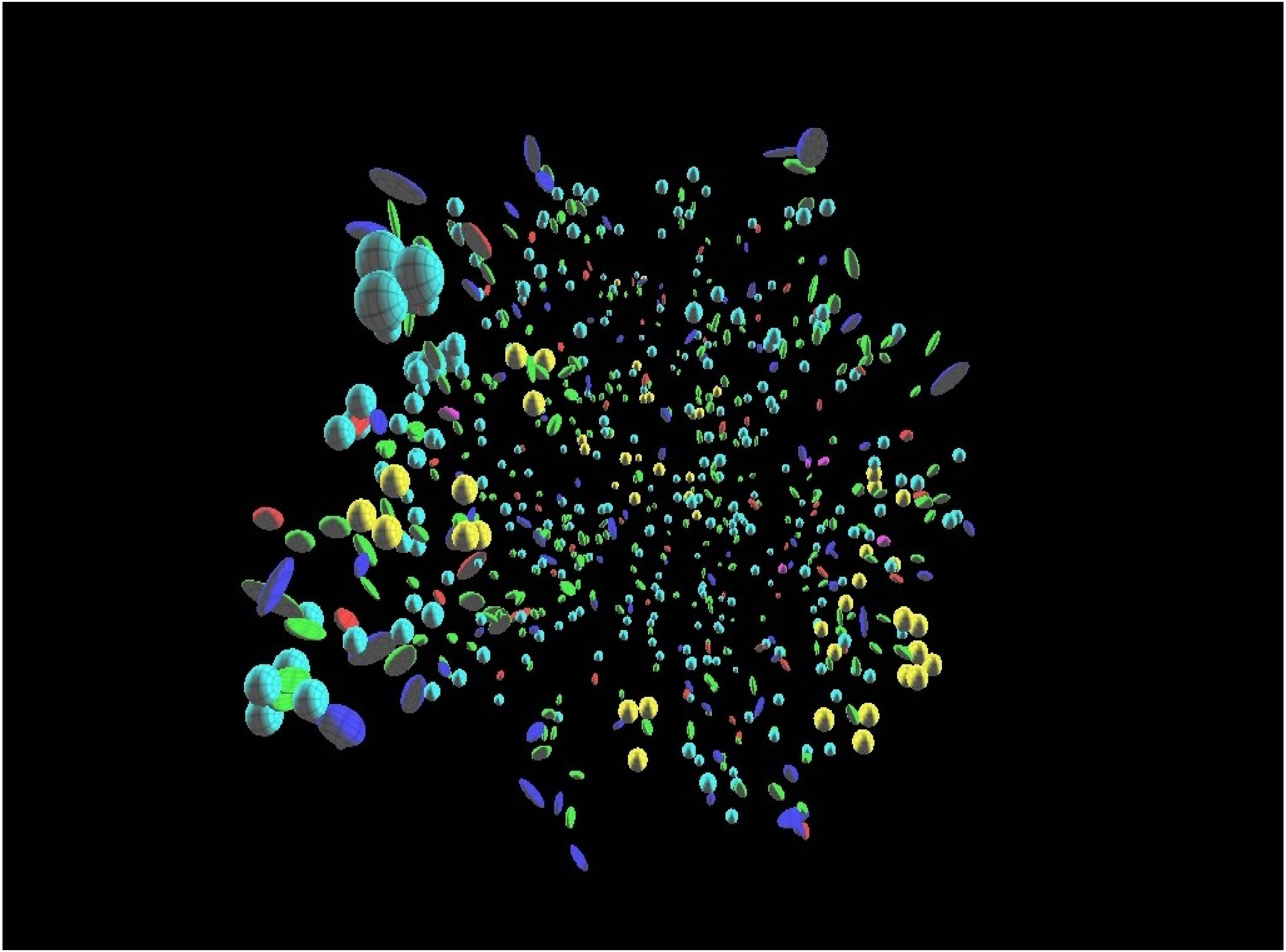
October 20, 2009

Introduction

- Proteins are polymers of animo acids joined together by peptide bonds.
- To minimize their free energy, $F = U - TS$, where U is the internal energy, T is the temperature and S is the entropy, proteins fold into complex three dimensional structures, which ultimately determine their chemical properties. The shape of these (tertiary) structures depends on the actual animo acid sequence (the primary structure) and the external conditions, such as temperature.
- Folding is entropically unfavourable (for the protein itself) and energetically favourable.

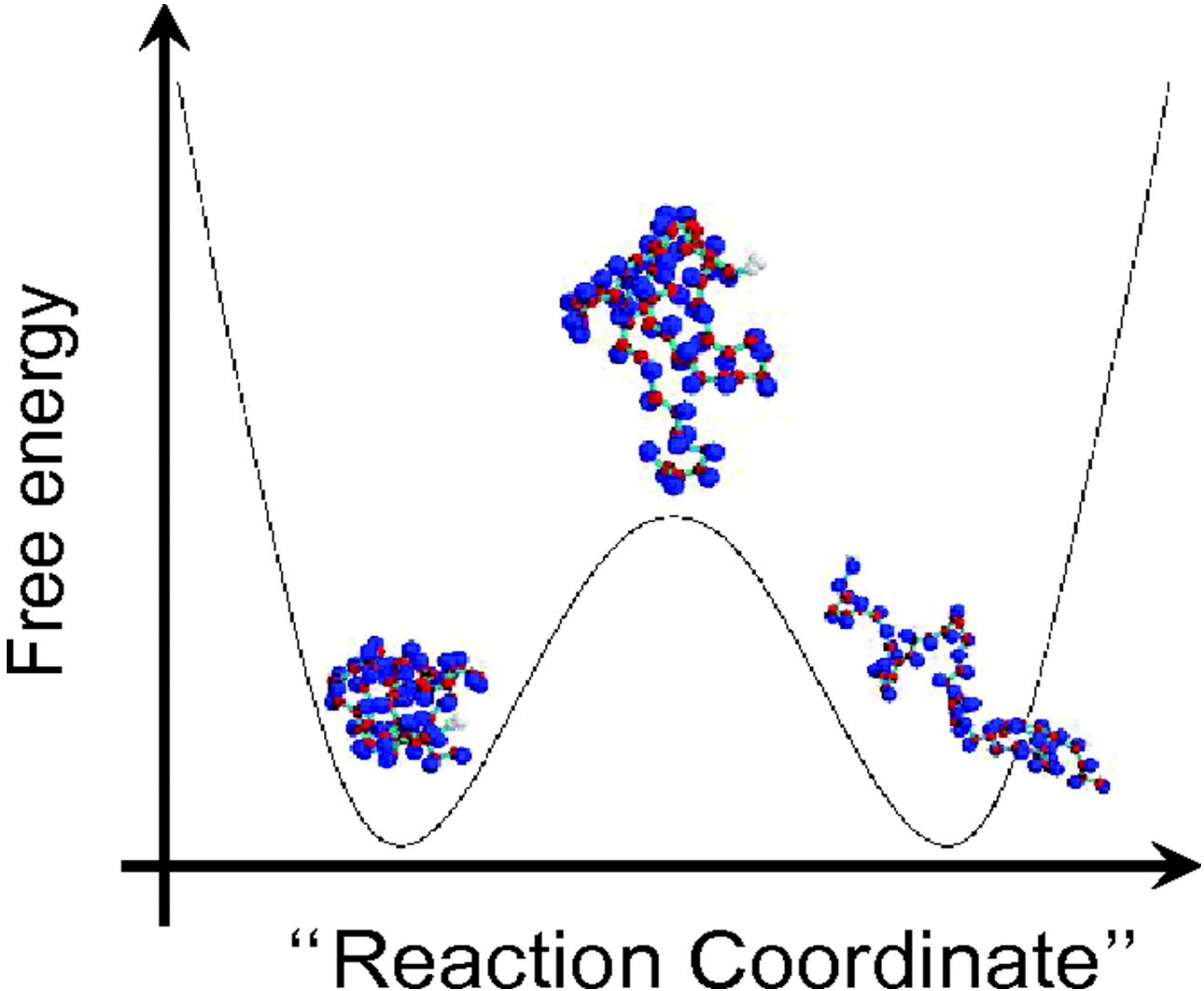
Folding Simulations

- Various numerical techniques for simulating the dynamics of protein folding exist; course grained models are often used to minimize simulation time. In such models, the residues – linked amino acids – are replaced by objects that approximate the combined effect of all the atoms in them.



