

UNIVERSITY OF OSLO

THIRD SEMESTER PROJECT IN AST1100

**From Byappo to Hiffre**  
Simulating a Satellite

*Gunnar Felix Lange  
(gunnalan)*

Presented for examination November 2016

## **Abstract**

We present an investigation into the modelling of satellite launches, by launching an imaginary satellite "VoyagerX" in a fictional solar system. We present all stages of the process, including the design of the satellite engine, the launch calculations, the transfer calculations and finally the landing calculations. VoyagerX manages to land safely on our fictional target planet, Hiffre, but not quite where we expected it to land.

# Contents

<b>Abstract</b>	<b>1</b>
<b>Index of common quantities</b>	<b>5</b>
<b>1 Introduction</b>	<b>6</b>
<b>2 Theoretical model &amp; Methods</b>	<b>7</b>
2.1 Building the rocket . . . . .	7
2.1.1 Theory of operation of a rocket engine . . . . .	7
2.1.2 Modelling our fuel . . . . .	7
2.1.3 Finding the number of cells and the amount of fuel needed .	9
2.1.4 Analytic calculations of the required fuel . . . . .	10
2.1.5 Method to model our engine . . . . .	11
2.2 Investigating the properties of our solar system . . . . .	12
2.2.1 Calculating the planetary orbits . . . . .	12
2.2.2 Choosing convenient units . . . . .	13
2.2.3 The habitable zone of our solar system and choosing a target planet . . . . .	14
2.2.4 Consistency checks for our solar system . . . . .	16
2.2.5 Exploring how our solar system looks from afar . . . . .	16
2.3 Orbital calculations . . . . .	18
2.3.1 First phase: Departure from home planet . . . . .	18
2.3.2 Second phase: Interplanetary space . . . . .	20
2.3.3 Third phase: Arrival at Hiffre, our target planet . . . . .	23
2.3.4 Computing the power required by our landing module . . . .	25
2.3.5 Methods . . . . .	26
2.4 Orientating the satellite in interplanetary space . . . . .	27
2.4.1 Finding the orientation of the satellite in interplanetary space	27
2.4.2 Finding the position of the satellite in interplanetary space .	30
2.4.3 Finding the velocity of the satellite in interplanetary space .	31
2.4.4 Orientating the satellite: Methods . . . . .	32

2.5	Investigating Hiffre, the target planet . . . . .	32
2.5.1	Using spectroscopy to determine molecular mass . . . . .	33
2.5.2	Method for determining the mean molecular mass . . . . .	35
2.5.3	Modelling the atmosphere . . . . .	36
2.6	Landing on our target planet . . . . .	39
2.6.1	Modelling the force from our atmosphere . . . . .	39
2.6.2	Getting into a close orbit around Hiffre . . . . .	40
2.6.3	Slowing our descent . . . . .	41
2.6.4	Picking out a landing spot and landing on Hiffre . . . . .	42
2.6.5	Communicating with VoyagerX . . . . .	43
2.6.6	Methods for landing on our target planet . . . . .	44
<b>3</b>	<b>Results</b>	<b>45</b>
3.1	Results from our engine simulation . . . . .	45
3.2	Result from our simulation of the solar system . . . . .	48
3.3	Results from our orbital calculations . . . . .	51
3.4	Results from our orientation software . . . . .	53
3.4.1	Software to determine orientation . . . . .	53
3.4.2	Software to determine position . . . . .	53
3.4.3	Software to determine the velocity of the satellite . . . . .	53
3.5	Results from the actual launch to get into a stable orbit . . . . .	54
3.6	Results from the spectroscopic investigation . . . . .	55
3.7	Results from our modelling of the atmosphere . . . . .	56
3.8	Results from our landing simulation . . . . .	57
<b>4</b>	<b>Discussion</b>	<b>61</b>
4.1	The engine simulation . . . . .	61
4.2	The simulation of our solar system . . . . .	63
4.2.1	The simulation of the planet orbits . . . . .	63
4.2.2	The investigations into the appearance of our solar system from afar . . . . .	63
4.3	The orbital simulation . . . . .	65
4.4	The orientation of our satellite . . . . .	66

4.5	The actual launch from Byappo . . . . .	66
4.6	The exploration of Byappo's atmosphere . . . . .	67
4.7	The landing . . . . .	68
<b>5</b>	<b>Conclusion/Outlook</b>	<b>70</b>
5.1	Conclusion . . . . .	70
5.2	Outlook . . . . .	70
	<b>Bibliography</b>	<b>72</b>
	<b>Appendices</b>	<b>74</b>
<b>A</b>	<b>Numerical methods</b>	<b>74</b>
A.1	Leapfrog integration . . . . .	74
A.2	The $\chi^2$ algorithm . . . . .	75
A.3	The Forward-Euler algorithm . . . . .	76
<b>B</b>	<b>Proof of selected results</b>	<b>77</b>
B.1	Finding the value of $\mathbf{G}$ in our new units . . . . .	77
B.2	Finding the extreme values of $\mathbf{x}_{\text{picture}}$ and $\mathbf{y}_{\text{picture}}$ . . . . .	77
B.3	Solving the triangulation equation . . . . .	78
B.4	Finding the velocity of the satellite . . . . .	79
B.5	Atmosphere calculations . . . . .	80
B.5.1	Adiabatic model . . . . .	81
B.5.2	Isothermal model . . . . .	82
B.5.3	Implementing the boundary conditions . . . . .	83
<b>C</b>	<b>Pertinent parameters</b>	<b>84</b>

# Index of common quantities

$T$	Temperature
$R$	Radius
$M_p$	Mass of planet
$R_p$	Radius of planet
$R_B$	Radius of Byappo (home planet)
$R_H$	Radius of Hiffre (target planet)
$M_*$	Mass of the star
$T_*$	Temperature of the star
$A_*$	Surface area of the star
$M_B$	Mass of Byappo (home planet)
$M_H$	Mass of Hiffre (target planet)
$M_{sat}$	Mass of satellite
$k$	Boltzmann's constant
$L$	Luminosity
$F_p$	Flux
$P$	Power
$\sigma$	Stefan-Boltzmann's constant
$u$	Unified atomic mass unit
$c$	Speed of light
$G$	Gravitational constant
$\sigma$	Standard deviation
$\alpha$	Field of view angle
$\theta$	Polar angle in spherical coordinates
$\phi$	Azimuthal angle in spherical coordinates
$FWHM$	Full width at half maximum

# 1 Introduction

Satellites have in the past decades become an essential tool for astrophysicists studying the universe. Just a few decades ago, all data about the distant universe came from ground-based telescope. Now, more and more satellites are travelling through space, gathering vital information about the cosmos.

This project describes in detail how such a satellite mission could be planned and executed. We imagine ourselves in a solar system, not at all different from our own, where sentient beings are constructing a satellite with which to explore their neighboring planets. This solar system consists of seven planets, each with a name and number. These are listed in table 7,in appendix C. Our sentient beings are on planet 0, Byappo. They are constructing a satellite, **VoyagerX**, with which they aim to explore their solar system. We discuss how to construct and calculate the engine for this satellite, how to explore the solar system prior to launch, how to launch the satellite and how to investigate the target planet. Finally, we will launch the lander module **HAL9001**, to actually land on our target planet.

Whilst this project is not entirely realistic, as we make a large number of assumptions throughout, is not entirely unreasonable either. Many of the calculations presented here, are the exact calculations done by actual rocket scientists.

We will also simulate an actual launch of the satellite. This is done through the AST1100SolarSystem module, available [here](#)<sup>1</sup>. This module, combined with the graphics module developed specifically for this project (available [here](#)<sup>2</sup>), creates a much more realistic setting to this project, and is the source for all rendered images of the solar system found in this project.

---

<sup>1</sup><http://www.uio.no/studier/emner/matnat/astro/AST1100/undervisningsmateriale/>

<sup>2</sup><http://www.rio.co.uk/MCAst/>

## 2 Theoretical model & Methods

We begin with an extended discussion of the theory required to get our satellite, VoyagerX, from our home planet Byappo to our target planet. We focus on the physics, leaving most of the mathematics either to the appendices or external references. We also discuss some technicalities relating to the implementation of the theory in either our numeric simulation or in our actual launch.

### 2.1 Building the rocket

In this section we give an overview of the design of our rocket, focusing on the fuel chamber, and present our fuel calculations.

#### 2.1.1 Theory of operation of a rocket engine

A rocket engine works by ejecting a mass of fuel from the bottom of the rocket, thereby increasing the rocket velocity. If we assume our rocket to be in deep space, far from gravitational interactions, there are no external forces acting on the rocket, and therefore momentum is conserved. Using Newton's second and third law then gives us the equation governing the motion of the rocket:

$$F_{rocket} = -\frac{dp}{dt} = -\frac{d}{dt}(mv) \quad (1)$$

Where  $F_{rocket}$  is the force on the rocket,  $m$  is the mass ejected and  $v$  is the velocity at which this mass is ejected (exhaust velocity). Our task is therefore to find an appropriate force which we want our rocket to output, and subsequently scale the right-hand side of equation 1 accordingly. We begin by investigating our fuel.

#### 2.1.2 Modelling our fuel

We will make a set of assumptions to model our fuel. These assumptions simplify our calculations, and are not too unreasonable. We will later include an error margin in our fuel calculations, to ensure that these assumptions do not have disastrous consequences. We assume the following:

- All particles in the engine are H<sub>2</sub> gas.
- The gas does not ignite

- The gas is ideal
- All particles exiting the engine are immediately refilled

Note in particular that this means that the density, pressure and temperature of the fuel is constant. It can be shown (see for example [Schroeder, 2000]) that the velocity distribution in an ideal gas follows the Maxwell-Boltzmann distribution. This is a Gaussian distribution, with parameters:

$$\mu = 0, \quad \sigma = \sqrt{\frac{kT}{m}} \quad (2)$$

Where  $\mu$  is the mean of the Gaussian distribution,  $\sigma$  is the standard deviation of the Gaussian distribution,  $m$  is the mass of the particles and  $k$  is Boltzmann's constant. With this relation, we can run simulations of gas particles, to find physical quantities, such as the pressure of the gas.

To maintain computational efficiency, we simulate only a small cell, and build our rocket motor by combining many such cells. Each cell is quadratic with length  $L$ . In each cell, we simulate particles with an initial velocity distribution given by the Maxwell-Boltzmann distribution (at a chosen temperature), as found in [Schroeder, 2000], with parameters given by equation 2. We let the initial positions of the gas particles be uniformly distributed in our cell. Because we assume our gas to be ideal, we neglect internal collisions of the molecules. When the particles collide with the bottom of the container, we count its mass,  $m$  and its velocity component in the direction normal to the bottom,  $v_i$ . We then reflect the particle back into the container and reverse its velocity component. Finally we apply equation 1 in a discretized form to calculate the force exerted by the particles on the container as:

$$F = -\frac{2 \sum mv_i}{\Delta t}$$

Where the sum extends over all particles that hit the bottom of the container in a single time step,  $\Delta t$ . Note that  $\Delta p = 2 \sum mv_i$ , as the velocity component,  $v_i$ , is reversed, for a total velocity change of  $2v_i$ .

Finally, we make a hole in the bottom of our container, to let the particles escape. This hole is quadratic with length  $L/2$ . We can then simply count the number of particles escaping through the hole, and compute the force exerted on the rocket from equation 1. We then again reflect the particles back into the container and reverse its velocity. This is a choice which ensures a small statistical bias, as we discuss in section 4.1.

### 2.1.3 Finding the number of cells and the amount of fuel needed

The number of cells determines the power output of our motor. We will show in section 2.1.4 that this does not affect total fuel usage. We therefore make an arbitrary choice; we require that our rocket should reach escape velocity from our home planet, Byappo, within 20 minutes. As we show in section 2.3.3, the escape velocity from a planet is generally given as:

$$v_{esc} = \sqrt{\frac{2GM}{R^2}} \quad (3)$$

Where  $M$  is the mass of the planet, and  $R$  is its radius. If we assume (somewhat unrealistically) that there is no gravity during the ascent <sup>3</sup>, we can find the required number of cells from:

$$v(t) = \int_0^t a(t') dt' = \int_0^t \frac{F}{m(t')} dt' = F \int_0^t \frac{1}{m(t')} dt'$$

Where  $m(t') = M_{sat} + M_{fuel} - \Delta m t'$ . Here  $M_{sat}$  is the total mass of the satellite,  $M_{fuel}$  is the total mass of fuel and  $F$  is the force from a single box and  $\Delta m$  is the mass of particles exiting a single cell per unit time. This is thus the velocity change due to a single cell. Simulating over 20 minutes gives:

$$v = F \int_0^{20} \frac{1}{M_{sat} + M_{fuel} - \Delta m t'} dt' \quad (4)$$

Where  $v$  is the velocity change due to a single cell over twenty minutes. Computing the integral gives:

$$v(t) - v_0 = \frac{F}{\Delta m} \ln \frac{M_{fuel} + M_{sat}}{M_{fuel} + M_{sat} - \Delta m t} \quad (5)$$

Assuming that the rocket starts from the surface of the planet, this gives the velocity change of a single cell as:

$$v = \frac{F}{\Delta m} \ln \frac{M_{fuel} + M_{sat}}{M_{fuel} + M_{sat} - \Delta m t}$$

We can then compute the number of boxes required simply by computing:

$$N = \lceil v_{esc}/v \rceil \quad (6)$$

---

<sup>3</sup>This may seem like a bizarre assumption, but it greatly simplifies calculations and, as earlier mentioned, the number of cells has no effect on the required fuel mass anyway.

Then the total velocity change over a certain time period is found by multiplying the quantities that depend on a single box in equation 4 by  $N$ , giving:

$$v(t) = NF \int_0^t \frac{1}{M_{sat} + M_{fuel} - N\Delta mt'} dt' \quad (7)$$

We will find an expression for the required fuel from this equation in section 2.1.4. However, to ensure the correctness of our simulation, we also compute this numerically. By choosing an  $M_{fuel}$ , we can iteratively compute the above integral, using the method described in section 2.1.5, until the required  $\Delta v$  has been reached. We can then compute the mass used in the boost from  $M_{used} = N\Delta mt$ .

#### 2.1.4 Analytic calculations of the required fuel

We can solve equation 7 analytically, by solving the integral. This gives:

$$v(t) - v_0 = \frac{F}{\Delta m} \ln \frac{M_{fuel} + M_{sat}}{M_{fuel} + M_{sat} - N\Delta mt}$$

Now we can employ a neat trick, originally devised by Daniel Heinsen, all credits go to him. We can simply require that we burn all fuel at the end of our burn, i.e. that  $M_{fuel} - N\Delta mt = 0$ . This gives:

$$v(t) = v_0 + \frac{F}{\Delta m} \ln \left( 1 + \frac{M_{fuel}}{M_{sat}} \right)$$

Which can be rewritten as:

$$M_{fuel} = M_{sat} \left( \exp \left( \frac{\Delta m(v - v_0)}{F} \right) - 1 \right) \quad (8)$$

Note that this is independent of the number of boxes - a large number of boxes means a large boost over a small time, a small number of boxes means a small boost over a large time. If there are no external forces doing work during the boost (as we have assumed, and which will be correct if the boosts are made over a short time interval), this makes no difference. Note that this assumes that all fuel is burned at the end of the boost. Thus we must input all boost we desire to calculate the total fuel needed. In practice, we determine  $M_{fuel}$  from equation 8, and check that this gives us our required velocity change by inputting the computed  $M_{fuel}$  into equation 7.

### 2.1.5 Method to model our engine

We implement four important consistency checks for our engine.

First, we check that our initial, mean kinetic energy is close to the theoretical value for the Boltzmann distribution which is shown in [Schroeder, 2000] to equal:

$$\bar{E}_k = \frac{3}{2}kT$$

Secondly, we check that our pressure, defined as  $P = F/L^2$ , follows the analytic expression, given by:

$$P = nkT$$

Where  $n$  is the number density (particles per cubic meter). Finally, we check that the mean speed (the absolute value of the velocity) follows the theoretical value, which is given by:

$$v = \sqrt{\frac{8kT}{\pi m}}$$

We compute the relative error of each of these quantities, defined as:

$$\epsilon_{\text{rel}} = 100 \cdot \frac{x_{\text{computed}} - x_{\text{analytic}}}{x_{\text{analytic}}} \quad (9)$$

Where  $x_{\text{computed}}$  is the value from our simulation and  $x_{\text{analytic}}$  is the analytic value. Thus an important consistency check that we implement is that  $\epsilon_{\text{rel}}$  is small.

The final check is that we animate a box containing a few gas particles, and check visually that they behave as we expect.

Another important aspect is that our comparison of the iterative solution of equation 7 with the analytic solution given by equation 8. We sketch how to compute the iterative estimate in pseudo-code below:

```

1 mass=m_sat+m_fuel
2 while v < v_esc:
3     v+=number_of_boxes*(F/mass)*dt
4     mass -= number_of_boxes*mass_per_second*dt

```

## 2.2 Investigating the properties of our solar system

In this section we present the model of our simulated solar system. We discuss orbital calculations, as well as consistency checks for these calculations. We also briefly discuss how our solar system may appear to extraterrestrials.

### 2.2.1 Calculating the planetary orbits

Newton's law of gravitation, which describes the force of gravity between two objects, is given by:

$$\vec{F}_G = \frac{Gm_1m_2}{r^3} \vec{r} \quad (10)$$

Where  $F_G$  is the force of gravity on the first mass,  $G$  is the universal gravitational constant,  $m_i$  are the masses of the two objects and  $\vec{r}$  is the vector pointing from  $m_1$  to  $m_2$ . We can combine this law with Newton's second law of motion to get a set of coupled differential equations. One simplification which we impose on these equations, is to assume a coplanar orbit of our planets<sup>4</sup>. This gives us the following set of equations:

$$\begin{aligned} \frac{d^2x}{dt^2} &= \frac{F_{G,x}}{m_1} \\ \frac{d^2y}{dt^2} &= \frac{F_{G,y}}{m_1} \end{aligned} \quad (11)$$

Where  $F_{G,x}$  and  $F_{G,y}$  are the components of the gravitational force in the  $x$  and  $y$  directions respectively. These equations can be reformulated as a set of first-order equations by introducing the velocity,  $v$ , such that:

$$\frac{dx}{dt} = v_x, \quad \frac{dv_x}{dt} = \frac{F_{G,x}}{m_1} = a_x(x, y) \quad (12)$$

And equivalently for the  $y$  direction. This gives four coupled, first-order, linear, differential equations for each planet-interaction. We can solve these equations by using the leapfrog method, described in appendix A.1. However, we impose an additional simplification: we assume the star to be the dominant object in the solar system, and therefore ignore interplanetary forces. This is justified if the star is significantly more massive than all other planets, which is the case as can be seen in appendix C. We make one further approximation: we can let the star be

---

<sup>4</sup>This is a good assumption for most solar systems, as they are formed from a single cloud of dust. By conservation of angular momentum, if this cloud collapses, it starts to spin faster. Consequently, the systems are flattened out by the Coriolis force. For more details, see e.g. [Katz et al., 2012]

stationary *at the origin*, and neglect the reciprocal force from the interaction of the star with the planets. With these simplifications, we can reduce the number of first order equations to  $4 \times$  number of planets (one for each  $xy$  component and one for velocity and position respectively). However, this also introduces many possible error sources. We will investigate the validity of our system model in section 2.2.4.

### 2.2.2 Choosing convenient units

When dealing with interplanetary distances, and the mass of celestial objects, standard SI units quickly become inconvenient. We will therefore adapt a different set of units for the interplanetary travel of our satellite, VoyagerX (though we will once again use SI units when approaching our target planet). These units are determined from physical considerations of our system.

