Finals Projects Symposium

ASU PHY494 Computational Methods in Physics

(Spring 2019)

Arizona State University, Department of Physics
Room PSH 356

Thursday May 2, 2019, 11:15am–2:00pm

1 Posters

#	title	authors
1	Path Integral Monte Carlo	Curtis Peterson, Sabrina Suhartono
2	Simulation of NaCl Crystal	Alan Gardner, Samantha Lake, Logan Leinbach
3	Benzene simulation	Nikolaus Awtrey, Justin Gens, Rick Sexton
4	Simulating J.J. Thompson's Cathode Ray Tube Experiment	Alex Gardeck, Sydney Olson
5	Agent-Based Modeling of Self-Driving Cars	David Lewis, Nathanael Mains, Albert Wang
6	Jupiter's Trojan Asteroids	J. Foster Morgan, Martin Flores, Avi Brahmbhatt
7	Examining Solar System Dynamics	Shane Bechtel, Kendric Knorr, Adrian Hernandez

2 Procedures

Mount your poster on the provided poster board and attach the number assigned to you (see table above). At the end of the symposium, return all mounting materials and take your poster with you.

Note that the instructor might take pictures of the posters as an additional record for grading purposes.

2.1 Q&A

- Each member of the team will be asked to engage in an individual Q&A with the instructor in front of the poster of about 6 minute duration. The Q&A will be graded and be part of the final grade.
- The TA will also ask questions and report their assessment to the instructor. However, they will only report positive evaluations, i.e., you cannot make your grade worse by talking to a TA.
- If you are not engaged in a Q&A then you are free (and encouraged) to look at other posters.

2.2 Poster prize

Participants will be able to vote for the best poster. Ballots will be provided: rank order the top three posters and drop your ballot into the collection container. The winning team will be awarded a prize (which will *not* influence the grade).

3 Abstracts

Abstracts[1] are listed in the order of appearance in the program.

[1] The 200-word limit to the abstract text is indicated by graying out any text beyond the limit. For a real conference, your abstract would have been truncated or rejected by the submission system.

#1 • Path Integral Monte Carlo • Curtis Peterson, Sabrina Suhartono

`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-path-integral-monte-carlo`

Ground state observables for N_2, O_2, and N-O diatomic molecules in their $X^1\Sigma_g^+$ (homonuclear) and $A^2\Sigma^+$ (heteronuclear) Born-Oppenheimer configurations have been calculated using Monte Carlo path integral simulations. The Born-Oppenheimer potentials have been approximated by Morse potentials. The chosen path integral Monte Carlo method consists of choosing an initial (potentially unphysical) bound state trajectory, then perturbing the trajectory and accepting or rejecting the perturbed trajectory based upon a Metropolis criterion that evaluates "total energy" as the integral of the energy in time over the trajectory (this is equivalent to the integral of the Lagrangian in time after an analytic continuation of the time variable to the negative imaginary axis). The probability that the system occupies a certain path is then distributed as the probability distribution of the canonical ensemble, similar to an Ising magnet. We find that our expectation values for the kinetic energy, total energy, and radial coordinate compare well with those calculated using a combination of an R-matrix propagation method for solving Schroedinger equations and the Hellman-Feynman theorem for calculating expectation values from eigenenergies. Future work will include the implementation of an algorithm that starting trajectories for quicker and more accurate calculations of the ground state wave functions.

#2 • Simulation of NaCl Crystal • Alan Gardner, Samantha Lake, Logan Leinbach

`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-salt-crystal/`

A crystal lattice of sodium chloride will be simulated using the Velocity Verlet algorithm in Python. The simulation can be run for cubic systems

of integer multiple unit cell length, with the ultimate goal to calculate the temperature fluctuation over time and to analyze the loss of energy by the simulation over time due to round-off error. The Lennard-Jones 6-12 potential and the electrostatic potential are both used to determine the force, which is then utilized for the Velocity Verlet to solve the equations of motion, giving the behavior of the atoms in the crystal lattice. The simulation is run in AKMA units with a time step of 0.05 AKMA units and is animated using VMD. Future work for the simulation is required to determine the cause of the simulation's instability in which a single atom is disrupting the entire lattice, which will allow for the temperature over time to be computed.

