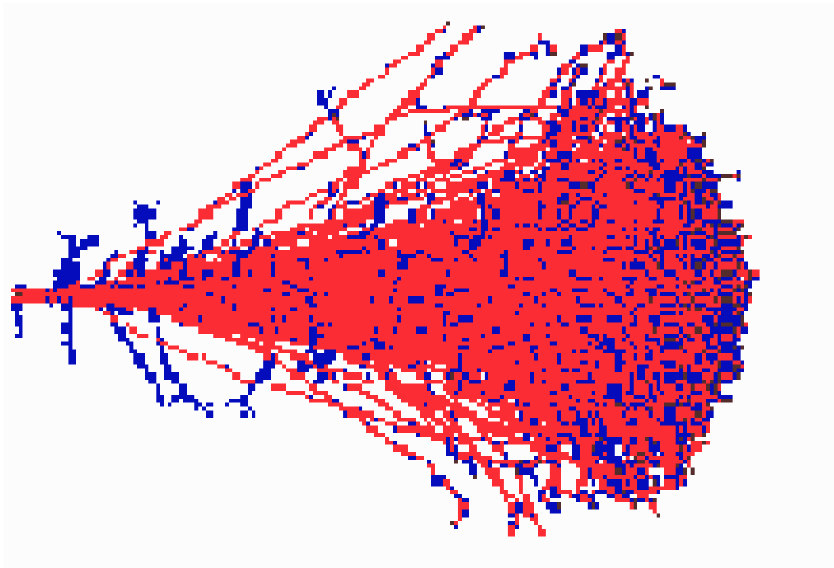
The Stopping of Ions in Matter



University of British Columbia

PHYS 210 Term Project Proposal

Sen Mei

October 22, 2009

Project Overview

- When an ion beam travel through matter, the ions lose energy and are scattered.
- If the target material is solid, the ions will eventually stop inside the material.
- The stopping distance depends on the mass and atomic number of the ions and target atoms, the initial energy of the ion, the displacement energy of target material, and the scattering effects due to interatomic collisions inside the target.

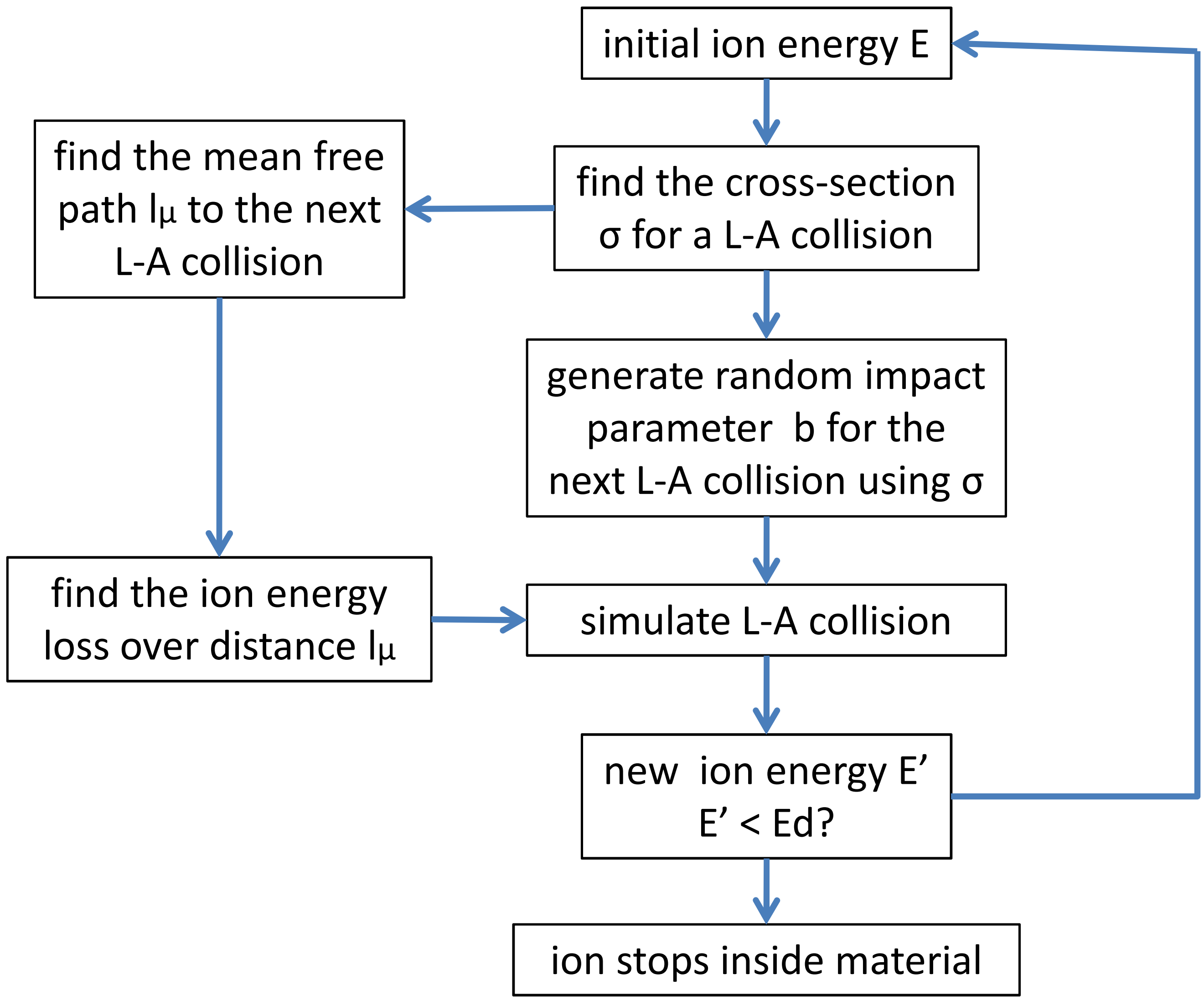
Project Goals

- To create a series of BASH and Maple scripts that simulate the stopping of ions in matter for a given set of parameters, using a Monte Carlo process to model large-angel collisions.
- To establish the correctness of the implementation by comparing test results with those produce by SRIM and/or Geant4 toolkit (which are validated by experimental data)
- To investigate common scenarios, such as alpha particle shielding.

Simulation Overview

- The simulation makes the following assumptions:
  - ions are independent of each other, thus only two-body collisions are considered;
  - no accumulated damage to the target material;
  - collisions that significantly alter trajectory of the ion (large-angle collisions) take place after the ion travels a certain distance from the previous such collision.
- The same simulation is done for each ion, the number of ions used will determine the quality of the results. For each ion, the simulation process will be outlined on the following slides.

Simulation Overview



Mathematical Formulation: Scattering Angle

- Input: m_{ion}, m_{tgt}, Z_{ion}, Z_{tgt}, ion energy E, and impact parameter b
- Output: scattering angle θ in center-of-mass frame

$$\theta_c = \pi - 2\beta \int_{x_0}^{\infty} \frac{f(r)}{r^2} dr$$

Where:

$$f(r) = \sqrt{1 - \frac{\phi(r)}{r\varepsilon} - \frac{\beta^2}{r^2}} \quad ; \qquad \beta = \frac{b}{a} \quad ; \qquad a = \frac{0.8854 a_{Bohr}}{Z_{ion}^{0.23} + Z_{tgt}^{0.23}}$$

$$\varepsilon = \frac{E_c \cdot a}{Z_{ion} \cdot Z_{tgt} \cdot e^2} \quad ; \qquad E_c = \left(\frac{m_{tgt}}{m_{ion} + m_{tgt}} \right) E$$

$$\phi(r) = \sum_{i=1}^{4} A_i \exp\left(-B_i \frac{r}{a} \right);$$

A_i, B_i are sequences of 4 constants

Numerical Approach to Scattering Angle

- I will instead compute θ_c using the algorithm employed in Geant4, which computes the integral using Lobatto quadrature of order 6

$$\theta_c = \pi - \frac{\pi\beta\alpha}{x_0}$$

Where:

$$\alpha \approx \frac{1+\lambda_0}{30} + \sum_{i=1}^{4} w_i\, f\left(\frac{x_0}{q_i}\right) \qquad \lambda_0 = \left(\frac{1}{2} + \frac{\beta^2}{2x_0^2} - \frac{\phi(x_0)}{2\varepsilon}\right)^{-1/2}$$

- We can the find the new ion energy and lab scattering angle by:

$$E' = E\left(1 - \frac{2m_{ion}m_{tgt}}{(m_{ion}+m_{tgt})^2}(1-\cos\theta_c)\right) \qquad \tan\theta = \frac{\frac{m_{tgt}}{m_{ion}}\sin\theta_c}{1+\frac{m_{tgt}}{m_{ion}}\cos\theta_c}$$

Mathematical Formulation (continued)

- For a uniformly distributed number k between 0 and 1:

$$b = \frac{1}{2}\sqrt{\frac{-\log k}{\pi N l_\mu}} \quad \text{if} \quad \sigma \le \pi\left(\frac{N^{-1/3}}{2}\right)^2$$

else: $$b = \frac{1}{2}\sqrt{k}$$

- l_μ is determined by: $$l_\mu = \frac{1}{\sigma N}$$

- The ion energy loss over distance l_μ is determined by:

$$E_{loss} = l_\mu \frac{2\pi N \cdot Z_{tgt}(Z_{ion})^2_{eff} \cdot e^4 \dfrac{m_{ion}}{m_{tgt}}}{E} \cdot \ln\left(\frac{4E}{11.5\dfrac{m_{ion}}{m_{atom}} Z_{ion}}\right)$$

Mathematical Formulation (continued)

- Now we are back at fining σ, the cross section for L-A collision:

  $$\sigma = 2\pi b_{\mu}$$

  where b is the largest impact parameter that will result in a predefined minimum energy loss below which collisions are considered small-angel collisions.

  $$E_{\min} = E\frac{4m_{ion}m_{tgt}}{(m_{ion}+m_{tgt})}\sin^2\left(\frac{\theta_c}{2}\right)$$

Visualization, Testing and Numerical Experiments

- The path of the each ion and the stopping range distribution will be plotted using the plotting facility of Maple or gnuplot.