A good length unit is the average distance between a planet and a star in a galaxy far far away (the planet is called "the earth" and the star is called "the sun" by earth's primate inhabitants.<sup>5</sup>) This distance is called an astronomical unit, and is equal to:

$$1\text{AU} = 1.49598 \times 10^{11}\text{m}$$

In equal spirit, we will define our mass unit as a solar mass,  $M_{\odot}$ , which equals

$$M_{\odot} = 1.989 \times 10^{30}\text{kg}$$

Finally, we let our time unit be given by the average time it takes the earth to revolve around the sun, the year. This is given by:

$$1 \text{ year} = 365.24 \times 24 \times 60 \times 60\text{s} = 3.1556736 \times 10^7\text{s}$$

We show in appendix B.1, that the gravitational constant,  $G$ , in these units, is given by:

$$G = 4\pi^2 \text{ AU}^3 \text{year}^{-2} M_{\odot}^{-1}$$

---

<sup>5</sup>This may seem ridiculous, but it is no more ridiculous than other units that are commonplace in modern astrophysics. Also, it greatly simplifies the expression for the gravitational constant, as we show in appendix B.1.

### 2.2.3 The habitable zone of our solar system and choosing a target planet

We define the habitable zone of our solar system as the belt around our star where the surface temperature of the star is between 260 K and 390 K.<sup>6</sup> To find the location of this in our solar system, we assume our star to be a perfect black body.<sup>7</sup> Stefan-Boltzmann's law gives the total power radiated from our star (luminosity) as:

$$L = A_* \sigma T_*^4 = 4\pi \sigma R_*^2 T_*^4 \quad (13)$$

Where  $A_*$  is the surface area of the star,  $R_*$  is the radius of the star,  $\sigma$  is Stefan-Boltzmann's constant and  $T_*$  is the temperature of our star. This power will distribute itself evenly across a spherical shell with radius  $r$ . This shell has area  $4\pi r^2$ , and thus the power per area (flux) is:

$$F_P = \frac{L}{4\pi r^2} = \frac{\sigma R_*^2 T_*^4}{r^2}$$

Assume now that light rays hit a planet normally (as they necessarily must in a two-dimensional solar system). Then the planet will expose a great circle to the light rays (the light rays do not "see" the curvature of the planet). This circle has area  $\pi R_p^2$ , where  $R_p$  is the radius of the planet. Thus, the total power received by planet is:

$$P_{in} = \frac{\pi \sigma T_*^4 R_p^2 R_*^2}{r^2} \quad (14)$$

As the planet heats up, it will in turn radiate back according to equation 13, with the starred quantities replaced with the equivalent quantities for the planet. Equating power out and power in gives:

$$\frac{\pi \sigma T_*^4 R_p^2 R_*^2}{r^2} = 4\pi R_p^2 \sigma T_p^4$$

So that finally:

$$T_p = T_* \sqrt{\frac{R_*}{2r}} \quad (15)$$

Or alternatively:

$$r = \frac{R_* T_*^2}{2 T_p^2} \quad (16)$$

---

<sup>6</sup>Which is approximately the temperature where water is liquid

<sup>7</sup>This is usually a good approximation for stars, but all black-body approximations in this section could be made more realistic, simply by including an emissivity,  $\epsilon$ , in the calculations.

Inserting values from appendix C and letting  $T_p = 260$  K and 390 K respectively, gives for the habitable zone,  $r_{\text{Habitable}}$  as:

$$r_{\text{Habitable}} \in [2.28\text{AU}, 5.13\text{AU}]$$

The habitable zone has been plotted together, with the remaining planets in our solar system, in the plot below:

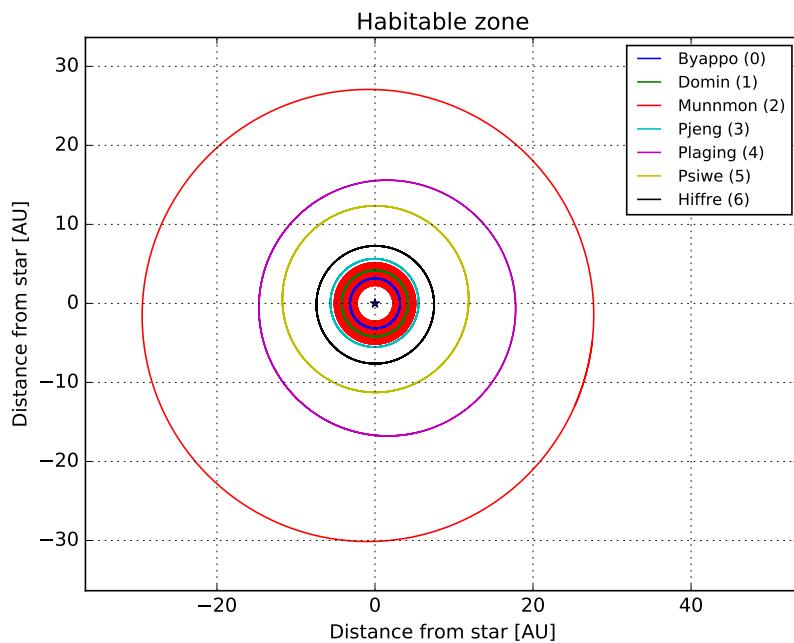


Figure 1: The orbits of the planets in our solar system. The habitable zone of our solar system has been indicated by a red donut.

Figure 1 shows that Byappo and Domin are inside the habitable zone. Pjeng is just on the edge of the habitable zone. Hiffre is the next planet out. We must now decide which planet to venture to.

The planets inside the habitable zone have been explored by previous missions. Therefore, we decide to plan our mission to Hiffre. Previous observations have shown that Hiffre is an interesting Ridged Planet, with a fascinating geological history and breathtaking nature.

#### 2.2.4 Consistency checks for our solar system

To ensure that our simulations are correct, it is important to have certain benchmarks to compare the results to. An example of such benchmarks would be conserved quantities. Two important conserved quantities are energy and angular momentum.

Mechanical energy should be conserved, because gravity is a conservative force. Thus the quantity:

$$\frac{1}{2}M_p v^2 - \frac{GM_* M_p}{r} \quad (17)$$

Which is the sum of potential energy and kinetic energy, should be constant throughout the simulation. Here  $M_*$  is the mass of the star and  $M_p$  is the mass of a planet. Note that the only reason why the expression for the potential energy is so simple is because we neglected the interplanetary forces and the forces on the star. If these were included, we would have to expand our definition of potential energy to include the interplanetary potentials.

Angular momentum should be conserved because there is no external torque. Thus the quantity:

$$\vec{l} = M_p \vec{v} \times \vec{r} \quad (18)$$

Should be constant for all times. Here  $\vec{r}$  is the position vector of the planet with mass  $M_p$ .

Equation 17 and 18 provide an important way to check our results.

#### 2.2.5 Exploring how our solar system looks from afar

The VoyagerX project is partially funded by, and developed in collaboration with, the BASA (Byappo's Agency for Space Alliances). In this section, we therefore investigate how our solar system would appear from afar, including both the velocity curve of our star and the light curve of our star. We wish to know if an extraterrestrial could detect the planets around our star.

##### The star's velocity curve

Our star is by far the most luminous object in our solar system. Therefore, this is also the object that any extraterrestrial is most likely to see. Note that, contrary to our earlier assumption, our planets also affect our star. This will cause it to oscillate slightly back and forth over time. This oscillation is what we use to determine the existence of extrasolar planets. We will therefore investigate how

these oscillations appear to an outsider observing our solar system. Note that this in general depends on the inclination angle (the angle between the surface normal of the solar system and the observer's line of sight) that the observer has to us. We will adapt the "best-case" scenario, where the inclination angle is  $90^\circ$ , so that the distant observer sees the full effect of the oscillation of the star. We will also assume that the observer sees along the  $x$ -axis. We therefore only consider the  $x$ -component of the velocity, and call it the radial component.

Note that, in order to achieve this, we will have to discard our previous assumption that the star is at rest. Instead, we will employ equation 10, where we now sum over all celestial bodies. To reduce computation time, however, we will only include the three most massive planets<sup>8</sup>. There is, however, a slightly problem with this - it is not at all obvious what initial velocity to give the sun. We solve this by requiring that the center of mass (COM) of our solar system should be fixed. The general equation for the velocity of the COM is:

$$\vec{v}_{\text{COM}} = \sum_i m_i \vec{v}_i \quad (19)$$

Where the sum extends over *all* celestial objects. Thus, we can fix the center of mass at the origin by requiring that the initial velocity of our star,  $\vec{v}^*$ , equals:

$$\vec{v}^* = \frac{1}{M_*} \sum_j m_j \vec{r}_j$$

Where the sum now extends over all celestial objects *except for our star*.

Our observations always have some noise, due to external disturbances. It is reasonable to assume that potential extraterrestrials may have similar problems. Therefore, we will include some Gaussian noise in our velocity curve. We will let this noise have zero mean, and a standard deviation equal to  $1/5$  of the maximum velocity of the star as an arbitrary, but reasonable, choice.

### The star's light curve

One additional possible indicator of extrasolar planets is a variation in the light from the star. The emitted flux will slightly dip if a planet goes in front of the star, as it absorbs some of the flux in the eclipse. We will implement a rather simplistic model to investigate this. We let all stars start at a distance of 1 AU away from the star, to subsequently let them travel towards the star, with the velocity that

---

<sup>8</sup>We justify this by running brief simulation with all planets and checking against the simulations with only the three most massive ones - to realize that there is little difference.

they have at time 0<sup>9</sup>. Once a planet is in front of our star, we will reduce the flux emitted by the star from 1 to:

$$F_{\text{reduced}} = 1 - \frac{R_p^2}{R_*^2} \quad (20)$$

Where  $R_p$  is the radius of the planet and  $R_*$  is the radius of the star. This models two discs overlapping each other.

We will also include some Gaussian noise in this calculation, for the same reasons as mentioned above. We let the noise have a mean of 0 and a standard deviation of 0.2, again as an arbitrary but reasonable choice.

## 2.3 Orbital calculations

In this section we present the basic methods used to compute the orbit of our satellite, VoyagerX. We will use a patched conic approximation (see e.g. [E.Barrabes et al., 2004]), i.e. we will at each stage assume that there is only one dominating body (our home planet Byappo, the star and our target planet Hiffre respectively). This approximation is valid if we are sufficiently far away from all other bodies. Whilst this will not be exact enough for our purposes, it will give us a starting point from where to start our simulations. We then later add a correction factor. We will attempt a hyperbolic escape from our home planet, followed by an elliptic transfer between the planets, and finally a hyperbolic capture by Hiffre, our target planet.

### 2.3.1 First phase: Departure from home planet

In this phase, we assume Byappo to be the dominating object. We wish to escape from Byappo's influence. One common way to classify this in astrodynamics is through the idea of a *sphere of influence*. This is an imaginary sphere around our planets, within which the patched conic approximation is assumed to hold (as explained in greater detail by [E.Barrabes et al., 2004]). The idea is to investigate where the force from the closer, but less massive, object is larger than the force from the distant but more massive object. The exact mathematical treatment of this topic requires perturbation theory, and is therefore not discussed here. The interested reader may find a rigorous treatment of this subject in [E.Barrabes et al., 2004] or [Burrows, 1966]. There it is shown that the radius

---

<sup>9</sup>For an explanation of what time 0 corresponds to, see section 2.3.2

of the sphere of influence,  $r_{SOI}$  is approximately equal to:

$$r_{SOI} \approx a \left( \frac{M_B}{M_*} \right)^{2/5} \quad (21)$$

Where  $a$  is the distance from the planet to the star (we will use the major axis of the planet),  $M_B$  is the mass of the home planet Byappo and  $M_*$  is the mass of our star. Note that  $r_{soi}$  describes the distance *from the home planet* at which we can largely neglect the influence of the home planet. This approximation is valid if  $M_* \gg M_B$ , and  $M_B \gg M_{sat}$ , i.e. that the star is significantly heavier than the planet, and the planet is significantly heavier than the satellite. Furthermore, the approximation is only valid if  $r_{soi} \gg R_B$ , i.e. that the planet is not too close to the star. All of these conditions are satisfied in our case, as can be seen from the table of values in appendix C.

We wish to make a hyperbolic escape of our satellite out to at least the height  $r_{SOI}$ , from the surface of the planet. The boost needed to achieve this can be computed from energy considerations. Equation 17 describes the energy in orbital motion. Note that the satellite has zero initial velocity relative to the planet. Thus the total initial energy of the satellite,  $E_{tot,i}$ , at the surface of the planet, is given by:

$$E_{tot,i} = -\frac{GM_B M_{sat}}{R_B}$$

Where  $R_B$  is the radius of the Byappo. The total potential in orbit, at a height of  $r_{soi}$  from our home planet Byappo,  $E_{pot,SOI}$ , is given by:

$$E_{pot,SOI} = \frac{1}{2}mv^2 - \frac{GM_B M_{sat}}{r_{SOI}}$$

Thus the energy difference,  $\Delta E$ , is:

$$\Delta E = GM_B M_{sat} \left( \frac{1}{R_B} - \frac{1}{r_{SOI}} \right)$$

This is thus the energy which we must give our satellite, in the form of kinetic energy,  $1/2M_{sat}v^2$ . Equating these expression gives us:

$$v = \sqrt{2GM_B \left( \frac{1}{R_B} - \frac{1}{r_{SOI}} \right)} \quad (22)$$

This is thus the velocity which we must give the satellite to get it a distance  $r_{SOI}$  away from the planet. Note that this is the velocity *relative to the planet*, as we assumed zero initial velocity of the satellite.

### 2.3.2 Second phase: Interplanetary space

#### Determining the magnitude and direction of required boost

Once we have arrived at the sphere of influence, we can commence the next stage of the patched conic approximation. This is an elliptic transfer. We now consider our satellite to be in an orbit around our star. This orbit will in general be elliptic, with an eccentricity close to that of the orbit of our home planet<sup>10</sup>. As can be seen in the table in appendix C, this eccentricity is fairly low for Byappo, i.e. the planet is in an almost circular orbit. We will therefore approximate the orbit of our satellite VoyagerX, after it has exited Byappo's sphere of influence, as a circular orbit around the sun. This simplifies the calculations immensely, and the error introduced by this will only be added to the error introduced by the patched conic approximation. Therefore, we will only need to tweak an already existent correction factor. Consequently, this assumption does not significantly complicate our problem.<sup>11</sup>

The reason for approximating the orbit as circular, is because of the existence of a famous analytic solution for transferring the satellite between two circular orbits, first published by German scientist Walter Hohmann - the Hohmann transfer. The idea of the transfer is to increase the velocity of a spacecraft in a circular orbit. This increase is done along the direction of the spacecraft's velocity, so that the spacecraft enters an elliptical orbit. The major axis of this ellipse is chosen in such a manner that it coincides with the major axis of the orbit of the target planet. The idea is then to hit the target planet precisely, so that we can boost directly into an orbit around it from the elliptical orbit. The idea of the Hohmann transfer orbit is illustrated in the figure below:

---

<sup>10</sup>This follows from energy considerations: we have only slightly increased the energy of our satellite, and it is still at approximately the same distance from the sun. Thus, if it orbits the sun, it should have a similar eccentricity.

<sup>11</sup>The avid reader may wonder why we do not simply boost the satellite into a circular orbit, using equation 28. This is because there are already errors which require a correction factor. Therefore we do not gain much by boosting into a circular orbit.

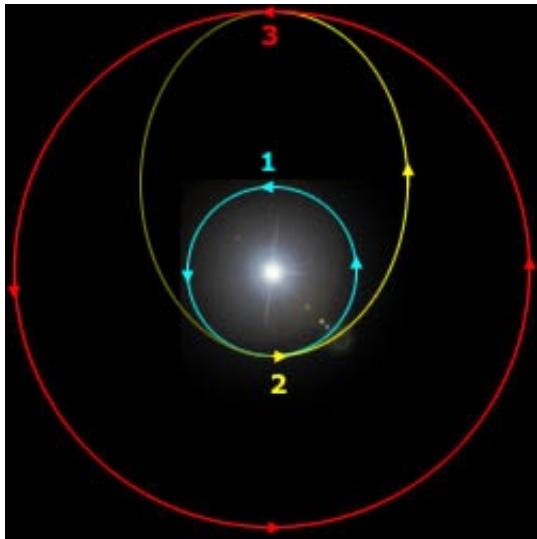


Figure 2: Visual representation of a Hohmann transfer. Note carefully the geometry of the problem. This image is taken from [Wilson, 2012], all courtesy go to the original creator.

The rigorous mathematical treatment of this subject, whilst not difficult, takes us to far afield. The full explanation of the Hohmann transfer equations can be found in [Widnall and Peraire, 2008]. There, it is shown that the *change* in velocity needed from the circular orbit to the elliptical orbit,  $\Delta v$ , is given by:

$$\Delta v = \sqrt{\frac{GM_*}{r_1}} \left( \sqrt{\frac{2r_2}{r_1 + r_2}} - 1 \right) \quad (23)$$

Where  $r_1$  and  $r_2$  are the radii of the two circular orbits - in our case, this is given (approximately) by the initial distances of the planets to the star at the time of launch,  $t_0$ . This gives the magnitude of the boost. The direction is, as earlier stated, along the velocity vector of the satellite. One important thing to note about this, is that the Hohmann transfer assumes instantaneous velocity changes. As we showed in section 2.1.4 however, the fuel usage is independent of the number of fuel cells, and consequently the acceleration of the rocket<sup>12</sup>. Therefore, we may assume instantaneous impulses.

With the magnitude of the velocity change determined, all that remains before we can launch towards Hiffre is to determine the time of launch,  $t_0$ . We let  $t = 0$  correspond to the earliest possible time we could launch VoyagerX (right after its

---

<sup>12</sup>Note that the mass of the fuel cells themselves may be ignored, as they can be ejected off the rocket if necessary.

completion), and wish to investigate how long we have to wait after this time.

### Determining the launch time, $t_0$

Note that we are boosting into an elliptical orbit with major axis given by:

$$a = \frac{r_1 + r_2}{2}$$

We do not know the parameters  $r_1$  and  $r_2$  yet, as they are the distance of the planets from the star at  $t = t_0$ . We will therefore approximate these values with the major axis of the planets,  $a_1$  and  $a_2$ . Now we employ Kepler's third law. This gives us approximately:

$$\frac{T^2}{a^3} = \frac{4\pi^2}{GM_*}$$

Note that this is only valid if the mass of the star is much larger than all other masses, which is indeed the case here. Inserting and solving gives:

$$T = \sqrt{\frac{\pi^2(a_1 + a_2)^3}{2GM_*}}$$

Note that, as illustrated in figure 2, we start the transfer at the perihelion of the ellipse and want to hit our planet at aphelion. This is only halfway around the ellipse, so that the relevant timeframe for our simulation is:

$$T_s = \pi \sqrt{\frac{(a_1 + a_2)^3}{8GM_\odot}} \quad (24)$$

This gives us the time taken to transfer between the orbits. From this, we can easily find the required start time. Note that we wish Hiffre to be at an angle of  $\pi$  radians from our starting position when we impact it, as shown in figure 2. Thus, after the time  $T_s$ , the planet should be at an angle  $\pi$  from our initial angle. The angular velocity of Hiffre,  $\omega_T$  can be found by once again applying Kepler's third law, which gives (using that  $\omega = 2\pi/T$ ):

$$\omega_T = \sqrt{\frac{GM_*}{a_2^3}} \quad (25)$$

The initial alignment,  $\phi$ , should then be:

$$\phi = \pi - \omega_T T_s \quad (26)$$

I.e., if Hiffre (target planet) starts at an angle  $\phi$  behind Byappo (home planet) at launch, Hiffre will be exactly at an angle  $\pi$  from the initial position of the home planet after a time  $T_s$ . Thus we can numerically find the time,  $t_0$  at which we want to launch the satellite, by requiring that the angle between the planets,  $\phi$  at  $t_0$  equals the angle dictated by equation 26.

