Simulation of the Solar System and Additional Experiments

By Emma Jane Allen

s2021695

Table of Contents

Background and Overview of Project

In 1543 Nicolaus Copernicus published his theory of a heliocentric model of the solar system (Brown, 2014) where planets orbit the Sun in circular motion. In the early 17th century, Johannes Kepler (Field, 1999) built upon these ideas to postulate three empirical laws of planetary motion, most notably Kepler's third law which is an empirical formula for calculating the time period of an elliptical orbit. Today, scientists use these principles to determine planetary orbits with immense precision, as demonstrated by the NASA Perseverance mission (NASA, 2020).

This project simulates the planets of the solar system in two dimensions. Data for each celestial body is read in from an excel file and an animation of the solar system is displayed in addition to the relevant total energy graphs. values for orbital period and total energy are written to an excel file. The exact output of the program depends on the program mode.

In program mode two, the total energy of Direct Euler and Beeman integration methods is graphically demonstrated. In this mode, the program takes a user input of which integration method to use to display the animation, then a total energy graph of both integration methods is displayed.

Program mode four predicts the optimal initial velocity of orbit for a satellite to get closest to Mars In this mode, the program runs the simulation (without displaying the animation) to determine the optimum initial velocity of orbit in a set range. Once found, the animation is ran displaying the satellite.

Program mode five investigates the influence of Jupiter on the orbital periods of the inner planets by updating position and velocity based solely on the Sun's influence and writing the value of the orbital period and percentage difference to a file.

Mathematical Background

The total energy, gravitational force and acceleration are calculated as presented in the course book (Britton Smith, 2023); these are not discussed further. The solar system is treated as a gravitational many-body problem, therefore numerical integration methods must be used to calculate the orbital paths. This program uses Beeman and Direct Euler methods.

Beeman:

The Beeman method is a variation of Verlet integration (Nikolic, Verlet Method, n.d.). It works by utilising a predictor formula and corrector for the velocity, the method is shown below:

$$\vec{r}(t+\Delta t) = \vec{r}(t) + \vec{v}(t) + \frac{1}{6}[4\vec{a}(t) - \vec{a}(t-\Delta t)]\Delta t^2 \qquad \text{[Equ.1]}$$

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \vec{v}(t) + \frac{1}{6}[2\vec{a}(t+\Delta t) + 5\vec{a}(t) - \vec{a}(t-\Delta t)]\Delta t^2 \qquad \text{[Equ.2]}$$

Where Δt is the small timestep, t is the current time, $\vec{r},\ \vec{v},\ \vec{a}$ are the vectors for position, velocity and acceleration respectively.

The error for velocity is $O(\Delta t^3)$ and $O(\Delta t^4)$ represents the error for position, this gives an overall error of $O(\Delta t^3)$, which provides a high degree of accuracy; an advantage over the Verlet method. However, the method is not self-starting, therefore program uses $\vec{a}(t - \Delta t)$ to be $\vec{a}(t)$ in the first iteration to start the program.

Direct Euler:

The Direct Euler algorithm is the equivalent of retaining the $O(\Delta t)$ terms of $x_{n+1} = x(t_n + \Delta t)$ and $v_{n+1} = v(t_n + \Delta t)$ in their respective Taylor expansions. The algorithm is presented below:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t)\Delta t \qquad \text{[Equ.3]}$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \vec{a}(t)\Delta t \qquad \text{[Equ.4]}$$

Where all symbols have their pre-defined meaning.

The error in one timestep is equal to $(\Delta t)^2$. Over time this error will accumulate, resulting in a global error of order Δt. Direct Euler is asymmetric as it will use information about the derivate only at the beginning of the interval when advancing the timestep, hence it has restricted accuracy. (Nikolic, Euler Method, n.d.). Due to being a first order expansion, Direct Euler is less computationally demanding in comparison to Beeman.

Both Direct Euler and Beeman have a time complexity of O(N), where N is the number of timesteps to solve the differential equation. the Direct Euler method requires one calculation of the position and velocity, while Beeman's algorithm requires three evaluations of the acceleration at each timestep. Therefore, Beeman is computationally more demanding per timestep.

