Satellites beyond the Solar System

Guner Aygin

17 December 2021

Abstract

This report outlines the steps taken to simulate a transfer orbit from Earth to Jupiter, and a gravity assist at interception. The ideal initial conditions to launch a satellite out of the Solar System were found to be a starting position of 8.101×10^{11} away from Jupiter, at an angle of $97.2°$ from Earth. The ideal additional initial velocity was found to be -9.75 m·s^{-1}. It was also determined that the gravitational pull of Mars would not be sufficient to perform such a manoeuvre.

1 Introduction

A lot of energy is required to overcome the gravitational potential of the Sun when trying to leave the Solar System. An efficient way of gaining this energy is to utilise the gravity of another planet to 'sling shot' out of the solar system, without the need for extensive thrust power, saving space agencies billions of dollars.

Jupiter is the largest planet in the solar system [1], and as such is the ideal planet to use when performing a gravity assist manoeuvre. This report outlines the theory of when to launch a satellite such that it intercepts the planet using the minimum energy possible, and the equations of motion which describe how the satellite moves due to the force of gravity in section 2. The algorithms used to simulate this are outlined in section 3, with the results explained in section 4.

2 Theory

2.1 Optimum launch time

The most efficient time at which to launch a satellite from Earth to Jupiter is when, after half an orbit, the satellite intercepts Jupiter; as seen in Figure 1. This requires the least amount of energy to achieve. To make sure the two bodies intercept at this point the position of the planets at launch time must be determined by calculating where they are half an orbit before the interception.

From Figure 1 we can see that the satellite travels in an elliptical orbit, anti-clockwise with the other planets, with a semi-major axis

$$a = \frac{1}{2}(R_E + R_J), \tag{1}$$

where R_E and R_J are the radii of Earth and Jupiter's orbits respectively. The period of the orbit only depends on the semi-major axis and is given by

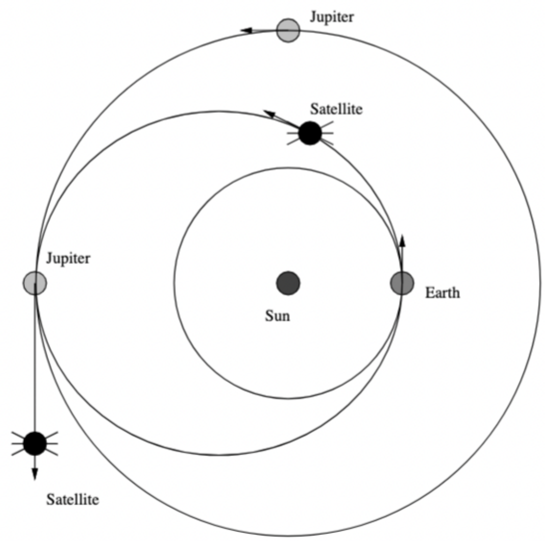

Figure 1: Diagram of the transfer orbit from Earth to Jupiter [2].

$$T_{\text{sat}} = \frac{2\pi a^{\frac{3}{2}}}{\sqrt{GM_{\odot}}}, \tag{2}$$

where G is the gravitational constant, and $M_{\odot}$ is the mass of the Sun. For simplicity, we assume that the planets lie in a 2D plane, with zero eccentricity to their orbits. We define the launch position of the satellite to be at coordinates $(R_E, 0)$, where $(0, 0)$ is the position of the Sun (as in Figure 1). To determine the position of Jupiter at launch, we need to go back from its final position at $(-R_J, 0)$ by $\frac{1}{2}T_{\text{sat}}$. As we are assuming no eccentricity, the velocity of Jupiter is always constant. The x and y position of an object in circular (or elliptical) motion are given by

$$x = r\cos\left(\frac{2\pi t}{T}\right), \tag{3}$$

$$y = r\sin\left(\frac{2\pi t}{T}\right), \tag{4}$$

where t is the time elapsed, r is the orbital radius, and T is the period of the object's orbit. In this scenario, $T = T_J$, and $r = R_J$. With equations 3 and 4 it is possible to find the position of Jupiter at launch, and thus the optimum time at which to launch the satellite. The velocity is the time derivative of position, given by

$$v_x = -\frac{2\pi r}{T}\sin\left(\frac{2\pi t}{T}\right) \tag{5}$$

$$v_y = \frac{2\pi r}{T}\cos\left(\frac{2\pi t}{T}\right) \tag{6}$$

2.2 Hohmann transfer orbit

The trajectory taken by the satellite as it changes from an orbital radius of R_E to R_J is known as a Hohmann transfer orbit [3]. As the satellite moves in an elliptical orbit, it has a constantly changing radius

$$r = \frac{a(1-e^2)}{1+e\cos\left(\frac{2\pi t}{T}\right)}, \tag{7}$$

where e is the eccentricity and here $T = T_{\text{sat}}$ [4]. At periapsis, the radius of the satellite's orbit is R_E, so using equation 7 we obtain $e = 0.678$.

