Simulation of a Solar System

Steinn Hauser, steinnhm

November 2017

Institute of theoretical astrophysics, Blindern University, Oslo
steinnhauser@mac.com

Abstract

This is a programmable model to accurately describe the problems encountered when simulating a solar system for analyses and attempting to land a space probe on the surface of one of the planets. The simulation uses numerical approximations to compute trajectories of the planets as well as the trajectory of the space probe landing on its designated planet. This model will include a few simplifications of the challenges of programming the true nature of, for example, our own solar system and its complexities. The simulation model will use Newton's Gravitational formula (in abundance) to calculate the behaviour of the solar system as well as predict what the trajectory of the probe will look like. Newton's gravitational formula states that the gravitational force F is equal to:

$$F = \frac{GMm}{r^2}$$

Where M is the mass of the central star, m is the mass of the other body in the system, r is the distance between the two, and G is Newton's gravitational constant. The simulation assumes that the planets in the solar system will be affected only by the gravitational pull of the central star; the model does not take into consideration gravitational forces and collisions between the planets. In addition, the central star will not be affected by the gravitational pulls of the planets surrounding it. When landing the probe, it will be affected only by the gravitational pull of the planet it is landing on and the air resistance encountered when entering the planet's atmosphere. The planet it will land on is also assumed to be static and not be accelerated by any gravitational forces. These computations will be carried out in python 2.7.12 and take place on a two dimensional plane for simplicity's sake. Euler's Method is the algorithm used to calculate the trajectories of the planets and the space probe landing. This method, when used accurately enough, has a small margin of error which is negligible in the final results. The simulation will be stopped when the outermost planet completes one full orbit around the central star and when the probe hits its planet's surface. The program took a total of 35 seconds to calculate and resulted in a successful simulation of the planet's trajectories, as well as a successful landing on the surface of one of the planets. A successful landing, in this case, is interpreted as an entry where the probe neither burns up in the atmosphere nor hits the ground at a fatal speed.

Introduction

In order to successfully navigate and simulate the solar system, its predictability and its tendency to follow the most fundamental laws of physics with a surprisingly small margin of error must be acknowledged. The largest factor on the planetary scale is the gravitational force of the central star acting on its surrounding planets, so any other factors the simulation does not take into account will not have catastrophic consequences for the results. This model will not take into account the forces from the planets acting upon the star since the star will be placed in the origin and if there were to be any deviation of the star from its initial position it would result in a far more complex model. All the planets have their respective initial positions and initial velocities which will dictate the trajectories they will follow. The gravitational force acting upon each planet will be calculated by Newton's Gravitational equation, which states:

$$\vec{F}_g = -\frac{GMm}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|}$$

Where $\vec{r}$ is a position vector pointing from the star (in the origin) to the respective planet, M is the mass of the star, m is the mass of the planet and G is Newton's gravitational constant $G \approx 6.67 \cdot 10^{-11}$. This formula, in addition to Newton's second law $\vec{F} = m\vec{a}$ will cause each of the planets to be accelerated inwards towards the sun and result in the planets moving in a trajectory depending on their initial conditions. If a planet were to have an initial speed over its escape velocity, it will inevitably have a hyperbolic trajectory away from our system. This does not have to be the case. The planet in question can also have a circular orbit with a centripetal acceleration inwards if the initial position and velocity are appropriate, otherwise the planet will have an elliptical orbit around the star. The same can be said for the probe landing on one of the planets. The initial parameter values will dictate what the trajectory looks like and therefore need to be fine-tuned to have as efficient a landing as possible. We will return to this later. However, a numeric calculation model is needed to calculate and simulate any of these trajectories in the first place. Enter Euler's Method:

