

Project Report: Project A

Benjamin Cox

Due 2018-03-29

Contents

1	Abstract	2
2	Overview	3
2.1	Objectives of Project	3
2.2	Requisites For A Simulation	3
3	Results and Discussion	3
A	Observables Data	4

1 Abstract

In this report we will discuss the design and implementation of an N-body simulation program. We will discuss details of the code and where choices were made, as well as justifications for those choices.

We will then dissect the results of running the simulation on the solar system for an extended time period, as well as comparing my results to those of another group.

We will conclude with an analysis of what went right and what went so horribly wrong in the execution and implementation of the simulator.

2 Overview

2.1 Objectives of Project

I have attempted the 'Astronomical N-Body Simulation' project, in which the Solar System is simulated according to Newtons Laws of Gravitation.

The objective of the project was to simulate the solar system by reading in the initial states of the celestial bodies and simulating their interactions with each other, consequently calculating their positions in the future. We were to create a file containing observables (apoapsis, periapsis, and period) and compare these to known values to determine the accuracy of the simulation.

2.2 Requisites For A Simulation

To create the N-body simulator I first needed to implement functions that could handle arbitrarily many particles. These are contained within the `GravUtils.py` file. Arguably the most important function in the file is `step_time`, because it contains the entire algorithm for the main body of the simulation. However it is neatened up by doing much of the calculations via calls to other functions.

To calculate all of the forces involved in an N-body system I used a nested loop. The first loop selected a body for which the forces were to be calculated and the second loop calculated the force of all other bodies upon that body. This loop was responsible for the majority of the runtime due to it having complexity n^2 .

Another coding choice was to contain the previous force applied on the particle within the particle class itself. This saved a lot of trouble that could have arisen with the Velocity Verlet algorithm.

Some deviations occurred from the design document. The most notable of those changes was that in the end ellipse fitting was not used for the orbits. This was due to time constraints.

To get around cumulative errors that occur when using very large numbers we rebased our calculations from metres and second to astronomical units and days.

3 Results and Discussion

A Observables Data

Period is measured in days, apoapsis and periapsis in AU.

Body	Calc. Period	Period (days)	Calc. Apoapsis	Apoapsis	Calc. Periapsis	Periapsis	Calc. Ecc	Ecc
Sol	-	-	-	-	-	-	-	-
Mercury	89	88	0.468	0.467	0.310	0.308	0.203	0.206
Venus	225	225	0.729	0.729	0.719	0.718	0.007	0.007
Earth	366	365	1.017	1.017	0.983	0.983	0.017	0.017
Moon	31	28	3.014e-3	2.710e-3	2.466e-3	2.424e-3	0.100	0.056
Mars	687	687	1.667	1.666	1.381	1.381	0.094	0.094
Jupiter	4333	4333	5.459	5.459	4.946	4.950	0.049	0.049
Saturn	10749	10759	10.078	10.124	9.007	9.041	0.056	0.057
Uranus	30672	30689	20.115	20.110	18.213	18.330	0.050	0.046
Neptune	60198	60182	30.345	30.330	29.810	29.810	0.009	0.009
Pluto	91395	90560	49.775	49.305	29.658	29.681	0.253	0.248
Halley	27322	27491	35.143	35.082	0.590	0.586	0.967	0.967