Equations 3 and 4 give the position of the planet during the transfer orbit, noting that r must be recalculated for every point. It is also possible to describe the trajectory of the satellite in terms of two sets of second-order differential equations [5], one for each dimension:

$$\frac{d^2\lambda}{dt^2} = GM'\frac{\lambda' - \lambda}{R^3}, \tag{8}$$

where $\lambda = \{x, y\}$ are the respective coordinates of the satellite, and $\lambda' = \{x', y'\}$ and M' are the respective coordinates and mass of the other gravitational body. R is the distance between the two bodies, given by

$$R = \sqrt{(x'-x)^2 + (y'-y)^2} \tag{9}$$

If there are N bodies that have a gravitational effect on the satellite, then the equations of motion are given by

$$\frac{d^2\lambda}{dt^2} = \sum_{i=1}^{N} GM'_i \frac{\lambda'_i - \lambda}{R_i^3}. \tag{10}$$

2.3 Impulse

The impulse J of an object is defined as

$$J = m\Delta v, \tag{11}$$

where m is the mass of the object, and Δv is the change of velocity. When the satellite launches from Earth it must thrust in the direction of its velocity vector, to increase its radius. This results in two impulses for the orbit, an impulse from the acceleration from Earth's orbit to the Hohmann transfer orbit, and an impulse from the deceleration from the Hohmann transfer orbit to Jupiter's orbit; given by

$$\Delta v_1 = v_\pi - v_E = \sqrt{\mu\left(\frac{2}{R_E} - \frac{1}{a}\right)} - \sqrt{\frac{\mu}{R_E}}, \tag{12}$$

$$\Delta v_2 = v_J - v_\alpha = \sqrt{\frac{\mu}{R_J}} - \sqrt{\mu\left(\frac{2}{R_J} - \frac{1}{a}\right)}, \tag{13}$$

where $\mu = GM_\odot$, and v_π and v_α refers to the velocity of the transfer orbit at perigee and apogee respectively. The initial velocity of the satellite is therefore v_π. The overall theoretical impulse for this transfer is $J = \Delta v_1 + \Delta v_2 = 14.42$ km·s^{-1}.

2.4 Gravity assist

A gravity assist is the acceleration of the satellite by the gravitational force of a planet, requiring no additional thrust from the satellite. It can be modelled as an elastic collision, where both kinetic energy and momentum are conserved. When one object is much more massive than the other, $m_2 >> m_1$, the two solutions to are

$$v_1 = u_1 \tag{14}$$
$$v_2 = u_2 \tag{15}$$

and

$$v_1 = 2u_2 - u_1 \tag{16}$$
$$v_2 = u_2 \tag{17}$$

Equation 14 and 15 tells us that nothing changes and that the two particles continue on their paths with the same velocity. As a gravity assist involves a change in velocity, the first solution does not describe a gravity assist. The second solution, however, does. Equation 16 tells us that one of the particles changes direction, due to the $-u_1$ term. As there is a change of direction, this implies that there needs to be a minimum of two dimensions for a gravity assist to be possible. This is investigated in section 3.3, where equation 8 is solved only for x. At $R = 0$ the differential equation equals infinity, so the equation cannot be solved for very small R. To get around this problem we instead use a Gaussian potential in the form

$$\frac{d^2x}{dt^2} = 2(x' - x)AGM \cdot \exp\left(-(x' - x)^2\right), \tag{18}$$

where A is a constant. At $R = 0$, the equation simply equals zero, and there are no infinities.

The maximum possible gravity assist $\Delta v = 11.3$ km$\cdot$s^{-1}. However, a gravity assist is not the head-on collision between the satellite and Jupiter, and so the actual assist will not be as large as this. The head-on collision is investigated in section 3.6.

2.5 Escape velocity

To overcome the gravitational potential of the Sun ($M_\odot \approx M_{\text{Solar System}}$ [6]) the velocity

$$v = \sqrt{v_x^2 + v_y^2} \tag{19}$$

at a distance R away from the Sun must be greater than the escape velocity, given by

$$v_{\text{esc}} = \sqrt{\frac{2GM_\odot}{R}}. \tag{20}$$

3 Methods

3.1 Simulating Jupiter's launch position

Simulations were performed using Python. To plot the trajectories of Earth and Jupiter we use equations 3 and 4 to find the position at each time, with t being an array of times between $t = 0$

and $t = T_J$. The larger the size of the array, the smaller the spacings will be, but the longer the code will take to run. A size of 10^5 suffices for sufficiently accurate simulations. A variable defined as 'launch position' represents going a time $\frac{1}{2}T_{sat}$, such that t[launch position] $\approx \frac{1}{2}T_{sat}$. This is used to find the position of Jupiter at launch time, by going back from its final position by a time t[launch position]. From there we find the distance between the Earth and Jupiter using equation 9, as well as the corresponding angle between them.

3.2 Simulating the transfer orbit

The transfer orbit can be simulated in two ways, via plotting the position of the satellite at multiple time intervals, or via solving equation 8. For the first method, similar to section 3.1, we need to find the position of the satellite at every time, using equations 3, 4, and 7, the accuracy of which is determined by the spacing between time intervals.

To solve the transfer orbit numerically, an ordinary differential equation (ODE) solver is required. The package used is *scipy.integrate.odeint* [7]. The package is only able to solve first-order equations, so each of our second order ODEs must be split into two first-order ODEs.

$$\dot{\lambda}(t) = v_\lambda(t),$$
$$\dot{v}_\lambda(t) = \frac{GM'(\lambda' - \lambda)}{R^3};$$

For one body, these equations are defined inside a function which takes parameters Z (a 4-dimensional array containing x, y, v_x, v_y) time, G and $M = M_\odot$. Initial conditions for Z at launch time must be stated, which we already know from sections 2.1 and 2.3. To simultaneously do this for both the satellite and Jupiter, Z becomes an 8-dimensional array, with variables for both bodies defined as they would normally be. The package then integrates the equations to return the definite integrals for the objects in motion.

3.3 Simulating a one-dimensional gravity assist

Using equation 18 and the ODE solver in section 3.2 we can obtain four different solutions based on different initial conditions. The first two conditions simulate what would happen in an attractive potential to a stationary or approaching particle in one dimension. The second two conditions simulate what would happen in a repulsive potential to a stationary or approaching particle.

