Orbital Mechanics of Gravitational Slingshots

Final Paper
15-424: Foundations of Cyber-Physical Systems
Adam Moran, amoran@andrew.cmu.edu
John Mann, jmannjr@andrew.cmu.edu

May 1, 2016

Abstract

A gravitational slingshot is a maneuver to save fuel by using the gravity of a planet to accelerate or decelerate a spacecraft. Due to the large distances and high speeds involved, slingshots require precise accuracy to accomplish — the slightest mistake could cause the whole mission to fail. Therefore, we have developed a cyber-physical system to model the physics and prove the safety and efficiency of powered and unpowered gravitational slingshots. We present our findings and proof in this paper.

1 Introduction

A **gravitational slingshot** is a maneuver performed to increase or decrease the speed of a spacecraft by simply approaching planetary bodies. A spacecraft's usefulness and maneuverability is basically tied to the amount of fuel it can carry, and the more fuel a spacecraft holds, the more fuel it needs to carry that fuel into orbit. Therefore, gravitational slingshots are a very appealing way to save mass, and therefore money, on deep-space missions since these maneuvers do not require any fuel. As missions conducted by national and private space programs become more frequent and ambitious, the need for these precise maneuvers will increase. Therefore, we have created a cyber-physical system that models the physics of a gravitational slingshot for a spacecraft approaching a planet.

In the "Approach" section of this paper, we give a brief overview of the physics involved with orbits and gravitational slingshots.

In the "Models and Properties" section of this paper, we describe what assumptions and simplifications we made to model these astrophysics in a way for us to prove our desired properties with KeYmaeraX.

2 Approach

This section describes the physics of gravitational slingshots, including concepts like orbital trajectory, escape velocity, and gravitational energy transfers.

2.1 Astrodynamics

Consider a spacecraft positioned 300 kilometers above the surface of Earth with no velocity. This spacecraft, since it's not moving, would begin to fall downward towards the surface of the planet. If the spacecraft were given a small forward motion, say ten meters per second, it would still land (or crash) on the surface of the planet, but it will do so some amount beyond the point where it would wind up if it had started completely stationary — call this distance d. The higher the initial speed of the spacecraft, the larger d becomes.

Now, consider what happens when d is larger than the radius of the planet. When this happens, the spacecraft will continue to fall towards the planet *forever* (unless outside forces act upon it). This perpetual path of motion is an **orbit**. An orbit is defined by several parameters, but the only parameter important to us is the semi-major axis. An orbit's semi-major axis is half the maximum diameter of the orbit's ellipse. Because the orbits we'll model in `slingshot.kyx` will be perfectly circular, we'll refer to the orbit's **radius** r_{orbit} instead of semi-major axis a_{orbit}.[1]

The linear velocity of the spacecraft v_{sc} needed to outrun the planet's gravity to attain a circular orbit can be calculated from the following formula (*Orbital Mechanics*, n.d.):

$$v_{sc} = \sqrt{GM\left(\frac{2}{r_{sc}} - \frac{1}{r_{orbit}}\right)}$$

G is Isaac Newton's universal gravitation constant, r_{sc} is the distance between the spacecraft and the planet, and M is the mass of the orbited planet. Because r_{sc} is always equal to r_{orbit} for circular orbits, we can simplify the above equation to Equation 1:

$$v_{sc} = \sqrt{\frac{GM}{r_{orbit}}} \tag{1}$$

If a spacecraft is to break out of orbit with a planet, it must reach a speed where it can totally outrun the gravity of the nearby planet. This speed is called **escape velocity**. It is calculated with the following formula:

$$v_e = \sqrt{\frac{2GM}{r_{orbit}}} \tag{2}$$

2.2 Gravitational Slingshot

A planet's gravity can be used to redirect or even speed up a spacecraft.

By approaching a planetary body from an angle other than the direction the body is moving, a change in velocity occurs due to an energy transfer effected by gravitational fields. Gravitational slingshots are very useful because they can change velocity without requiring any fuel.