Analysis of Simulated Data

• From simulated positional and energetic data, one can find the mean bond distances between residues at different times during the folding process and and create free energy surfaces/contours, which can then be used to determine the structure of the transition state(s).
• I will need to write MATLAB programs to read very large data files obtained via simulations, calculate the necessary quantities and plot the results. This will be followed by a detailed analysis of these results.


Free energy
"Reaction Coordinate"

Timeline

Dates	Activities
10/23–10/29	Do basic research & design code
10/30–11/05	Implement code
11/06–11/12	Test code
11/13–11/19	Run numerical experiments, begin presentation & report
11/20–11/26	Analyze data, continue work on presentation & report
11/27–11/30	Polish presentation and work on final draft of report
12/01	Give presentation!
12/01–12/03	Finish final draft of report
12/04	Submit report!


APPLAUSE !!

Motion of Unstable Charged Particles

Natassia Orr

Phys 210

Overview

- Many atoms and elementary particles are charged and/or experience decay.
- As a particle moves, momentum carries it in the direction of travel, while eternal forces may cause acceleration in any direction.

Project Goals

- Write a script to solve for the motion, x(t), of a system of particles when given the initial parameters.
- Display the results visually in an easily understood fashion.

Mathematical Formulae

$$N = N_0 e^{-t/\lambda}$$

$$F = \frac{q_1 q_2}{4\Pi \varepsilon_0 r^2} \hat{r}$$

$$p = mv = Fdt$$

$$F = qv \times B$$

$$F = ma$$

The formulae on the right are Newtonian equations of motion and decay.
The formulae on the left are the electric and magnetic force exerted on charged particles.

Numerical Approach

- Since a small number of particles must decay discretely instead of continuously, N will be rounded to the nearest integer.
- The temporal domain will be taken from $t_0=0$ to the time when the last particle decays.
- Velocities will be restricted to absolute values of less than 0.1c.

Testing

- Parameters of known particles will be input to the code and output will be compared to recorded experimental data.
- Testing will be done in three parts: decay, electro-magnetic attraction/repulsion, and motion.

Numerical Experiments

- 5-10 Particles will be defined in the system and their interactions observed.

Timeline

Oct 20-25 Research unstable particle experiments
Oct 26-Nov1   Write and test code for exponential decay
Nov2-8   Write and test code for electromagnetic force,
    begin report
Nov9-15   Write and test code for the system in motion,
    work on report
Nov16-22   Complete testing of code, create
    visualization
Nov23-29   Finish presentation and report
Nov30-Dec3   Edit final draft of report, give presentation
Dec4   Report due


APPLAUSE !!

N-body Simulation in Dynamic Space

Yuuki Omori

Oct/20/2009

Overview

What is a N-Body simulation?

- Use of Computers

- Visualization

Project Goal

-To write a Matlab code which numerically calculates the motions of gravitationally interacting objects within a dynamic space.

-To test various initial conditions and parameters.

-To produce a visualization of the calculations.

Physics of Stationary Space N-Body Simulations

- Gravitational Interaction between 2 objects

$$F = -G\frac{mM}{\left|\mathbf{r}-\mathbf{R}\right|^2}\frac{\mathbf{r}-\mathbf{R}}{\left|\mathbf{r}-\mathbf{R}\right|}$$

Physics of Stationary Space N-Body Simulations

- Gravitational Interaction between 2 objects
-Force is a vector
-Potential is a scalar

$$F = -\nabla U$$

$$U = G\frac{mM}{|\mathbf{r} - \mathbf{R}|}$$

Physics of Stationary Space N-Body Simulations

For N particles:

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

Can get force by using:

$$F = -\nabla U$$

Physics of Stationary Space N-Body Simulations

Force causes particles to accelerate.

The i-th particle will move according to:

$$r_{(n+1),i} = r_{(n),i} + \dot{r}_{(n),i}dt + \frac{1}{2}\ddot{r}^2_{(n),i}(dt)^2$$

Where n is the step number and dt is the step time.

Dynamic space

In a non-static space, we also have the consider the rate in which the space is changing. Consider a scale factor A(t) which relates static to non-static space.We can then define:

$$\rho = A(t) \mid r \mid$$

$$\beta = \dot{\rho} = \dot{A}(t) \mid r \mid = \frac{\dot{A}(t)}{A(t)}\rho$$

$$\gamma = \ddot{\rho} = \frac{\ddot{A}(t)}{A(t)}\rho$$

Dynamic space

Using this, the motion of the i-th particle will be

$$r_i(t) = \rho + \beta t + \frac{1}{2}\gamma t^2$$

Numerical Approach

Starting off with initial conditions, I will use Symplectic Euler's method. In this scheme,

$$\ddot{r}_{(n),i} = \frac{F_{(n),i}}{m_i}$$

$$\dot{r}_{(n+1),i} = \dot{r}_{(n),i} + dt\ddot{r}_{(n),i}$$
$$r_{(n+1),i} = r_{(n),i} + dt\dot{r}_{(n+1),i}$$

Numerical Approach

plug these into

$$\rho = A(t) \mid r \mid$$

$$\beta = \dot{\rho} = \dot{A}(t) \mid r \mid = \frac{\dot{A}(t)}{A(t)} \rho$$

$$\gamma = \ddot{\rho} = \frac{\ddot{A}(t)}{A(t)} \rho$$

$$r_i(t) = \rho + \beta t + \frac{1}{2} \gamma t^2$$

Testing And Numerical Experiments

Testing - Set $N = 1$, restrict to 1D and see if it recovers classical kinematics.

Different functions for A(t)
-Stationary case
-Constant expansion
-Cycloid expansion

-Various initial conditions

Project Time Line

Date	Goal
10/23	Basic research, code design
10/30 $\mapsto$ 11/05	Implement code
11/06 $\mapsto$ 11/12	Test code and experiment
11/13 $\mapsto$ 11/19	Begin presentation work and report
11/20 $\mapsto$ 11/26	Analyze data, work on presentation
11/27 $\mapsto$ 11/30	Work on final draft of report
12/01	Give presentation
12/01 $\mapsto$ 12/03	Finish final draft of report
12/04	Submit report


APPLAUSE !!

Formation of The Saturnian Rings through N-body simulation

PHYS 210 Term Project Proposal

Kasun Somaratne

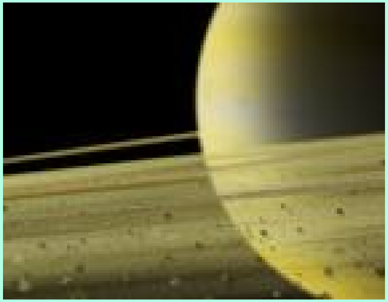
October 20, 2009

Overview & Motivation

- The motion of neutral particles in a N-body system is goverened primarily by the gravitational interaction between the particles. These particles obey the Newton's laws of motion.
- Collisions between particles can be either elastic, where both momentum and kinetic energy are conserved or inelastic, where only the momentum is conserved.
- Such a N-body system can be used to simulate the evolution of a planetary system where gravitational forces dominate the interaction between particles.
- Gravitational interactions and collisions between particles can be used to study the formation of planetary disks (asteroid belt, rings of Saturn etc.)