Euler's Method

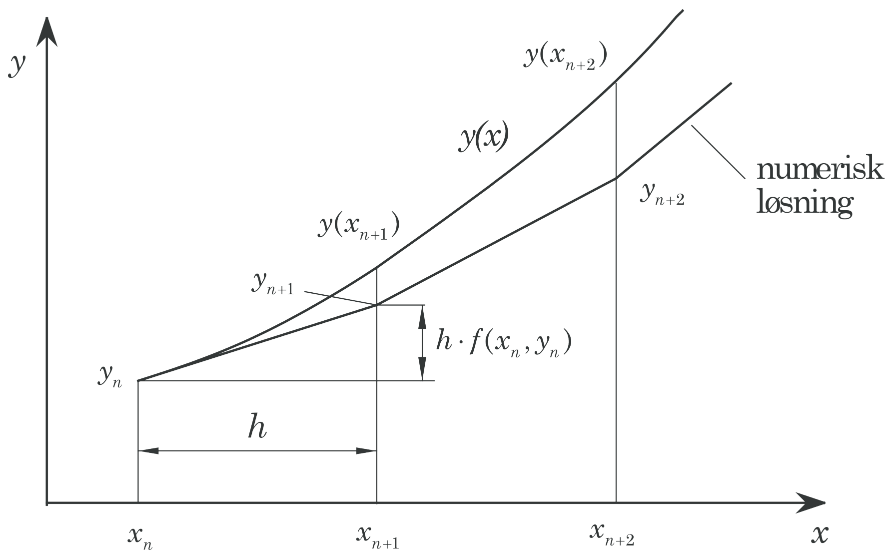

Figure 1: Numeric calculation vs true trajectory

The most central concept in a numerical calculation of trajectories is using the slope of a curve to calculate where the curve will be at a later point. As long as the slope of the curve is known at all points on the x-axis and the curve has a start point (x_0, y_0), there will be an equivalent non-trivial numerical approximation of the true values of the curve. This approximation is calculable by Euler's Method, which is an algorithm that takes the derivative (slope) of a curve as an input at any given point and produces an extension to the next point in the curve accordingly (as illustrated in figure 1). Figure 1 uses the parameter h as the distance between two points. However, in this case the simulation is in the xy-plane and uses a third "time dimension" where h is a small time interval Δt which will update the position (x, y) accordingly. From figure 1 it is clear that the smaller the time step Δt is, the more accurate the numerical approximation will become. However, the shorter the time interval is, the more calculations will be needed for the simulation and the longer time the program takes to run. Finding a balance of accuracy and quick simulation run time will be addressed at a later point. Writing Euler's method mathematically includes adding a short extension of the tangent vector to our previous position vector to follow a curve as closely as it can, depending on the time step Δt (the shorter the tangent vector the more exact the numeric curve becomes). This allows the simulation of the velocity of any given object so long as we have its acceleration (velocity derivative) and the trajectory patterns become very predictable once we have the velocity vectors (position derivative). Using the formula $\ddot{\vec{r}} = \dot{\vec{v}} = \vec{a}$ returns:

$$\vec{r}(t+\Delta t)=\vec{r}(t)+\dot{\vec{r}}(t)\Delta t$$

$$\vec{v}(t+\Delta t)=\vec{v}(t)+\dot{\vec{v}}(t)\Delta t$$

Decomposing the position and velocity vectors into their respective x and y components, making them functions of time $\vec{r}(t)=\vec{r}(x(t),y(t))$ returns:

$$\vec{r}(x(t+\Delta t),y(t+\Delta t))=\vec{r}(x(t),y(t))+\vec{v}(v_x(t),v_y(t))\Delta t$$

$$\vec{v}(v_x(t+\Delta t),v_y(t+\Delta t))=\vec{v}(v_x(t),v_y(t))+\vec{a}(a_x(t),a_y(t))\Delta t$$

If a_x and a_y are calculable, the final requirements are simply the initial values x_0, y_0, v_{x0} and v_{y0} at time t=0. Here is an example of how the calculations look:

t=0

$$\vec{r}(x_1,y_1)=\vec{r}(x_0,y_0)+\vec{v}(v_{x0},v_{y0})\Delta t$$

$$\vec{v}(v_{x_1},v_{y_1})=\vec{v}(v_{x_0},v_{x_0})+\vec{a}(a_{x_0},a_{y_0})\Delta t$$

