New Mexico

Supercomputing Challenge

Final Report

March 19, 2015

1019

Deming High School

Team Members:

Ray Stubbs

Teacher:

Sandra Almanza

Project Mentor:

None

Proposal

As a project for the supercomputing challenge I will make a gravity modeling environment. The program will allow its user to create astronomical bodies with a specified mass and initial velocity. When the model is run all created bodies will be attracted to each other, and move toward each other at speeds determined by the objects' masses and distances from each other.

I will write the program in Java and use Newton's Law of Universal Gravitation to model the gravity between bodies. I will use Kepler's Laws to test whether my model is accurate. The purpose of my project is to simulate the effects of gravity on astronomical bodies. I hope to use this model to research what would happen to the orbit of the Earth if Mercury, the smallest planet and the one closest to the Sun, was consumed by that Sun. Would Earth still be habitable? This is the question I plan on answering.

Executive Summary

In my research I attempt to discover the changes that would occur in our Solar System if Mercury was consumed by the Sun. I predict that the orbits of all the other planets will be shifted slightly inward. This would not be good for humans, since right now the Earth exists at the perfect distance from the Sun to support our life. If the Earth's orbit is shifted too far in toward the Sun, or too far away from the Sun, then human kind and all other life on Earth, could perish.

Report

Our universe is an amazing phenomenon, the rules that govern it so perfect. One of these rules that has perhaps one of the greatest effects on the universe, and causes the most change in it is the law of gravity. This law is the law that causes the attraction of masses in the universe. It is the law that holds the planets together, fuels the stars, creates and maintains Solar Systems, and keeps galaxies from falling apart. The law of gravity is the reason that we, as humans, and everything else in our lives can exist. This law is very stable, and has kept our world alive for billions of years; however gravity is also unforgiving and must follow its own law.

Earth exists in a small region of the Solar System that can support liquid water and thus support life, called the Circumstellar Habitable Zone. According to Wikipedia the habitable zone in our Solar System ranges from about 0.725 to 3.0 AU, this gives us a pretty small tolerance.

In my experiment I study the changes that would occur in the Earth's orbit if the Sun consumed Mercury. To clarify, by consume I mean to pull it in and absorb its mass. I study these changes and attempt to answer the simple question: Would mankind survive?

Tools

I chose to conduct my investigation with GravSim, a gravitational modeling environment that I wrote in Java. Though still in development GravSim allows for the gravitational simulation of an indefinite number of astronomical bodies in 2D, vacuum space. The simulation considers the mass of each object, and their proximity with each other when calculating their motions. The environment supports adjustable step sizes, this allows the user to trade simulation speed for greater accuracy, so in theory, with a small enough step size simulations can produce near-perfect results, however this could take an impractical amount of time, so I'll use less accurate, faster simulation settings for my investigation. Because this tool is based upon a 2D universe and the step sizes are kept fairly large, the predictions are purely theoretical and may not be extremely accurate.

I verified the validity of my model by setting up a simulation of the Earth's orbit around the Sun. The simulation satisfied all of Kepler's Laws; the orbit of the Earth is an ellipse, with the Sun at one of the foci, the arc area formed by the Earth's sweeping across space is equal during an equal period of time independent of the phase of the orbit, and the ratio between the square of the orbit period over the cube of the distance between the Earth and Sun satisfy the T^2/R^3 = ~1 of Kepler's Third Law. Also the Earth maintains a semi-constant orbital distance that matches that of the real Earth, and the orbital period is ~365 days, which is the length of one real year.