3.4 Simulating the gravitational potential of Jupiter

To incorporate the gravitational potential of Jupiter one needs to utilise equation 10 and create a function with two mass variables instead of one: $M_\odot$ and M_J. The ODEs for Jupiter remain unchanged, whereas the ODEs for the satellite have an additional term corresponding to the field strength of Jupiter:

$$\dot{v}_\lambda(t) = \frac{GM_\odot(-\lambda)}{R_1^3} + \frac{GM_J(\lambda'_J - \lambda)}{R_2^3}.$$

The solution to this function will automatically give rise to a gravity assist, due to the additional potential.

3.5 Optimising the gravity assist

To increase the acceleration from the assist we can change the initial velocity and position of the satellite. Multiple ODEs are simultaneously solved to see which conditions give rise to a successful gravity assist, where successful means the probe will leave the solar system.

An array of different additional velocities between -50 m$\cdot$s^{-1} and 50 m$\cdot$s^{-1} were added to the initial velocity of the satellite by employing a list of conditions to be solved using a 'for loop'. The absolute velocities are calculated using equation 19. At any point, after the gravity assist has taken place, the velocity of the satellite minus the escape velocity reveals the fate of the satellite. The difference in velocity is either negative, signifying the gravity assist was not strong enough to expel the satellite, or positive, signifying a successful gravity assist. The largest positive velocity difference represents the velocity to launch the satellite to achieve the greatest acceleration.

To change the initial position we need to account for the fact that a change in position results in a change of velocity components. To change the initial conditions we use equations 3, 4, 5, and 6. An array of times between $-T_E$ and T_E is created, corresponding to different planetary positions. In the same way as before, the velocity of the satellite at a point after the gravity assist is compared with the escape velocity, to determine the best position to launch a satellite for the maximum gravity assist.

3.6 Simulating a head-on collision between the satellite and Jupiter

We can simulate the required velocity needed for the satellite to launch backwards and collide head-on with Jupiter in a similar way to how we found the optimum initial velocity in section 3.5. There is only a very small window where we would be able to see a collision so a little bit of trial and error is needed to do this, but we know that the velocity needs to be much greater (more negative) than -40 km$\cdot$s^{-1} to counteract the velocity of the Earth.

Each of the different velocities which result in a successful gravity assist would send the satellite off on different trajectories, based on the angle the satellite collides with Jupiter. This can be shown by plotting the trajectory of the satellite for a time $T >> \frac{1}{2}T_{\text{sat}}$.

3.7 Simulating a gravity assist with Mars

We can simulate the effect of launching a satellite to Mars instead of Jupiter in much the same way, by simply changing the mass and distances in the functions used in sections 3.1 and 3.4. Analysing different initial velocities, described in section 3.5 we can determine which starting conditions are suitable for a successful gravity assist, using Mars.

4 Results and Analysis

4.1 Launch position

The distance between Jupiter and Earth at the launch position was found to be 8.101×10^{11} m. This distance corresponds to two different positions, one which can be seen in Figure 2, and one which is symmetric in the y-axis. Jupiter needs to be positioned ahead of Earth, as in Figure 1, at an angle of 97.2°, which is in line with the expectations that Jupiter needs to be just over 90° from Earth.

4.2 Transfer orbit

The transfer orbit solved via the time interval method can be seen in Figure 2. As we can see, it looks and behaves exactly as is expected, intercepting Jupiter after exactly half an orbit (2.73 years). Solving the ODEs for Jupiter and the satellite simultaneously gives the same result, proving that the ODE solver works, and can be used to simulate the gravitational potential described in section 3.4.