In the simplest case, a craft approaches a planet at 180° difference between their velocities. The craft can enter an orbit around the planet, gain some speed and exit the orbit with a higher velocity than when it entered. Specifically, it will leave the planet with it's original velocity plus the planet's velocity *relative* to the planet (thus resulting in a velocity of twice the planet's plus original). This simple case is illustrated below.

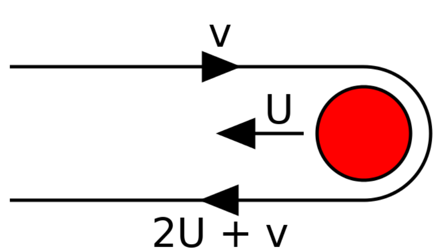

Fig 1: Gravitational Slingshot Speed Changes (*Gravity assist*, n.d.)

[1] Would you like to know more? Read about orbits in the rocket science primer found in Appendix C.

In a more general case, we can observe a change in velocity based off the angle between the incoming and outgoing trajectories. This obeys the following physical formula:

$$\Delta v = 2|v_{planet}|cos(\theta)$$

These values are all from the planet's perspective. As we can see, a sharper angle yields a bigger boost up to the 180° mark. Gravitational assists are used to give the craft extra kinetic energy to reach it's destination. This was done using partial slingshots with the *Voyager* probes sent to the outer Solar System.

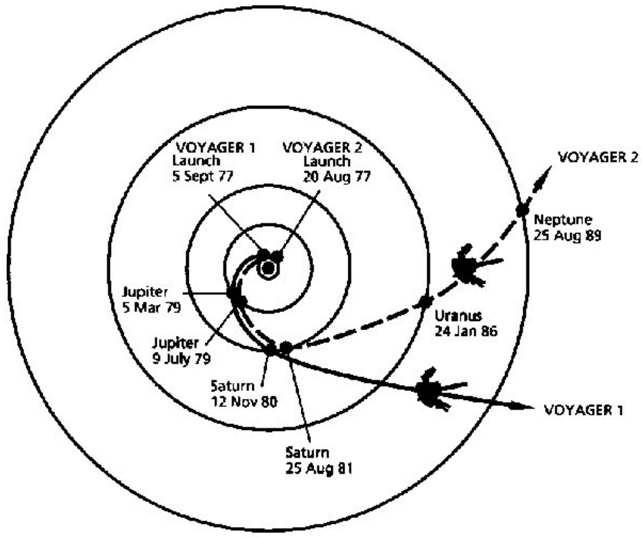

Fig 2: *Voyager* Trajectories (*Voyager program*, n.d.)

2.3 Powered Slingshot

In the 2014 Christopher Nolan science-fiction film *Interstellar*, the crew of the *Endurance* was able to travel from Dr. Mann's Planet to Edmund's Planet by performing a **powered gravitational slingshot** or **Oberth maneuver** around the black hole Gargantua.

A powered gravitational slingshot is when a spacecraft runs its engines while encountering a planet. Running engines not only increases the velocity of a spacecraft, but also burns fuel, which decreases mass. Therefore, a powered gravitational slingshot can significantly increase the effectiveness of a flyby.

If the spacecraft begins the burn when travelling at v_{sc} and burns the engines enough to change its velocity Δv, then the change in orbital energy is:

$$\Delta E_{orbit} = v\Delta v + \frac{1}{2}(\Delta v)^2$$

The closer the spacecraft is to the body, the faster its orbital velocity is, and therefore, the burn becomes more effective (*Oberth effect*, n.d.).

3 Models and Properties

Our model contains a spacecraft moving around the target planet while behaving according to orbital physics. The model is a hybrid system that combines discrete decisions with continuous evolutions of physics. This model simplifies live systems in various ways detailed in the below 'Assumptions and Limitations' section. Although this model simplifies various properties, we believe it creates a representative system that could be expanded to model all appropriate physics with the right tools.

There are 3 main components to our model: control assignments, differential equations, and loop invariants and safety/efficiency conditions.