t=$1\Delta t$

$$\vec{r}(x_2,y_2)=\vec{r}(x_1,y_1)+\vec{v}(v_{x1},v_{y1})\Delta t$$

$$\vec{v}(v_{x_2},v_{y_2})=\vec{v}(v_{x_1},v_{x_1})+\vec{a}(a_{x_1},a_{y_1})\Delta t$$

t=$2\Delta t$

$$\vec{r}(x_3,y_3)=\vec{r}(x_2,y_2)+\vec{v}(v_{x2},v_{y2})\Delta t$$

$$\vec{v}(v_{x_3},v_{y_3})=\vec{v}(v_{x_2},v_{x_2})+\vec{a}(a_{x_2},a_{y_2})\Delta t$$

$$\vdots$$

t=$i\Delta t$

$$\vec{r}(x_{i+1},y_{i+1})=\vec{r}(x_i,y_i)+\vec{v}(v_{x_i},v_{y_i})\Delta t$$

$$\vec{v}(v_{x_{i+1}},v_{y_{i+1}})=\vec{v}(v_{x_i},v_{y_i})+\vec{a}(a_{x_i},a_{y_i})\Delta t$$

..and so on. Here the acceleration vector $\vec{a}$ needs to be updated each time step for the simulation to run. Another variant of this is the Euler-Chromer Method, which is the same except instead of adding the slope of the position curve in a point i, you add the derivative from the point $i+1$. This means the velocity must first be calculated normally, and once $\vec{v}(i+1)$ is known, $\vec{r}(i+1)$ is calculated in the following fashion:

$$\vec{r}(i+1)=\vec{r}(i)+\vec{v}(i+1)\Delta t$$

This is the method variant which will be used in the simulation, as it is more accurate in describing the true trajectory of the object.

Methods

The methods of the simulation will mostly include knowing where the planet/probe is at a given time, and accelerating it accordingly. The Euler-Chromer method will cause the acceleration to affect the velocity vector, and in turn, the position vector will also be updated. This method is a great one because as long as all the forces of the system ($\vec{F}_1 + \vec{F}_2 + ...$) are accounted for (as well as the initial parameter values), the trajectory for an object with mass m over a given time $T = N\Delta t$ can be calculated quite easily by the following steps:

$$\sum \vec{f} = \vec{F}_1 + \vec{F}_2 + ...$$

$$\vec{F} = \sum \vec{f}$$

$$\vec{a} = \vec{F}/m$$

$$\vec{v}(i+1) = \vec{v}(i) + \vec{a}\Delta t$$

$$\vec{r}(i+1) = \vec{r}(i) + \vec{v}(i+1)\Delta t$$

..and so on, from $i = 0$ to $i = N$.

Solar System Method

Since the planets orbit independent of each other, each planet's full orbit can be calculated before moving on to the next. The simulation will run until the outermost planet has completed one orbit, so the program will begin by calculating the one farthest from the origin to find out how long the simulation will run. Once the total simulation time T is known, each planets course can be calculated using a for loop that simulates a given planets position and velocity using the Euler-Chromer method:

$$\vec{v}_{i+1} \approx \vec{v}_i + \vec{a}\Delta t$$

$$\vec{r}_{i+1} \approx \vec{r}_i + \vec{v}_{i+1}\Delta t$$

To find the number of calculations, the total simulation time T is divided by our time step intervals Δt:

$$N = \frac{T}{\Delta t}$$

Where Δt can be increased or decreased to get faster or more accurate simulations, and the number of calculations N must be a natural number integer $N \in \mathbb{N}$. The bulk of the solar system simulation will be a for loop that cycles through each planet, calculating its total course with N steps before moving on to the next planets course. These planet trajectories will be saved in a DxPxN array where D is the dimension of the calculation (here D=2), P is our number of planets (here P=7), and N is the number of calculations (steps) made. The following equations were used to calculate the acceleration vector $\vec{a}$:

$$\vec{F} = -\frac{GMm}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|}$$

$$\vec{F} = m\vec{a} \Rightarrow \vec{a} = -\frac{GM}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|}$$

$$|\vec{r}| = \sqrt{x^2 + y^2}$$

Where $\vec{r}$ is the planets position vector, and (x, y) is the planets position in the xy-plane.