Method

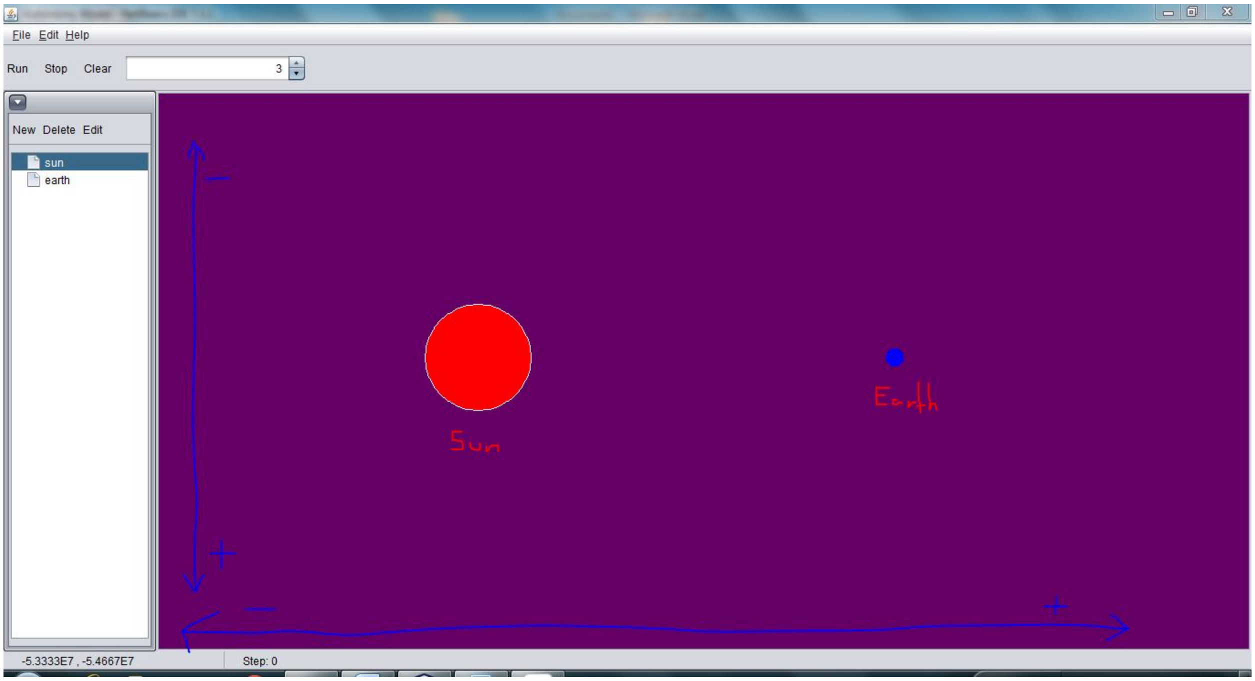
My experiment is rather simple, first I setup a simulation of the Earth's orbit under normal conditions, with the Sun and Earth at their proper masses. I setup GravSim to log the X coordinate of the Earth each time its Y coordinate changes polarity, that is, it logs the Earth's distance from the Sun at aphelion (max orbit distance) and perihelion (min orbit distance). Next I setup a similar simulation; only in this one the Sun has the combined mass of itself and Mercury. After running each simulation for a time, I analyze the log data and see if the orbit of the second simulation was smaller than that of the first and if so then by how much. If the smallest orbit distance (perihelion) is still higher than the minimum habitable orbit distance (0.7E6 km from the Sun), then mankind will survive this devastating disaster, otherwise we are doomed.

Initial Setup for Simulation 1 (Normal Masses)					
	Mass (Kg)	X (Km)	Y (Km)	Initial Velocity (Km/s)	Angle (π*rad)
Sun	1.989E30	0	0	0	0
Earth	5.9722E24	152E6	0	29.29	0.5

Note: *I set the initial X coordinate of Earth at the maximum distance from the Sun that its orbit ever reaches, and its initial velocity to the minimum velocity that the Earth ever travels at. This is because the Earth travels slowest when it is farthest from the Sun since gravity generates less force from a distance.*

Initial Setup for Simulation 2 (Sun Mass = Sun + Mercury)					
	Mass (Kg)	X (Km)	Y (Km)	Initial Velocity (Km/s)	Angle (π*rad)
Sun	1.989003301E30	0	0	0	0