- Test that the simulation exhibits the expected behaviour as each input is individually modified.
- The main check for the implementation will be done by comparing outputs generated by the program to those generated on the same inputs by SRIM or Geant4 toolkit .

- Investigate the effectiveness of different materials as shielding against alpha particles and various nuclear fragments produced in fission reactions

Project Timeline

Dates	Activities
Oct. 23 – Oct. 29	Do basic research, complete mathematical formulations, design code
Oct. 30 – Nov. 5	Implement code on the simulation for each ion
Nov. 6 – Nov. 12	Finish implementation
Nov. 13 – Nov.19	Testing and debugging code, start working on final presentation
Nov. 20 – Nov. 26	Run numerical experiments, start working on final report
Nov. 27 – Nov. 30	Polish final presentation
Dec. 1	Give final presentation
Dec. 1 – Dec. 3	Complete the final report
Dec. 4	Submit the final report

References:

An algorithm for computing screened Coulomb scattering in GEANT4

M.H. Mendenhall, R.A. Weller / Nucl. Instr. and Meth. in Phys. Res. B 227 (2005) 420–430

Notes on simulation of primary damage production by Monte Carlo methods SRIM

http://www.nuc.berkeley.edu/courses/classes/NE-220/


APPLAUSE !!

Modeling N Charges on a Sphere

Physics 210 Term Project Proposal

Steven Janssens — 1 November 2009

Overview

- N like charges bound to a spherical surface will seek to maximize their distance from one another and thus minimize their potential energy
- Coulomb's Law describes the electrostatic force reponsible for this behavior

Project Goals

- To write a program that models N charges on a sphere beginning at either random locations or at specific point(s) and falling to their equilibrium positions
- Study the equilibrium configurations and the lattice structure that results
- With time permitting, generalize model to other shapes

Mathematical Formulation

- The Coulomb force on a particular charged particle, j, is the sum of the Coulomb forces exerted on it by all the other particles is given by,

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

- A frictional force proportional to velocity will also need to be implemented to dissipate energy from the system allowing the charges to reach an equilibrium state given by,

$$F_f = -\gamma v \tag{2}$$

Numerical Approach

- Finite difference approximation will be used to solve the differential equations
- Initial positions will either be set randomly or specifically set
- Particles will initially be at rest

Testing & Numerical Experiments

- Examine how the number of time steps required for the solution to reach equilibrium varies with increasing the number of particles N and with a variety of initial conditions
- Examine the lattice structure that forms among the particles and examine how it changes with increasing N and different initial conditions
- Visualization and Plotting Tools
  - xfpp3d/pp3d will be used to create animations
  - PGPLOT will be used for any plots in my report

Project Timeline

Dates	Activities
Now – Nov. 6	Research, derive equations & design
Nov. 7 – Nov. 20	Implement & test the code
Nov. 21 – Nov. 25	Run numerical experiments, begin presentation & report
Nov. 25 – Nov. 28	Analyze data, continue presentation & report
Nov. 29 – Nov. 30	Polish presentation & work on report
Dec. 1	Give presentation
Dec. 1 – Dec. 3	Finish report
Dec. 4	Submit report

References

- Erber, T., Hockney, G. M., 1999, J. Phys. A: Math. Gen, 24, L1369
- http://en.wikipedia.org/wiki/Coulomb's_law


APPLAUSE !!

Fire simulations

Using cellular automata

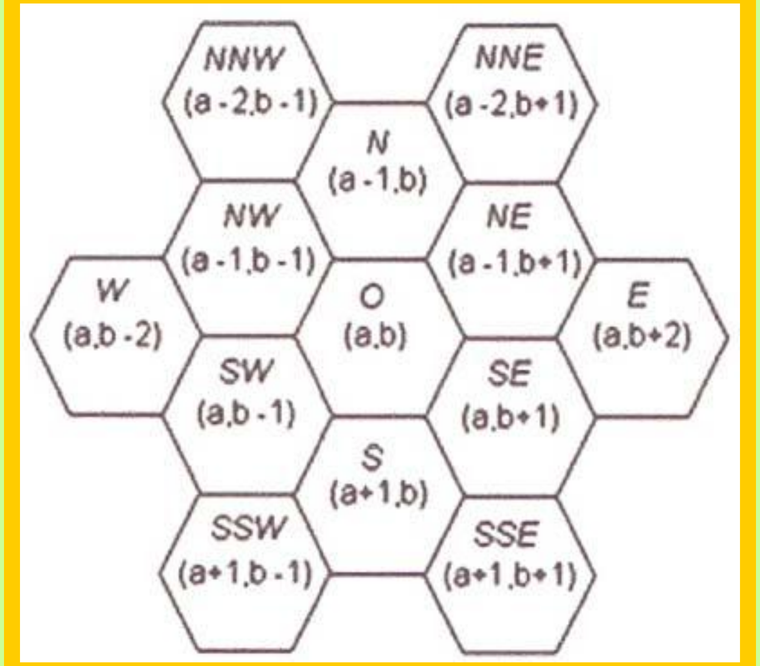
Phys 210 Term Project Proposal

Katherine Van Dijk

October 15, 2009

Overview

- Cellular automata make a good forest fire simulations with some complexity.
- Other models may use wave propagation techniques to determine temperature fields and fire propagation. (Uses turbulent fluid flow calculations)
- I will use hexagonal cells, where the function of each will probably only rely on the 6 close neighbors.
- Maintain flat surface of forest and no wind (for simplicity)



Project Goals

- Write a Matlab procedure to simulate the spread of fires using 2D cellular automata.
- Learn more about programming and the use of cellular automata to do simulations.
- Create a model that qualitatively agrees with actual forest fires.
- Investigate a variety of initial conditions of spread rates. Determine how long it takes for a percentage of forest to burn for specific conditions, for my model.

Mathematical Formulation

- $S_{ab}(t+1) = g * (S_{\{a+\alpha, b+\beta\}}(t) + {}_{\{(\alpha,\beta)€V_n\}}\Sigma\, [\mu_{\{\alpha\beta\}}(a, b) * S_{\{a+\alpha,b+\beta\}}(t)] + {}_{\{(\alpha,\beta)€V_d\}}\Sigma\, [\mu_{\{\alpha\beta\}}(a,b) * S_{\{a+\alpha, b+\beta\}}(t)])$
- $\mu_{\{\alpha\beta\}}(a,b) = \omega_{\{\alpha\beta\}}(a,b) * h_{\{\alpha\beta\}}(a,b) * r_{\{\alpha\beta\}}(a,b)$
- $S_{ab}(t) =$ [burned out area of (a,b) at time t] / [total area of (a,b)]

- $\mu_{\alpha\beta}(a,b)$ = function of wind; $\omega_{\alpha\beta}(a,b)$, height/topography; $h_{\alpha\beta}(a,b)$, and fire spread rate; $r_{\alpha\beta}(a,b)$.
- g = discretization function
- S_{ab} = state of current cell, function of it's own state one time step earlier, and all its neighbors.
  - So a function of up to 13 variables in hexagonal cells, where distant neighbors are included.

Numerical Approach

- Graphs & Diagrams to show fire fronts and propagation.
- Calculate the number of time steps required to burn a percentage of forest area for given situations in my model.

Testing and Numerical Experiments

- Compare constant rates of spread with known fire propagation front.
- Vary spread tendency, e.g. how likely an area (cell) will set on fire.
- May incorporate a “random” wind factor, therefore the cells with have some random chance to set on fire without a direct neighbor on fire.
  - Would be the equivalent of changing winds
- How long will it take to stop burning given certain spread rates and initial amount of burning forest?
  - Plot graphs for model

Project Timeline

Dates	Activities
Oct 23 – Oct 29	Research, derive equations & design code
Oct 30 – Nov 05	Design & Implement code
Nov 06 – Nov 12	Test code
Nov 13 – Nov 19	Run experiments, begin presentation & report
Nov 20 – Nov 26	Analyze data, cont. pres. & report
Nov 27 – Nov 30	Polish presentation and work on final draft of report
Dec 01 – Dec 03	Give presentation & Finish final draft of report
Dec 04 (11:59 pm)	Term project due

References

- “Simulation of forest fire fronts using cellular automata” Authors: A. Hernández Encinas, L. Hernandez Encinas, S. Hoya White, A. Martín del Rey, G. Rodríguez Sánchez **ScienceDirect** *Advances in Engineering Software* 38 (2007) 372-278

- “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

- “Self organized criticality and cellular automata” Author: Micheal Creutz, April 27, 2007

- http://demonstrations.wolfram.com/ForestFireSimulationUsingARandomizedHexagonalAutomaton/

- http://www.shodor.org/interactivate/activities/Fire/?version=1.6.0_14&browser=MSIE&vendor=Sun_Microsystems_Inc.&flash=10.0.22


APPLAUSE !!