Project Goals

- To write a Matlab programme to govern the evolution of N particles by discrete time stepping.
- To investigate the correctness of the code and its limitations through various test cases (simple particle collisions, 2-body system, comparison with other N-body simulations etc.)
- To use the programme to simulate various planetary systems with different initial conditions.



chandra.harvard.edu

Mathematical Formulations

Gravitational interaction

- The evolution of n-particle system under pure gravitational interaction can be written as

$$m_j\vec{a_j} = \sum_{k\neq j}^{n} \frac{m_j m_k(\vec{r_k} - \vec{r_j})}{|\vec{r_k} - \vec{r_j}|^3}, j = 1..n \tag{1}$$

- Eq (1) will be solved for n particles with given intial position and velocity vectors in 3 dimensions.

Particle collisions

- Perfectly inelastic collisions - only momentum is conserved, particles stick together.

$$\vec{v_f} = \frac{m_1\vec{v_{1,i}} + m_2\vec{v_{2,i}}}{m_1 + m_2} \tag{2}$$

Mathematical Formulations (continued)

- Totally elastic collisions - Both kinetic energy and momentum conserved.

$$\angle xyz_n = \cos^{-1}\left(\frac{\Delta x_n v_{nx} + \Delta y_n v_{ny} + \Delta z_n v_{nz}}{\sqrt{(\Delta x_n + \Delta y_n + \Delta z_n) * (v_{nx} + v_{ny} + v_{nz})}}\right) \tag{3}$$

- Eq (3) is used to calculate the 3D angle between the particles. Then it can be treated as a 1D collision by rotating the frame of reference.

Roche limit

- Roche limit is the orbital radius **d** at which the tidal forces on a satellite by the body it orbits is stronger than the gravitational force holding the satellite together.

$$d = R\left(2\frac{\rho M}{\rho m}\right)^{1/3}$$

Numerical Approach

- Each particle in the n-body system will be spherical with an initial mass **m**, radius **r**, position $[s_x s_y s_z]$ and a velocity $[v_x v_y v_z]$

- Eq (1) can be simplified first by calculating the center of mass $\vec{R}$ of the system at a particular time **T**.

$$\vec{R} = \frac{\sum m_i \vec{r_i}}{\sum m_i} \tag{4}$$

- The evolution of the system is carried out by discrete time stepping. The time interval Δ t will be chosen by considering the shortest distance between any two particles and their velocities at a given time.

- Depending on the initial velocities and masses, the collision will be either totally elastic or totally inelastic.

Visualization and Plotting Tools

- I hope to use **xvs** to visualize the evolution of the n-body system and to generate mpeg animations.
- I will use **sm** or **gnuplot** to generate plots to be included in my report.

Testing

- Test the interaction of a simple 2-body system with various initial conditions and collision scenarios.
- Check if the implementation correctly predicts the Roche limit for a clump of particles orbiting a relatively massive particle.
- Compare the results from my programme with other n-body simulation softwares. (i.e. Planets etc.)

Numerical Experiments

- Attempt to simulate the formation of the rings of Saturn from the disruption of a moon.
- investigate the sweeping effect of a massive planet orbiting within a ring of planetoids around a star.
- Attempt to correctly predict the positions of the 4 Galilean moons of Jupiter after a given amount of time.

www.nature.com

Project Timeline

Dates	Activities
10/22-10/29	Research & design code
10/30-11/05	Implement code
11/06-11/12	Test code
11/13-11/30	Run numerical experiments, begin presentation & report
12/01	Final Presentation!
12/01-12/03	Finish final draft of Report
12/04	Submit report!

References

- http://www.atmos.uiuc.edu/courses/atmos100/userdocs/3Dcollisions.html
- http://en.wikipedia.org/wiki/N-body_ problem
- http://scienceworld.wolfram.com/physics/RocheLimit.html
- http://laplace.physics.ubc.ca/210/Doc/term-projects/kdv.pdf

QUESTIONS?

COMMENTS?

SUGGESTIONS?


APPLAUSE !!

The Gutenberg-Richter Law for Earthquakes

Phys 210 Project Proposal

Gian Matharu – 20th Oct 2009

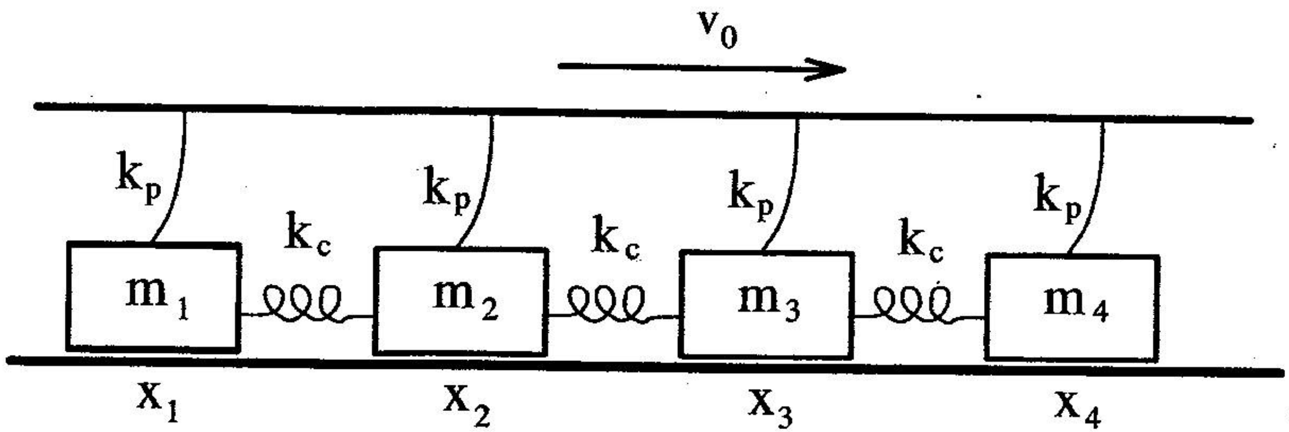
Overview

- The Gutenberg Richter power law represents a probability distribution for earthquakes occurring at particular magnitudes (only applicable to small M).
- A simplified model will be used to try and determine the properties a system must have to exhibit a power law distribution of earthquake sizes.

Project goals

- To write a MATLAB code that will solve the equations of motion for the model to simulate an earthquake focusing on frictional forces.
- Verify the results of the model by plotting a number of different graphs and then comparing with expected behaviour.
- Use different starting arrangements to progressively improve the realism of the model simulation.

Simplified Model



- Model represents two plates slowly moving relative to each other.
- Crust is represented as a single row of blocks connected by springs which exert forces on each other as described by Hooke’s Law.
- Bottom plate exerts a frictional force on the blocks preventing them from moving until the force from the springs overcomes the frictional force.
- This event is interpreted as an earthquake.
- The frictional force eventually stops the movement of the blocks and from there the process repeats.