Once the acceleration is computed, the Euler-Chromer method only needs initial conditions x_0, y_0, vx_0 and vy_0 which describe the planets position and velocity vectors at $t_0 = 0$. These initial conditions vary from planet to planet, so the values will be reset each time the program finishes one planet calculation loop to compute the next one's. The program will begin by calculating the outermost planet (largest absolute distance from the star), and set the simulation period T accordingly when an orbit is completed. Measuring when an orbit is completed involves measuring when the planet has the same position as the start position within some tolerance δr. Once the simulation of the outermost planet is completed, the program will calculate the rest of the planets with the same simulation time T and with the same time interval Δt. This can result in some of the innermost planets completing multiple orbits, but this will not have any consequences for the model. After this simulation is completed, the program will begin calculating the trajectory of a probe landing on one of the planets.

Probe Landing Method

This simulation is tricky in more ways than one. On one hand, there is just one trajectory to compute now, as the planet the probe will attempt to land on will not be affected by any outside forces. However, there is an additional force acting upon the probe caused by the planets atmosphere. The model which will describe this gas resistance (drag force) is as follows:

$$F_D = \frac{1}{2}\rho A v^2$$

Where ρ is the density of the atmosphere at any given position, A is the area (in square meters) of a parachute which will slow the probe down in entry, and v is the velocity of the probe. This force is a more complicated one than say, gravity, as it depends on the position of the probe to calculate the density of the atmosphere (the density is largest at the planet's surface) as well as depending on the velocity of the probe to dictate the magnitude of the force.

Since the planet has a radial symmetry, it is safe to say that the gas density of the planet is spherically symmetric around the whole planet. This means that the density at any given position only depends on the probe's height off the ground. The density of the planet as a function of the height off the ground can be modelled by the equation:

$$\rho(h) = \rho_0 e^{-h/h_{scale}}$$

Where h is the probe's height off the planets surface, ρ_0 is the surface density of the planet (density at h=0), and h_{scale} is the scale height of the atmosphere, expressed by $h_{scale} = 75200/g$. Where g is the gravitational acceleration at the surface of the planet. This acceleration is calculable by Newton's second law in addition to Newton's gravitational formula:

$$ma = \frac{GMm}{r_p^2} \Rightarrow a = g = \frac{GM}{r_p^2}$$

Where r_p is the radius of the planet, M is the mass of the planet and G is Newton's Gravitational constant.

Implementing the drag force also involves giving it a direction in which it will accelerate the probe. The drag force is similar to a frictional force on ground, where the force will attempt to nullify a velocity in a certain direction and drag it in the opposite direction of its velocity. In other words, the force's direction depends on the probe's current velocity vector and will act as a resistance to it. In mathematical terms, this can be written as the force having the direction of the negative velocity vector:

$$\vec{F}_D = \frac{1}{2}\rho(h) A v^2 \frac{(-\vec{v})}{|\vec{v}|}$$

Since $v = |\vec{v}|$, the expression can be simplified to:

$$\vec{F}_D = -\frac{1}{2}\rho(h) A |\vec{v}|\vec{v}$$

The area of the parachute can be changed and corrected for a safe landing, but the general goal is to have it be as small as possible to minimize probe weight, and material costs.

Now that all the forces acting on the probe are accounted for, the Euler-Chromer method can be initiated:

$$\vec{F} = \sum_i \vec{f_i}$$

$$\sum_i \vec{f_i} = \vec{F}_G + \vec{F}_D = \frac{GMm}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|} - \frac{1}{2}\rho(h)A|\vec{v}|\vec{v}$$

$$\vec{a} = \vec{F}/m$$

$$\vec{v}(i+1) = \vec{v}(i) + \vec{a}\Delta t$$

$$\vec{r}(i+1) = \vec{r}(i) + \vec{v}(i+1)\Delta t$$

from $i = 0$ to $i = N$. Where the probe's height off the ground is found by subtracting the planet's radius from the absolute value of the position vector:

$$h_i = |\vec{r}_i| - r_p$$

In this case, the initial parameters must be estimated and fine-tuned for an ideal landing trajectory. This landing trajectory needs to be such that the probe does not enter the atmosphere at too high a speed, causing the drag force to rip the parachute apart. To accomplish this, it is best to have the probe pass slightly through the atmosphere multiple times in an elliptical fashion in order to slow it down properly before entry. The simulation must also account for the probe not hitting the planet's surface too hard on landing. If the probe were to hit the surface at a speed of roughly $3\frac{m}{s}$, the instruments and any astronauts inside would be totaled. Finding the initial conditions that produce a safe landing involves a lot of trial and error.