- Within control assignments, the thrust of the craft is assigned within various scope considerations. After the thrust is assigned, the differential equations are executed.
- The differential equations include those for accessing sine and cosine functions, changing velocity based on thrust and cosine, and changing theta (the current angle traveled) based on our current velocity.
- There are 2 main conditions that should remain true after each execution of the loop (the loop invariants). The first states that if the craft has not reached the desired exit angle (`slingtheta`), our velocity should be smaller than escape velocity. **This is the crux of what we are proving with our model.** The second is that the craft orbit radius should be sufficiently above the surface of the planet (150km is used in our model).

The figure below shows how the craft enters the gravitational influence of the planet, travels around it until hitting slingtheta where it exits the gravitational influence with a velocity increase courtesy of the planet.

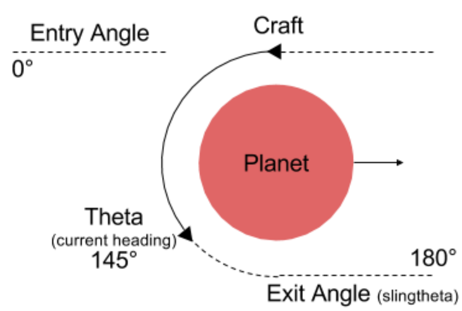

Fig 3: Slingshot Angles Diagram

This next figure shows how the orbit radius and planet radius are related.

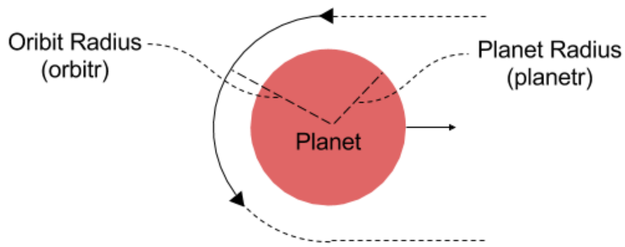

Fig 4: Slingshot Orbit Diagram

3.1 Design Choices

3.1.1 The Spacecraft

In our model, we use polar coordinates to express the position of the spacecraft.

orbitr is the distance between the spacecraft and the planet's center of mass in meters.

We define theta (or θ) to be the current angle between traveled since entering the slingshot at $\theta = 0$. The planet is assumed to be moving opposite the direction of entry into the slingshot (this achieves maximum slingshot effect). The spacecraft exits the slingshot at the exit angle, dubbed *slingtheta*. The differential equation for θ follows the circular motion equations for angular velocity. Just as a distance differential would be velocity, the θ differential equation involves angular velocity. Angular velocity is related to both linear velocity and radius of the referenced circle, `orbitr` (*Circular motion*, n.d.).

$$v = \omega * r$$
$$\omega = \frac{v}{orbitr}$$
$$\theta' = \frac{v}{orbitr}$$

Thus, as the craft moves faster, it completes the angle theta quicker and in turn the slingshot quicker.

3.1.2 The Planet

Planets orbiting stars have motion too. However, in our model, we decided to center our frame at the planet, effectively ignoring the planet's position. This allows us to simplify the model but lacks the descriptiveness the full physical model would entail. The planet that the craft slingshots around has radius *orbitr* (measured in kilometers). If *orbitr* is ever less than *planetr*, then our spacecraft has crashed onto the surface of the planet. Because most planets in the Solar System have atmospheres, we require the following safety condition to hold throughout the maneuver:

$$planetr + h_{atmosphere} \leq orbitr \tag{3}$$

In our model, $h_{atmosphere}$ is equal to 150 kilometers.

3.2 Thrust Control Assignment

Thrust is assigned non-deterministically, meaning thrust could be assigned to be anything that satisfies the test following that assignment. This is important because although thrust should not be fixed to a single value, it needs to fall within some physical constraints. In addition, we must choose our thrust such that our scaling constant, x, is valid. The use and definition of this scaling constant is discussed in section 3.3.