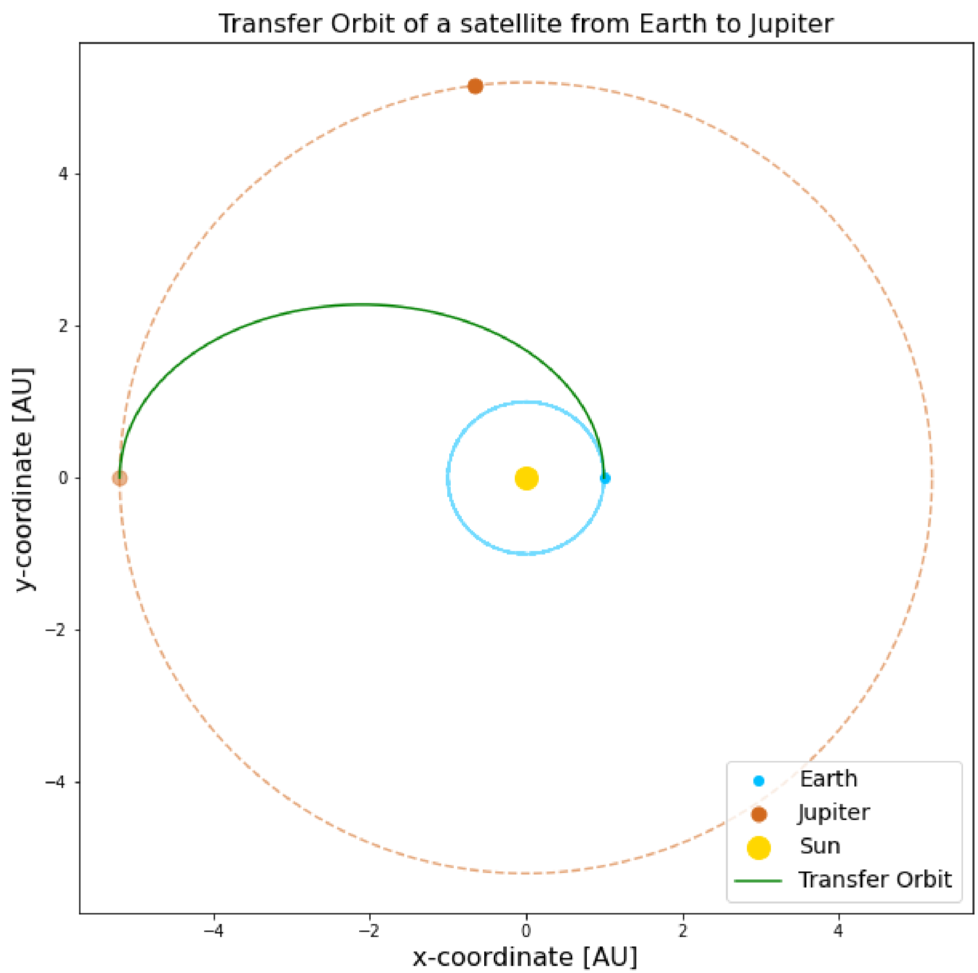

Figure 2: Simulated transfer orbit from Earth to Jupiter using the time interval method.

4.3 The one-dimensional potential

Figure 3 shows how an initially stationary satellite is attracted towards Jupiter, passing through $R = 0$ with maximum velocity, before decelerating and falling back into the potential, oscillating between its initial position and -initial position. Figure 4 shows how the approaching satellite is accelerated towards the potential, again with a maximum velocity at the centre, before decelerating back down to its initial velocity and continuing on its way. This is not a gravity assist as the final velocity is equal to the initial velocity. This proves that a one-dimensional gravity assist is not possible.

If we instead make it such that the potential is repulsive, we see from Figure 5 that a stationary satellite is accelerated away from the potential until it escapes the reach of the potential, where it continues with a much greater velocity than it started with. In Figure 6 we see that an approaching satellite's velocity decreases until it reaches a turning point, and is accelerated in the other direction, with a final velocity greater than the initial one (but in the opposite direction). Therefore we can conclude that, if gravity was a repulsive force a gravity assist would be possible in one dimension.

4.4 Initial gravity assist

Figure 7 shows that a gravity assist occurs from solving the ODEs when Jupiter's potential is included. The larger velocity results in a larger orbital radius, as expected from equations 5 and 6. The assist does not however give the satellite enough velocity to escape the Solar System, as we can see that the satellite proceeds to fall back into orbit.

4.5 Optimised gravity assist

The velocities which result in a successful gravity assist can be seen in Figure 8. The large gap in the data is because certain velocities caused the satellite to approach too close to Jupiter, and resulted in infinities trying to be solved, which produces errors in the code. As such, these troublesome values were omitted from the list of initial conditions. The graph shows that any initial additional velocity between $-12.5 \leq v_{\text{add}} \leq -7$ m·s^{-1} results in a successful gravity assist, with the greatest acceleration occurring with an initial additional velocity of -9.75 m·s^{-1}.

Changing the initial position does not result in a gravity assist large enough to expel the satellite out of the Solar System, as we can see in Figure 9. It does allow us to see that the best time to launch to Jupiter is 24 days before the previously calculated launch time.

One aspect which was not fully explored was combining the optimum position with changing the initial velocity, to find the overall most efficient transfer and gravity assist.

4.6 Head-on collision with Jupiter

We found that the satellite would need to be launched with an additional initial velocity in the region of $-77,070$m·s^{-1} to collide with Jupiter and rebound out of the Solar System. Slight deviations in the velocity resulted in different gravity assists and trajectories, which can be seen in Figure 10. The largest assist occurred with an initial additional velocity of $77,070.5$ m·s^{-1}. The acceleration obtained via the head-on collision was far greater than from the original gravity assist, which was to be expected from our modelling of elastic collisions. However too much energy is required to thrust the rocket to a velocity of $77,000$ m·s^{-1}, and so a head-on gravity assist would never be favourable.

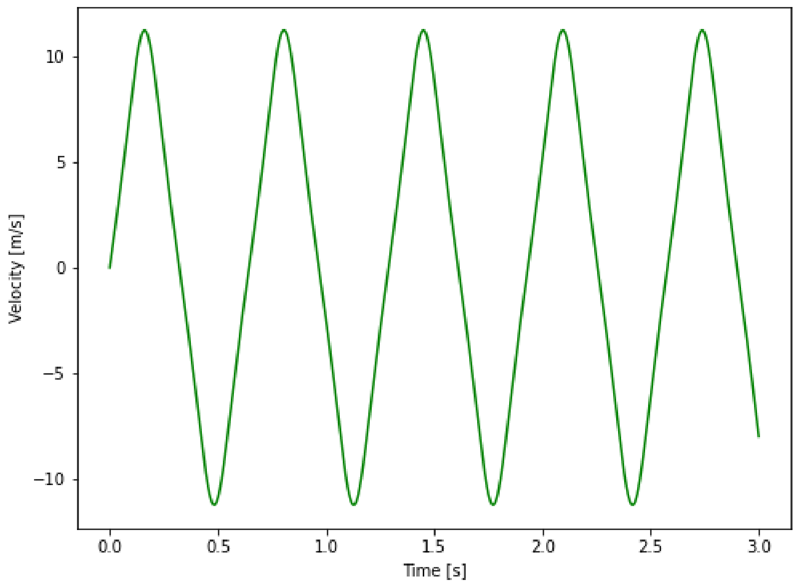

Figure 3: Simulation of how the velocity of a stationary particle behaves in an attractive potential in only one dimension.