### 2.3.3 Third phase: Arrival at Hiffre, our target planet

To commence the last phase of our patched conic approximation, we must be close enough to Hiffre to achieve a stable orbit. Additionally, we want to be close enough to take pictures of the planet, to get information about the atmosphere.

#### Achieving a stable orbit

This is in general, once again a question that must be answered by perturbation theory. Notice, that the sphere of influence description provided in section 2.3.1 is no longer useful, as we are now looking to attain a stable orbit over an extended period of time. It is therefore not sufficient to be far enough out that the forces from the planet are stronger - all other forces must be *utterly* negligible to achieve a stable orbit with reasonably large timesteps. We decide to impose a simple condition: we require the force from the planet to be  $k$  times stronger than the force from the sun, and then conduct numerical experiments to establish an appropriate value of  $k$ . Note that it is rather straightforward to establish the radius at which this occurs, simply by requiring that the force from the planet be  $k$  times stronger than the force from the star, i.e:

$$\frac{GM_H M_{sat}}{r_{sp}^2} = k \frac{GM_* M_{sat}}{r_{ss}}$$

Where  $r_{sp}$  is the satellite-planet distance,  $r_{ss}$  is the satellite-star distance and  $M_H$  is the mass of the target planet, Hiffre. Solving, gives:

$$r_{sp} = r_{ss} \sqrt{\frac{M_H}{M_*} \frac{1}{k}} \quad (27)$$

We find numerically that choosing  $k = 10$  gives a sufficiently stable orbit over multiple years.

Now that we know the distance to the planet at which we wish to arrive, we need to find what boost (direction and magnitude) we need to give the satellite in order to achieve a stable circular orbit around our target planet. The magnitude is easily found by equating centripetal acceleration to the gravitational force:

$$G \frac{M_H M_{sat}}{r_{sp}^2} = M_{sat} \frac{v_{SO}^2}{r_{sp}}$$

Where  $v_{SO}$  is the speed of the satellite *relative to the planet* necessary for a stable orbit. Solving for the speed gives:

$$v_{SO} = \sqrt{\frac{GM_H}{r_{sp}}} \quad (28)$$

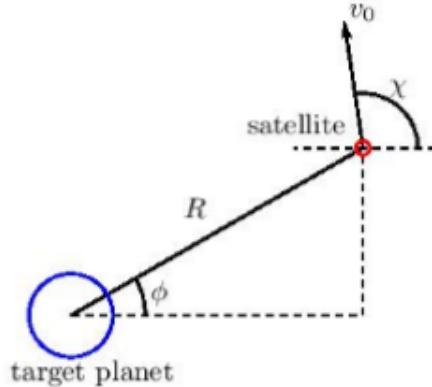


Figure 3: The geometry of VoyagerX’s approach to Hiffre. Image taken from [Hansen and Hagala, 2016a], all courtesy go to the original creator.

The direction can be found from geometric considerations. Consider the situation shown in figure 3

Let us assume that the satellite orbits clockwise (otherwise we must simply flip all signs). Note that, for circular motion, the velocity must be orthogonal to the position vector. Consider the situation where  $\phi = 0$ . Then it is clear that all velocity must be in the  $y$ -direction. On the other hand, if  $\phi = \pi/2$ , all velocity must be in the *negative*  $x$ -direction. From this, and elementary trigonometry, it follows that the velocity of the satellite in orbit must be:

$$\vec{v}_{SO} = \begin{pmatrix} -v_{SO} \sin \phi \\ v_{SO} \cos \phi \end{pmatrix}$$

To find the boost required, we must find the *change* in velocity,  $\Delta v_{SO}$ . This is given by the final velocity of the satellite,  $\vec{v}_{SO}$ , minus the initial velocity of the satellite,  $\vec{v}_0$ . This velocity can easily be found by decomposing the vector  $\vec{v}_0$  in figure 3. By a similar argument as earlier,  $v_0$  is purely in the  $x$ -direction if  $\chi$  is zero, and it is purely in the  $y$ -direction if  $\chi$  is  $\pi/2$ . Thus:

$$\vec{v}_0 = \begin{pmatrix} v_0 \cos \chi \\ v_0 \sin \chi \end{pmatrix}$$

Thus subtracting  $\vec{v}_0$  from  $\vec{v}_{SO}$  gives the direction and magnitude of the boost:

$$\Delta \vec{v}_{circ} = \begin{pmatrix} -v_{SO} \sin \phi - v_0 \cos \chi \\ v_{SO} \cos \phi - v_0 \sin \chi \end{pmatrix} \quad (29)$$

The velocities of the planets can easily be found with a forward Euler scheme, as described in section A.3. Note, however, that this is all relative to the planet.

In our system, however, the sun is at rest at the origin and all planets move. Therefore, we must implement that  $\vec{v}_0 = \vec{v}_{sat} - \vec{v}_H$ , where  $\vec{v}_{sat}$  is the velocity of the satellite and  $\vec{v}_H$  is the velocity of the target planet Hiffre. Note further, that the angles are also given relative to the planet. We can find  $\phi$  by realizing that:

$$\tan \phi = \frac{y_{sat}}{x_{sat}}$$

Where  $x_{sat}$  and  $y_{sat}$  are the components of the relative position,  $\vec{r}_{SH}$  between the satellite and Hiffre, given by  $\vec{r}_{SH} = \vec{r}_S - \vec{r}_H$ , where  $\vec{r}_S$  is the position of the satellite and  $\vec{r}_H$  is the position of Hiffre. Thus we can compute the vector  $\vec{v}_{SO}$ . Once this has been found, we can easily compute  $\Delta v_{circ}$ , without computing  $\chi$ , as we can simply compute  $\vec{v}_0 = \vec{v}_{sat} - \vec{v}_H$ , and then subtract  $\vec{v}_0$  from our calculated  $\vec{v}_{SO}$ .

### Being close enough to take a picture

Assume our satellite takes pictures with a resolution of  $P \times P$  pixels. We denote the field of view of our camera by  $F$ , and the radius of Hiffre by  $R_H$ . For the planet to show up in more than a singel pixel, the angular separation of the top and the bottom of the planet must be larger than one pixel. The height of each pixel is given by  $F/P$ , which therefore is the size we must compare our angular separation against. Drawing a triangle with our line of sight and the radius of Hiffre as sides, one easily sees that the following holds true for the angular separation  $\theta$ :

$$\tan \theta = \frac{R_H}{r}$$

Where  $r$  is the distance to Hiffre at which we take the picture. Now we use that, for small angles,  $\tan \theta \approx \theta$ , to get the angular separation as  $\theta \approx R_H/L$ . Equating these expressions for the angular separation gives:

$$\frac{F}{P} \approx \frac{R_H}{r} \implies r \approx \frac{R_H P}{F} \quad (30)$$

Thus, if the distance is lower than this, Hiffre will show up in more than one pixel. We know from our engineers that the field of view of our camera,  $F$  is approximately  $70^\circ$ , and that the number of pixels,  $P$  is approximately 2000. Note carefully that the small-angle approximation is valid only for radians. Therefore, we must convert  $F$  to radians. Doing the calculation shows that the distance given in equation 27 is smaller than the distance given in equation 30. Thus, we adopt the  $r$  from equation 27 as our target distance to Hiffre. This is done more explicitly in section 4.3.

#### 2.3.4 Computing the power required by our landing module

VoyagerX has a landing module, HAL9001, which we will launch from VoyagerX, to actually land on Hiffre. We will power this module with solar panels. We there-

fore need to know the size of solar panels that are required to power the module.

We known from the manufacturer that the solar panels have an efficiency of 12 %. We also know, from equation 14, what power the satellite receives from the star at a distance  $r$  from the star. Assume the solar panels have an area of  $A$ . Then, the total power incident on the solar panels is:

$$P_{\text{in,solar}} = AF_P = \frac{A\sigma R_*^2 T_*^4}{r^2}$$

However, only 12 % of this is useful power. We therefore introduce an efficiency factor,  $e$ , which is a dimensionless parameter describing the efficiency of the solar panels. Furthermore, we know that the lander requires a power,  $P_{\text{required}}$  of at least 40 W to operate. We can then find the required area as:

$$A = \frac{P_{\text{required}} r^2}{e \sigma R_*^2 T_*^4} \quad (31)$$

### 2.3.5 Methods

#### Computing the total boost

An important realization is that the two boosts we plan to perform are both in the direction of the velocity vector. From this, it follows that we can add these boost together, and simply leave our planet with a total velocity change,  $\Delta v$  given by the sum of equation 22 and 23, along the direction of the velocity vector.

#### Adjusting our parameters in the simulation

In section 2.3.1 to 2.3.3, we have made a significant number of simplifications. We must therefore conduct numerical experiments to find the correction factors required for both the start time,  $t_0$  and the boosts,  $\Delta v$ .

Ideally, we want as low  $\Delta v$  as possible, to minimize fuel usage. However, we need enough  $\Delta v$  to hit the planet. Ideally, we should hit the target planet with a low velocity, to get into a circular orbit without having to reverse our velocity. Thus  $t_0$  and  $\Delta v$  should be chosen such that  $\Delta v$  is as low as possible, so that we pass behind the target planet with as low velocity as possible. This gives us a way to check our chosen correction factors. Thus we guess a set of correction factors, and adjust them up or down, each time evaluating if we have gotten closer to our ideal situation or not.

#### Implementing safety margins for the real launch

Whilst it is possible to adjust  $\Delta v$  and  $t_0$  so that the satellite hits perfectly in our

simulation, this is a waste of time, as there may be errors in our real launch which prevent us from reaching the required precision. We therefore implement a safety margin: we make an additional, small, boost,  $\Delta v_b$  in the direction of the planet once we are less than 0.01AU away from the planet. This small boost ensures that we hit the planet, and do not fly past it. The direction of the boost should be in the general direction of the planet. However, by the time the satellite arrives there, the planet will have moved. Therefore we boost towards where the planet will be at time  $\Delta t_b$  from now instead. The magnitude of  $\Delta v_b$  and  $\Delta t_b$  can be determined by numerical experiments, as described in the previous section.

One additional safety feature which we implement is that we add enough fuel to boost an additional 10% of the total velocity, i.e. we compute  $\Delta v$ ,  $\Delta v_b$  and  $\Delta v_{circ}$  and then let our fuel estimate be given by the amount of fuel necessary to reach a velocity  $v = 1.1(\Delta v + \Delta v_b + \Delta v_{circ})$ . This ensure that we have enough fuel to fine-tune our position in the real launch.

### Numerical considerations

For this part of the project, we use different timesteps at different positions. Clearly, we need small timesteps whenever there is large acceleration. Thus, we use small timesteps close to the home planet and the target planet. Once we are more than 0.1 Au away from either planets, however, we use larger timesteps, to reduce computation time.

## 2.4 Orientating the satellite in interplanetary space

Orientation in interplanetary space is not an easy task, due to the isotropic nature of space - there are few waypoints. For the real launch, however, we will need to find our position, velocity and orientation at any point in time. In this section, we will discuss how to do this.

### 2.4.1 Finding the orientation of the satellite in interplanetary space

Our satellite comes equipped with a camera, with which it can take pictures in the forward direction. By comparing these pictures with the pictures of the sky taken from our home planet, we can find the orientation of the satellite, expressed in terms of the angles  $\phi$  and  $\theta$ , in spherical coordinates, measured from our home planet. We here adapt the definition of these angles commonly used in physics:  $\theta$  defines the polar angle and  $\phi$  defines the azimuthal angle.

The image returned by the satellite is a stereographic projection of the celes-

tial sphere that the satellite sees, i.e. it is a projection of a sphere unto a square. We will compare these to stereographic projections made from our home planet. In general, stereographic projections work by mapping each point of the celestial sphere that is visible to the observer (with a specified field of view) unto the tangent plane, centered at some coordinates  $\theta_0, \phi_0$ . As our satellite will be travelling in a two-dimensional plane, we will assume the satellite to be in the equatorial plane, i.e.  $\theta_0 = \pi/2$  for all projections<sup>13</sup>. We can then define tangent plane coordinates,  $x_{\text{picture}}$  and  $y_{\text{picture}}$ , where the origin of this coordinate system coincides with  $(\theta_0, \phi_0)$ . This is the grid on which our pixels will fall. Note that,  $\theta$  increases as we go further down the sphere. Furthermore, pixel entries, like matrix elements, are counted positively downwards (i.e. in the y-direction). Therefore, it makes sense to flip our y-axis, so that it goes from low values to high values. We will assume this to be the case in the rest of this section.

The mathematical transformation between  $x_{\text{picture}}$ ,  $y_{\text{picture}}$  and  $\theta, \phi$  are shown in [Hansen and Hagala, 2016b]. They are given by:

$$x_{\text{picture}} = k \sin \theta \sin(\phi - \phi_0) \quad (32)$$

$$y_{\text{picture}} = k(\sin \theta_0 \cos \theta - \cos \theta_0 \sin \theta \cos(\phi - \phi_0)) \quad (33)$$

Where:

$$k = \frac{2}{1 + \cos \theta_0 \cos \theta + \sin \theta_0 \sin \theta \cos(\phi - \phi_0)}$$

From these equations, we can find the maximum and minimum values of  $x_{\text{picture}}$  and  $y_{\text{picture}}$ , as a function of our maximum field of view in the  $\theta$  direction,  $\alpha_\theta$ , and our maximum field of view in the  $\phi$  direction,  $\alpha_\phi$ . This will enable us to determine our grid of  $x_{\text{picture}}$  and  $y_{\text{picture}}$  values. This is done in detail in appendix B.2, where we show that:

$$x_{\text{picture,max/min}} = \pm \frac{2 \sin(\alpha_\phi/2)}{1 + \cos(\alpha_\phi/2)} \quad (34)$$

$$y_{\text{picture,max/min}} = \pm \frac{2 \sin(\alpha_\theta/2)}{1 + \cos(\alpha_\theta/2)} \quad (35)$$

Where we once again stress that  $y_{\text{picture,max}}$  is at the *bottom* of the square.

With  $x_{\text{picture,max}}$  and  $y_{\text{picture,max}}$  established, we can now make a grid with the same number of pixels as the image we receive from the satellite, and with the angular differences in the  $xy$  direction determined by equations 34 and 35. We can transform the  $xy$  values of this grid back into spherical coordinates  $\theta, \phi$ , by using

---

<sup>13</sup>The observant reader may object, but as we launch from equator, and all our models are two-dimensional, this is a reasonable assumption.

the inverse transformations of equations 32 and 33. The detailed mathematical treatment of this can be found [HERE](#). We simply state the result:

$$\theta = \frac{\pi}{2} - \arcsin \left( \cos c \cos \theta_0 + \frac{y_{\text{picture}} \sin c \sin \theta_0}{\rho} \right)$$

$$\phi = \phi_0 + \arctan \left( \frac{x_{\text{picture}} \sin c}{\rho \sin \theta_0 \cos c - y_{\text{picture}} \cos \theta_0 \sin c} \right)$$

With:

$$\rho = \sqrt{x_{\text{picture}}^2 + y_{\text{picture}}^2}, \quad c = 2 \arctan \left( \frac{\rho}{2} \right)$$

Note that these equations simplify significantly in our case, as  $\theta_0 = \pi/2$ . This gives finally, for the inverse transformations:

$$\theta = \frac{\pi}{2} - \arcsin \left( \frac{y_{\text{picture}} \sin c}{\rho} \right) \quad (36)$$

$$\phi = \phi_0 + \arctan \left( \frac{x_{\text{picture}} \tan c}{\rho} \right) \quad (37)$$

Now we have all equations required to find the orientation of our satellite. The method can be briefly summarized as:

- Make an  $xy$  grid centered at  $\theta = \pi/2$  and  $\phi$  given by a chosen  $\phi_0$ . Determine the max/min values of  $xy$  from equation 34 and 35. Determine the number of grid points from the number of pixels in the image from the satellite.
- Compute the angles  $\phi$  and  $\theta$  corresponding to each point  $x$  and  $y$  from equation 36 and 37.
- Insert the sliver of the celestial sphere (as seen from the home planet) corresponding to each  $\theta$  and  $\phi$  at the corresponding point in the  $xy$  grid.
- Repeat this process for different  $\phi_0$  to get a sufficient number of projections of the celestial sphere (in our case we let  $\phi_0 \in [0^\circ, 360^\circ]$ ) to cover the celestial sphere with a relatively coarse grid).
- Use a least square method as described in appendix A.2, comparing each of the images of the celestial sphere as seen from the home planet with the image from the satellite. Then find the image which fits best.
- Determine the  $\phi_0$  (and  $\theta_0$ ) corresponding to the best-fit image, and use this as an (hopefully) excellent approximation to the orientation of the satellite.

### 2.4.2 Finding the position of the satellite in interplanetary space

We have a radar aboard our satellite, which enables us to determine the distance to all planets, as well as the sun. We can then use a version of a triangulation algorithm to find our position.

We know the distance from the satellite to the planet/star,  $r$ , and thus we know that the satellite is somewhere on a circle with radius  $r$  around the celestial object. By using three different celestial objects, we should be able to find the point where all three circles intersect, which will be the position of the satellite..

The general equations for tree circles is given by:

$$(x - a)^2 + (y - b)^2 = r_1^2 \quad (38)$$

$$(x - c)^2 + (y - d)^2 = r_2^2 \quad (39)$$

$$(x - e)^2 + (y - f)^2 = r_3^2 \quad (40)$$

These three equations can be solved analytically to find the intersection point. We begin by solving equation 38 and 39. The details can be found in appendix B.3, but it results in the following equations for  $x$  and  $y$ :

$$y_{1,2} = \frac{-2(\gamma\lambda - b) \pm \sqrt{4(\gamma\lambda - b)^2 - 4(\lambda^2 + 1)(\gamma^2 + b^2 - r_1^2)}}{2(\lambda^2 + 1)} \quad (41)$$

$$x_{1,2} = \lambda y + \gamma + a \quad (42)$$

Where:

$$\gamma = \frac{r_1^2 + c^2 + d^2 - a^2 - b^2 - r_2^2}{2(c - a)} - a$$

$$\lambda = \frac{b - d}{c - a}$$

The points  $(x_1, y_1)$  and  $(x_2, y_2)$  can then be inserted into equation 40. The points which satisfy equation 40 (or, due to numerical errors, almost satisfy it) give the position of the satellite.

Note that there is one potential problem with this method: if any of the three circles are collinear, the method fails, as it is no longer possible to find a unique tangent point. To accommodate for this, we test with the sun and multiple planets, discarding the solutions where both  $(x_1, y_1)$  and  $(x_2, y_2)$  give an answer close to 0 in equation 40. Specifically, we always test with the sun, and then iterate through all the other planets, taking pairs of two.

We test this method by choosing 1000 random positions in our solar system, and letting the algorithm find these positions.