Numerical Formulas (Newton's 2nd Law)

- By combining all the forces resultant from the springs and friction Newton's 2nd law gives:

$$m_i \frac{d^2 x_i}{dt^2} = k_c(x_{i+1} + x_{1-i} - 2x_i) + k_p(v_0 t - x_i) + F_f$$

- Which can be written as two separate differential equations

$$\frac{dx_i}{dt} = v_i$$

$$m_i \frac{dv_i}{dt} = k_c(x_{i+1} + x_{1-i} - 2x_i) + k_p(v_0 t - x_i) + F_f$$

Approach – Velocity and Displacement

- I will use parameter values that are known to provide fairly typical behaviour (k_c, m, k_p, F_0, v_0).
- For first simulation I will assume that all the blocks are in their equilibrium positions and stationary.
- By representing time as a series of increments Δt I will look to calculate the velocity of N blocks at time t and estimate the position of the block at the next step.
- Will also calculate the force on all of the blocks at each step.
- Repeat the simulation but adjust the initial configuration of the blocks such that their displacements lie randomly within ±0.001

Plots

- Will use MATLAB to make plots of velocity and displacement against time to examine behaviour of model.

Gutenberg-Richter Law

- Law states that

$$P(\mathcal{M}) = AM^{-b} = Ae^{-b\mathcal{M}}$$

- Can find M using the following summation

$$M = \sum_{n=time} \Big(\sum_{i=blocks} v_i t \Big)$$

- I will then find the magnitude of the earthquake using

$$\mathcal{M} = \ln{(M)}$$

Verification

- To verify the model I will look to produce a probability of a certain magnitude against magnitude that shows some form of a linear realtionship

Project Timeline

Date	Activities
10/23-10/29	Consider equations for velocity and displacement and how to implement them into code.
10/30-11/05	Implement code
11/06-11/12	Test code and implement Gutenberg Richter Law equations into code.
11/13-11/19	Compile graphical results, Begin presentation and report.
11/20-11/26	Test other initial conditions, continue report and presentation
11/27-11/30	Polish Presentation , work on final draft of report
12/01	Give presentation
12/01-12/03	Finish final draft of report
12/04	Submit report

References

- Giordano – *Earthquakes and Self organized criticality*


APPLAUSE !!

Ray tracing through various optical components

PHYS 210 Term Project

Shawn Wu

October 19, 2009

Overview & Project Goals

- Light hits different surfaces all the time. And what is interesting is how the light rays will travel after it hits a surface.
- If a light hits a prisms, dispersion will occur, therefore we'll see different colours if we shine a white light through a prism.
- The goal here is to write an MATLAB (octave) code which will produce a graph and show how the ray will travel after it hits an optical component.
- Optical component can include lenses, mirrors, and prisms.

Mathematical Formulation

- I am going to use the simple lenses and mirror equation:
  - $1/p + 1/q = 1/f$
    - Where *p* and *q* are the object and image distance from the lens and *f* is the focal length of the lens.
  - The equality of angle of incident/reflection for mirrors
  - I will also use Snell's law for prisms
    - $n_1 \sin(\theta_1)=n_2 \sin(\theta_2)$

Numerical Approach

- Using the thin-lens approximation, we can assume that a lens has a negligible thickness compared to its focal length.
  - $1/p + 1/q = 1/f$
- Because the index of refraction (n) is different at each value of lamda, each colour will be refracted by a different amount, so we see the full "rainbow" of colours emerge from a prism if we shine white light on it.
  - $n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$

Testing & Numerical Experiments

- Mirrors: Test to see if the angle of reflection and angle of indication equals
- Lenses: Given the distance of object and image to a lens, test to see if the focal length of graph agrees with the calculated value.
- Prisms: Given a particular index of refraction, compute to see that the colour it refracted will correspond to the wavelength.

Project Timeline

Dates	Activities
Oct 23 – Oct 30	Do basic research, & design codes
Oct 30-Nov 6	Implement code
Nov 6- Nov 13	Test code
Nov 13- Nov20	Run numerical experiments, begin report
Nov20- Nov 27	Continue on report
Nov 27 – Nov 30	Wrap up report and presetatoin
Dec 03	Presentation, and submit report.

References

http://en.wikipedia.org/wiki/Thin_lens
http://www.lhup.edu/~dsimanek/scenario/raytrace.htm

QUESTIONS?
SUGGESTIONS?

THE END


APPLAUSE !!

Galaxy Interaction
---Toomres' Model

PHYS 210 Term Project Proposal

Liang Jiajie (Javen)

Oct 21, 2009

Overview

- The Toomres' Model is a model showing the interaction between two galaxies when they come close to each other.
- The simulation was done by Alar Toomre and his brother Juri Toomre in 1970s. They only used Newton's simple gravitational laws, but the results they got nicely resemble some actual interacting galaxies in space.
- In the Toomres' Model, only the gravitational effect of the heavy objects (the galaxy cores) are considered, and the light objects (the stars) only act as “tracers” to show the interacting effects.

Project Goals

- To write an octave code to rebuild the Toomres' Model and simulate the galaxy interactions with technique mentioned in previous points.
- To investigate different initial conditions of the galaxy , such as mass, initial velocity, etc.
- To try to produce some similar effects comparing to actual galaxy interaction.

Mathematical Formulation

- The equation required is actually just Newton's simple gravitational law:

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

- Plug in the equation F=ma and simplify, the gravitational equation becomes:

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

Numerical Approach

- Because this model is a 3-dimension model, I will use a vector approach:

final acceleration: $a = (a_x, a_y, a_z)$

initial acceleration: $a_0 = (a_{x0}, a_{y0}, a_{z0})$

position of two interacting object:

$d_1 = (d_{x1}, d_{y1}, d_{z1})$ $d_2 = (d_{x2}, d_{y2}, d_{z2})$

distance between two object:

$$r = \sqrt{(d_{x2} - d_{x1})^2 + (d_{y2} - d_{y1})^2 + (d_{z2} - d_{z1})^2}$$

unit vector of distance:

$$d_u = \frac{d_2 - d_1}{r}$$

Numerical Approach (cont.)

- And then plug every thing in this formula:

$$a = a_0 \square \frac{d_{unit} \times G \times m_2}{r^2}$$

- This formula will give me the resulted acceleration of object 1 due to the the gravitational attraction of object 2.
- I will assume the mass of small objects are so small that the gravitational effects from them are negligible. In other words, I will only consider the gravitational effects, on both heavy and light objects, from the heavy objects.
- Given that the vector of initial velocity is v_0, the vector of final velocity v will be:

$$v = v_0 \square a \times t$$

- At this point, all I need to get the final velocity are d_1, d_2, m_2, a_0, v_0 and t, while t is changing, and a_0 at the very beginning is (0, 0, 0), then it changes with time. All other four parameters should be either set as initial condition or obtained after

Numerical Approach (cont.)

the galaxies construction.

- The method of galaxy construction is still a work in progress, and it is one of my main tasks this week.

Visualization

- My current plan is to use octave to do both the calculation and the visualization. However, I may use better visualizing software if possible.

Testing & Numerical Experiments

- Compare the final simulation with some other existing simulation from Internet, and also take advantage of the GalaxyColliding program in astronomy lab.
- Try to identify some basic characteristics which resemble the existing interacting galaxies. For example, the tidal tails of Antennae galaxy.
- Investigate the interaction between the two galaxy cores with different initial condition, and take a close look at the data obtained (if any.)
- Pick several small objects and investigate their motions with different initial condition of the galaxy cores, and observe the data (if any.)
- Try to have various numbers of small objects in the same initial condition and see if the effects are consistent. (I will have 50 small objects for each galaxy core during development, and try to have maybe 300 to 500 for each in the final product.)