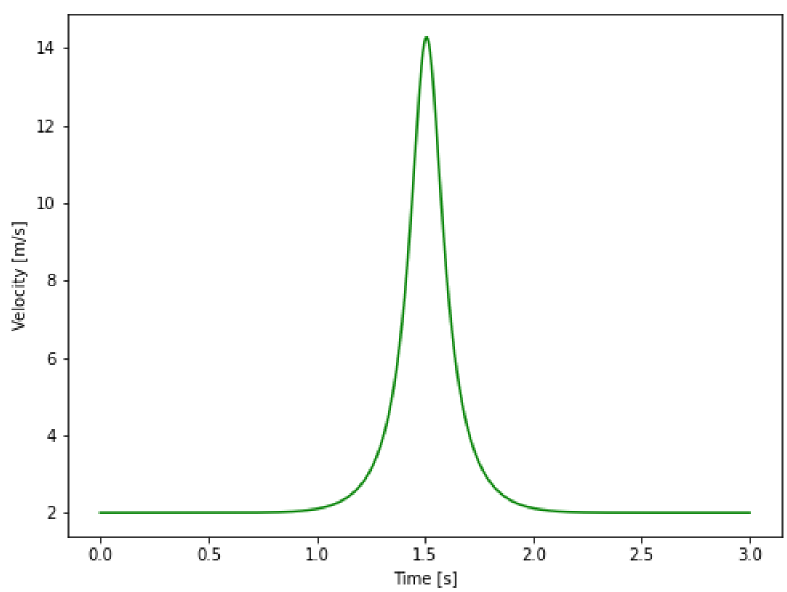

Figure 4: Simulation of how the velocity of an approaching particle behaves in an attractive potential in only one dimension.

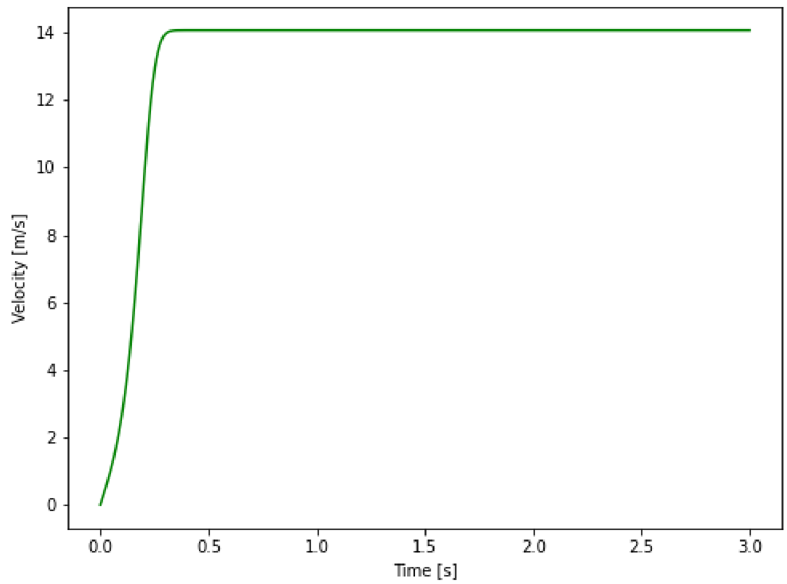

Figure 5: Simulation of how the velocity of a stationary particle behaves in a repulsive potential in only one dimension.

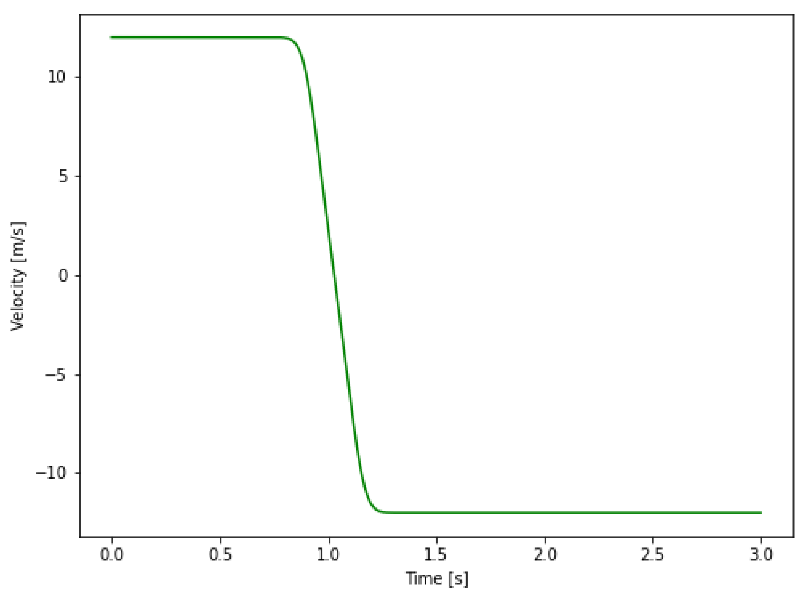

Figure 6: Simulation of how the velocity of an approaching particle behaves in a repulsive potential in only one dimension.