Numerical Analysis of 1D Time-Dependent Schrödinger Equation

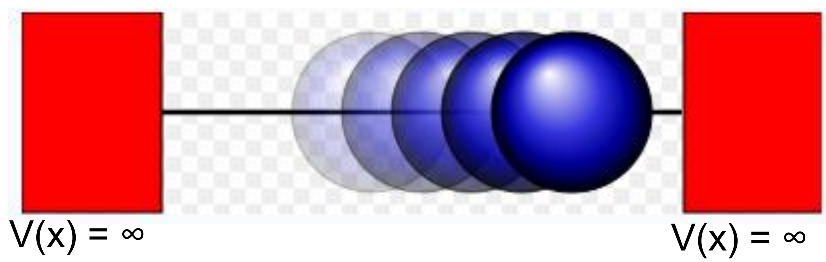
PHYS 219 Term Project Proposal

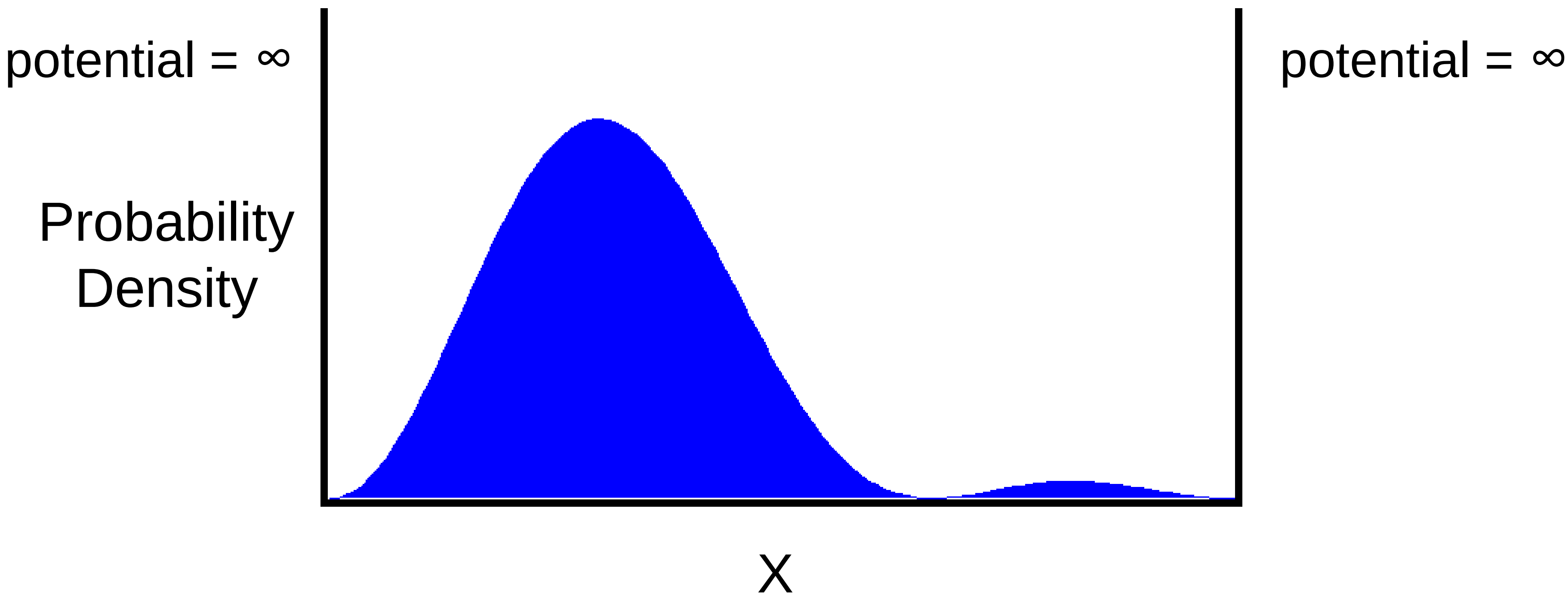
Anffany Chen

October 22, 2009

Overview: Schrödinger Equation

- Schrödinger Equation is used to predict the future behavior of a dynamic system.
- Particle in a box





Project Goals

- To write an MATLAB code which solves the 1D Time-Dependent Schrödinger Equation numerically: a program that takes input $\mathbf{x_0,t_0}$ and outputs $\mathbf{\Psi(x_0,t_0)}$

- To plot the time evolution of the probability density of a particle in a box based on the numerical results

- To test the implementation of the code by checking if the outputs are in agreement with the **conservation of total probability**

- To investigate various **initial conditions** and **potential energy equations**

Mathematical Formulation: 1D Time-Dependent Schrödinger Equation

- a wave equation in terms of the wavefunction Ψ(x,t)

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\,\psi(x,t)$$

where $|\Psi(x,t)|^2$ ≡ probability density
V(x) ≡ potential energy formula

- Domain: $0 \le X \le 1$, $t \ge 0$

- Initial conditions: $\psi = \sqrt{2}\sin\pi x$ at $t = 0$

- Boundary conditions: $\psi_1^{n+1} = \psi_{\mathbf{nx}}^{n+1} = 0$

But Ψ(x,t) cannot be solved algebraically in most cases!

Numerical Approach: Crank-Nicolson Method

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\,\psi(x,t)$$

$$\downarrow$$

$$i\frac{\psi_j^{n+1}-\psi_j^n}{\triangle t} = -\frac{1}{2}\left(\frac{\psi_{j+1}^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1}}{\triangle x^2}+\frac{\psi_{j+1}^{n}-2\psi_j^{n}+\psi_{j-1}^{n}}{\triangle x^2}\right) + V_j\ \frac{1}{2}\left(\psi_j^{n+1}+\psi_j^n\right)$$

where $\psi_j^n \equiv \psi(j\triangle x, n\triangle t)$, $V_j \equiv V(j\triangle x)$

$$\downarrow$$

$$\mathrm{C}_j^{+}\psi_{j+1}^{n+1} + C_j^{o}\psi_j^{n+1} + C_j^{-}\psi_{j-1}^{n+1} = -C_j^{+}\psi_{j+1}^{n} - C_j^{o*}\psi_j^{n} - C_j^{-}\psi_{j-1}^{n}$$

where $\mathrm{C}_j^{+} = C_j^{-} = \frac{-1}{2\Delta x^2}$

$$\mathrm{C}_j^{o} = \frac{1}{\Delta x^2} - \frac{i}{\Delta t} + \frac{V_j}{2}$$

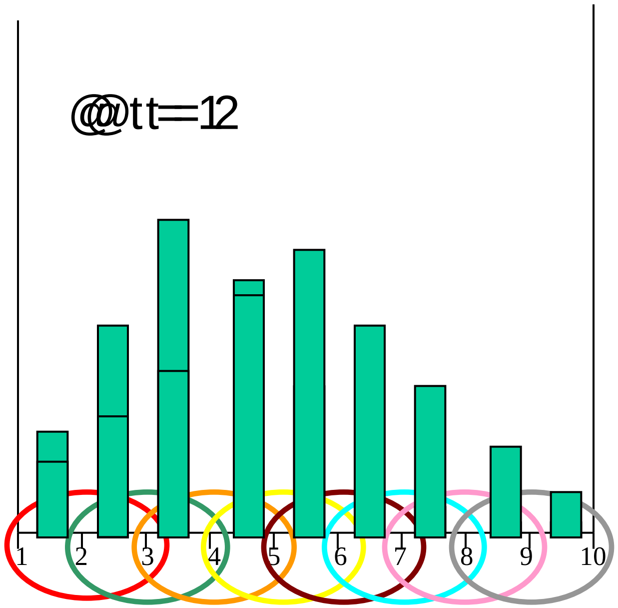
Numerical Approach: Crank-Nicolson Method (continued)

$$C_j^+ \psi_{j+1}^{n+1} + C_j^o \psi_j^{n+1} + C_j^- \psi_{j-1}^{n+1} = -C_j^+ \psi_{j+1}^n - C_j^{o*} \psi_j^n - C_j^- \psi_{j-1}^n$$

$$\psi = \sqrt{2} \sin \pi x$$

$$C_j^+ \psi_1^1 + C_j^o \psi_2^1 + C_j^- \psi_3^1 = -C_j^+ \psi_1^0 - C_j^{o*} \psi_2^0 - C_j^- \psi_3^0$$
$$C_j^+ \psi_2^1 + C_j^o \psi_3^1 + C_j^- \psi_4^1 = -C_j^+ \psi_2^0 - C_j^{o*} \psi_3^0 - C_j^- \psi_4^0$$

V = ∞



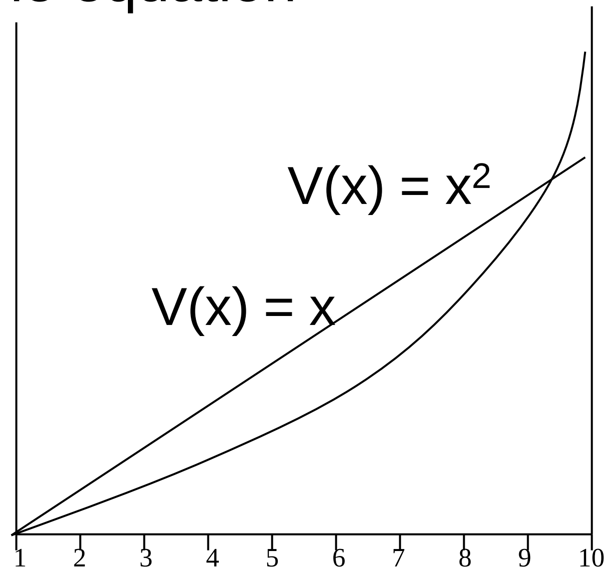
V = ∞

Testing & Numerical Experiments

Numerical Experiments

- Investigate various initial conditions and potential energy equations for the equation

$$\psi = \sqrt{2}\sin \pi x$$



V(x) = sinx!