Project Timeline

Dates	Activities
10/19~10/25	Improve the project planning with suggestions, do research, derive equations & begin designing code
10/26~11/1	Finish designing and begin Implement code
11/2~11/8	Finishing implementing code & basic testing
11/9~11/15	Run Numerical experiments & begin presentation
11/16~11/22	Basic data analysis, continue work on presentation & begin report
11/23~11/30	Finish presentation, practice & continue work on report
12/01~12/03	Give presentation & finish report
12/04	Submit report

References

- http://www.cv.nrao.edu/~jhibbard/students/GStinson/stinson.html
- http://en.wikipedia.org/wiki/Alar_Toomre

QUESTIONS?
COMMENTS?
SUGGESTINGS?

THANKS!


APPLAUSE !!

1D Plasma Simulation

Phys210 term project proposal

Magnus Haw Oct.20

Overview

- A plasma is a mixture that contains positively and negatively charged mobile particles
- System evolves over time and can develop instabilities and oscillations.

Project Goals

- To write a python code which models plasma behavior in one dimension using a particle-mesh approach (& visualize using ImageMagick).
- To establish validity of the implementation through convergence tests and comparison with known solutions
- To experiment with initial conditions (two particle test)

$$f(x,y,z,u_x,u_y,u_z) = \frac{N exp[-\frac{m}{2\kappa T}(u_x^2 + u_y^2 + u_z^2) - \frac{\Phi(x,y,z)}{\kappa T}]}{\int\int\int\int\int\int exp[-\frac{m}{2\kappa T}(u_x^2 + u_y^2 + u_z^2) - \frac{\Phi(x,y,z)}{\kappa T}] \;\; dx \;\; dy \;\; dz \;\; du_x \;\; du_y \;\; du_z}$$

Basic Assumptions

- 1 spatial dimension
- Non-relativistic & collision-less
- Strong magnetic field in x direction
- System is periodic in x
- System is composed of equal numbers of moving electrons and uniformly-spaced stationary protons.

1D Plasma Model

$$\frac{df}{dt} = -v\frac{\delta f}{\delta x} + \frac{e}{m_e}E\frac{\delta f}{\delta v} \tag{1}$$

$$E = -\frac{\delta \phi}{\delta x} \tag{2}$$

$$\frac{\delta^2 \phi}{\delta x^2} = -\frac{\rho}{\epsilon_0} \tag{3}$$

$$\rho = -e\int f(x, v, t)\delta v + \rho_0 \tag{4}$$

Numerical Methods (Particle Mesh single iteration)

1. Use particle position to assign an equivalent charge density to each mesh point
2. Calculate electric potential using Poisson’s equation
3. Find the electric field at each point on the mesh; interpolate force at each particle position.
4. Move each particle

Numerical Methods (cont'd)

Assigning charge to mesh points: divide charge of each particle between 2 nearest mesh points based on distance from each. This gives the charge density ρ at each mesh point.

$$- > \tag{1}$$

Calculating the potential at each grid point (solve by matrix methods):

$$\phi_{p-1} - 2\phi_p + \phi_{p+1} = \Delta x^2 \frac{\delta^2 \phi}{\delta x^2} + O(\Delta x^3) = -\Delta x^2 \frac{\rho_p}{\epsilon_0} \tag{2}$$

Find e-field:

$$E_p = -\frac{\phi_{p+1} - \phi_{p-1}}{\Delta x} \tag{3}$$

$$p = floor(x_i), f = x_i - p \tag{4}$$

$$E_{x_i} = (1-f) * E_p + f * E_{p+1} \tag{5}$$

Moving the particles ("n" exponents refer to time iteration):

$$\frac{v_i^{n+1/2} - v_i^{n-1/2}}{\Delta t} = \frac{e}{m_e} E(x_i^n) \tag{6}$$

$$v_i^{n+1/2} = \frac{x_i^{n+1} - x_i^n}{\Delta t} \tag{7}$$

Tests/Experiments

- Verify numerical convergence for smaller time steps.

- 2 particle test- 2 electrons; should oscillate.
- Experiment with various initial charge/velocity distributions

Timeline

Dates	Progress
Week 1	Finish research & code implementation
Week 2	Test code & debug
Week 3	Run numerical experiments & begin presentation & report
Week 4	Analyze results & continue work on presentation & report
Week 5	Practice presentation & work on report
Dec 4.	Submit final draft of report

References

1 Froese, Aaron., Comparison of Landau Damping in Two Computer Models, 2005 senior thesis
2 Uman, Martin A., Introduction to Plasma Physics. McGraw-Hill, New York, (1964).
3 Pang, Tao., An Introduction to Computational Physics. University Press, Cambridge, (2006)

Questions?


APPLAUSE !!

Physics 210 Term Project Proposal

Electrostatic Interaction and Equilibrium of an n-particle System on the Surface of a Sphere

By Ryan Lovelidge

October 20, 2009

Overview

- Charged particles interact according to Coulomb`s Law
- When confined to the surface of a sphere these particles will self arrange to their lowest energy state, i.e. the individual x, y and z components of the electrostatic forces from the other particles on the sphere will be in balance.

Note:

- In reality, the charged particles will not want to remain on the sphere so any force that is not tangential to the particles` position on the sphere will be discarded.
- A drag force will have to be incorporated to prevent the particles from moving indefinitely.

Project Goals

- To write a Matlab program to demonstrate the equilibrium positions of n-charged particles (for n greater than one) on the surface of a sphere.
- To test the program via repeated experiments based on the parameters of the formulae.
- To investigate a variety of initial conditions, including the number of particles interacting and their respective charges.

Mathematical Formulae

Newton's Second Law:

$$F_i = m_i a_i$$

The mass of the *i*th particle will be adjusted to keep the acceleration reasonable .

Coulomb's Law in vector form:

$$F_{E_i} = kq_i \sum_{n=2}^{j} \frac{q_j}{r_{ij}^2}$$

- F is the electric force
- k is Coulomb's constant
- q*i* and q*j* represent the charges of the *i*th and *j*th particles
- r*ij* is the separation distance between the *i*th and *j*th particles

$$F_{E_i} = m_i a_i$$

The above equation will provide a relationship between the acceleration of the *i*th particle and its charge.

Drag Force:

$$F_{drag_i} = -\gamma v_i$$

Gamma is a positive, adjustable constant which controls the amount of friction and v*i* is the velocity of the *i*th particle.

Numerical Approach and Testing

- These are difficult to carry out at this stage of the project, however once the formulae have been implemented in the code, testing will be done by cycling through values of the variables and evaluating the formulae in a repeating loop pattern to observe the resulting behaviour of the particles on the sphere.

Timeline

- October 21-27: Design the code that the equations will fit into
- October 28-November 3: Implement the code
- November 3-9: Test the code
- November 9-20: Run the numerical experiments
- November 21-27: Analyze the data and begin the presentation & report
- November 28-30: Finish the presentation and begin work on the report
- December 1: Give the presentation
- December 1-3: Finalize the report
- December 4: Submit the report


APPLAUSE !!

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

PHYS 210 Term Project Proposal

Earl Lin

October 20, 2009

Overview

- Using Newton's 2^{nd} Law $F = ma$ to find the position of any numbers of identical charged particles on a spherical surface at any given time.
- The particles will be stationary after equilibrium and maintain a special geometrical shape depending on the number of particles used.

Project Goals

- To write an MATLAB (octave) code which solves the equation numerically
- To test and investigate on various initial conditions for the equation to satisfy many known particle structure (i.e. linear, tetrahedral, octahedral, etc.)