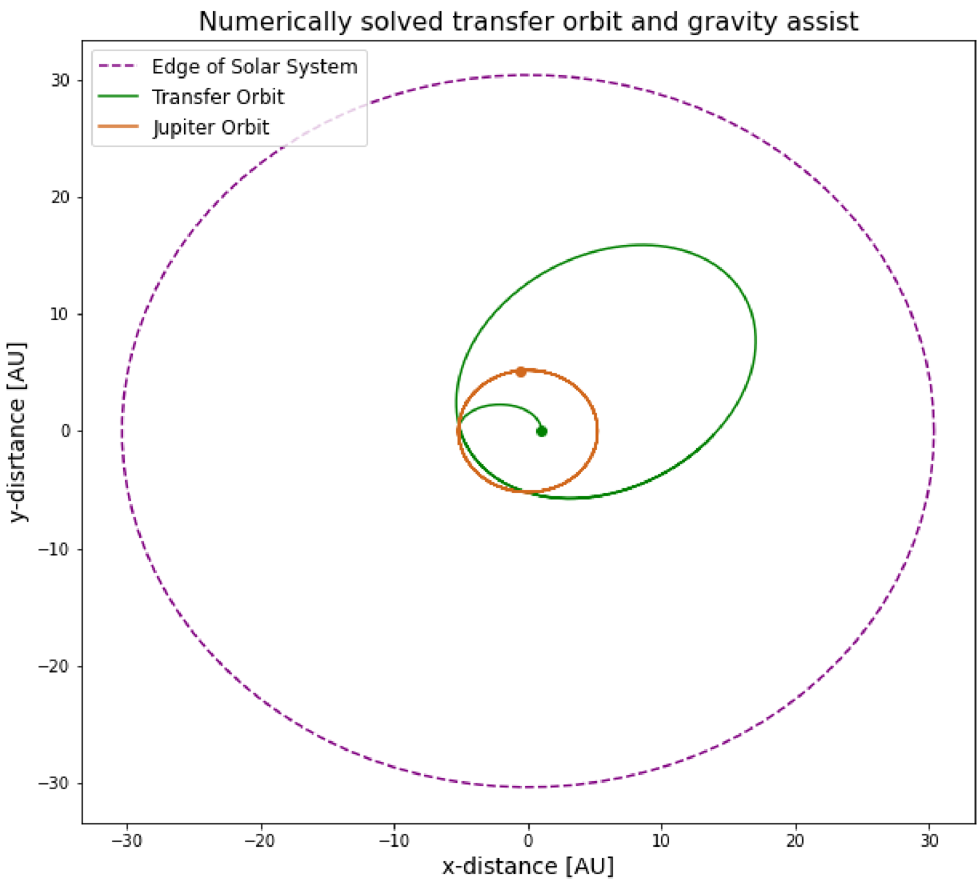

Figure 7: Simulation of how the satellite orbit changes due to the unaltered gravity assist from Jupiter. Simulation run for $t = 10T_{\rm sat}$.

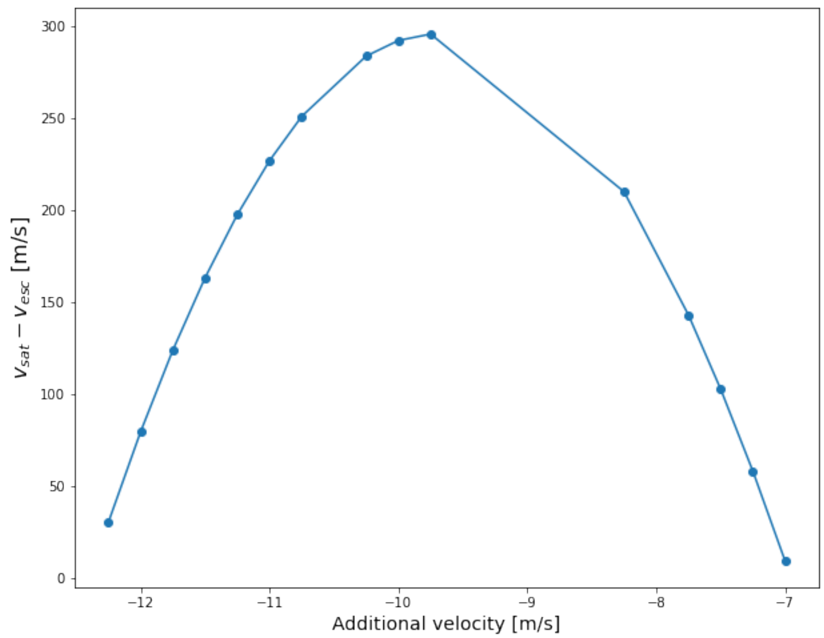

Figure 8: Graph showing which initial additional velocities give rise to a post gravity assist velocity greater than the escape velocity.

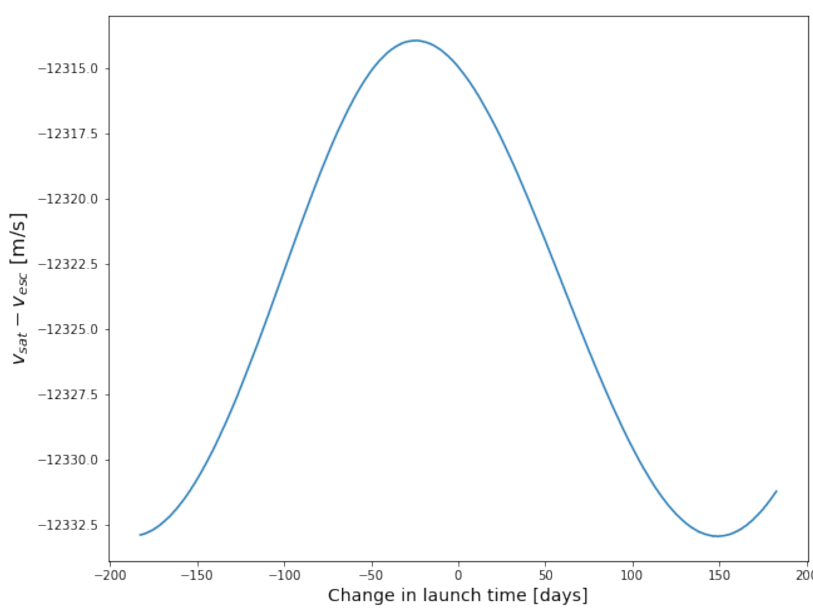

Figure 9: Difference in satellite velocity and escape velocity for different launch times before or after the initially calculated launch time.