3.2.1 Physics of Thrusting

We do not want the thrust of the craft to cause it to escape orbit. Therefore, we want our thrust to be less than our escape velocity minus current velocity. Note that thrust is represented in units of acceleration (kilometers per square second), **not** units of force.

$$
\begin{aligned}
thrust &< \left(\frac{2GM}{r_{orbit}}\right)^{1/2} - v \\
\Longrightarrow thrust + v &< \left(\frac{2GM}{r_{orbit}}\right)^{1/2} \\
\Longrightarrow (thrust + v)^2 &< \left(\frac{2GM}{r_{orbit}}\right)
\end{aligned}
$$

This final condition becomes part of our thrust test to ensure our thrust is not too powerful to escape orbit. We've consciously chosen to not model when the test fails. The test is a statement that we've chosen not to model space physics outside of orbit. All of our differential equations rely on being in the gravity influence of the planet. In order to model when the test fails (that is, thrust puts us over escape velocity), we would need an entirely different set of physics that would be hard or impossible to model in KeYmaeraX.

3.3 Scale Factor Considerations

To use cosine and sine within KeYmaeraX, we utilize the concept of differential ghosts. Two differential equations are of interest:

```
c' = -s
s' = c
```

As the differential equations are executed repeatedly, c becomes a cosine wave and s becomes a sine wave. This is important as we look back to how velocity is impacted by the slingshot maneuver.

$$\Delta V = 2 * planetv * cos(\theta)$$

Although, as our system evolves, it produces $cos(t)$, not $cos(\theta)$. Ideally, we would manipulate the period of cosine to fit our needs, however, with the current version of KeYmaeraX, this is not possible. As a simplification, we chose to scale the other components of velocity (i.e. from thrust) to bring velocity in line with our use of $cos(t)$. To explain this more clearly, we can examine the velocity differential equation from our model:

$$v' = x * thrust + c \tag{4}$$

In this equation, c is cosine and thrust is in terms of acceleration (km/s^2) as noted above. The equation is derived from the idea that change in velocity should equal thrust plus deltaV due to the gravity slingshot:

$$v' = thrust + 2 * planetv * cos(\theta)$$

We've chosen to instead use $cos(t)$ (or 'c') in place of $2(planetv * cos(\theta))$. To compensate, we've scaled velocity to simulate stretching the period of cosine and multiplying by the various constants. In turn, we've limited the domain of c to be $-1 < c \leq 0$. This will prevent our cosine function from completing the wave (which would correspond to traveling 360deg around the planet. The maximum slingshot angle is 180 deg which is what we've limited the cosine wave to.

Since we've scaled thrust by `x` for the above reasons, we need to mathematically define `x`. Clearly it must be in terms of `c`, the cosine, and `thrust` the control assignment. In fact, since `v` must be greater than 0, we can analyze Equation 4 as follows:

$$
\begin{aligned}
& 0 < x * thrust + c \\
\implies & -c < x * thrust \\
\implies & c > x * thrust \\
\implies & x < \frac{c}{thrust}
\end{aligned}
$$

We use both this final property and the square of that property (for negation reasons) in the test for thrust to ensure thrust satisfies those properties. This is inherently better than having these conditions in the domain constraint since it is more appropriate to chose a thrust that satisfies our proof properties than to just assume the domain excludes thrusts that don't.

3.4 Proof Techniques

In the proof, we use a loop invariant to show which properties will be true after each iteration of the loop. The properties of the loop invariant are:

1. $(theta \leq slingtheta) \implies (v^2 \leq ((2 * G * planetm)/orbitr))$
2. $orbitr > 0$
3. $thrust > 0$
4. $c^2 + s^2 = 1$
5. $planetr + 150 \leq orbitr$

The final safety condition is the same as the first loop invariant. This means that proving the loop invariant will directly imply the post conditions. The loop invariant conditions follow from the control decision tests (for thrust) and the properties of the continuous evolution of the hybrid program.

Invariants 2 and 5 hold by the precondition and the fact that they are constant. Invariant 3 holds by the precondition and the test for thrust ensuring that it is positive. Invariant 4 is true by the evolution properties of cosine and sine. Invariant 1 is true by the thrust test and preconditions. The most crucial part of the proof is the differential invariant cut. This cut in the following properties: $c^2 + s^2 = 1$ & $planetr + 150 \leq orbitr$ & $v^2 \leq (2 * G * planetm)/orbitr$ & $thrust > 0$. Thus, we've been able to prove that while we haven't reached our desired slingshot angle, we are still within orbit (our primary proof goal).