Mathematical Formulation

- The motion on charges from Newton's Law

$$F_{net} = \sum m_i a_i$$

Where i, j intergers $j \neq i$

$$a_i(t) = (1/m_i m_j)[(1/4\pi\varepsilon_o)(q_i q_j)^2/(r_j - r_i)^2 - \gamma(v_i + v_j)]$$

- Use the equation to solve

$$a_i(t) = [v_i(t+dt) - v_i(t)]/dt$$

$$v_i(t) = [r_i(t+dt) - r_i(t)]/dt$$

- With initial conditions $1 < i < 4\pi r^2$, $1 \leq t \leq t_{max}$

Numerical Approach

Calculation is done in Cartesian coordinates

$1/4\pi\varepsilon_o = c_o$, $r_i = (x_i, y_i, z_i)$

Testing & Numerical Experiments

- Test the effect of particle that are not uniformly charged or have different mass
- Evaluate for any number of particles their half time for equilibrium

Project Timeline

Dates	Activities
10/26 – 10/30	DO basic research, derive equations and design code
11/02 – 11/06	START writing and testing code
11/09 – 11/13	CONTINUE writing an testing code
11/16 – 11/20	Test with numerical experiments, begin presentation & report
11/23 – 11/27	Compare results and finish presentation & report
11/30	Finalize Project Presentation
12/01	Project Presentation
12/02-12/03	Finalize Project Report
12/04	Hand in Project

References

- Wikipedia for Formula on Coulomb and Newton’s Law
- PHYS210 instructor Matthew W. Choptuik


APPLAUSE !!

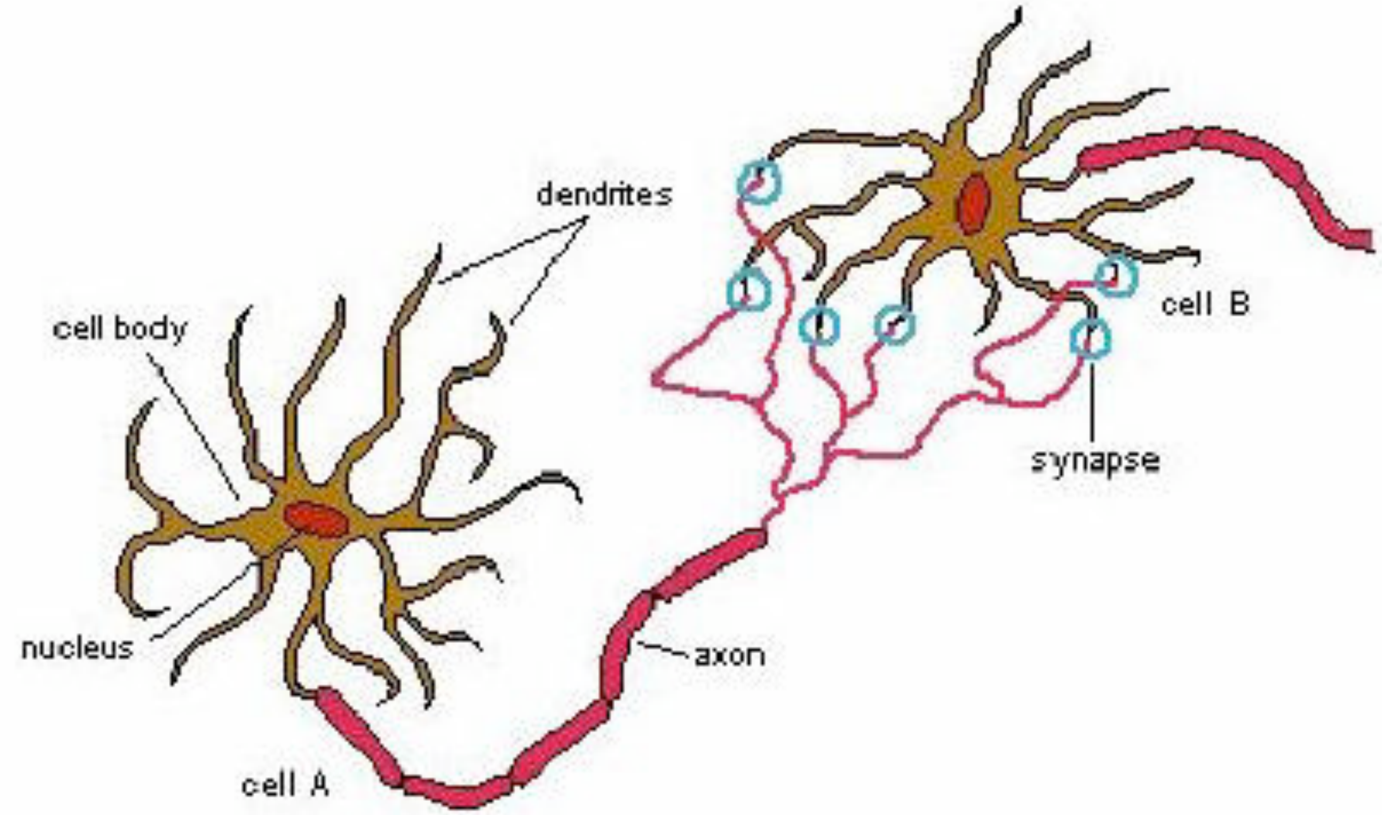
Neural Networks

Physics 210 Term Project Proposal

Carmen Huang

October 20, 2009

Overview



-the brain consists of many basic units called neurons

-neurons are electrically active and communicate with other neurons through electrical signals carried by dendrites and axons

-firing rate is a function of all its inputs, which is non linear

-often modeled as an on/off device

-*Model a neuron as a simple Ising spin*

-assume that the neuron has two possible states: firing and not firing ($s = \pm 1$)

Project Goals

-To write a MATLAB (octave) code which calculates the energy function of the Ising model

-To stimulate a neural network memory by creating patterns on an array of spins

-To calculate the interaction energies of the neurons

-To use the Monte Carlo procedure to ensure that the system will always evolve in time to states with the same or lower energy levels

Mathematical Formulation

-the energy function of the neural network/spin system

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

- $J_{i,j}$ is related to the strength of the synaptic connections
- sum is over all pairs of spins i and j in the network

-the equation to calculate the interaction energy to store a single pattern

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

-the equation to calculate the interaction energies to store many patterns

$$J_{i,j} = 1/M^* \Sigma\, s_i(m) s_j(m)$$

Numerical Approach

-the energy function enables the modeling of the neural net as an Ising model

-create a pattern on a 100×100 grid

-for our memory to recall a pattern, we require that the spin directions change in time in such a way that the spin configuration eventually ends up in the desired state

-if $\triangle E_{flip}$ is negative (if flipping the spin would lower the energy), the spin is reversed

-assume effective temperature is zero

-Monte Carlo rules ensure that the system will always evolve to states with the same or lower energy

Testing and Numerical Experiments

- Testing
  - Use Monte Carlo procedure repeatedly, giving every spin a chance to flip
  - Have the network find the desired pattern by locating the nearest energy minima
- Numerical Experiments
  - investigate patterns with different percent of randomness and find when the system is unable to find the correct pattern
  - investigate patterns that are rescaled or rotated

Project Timeline

Dates	Activities
10/20 - 10/28	Do basic research, derive equations and design code
10/30 - 11/05	Implement code
11/06 - 11/12	Test code
11/13 - 11/19	Run numerical experiments, begin presentation & report
11/20 - 11/26	Analyze data, continue work on presentation & report
11/27 - 12/02	Polish presentation and work on final draft of report
12/03	Give presentation
12/03	Finish final draft of report
12/04	Submit report

References

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

-http://www.cooper.edu/engineering/chemechem/MMC/tutor.html


APPLAUSE !!