Testing

- Conservation of total probability

$$I \equiv \int_0^1 \psi(x,t)\,\psi^*(x,t)\,dx = 1$$

Project Timeline

Dates	Activities
10/23 – 10/29	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

"Hints for Solution of the Time-Dependent Schrödinger Equation" http://laplace.physics.ubc.ca/People/matt/Teaching/05Fall/PHYS410/Projects/schrodinger.pdf

Griffiths, David J. *Introduction to quantum mechanics*. Pearson Education. Toronto: 2005.

"Particle in a box" http://en.wikipedia.org/wiki/Particle_in_a_box


APPLAUSE !!

Ray Tracing Through Series of Lenses and Spherical Mirrors

PHYS 210 Term Project Proposal

Ye Cheng Chen (Lewis)

October 22 2009

Overview

- Ray tracing is a graphical method for locating images of objects formed by rays passing through mirrors or lenses from multiple directions.
- It is mainly used for checking against the results of algebraic computations (of image distance, focal length, etc) obtained from equations of mirror or lens.

Project Goals

- To write a GUI code for setting up a lens and another code for spherical mirror, allowing rays to pass through them and converge to or diverge away from focal point.
- To write a MATLAB code which easily solves the equations of spherical mirrors and lenses for focal length, radius, etc.
- To check the results obtained from the graphical setting and the ones obtained from MATLAB.
- To ensure the rays are bounded by a rectangular box, allowing rays move continuously until exit.

Mirror Reflection and Equation

- The angle of incidence equals the angle of reflection when a ray is reflected by a spherical mirror with the presence of its curvature.
- The mirror better be double-sided, so that the ray can be reflected on both sides. Cool!
- I will use the concept above to set up the mirror and use MATHLAB to solve the mirror equation below numerically (easy part)

$$\frac{1}{o} + \frac{1}{i} = \frac{1}{f}$$

- Where o is the distance of the object from the mirror
- i is the distance of the image from the mirror
- f is the focal length of the mirror

Lens Refraction and Equation

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$

- This formula is used to predict the path of rays entering and leaving a lens that has an index of refraction (though I am not sure if this formula helps me with lenses. Investigation is ongoing).
- The angle of refraction is associated with the center of curvature of a lens, C.
- The equation for numerical computation is

$$\frac{1}{f} = (n-1)(\frac{1}{r_1} - \frac{1}{r_2})$$

- Where n is the index of refraction.

Challenges and Visualization

- I need a 2-D Cartesian coordinate graphic tool to set up the system.
- Notice this project proposal aims for paraxial rays entering lenses and mirrors. The biggest challenge is how I can set up the lens such that a ray passing through the lens does not depend on the focal point but only the index of refraction of the lens and the angle of incidence.
- Once the challenge is solved, and when time is permitting, I am interested in showing how non-paraxial rays behave when they pass through a lens and possibly generate dispersion of polychromatic (multiple color) rays, and such dispersion is presumed to depend on the index of refraction of the lens and wavelength only.
- It’s going to be hard but interesting.
- Possible parameters in GUI code are index of refraction of lens (n), curvature of lenses and mirrors (C), initial angle (θ), and probably wavelength (w) (I am not even sure at this point).
- I will eventually use xvs or xsffs for visualization of my ray tracing, but I should start this with Python since I knew it has a pretty good GUI tool.

Project Timeline

Date	Activities
10/24—10/31	Do basic research, derive equations, design and implement code
11/01—11/07	Test Code and possibly challenge on additional features
11/08—11/15	Run numerical experiments, begin presentation and report, decide to add features or delete them due to its difficulty
11/16—11/23	Analyze data, continue work on presentation and report
11/23—11/30	Polish presentation and work on final draft of report
12/01	Give presentation
12/02—12/03	Finish final draft of report
12/04	Submit Report

References

- David Halliday, Robert Resnick, Kenneth S. Krane, *Physics*, John Wiley & Sons, Inc., New York, 2002


APPLAUSE !!

Solving 2D Poisson Equations for Electrostatics

Phys 210 Term Project

Brian Sin

Oct 21st, 2009

Overview

- **Poisson's equation** is a partial differential equation with broad utility in electrostatics, mechanical engineering and theoretical physics (http://en.wikipedia.org/wiki/Poisson's_equation\)
- For this project, only electrostatics will be concerned with

Objective

- To create a Matlab (octave) code which implements a finite difference approximation method to solve 2D Poisson equations for electrostatics
- To investigate a variety of initial conditions for the equation
- To present all solutions with a visualization

Poisson Equation

- In 3D:

$$\left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2} + \frac{d^2}{dz^2}\right)u(x, y, z) = f(x, y, z)$$

- In 2D:

$$\left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2}\right)u(x, y) = f(x, y)$$

Where, ***f*** is the source function

Pre-analysis

- Poisson Equation:

$$\Delta u = f(x, y)$$

- For vanishing ***f***, this equation becomes:

$$\Delta u = 0$$

- In Cartesian coordinates:

$$\Delta u = \frac{d^2 u}{dx^2} + \frac{d^2 u}{dy^2} = 0$$

- In polar coordinates:

$$\Delta u = \frac{1}{r}\frac{d}{dr}\left(r\frac{du}{dr}\right) + \frac{1}{r^2}\left(\frac{d^2 u}{d\theta^2}\right) = 0$$

Pre-analysis (cont'd)

- Angle becomes insignificant as ***r*** becomes arbitrarily large. Therefore, we can ignore the second term:

$$\Delta u = \frac{1}{r}\frac{d}{dr}\left(r\frac{du}{dr}\right) = 0$$

- For this equality to be true:

$$r\frac{du}{dr} = \text{constant}$$

$$\therefore \frac{du}{dr} = \frac{\text{constant}}{r}$$

- **u** behaves like a decay function

Initial Domain

- Unit square:

$$0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

Initial Boundaries

- Dirichlet conditions:

$$u(0, y) = 0$$
$$u(x, 0) = 0$$

- Neumann conditions:

$$\frac{du}{dy} = 0 \text{ when } x = 1 \quad \frac{du}{dx} = 0 \text{ when } y = 1$$

Timeline

- 10/23-10/29 design code
- 10/30-11/05 Implement code
- 11/06-11/12 Test code
- 11/13-11/19 experiments, begin presentation & report
- 11/20-11/26 Analyze data, continue work on presentation & report
- 11/27-11/30 Polish presentation and work on nal draft of report
- 12/01 Give presentation
- 12/01-12/03 Finish final draft of report
- 12/04 Submit report

Thanks for your attention


APPLAUSE !!

Formation of Aggregates in an N-Body Gravitational System

PHYS 210 Term Project Proposal

Ian Fraser
October 22, 2009

Overview:

- An n-body gravitational particle simulation in two dimensions using finite difference approximations
- Looking at the formation of aggregates in this system

Project Goals:

- To write a MATLAB (octave) code which numerically calculates the motions of gravitational particles
- To create a realistic simulation of the formation of aggregates
- To look at varying initial conditions and see how they affect the formation of aggregates

Mathematical Formulation (Equations of Motion)

- The vector equation of Force in this system is:

$$\vec{F} = \frac{G m_1 m_2}{(|r_1 - r_2|)^2} \cdot \hat{r}_{12} \qquad (1)$$

- Where G is the gravitational constant, m is mass of the particle, $|r_1 - r_2|$ is the distance between particles, r_{12} is the unit vector from particle 1 to 2; and F is the force on particle 1 from 2.
- The total force on one particle is the summation of all the forces from all the other particles:

$$\vec{F}_T = \Sigma \vec{F} \qquad (2)$$

- From this the acceleration can be calculated for the particle:

$$\vec{a}_1 = \frac{\vec{F}_T}{m_1} \qquad (3)$$

Numerical Approach

- The problem is then a set of second order ordinary differential equations, relating the acceleration to the position of all the particles in the system

$$\partial^2 \vec{r}_i / \partial t^2 = \vec{F}_i / m_i \qquad (4)$$

$$\vec{v}_i \equiv \partial \vec{r}_i / \partial t \qquad (5)$$

- With the initial conditions of the position, r_1, the velocity, v_1, and the mass, m, known for each particle there then exists a unique solution.
- This can then be solved using Euler's Method

Visualization

- I will use the visualization and plotting tools of MATLAB

Testing & Numerical Experiments

- Testing
  - First, model simple 2-3 particle interactions, making sure that particles do merge properly with no extreme adverse effects
  - Adjust energy loss when aggregates are formed to make sure system can come to an equilibrium
- Numerical Experiments
  - Investigate if initial conditions are symmetric will aggregate formations be symmetric
  - Investigate the number of aggregates that form based on varying initial conditions
  - Investigate number of particles or aggregates that are ejected out of the system
  - Investigate equilibrium if a much larger particle, representative of a black hole, is centered and has initial condition of $v=0$.