As theta increases, our velocity increases due to thrust or simply the gravity of the slingshot itself. This in turn causes theta to increase even more and the slingshot is underway. It is very important that the proof guarantee that when the craft hasn't exited the slingshot, it is going slow enough to maintain it's position within the gravity influence of the planet. A future (stronger) proof would state that precisely when theta equals slingtheta, we've escaped the gravity slingshot and our speed has been incremented twice that of the planet's velocity plus our initial velocity plus any velocity gained from thrust. It's important to note that this is a great first step and is still meaningful for modeling gravitational slingshots.

Note: The described hybrid system and proof can be found in files slingshot.kyx and slingshot.kyt.

3.5 Assumptions and Limitations

In order to come to a middle-ground on a system so complex that we couldn't prove it and one so simple it's not realistic, we made some assumptions and compromises.

First, we abstracted certain properties of our spacecraft away. We assumed that its engines were capable of producing any speed needed. Furthermore, we assumed that the engines had unlimited fuel, meaning that not only were they able to run forever, but also never decreased the mass of the spacecraft.

Secondly, we had initially wanted to have `orbitr` change with `thrust`. When burning your engines in orbit, if you thrust straight forward (in the direction of travel), your orbit radius will get larger. However, you can compensate for this by thrusting towards the planet which reduces your orbit radius. It became very hard to prove key points about velocity in our model when the orbit radius was also changing. This was due to the escape velocity equation:

$$v^2 \le \frac{2 * G * M}{r_{orbit}}$$

When neither `v` nor `orbitr` were constant, the proof became extraordinarily more complex. We chose to make a simplification: thrust would assumed to be applied in a direction that would simultaneously propel forward and inward (towards the planet) so as to keep `orbitr` constant. Although this often is not fuel efficient in real space travel, it is not impossible to do. Thus, we believe it is a fair simplification to our model.

Thirdly, working with cosines creates scenarios that are hard to prove. As they are not in first-order real arithmetic, KeYmaeraX provides less help in solving them. We required the use of cosine in our velocity differential since the gravity slingshot mechanics require them. We originally tried to eliminate this cosine term by digging into where the slingshot equation, $\Delta V = 2 * planetv * cos(\theta)$, came from. Unfortunately, it has its roots in conservation of energy equations. Since these equations look at the energy of the system before the event (slingshot) and after, there are no useful continuous evolution equations we can use for our differential equations. So, we had to use the cosine function. To do this, we took tricks from previous labs where circular dynamics were in play. This afforded us the use of `cos(t)`. We needed `cos(theta)`, so we had to apply some tricky scaling to other portions of v' to bring everything to a single scale (that of `cos(t)`. This is detailed in the 'Scale Factor Considerations' section above.

Lastly, we did not utilize the full set of physical dynamics that we originally wanted to. Ideally, we would have had the planet move in an orbit-like fashion and the incoming craft would be pulled into orbit as it neared the planet's gravitational influence. Instead, we went with a fixed-frame approach where the center of the planet is at coordinates $(0, 0)$. Although this is a significant simplification, it's still realistic. When analyzing a physical model, your viewpoint can be any point as long as you are consistent (which we believe we were). Having a more complete physical model of space would have made our model more realistic but harder to prove properties about.

4 Conclusion

With our model and proof, we have shown that while performing a gravitational slingshot, a spacecraft can control its thrust and remain inside the gravity influence of a planet until the ejection angle. Our goal was to show that we could perform such a maneuver with a simulation using accurate physical laws. What we achieved with the development of our model and proof is a first step to creating more complicated and accurate models and proofs that would allow national space agencies and private organizations to confidently execute these precise maneuvers.

A more complete model that involves factors such as change in orbiting radius and acceleration due to spacecraft mass reduction would be a subject for future work that could have exciting and meaningful implications for both cyber-physical systems and space travel.

5 Summary of Deliverables

The computer programs and proof created for this project are found in the appendixes of this document:

- Appendix A includes the full KeYmaeraX Model, which is also found in slingshot.kyx .
- The proof for this model can be found in Appendix B or in slingshot.kyt.

Equal work was performed by both project members.