Toomre model for Galaxy Collisions

Max Tims

Physics 210: Introduction to Computational Physics (Fall 2009)

October 20, 2009

Instructor: Matthew W. Choptuik

Picture: http://geology.com/news/2006/10/image-of-largest-galaxy-collision.html

Overview

- In 1964 Alar Toomre devised an instability criterion for differentially rotating disks now known as the Toomre Instability.

$$Q = \begin{cases} \dfrac{\sqrt{\langle c_r^2 \rangle}\kappa}{3.36 G\sigma_0} & \text{for stars,} \\ \dfrac{a\kappa}{\pi G \sigma_0} & \text{for gas,} \end{cases}$$

- Toomre conducted the first computer simulations of galaxy mergers in the 1970s with his brother Jüri Toomre. Although the small number of particles in the simulations obscured many processes in real galactic collisions, Toomre and Toomre were able to identify tidal tails in his simulations similar to those seen in the universe. I will attempt to also model galactic collisions but with an increased number of particles by using time-saving corrections to the equations of motion.

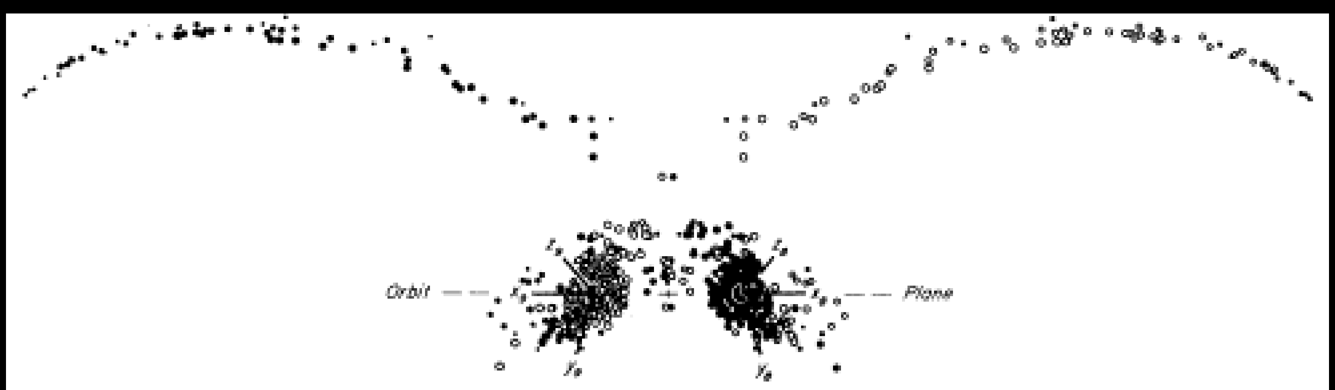


FIG. 23.—Symmetric model of NGC 4038/9. Here two identical disks of radius $0.75R_{\min}$ suffered an $e \approx 0.5$ encounter with orbit angles $i_8 = i_9 = 60°$ and $\omega_8 = \omega_9 = -30°$ that appeared the same to both. The above all-inclusive views of the debris and remnants of these disks have been drawn exactly normal and edge-on to the orbit plane; the latter viewing direction is itself 30° from the line connecting the two pericenters. The viewing time is $t = 15$, or slightly past apocenter. The filled and open symbols again disclose the original loyalties of the various test particles.

Top picture: http://upload.wikimedia.org/wikipedia/en/1/1e/ToomreandToomreF23.png
Bottom picture: http://upload.wikimedia.org/wikipedia/commons/c/c2/NGC40384039_large

Project Goals

- To write a MATLAB (octave) code using Toomre's Stability criteria and the general equations of gravity as they relate to galactic interactions
- To use the code to model various interactions; duplicate some of the common galaxy collisions that have been seen in space.

Left picture:http://www.sott.net/image/image/8378/galaxy_collision.jpg
Middle Picture: http://antwrp.gsfc.nasa.gov/apod/image/0807/ngc5426_gemini_big.jpg
Right Picture: http://mrbarlow.files.wordpress.com/2009/04/galaxy-collision.jpg

Mathematical Formulation

- The force on a particle α is simply summing the contributions form all the other particles in the simulation,

$$F_\alpha = \sum_{\beta \neq \alpha} G m_\beta \frac{r_\beta - r_\alpha}{\left| r_\beta - r_\alpha \right|^3} \qquad \text{(i)}$$

- As α and β approach each other closely, the force becomes large which is problematic. The divergence $F_{\alpha\beta}$ as $r_\alpha \to r_\beta$ is an awkward computation because it implies that the equations of motion of particles α and β have to be integrated with very small timesteps... This can slow the integration down to almost a halt.

- Thus, a **softening** must be employed,

$$F_\alpha = \sum_{\beta \neq \alpha} G m_\beta S_{\mathrm{F}}\left(\left| r_\beta - r_\alpha \right|\right) \frac{r_\beta - r_\alpha}{\left| r_\beta - r_\alpha \right|}$$

- The $\frac{1}{r^2}$ gets replaced by $S_F(r)$ (the *force softening kernel*) where $r = r_\beta - r_\alpha$.
- $S_F(r)$ tends to $\frac{1}{r^2}$ for values of the argument bigger than the *softening length ε,* thereby the numerical solution is well behaved without employing special regularization treatments for close encounters.
- The *softening kernel* that will be used is,

$$S(\mathrm{r}) = -\frac{r^2 + \frac{3}{2}\varepsilon^2}{(r^2 + \varepsilon^2)^{\frac{3}{2}}} \qquad \text{(ii)}$$

- We then write the equation of motion as

$$\ddot{r}_\alpha = -G \sum_{\beta=1;\beta\neq\alpha}^{N} \frac{m_\beta (r_\alpha - r_\beta)}{(\left|r_\alpha - r_\beta\right|^2 + \varepsilon^2)^{\frac{3}{2}}} \qquad \text{(iii)}$$

- where G is the gravitational constant and the summation is over all the other particles of mass m_β and coordinates r_β.

Numerical Approach

- Because most galaxies are very large (and the two I use will also be large) direct simulation is extremely time consuming. Therefore, it is necessary to utilize methods which speed up the calculations while retaining a “true collision”. Also, another consequence is that because of my computational capabilities I will have to limit number of particles to $N \leq 10^4$ (could be less).

- A spherical boundary of radius S containing total mass M has the equation of motion $\ddot{S} = -\frac{GM}{S^2}$ (iv).

  Co-moving coordinates for each galaxy $p_\alpha = r_\alpha / S^2$ are introduced by scaling the physical coordinates in terms of the scale factor.

- Softened potential= $m/(r^2+έ^2)^{1/2}$ where έ is the half-mass radius of a galaxy. where (from (i)) ε=έ/S

- Combining (i) and (iv), the corresponding co-moving equation of motion is

$$\ddot{p}_\alpha = -\frac{2\dot{S}}{S}\dot{p}_\alpha - \frac{G}{S^3}\left(\sum_{\beta=1;\beta\neq\alpha}^{N} \frac{m_\beta(r_\alpha - r_\beta)}{\left(\left|r_\alpha - r_\beta\right|^2 + \varepsilon^2\right)^{\frac{3}{2}}} - Mp_\alpha\right) \qquad \text{(v)}$$

Numerical Approach

- There are a few more time-smoothing methods I can use to help the equation be integrated,

$$t' = S^{3/2}$$

which gives the velocity relations

$$p'_\alpha = S^{3/2}\dot{p}_\alpha$$

$$v_\alpha = \frac{S' p_\alpha}{S^{3/2}} + \frac{p'_\alpha}{S^{1/2}}$$

- a second differentiation give us

$$p''_\alpha = -\frac{S'}{2S} p'_\alpha - G \sum_{\beta=1;\beta\neq\alpha}^{N} \frac{m_\beta (r_\alpha - r_\beta)}{(\left|r_\alpha - r_\beta\right|^2 + \varepsilon^2)^{\frac{3}{2}}} + GMp_\alpha \qquad \text{(vi)}$$

- Finally particles crossing the boundary may be subject to a mirror reflection in order to conserve the co-moving mean density. Any particle ($p_\alpha > 1$, $p'_\alpha > 0$) is assigned an equal negative co-moving radial velocity, then new polynomials are used to avoid discontinuity. The total energy corresponding would be,

$$\Delta E_\alpha = \frac{2 m_\alpha S' (S' p_\alpha + S p'_\alpha - S')}{S^3} \qquad \text{(vii)}$$

where p'_α is the old radial velocity.