Project Timeline

Dates	Activities
Week 1: Oct 23	Do basic research, derive equations & design code
Week 2: Oct 30	Implement code
Week 3: Nov 6	Test code
Week 4: Nov 13	Run numerical experiments, begin presentation & report
Week 5: Nov 20	Analyze data, continue work on presentation & report
Week 6: Nov 27	Polish presentation and work on final draft of report
Dec 1	Give Presentation!
Dec 4	Submit Final Report!

References

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


APPLAUSE !!

Toomre’s Model of Galactic Merger and Effects on Satellite Galaxy

PHYS 210 Term Project Proposal

Chenruo Qi

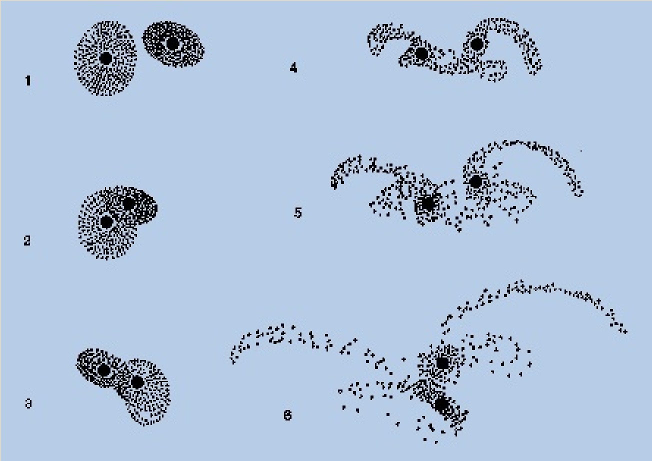
October 21, 2009

Overview

- Galaxy mergers occur when two or more galaxies collide.
- The word 'collide' is misleading. The stars do not actually collide with each other due the vast "empty" space between them.
- Strong gravitational interactions significantly change the morphology(shape) and kinematics of each galaxy.
  - When one galaxy is much larger than the other, the larger often "eats" the smaller.
    - Might be what happened to M32, a satellite galaxy of Andromeda(M31).
  - If two spiral galaxies, with about the same mass, collide at appropriate angles and speeds, they are likely to become an elliptical galaxy.
    - The Milky Way and Andromeda will collide in about 3 billion years, and the outcome is probably an elliptical galaxy.

Overview (Continued)

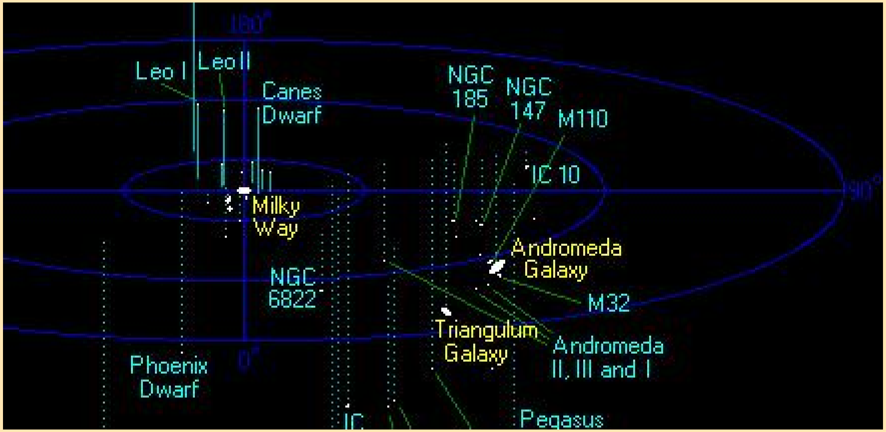
- Alar Toomre did the first galaxy merger simulation in 1970s with his brother.
  - A massive particle represents the bulge of the galaxy.
  - Small particles represent stars orbiting around the bulge.
  - Ignore ISM(gas and dust) and dark matter.



Simulations of the Antennae Galaxies

Project Goals

- To write a Matlab/Java code which simulates the collision of two galaxies. One of them has a satellite galaxy.
- To test the implementation with different test scenarios. (e.g. without the satellite, no angles involved, etc.)
- To investigate how will the collision affect the satellite galaxy.



Formulations

- For each particle, its motion is determined by

$$m_j \vec{a_j} = \sum_{k=1}^{N} \frac{G m_j M_k}{|\vec{r_k} - \vec{r_j}|^2} \hat{r} \tag{1}$$

where $\hat{r}$ is the unit vector pointing towards the bulge.

$$\hat{r} = \frac{\vec{r_k} - \vec{r_j}}{|\vec{r_k} - \vec{r_j}|}$$

- Kepler's Law - for elliptical orbits

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

- Centripetal force - for circular orbits

$$\frac{m\vec{v}^2}{r} = \frac{GMm}{r^2} \hat{r} \tag{3}$$

where $\hat{r}$ points toward the center of the bulge.

Numerical Approach

- The bulge of each galaxy will be assigned a mass and a radius.
- All stars are in circular orbits with the same mass.
- The gravitational forces between stars are neglected. So The force acting on each star is

$$m_j \vec{a_j} = \sum_{k=1}^{3} \frac{m_j M_k}{|\vec{r_k} - \vec{r_j}|^2} \hat{r} \tag{4}$$

- The time complexity is O(N). (i.e. if I have two bulges, then I need to do two computations for each particle. So for n particles, only 2n computations need to be done)
- Assume there are no collisions between stars. When bulges collide, they simply pass through each other.

Numerical Approach (Continued)

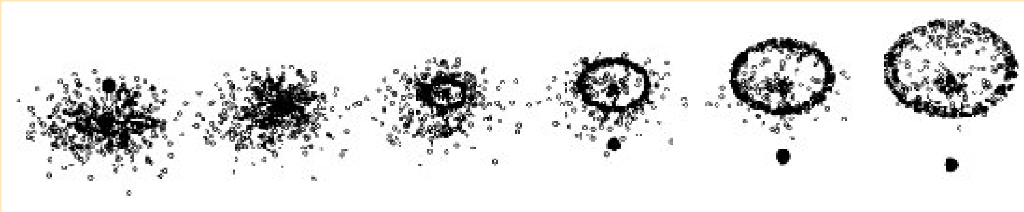
- The satellite is in a stable orbit.
- No interchange of mass between the galaxy and its satellite before the collision.
- The stars of interest will be marked in distinct colcour, so it is easy to see what is going on.

Visualization and Plotting Tools

- Use **XVS** for generation of mpeg animations.
- Use Matlab's plotting facility for plots in my report.

Testing

- Test the collision of two galaxies without a third satellite galaxy.
- Compare the numerical results with other simulations.
- Check if then head-on collision of a low mass galaxy and a disk galaxy creates a ring.



Numerical Experiments

- Simulate galaxy collisions with various initial conditions.(e.g. velocities, angles, etc.)
- Show the effect on M32 caused by Andromeda-Milky Way collision.

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, work on presentation & report
12/01	Final Presentation!
12/01-12/03	Finish final Report
12/04	Submit report!

References

- http://scicom.ucsc.edu/SciNotes/9701/full/features/galaxy/Toomre.html
- http://www.etsu.edu/physics/bsmith/collisions/collisions.html
- http://en.wikipedia.org/wiki/Alar_Toomre
- http://bh0.phas.ubc.ca/ matt/Teaching/05Fall/PHYS410/Projects/kdv.pdf

QUESTIONS?


APPLAUSE !!