Earth	5.9722E24	152E6	0	29.29	0.5



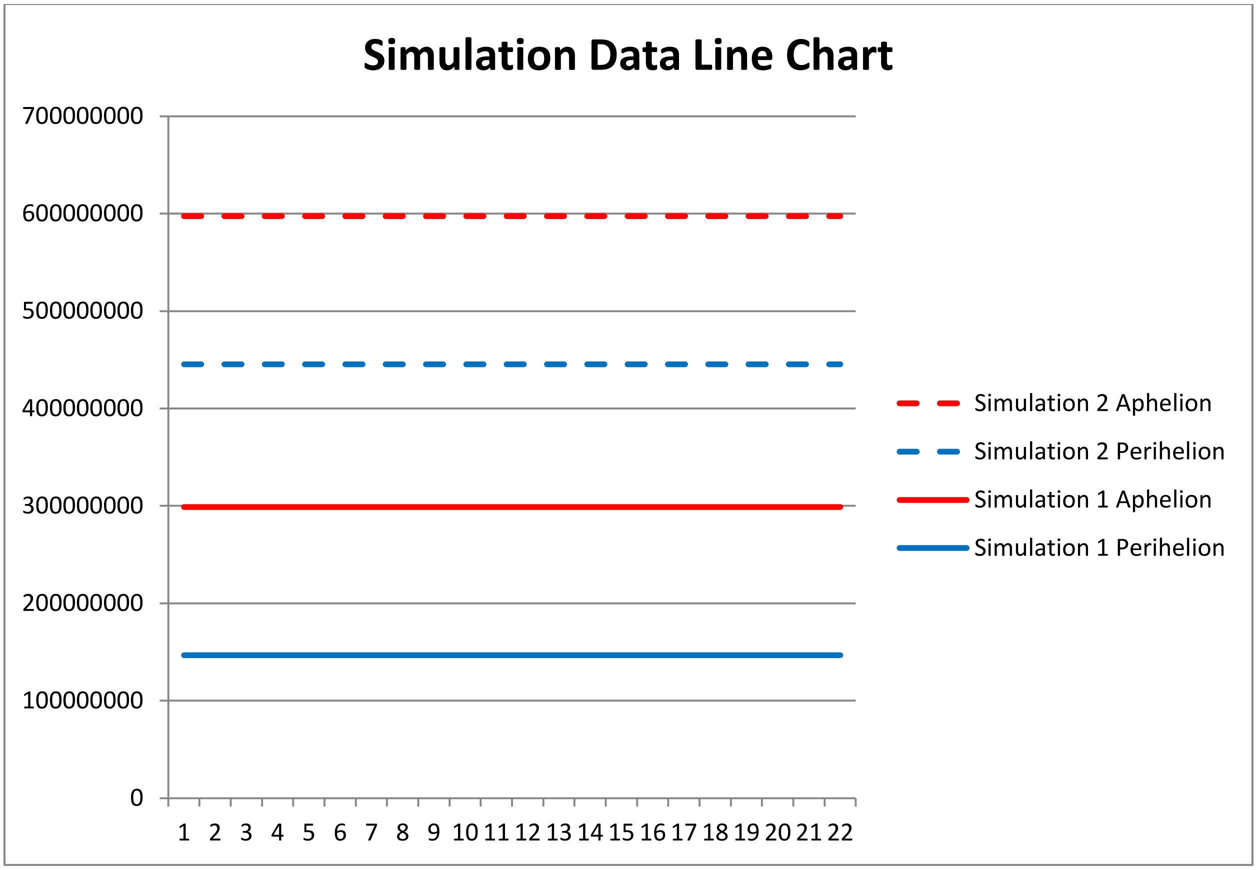
Initial setup for the simulations, note that the radius' are scaled to make them visible; at the current zoom state neither the Sun nor the Earth would be visible with their appropriate radius'.

Simulation Data

This is the log data generated by GravSim.

