Planetary & Satellite Orbit Simulation Using Beeman's Algorithm

Introduction

Using the Beeman algorithm on an object-oriented Python program, the orbits of Mercury, Venus, Earth, and Mars round the Sun were animated. The Beeman algorithm is a numerical integration method which predicts the position of the next timestep of a body by combining the current and previous acceleration of the body [1]. And the acceleration of each of the bodies is calculated using Newtonian mechanics with planetary data from a NASA database [2].

Having simulated the orbits of the planets, 3 separate experiments were conducted to test the simulation. The first was to test, print and compare the orbital periods of the simulated planets round the sun to periods calculated by NASA. The second experiment is to check whether the total (kinetic + potential) energy of the modelled system is conserved. The final experiment is to simulate the launch of a satellite, from Earth and perform a fly-by of Mars and ideally return it back to Earth. This experiment will also be compared to results from NASA's Mars Perseverance mission [3].

Methods

The entire program is broken into separate files within a folder; the main Project file that runs the code, an Energy text file where values of energy are printed in, and data text files for each planet and satellite which are used as inputs to create each of the objects.

The planet text files all have the same exact format containing information on planet's name, color, relative size, mass, position, and initial velocity; all in standard units. This strict format facilitates the reading of the files. The relative size was determined experimentally to provide an appropriate visual representation as shown in the image in figure 1. The positions and velocities were split into X and Y components using NumPy arrays, facilitating calculations by not having to break calculations into components.

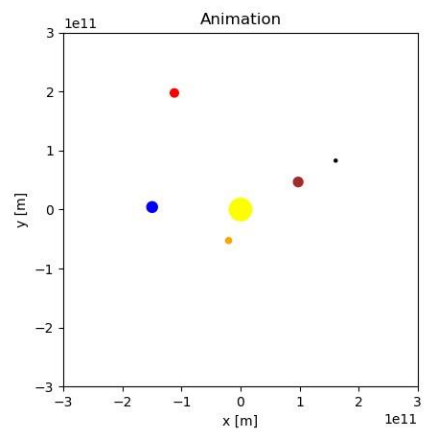

Figure 1- Snapshot of Animation

The initial position of each planet was assumed to be along the X-axis at the radius of orbit of each planet setting the average velocity of the planets only in the Y-component. This is true, in exception for the satellite, which contains initial velocity in both the X and Y component, this allows the satellite to travel outwards from Earth, towards Mars.

The energy text file is a file that gets wiped automatically every time the project file is run using the truncate() function and then gets written on with every total energy value calculated per timestep. This avoids clutter and confusion with previously calculated energy values.

The main project file is broken up into two separate classes, the body, and the simulation class. The body class creates each object from the sun to the satellite, and within, the body class contains variables and methods that belong only to that one specific body, such as the positions, velocity, or orbital periods. The simulation class contains a list of all these bodies, and the remaining methods and instance variables that include two or more of the bodies together, like the timesteps, the animation function and the calculations for the potential energies which require interactions between bodies. This ensures that the body class does not interact with other bodies simplifying the code.

As mentioned, the code begins by clearing the previous energy text file. It then runs the simulation class which starts with a constructor method reading the input files and sends them to the body class building all the planets and satellite. Each body built is then appended to a bodies list in the simulation constructor. This bodies list is used as it facilitates looping the calculations of each planet. The simulation constructor also contains lists needed to create the graphs and an instance variable for the number of iterations to run the program.

Then, the initialize function within the simulation class is summoned in main, this function is used to compute calculations for the first timestep of the animation by assuming that the previous acceleration of the planet is the same as the current acceleration, necessary to start the Beeman algorithm.

Then the createpatches function is summoned in main which begins the animation and loops the run_simulation for the assigned iterations and timestep. The timestep is currently set for a day as it provides an equilibrium between computational stress and precision of calculations. However, some experiments were done for 12- and 3-hour timesteps to see the effects of the timestep on the precision of results.

The run simulation function also summons a check_period method in the body class. This method is used to print the orbital periods of the planets through only one if statement. This if statement checks that the current y position is positive, that the previous y position was negative and that the orbit period print is true. This truth statement is used to skip printing the period of the satellite and prevent repeated prints of the orbital periods.