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

PHYS 210 Term Project Proposal

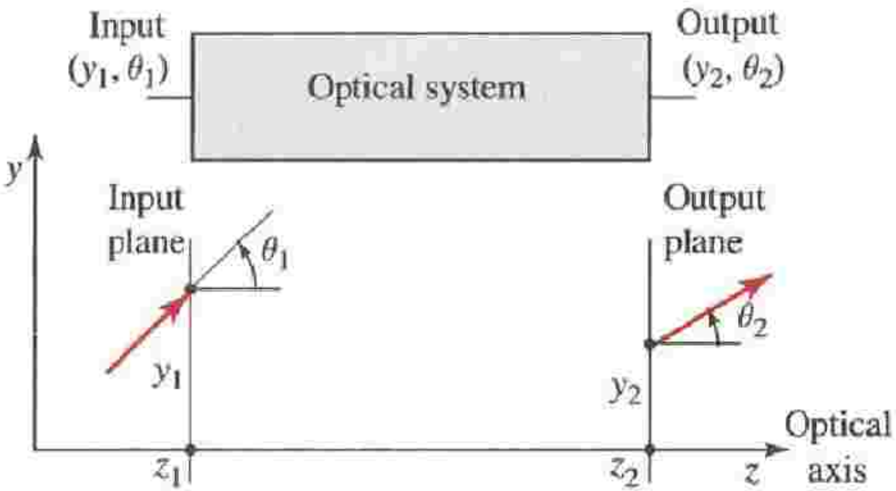
Yifei Zhou (Alfie)

October 22, 2009

- Overview
  - a method for calculating path of waves or particles through a system with regions of varying propagation velocity and reflecting surface
  - ray intersects the objects in the system, could have results like reflection, refraction and shadow

- Project Goals
  - Design a visualized system, a combination series of mirrors, prisms and lens
  - Write a script code
  - Use MATLAB to analysis ray transfer [2*2 Matrix]
  - To generate the image of the scene

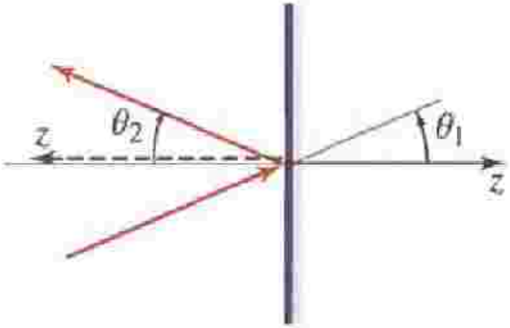
- Mathematical
  - Initial components: x_0 , y_0 , v_{X0} , v_{Y0}
  - Any given time: x_n , y_n
  - Then $x_{n+1} = x_n + v_x \Delta t$, $y_{n+1} = y_n + v_Y \Delta t$
  - Difference Equation for the Ray Position:



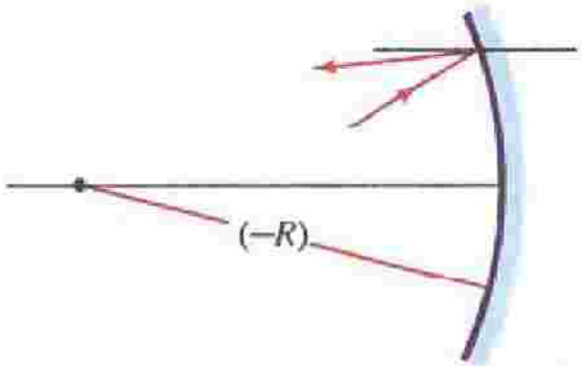
$$y_2 = Ay_1 + B\theta_1$$
$$\theta_2 = Cy_1 + D\theta_1,$$

$$\begin{bmatrix} y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} y_1 \\ \theta_1 \end{bmatrix}$$

- Mathematical
  - Matrices of Simple Optical Components



$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



Concave: $R < 0$; convex: $R > 0$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \frac{2}{R} & 1 \end{bmatrix}$$

- Numerical Approach
  - Paraxial approximation: $\sin\theta \approx \theta$
  - Boundary condition: done when light ray leaves

- Testing & Numerical Experiments
  - Define little line segment & intersects with all optical elements
    - Make Δt as small as possible: $\Delta t \Rightarrow \Delta t/2$
  - Objects will not overlap
  - Testing different light rays

- Project Timeline

Dates	Activities
10/23-10/29	Researching, Design system
10/30-11/05	Design code
11/06-11/12	Implement code, Test code
11/13-11/19	Run numerical experiments,
11/20-11/26	Analysis data, Preparing presentation
11/27-11/30	Polish presentation, Work on final draft
12/01	Give presentation
12/01-12/03	Final draft of report
12/04	Submit report

- References

  - Professor Matthew Choptuik
  - Saleh B. Teich M “Fundamental of Photonics” 2nd edition
  - http://en.wikipedia.org/wiki/Ray_tracing_(physics)
  - http://www.siggraph.org/education/materials/HyperGraph ce/rtaccel.htm


APPLAUSE !!

Simulation of Rigid Body Dynamic in Two Dimension

PHYS 210 Term project proposal

Jason Zhuang

October 22,2009

Overview

- In physics, a rigid body is an idealization of a solid body of finite size in which deformation is neglected. In other words, the shape of the object does not changed due to any external force.
- rigid body dynamics is the study of the motion of rigid bodies. Unlike particles, which move only in three degrees of freedom (translation in three directions), rigid bodies occupy space and have geometrical properties, such as a center of mass, moments of inertia, etc., that characterize motion in six degrees of freedom (translation in three directions plus rotation in three directions).

Project goals

- To write an MATLAB(octave) code to simulate the whole system.
- Test the code with different initial cases.
- Produce graphs for visual representation.
- Videos if possible.

Formulas

- Newton's Second Law states that the rate of change of the linear momentum of a particle with constant mass is equal to the sum of all external forces acting on the particle:

- $$\frac{d(mv)}{dt} = \sum_{i=1}^{n} fi$$

- Rigid body angular momentum

$$Mb_{G/O} \times \frac{d^2 R_o}{dt^2} + \frac{d(I\omega)}{dt} = \sum_{j=1}^{N} \tau_O, j$$

where I, the moment of inercia sensor, is defined using

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}$$

$$\mathbf{I} = \begin{pmatrix} \int (y^2+z^2)\,\mathrm{d}m & -\int xy\,\mathrm{d}m & -\int xz\,\mathrm{d}m \\ -\int xy\,\mathrm{d}m & \int (x^2+z^2)\,\mathrm{d}m & -\int yz\,\mathrm{d}m \\ -\int xz\,\mathrm{d}m & -\int yz\,\mathrm{d}m & \int (x^2+y^2)\,\mathrm{d}m \end{pmatrix}$$

- and w, the angular velocity is given by

$$\boldsymbol{\omega} = \omega_x \hat{\mathbf{i}} + \omega_y \hat{\mathbf{j}} + \omega_z \hat{\mathbf{k}}$$

- the angular momentum, L is then defined as

$$L = \sum_{i=1}^{N} r_i \times p_i = \sum_{i=1}^{N} m_i r_i \times v_i$$

Testing & Experiments

- Start with a simple collision between two rigid body in a open space.
- Extend the module to have multiple bodies colliding each other.
- Finish with a closed space simulation: closely examinate the motion of several rigid bodies in a closed space in 2D. (behaviors like bouncing, colliding and rotation of objects in a circle or square.)

Timeline

Dates	Activities
10/23-10/29	Do basic research, derive equations & design code
10/30-11/05	Implement code
11/06-11/12	Test code
11/13-11/19	Run experiements, and start prepare presentation and report
11/20-11/26	Analyze data, keep on working the presentation and report
11/27-11/30	Finish preparing for presentation.
12/01	Give presentation
12/01-12/03	Final draft
12/04	Report submittion

Reference.

- http://en.wikipedia.org/wiki
- http://www.youtube.com
- http://www.kwon3d.com/theory/rigid.html

Question?


APPLAUSE !!

Simulation of Electrically Charged Pendulum

Phys 210 Term Project Proposal

Benson Chang
Oct 22, 2009

- Overview
  - A simple pendulum under the effects of gravity displays a simple harmonic motion
  - What happens when another force is introduced?
  - A charged particle at the end of a pendulum arm above a plane of charged particles will present a different motion due to the force exerted on the electrostatic force on the pendulum from each of the charged particles below.

- Project Goals
  - Write a MATLAB (octave) code that aggregates the forces on the pendulum to predict/simulate the pendulum's path of motion

- Formulas Required

$$V = \frac{4}{3}\pi r^3$$

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

$$E = \frac{1}{4\pi\varepsilon_0}\frac{q}{r^2}.$$

$$F = k_e \frac{q_1 q_2}{r^2},$$

$$k_e = \frac{1}{4\pi\varepsilon_0} = \frac{\mu_0 {c_0}^2}{4\pi} = \frac{{c_0}^2}{10^7}\frac{H}{m} =$$
$$= 8.987\,551\,787\,368\,176\,4 \times 10^9\ \mathrm{N \cdot m^2 \cdot C^{-2}}.$$

$$\mathbf{F} = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2(\mathbf{r}_1 - \mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|^3} = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r^2}\hat{\mathbf{r}}_{21},$$

- Simulation Approach
  - The equations used for this are simple ones covered in first year physics.
  - Locations of the pendulum and all the charges on the plane will be determined in 3 dimensional Cartesian Coordinate system.
  - The vector between two points will be key for the computations.

- Simplifications
  - Magnetic force is disregarded.
  - The pendulum will be forced to act as a charge on the surface of a sphere.

- Project Timeline

Dates	Activities
10/23 – 10/29	Finish 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 on presentation/report
11/27 – 11/30	Finish presentation and start finalizing report
12/01	Give presentation
12/01 – 12/03	Finish report
12/04	Hand in report

- References

  - Matthew Choptuik, Phys 210, University of British Columbia, 2009
  - Fundamentals of Physics, Halliday/Resnick/Walker, John Wiley & Sons, 2005
  - Various Wiki pages ie. http://en.wikipedia.org/wiki/Coulomb_law


APPLAUSE !!

Finite Difference Solution of the Time Dependent Schrödinger Equation

PHYS 210 Term Project Proposal

Ziad Al

October 22nd, 2009

Erwin Schrödinger

Overview

- The Schrödinger Equation is the law of quantum mechanics.
- It is as important to quantum mechanics as are Newton's laws important to classical mechanics.
- It predicts the future behaviour of dynamic systems.
- The solution to the equation is the wave function $\psi\ (x,t)$

Project Goals

- To write a MATLAB (Octave) code which solves the Schrödinger Equation numerically using finite difference techniques.
- Develop methods of testing and implementing the code.
- Compare obtained solutions with known ones.
- Test a variety of initial conditions and observe their effects.