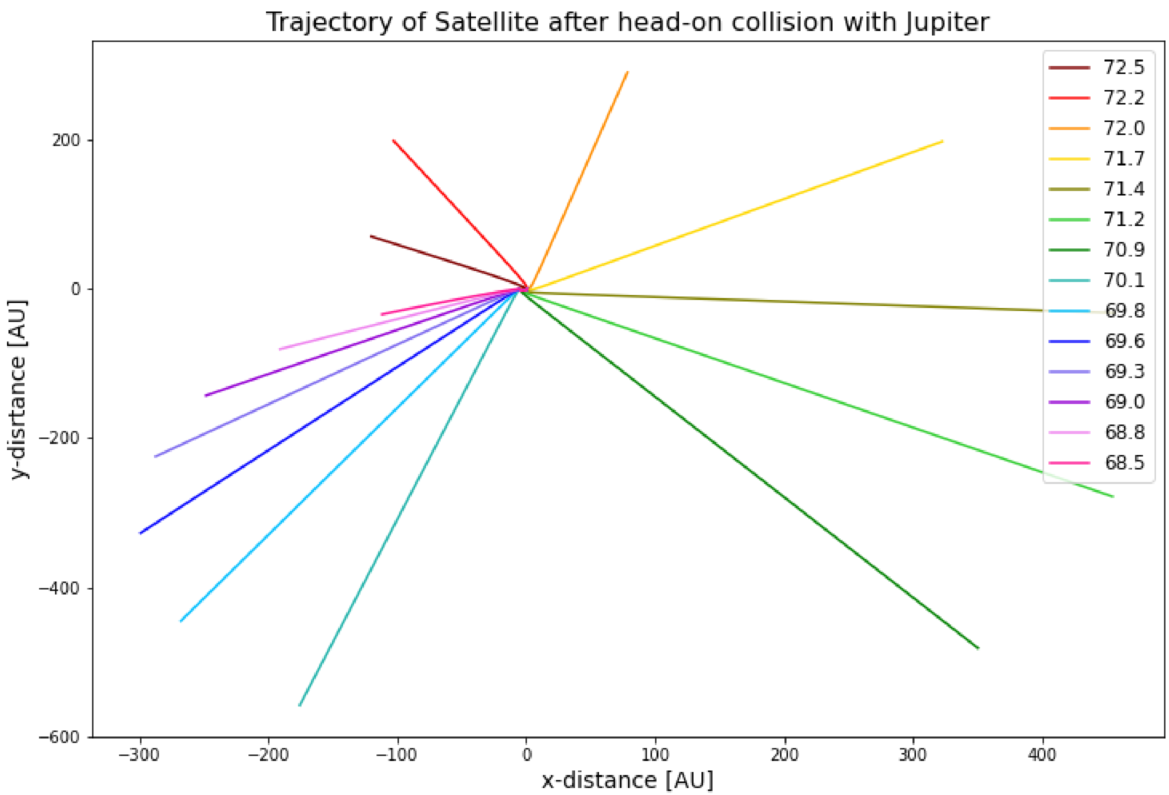

Figure 10: Trajectory of the satellite after a head on collision with Jupiter, with different initial additional velocities. The legend indicates the last three significant figures of the velocity (i.e. $77,072.5$ m$\cdot$s^{-1}). The simulation was left to run for a time $t = 20T_{\text{sat}}$.

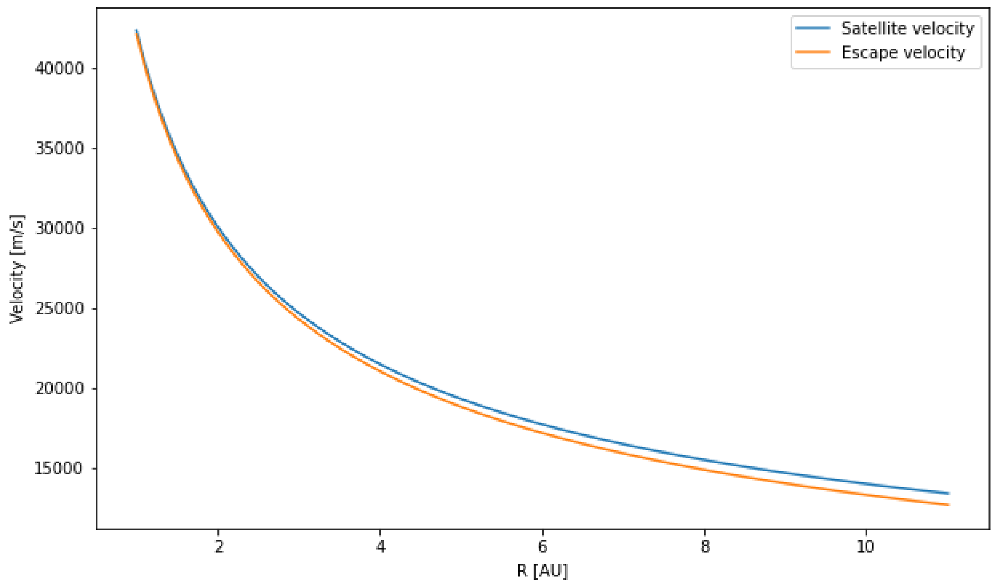

Figure 11: Graph showing the velocity of the satellite and escape velocity as a function of R. We see that the satellite always has $v > v_{\text{esc}}$.