6 Related Work

Space exploration has been entering new territory in the last 10 years. More and more commercial and private launches have occurred and stirred up interest in what to explore next. Journeys to Mars and beyond require a very precious resource in space: fuel. The hardest part about sending weight beyond our atmosphere is having enough fuel to do so. Maneuvers are planned to be the most efficient possible. But, there is a technique that can help reduce the fuel weight or make the fuel go further: gravity. Using the gravity of a comparatively giant body (like a planet) is like getting free fuel.

Even though the planets don't align very often to allow a massive chain gravitational slingshot executed by the *Voyager* probe, there are often smaller chances to capitalize on the planetary alignments for a boost in deltaV. Utilizing the Moon as a gravitational slingshot to get to Mars will be a key part in transporting people and supplies to the red planet. This topic is very important to future orbital techniques and will without a doubt be discussed when transporting any quantity of bulk beyond our atmosphere.

7 Works Cited

Circular motion. (n.d.). https://en.wikipedia.org/wiki/Circular_motion. (Accessed on 11 April 2016)

FAA. (n.d.). *Section iii.4.1.5 maneuvering in space.* https://www.faa.gov/other_visit/aviation_industry/designees_delegations/designee_types/ame/media/Section\%20III.4.1.5\%20Maneuvering\%20in\%20Space.pdf. (Accessed on 04/28/2016)

Gravity assist. (n.d.). https://en.wikipedia.org/wiki/Gravity_assist. (Accessed on 10 April 2016)

Oberth effect. (n.d.). https://en.wikipedia.org/wiki/Oberth_effect. (Accessed on 1 May 2016)

Orbital mechanics. (n.d.). https://en.wikipedia.org/wiki/Orbital_mechanics#Elliptical_orbits. (Accessed on 1 May 2015)

Voyager program. (n.d.). https://en.wikipedia.org/wiki/Voyager_program. (Accessed on 10 April 2016)

8 Appendix A: KeYmaeraX Model

```
/*
AndrewID(s): amoran, jmannjr
KeYmaeraX version: 4.2b1
Proves with bundled tactic: Y
*/

Functions.
  R G. /* Gravitational constant: G=6.673 x 10^-11 N*(m^2)*(kg^2) */
  R slingtheta. /* Desired slingshot arc angle */
  R vi. /* Initial velocity */
  R u. /* Standard Gravitational Parameter */
  R orbitr. /* Distance from craft to center of planet */
  R planetm. /* Planet's mass */
  R planetr. /* Planet's radius */
End.

ProgramVariables.
  R v. /* Craft's linear velocity */
  R thrust. /* Craft's thrust */

  R theta. /* current completed arc angle */
  R t. /* time */

  R c. /* cosine */
  R s. /* sine */
  R x. /* scale */
End.

Problem.
(
  /* Initial Conditions */
  v >= 0 &
  orbitr > 0 &
  planetm > 0 &
  thrust > 0 &

  /* Current completed angle (theta) starts at 0 and is less than desired angle */
  theta = 0 &
  theta < slingtheta &

  /* Enter at less than escape velocity */
  v^2 <= ((2*G*planetm)/orbitr) &

  /* Orbit must be sufficiently above the ground (150km) */
  planetr + 150 <= orbitr &

  /* Cosine and Sine setup */
  c^2 + s^2 = 1 &
  c = 0 &
```

```
  s = 1 &

  /* Scaling factor setup */
  x^2 <= (c^2/thrust^2) &
  x <= c/thrust
)
->
[
  {
    {
      /* Control Assignments */
      /* Thrust is set non-deterministically for user control */
      /* Thrust is less than escape velocity minus current velocity */
      /* Thrust is chosen so that the scaling factors are satisfied */
      thrust :=*; ?((thrust+v)^2 < ((2*G*planetm)/orbitr) & thrust > 0
      & x < (c/thrust) & x^2 <= (c^2/thrust^2));
    }
    {
      /* cosine and sine functions */
      c' = -s,
      s' = c,

      /* Velocity changes by scaled thrust amount plus gravity assist (cosine). */
      v' = x*thrust + c,


      /* Angle changes by angular velocity which is v/orbitr */
      theta' = v/orbitr,

      /* Non-changing variables */
      planetr' = 0,
      orbitr' = 0,
      thrust' = 0,
      slingtheta' = 0,
      planetm' = 0,
      G' = 0,

      /* Increment time */
      t' = 1

      /* Make sure we never go backwards */
      & v >= 0

      /* Make sure cosine stays between 0 and -1 to prevent a looping orbit. */
      & c > -1
      & c <= 0

      /* Make sure orbitr is positive so that we won't crash (or divide by 0) */
      & orbitr > 0
    }
  }*@invariant(
```

```
    /* While we haven't reached our slingshot desired angle,
     * we are going less than escape velocity
     */
    ((theta <= slingtheta) -> (v^2 <= ((2*G*planetm)/orbitr))) &

    /* To prevent division by 0, make sure orbitr and thrust are always positive */
    orbitr > 0 &
    thrust > 0 &

    /* cosine plus sine should always equal 1 */
    c^2 + s^2 = 1 &

    /* Inside planet gravity */
    planetr + 150 <= orbitr
    )
]((theta <= slingtheta) -> (v^2 <= ((2*G*planetm)/orbitr)))
End.
```