Program Design and Structure

The program was written in an Object Orientated Programming (OOP) style where each celestial body is an object with its data encapsulated in various variables and methods. OOP properties such as encapsulation, modularity, inheritance and polymorphism were utilised throughout the project. Inspiration was taken from the solution code as to how to structure the program and complete the integration.

The program mode is determined by the user. This implementation means the user does not have to modify the code to run a particular experiment, making the simulation accessible to those without coding experience. The program is split into three main classes and six subclasses. Figure 1 shows the Unified Modelling Language (UML) representation of the program.

CelestialBody and CelestialSystem classes form an aggregation relationship as the CelestialSystem consists of CelestialBodies. Inheritance was used to form one subclass of CelestialSystem for each program mode. Two subclasses of CelestialBody were created for different experiment requirements. This methodology reduced code duplication and unnecessary if statements, improved maintainability, readability, and highlights program mode differences.

To determine a new year, the planet's y coordinate is compared to the Sun's y coordinate to account for the sun's movement. Time periods are calculated and stored in a data frame; if the timestep is a multiple of 2000, the data written to file. This approach aims to improve efficiency. This methodology was utilised for writing total energy to file.

A try-except clause was used to set the initial position, velocity and acceleration of a celestial bodies, capturing the zero-division encountered with the sun case. This was done to avoid if statements and improve speed. All bodies are begin aligned on the x-axis, this is unrealistic and could contribute to the discrepancies between simulation results and literature values in the experiments.

Celestial bodies are not sized based on their radius. As to the sun's mass being far greater this would lead to inner planets being too small to see. A consequence is that a satellite and its target planet appear to collide on the animation when they have not collided.