The Schrödinger Equation

$$i\hbar \frac{\partial}{\partial t}\Psi(x,t) = \frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\Psi(x,t) + V(x)\Psi(x,t) \qquad (1)$$

- where $\hbar = 2m = 1$
- The initial boundary conditions

$0 \le x \le 1$ and $0 \le t \le t_{max}$ (2)

with initial conditions

$\psi(0,t) = \psi(1,t) = 0$ (3)

Numerical Approach

- The equation will be solved using the Crank-Nicolson Finite Difference method.

$$i\frac{\Psi_j^{n+1}-\Psi_j^n}{\Delta t}=-\frac{1}{2}\left(\frac{\Psi_{j+1}^{n+1}-2\Psi_j^{n+1}+\Psi_{j-1}^{n+1}}{\Delta x^2}+\frac{\Psi_{j+1}^{n}-2\Psi_j^{n}+\Psi_{j-1}^{n}}{\Delta x^2}\right)-V_j^{n+1}\frac{1}{2}\left(\Psi_j^{n+1}+\Psi_j^n\right) \quad (4)$$

- Equation 4 can be rewritten in the form of

$$c_j^+\Psi_{j+1}^{n+1}+c_j^0\Psi_j^{n+1}+c_j^-\Psi_{j-1}^{n+1}=S_j \qquad j=2,3,\ldots,nx-1 \qquad (5)$$

Numerical Approach (continues)

- Where

$$\Psi_j^n \equiv \Psi(j\Delta x, n\Delta t) \qquad (6)$$

$$V_j \equiv v(j\Delta x) \qquad (7)$$

$$\Psi_1^{n+1} = \Psi_{nx}^{n+1} = 0 \qquad (8)$$

Precise formula for $\mathrm{c}_\mathrm{j}^+, \mathrm{c}_\mathrm{j}^0, \mathrm{c}_\mathrm{j}^-, \mathrm{S}_\mathrm{j}$ must be determined.

The conservation law for ψ (x,t) is

$$I \equiv \int_0^1 \Psi(x,t)\Psi^*(x,t)dx = \text{constant} \qquad (9)$$

- Numerical Testing
  - I plan on placing a Gaussian wave-packet between boundaries of infinite energy at x=0 and x=1 and investigate how it behaves.
  - Introduce a barrier to see the effects of quantum tunnelling.
  - I plan on plotting the conservation of probability to confirm that it remains constant

- Plotting Tools
  - I will use MATLAB plotting facilities or I will try to generate animated graphs via maple.
  - Potentially use *xvs* to generate mpeg animations.

- Project Timeline

DATES	ACTIVITIES
Oct. 23 Week1	Do basic research, derive equations & design code
Oct. 30^{th} Week2	Implement code
Nov. 6^{th} Week3	Test code
Nov. 13^{th} Week4	Run numerical experiments, begin presentation & report
Nov. 20^{th} Week5	Analyze data, continue work on presentation & report
Nov. 27^{th} Week6	Polish presentation and work on final draft of report
Dec. 1^{st}	Give Presentation
Dec.1^{st} – Dec.3^{rd}	Finish final draft of report
Dec.04^{th} Week7	Submit Report

References

- Randall D. Knight, *Physics for Scientists and Engineers: a Strategically Approach*, Pearson (2008)
- http://www.missioncollege.org/depts/physics/P4poe/P4D/Schrodinger.htm
- http://hyperphysics.phy-astr.gsu.edu/HBASE/quantum/schr.html#c1
- http://en.wikipedia.org/wiki/Schr%C3%B6dinger_equation
- http://osulibrary.oregonstate.edu/specialcollections/coll/nonspcoll/catalogue/portrait-schrodinger-600w.jpg
- http://laplace.physics.ubc.ca/People/matt/Teaching/05Fall/PHYS410/Projects/schrodinger.pdf
- Matthew W. Choptuik (2009)


APPLAUSE !!

N-Body Simulation of Gravitational Interactions

PHYS 210 Term Project Proposal

Matthew Chin
October 22, 2009

- Overview
  - Basically, the N-body problem involves predicting the motion of N particles subject to only gravitational interactions between particles.
  - So given any initial velocities and positions, the velocities and positions at any time t can be determined.
- Project Goals
  - Implement MATLAB (octave) code to simulate the N-body problem and use visualization software to display the results.
  - Simulate a variety of initial conditions and number of particles.
  - Test the results by analyzing energy and momentum conservation.

- Mathematical Formulation (Equations of Motion)
  - The main equation of motion is Newton's second law,

$$\vec{F}_i = m_i(\frac{d^2\vec{x}}{dt^2})_i$$

    where i=1,2,…,N

    where m_i is the mass of the i^{th} particle,

    F_i is the force on the i^{th} particle,

    and x is the position of the i^{th} particle.
  - With the force being the 2-dimensional vector addition of the gravitational attraction by all the other particles,

$$m_i(\frac{d^2\vec{x}}{dt^2})_i = G\sum_{i\neq j}\frac{m_i m_j(\vec{x}_j - \vec{x}_i)}{|\vec{x}_j - \vec{x}_i|^3}$$

    this includes the direction vector of the force.
  - The system will have initial position and velocity specified by x_0 and v_0.

- Numerical Approach
  - The simulation will be run using finite difference approximations, which approximates the derivative of a function by

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

    with Δt being sufficiently small

    and with v_0 being the velocity value for the previous time step.
  - To reduce error when accelerations are high, Δt will be a function dependent on the minimum radius between any two particles.
  - To apply this to the simulation rearrange for the next time step,

$$f(\vec{v}_0 + \Delta t) = \Delta t f'(\vec{v}_0) + f(\vec{v}_0)$$

  - v_0 can be replaced by x_0 in order to solve for position as well.
  - The new velocities and positions will be used to calculate force for the next time step and so on.
  - Some sort of softening factor will also have to be added in order to limit force when particles are within a certain range to decrease error and computational time.

- Testing
  - Check to make sure there is not a significant change in energy and momentum of the system since these are conserved quantities.

- Numerical Experiments
  - Investigating a variety of initial positions and velocities, which would result in varying the energy of the system and its constituents. These different configurations will be analyzed in addition to any other phenomena that are observed during testing.
  - Overall I'm just going to analyze the results with different initial values and find if there's anything interesting to investigate further.

- Project Timeline

Dates	Activities
Oct. 23-29	Do basic research, derive equations & design code
Oct. 30-Nov. 05	Implement code
Nov. 06-12	Test code
Nov. 13-19	Run numerical experiments, begin presentation & report
Nov. 20-26	Analyze data, continue work on presentation & report
Nov. 27-30	Polish presentation and work on final draft of report
Dec. 1	Give presentation!
Dec. 1-3	Finish up final draft of report
Dec. 4	Submit report!

- References

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


APPLAUSE !!

Simulation of the Toomre model of galaxy collisions

PHYS 210 Term Project Proposal

Duncan Provan

October 21, 2009

Overview

- Toomre equations named after Alar and Juri Toomre – worked on galaxy simulations in the 1970’s
- Equations for predicting the result of two galaxies merging
- Will find the end result when all has settled into equilibrium

Project Goals

- Write an octave code that will solve the system of equations
- Experiment with various initial conditions (velocity and position) to produce various solutions

Mathematical Formulation

- Newton's Law of Gravitation:
- F = G*m1*m2/r^2
- A relationship between period, radius and mass:
- (P/(2*pi))^2 = a^3/(G*(M+m))

Assumptions

- Mass of each star = 0
- Orbits are always circular

Numerical Approach

- I will use a finite difference approximation to solve
- Initial conditions for velocity and position will be specified
- My range for values of N will be set somewhere between 100 and 500

Visualization

- Create an mpeg file using MATLAB software

Numerical Experiment

- Once the code is working, I will be experimenting with various initial conditions and will be able to produce all kinds of experiments/results
- All experiments will be collision-less

Project Timeline

Dates	**Activities**
10/22 – 10/29	Basic research, design code
10/30 – 11/5	Implement code
11/6 – 11/12	Test code
11/13 – 11/19	Run experiments, begin presentation/report
11/20 – 11/26	Analyze data, continue presentation/report
11/27 – 11/30	Polish presentation and start final draft on report
12/01	Give presentation
12/01 – 12/03	Finish final draft of report
12/04	Submit report

References

- www.wikipedia.org
- "A Gentle Introduction to Numerical Simulations of Galaxy Mergers", Christopher Q. Trinh (2006)
- Eric Walker


APPLAUSE !!

N-Body Simulation in 3 Dimensions

Phys 210 Term Project Proposal

William Kellett

October 22, 2009

OVERVIEW & PROJECT GOALS

- Gravity works upon every object in a system.
- A system of N particles with bodies, with random velocities and masses will interact in a specific way due to gravity
- To determine how they shall move, an approximation will need to be used

- To write a Matlab (Octave) code which will accurately approximate the movement of N-bodies due to gravity exerted upon each other
- To test the code for correctness using with known solutions such as two body motion and three body motion.
- To investigate a variety of initial conditions.