9 Appendix B: KeYmaeraX Proof

```
implyR(1) & loop({`(theta<=slingtheta->v^2<=2*G*planetm/orbitr)&orbitr>0&thrust>0&
c^2+s^2=1&planetr+150<=orbitr`},1) <(
    QE,
    QE,
    composeb(1) & composeb(1) & randomb(1) & allR(1) & testb(1) & implyR(1) &
    diffInvariant({`c^2 + s^2 = 1 & planetr + 150 <= orbitr & v^2 <= (2*G*planetm)/orbitr &
    thrust > 0`}, 1) & diffWeaken(1) & implyR(1) & andR(1) <(
        QE,
        andR(1) <(
            QE,
            andR(1) <(
                QE,
                andR(1) <(
                    QE,
                    QE
                )
            )
        )
    )
)
```

10 Appendix C: Rocket Science Primer

10.1 Orbits

The intuitive explanation of how orbits work can be found in the "Astrodynamics" section of the main document.

Orbits are defined by the following parameters:

- The **eccentricity** e describes the shape of the orbit (how elliptical or how hyperbolic it is).
- The **semi-major axis** a is the average of the orbit's highest radius (apogee) and orbit's lowest radius (perigee).

There are three types of orbits, depending on the eccentricity e:

1. A **circular orbit** occurs when $e = 0$.
2. An **elliptic orbit** occurs when $0 < e < 1$.
3. An **escape trajectory** occurs when $e > 1$. When $e = 1$, the shape of the orbital path is parabolic. When $e > 1$, the shape of the orbit is hyperbolic.

10.2 The Rocket Equation

Force applied to an object is equal to an acceleration times the object's mass. The more massive the object is, the more force one must apply to accelerate it. The less massive the object is, the less force one must apply to accelerate it.

Fuel has mass. Therefore, as fuel is burned, engines become more effective. Because fuel is limited, how much the rocket can speed up or slow down is finite. The capacity for a spacecraft to change its velocity is referred to as a **delta-v budget** and is given by the Tsiolkovsky Rocket Equation:

$$\Delta v = v_e \ln\left(\frac{m_0}{m_f}\right)$$

...where:

- v_e is the engine's exhaust velocity.
- m_0 is the wet mass of the spacecraft (the mass when full of fuel).
- m_f is the dry mass of the spacecraft (the mass when empty of fuel).

10.3 Orbital Transfers

An object in orbit stays in orbit, unless an outside force acts upon it. Due to the relationship between orbital speed and altitude, it stands that changing the speed of the orbit will also change the introduction of energy into the system.

If the spacecraft is orbiting at distance r_1 at velocity v_1 and wishes to reach an altitude of r_2, then the desired change in velocity is:

$$\Delta v = \sqrt{2\left(\frac{\mu}{r_1} - \frac{\mu}{r_1 + r_2}\right)} - v_1$$

(FAA, n.d.)