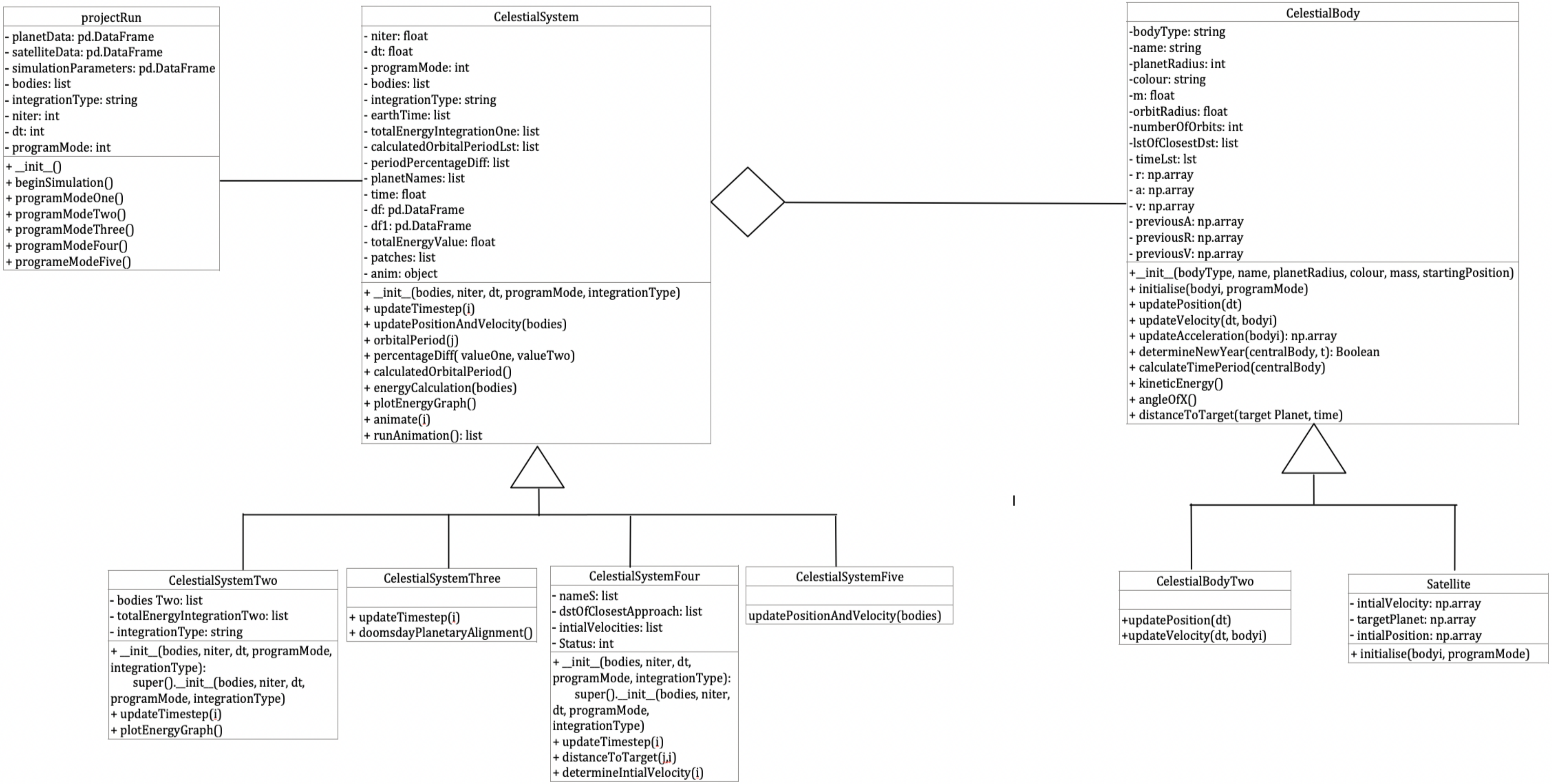

Figure 1: Program UML Representation

Results and Discussion

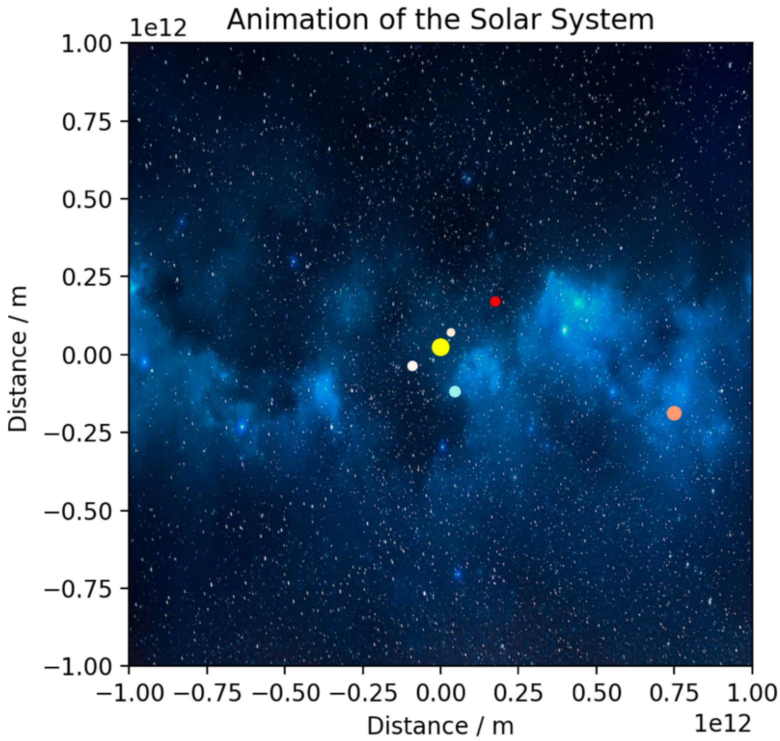

Figure 2: One frame of Solar System Animation

The animation displayed in figure 2 was used for all experiments. By observation one can see the circular orbits of the planets, implying a low eccentricity.

Orbital Period:

The program was run in mode one for 197.87 Earth years. The values for orbital periods were written to a file. Using the excel function AVERAGE, the average simulation value of period was calculated and presented in figure 3.

Planet	Literature value / Earth years *(Williams, 2023).*	Planetary influence considered	
		Value / Earth Years	Percentage Difference
Mercury	0.240919	0.240945	0.25689
Venus	0.615452	0.61541	0.26950
Earth	1.0006	1.0005	0.29570
Mars	1.8913	1.8910	0.28224
Jupiter	11.894	11.870	0.27068

Figure 3 Literature Orbital Period in comparison with simulation values

The low percentage differences suggest Beeman integration is accurate and well implemented. The simulation values are lower than the true values, there are various reasons for this, notably this is a 2D simulation therefore distances between planets will be slightly incorrect, only planets up to Jupiter have been included therefore gravitational force the outer planets were not considered. Planets are initialised along the positive x-axis this alignment is a rare occurrence, which may impact the orbital period in the relatively short period that they were measured. Furthermore, the approximation's limit and overall error of $O(\Delta t^3)$ have contributed.