### 2.4.3 Finding the velocity of the satellite in interplanetary space

The velocity of the Satellite can be determined by finding the position of the  $H_{\alpha}$  spectral line, which is usually found at 656.3 nm, in the spectral images from the satellite. This will give us a wavelength shift,  $\Delta\lambda$ , which, by the standard Doppler shift formula, is related to the velocity of the reference stars as:

$$v_{star} = \frac{\Delta\lambda}{\lambda}c \quad (43)$$

Where  $c$  is the speed of light. Note that this will give the velocity of the star relative to the satellite. We can alternatively employ the same formula to find  $v_{star}$  as seen from our star (origin of coordinate system), by inputting the  $\Delta\lambda$  seen from our star. By comparing the values of  $v_{star}$  found by the satellite with the values found from the star, we can find the velocity of our satellite.

The radial velocity,  $v_{star,sat}$  of a reference star as seen from the satellite (found from equation 43) can be used to find the velocity of the satellite,  $v_{sat}$  (as seen from our home planet) in the direction of the star, by employing:

$$v_{sat} = v_{star,home} - v_{star,sat}$$

Where  $v_{star,home}$  is the velocity of the star seen from our home planet. Thus, we can find the radial velocity of the satellite, as seen from our star, in two different directions - the direction of the two reference stars. However, we *know* the position of the stars, as seen from the star - they are at angles  $\phi_1$  and  $\phi_2$  respectively. Thus we can define units vectors in the direction of the stars, which will be linearly independent (and therefore span the entire solar system) *as long as the stars are not collinear*<sup>14</sup>. Let  $v_{sat,1}$  and  $v_{sat,2}$  denote the radial velocity of the satellite in the direction of the reference star 1 and 2 respectively, as seen from our star, and let  $v_{sat,x}$  and  $v_{sat,y}$  denote the (unknown) velocity components of the satellite, as seen from our reference star, in the standard coordinate system. We show in appendix B.4 that the velocity of the satellite can then be found as:

$$\begin{pmatrix} v_{sat,x} \\ v_{sat,y} \end{pmatrix} = \frac{1}{\sin(\phi_2 - \phi_1)} \begin{pmatrix} \sin \phi_2 & -\sin \phi_1 \\ -\cos \phi_2 & \cos \phi_1 \end{pmatrix} \begin{pmatrix} v_{sat,1} \\ v_{sat,2} \end{pmatrix} \quad (44)$$

Thus, if we know the angles of the reference stars and the radial velocity of the satellite in the direction of our reference stars (relative to our star) we can find the velocity of the satellite in xy coordinates.

---

<sup>14</sup>If the stars *are* collinear, we will have to choose a different set of reference stars, as  $\phi_1$  and  $\phi_2$  are (almost) time-independent

#### 2.4.4 Orientating the satellite: Methods

We present, in Pseudo-code, how we implement the least-square method for finding the best image:

```
1 diff= 100000000 #Arbitrary large number
2 j=0
3 for k in reference_sphere_images:
4     diff_img=(image-reference_sphere[k])**2
5     least_square=np.sum(diff_img)
6     if least_square < diff:
7         diff = least_square
8         j=k
9 print j
```

### 2.5 Investigating Hiffre, the target planet

Before we can attempt a safe landing on Hiffre, we need an excellent model for its atmosphere, to avoid a potential catastrophe during landing. We can make some simplifying assumptions for the atmosphere<sup>15</sup>.

- We assume our atmosphere to rotate perfectly with Hiffre
- We assume the atmosphere to be adiabatic up to a temperature  $T_0/2$ , where  $T_0$  is the surface temperature of Hiffre, and isothermal after this point.
- We assume Hiffre to have no variation of surface temperature,  $T_0$  at different longitudes/latitudes.
- We assume the percentage composition of gases that we find in our atmosphere to be equal.
- We assume Hiffre's atmosphere to be in hydrostatic equilibrium.
- We assume all gases in Hiffre's atmosphere to be ideal

We then build an accurate model for this simplified atmosphere. For this, however, we require the mean molecular mass of Hiffre's atmosphere, as well as its surface temperature. We can find the surface temperature from equation 15.<sup>16</sup>

---

<sup>15</sup>These may seem unrealistic (and they are), however, they simplify calculation and we hope that they will be sufficiently realistic.

<sup>16</sup>The attentive reader may notice that we assumed our planet to have no atmosphere when deriving this equation, whereas we here use this temperature to model the atmosphere. This observation is correct, however, this is the only model of  $T_H$  available.

Inserting values gives that the surface temperature of Hiffre,  $T_H = 216\text{K}$ . In this section, we describe how to find the mean molecular weight,  $\mu$ , of the atmosphere from spectroscopic measurements, and how to use this value to model Hiffre's atmosphere.

### 2.5.1 Using spectroscopy to determine molecular mass

The gases present in Hiffre's atmosphere will absorb some of the electromagnetic radiation from the ground. This will result in absorption lines, which we should be able to detect with spectroscopy. If we can detect an absorption line, and assign it a wavelength, we will be able to determine what gases are present in Hiffre's atmosphere, as the absorption line from each gas is characteristic of that gas. Once this is known, we can determine the mean molecular weight,  $\mu$ , of our atmosphere. However, there are multiple interfering factors which may make this task difficult.

One of the main problem, is that our instrument is not perfect, and that some of our assumptions about the atmosphere may be inaccurate. As a result, we will have a significant amount of noise in our data, which leads to multiple problems. First and foremost; the noise makes it difficult to detect lines. Additionally, however, the noise may result in false positives.

Additional problems arise from the fact that our satellite is moving relative to Hiffre's atmosphere, and that different parts of Hiffre's atmosphere are moving at different speeds relative to us. The relative difference in motion will mean that the spectral line is observed at numerous wavelengths, and the motion of the satellite may Doppler shift the entire absorption line.

With all these factors considered, we assume our spectral lines to have a Gaussian spectrum (due to varying velocity of the gases in our atmosphere), with a through close to (but not necessarily at) the expected absorption line. More exact thermodynamical calculations show (according to [Hansen, 2016b], give that a good model for the flux,  $F$ , as a function of wavelength,  $\lambda$  at an absorption line is:

$$F(\lambda) = F_{\text{cont}}(\lambda) - (F_{\text{min}} - F_{\text{cont}}(\lambda))e^{-(\lambda-\lambda_0)/(2\sigma^2)} \quad (45)$$

Where  $F_{\text{cont}}$  is the continuum flux (the flux when there is no absorption line present),  $F_{\text{min}}$  is the minimum flux (depth of the absorption line),  $\lambda_0$  is the wavelength at which we expect absorption, and  $\sigma$  is a parameter which classifies the width of the spectral line. Our satellite normalizes the continuum flux to 1. Thus  $F_{\text{cont}} = 1$ .  $\lambda_0$  depends on the gas we are looking for. Standard values for the position of spectral lines can be found for example [here](#). Note that, as earlier stated,

$\lambda_0$  may not correspond exactly to these values, due to the Doppler shift.

From thermodynamic considerations, it is clear that  $\sigma$  should depend on temperature (as the gases move faster, they have a larger range of Doppler shift). Careful calculations, explained in [Hansen, 2016b], show that the Gaussian profile specified above has a FWHM (full width at half maximum), given by:

$$\text{FWHM} = \frac{2\lambda_0}{c} \sqrt{\frac{2kT \ln 2}{m}} \quad (46)$$

Where  $c$  is the speed of light and  $m$  is the mass of the molecule we are looking for. As shown by [Hansen, 2016a], FWHM is related to the standard deviation,  $\sigma$ , of a Gaussian distribution as:

$$\sigma = \frac{\text{FWHM}}{\sqrt{8 \ln 2}} \quad (47)$$

To implement the model specified by equation 45, we need to find the optimal combination of the parameters explained above. This is done using the  $\chi^2$  algorithm, as detailed in appendix A.2. Note that this algorithm requires an estimate for the standard deviation of the noise of each measurement,  $\sigma_\chi$ . Luckily, VoyagerX has excellent equipment on board to estimate this. However, we still need to decide on an appropriate range of the parameters specified in equation 45.

We know that our satellite, VoyagerX, has a maximum velocity relative to Hiffre,  $v_{\text{VoyagerX,max}}$ , of 10km/s. Using the Doppler shift formula (equation 43), we can then determine the maximum shift in  $\lambda_0$ ,  $\Delta\lambda_0$ , simply as:

$$\Delta\lambda_0 = \frac{\lambda_0}{c} v_{\text{VoyagerX,max}}$$

Thus letting  $\lambda_0$  vary from  $\lambda_0 - \Delta\lambda$  to  $\lambda_0 + \Delta\lambda$  ensures that we hit all possible  $\lambda_0$ .

We can estimate  $F_{min}$  simply by realizing that it is extremely unlikely for the absorption line to be much larger than the continuum (background) flux. We therefore estimate that  $F_{min} \in [0.7, 1]$ .

We also need to find a range for  $\sigma$ , the possible widths of the spectral line. To do this, we assume that all gases in Hiffre's atmosphere have a temperature between 150 K and 450 K<sup>17</sup>. Inserting these temperatures into equation 46, gives us an estimate of the FWHM, which is directly proportional to  $\sigma$ . Here some care

---

<sup>17</sup>This is in all probability a reasonable estimate, as the surface temperature was shown earlier to be about 216 K

is required, however. The boundaries for  $\sigma$  are not sharp, due to the Gaussian nature of the Doppler shift in the atmosphere. Therefore, we choose a larger range for  $\sigma$ , to include potentially broader lines. After some trial and error, we decide to let

$$\sigma \in [\text{FWHM}_{low}/20, 2\text{FWHM}_{high}]$$

Where  $\text{FWHM}_{low}$  is the FWHM for the lower boundary of temperature (150 K) and  $\text{FWHM}_{high}$  is the FWHM for the upper boundary of the temperature (450 K).

With all these estimates for the parameters in place, we can then use a multi-dimensional  $\chi^2$ , trying all possible combinations of the parameters  $\sigma$ ,  $\Delta\lambda$  and  $F_{min}$ , and extracting the one for which the  $\chi^2$  is the smallest. We can then compute the temperature of the atmosphere (by combining equation 46 with 47 and solving for  $T$ ) and the velocity of VoyagerX (from equation 43). Then, we can combine the visual fit to the spectrum and the  $\chi^2$  values with the information about the predicted velocity of VoyagerX and the temperature of the atmosphere, to establish which absorption lines correspond to actual gases, and which are flukes.

Once it has been established what gases are present in the atmosphere, the mean molecular mass can be easily computed. By assumption, the percentage composition of gases in the atmosphere are equal, therefore the mean molecular mass is simply:

$$\mu = \frac{\sum_{i=1}^N m_i}{N} \quad (48)$$

Where  $N$  represents the number of different gases present in our atmosphere and the  $m_i$  are the molecular masses corresponding to these gases.

### 2.5.2 Method for determining the mean molecular mass

As noted in the section about  $\chi^2$  (appendix A.2), we implement a brute force approach to find the optimal combination of parameters. This means that we must severely limit ourselves in terms of how many points we check.

The first question that we need to address is how much of the spectrum we want to fit to. This is the innermost sum in the  $\chi^2$  algorithm, and therefore the most significant one. We expect that the relative variations of wavelength increase with  $\lambda_0$ , according to equation 46. Our spectrum starts at 600nm, and we find that approximately 0.1nm on each side of the spectrum is sufficient to capture  $\lambda_0$ . We therefore implement the condition that the variations to either side of the expected

wavelength,  $\lambda_0$  that we investigate is given by:

$$\delta\lambda = \frac{0.1}{600}\lambda_0$$

We test 30 values of  $\sigma$ , 30 values of  $F_{min}$  and 300 values of  $\lambda_0$ . The algorithm is shown in Pseudo-code below:

```

1  smallest_chi=10000000000 #Arbitrary , large , number
2  for i in range(F_min):
3      for j in range(sigma):
4          for k in range(lambda_center):
5              chi_square[i,j,k]=np.sum((data-model)**2/(noise[i,j,k])**2)
6
7
8  for i in range(F_min):
9      for j in range(sigma):
10         for k in range(lambda_center):
11             if chi_square[i,j,k] < smallest_chi:
12                 smallest_chi=chi_square[i,j,k]
13                 index_F=i
14                 index_sigma=j
15                 index_lambda=k

```

Note that the spectrum is initially given as a text file. For computational efficiency, this should be converted to an .npy file.

### 2.5.3 Modelling the atmosphere

As mentioned in the beginning of this section, we model the atmosphere as being adiabatic up to the distance from the planet,  $r_{T_0/2}$ , where the temperature of the atmosphere equals half the surface temperature of Hiffre,  $T_0$ . After this point, we model the atmosphere as isothermal. This is a common model of an atmosphere. As described by [Fitzpatrick, 2006], at low altitudes, the air is constantly moving (due to winds and small scale temperature-differences). Thus the air is well mixed, and we may assume a packet of air to be at mechanical equilibrium (same pressure) with its surrounding. Furthermore, as air is a poor heat conductor, we may assume that it does not exchange any heat with the surrounding. This leads to an adiabatic model of the atmosphere for low altitudes, which can be described as:

$$P^{1-\gamma}T^\gamma = A \quad (49)$$

Where  $P$  is the pressure of the atmosphere,  $T$  is the temperature and  $A$  is a constants.  $\gamma$  adiabatic index, which for our atmosphere is given by:

$$\gamma = 1.4$$

At higher altitudes, as explained by [Fitzpatrick, 2006], the air is less well mixed. Therefore pressure can drop more quickly. We therefore describe the atmosphere at high altitudes by an isothermal model, i.e. by:

$$T = B \quad (50)$$

Where  $B$  is a constant. The transition point at  $r = r_{T_0/2}$  is chosen, as it is in-between these two regimes. We combine equation 49 and 50 with two further assumptions stated in the beginning of this section: the assumption that Hiffre's atmosphere is in hydrostatic equilibrium, and the assumption that all gases in the atmosphere are ideal. If hydrostatic equilibrium holds, then it is shown by [Hansen, 2016c] that the pressure,  $P$ , obeys the following equation:

$$\frac{dP}{dr} = -\frac{G\rho(r)M_T}{r^2} \quad (51)$$

Where  $\rho$  is the density,  $r$  is the distance from core of the planet and  $M_T$  is the mass of Hiffre. Furthermore, if the gases are ideal, they obey the ideal gas law:

$$P = \frac{\rho k T}{\mu u} \quad (52)$$

Where  $u$  is the atomic mass unit. With all these equations in place, we can now develop a model for the pressure, temperature and density of Hiffre's atmosphere.

### Analytic solution

We can solve equation 49 and 50 independently. As shown in appendix B.5, this leads to the following equations for the adiabatic part of the atmosphere:

$$\begin{aligned} T_{\text{adiabatic}} &= \frac{\gamma - 1}{\gamma} \frac{GM\mu u}{k} \frac{1}{r} + C_1 \\ P_{\text{adiabatic}} &= AT^{\frac{\gamma}{\gamma-1}} \\ \rho_{\text{adiabatic}} &= \frac{\mu u}{k} AT^{\frac{\gamma}{\gamma-1}} \end{aligned} \quad (53)$$

Where  $C_1$  and  $A$  are constants. For the isothermal region, the corresponding equations are:

$$\begin{aligned} T_{\text{isothermal}} &= T_0/2 \\ P_{\text{isothermal}} &= C_2 \exp(2GM\mu u / rkT_0) \\ \rho_{\text{isothermal}} &= \frac{2\mu u}{kT_0} C_2 \exp\left(\frac{2GM\mu u}{rkT_0}\right) \end{aligned} \quad (54)$$

Where  $C_2$  is a constant. The constants  $A$ ,  $C_1$  and  $C_2$  must be fixed by boundary conditions. One such condition is the fact that the temperature at the surface of

the planet,  $T_0$ , is known from equation 15. Thus, we require that  $T_{\text{adiabatic}} = T_0$  when  $r = r_p$ , where  $r_p$  is the planet radius. Furthermore, the density at the surface of the planet,  $\rho_0$ , is known from previous missions. Thus we require that the density,  $\rho_{\text{adiabatic}}$  equals  $\rho_0$  at  $r = r_p$ . This fixes  $A$  and  $C_1$ . To fix  $C_2$ , we impose a continuity condition: we require that the density is continuous at the transition point,  $r_{T_0/2}$ , i.e. that  $\rho_{\text{adiabatic}} = \rho_{\text{isothermal}}$  at  $r = r_{T_0/2}$ .

We show how to implement these boundary conditions in appendix B.5.3. With all of this in place, this leads, finally, to the following model for Hiffre's atmosphere:

$$T(r) = \begin{cases} \frac{\gamma-1}{\gamma} \frac{GM\mu u}{k} \left( \frac{1}{r} - \frac{1}{r_p} \right) + T_0 & \text{for } r \leq r_{T_0/2} \\ T_0/2 & \text{for } r > r_{T_0/2} \end{cases} \quad (55)$$

$$P(r) = \begin{cases} \frac{1}{\mu u T_0^{1/(\gamma-1)}} \rho_0 k T(r)^{\frac{\gamma}{\gamma-1}} & \text{for } r \leq r_{T_0/2} \\ \frac{k T_0 \rho_{T_0/2}}{2 \mu u} \exp \left( GM \mu u \left( \frac{1}{r k T} - \frac{2}{r_{T_0/2}} \right) \right) & \text{for } r > r_{T_0/2} \end{cases} \quad (56)$$

$$\rho(r) = \begin{cases} \rho_0 \left( \frac{T}{T_0} \right)^{1/(\gamma-1)} & \leq r_{T_0/2} \\ \rho_{T_0/2} \exp \left( \frac{2GM\mu u}{k T_0} \left( \frac{1}{r} - \frac{1}{r_{T_0/2}} \right) \right) & \text{for } r > r_{T_0/2} \end{cases} \quad (57)$$

Where  $\rho_{T_0/2}$  is the density at the transition point,  $r = r_{T_0/2}$ . by solving the temperature equation for  $T(r) = T_0/2$ , we can explicitly write out  $r_{T_0/2}$  as:

$$r_{T_0/2} = \left( \frac{1}{r_p} - \frac{T_0}{2} \frac{\gamma}{\gamma-1} \frac{k}{GM\mu u} \right)^{-1} \quad (58)$$

These are the equations which we use to model our atmosphere.

## Numerical solution

We also implement a numeric model of our atmosphere. This is mainly to check that the rather convoluted equations above have been solved and implemented correctly. This is significantly easier to implement, as we can simply solve equation 51 and 52 iteratively, using the appropriate expression for the atmosphere, dependent upon temperature. A pseudo-code for this is shown below. Note that the constant  $A$  can be determined if  $T_0$  and  $P_0$  are known.

```

1 P_0=(self.k*self.rho_0*self.T_0)/(self.mu*self.atomic_mass_unit)
2 adiabatic_constant=(P_0**((1-self.gamma)))*(self.T_0**((self.gamma)))
3 while T > self.T_0/2.0:
4     P+=G*m*rho[-1]/(float(r)**2)*dr
5     T=adiabatic_constant/(P**((1-self.gamma)))*((1.0/self.gamma))
6     rho.append(ideal_gas_constant*(P/float(T)))
7     r+=dr
8
9 while rho[-1]>1e-14: #As far out as we want to go
10    P+=mu_gravity*rho[-1]/(float(r)**2)*dr
11    rho.append(ideal_gas_constant*(P/float(T)))
12    r+=dr

```

## 2.6 Landing on our target planet

Having developed a model for our atmosphere, we can now simulate the actual landing. Note that, as discussed in section 2.2.2, we will now switch back to SI units. We will also discard our 2 dimensional approximation, and simulate fully 3D, assuming VoyagerX to have an initial velocity and position in the xy-plane. We will ensure that we get satisfactory results from our simulations before we attempt to land on Hiffre. In this section we present how we model the force from the atmosphere on the satellite, and how we get into a close planetary orbit, given that we are in a orbit around Hiffre with radius dictated by equation 27. We then discuss how to slow down our descent, to ensure a smooth landing on the surface of Hiffre. Finally, we discuss how to pick out and land on a specific spot on Hiffre.