The run_simulation function itself loops the calculations methods of the Beeman algorithm for each planet and satellite returning their position for the animation. Run_simulation also summons methods to updates the time, calculate and write into a file the energy of the system. It also appends calculations of energy and distance of satellite to Mars to a list for each timestep. And after a certain number of iterations, an if statement is used to prints the minimum distance of satellite to Mars, the graphs for energies and the graph of distance of satellite to Mars over time from the mentioned lists.

Results and Discussion

The first experiment was to calculate the orbital period of the planets via printing the time elapsed for the planet to complete one orbit and compare them to the literature value of the periods obtained by NASA. Below is the table of the calculated periods for 24, 12 & 3 hour timesteps together with the literature values [2].

Planet	Orbit periods in days for different timesteps			Literature value of Orbit [days] [2]
	24-hour intervals	12-hour intervals	3-hour intervals	
Mercury	86.0	85.5	85.4	88.0
Venus	225.0	224.5	224.1	224.7
Earth	366.0	366.0	365.8	365.2
Mars	685.0	685.0	684.8	687.0

As can be seen in the table above, the effects of the change in timestep are very small on the results showing a slight decreasing trend, moreover, the results from the periods from Beeman algorithm all fall under a 3% error from literature value. This error is possibly attributed to the assumption that all planets start on the x-axis or the uncertainty in values from NASA's database [2]. However, all the results are physically reasonable and accurate to 3% error.

The second experiment involved calculating the total energy of the system over time for different timesteps, and to determine whether the total energy in the system is conserved or not. The code would print a graph of the total energy over time after a certain number of iterations in the program, in all cases, this was taken to be 4000 iterations, which for the 1-day timestep equates to 4000 days, enough time to see if there is any long-term drift in energy.

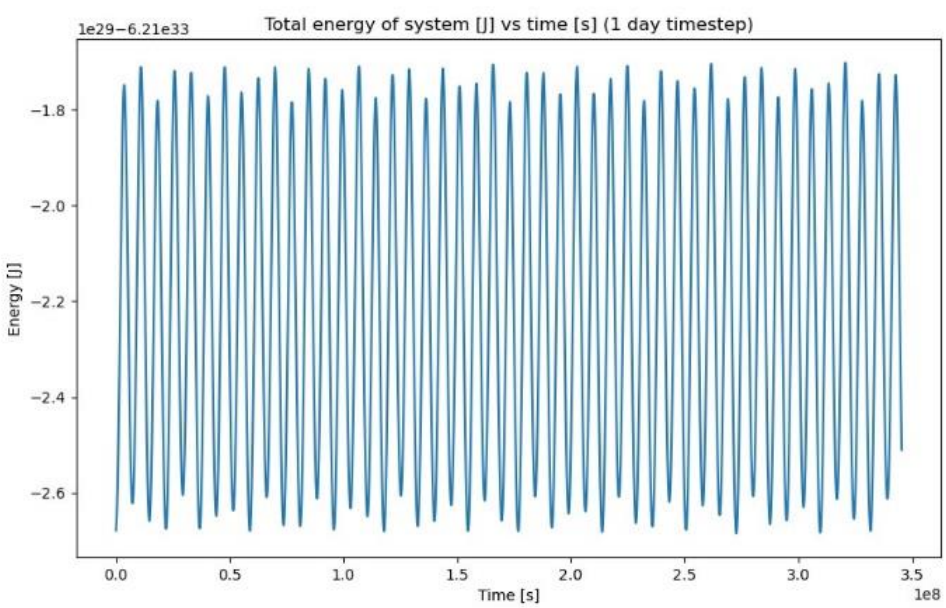

Figure 2- Graph of the total energy of the system [J] over time [s] for 1-day timesteps after 4000 iterations.