MATHEMATICAL FORMULATION

- The equation for gravity is

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

- However this can be spilt up into it's components such that

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

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

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

- Where $r = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$
- and $\Delta x = x_i - x_o \quad \Delta y = y_i - y_o \quad \Delta z = z_i - z_o$

Numerical Approach

- Using this we can determine the Distance using the equations of motion (shown just for the x-direction)

$$x = x_o + vt + \frac{a}{2}t^2$$

$$v = v_o + at$$

$$a = G\left(\sum_{i}^{n} \frac{m_i \Delta x}{r^3}\right)$$

- Then using the Runge-Kutta Method I shall approximate the changes

VISUALIZATION

- I shall use Matlab/Octaves built in visualization programming to generate my graphs and any visualizations necessary.

TESTING

- Using my program I can test using known solutions for 2 body systems and 3 body systems.

PROJECT TIMELINE

Dates	Activities
10/23 -10/29	Do Basic Research, Derive Equation & Design Code
10/30-11/05	Implement Code and Test Code
11/06-11/12	Test Code, Begin Numerical Experiments
11/13-11/19	Run Experiments, Begin Experiments & Report
11/20-11/26	Analyze Data, Continue Work on Presentation & Report
11/27-11/30	Polish Presentation & Work On Final Draft
12/01	Give Presentation
12/01-12/03	Finish Final Draft
12/04	Submit Report

- http://en.wikipedia.org/wiki/Three-body_problem
- http://en.wikipedia.org/wiki/Two-body_problem
- http://en.wikipedia.org/wiki/N-body_problem
- http://faculty.ifmo.ru/butikov/ManyBody.pdf

REFERENCES


APPLAUSE !!

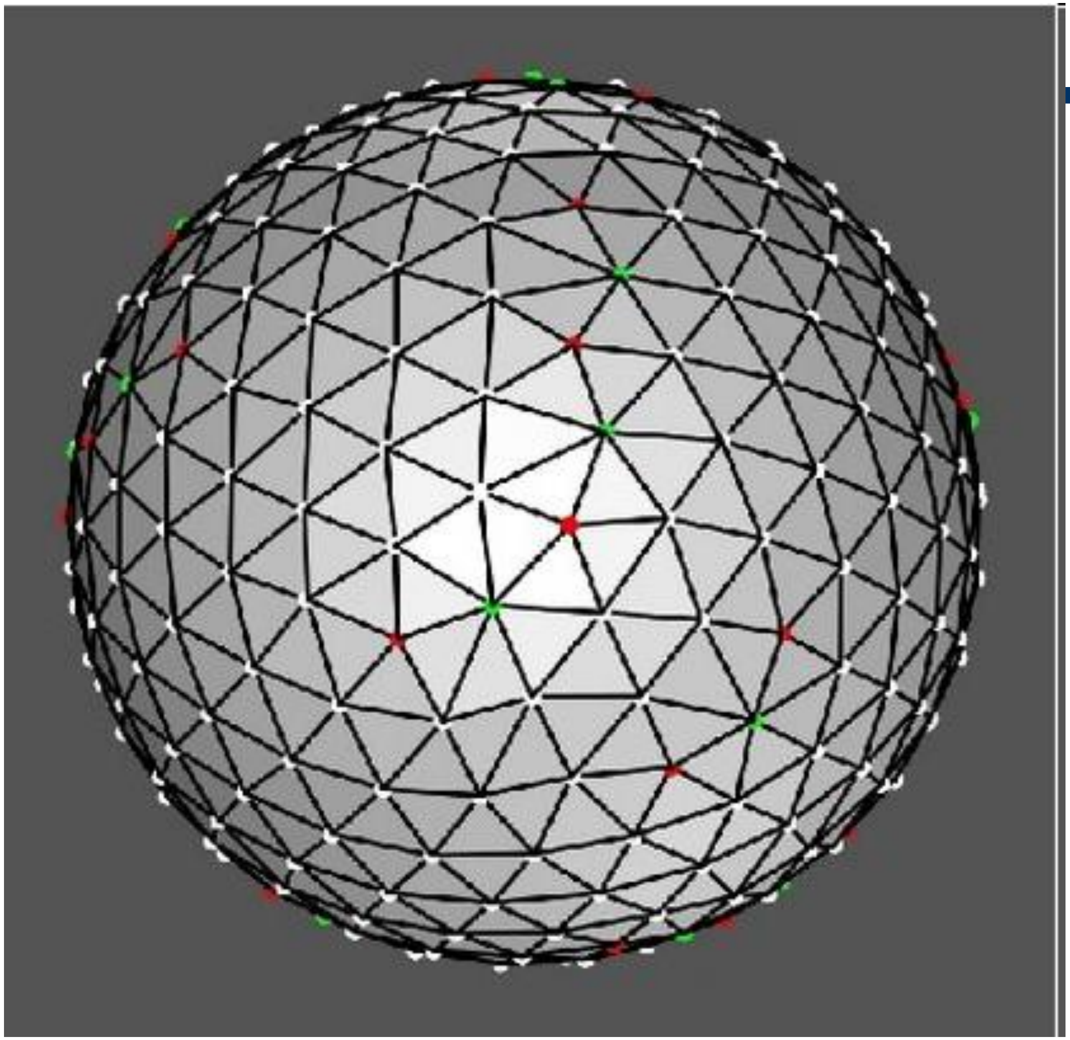
Distributing N Charges on the Surface of a Sphere

By: Henry T

Overview

- Thomson problem
  - Given N charges confined to the surface of a sphere, what is the arrangement of charges which minimizes the total electrostatic energy?
  - Originally part of J.J. Thomson's Plum-Pudding model of the atom
- Applications
  - Protein arrangement in the shells of spherical viruses
  - States of dusty plasma
  - Colloidal crystals

N = 1000



Project Goals

- To write a MATLAB code that will create a 3D visualization of how N charges distribute themselves when bounded to the surface of a sphere
- To check the accuracy of the code by comparing the results to known values
- To possibly investigate various initial conditions
  - Some N have more than one possible equilibrium configuration

Mathematical Formulation

- Coulomb's Force

For the electrostatic force on charge q from charges q_1 to q_N

$$F_E(q, q_i, r, r_i) = kq \sum_{i=1}^{N} \frac{q_i (r - r_i)}{|r - r_i|^3}$$

- Where r is the position of charge q, r_i is the position of q_i, and k is Coulomb's constant ($k = 8.987551787368\ldots x10^9$ $N \cdot m^2 \cdot C^{-2}$)

- Electric Potential Energy

$$U_E(q, q_i, r, r_i) = kq \sum_{i=1}^{N} \frac{q_i}{|r - r_i|}$$

Numerical Approach

- Simplifications
  - All charges = +1
  - Radius of sphere = 1 unit (with centre as origin of coordinates)
  - Physical constants ignored (Coulomb's constant in this case)

$$F_E = \sum_{i=1}^{N} \frac{(r - r_i)}{|r - r_i|^3} = \sum_{i=1}^{N} \frac{1}{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$$

$$U_E = \sum_{i=1}^{N} \frac{1}{\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}}$$

Numerical Approach

- +1 charges will begin at random positions on sphere
- Each charge will move in the direction of the net force it experiences while constrained to the surface of the sphere.
  - Iteration of moving tangent to the sphere and then projecting the point back on to sphere
  - A set of linear equations will have to be solved

Testing and Experimentation

- Will test for known easy-to-identify configurations such as octahedrons
- Will compare potential results to known values
- Will test for initial values of various N that give another equilibrium configuration

Charges	Potential	Charges	Potential	Charges	Potent
38	593.0385036	71	2190.649906	104	4822.87
39	626.389009	72	2255.001191	105	4919.00
40	660.6752788	73	2320.633884	106	5015.98
41	695.9167443	74	2387.072982	107	5113.95
42	732.0781075	75	2454.369689	108	5212.81
43	769.1908465	76	2522.674872	109	5312.73
44	807.1742631	77	2591.850152	110	5413.54
45	846.1884011	78	2662.046475	111	5515.29
46	886.1671136	79	2733.248358	112	5618.04
47	927.0592707	80	2805.355876	113	5721.82
48	968.7134553	81	2878.52283	114	5826.521
49	1011.557183	82	2952.569675	115	5932.181

Project Timeline

Oct 26 – 30	Research and design code
Nov 1 – 6	Write code
Nov 8 – 13	Write code and debug
Nov 15 – 20	Run experiments & begin report
Nov 22 – 27	Run experiments, analyze data, continue report, and prepare presentation
Nov 28 – 30	Prepare presentation
Dec 1	Presentation
Dec 2 – 4	Finish Report (and prepare for being slaughtered in finals)

References

No MLA/APA/etc. format here

- Wikipedia
- Google
- http://tracer.lcc.uma.es/problems/thomson/thomson.html
- http://adil.m.mughal.googlepages.com/thethomsonproblem2
- http://www.science.uwaterloo.ca/WWSEF/05Awards/05forsyth.pdf
- http://www.iop.org/EJ/article/0305-4470/24/23/008/ja912308.pdf?request-id=3c046918-e58c-449c-8be9-4e073dff63c6
- http://trond.hjorteland.com/thesis/node26.html


APPLAUSE !!