### 2.6.1 Modelling the force from our atmosphere

Now that we know how the density of the atmosphere varies with the distance,  $r$ , modelling the atmosphere is relatively straightforward. As stated in section 2.5.3, we assume our atmosphere to rotate uniformly with our planet. The angular velocity of this rotation can be found once the rotational period of Hiffre,  $T_H$  is known. This is known from earlier observations, to be about 13.5 days. Then we can find the angular velocity of Hiffre,  $\omega_H$ , as  $\omega_H = 2\pi/T_H$ . Letting the  $z$ -axis point along the poles of Hiffre, the velocity of the atmosphere,  $\vec{v}_A$ , at a distance  $\vec{r}$  from the center of Hiffre can be found from:

$$\vec{v}_A = \omega_H \mathbf{k} \times \vec{r} \quad (59)$$

We model the force from the atmosphere on the satellite with the drag equation<sup>18</sup>, which is given by:

$$\vec{F} = -\frac{1}{2}\rho(r)C_D A |\vec{v}_{rel}| \vec{v}_{rel} \quad (60)$$

---

<sup>18</sup>Note that this assumes the Reynolds number of our atmosphere to be large enough to produce turbulence

Where  $\vec{F}_D$  is the drag fore,  $C_D$  is a drag coefficient,  $A$  is the surface area of VoyagerX (and later of the lander module, HAL9001), which can be found in appendix C, and  $\vec{v}_{rel}$  is the relative velocity. We will assume that  $C_D = 1$ , as this has been shown, by previous missions through our solar system, to be a good approximation for the atmosphere of Hiffre. We can find  $\vec{v}_{rel}$  as  $\vec{v}_{rel} = \vec{v}_{VoyagerX} - \vec{v}_A$ , where  $\vec{v}_{VoyagerX}$  is the velocity of the satellite relative to a stationary Hiffre.

The engineers responsible for our satellite VoyagerX, and our lander module HAL9001, advised that the forces on them should not exceed 25000 N, and guaranteed that they will fail if the force ever exceeds 250 000 N. Thus, we will have to ensure that the combined effect of the gravitational force from Hiffre and the drag force from Hiffre's atmosphere stays below this number at all time.

### 2.6.2 Getting into a close orbit around Hiffre

One way of ensuring that the force stays low is to let VoyagerX orbit at the outer rim of the atmosphere, and then launch the lander module, HAL9001, from this position. This has the added benefit that VoyagerX can circulate the planet many times, gathering large amount of data in the process. We can find, by trial and error, at which density of the atmosphere,  $\rho_{crit}$ , from the planet the satellite manages to stay in a stable orbit for an extended period of time, and then maneuver the satellite as close to that distance as possible. To do this, we will again employ a Hohmann transfer, as explained in section 2.3.2. However, this maneuver is different in two crucial ways: the maneuver must be much more precise, and it must be in the opposite direction (from a low orbit to a high orbit). The fact that we must use a Hohmann transfer the other way around, can be easily adjusted for by reversing the Hohmann transfer needed to leave the elliptical orbit. As shown by [Widnall and Peraire, 2008], this leads to the following boost:

$$\Delta v = -\sqrt{\frac{GM}{r_1}} \left( 1 - \sqrt{\frac{2r_2}{r_1 + r_2}} \right) \quad (61)$$

Where  $r_1$  is the initial distance from the planet, given by equation 27, and  $r_2$  is the distance to the edge of the atmosphere, measured from the center of Hiffre. To get increased precision, we remember that Hohmann assumes perfectly circular orbits. Luckily, we can easily achieve this, by employing equation 28. Thus, by boosting into a circular orbit before employing the Hohmann burn, we should be able to get highly precise results. To see more of Hiffre, we will get into a polar orbit, from the equatorial orbit at which we arrive. In a polar orbit, we will be able to see more of Hiffre, as it spins beneath us. Thus we will need three boosts to get into a closed orbit:

- Change the initial velocity of the satellite so that the satellite has velocity dictated by 28, with  $r$  given by the initial distance to the planet. To see more of the planet, we will let the new velocity be in a polar orbit (along the z-axis).
- Apply the boost given by equation 61, to get into an elliptical orbit, taking VoyagerX closer to Hiffre
- After the transfer time  $T_s$ , dictated by equation 24, boost again, so that the total velocity of the satellite is given by equation 28, with  $r$  now being the outer edge of the atmosphere.

### 2.6.3 Slowing our descent

Once VoyagerX is in an orbit at the outer edge of the atmosphere, we can launch the lander module, HAL9001. However, as shown in C, the surface area of HAL9001 (when the solar panels are not yet unfolded) is only  $0.3 \text{ m}^2$ . This is not sufficient for HAL9001 to be stopped by the atmosphere. Therefore, we equip HAL9001 with two tools: a landing parachute and a landing engine.

#### Calculating the size of the parachute

The engineers of HAL9001 have made the lander module resistant to impacts at a radial velocity of up to 3m/s. Due to potential errors in our atmosphere calculation and our identification of the gases in the atmosphere, we aim for a safety margin of 0.2 m/s, so that we aim for an impact velocity of 2.8 m/s. Thus, we want to calculate the size of the parachute necessary to achieve this impact velocity. We find the terminal speed of the satellite,  $v_t$ , by equating the gravitational force from equation 10 to the drag force (equation 60). This gives (working with magnitudes):

$$G \frac{M_{HAL} M_H}{r^2} = \frac{1}{2} M_{HAL} \rho C_D A v_t^2 \quad (62)$$

Where  $M_{HAL}$  is the mass of the lander module and  $M_H$  is the mass of Hiffre. Solving for  $v_t$ , and realizing that, close to the surface,  $\rho \approx \rho_0$ , gives the terminal speed (close to the surface) as:

$$v_t = \sqrt{\frac{2GM_{HAL}M_H}{\rho_0 C_D A r_p^2}} \quad (63)$$

Where  $r_p$  is the planet radius. Note that this is the terminal speed, whereas the key factor for the landing is the radial velocity, i.e. the radial component of equation 63. However, some simple physical considerations show that this is nonetheless

an excellent approximation to the required velocity. Notice that the drag force always acts opposite to the current velocity. Thus, if there is any initial velocity in the tangential direction, it will quickly die out. This is not the case with the velocity in the radial direction, as the gravitational force provides an acceleration in this direction. As the atmosphere is quite dense near the surface of the planet, all tangential velocity should have vanished by the time HAL9001 hits the surface. Thus, we may use the above equation to determine the surface area of the lander module required to land safely, by solving for  $A$ . This gives:

$$A = \frac{2GM_{HAL}M_H}{\rho_0 r_p^2 v_t^2} \quad (64)$$

### The landing engine

As we are unsure of the composition of our atmosphere at the time of launch, we will also include a landing engine, in case our parachute is not large enough. This will give us a force upwards, slowing down the landing module. If the terminal speed,  $v_t$ , is significantly larger than our attempted 2.8 m/s, we will have to engage the landing engine to slow us down to a safe speed,  $v_{safe}$ . Assume that we have reached terminal velocity when engaging the landing engine. The landing engine will produce an additional force,  $F_L$  on the satellite, pointing away from the planet. This gives a new final velocity, which can be found by equating forces:

$$F_D + F_L = F_G$$

Where  $F_G$  is the force of gravity. Assuming that I am travelling with velocity  $v_t$  before the landing engine engages, we can now use equation 62 to rewrite the gravitational force. This gives (close to the surface of Hiffre):

$$\frac{1}{2}M_{HAL}\rho_0 C_D A v_t^2 = \frac{1}{2}M_{HAL}\rho_0 C_D A v_{safe}^2 + F_L$$

Which gives the required force from the landing engine as:

$$F_L = \frac{1}{2}M_{HAL}\rho_0 C_D A (v_t^2 - v_{safe}^2) \quad (65)$$

#### 2.6.4 Picking out a landing spot and landing on Hiffre

Once we are in a stable, polar, orbit about our satellite, we can take pictures of the planet. This will enable us to pick a suitable landing spot. By waiting for long enough, we will be able to see every part of the planet's surface if we are in a polar orbit. As can be seen in C, the rotational period of Hiffre is fairly high. Therefore, we may approximate Hiffre as stationary during our descent.

Once we have found a suitable spot, we can make a single simulation, choosing an initial velocity of HAL9001 as described in section 2.6.6, to find at what longitude and latitude we impact Hiffre. This can be found by applying the inverse spherical coordinate transformations, described [Weisstein, ]. We must, however, make a slight modification. We measure the angle such that the planet is stationary at the time when we enter in our orbit close to the planet. At this point, the angles are fixed on a coordinate system with the planet at the center. After this time, however, the planet rotates in the counter-clockwise direction, with the rotation axis pointing in through the North pole. Thus,  $\theta$  (the polar angle) stays constant, but  $\phi$  varies with the angular velocity of Hiffre,  $\omega_H$ . This gives the angles as:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \phi &= \arctan\left(\frac{y}{x}\right) - \omega_H t \\ \theta &= \arccos\left(\frac{z}{r}\right) \end{aligned} \tag{66}$$

Where  $x$ ,  $y$  and  $z$  are the coordinates of our landing site, and  $t$  is the time of impact (measured from when we first enter the planetary orbit).

Once we have established an initial impact site in our simulation, we can go back and change the time of launch. As everything is radially symmetric, this will not change anything physical, and therefore the change in angle will be the same as in the first simulation, no matter the start time. Therefore, we can then simply move the initial position, until we land at our designated spot.

### 2.6.5 Communicating with VoyagerX

We can communicate with VoyagerX through simple commands, such as "orient", "boost" or "launch". However, one command requires special attention: the image command. This command, when we are close to our target planet, takes spherical angles as an argument. Note that these angles are relative to the coordinate system where Hiffre is fixed at the origin. Additionally, we want the angles facing "the other direction", as we want to look down on our planet. This can be easily achieved: we use equation 66, but set  $\omega_H$  to zero (as the pictures are not taken relative to a rotation coordinate system). Then we simply input the spherical coordinates of the position that is *on the other side of Hiffre*. If this is done correctly, VoyagerX will look down towards Hiffre at all times, which enables us to take pictures of the planet, extracting much valuable information.

### 2.6.6 Methods for landing on our target planet

For these simulations, we use the leapfrog method which computes the force at equal points of the position and the velocity, as described in appendix A.1.

To get large quantities of data close to the surface, we release our lander, HAL9001, with a velocity that is only slightly less than the velocity of our satellite, VoyagerX. We experiment numerically, until we find a factor,  $v_{\text{corr,factor}}$ , by which we can reduce the velocity of the satellite, so that the lander module completes at least one entire revolution around Hiffre. We then choose HAL9001's velocity to be given by  $v_{\text{HAL}} = v_{\text{corr,factor}} v_{\text{sat}}$ . Note that the lander module will be in an elliptic orbit around Hiffre, as the satellite is initially in a circular orbit, and we release HAL9001 with a velocity that is slightly less. We open the parachute once HAL9001 has circulated at least once around our planet, and at a low height,  $h_{\text{parachute}}$ , to ensure that the lander goes straight down.

Due to time constraints, we were not able to run sufficient simulations to land at any spot. Therefore, we ran a simulation where we landed at a spot that was acceptable (in particular: which was on the day side of the planet, and not in a lake), and decided to land on this spot. We recorded both the impact site and impact time in our simulation, and the actual impact site and impact time.

### 3 Results

In this section, we present all our results, with a brief explanation. We postpone an extended discussion and evaluation of these results until section 4.

#### 3.1 Results from our engine simulation

A snapshot from our 3D simulation of the particle, as discussed in section 2.1.5, is shown below.

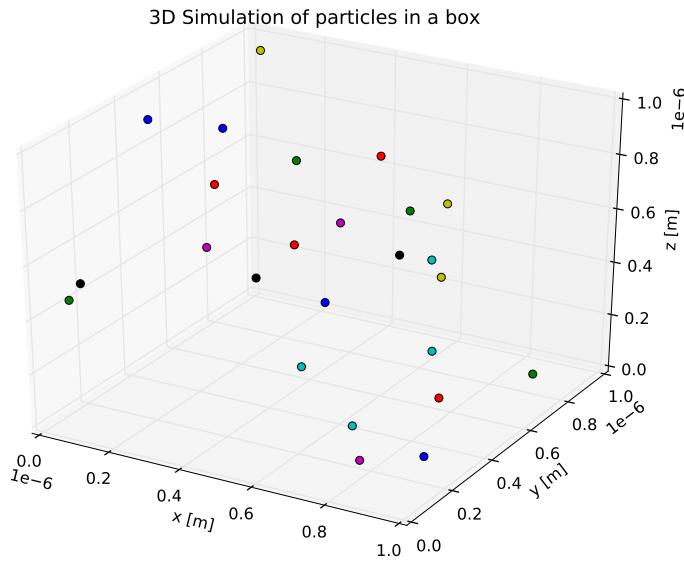


Figure 4: Snapshot from 3D simulation of 25 ideal gas particles confined in a cube with length  $10^{-6}$ m. Total simulation time was  $10^{-9}$  s. Note that the size of the gas particles has been grossly exaggerated and the number of particles drastically reduced, for a better visual representation.

A summary of our engine parameters is shown in table 1 below. We have also included the mass estimates. These were computed using the  $\Delta v$  derived in section 3.3.

Table 1: Parameters of our engine, from simulation.

Force from a single fuel cell, $F$ [N]	$1.72 \times 10^{-9}$
Number of particles escaping per second from a single cell, $n$ [ $s^{-1}$ ]	$6.35 \times 10^{13}$
Number of fuel cells, $N$	$8.94 \times 10^{12}$
Mass per second from one fuel cell, $\Delta m$ [ $kg\ s^{-1}$ ]	$2.11 \times 10^{-13}$
Analytic mass of fuel needed, $M_{fuel}$ [kg]	22755
Numeric mass of fuel needed, $M_{fuel,Numeric}$ [kg]	22755
Mass remaining at the end of the boost $M_r$ [kg]	1100

The results from the consistency checks discussed in section 2.1.5, are shown in table 2 below:

Table 2: Results from the consistency checks of our engine.

Percentage error in initial kinetic energy, $\epsilon_{E_k}$ [%]	0.00581
Percentage error in average speed, $\epsilon_v$ [%]	0.119
Percentage error in average pressure, $\epsilon_P$ [%]	0.132
Percentage difference in analytic and numeric mass estimates, $\epsilon_m$ [%]	$7.75 \times 10^{-5}$

Finally, we show the time development of the mass remaining and the velocity achieved by the satellite:

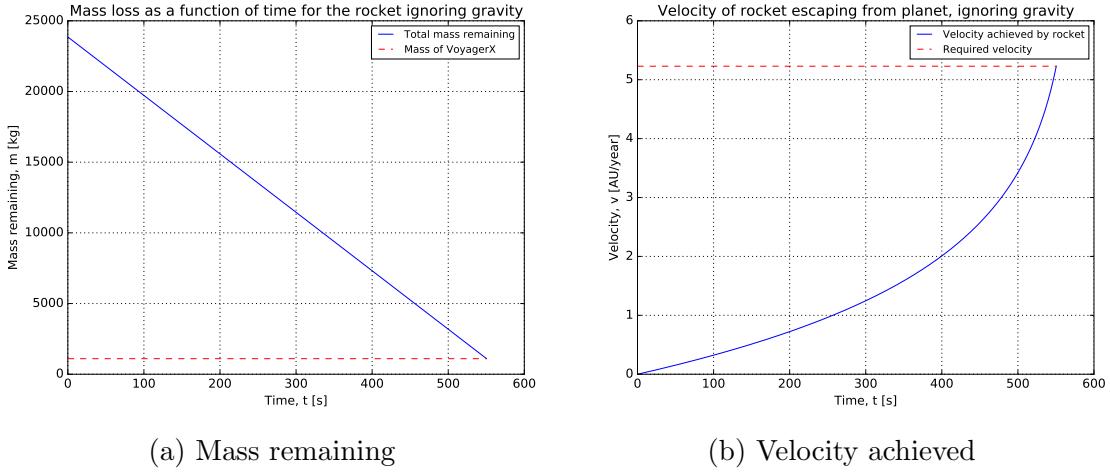


Figure 5: Plot of the numeric solution of equation 5, including both the velocity attained by our satellite, VoyagerX, and the mass used. The dotted lines are respectively the mass of the satellite itself,  $M_{sat}$  and the required velocity change,  $\Delta v$ . These simulations were run with an initial mass of 22755 kg and a required  $\Delta v$  of 5.23 AU/year, using a step size of  $10^{-3}$  s.

### 3.2 Result from our simulation of the solar system

A plot of the orbit of the planets is shown in the figure below:

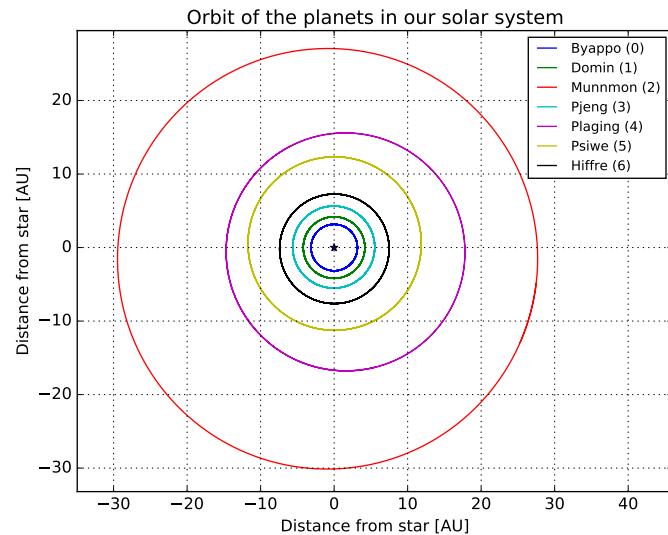
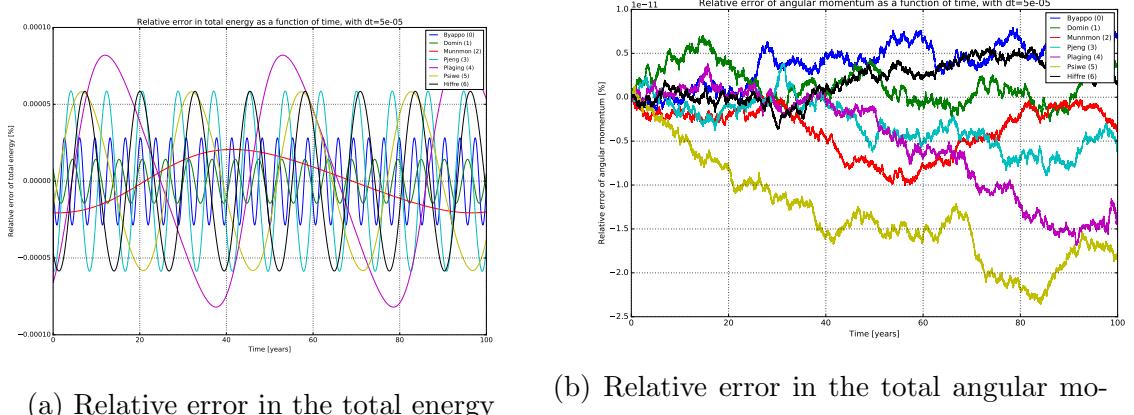


Figure 6: The orbits of all planets in our solar system, with our star indicated in the middle. This simulation was run for 100 years, with 20000 steps per year.