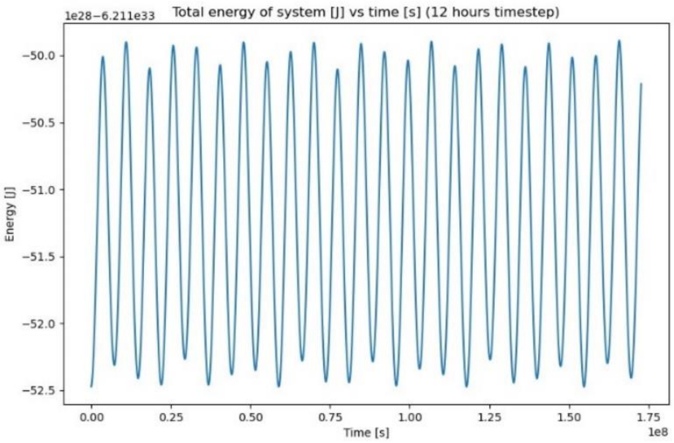

Figure 3- Graph of the total energy of the system [J] over time [s] for 12 hours timesteps after 4000 iterations.

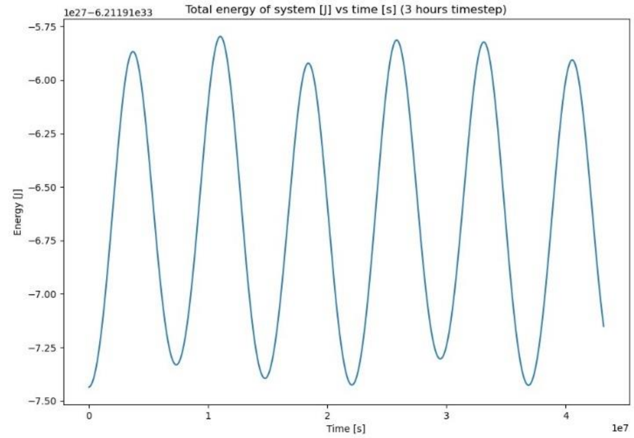

Figure 4- Graph of the total energy of the system [J] over time [s] for 3 hours timesteps after 4000 iterations.

As can be seen in figures 2, 3 and 4, the total energy is showing no long-term drift and seems to slightly oscillate. Moreover, it can be seen, that the smaller the timestep, the smaller the oscillations in energy, whereby the oscillations in energy for the 1-day timestep is of order 1e29J, for 12-hour steps is 1e28J and 1e27J for a 3-hour timestep. However, the total energy is conserved at approximately -6.21e33J which corresponds to less than 1% error to the value obtained by San Jose University of -6.2e33J [4]. This result is both physically reasonable due to the conservation of energy and lies within 1% accuracy from other experimental results. The oscillations can be attributed to uncertainties in the Beeman algorithm.

The final, and most complex experiment is to simulate the launch of a satellite from Earth and perform a fly-by of Mars. To do so, a similar trajectory that NASA's perseverance mission was used, whereby the satellite was launched towards the future position of Mars as shown in the figure 5 below [3].

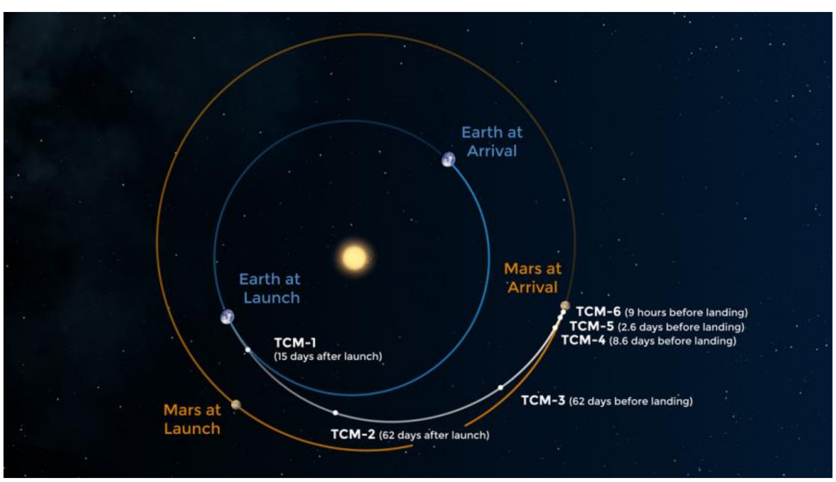

Figure 5- NASA's Perseverance mission satellite's trajectory to Mars [3].