4.7 Can we escape the Solar System using Mars?

The velocities which give rise to a gravity assist with Mars are all insufficient to launch the satellite out of the Solar System. The only velocities which result in the satellite being expelled are velocities above 9380 m$\cdot$s^{-1}. If we look at Figure 11 we can see that no gravity assist even occurs at this velocity, the satellite begins with a velocity greater than the escape velocity, and will always escape. When the simulation was run without Mars, the satellite still escapes, proving this point. Therefore a gravity assist is not possible using Mars alone.

5 Conclusion

The code was successfully able to determine the correct time to launch a satellite to another planet, using the minimum energy possible. We were able to simulate the transfer orbit using a time interval to track the position at every time, and using an ODE solver to integrate the equations of motion, which proved to be a far more convenient method, as we could easily incorporate the gravitational potential of Jupiter as well as the Sun, and see the resulting gravity assist that occurs. Both the launch position and launch velocity were optimised to provide the maximum gravity assist possible. Further work could be done to optimise the two simultaneously, which would require changing both x and y components of initial velocity, or redefining the grid such that the new optimum launch position sits at $(R_E, 0)$. A one-dimensional gravity assist was proven to be impossible unless gravity was a repulsive force. We also showed that the maximum gravity assist occurs when the two bodies collide head-on, and we simulated the different trajectories that resulted from this head-on collision. It was concluded that the gravitational field of Mars was too weak to provide a sufficient gravity assist to launch a satellite out of the Solar System.

References

[1] Cox B, Cohen A. Wonders of the Solar System. BBC. (2010).

[2] Satellites beyond Jupiter. University of Birmingham: unpublished handout. (2021).

[3] Widnall S, Peraire J. Orbit Transfers and Interplanetary Trajectories. MIT. (2008).

[4] Winn JN. Transits and Occultations. (2014). arXiv:1001.2010

[5] Njeunje J, Sewwandi de Silva D. The Equations of Planetary Motion and their Numerical Solution. (2018).

[6] Weidenschilling SJ. The distribution of mass in the planetary system and solar nebula. Astrophys Space Sci 51, 153–158. (1977). doi.org/10.1007/BF00642464

[7] scipy.integrate.odeint. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html. [Accessed Dec 2021].

A Appendix: Derivations

A.1 Orbital period

$$F_{Gravity} = F_{Centripetal}$$
$$\frac{GMm}{r^2} = \frac{mv^2}{r}$$
$$v = \sqrt{\frac{GM}{r}}$$

Using the knowledge that, for circular motion, $v = \frac{2\pi r}{T}$

$$T = \frac{2\pi r^{3/2}}{\sqrt{GM}}$$

A.2 Second order equations of motion

In 2D, the components of force acting on the satellite is

$$F_\lambda = F\frac{\lambda' - \lambda}{R}$$

From Newton's second law we know that

$$\begin{aligned} F &= ma \\ &= m\frac{dv}{dt} \end{aligned}$$

Therefore, we obtain the equation

$$F_\lambda = m\frac{d^2\lambda}{dt^2}$$

Combining the two equations we arrive at

$$\frac{d^2\lambda}{dt^2} = GM'\frac{\lambda' - \lambda}{R^3}$$

A.3 Impulse

The velocity of the Earth and Jupiter are constant due to our circular motion approximation, given by

$$v_E = \sqrt{\frac{\mu}{R_E}}$$
$$v_J = \sqrt{\frac{\mu}{R_J}}$$

The velocities of the transfer orbits are determined by using the conservation of energy.

$$\frac{1}{2}mv_\pi^2 + \frac{\mu m}{R_E} = \frac{1}{2}mv_\alpha^2 + \frac{\mu m}{R_J}$$

Therefore we obtain the two solutions

$$v_\pi = \sqrt{\mu\left(\frac{2}{R_E} - \frac{1}{a}\right)}$$
$$v_\alpha = \sqrt{\mu\left(\frac{2}{R_J} - \frac{1}{a}\right)}$$

And as

$$\Delta v_1 = v_\pi - v_E$$
$$\Delta v_2 = v_J - v_\alpha$$

we obtain the equations given in equation 12 and 13

A.4 Elastic collision

Kinetic energy and momentum are conserved.

$$\frac{1}{2}m_1u_1 + \frac{1}{2}m_2u_2 = \frac{1}{2}m_1v_1 + \frac{1}{2}m_2v_2$$
$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2$$

These equations can be simultaneously solved:

$$m_1(v_1^2 - u_1^2) = m_2(u_2^2 - v_2^2)$$
$$m_1(v_1 - u_1) = m_2(u_2 - v_2)$$

When $m_2 >> m_1$, the two solutions are

$$v_1 = u_1$$
$$v_2 = u_2$$

and

$$v_1 = 2u_2 - u_1$$
$$v_2 = u_2$$

A.5 Escape velocity

For an object to escape from the gravitational pull of a massive body, the sum of its kinetic and potential energies must equal zero.

$$\frac{1}{2}mv_{esc}^2 + \frac{-GMm}{r} = 0$$

$$v_{esc} = \sqrt{\frac{2GM}{r}}$$