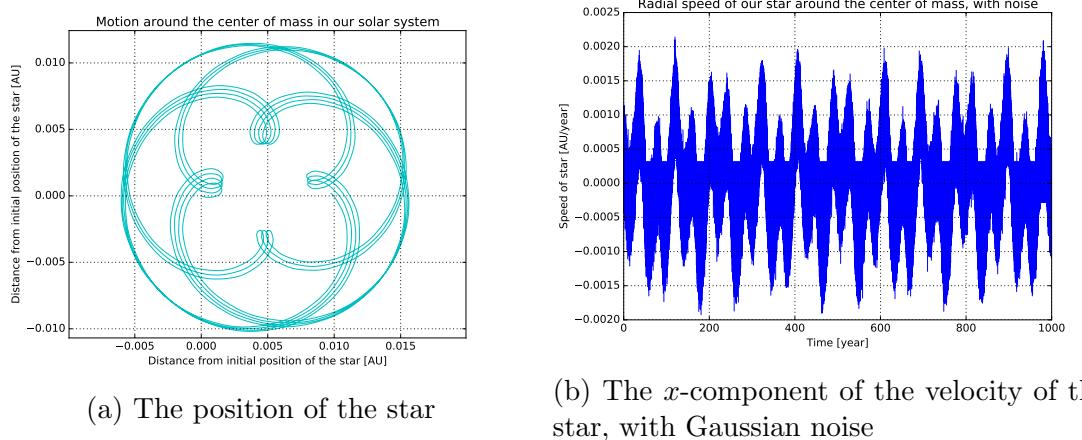
We also plot the relative error of the total energy and the angular momentum as a function of time in the plot below: Finally, we present the radial velocity curve and the light curve of our star, used to investigate whether or not extraterrestrials could observe our solar system.



(a) Relative error in the total energy

(b) Relative error in the total angular momentum

Figure 7: The relative error of total energy and angular momentum as a function of time, expressed in percentages. In this simulation, the number of grid points  $N = 20000$  per year. We simulate 100 years.



(a) The position of the star

(b) The  $x$ -component of the velocity of the star, with Gaussian noise

Figure 8: The motion of our star, being affected by the three most massive planets in our solar system. The star starts at the origin, with velocity such that the center of mass stays stationary. We also plot the  $x$ -component of the velocity, with Gaussian noise. We simulate for 1000 years, with 20000 steps per year.

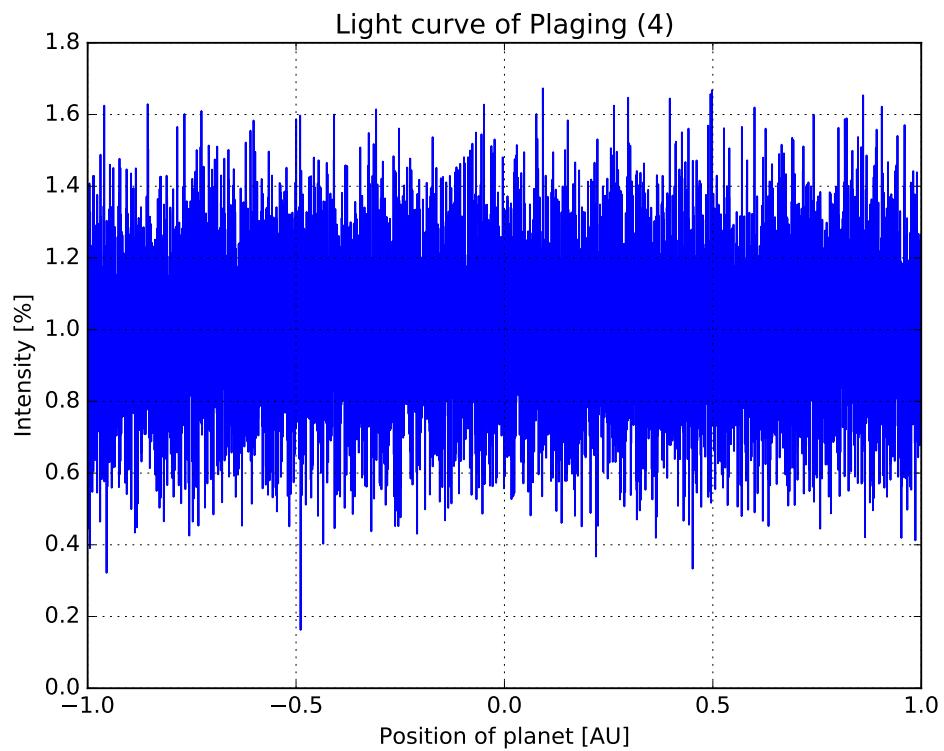


Figure 9: The light curve of our star, in a simplified model described in section 2.2.5, with Gaussian noise added. This corresponds to the most massive planet (Plaging) eclipsing our star.

### 3.3 Results from our orbital calculations

The path of our satellite, VoyagerX, from shown in the figure 10 below:

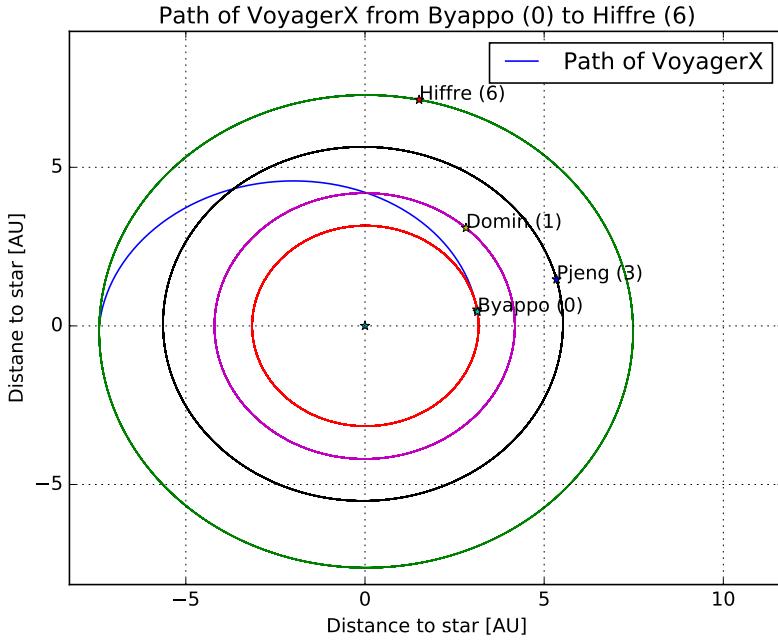
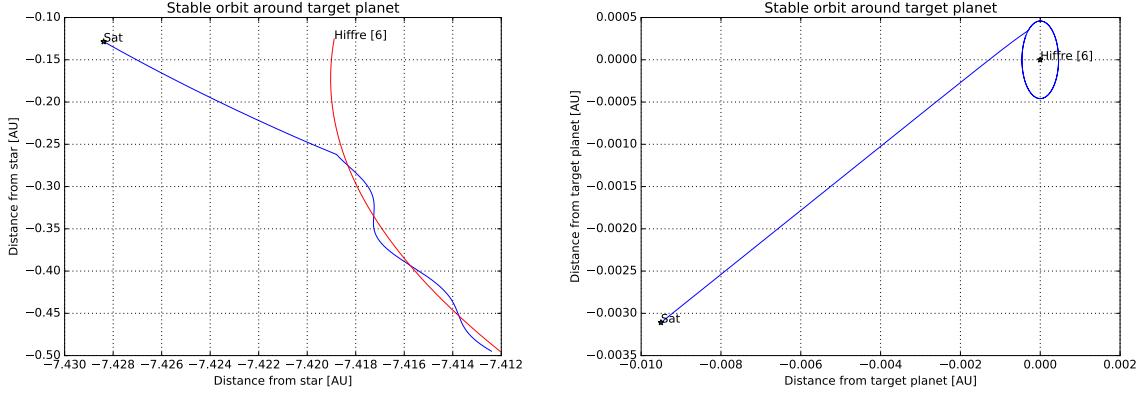


Figure 10: Path of VoyagerX through interplanetary space, including only the launch and the first Hohmann boost, detailed in equation 23. We simulate from  $t_0 \approx 3.57$  years to  $t \approx 7.10$  years. with  $N_1 = 20000$  timesteps per year whilst the satellite is more than 0.1 AU away from all planets, and with  $N_2 = 2 \times 10^6$  timesteps per year whilst close to a planet.

We ran this simulation with the number of timesteps, both close to the planet and in interplaneatry space, increased by a factor 10 for the final simulations, used to determine what commands to give the satellite. However, these timesteps were inconvenient for plotting. There was also no visible difference between these plots. This boost brought us within 0.1 AU of Hiffre. Whilst we could finetune our parameters further, to get closer, we decided that - due to uncertainties in the actual launch, it would be better to implement an additional boost towards the satellite at a distance of 0.1 AU, to get us within the distance specified by 27. We boost in the direction that the satellite will have at a time  $t + \Delta t_b$ . This boost, together with the circulation boost, is shown in figure 11.



(a) Path of VoyagerX as seen from the star    (b) Path of VoyagerX as seen from Hiffre

Figure 11: The correction and injection boost of our satellite, VoyagerX, as seen from both the star and the target planet. This simulation was run with  $2 \times 10^6$  timesteps per year, starting from the final point and time shown in figure 10

Some crucial output from our simulation is summarized in the table below:

Table 3: Some important quantities from our orbital calculations

Computed launch time of VoyagerX, $t_{\text{launch,computed}}$ [Year]	3.59
Computed velocity change for boosting out of SOI, $\Delta v_{\text{SOI}}$ [AU/year]	3.53
Computed velocity change for Hohmann transfer boost, $\Delta v_{\text{Hohmann}}$ [AU/year]	1.01
Computed velocity change for the correction boost at 0.1 AU, $\Delta v_{\text{corr}}$ [AU/year]	0.854
Computed velocity change for injecting into stable orbit, $\Delta v_{\text{SO}}$ [AU/year]	0.212
Distance to Hiffre at which stable orbit was achieved, $r_{\text{stable}}$ [AU]	$4.59 \times 10^{-4}$
Correction boost wait time, $\Delta t_b$ [Year]	0.00544
Correction factor for the start time, $\Delta t_{\text{launch}}$ [Year]	-0.0280
Correction factor for the velocity change, $\Delta v_{\text{err}}$ [AU/year]	-0.850
Safety margin for the velocity change, $\epsilon_{\text{Vel}}$ [%]	10

So that the launch time is given by:

$$t_{\text{launch}} = t_{\text{launch,computed}} + \Delta t_{\text{launch}} = 3.562 \text{ year}$$

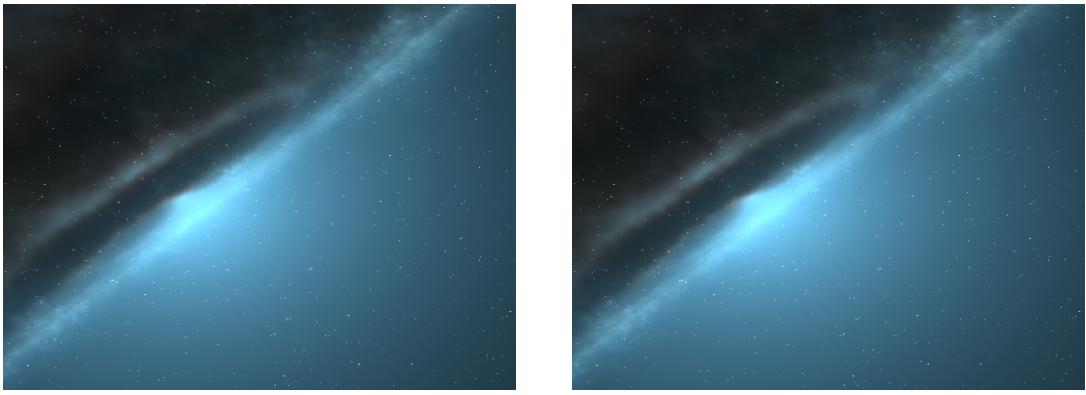
And the final total velocity change required is:

$$\Delta v_{\text{tot}} = (1 + \epsilon_{V_{el}}) (\Delta v_{\text{SOI}} + \Delta v_{\text{Hohmann}} + \Delta v_{\text{corr}} + \Delta v_{\text{SO}} + \Delta v_{\text{err}}) = 5.23 \text{ AU/year}$$

## 3.4 Results from our orientation software

### 3.4.1 Software to determine orientation

The test image was taken at a time  $t_0 = 3.601$  years. The test image is shown next to the image of the celestial sphere (found from our least-square algorithm) as seen from our home planet, Byappo, in figure 12. The software found that this image was taken at an angle  $\phi_0 = 11^\circ$ , with a least-square error of 0.



(a) Reference image taken by our satellite. (b) Image taken from Byappo.

Figure 12: The reference image taken by VoyagerX, of interplanetary space, right after launch, compared with the image of the celestial sphere taken from Byappo.

### 3.4.2 Software to determine position

We tested our software to determine the satellite position by randomly picking 1000 positions, in the range  $[-20\text{AU}, 20\text{AU}]$  from our star. The software managed to pinpoint the position exactly to within  $10^{-5} \text{ AU}$  for each of the 1000 points.

### 3.4.3 Software to determine the velocity of the satellite

We tested our software for determining the velocity by checking the velocity if the doppler shift observed by VoyagerX is zero (giving it the same velocity as the stars), and if the Doppler shift is the same as the one observed from Byappo

(giving it zero velocity relative to Byappo). In both of these cases, the program reproduced the expected results. We then checked the actual wavelength shifts at time  $t_0 = 3.601$  years. A list of important parameters for this simulation is shown in table 4 below.

Table 4: Parameters from our velocity finding software

Angle of reference star 1 relative to home star, $\phi_1$ [°]	171
Angle of reference star 2 relative to home star, $\phi_2$ [°]	87.8
Wavelength shift of reference star 1 relative to home star, $\Delta\lambda_{\text{star},1}$ [m]	$-0.0201 \times 10^{-9}$
Wavelength shift of reference star 2 relative to home star, $\Delta\lambda_{\text{star},2}$ [m]	$-0.0141 \times 10^{-9}$
Wavelength shift of reference star 1 relative to VoyagerX, $\Delta\lambda_{\text{sat},1}$ [m]	$-0.0417 \times 10^{-9}$
Wavelength shift of reference star 2 relative to VoyagerX, $\Delta\lambda_{\text{sat},2}$ [m]	$-0.0825 \times 10^{-9}$

### 3.5 Results from the actual launch to get into a stable orbit

After the initial boost (consisting of  $\Delta v_{\text{SOI}}$  and  $\Delta v_{\text{Hohmann}}$ ) we consistently checked the satellite's distance to Hiffre. VoyagerX was able to approach closer than indicated by our simulations. In fact, after a time  $t_0 = 7.12$  years, VoyagerX was able to come close enough to boost directly into a circular orbit, without performing a correction boost. However, the magnitude of this boost was slightly larger, namely  $\Delta v_{\text{SO,actual}} = 0.884$  AU/year. Thus, the actual boost required by our satellite,  $\Delta v_{\text{tot,actual}}$  was:

$$\Delta v_{\text{tot,actual}} = \Delta v_{\text{SOI}} + \Delta v_{\text{Hohmann}} + \Delta v_{\text{err}} + \Delta v_{\text{SO,actual}} = 4.57 \text{AU/year}$$

Positions measurements after this boost showed that VoyagerX was in an elliptic orbit, as the distance to Hiffre varied between  $4 \times 10^{-4}$  AU and  $3 \times 10^{-4}$  AU over the three years that we orbited Hiffre. Our satellite returned the following image of it orbiting Hiffre:

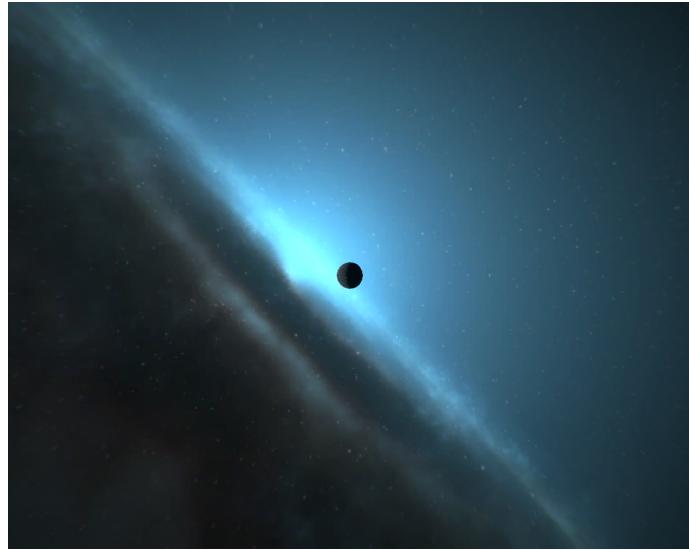


Figure 13: Snapshot of VoyagerX orbiting Hiffre

### 3.6 Results from the spectroscopic investigation

A complete list of the parameters associated with the gases we checked for in Hiffre's atmosphere is shown below. Here  $v_{\text{sat}}$  is the velocity of the satellite relative to the target planet,  $\lambda_0$  is the theoretical location of the absorption line, whereas  $\lambda_{0,\text{obs}}$  is the observed location of the absorption line.

Table 5: Parameters from the spectroscopic analysis of Hiffre's atmosphere

Species	$\lambda_0$	$\chi^2$	$F_{\min}$ [W m $^{-2}$ ]	$\sigma$ [nm]	$\lambda_{0,\text{obs}}$ [nm]	Mass [g/mol]	$v_{\text{sat}}$ [m/s]	Temperature [K]
O <sub>2</sub>	630	916	0.77241	0.00085	630.02	31.9988	9598	629.82
O <sub>2</sub>	690	1015	0.76207	0.00018	689.99	31.9988	-4783	23.55
O <sub>2</sub>	760	1009	0.79310	0.00048	759.98	31.9988	-7056	138.012
H <sub>2</sub> O	720	927	0.92759	0.00042	719.99	18.0153	-5854	66.28
H <sub>2</sub> O	820	1185	0.76207	0.00128	820.03	18.0153	9732	474.64
H <sub>2</sub> O	940	1238	0.70000	0.00147	940.03	18.0153	9733	476.37
CO <sub>2</sub>	1400	1950	0.70000	0.00162	1400.05	44.0095	9865	637.16
CO <sub>2</sub>	1600	2198	0.70000	0.00035	1600.01	44.0095	2508	22.77
CH <sub>4</sub>	1660	2197	0.92759	0.00403	1660.05	16.0425	9866	1022.34
CH <sub>4</sub>	2200	2918	0.86552	0.00251	2200.07	16.0425	9865	225.79
CO	2340	3177	0.81379	0.00293	2340.08	28.0101	9799	474.83
N <sub>2</sub> O	2870	3878	0.85517	0.00153	2870.09	44.0128	9598	135.25

Two samples plots of the fits are shown in figure 14 below.