The condition of velocity for closest approach for 1-day timesteps was obtained experimentally to be approximately $(11297{,}29320)ms^{-1}$. This would make the satellite flyby Mars at 135 million meters after 188 days with a cruising speed of $31421ms^{-1}$ as shown in figure 6. However, the Perseverance mission used a cruising speed of $11000ms^{-1}$, three times less than the speed obtained. Also, the Perseverance satellite took 202 days, to reach Mars, 14 more than that obtained [5].

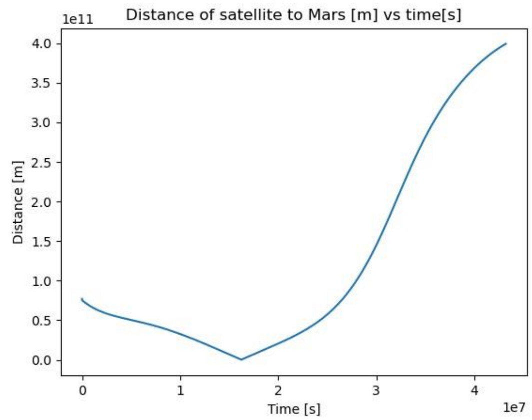

Figure 6- Distance of satellite to Mars over 500 days with minima of 135million meters at 188 days.

This deviation in results is from the assumption that the satellite is released when Earth and Mars are perfectly aligned with the Sun. However, this was not the case for the Perseverance mission as Mars was slightly ahead of Earth's orbit during launch. Moreover, the perseverance satellite used path corrections to reach Mars which the program does not use, hence the deviation of 135 million meters, which is still relatively small for interplanetary travel. This deviation is also caused by errors in precision caused by the timesteps.

To obtain more reliable results, using exact planetary positions would provide a more appropriate starting condition for the satellites launch. The code could also implement path corrections to perfect the landing, and to them make the satellite return to Earth, which is currently not the case in the program. Overall, the experiment provides physically reasonable results and a decent plan of approach but is not fully comparable to the Perseverance mission due to the caveats mentioned.

Conclusions

The program set out to provide a physical representation of the planets orbit round the sun and provide results to a set of experiments. The first experiment for the orbital periods was completed satisfactorily over several timesteps with results all within under 3% literature values [2]. The second experiment for the energy over time showed miniscule oscillations of the energy with range of oscillation proportional to the size of the timestep. The oscillations were round the value of -6.21e33J which corresponds to less than 1% error to the value obtained by San Jose University of -6.2e33J [4].

The final experiment provided a physically reasonable plan of approach for a satellite to reach Mars whereby the satellite would make a flyby at 135million meters after 188 days with a cruising speed of $31421ms^{-1}$. However, when comparing to the perseverance mission, NASA used a cruising speed of $11000ms^{-1}$, three times less than the speed obtained and a journey of 202 days [5]. However, the deviation in results is from the assumption that the satellite is released when Earth and Mars are perfectly aligned with the Sun, which was not the case for the Perseverance mission.

Therefore, by using exact planetary positions from a database, and by creating a method to correct the path of the satellite, as done by NASA, the program would provide more reliable and comparable results for the satellite experiment. On the other hand, it would be interesting to test the effect of Jupiter, Saturn and Neptune on the orbits of the planets (if any), as these planets are massive.

Bibliography

[1] "Beeman's Algorithm - Wikipedia". 2022. *En.Wikipedia.Org*. https://en.wikipedia.org/wiki/Beeman%27s_algorithm.

[2] "Planetary Fact Sheet". 2022. *Nssdc.Gsfc.Nasa.Gov*. https://nssdc.gsfc.nasa.gov/planetary/factsheet/.

[3] "Trip To Mars". 2022. *Mars.Nasa.Gov*. https://mars.nasa.gov/mars2020/timeline/cruise/.

[4] San José State University "The Potential, Kinetic And Total Energies Of The Planets". 2022. *Applet-Magic.Com*. http://www.applet-magic.com/planetenergy.htm.