Simulation Data			
Simulation 1		Simulation 2	
Perihelion	Aphelion	Perihelion	Aphelion
146729357.860304	151999998.669940	146729310.001625	151999998.637151
146729356.762856	151999997.320881	146729308.898725	151999997.248172
146729355.646007	151999995.952490	146729307.768735	151999995.832655
146729354.509389	151999994.564415	146729306.611206	151999994.390202
146729353.352631	151999993.156304	146729305.425705	151999992.920385
146729352.175348	151999991.727818	146729304.211795	151999991.422805
146729350.977174	151999992.556574	146729302.969041	151999992.584345
146729349.757728	151999991.419873	146729301.696987	151999991.422080
146729348.516631	151999990.263776	146729300.395183	151999990.232836
146729347.253497	151999989.087929	146729299.063171	151999989.016197
146729345.967944	151999987.891963	146729297.700518	151999987.771753
146729344.659574	151999986.675516	146729296.306760	151999986.499066
146729343.328012	151999985.438228	146729294.881436	151999984.522735
146729334.292200	151999983.090458	146729293.424094	151999983.596637
146729332.540717	151999982.158298	146729291.934260	151999982.643052
146729330.762648	151999981.206228	146729290.411478	151999981.661552
146729328.957602	151999980.233879	146729288.855276	151999980.651697
146729327.125166	151999979.240879	146729287.265187	151999979.613058
146729325.264955	151999978.226847	146729285.640739	151999974.581543
146729323.376550	151999972.474880	146729272.810256	151999973.900614
146729321.459531	151999971.777825	146729270.667419	151999973.191612
146729319.513490	151999971.060628	146729268.486600	151999972.454096

Note: *The perihelion and aphelion lose a Km of distance with each orbit, this is a consequence of using large step sizes, the results are less accurate. Luckily the inaccuracy in these simulations follow somewhat of a pattern, losing one Km on each orbit, so to patch up the data I add 22 Km (one for each sample) to each of the data sums before calculating the average.*



Discussion

The conclusion of the experiment is based on the average values of the perihelion and aphelion of **Simulation 1** and **Simulation 2**. In the calculations I add 22 to the sums of the data values to make up for the inaccuracy mentioned in the above note.

Average Values			
Simulation	**Type**	**Calculation**	**Value**
1	Perihelion	$(22 + \sum_{0}^{22} perihelion)/22$	146729342.09363
	Aphelion	$(22 + \sum_{0}^{22} aphelion)/22$	151999987.0998
2	Perihelion	$(22 + \sum_{0}^{22} perihelion)/22$	146729293.7921
	Aphelion	$(22 + \sum_{0}^{22} aphelion)/22$	151999987.12701

As can be seen in the results, the aphelions for the two simulations are almost exactly the same, and the perihelions are also pretty close. And the perihelion of **Simulation 2** does not even come close to dropping below the 700,000 habitable zone limit.

Conclusion

In conclusion, we have no reason to fear being pulled out of our habitable zone, if the Sun consumes Mercury and absorbs its mass, the orbit of our Earth would hardly change at all; at most the average temperature of the Earth might rise a little due to the small change (about 48.30153 Km) in the orbit size. Of course this simulation only considers gravity though, so there is always the chance that we'll be exterminated by the heat blast that'll occur when Mercury is consumed, but that consideration was not part of my experiment.

Personal Statement

My most significant accomplishment in this project was how much I learned. I've never worked much with Java graphics so this project gave me a chance to experiment with them, and I had to figure out how to get everything in the program to work with the graphics, which proved to be quite a task. I also began working in a new programming language, Go, to speed up the model; however there is no Go code in the final product due to the language's lack of GUI support. GravSim, the modeling environment, itself is what I am proud of; not so much the investigation since I think that could have gone *much* better. I'm proud of the program because it worked exactly as I'd foreseen which does not always happen with my programs.

Acknowledgements

I'd like to acknowledge my coach Ms. Almanza for all the time and effort she put into this program, the NetBeans development team for developing a great Java development environment which I used to develop GravSim, Wikipedia for providing me with great research leads and numbers, and for the Supercomputing Challenge Staff and volunteers for all the hard work they put into the program and for making the program itself possible. Thank you all very much.