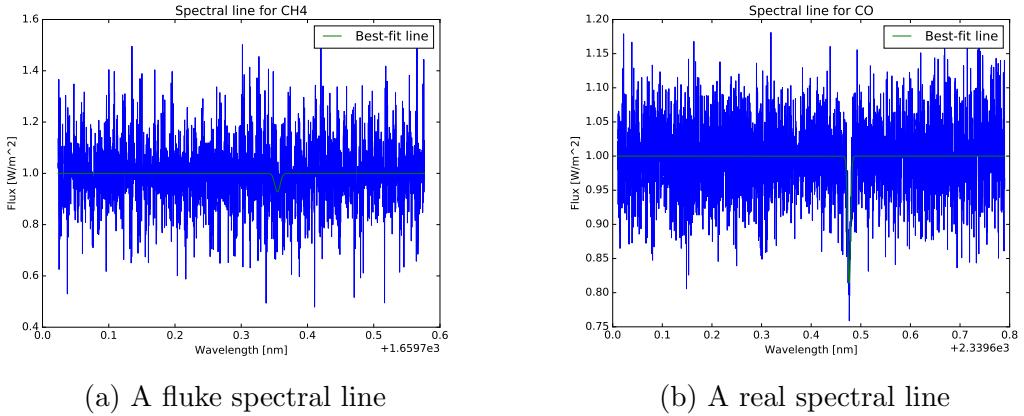


Figure 14: Two sample absorption lines, one being identified as a fluke and the other one being identified as a real absorption line, using a combination of visual clues and the parameters in table 5.

### 3.7 Results from our modelling of the atmosphere

A plot comparing the analytic and numeric solution of the atmosphere is shown below. The largest relative error between the analytic and numeric solution was 0.02% for this run, whereas the mean relative error was 0.0099 %. This error decrease with decreasing step size.

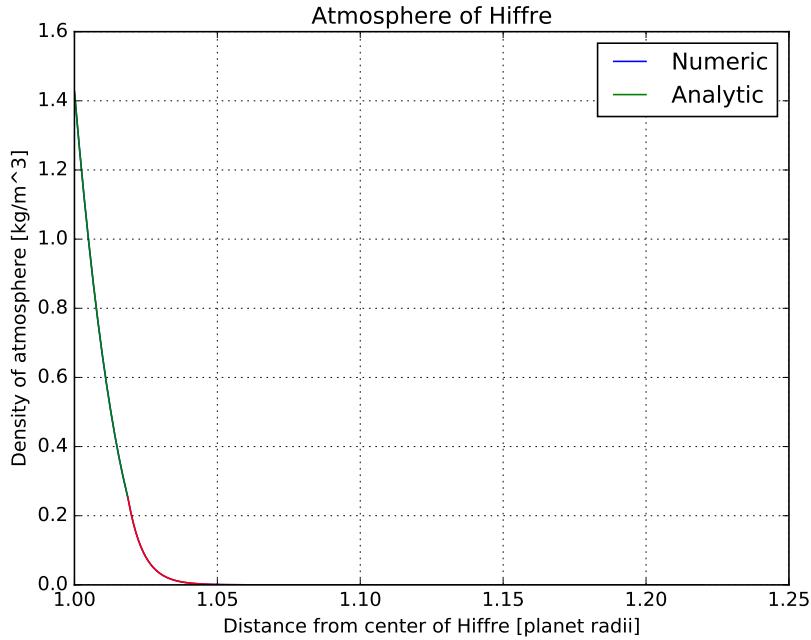


Figure 15: Plot of our model of the atmosphere of Hiffre, showing both the analytic and numeric solution. This was produced with steps a step size of 0.1 meters. Note that we have plotted the density as a function of planet radii, for improved readability.

### 3.8 Results from our landing simulation

The simulated trajectory of both VoyagerX and HAL9001 are shown in figures below: We also include two image shot by VoyagerX whilst close to the planet, to demonstrate how one may pick a landing site:

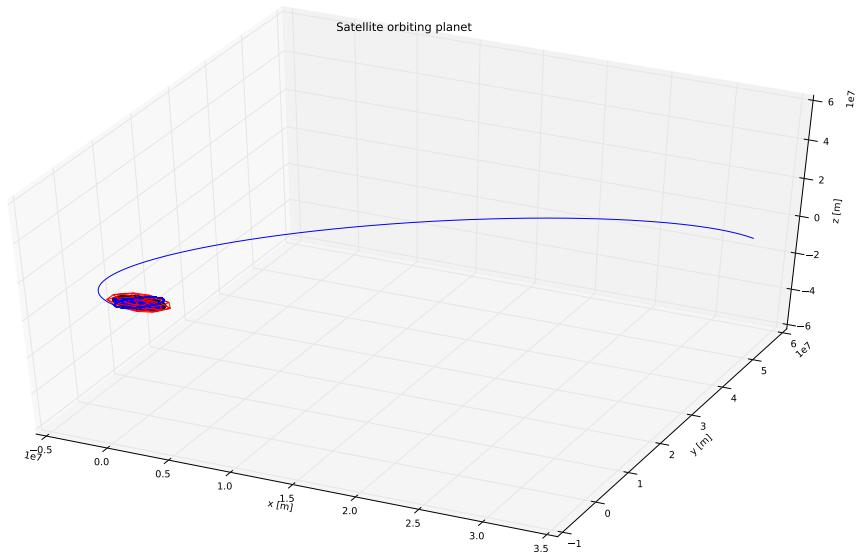


Figure 16: Simulation of VoyagerX's descend to Hiffre. We simulate with 10 steps per second, for approximately 4 days. The red grid is the upper boundary of our atmosphere.

Simulation of HAL9001 landing on Hiffre

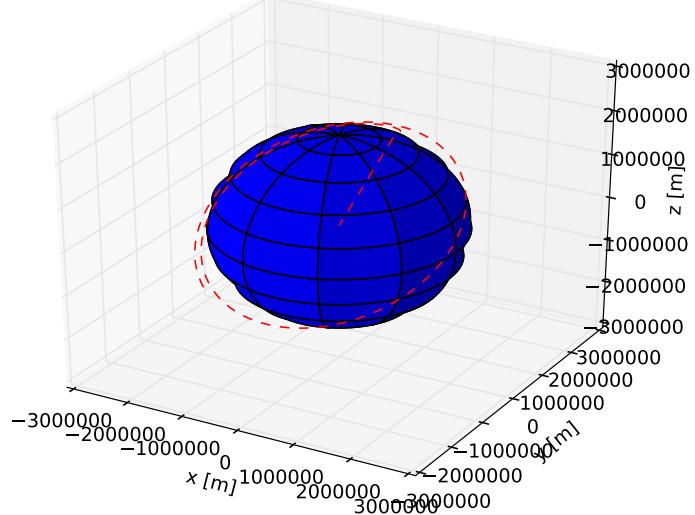
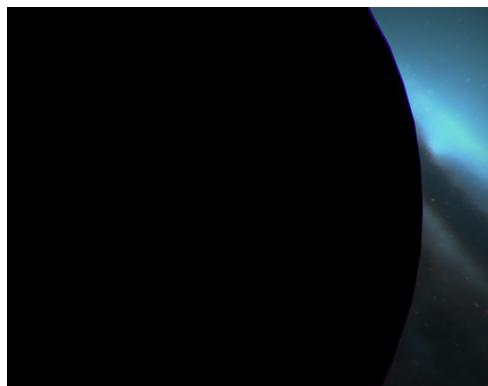
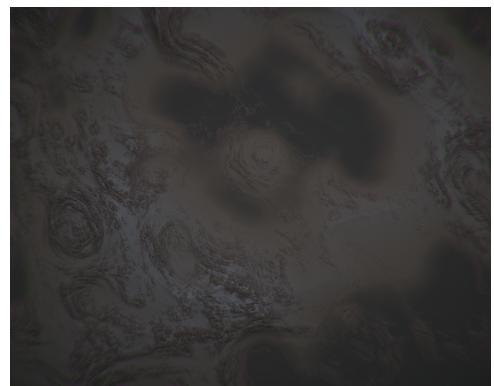


Figure 17: Simulation of HAL9001's descend to Hiffre. We simulate with 200 timesteps per second, for a total of 10 hours (though the satellite hits the planet after approximately 6.38 hours)



(a) An image of Hiffre's night side



(b) A close-up of Hiffre's surface

Figure 18: Two images showing VoyagerX's close-up of Hiffre.

A table summarizing the parameters from this part is shown below:

Table 6: Some of the important parameters from our landing on Hiffre.

Density at the outer edge of atmosphere, $\rho_{crit}$ [kg m <sup>-3</sup> ]	$10^{-13}$
Area of parachute required, $A_{parachute}$ [m <sup>2</sup> ]	47.7
Factor by which we reduce the satellite velocity to launch HAL9001, $v_{corr,factor}$	0.026
Height above surface at which parachute is deployed, $h_{parachute}$ [km]	230
Simulated azimuthal impact angle, $\phi_{simulated}$ [°]	353.2
Actual azimuthal impact angle, $\phi_{actual}$ [°]	353.2
Simulated polar impact angle, $\theta_{simulated}$ [°]	29.1
Actual azimuthal impact angle, $\phi_{actual}$ [°]	38.3
Simulated impact time in seconds after achieving stable orbit, $t_{impact,simulated}$ [s]	212970
Actual impact time in seconds after achieving stable orbit $t_{impact,actual}$ [s]	212992
Simulated radial impact velocity $r_{impact,simulated}$ [m/s]	2.8
Actual radial impact velocity, $r_{impact,actual}$ [m/s]	2.8

## 4 Discussion

In this section we provide an extensive discussion of the results from the previous section.

### 4.1 The engine simulation

As can be seen from table 1, a single fuel cell has very little effect, and we need a large number of fuel cells to achieve a viable engine. This shows that we indeed started with a microscopic cell. As such, there may be a potential problem of scale that we have not considered. A large engine, with a single combustion chamber, may well behave different than the cells simulated here, as a qualitatively different behavior is frequently observed for large thermodynamic systems, than for small systems.

An additional unrealistic approximation made by us is that the exhaust gas consists purely of hydrogen, and that it is an ideal gas. However, our boost calculations were sufficient and VoyagerX was able to arrive at our home planet. Thus, these approximation do not seem to be a large problem.

It is reassuring to see the small errors in table 2. These confirm the correctness of our numeric implementation of the ideal gas. Note furthermore the exactness of the results displayed in figure 5. VoyagerX manages to reach the required velocity exactly at the same time as all fuel is used. This confirms the correctness of equation 5. In fact, numerical errors sometimes made the remaining mass dip just below the mass of VoyagerX. This is not a problem however, as our fuel calculations assumed that we wanted achieve a boost of  $\Delta v_{tot} = 5.23$  AU/year, whereas VoyagerX in reality only required a boost of  $\Delta v_{tot,actual}=4.57$  AU/year. This demonstrates that our fuel calculations were accurate.

One of the trickier thing was deciding how to deal with the gas particles that escaped our simulation cell through the nozzle. There were three immediately viable options:

1. Replace the particles with new particles, with velocities following the same Gaussian distribution as earlier, and position following the same normal distribution
2. Fill in particles from the top, that is to say, let the new particles have velocities dictated by a Gaussian distribution, but let their position be at the top of the box

- Simply reflect the particles back, i.e. reverse their velocity and place them "on" the hole.

**Option 1** The first option seems viable at first, but it introduces a statistical bias. This is because the particles with a large (negative)  $v_z$  component will be siphoned off gradually. This affects the velocity distribution, reducing the effective velocity. This results in a slight pressure drop over time, which we did observe when we programmed this approach. Additionally, there will be more particles at the top of the box than at the bottom, as the z-component of the new particles will be higher than the one of the older particles no matter where the new position is drawn (because we take away the particles with the lowest z-component). This implies that the positions will no longer be normally distributed, which creates the problems discussed in the paragraph on option 2.

**Option 2** This option seems physically most realistic, but again disrupts the numerical stability by introducing a statistical bias. In fact this bias is even worse than option 2, as the position distribution is even more skewed (with a much larger probability that particles will be at the top of the box). This breaks the symmetry of the pressure, as there will now be a higher probability of a particle hitting the top of the box than a particle hitting the bottom of the box. This is therefore also not a viable option, if one wants to uphold the assumption of constant pressure.

**Option 3** This option disrupts the system the least. Essentially, it only counts how many particles hit a specific part of the box, without interfering with the motion of the particles in the box. To fulfill the assumption of constant pressure, this therefore seems to be the best approach out of the three we considered, and was therefore the one we adapted.

## 4.2 The simulation of our solar system

### 4.2.1 The simulation of the planet orbits

Looking at figure 6, it seems clear that we have chosen a sufficiently small step size (which was found by trial and error). The orbit are stable for 100 years. This is to some degree confirmed by the errors shown in figure 7. Note first the scale on the axis of the relative error in angular momentum. It seems clear that this error is almost exclusively caused by numerical error, as the relative error marginal. This is reassuring because, as explained by [Young, 2014], leapfrog integration should exactly conserve angular momentum. The relative error in the energy, on the other hand, has significantly larger variations. Notice how these are almost perfectly periodic, with a mean of zero. As also explained by [Young, 2014], leapfrog integration conserves energy over one period which is the behavior we observe. There is, however, one interesting point. Notice how the relative error in the total energy of Munnmon (the planet that is furthest out) does not start at zero. This is surprising, as we calculate the relative error relative to the initial energy (at time 0). Thus it seems that the total energy of Munnmon (and some other planets) *jumps* initially, to later settle into the observed periodic pattern. This is an interesting observation, which may warrant further investigation in the future.

### 4.2.2 The investigations into the appearance of our solar system from afar

As can be seen in figure 8, our star wobbles about the center of mass (which is in the middle of the figure). The motion is clearly made up of multiple periodic oscillations, superimposed on top of each others - namely the orbit of the planets. One interesting feature is that, whilst there is a clear periodic structure to the motion, the curves are not perfectly on top of each other. This may have one of two possible reasons:

- Numerical errors in the experiment, i.e. our timestep being too low
- The alignment of the three most massive planet not being exactly equal at each oscillation

We tested the first of these points by increasing the time step, however, the same pattern was observed. This indicates that these variations are indicative of the complex nature of these many-body problems. The motion of the star is dominated by the motion of the three planets, however, the relative position of the planets are not the same at each revolution. Therefore, the pattern changes slightly over time.

When looking at the radial velocity curve of the star, there is a clear periodic pattern. Even through the noise, there are clearly periodic oscillations present in the data. Thus, extraterrestrials would be able to see our solar system by measuring the Doppler shift (and thereby the velocity) of our star - at least if they have the optimal angle of incidence, and are not too far away.

The opposite is the case with the light curve of the star. Here, even the eclipse of the most massive planet is not visible through the noise. This is not particularly surprising, seeing as the light curve without noise, for the heaviest planet, is shown in figure 19. This shows that the effect is very small. This may be due to our sim-

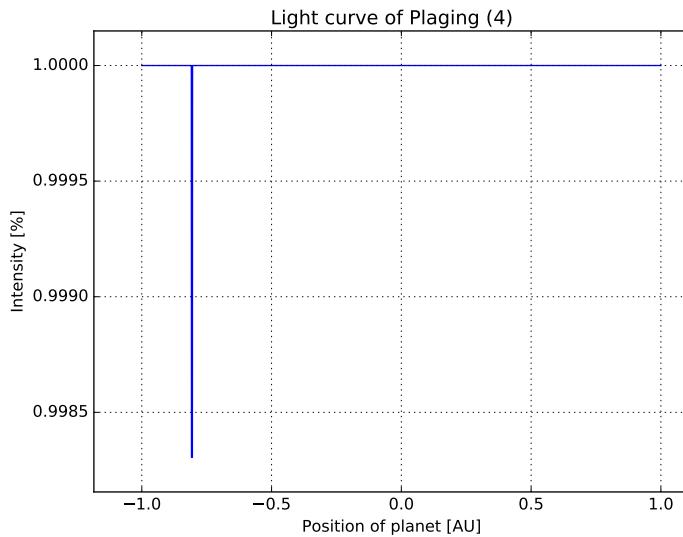


Figure 19: Light curve of the most massive planet, using the simplified model from section 2.2.5, without noise.

plified model however, seeing as the velocities, and therefore the eclipse time, may be slightly different with a different line of sight. Furthermore, some planets may eclipse the star at the same time, combining into a larger effect - though this is very unlikely. Despite all this however - the effect is so small, compared to the noise, that it seems very unlikely that it would be possible to observe our planets by observing the light curve of our star.

### 4.3 The orbital simulation

Calculation show that distance to Hiffre given by the stable orbit requirement in equation 27, is given by:

$$r = 0.0004589 \text{AU}$$

If we set  $k = 10$ , and approximate the satellite-star distance by the satellite-star distance at a distance of 0.01 AU<sup>19</sup>. The distance dictated by equation 30, requiring that the planet be larger than one pixel, is given by:

$$r = 0.0225 \text{AU}$$

This justifies our choice of  $r = 0.0004589$  as the distance at which we wish to achieve a stable orbit.

Figure 10 shows that we were almost able to reproduce the perfect Hohmann burn, traversing half an ellipse. This justifies our patched conic approximation, as it indeed seems that the satellite was barely influenced by anything other than the star in interplanetary space. Note that this was somewhat of a lucky coincidence. As can be seen in the figure, VoyagerX does cross two other planetary orbits before it hits Hiffre. The stars in the figure indicate the initial position of the planets, and all planets rotate anti-clockwise. Thus, VoyagerX may well have been close to being influenced by Domin and Pjeng. This may give a partial explanation of the correction factors that we had to employ.

It is nonetheless somewhat surprising that the correction factor for the velocity is as big as it is. Note that we initially simulated with a much smaller correction factor, and also managed to arrive at Hiffre (by tweaking the starting time slightly). However, we then arrived with a much larger velocity, so that the correction and injection burns were significantly larger, resulting in a much larger fuel usage - in some cases as high as  $5 \times 10^5$  kg! This seems to indicate that a "pure" boost out of the sphere of influence, followed by a Hohmann transfer, may not have been ideal in this case - possibly due to the interference from other planets. Note also, that boost out of the sphere of influence may have been slightly too big, as the gravitational well of the other planets may have pulled VoyagerX out of the gravitational well of Byappo.

As figure 11 shows, we were able to get into orbit around Hiffre, in a clockwise orbit. Notice that we initially overshot the position of Hiffre slightly, before our

---

<sup>19</sup>This is where we perform the correction boost. Note that this is not a very crude approximation, as the actual distance to the star at this point is more than 6 AU, so the error we make in the distance is less than 1/6 of 1 %.

correction boost. This is probably not ideal. Luckily, this was not necessary in the actual launch of VoyagerX, as we did not need the correction, as discussed in section 4.5.

Finally, we calculate the surface area needed for HAL9001 to maintain its power at the surface of Hiffre. This can be computed from equation 31. We approximate the distance to Hiffre by Hiffre's semi-major axes, as this will be the point where HAL9001 receives the least power. Inserting values gives:

$$A = 0.675\text{m}^2$$

This is therefore the area of solar panels which we must give our lander. Note that the initial surface area of the lander is only  $0.3 \text{ m}^2$ . This is because we "tuck away" the solar panels, unfolding them at the surface.

#### 4.4 The orientation of our satellite

As figure 12 shows, the orientation software worked excellently, as the images correspond well each other. In fact, the error made in the least square approximation was exactly 0. As the Satellite moved further from our home planet Byappo, this error presumably changed, as the distant objects shifted slightly. Of course, this may still be an effect that is too negligible to observe.

Whilst our orientation software worked consistently to the required precision, it did sometimes have errors as large as  $10^{-6}$  AU, which still is a fairly significant distance. It is not entirely obvious why this would be the case, as our solution should be exact. It is important to note, however, that a fairly large number of rather intricate calculations are necessary, and thus the numerical error may accumulate. This is a potential explanation, though it may be worth investigating this further.

#### 4.5 The actual launch from Byappo

As figure 13 shows, VoyagerX was able to orbit Hiffre, albeit not quite as in our simulations. The satellite did not require an additional correction boost. It also ended in an elliptic orbit, rather than a circular orbit.