The results of the probe landing, as well as the solar system simulation, are as follows:

Results

Plotting the x and y positions of the solar system planets produced the following graph:

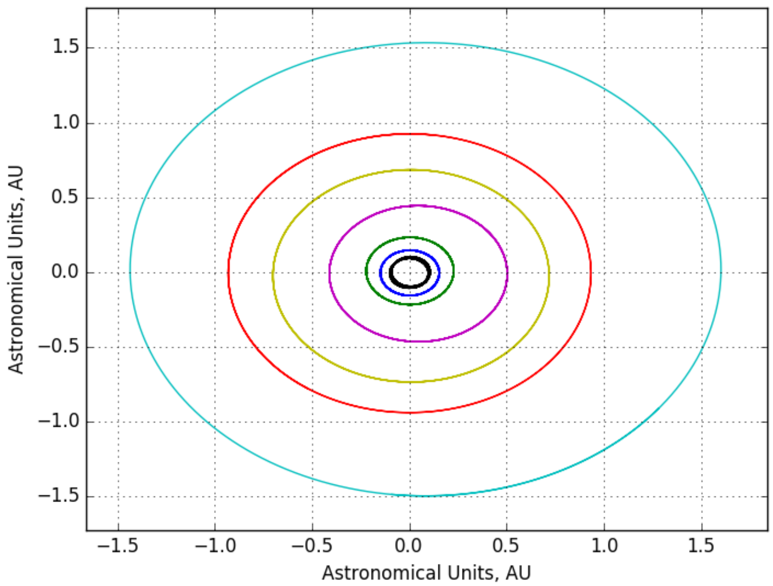

Figure 2: Trajectories of the planets

These are the trajectories of the seven different planets of the solar system with the star in the origin $(0, 0)$. Following is the plotted data of the probe landing:

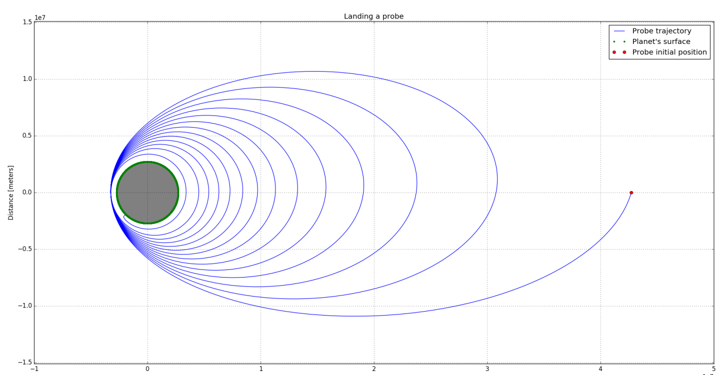

Figure 3: Trajectory of the probe

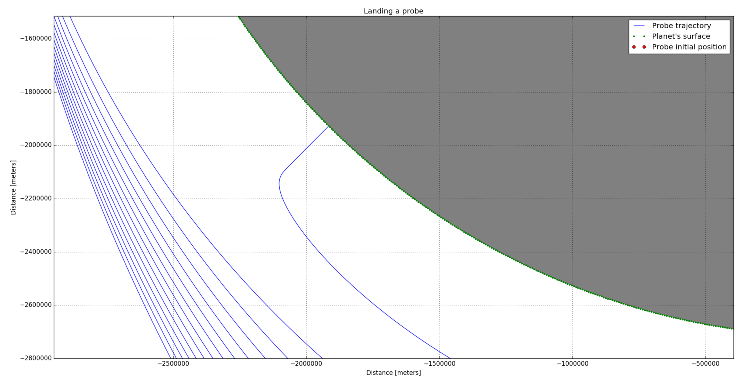

Figure 4: Probe landing

This trajectory was achieved with the initial conditions:

$$\vec{r}_0 = x\hat{x} + y\hat{y} = (42717km, 0)$$
$$\vec{v}_0 = v_x\hat{x} + v_y\hat{y} = (-70\frac{m}{s}, -245\frac{m}{s})$$
$$A = 8m^2$$