#3 • Benzene simulation • Nikolaus Awtrey, Justin Gens, Rick Sexton
`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-benzene`

The goal of this project was to create molecular dynamics code that would simulate the interactions of benzene molecules via the velocity Verlet algorithm. The interactions were modeled using the Morse potential and a weighted Lennard-Jones potential that functioned based on neighboring particles internal to each molecule. For multiple benzene molecules, Lennard-Jones interactions were implemented to account for inter-molecular forces. To obtain a stable molecule in simulation, dihedral potentials were added between carbon atoms. Bond parameters for the Morse potential were obtained from existing force-fields and used in the intra-molecular force calculations. The bond parameter don't give correct forces to produce a stable molecule with the current implementation. Adjustments need to be made to the parameters to determine a combination that gives a stable molecule. In the future it is possible that adding more forces and correcting the current parameters will produce a stable simulation.

#4 • Simulating J.J. Thompson's Cathode Ray Tube

Experiment • Alex Gardeck, Sydney Olson
`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-cathode-ray`

A model was created to simulate the Cathode Ray Tube Experiment that was performed by JJ Thompson in 1897. The simulation requires inputs of accelerating potential, external electric field, and external magnetic field. A runge-kutta integrator was used to determine the trajectory of an electron in the cathode ray tube by accounting for all the forces acting on the electron in each region. The trajectory was simulated four times for when the external electric and magnetic fields are both off, when they are both on, and both cases when only one of them is on. The trajectories were plotted for each simulation, and the final deflections were compared to the analytical solution obtained from classical electrodynamics. The model yielded values of defection that were very accurate compared to the theoretical values obtained from using Galileo's equations. All the percent error values between the theoretical solutions and the model's solutions were under 0.7%. Additionally, an animation was generated to display the path of the electron in the cathode ray tube. Further work would consider the relativistic effects due to the high velocity of the electron, and the effects of spin.

#5 • Agent-Based Modeling of Self-Driving Cars • David Lewis, Nathanael Mains, Albert Wang
`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-ghost`

Self-driving cars are quickly becoming the option of the future in the automobile industry. We can try to analyze how this integration will affect current traffic flows as well as predict methods that will help ease the integration process. Agent-based modeling (ABM) is a particularly useful method to answering these questions because it allows the user to create individual entities ("agents") that make decisions based upon defined rulesets and situations within a simulation. By applying this method we studied in this project the behavior of self-driving vehicles inside of

a simple road network. In this project, we have created an ABM using the Python package Mesa. A basic road network consisting of an intersection and multiple road segments was created, and individual agents were programmed with simple behaviors and rules, allowing us to look for emergent complexity. These agents were then released into the model to interact based upon these rules. The results of this were then analyzed using Jupyter notebook to plot and compare various parameters with the rules used in each agent. This project could be further expanded by growing the road network, or adding various rulesets to change how these agents move and interact.

#6 • Jupiter's Trojan Asteroids • J. Foster Morgan, Martin Flores, Avi Brahmbhatt

`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-help-i-forgot-the-integral-of-c`

At sixty-degree angles from Jupiter and in the same orbit lie two fields of asteroids, one trailing and one ahead of the planet. Using the velocity Verlet algorithm and matplotlib we created a stable time-dependent model (stable up to nearly twelve of Jupiters orbits) of the system to analyze whether our current theoretical understanding is sufficient to describe what is physically observed. In theory, without Jupiter the asteroid clusters should break free of their orbit and spiral into the sun. We find that the influence from Jupiter is nearly insignificant in keeping the asteroids in orbit. We also find that unlike previously created models, the influence that Saturn has on the system does not change the stability of the orbiting clusters.

#7 • Examining Solar System Dynamics • Shane Bechtel, Kendric Knorr, Adrian Hernandez

`https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-solar-system-dynamics`

The goal of this project is to simulate the solar system comparing the dif-

ference between including/excluding interplanetary gravitational forces, the effect of a stationary non-negligible gravity source, the removal of planetary mass objects, and comparing our simulated orbit of the planet Mercury-which does not account for general relativity-to the observed orbit using data retrieved from NASA. The velocity verlet integrator is used in order to ensure long-term stability. The primary result from each of the simulations was the rms of either the difference in angular velocity or distance in the two compared situations. We found that the rms values of angular velocity were generally largest for planets closer to the sun while the position rms was more comparable between the bodies examined. The primary result observed is the observation that while simplifications of the solar system model do create inaccuracies, they mostly present an adequate view that works for most bodies in the solar system.