Energy Conservation (Beeman Integration):

The program was run in mode one for 110.9 Earth years. A graph was produced (Figure 1) and values were written to the file.

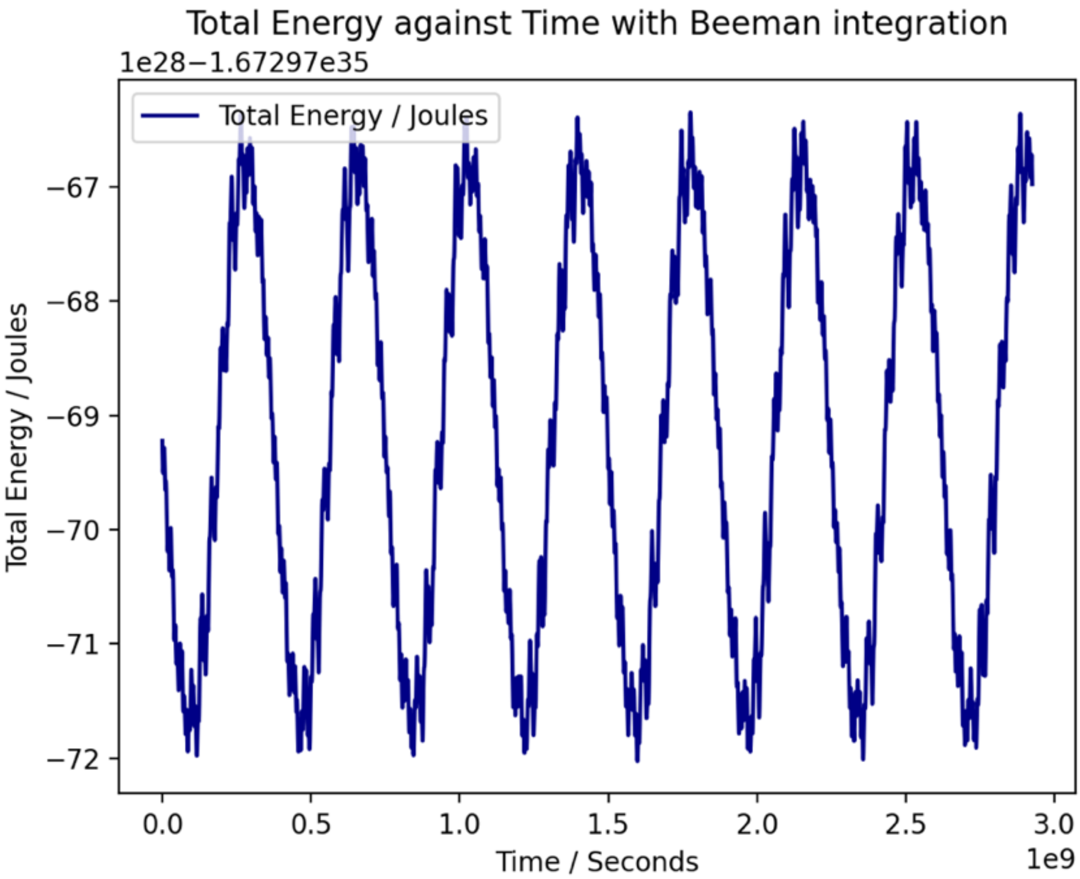

Figure 4 Total Energy against time for Beeman Method integration

When looking at the data, one can see energy is constant to 14 significant figures. This gives a total energy value of $-1.6729769222233 \times 10^{-35} J$. Figure 4 demonstrates the oscillations about this energy. These oscillations correspond to the planets being in different positions in their orbits.

Satellite to Mars

The simulation is run in mode four, to find the closest distance, the user can vary the velocity range tested. The minimum distance was found to be 6.9322 × 10^8 m, with a starting velocity of [7020, 7020] ms^{-1}. This occurred 3.15 Earth years after launch.