Comments

A thing worth noting for the planet orbits is that the planets have trajectories that are approximately circular. This means that the centripetal acceleration inwards towards the star can be approximated to $a \simeq v^2/r$. This means that the numeric values can be put to the test by the analytic ones to make sure the simulation is running appropriately. These results are verifiable with Kepler's laws; according to Kepler's Laws, the velocity on circular orbit that encompasses all gravitation mass can be simplified to:

$$F = \frac{GMm}{r^2}, F = m\frac{v^2}{r}$$
$$\frac{GMm}{r^2} = m\frac{v^2}{r}$$
$$v = \sqrt{\frac{GM}{r}}$$

Where G is the gravitational constant, M is the mass of the star, and r is the distance from the star to the planet in question. This analytic solution to the numeric planet velocities can be used to check that the program calculations were sufficiently accurate. Also notable is that the planets closest to the center seem to have thicker lines plotted, which is simply because the planets at the center have completed several orbits (with slight variants in trajectories) in the same time that the outermost planet completed one.

The results of the probe landing were very successful as the probe entered the atmosphere at a very low planetary orbit which produced a low entry velocity. This is optimal as the heat shield of the probe cannot take a drag force over $25000N$, and the drag force is proportional with v^2. The probe landed on the ground safely as a result. The initial conditions for the probe trajectory were not easy to find to achieve a good landing, and were found through much trial and error. The probe would often times orbit the planet without coming close enough to the atmosphere to slow down sufficiently to make a landing at all (the probe had an elliptical orbit). Other times the probe would have a trajectory entering the atmosphere too harshly, and would either be destroyed by the drag force or hitting the ground too hard. In the case of the initial values used, the drag force on the probe went through a verification each time step so as to check whether or not they were too high. Equivalently, the velocity of the probe had to be checked at landing, to make sure it was not over $3\frac{m}{s}$. It is largely only the radial velocity which has consequences for the probe landing, but seeing as the drag force works nullifies the angular velocity, measuring the absolute velocity has the same result.

Conclusion

The simulation of the solar system had many simplifications, but was nicely modelled and simulated with the Euler-Chromer method. If the simulation were to be modelled more accurately (where all bodies are affected by all gravitational forces), the simulation model would not be to calculate one at a time as these simplifications have allowed. But rather to update all bodies positions and velocities at each time step, resulting in a more dynamic and complex system.

The model for the probe was a challenge as the goal was to achieve as safe a landing as possible. The ideal way to land a probe is to have it slow down as much as possible before entering the planet's atmosphere. This can be accomplished through external forces such as combustion engines, or slowing it slightly each orbit with the planets atmosphere. Other factors like this could be the gravitational force of an external body such as a moon or the central star, but these were not implemented into this model.

Summary

The behaviour of a solar system is neither unpredictable nor difficult to analyze under the right circumstances. The only requirements are an understanding of the fundamental forces and sufficient computational power, as well as sufficient data about the planets and central star (masses, densities, atmosphere data, etc.). These factors, in addition to a measurement of an objects initial position and velocity, are all that is needed to program a reasonably accurate simulation of the cosmos, and any mission to traverse through it.

References

Frode K. H. Astronomy Lecture notes

http://www.uio.no/studier/emner/matnat/astro/AST2000/h17/undervisningsmateriale-2017/forelesningsnotater/part1b.pdf

AST2000SolarSystemViewer Module for planet information (seed=52772)

http://www.uio.no/studier/emner/matnat/astro/AST2000/h17/undervisningsmateriale-2017/filer-til-oppgaver/ast2000solarsystemviewer_v2.pyc