Testing and Numerical Experiments

- Testing will involve running multiple simulations of different interactions.
- I will vary the angles of the galaxies as they collide, the speed at which they collide, and the size of the galaxies in order to model some of the real interactions we can see right now.
- As this project is much more of visuals based project, the numerical experiments are harder to evaluate.
- I could however figure out the various equations at different time intervals to see how the speeds of the galaxies change, the number of stars still left in the galaxy after the collision, etc.

Project Timeline

Dates	Goals
10/21-10/28	Basic research, derive equations and design cod
10/29-11/05	Implement code
11/06-11/12	Test code
11/12-11/19	Run numerical experiments, begin presentation & report
11/20-11/26	Analyze data, continue work on presentation & report
11/27-11/30	Polish presentation and work on final draft of report
12/01-12/03	Finish final draft of report
12/03	Give presentation
12/04	Submit report

References

- Galactic dynamics and n-body simulations : lectures held at the Astrophysics School VI, organized by the European Astrophysics Doctoral Network (EADN) in Thessaloniki, Greece, 13-23 July 1993 / G. Contopoulos, N. Spyrou, L. Vlahos, (eds.)
- Collisions, rings, and other Newtonian N-body problems / Donald G. Saari, American Mathematical Society, c2005.
- Galactic dynamics / James Binney and Scott Tremaine, Princeton : Princeton University Press, c2008.
- Disk Stability of the Milky Way Near the Sun, LUO Xin-lian, PENG, Department of Astronomy, Nanjing University, Nanjing 210093, Chinese Academy of Sciences-Peking University Joint Beijing Astrophysical Center,Galactic-scale star formation by gravitational instability Mordecai-Mark Mac Low1, Yuexing Li1,2 and Ralf S. Klessen3
- Bate, M. R., Bonnell, I. A. & Price, N. M. 1995, MNRAS
- http://www.galaxydynamics.org/
- http://www.ifa.hawaii.edu/~barnes/transform.html
- http://burro.astr.cwru.edu/models/models.html
- http://en.wikipedia.org/wiki/Alar_Toomre


APPLAUSE !!

Effects of a Central Mass on Binary Formation

PHYS 210 project proposal

Joshua Wienands

Oct. 20th, 2009

Overview and Goals

- Use finite difference techniques to investigate the effects of a large central mass on binary formation in a group of N gravitationally interacting bodies
- Code an N-body simulation, run both with and without a central mass
- Code in C for efficiency and larger possible N

Mathematical Formulation

- $g(r) = - Gm_2/r^2$; $r^2 = (x_2-x_1)^2 + (y_2 - y_1)^2$
- $F_g = g(r)*m_1 = m_1*a(t) = d^2r/dt^2$

- $F_{gx} = F_g*(x_2-x_1)/r$
- $v_x(t) = v_{0x} + \int a_x(t)dt$
- $x(t) = x_0 + \int v_x(t)dt$
- Same formulae for y direction

Mathematical Formulation

- Central Mass
- This would be treated like any other body, but with specific initial position (origin), velocity (0), and mass (adjustable, but much larger than the other masses)

Numerical Approach

- Discretize time:

  $t = nh; \;\; n = 0, 1, 2, \ldots, n_{max}$

- Use adaptive time steps for close bodies

  \- if distance b/t two bodies is small, calculate in-between positions for those for higher accuracy

Numerical Approach

Approximate equations (Euler Method):

$v_x(t + h) = v_x(t) + h*a_x(t) + O(h^2)$

$x(t + h) = x(t) + h*v_x(t) + O(h^2)$

- Have error order h^2
- 4th order Runge-Kutta method
- Error of order h^5

Runge-Kutta Method

- $x(t_n + h) = x(t_n) + 1/6(k_1 + 2k_2 + 2k_3 + k_4)$
- $k_1 = h*f(x(t_n))$
- $k_2 = h*f(x(t_n + h/2) + k_1/2)$
- $k_3 = h*f(x(t_n + h/2) + k_2/2)$
- $k_4 = h*f(x(t_n + h) + k_3)$
- In this case, $f(x(t_n))$ is the acceleration at time t_n

Advantages/Disadvantages

- The 4th order Runge-Kutta Method is more accurate, but is more expensive
- It may be faster and just as accurate to use the Euler Method, with a smaller discretization
- Test with relatively small N

Testing & Numerical Experiments

- Keep track of "constants of motion" and make sure error converges at smaller discretizations

- Run at several N, both with and without a central mass
- Compare the frequency of binary formation and the duration

Timeline

10/21-10/26	Research, Design code
10/27-11/01	Implement Code: static time steps
11/02-11/05	Test Code
11/06-11/10	Implement Code: adaptive time steps
11/11-11/15	Test Code
11/16-11/23	Run Experiments, begin work on report
11/24-11/30	Analyze Data, continue work on report
12/01-12/03	Finish report
12/03	Give presentation, finalize report
12/04	Submit

http://en.wikipedia.org/wiki/Runge–Kutta_methods
http://en.wikipedia.org/wiki/Newtonian_gravity

Questions?


APPLAUSE !!

Two Lane Highway Traffic Simulations Using the Nagel-Schreckenberg-Model

Term Project Proposal

Andy Lin

Overview

- Will be simulating two lane highway traffic with cellular automata.
- NS Model gives a set of rules that each cell will have to obey which creates a realistic model of traffic
- The rules will cover the acceleration, deceleration and randomness of drivers

Project Goals

- To research and implement lane changing rules in order accurately simulate the two lane aspect of traffic
- To write a code which will properly abide by the NS model and also lane changing rules.
- To investigate the effects that varying conditions max speed and traffic density will have on flow rate

Mathematical Formulation

- The transition probability, W, of a car moving from one cell to the next cell is written as:

$$W = (1 - p)\frac{1+\sigma_i}{2}$$

where $\sigma_i = \pm 1$,

+1 for occupied cells and -1 for empty cells, and p is a randomization parameter.

-The initial velocity of the cars is V_{max} which will be vary for different tests depending on different speed limits.

- First test will use V_{max} = 5 sites per time-step which corresponds to 90km/hr.

Mathematical Formulation (continued)

-The flow rate, q, at a specific cell, i, is give by the equation:

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

where $n_{i,i+1}(\ ) = 1$ when a car is detected between cells i and i+1

Testing and Numerical Experiments

- Investigate how different V_{max} and traffic density will effect the flow rate of the cars
- Will plot flow rate vs. traffic density graphs and correlate them to real traffic data

Project Timeline

Dates	Work
10/21 – 10/27	Basic research and design code
10/28 – 11/10	Implement and Test code
11/11 – 11/17	Run experiments, begin presentation and report
11/18 – 11/24	Analyze data, work on presentation and report
11/25 – 12/02	Finish presentation and report
12/03	Final presentation
12/04	Hand in report

References

- Nagel and Schreckenberg, "A cellular automaton model for freeway traffic" J Phys I France 2 (1002) 2221-2229
- A. Benyoussef et al, “Traffic Flow in a 1D Cellular Automaton Model with Open Boundaries”, Journal of Physics 428-440 (2001)


APPLAUSE !!