In comparison, the NASA perseverance probe took 203 days to arrive at Mars. This is more than three times less than the simulation, however perseverance used multiple trajectory correction manoeuvres to adjust the flight path, including these in the code could increase the accuracy of results.

The satellite did not return to Earth, the closest distance to Earth was not calculated.

Experiment One – Energy conservation comparison:

The program was run in mode two. The energy for both Beeman and Euler was written to the planetData file and a graph was displayed at 161.7 Earth years.

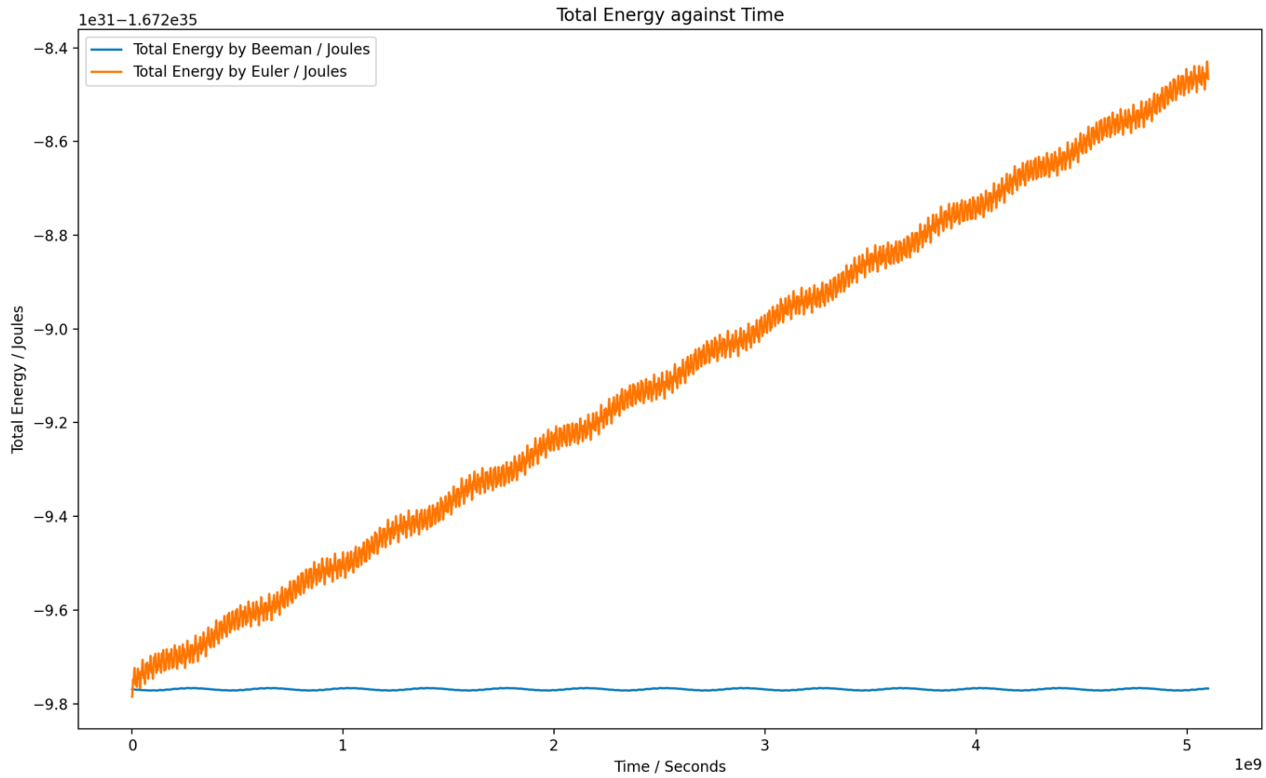

Figure 5 Total energy in Joules against time in seconds

Figure 5 demonstrates that Beeman is well conserved however Euler will diverge. As previously discussed, the error on Euler is equal to the timestep but will accumulate over time. Therefore, despite the small timestep of 0.00095 Earth years Direct Euler gets significantly worse over time. Moreover, the Direct Euler method does not consider the change in acceleration of the particles over the timestep which can lead to further inaccuracies.

The Beeman algorithms calculates the position and velocity of each particle to the fourth order Taylor expansion. One reason the total energy is conserved over long simulation times is because accelerations are recalculated at each step.

Experiment Two – Jupiter's Influence on Planetary Orbits

The simulation was run in mode five for 197.87 Earth years and the orbital periods were written to a file. Using the excel function AVERAGE, the average simulation value of period was calculated and presented in figure 6.

Planet	Literature value / Earth years *(Williams, 2023).*	Inter-Planetary influence considered		Inter-Planetary influence not considered	
		Value / Earth Years	Percentage difference	Value / Earth Years	Percentage Difference
Mercury	0.240919	0.240945	0.25689	0.240956	0.25733
Venus	0.615452	0.61541	0.26950	0.615439	0.27021
Earth	1.0006	1.0005	0.29570	1.0005	0.29454
Mars	1.8913	1.8910	0.28224	1.8912	0.27930
Jupiter	11.894	11.870	0.27068	11.894	0.28689

Figure 6: Literature Orbital Period in comparison with simulation values from both considering and not considering inter-planetary interactions.

Including solely the force from the sun instead of the gravitational force from all planets has negligible overall effect. As demonstrated by the data, the percentage difference between the actual and simulation values changes slightly. This could be due to how the gravitational force affects the acceleration which affects the velocity and position but the contribution to the net force that is lost by discounting inter-planetary forces is small in comparison to the force by the Sun, even in the case of Jupiter as it is 1000 times less massive than the Sun.

Conclusions

The animation displayed the orbital motion of the planets of the solar system in two dimensions. The orbital period and total energy were determined and written to an excel file. Optimum launch velocity for a satellite going to Mars was determined to be [7020, 7020] ms^{-1} and was 6.9322×10^{8}m, 3.15 Earth years after launch.

From the experiments, one concludes that that Beeman is much more accurate integration method which will conserve energy over long periods of time whereas Direct Euler will not. Furthermore, ignoring the inter-planetary interactions has little impact on the orbital period of the four innermost planets as it gives an overall percentage difference of 0.2670 in comparison to 0.2665 when considering all inter-planetary interactions.

To improve results, the code could be run for longer to get more data points for average values of orbital periods. To do this, further optimisation of the program is needed, specially addressing the latency when data is written to excel, for times greater than 100 years. This could be achieved by writing chunks of data accumulated in 50 years to a file, deleting this data and continuing.

Further extensions for this project include extending the animation to three dimensions. This would mean one is able to take the planet tilt into account making the simulation more realistic and not starting the bodies at the x axis. Moreover, the bodies' radius could be considered to determine their size in the animation.

Word Count: 2000

Bibliography

Britton Smith, M. Z. (2023). *Solar*. Retrieved from Computer Simulation Course Book 2023: https://www2.ph.ed.ac.uk/AardvarkDeployments/Protected/98172/views/files/Python/Deployments/CompSimLearn2223/deploymentframeset.html#tree=Python:Organisations:CompSimBeforeCourse2223

Brown, C. S. (2014). *A Sun-Centered View of the Universe*. Retrieved from Khan Academy: https://www.khanacademy.org/humanities/big-history-project/big-bang/how-did-big-bang-change/a/nicolaus-copernicus-bh

Field, J. V. (1999). *Johannes Kepler*. Retrieved from Mac Tutor: https://mathshistory.st-andrews.ac.uk/Biographies/Kepler/

NASA. (2020). *Cruise*. Retrieved from NASA science 2020 mission perservance rover: https://mars.nasa.gov/mars2020/timeline/cruise/

Nikolic, P. B. (n.d.). *Euler Method*. Retrieved from Computational Methods of Physics Phys: https://www.physics.udel.edu/~bnikolic/teaching/phys660/numerical_ode/node1.html

Nikolic, P. B. (n.d.). *Verlet Method*. Retrieved from Computational Methods of Phys: https://www.physics.udel.edu/~bnikolic/teaching/phys660/numerical_ode/node5.html

Williams, D. D. (2023, February). *Planetary Fact Sheet - Metric*. Retrieved from NASA: https://nssdc.gsfc.nasa.gov/planetary/factsheet/