The fact that VoyagerX was able to get there without a correction boost is pleasant but surprising. In our simulation, our satellite was almost able to arrive at

the planet without the correction boost, however, we implemented the correction boost for safety, in case some of our assumptions were too crude, or the timestep too rough. Note that this was a rather unstable problem, as slight changes in our initial conditions made us miss Hiffre. This justifies our use of the correction boost.

The explanation of the elliptical orbit is most likely a result of a simple instructional error. We misjudged the time delay necessarily between each command to our satellite, and therefore launched the satellite approximately 0.03 years after orienting the satellite and finding the velocity necessary to achieve a circular orbit. The result was presumably that our orbital calculations were slightly off, which resulted in us giving slightly wrong commands to VoyagerX. Therefore, it moved into an elliptical, rather than a circular orbit. Luckily, this was not a big problem, as we easily rectified it when simulating the closer orbit, as explained in section 2.6.

## 4.6 The exploration of Byappo's atmosphere

After careful consideration of the parameters in table 5, and after consulting the visual representation of the gases, we conclude that there appears to only be a single gas in the atmosphere, namely CO. This was established due to the following factors:

- None of the elements with multiple spectral lines have consistent velocity or temperature estimates. Furthermore; in none of them, was the spectral line clearly visible in all plots. CH<sub>4</sub> has a consistent velocity estimate, but the lines were no convincing, as shown in figure 14.
- Whilst N<sub>2</sub>O had an acceptable velocity estimate, its temperature was rather low. Additionally, graphically, the line was clearly not present.
- The line for CO is clearly visible, as can be seen in figure 14, and the other parameters are also reasonable.

Thus, the mean molecular mass was easy to estimate, giving:

$$\mu = 28.0101 \text{ g/mol}$$

Inserting into equation 15, gives that the surface temperature of Hiffre is approximately:

$$T_0 = 216\text{K}$$

Which, as earlier noted, is not within the habitable zone.

As can be seen in figure 15, the numeric and analytic expressions for the atmosphere are in agreement. This indicates that we have calculated correctly. Furthermore, note how quickly the atmosphere rises from being close to zero. This justifies our choice to stay where the density of the atmosphere is less than  $10^{-13} \text{ kg m}^{-3}$ .

## 4.7 The landing

We see in figure 16 that we manage to get close to our planet and achieve a circular orbit with the Hohmann burn and the circular burn, as planned. We check this by investigating how stable our orbit around Hiffre is, once we are close, and find that it changes with less than 0.5 kilometers as we orbit. Thus, we are able to achieve a stable orbit.

As can be seen in table 6, we achieve precisely the impact velocity that we wanted. HAL9001 managed to land safely on Hiffre, without either burning or crashing into the planet. However, we do not quite manage to hit the same spot in our simulation as we hit with the actual satellite.

Notice how we manage to hit the correct azimuthal angle, but not the correct polar angle. The most likely explanation for this is the large Coriolis force visible in figure 17. There, it is clear that the landing module is taken by the atmosphere, and blown slightly off the trajectory. This indicates that we should perhaps have used a lower time step in the simulation, once we were close enough to the planet. Notice further that the impact time also differs slightly, which also indicates a discrepancy between our computed results and the simulation.

One alternative explanation to this, may be a faulty estimate of the density of our atmosphere,  $\mu$ . Notice that the desired impact velocity, given by equation 64, is independent of the density of the atmosphere. It is therefore not surprising that we managed to achieve our desired impact velocity. However, if this density was underestimated, we may get a larger force from the atmosphere, leading to a larger Coriolis force, thereby dragging the satellite along, increasing the landing time. We could easily have estimated  $\mu$  wrongly, as it was difficult to distinguish a fluke from a real line, and because we had to restrict ourselves to such a narrow range of the data and parameters.

It is not at all obvious which of these sources of error was the main reason for the discrepancies. Either way, it was luckily not disastrous; as the following image from the surface of Hiffre shows, HAL9001 was able to land safely.

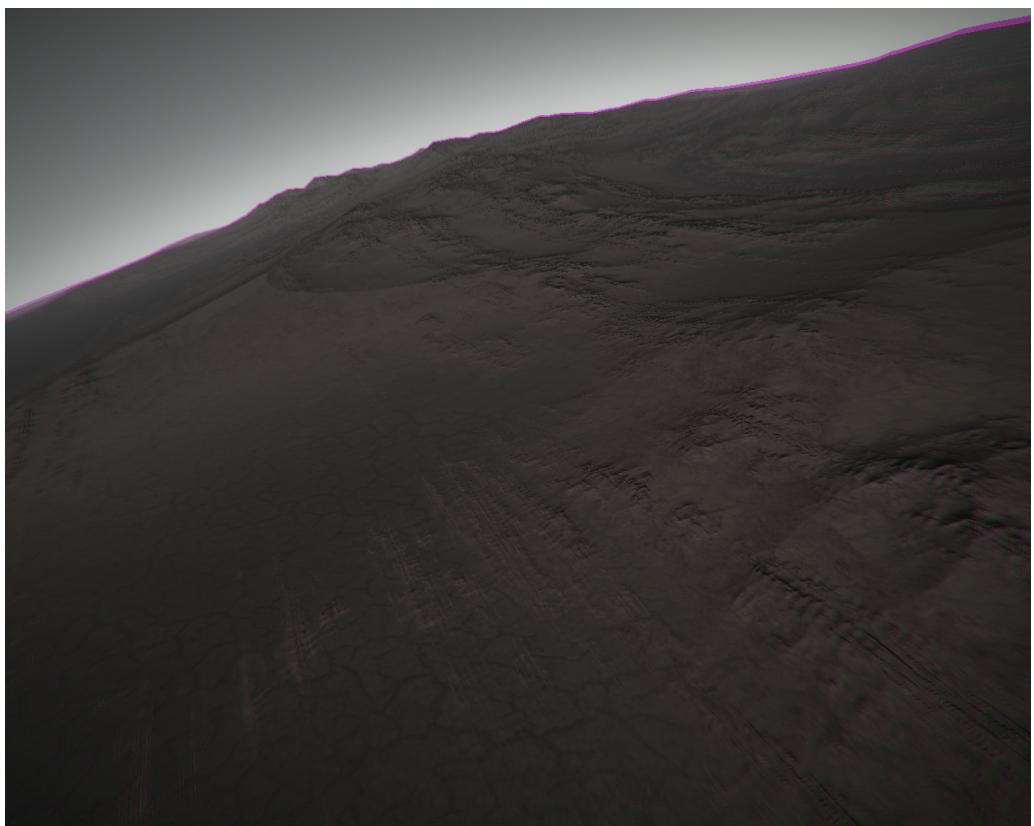


Figure 20: Image taken by HAL9001 on the surface of Hiffre, after a successful landing.

## 5 Conclusion/Outlook

### 5.1 Conclusion

We have successfully landed our satellite on our target planet Hiffre. To do so, we have discussed engine simulations, planetary orbit calculations, satellite transfers, orientation software and atmosphere models. We have presented all basic steps to achieve a successful satellite mission.

We have made significant simplifications along the way, however. A real satellite mission requires much more precise modelling, and less assumptions. We were also not quite able to land where we expected to land. Nonetheless; most of our results seem reasonable, and are in agreement with the theoretical models we have developed. Our simulations passed all consistency checks which we devised, and correctly predicted the impact time, impact velocity and the polar angle.

By far the most time-consuming and frustrating part of this project, was determining the correction parameters necessary to make our model. This was difficult, as it was based purely on trial and error, and some simulations ran for multiple minutes each time. This demonstrates the necessity of well-programmed algorithms, but it also demonstrates that it may be worthwhile to further develop the analytic solutions, to get more exact results which require less tweaking.

### 5.2 Outlook

There are many ways in which this project could be developed further. Most prominent among these is of course the investigation of why we did not land where we expected to land. Furthermore, it would be very interesting to develop an algorithm which could determine a desired landing spot, based on coordinates chosen by the user. Due to time constraints, this was not possible, but it would have been optimal.

Another future project would be to investigate closely the assumptions that we have made throughout this project, and try to develop solutions which do not require these assumptions. It would be particularly interesting to see a solution to the satellite transfer which did not depend on a patched conic approximation, but which took into account all forces at all times. This may also reduce the need for a correction factor, though it would make the project significantly more advanced.

Another interesting aspect would be to investigate a more general orientation

algorithm, which is not dependent upon the satellite being in the equatorial plane. Once again, this would make the problem significantly more complex, but also more applicable.

A final point for future research may be the atmosphere calculations. These were rather rough, in particular with respect to determining  $\mu$ . Seeing as  $\mu$  might have been one of the main sources of error, as discussed in section 4.7, it would be worthwhile to investigate more sophisticated statistic techniques for extracting  $\mu$  from the spectral data - though this is of course a field in its own right. The atmosphere itself could also have been modelled more complexly, though this would have to be a project for an atmospheric physicist.

# Bibliography

## Acknowledgment

The author would like to express his gratitude for the support of his supervisors Robert Hagala and Frode Hansen, without whom, this project would not have been possible.

Additional gratitude must be expressed towards Daniel Heinesen, for interesting discussions and exchange of ideas throughout this project.

## References

- [Burrows, 1966] Burrows, R. R. (1966). The classical "sphere of influence". *NASA*.
- [E.Barrabes et al., 2004] E.Barrabes, G.Gomez, and J.Rodriguez-Canabal (2004). Gravitational assisted trajectories. <http://www.ieec.cat/hosted/webastro04/notes/gravity.pdf>.
- [Fitzpatrick, 2006] Fitzpatrick, R. (2006). The adiabatic atmosphere. <http://farside.ph.utexas.edu/teaching/sm1/lectures/node56.html>.
- [Hansen, 2016a] Hansen, F. (2016a). Ast1100 lecture notes part 1a statistics and gas dynamics: building the rocket engine. <http://www.uio.no/studier/emner/matnat/astro/AST1100/>.
- [Hansen, 2016b] Hansen, F. (2016b). Ast1100 lecture notes part 1d electromagnetic radiation. <http://www.uio.no/studier/emner/matnat/astro/AST1100>.
- [Hansen, 2016c] Hansen, F. (2016c). Ast1100 lecture notes part 1e hydrostatic equilibrium: modelling the planetary atmosphere. [http://www.uio.no/studier/emner/matnat/astro/AST1100/](http://www.uio.no/studier/emner/matnat/astro/AST1100).
- [Hansen and Hagala, 2016a] Hansen, F. and Hagala, R. (2016a). Satellite project, ast1100 part 3. [http://www.uio.no/studier/emner/matnat/astro/AST1100/](http://www.uio.no/studier/emner/matnat/astro/AST1100).
- [Hansen and Hagala, 2016b] Hansen, F. and Hagala, R. (2016b). Satellite project ast1100 part 4. [http://www.uio.no/studier/emner/matnat/astro/AST1100/](http://www.uio.no/studier/emner/matnat/astro/AST1100).
- [Katz et al., 2012] Katz, B., Dong, S., Fabrycky, D., and Socrates;, A. (2012). Are planetary systems flat? <https://www.ethz.ch/content/dam/ethz/special-interest/phys/astronomy/astro-dam/documents/education/seminars/2011-2012/Tremaine.pdf>.

[Schroeder, 2000] Schroeder, D. V. (2000). *An Introduction to Thermal Physics*. Pearson.

[Weisstein, ] Weisstein, E. W. Spherical coordinates.  
<http://mathworld.wolfram.com/SphericalCoordinates.html>.

[Widnall and Peraire, 2008] Widnall, S. and Peraire, J. (2008). Orbit transfers and interplanetary trajectories.

[Wilson, 2012] Wilson, J. (2012). Hohmann transfers.  
<http://jwilson.coe.uga.edu/EMAT6680Fa05/Bacon/hohmanntransfers.html>.

[Young, 2014] Young, P. (2014). The leapfrog method and other “symplectic” algorithms for integrating newton’s laws of motion.  
<http://young.physics.ucsc.edu/115/leapfrog.pdf>.

# Appendices

## A Numerical methods

There are three numerical methods which we employ multiple times in this project report: the Leapfrog integration algorithm, the  $\chi^2$  (or least square) algorithm and the Forward-Euler method for estimating derivatives. We will here give a brief overview of both of these methods. More extensive information on these methods are easily found, see e.g. [Young, 2014] or [Hansen, 2016b].

### A.1 Leapfrog integration

Assume that we have a second order differential equation of the form:

$$a = f(t, x, v) \quad (67)$$

Where  $a = x(t)''$  and  $v = x(t)'$ . Let us first assume that  $g$  is not a function of  $v$ . In this case, we discretize our equations at points  $t_i$ , giving us solutions at points  $x(t_i) = x_i$ . A standard way of solving such discretized equations is e.g. the forward Euler, where we simply compute the velocity in the next step by assuming it to be a linear function of time, i.e.  $v_i = v_{i-1} + x_i t_i$ , and similarly for the acceleration. This algorithm, however, is not optimal because it only considers points forward in time. Therefore, if one were to Taylor expand the function  $v(t)$ , the error would already be in the second derivative. An alternative, more elegant, solution is provided by the Euler-Cromer algorithm, which uses the previous acceleration to compute the velocity, but the new velocity to compute the position. The leapfrog method is in between these two. It alternates between velocity and position, always computing them half a timestep from each other. As discussed by [Young, 2014], this leads to a more stable algorithm, particularly for periodic motion (such as planetary movements). In this case, the leapfrog algorithm *exactly* conserves angular momentum, and is also expect to conserve energy over a full period of oscillation. A general scheme for the leapfrog algorithm can be written as follows:

$$\begin{aligned} x_i &= x_{i-1} + v_{i-1/2} dt \\ v_{i+1/2} &= v_{i-1/2} + a(t_i, x_i) dt \end{aligned} \quad (68)$$

This can be implemented numerically as shown in the following Pseudo-code:

```

1 v[0]=v0+0.5*dt #Leapfrog step
2 x[0]=x0
3 t[0]=t0
4 for i in range(1,N):
5     x[i]=x[i-1]+v[i-1]*dt
6     t[i]=t[i-1]+dt
7     v[i]=v[i-1]+a(t[i], x[i])*dt

```

Notice, that we never know the position and the velocity in the same time step. This is generally not a problem, unless  $a$  is a function of  $v$  (this is the case for our atmosphere calculations in section 2.5.3). In this case, we must take an additional step to get the velocity in the same timestep as the position. This is easily done; we simply subtract the half-step from earlier in every iteration, to give a (hopefully) good approximation to the velocity in the step  $i$ . This is shown in Pseudo-code below:

```

1 v_temp_previous=v0
2 v[0]=v0+0.5*dt #Leapfrog step
3 x[0]=x0
4 t[0]=t0
5 for i in range(1,N):
6     x[i]=x[i-1]+v[i-1]*dt
7     t[i]=t[i-1]+dt
8     v_temp=v[i-1]+a(t[i], x[i-1], v_temp_previous)
9     v[i]=v[i-1]+a(t[i], x[i], v_temp)*dt
10    v_temp_previous=v_temp

```

Note that this is not the most efficient way to do this - we do not have to store the velocity for more than one timestep, and we can store the acceleration without having to store the temporary velocity. However, this way of writing the code gives illustrates the algorithm well.

## A.2 The $\chi^2$ algorithm

This is an algorithm for selecting from a set of parameters, the parameters which make a certain quantity minimal. Assume, therefore, that we have a model for our data,  $F(x_1, x_2, \dots, x_n)$ , which is dependent upon  $n$  parameters. We wish to choose  $x_1, \dots, x_n$ , so that the difference between the observed data and the model is as small as possible. We measure this difference by:

$$\chi^2(x_1, x_2, \dots, x_n) = \sum_i \frac{(F(x_1, x_2, \dots, x_n) - f_i)^2}{\sigma_i^2} \quad (69)$$

Where  $f_i$  is our observed data point at a point  $i$ ,  $\sigma_i$  is the standard deviation of the noise of measurement  $i$ , and the sum extends over all data points. Our goal is thus to find the combination of  $x_1, x_2, \dots, x_n$  that minimize  $\Delta$ . We do this iteratively -

we loop through the possible parameter combinations and calculate  $\chi$  for each of them. Then we find the combination for which this is the smallest. This is the  $\chi^2$  algorithm. In the case of no noise (such as the case in section 2.4, where we try to find the image that best fits a given image from a range of reference images), this method simplifies to the least-square method. In this case, we can simply minimize the following quantity:

$$S(x_1, x_2, \dots, x_n) = \sum_i (F(x_1, x_2, \dots, x_n) - f_i)^2$$

Note that, because we adapt a brute-force approach, trying all possible combinations of parameters, we will have to severely limit the number of points and parameters we consider, to make this numerically feasible. We present a brief pseudo-code showing a possible implementation of this below. Note that this process can be vectorized, which leads to an immense speed-up. However, we present the most readable code below.

```

1 for i in range(parameter_1_values):
2     for k in range(parameter_2_values):
3         .
4         .
5         .
6         for y in range(parameter_n_values):
7             chi_square[i, k, ..., y]=0
8             for z in range(number_of_observation_points):
9                 chi_square[i, k, ..., y]+=((F(i,k,...,y)-obs(z))/(sigma(z)))**2
10
11 best_combination=min(chi_square)
12

```

### A.3 The Forward-Euler algorithm

We use this algorithm to estimate derivatives, in particular when calculating the velocity of our planets. The idea is to take the definition of the derivative:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

And discretize it, according to:

$$f'(x) \approx \frac{f_{i+1} - f_i}{h} \quad (70)$$

Where  $h$  is the distance between two points. A scheme to implement this, in Psueo-code, is shown below:

```

1 dfdx=(f[i+1]-f[i])/h

```

## B Proof of selected results

### B.1 Finding the value of G in our new units

We know from equation 28 what the velocity of our planet must be in order to achieve circular motion. Let us therefore assume that earth travels in a perfect circle. Then it will complete one entire revolution in one year. In one year, the earth travels  $2\pi$  AU, so that the speed is given by:

$$v_{\text{Earth}} = 2\pi \text{ AU/year}$$

Inserting this into equation 28, and using that  $M_T = M_\odot = 1$ , and  $r = 1\text{AU} = 1$  gives:

$$v_{\text{SO}} = \sqrt{GM_\odot/\text{AU}} = 2\pi\text{AU/year}$$

Which gives:

$$G = 4\pi^2 \text{ AU}^3 \text{year}^{-2} M_\odot^{-1}$$

As was to be shown.

### B.2 Finding the extreme values of $x_{\text{picture}}$ and $y_{\text{picture}}$

We wish to find the maximum of equation 32 and 33 respectively. First, note that  $x_{\text{picture}}$  and  $y_{\text{picture}}$  describe a square. Thus, the maximum of  $x_{\text{picture}}$  will be constant over the entire range of  $y_{\text{picture}}$ , and vice-versa. The same also holds for the minimum. Therefore, we can choose a convenient spot along the respective axis at which to calculate the maximums/minimums. One convenient choice is to find max/min of  $x_{\text{picture}}$  at  $y_{\text{picture}} = 0$ , and equivalently for the max/min of  $y_{\text{picture}}$ . At the maximum of  $x_{\text{picture}}$  it is clear that  $\phi - \phi_0 = \alpha_\phi/2$ , and (choosing to find the maximum along  $y_{\text{picture}} = 0$ )  $\theta = \theta_0 = \pi/2$ . Inserting these values into equation 32 gives:

$$x_{\text{picture},\max} = \frac{2 \sin(\pi/2) \sin(\alpha_\phi/2)}{1 + \cos(\pi/2) \cos(\pi/2) + \sin(\pi/2) \sin(\pi/2) \cos(\alpha_\phi/2)} = \frac{2 \sin(\alpha_\phi/2)}{1 + \cos(\alpha_\phi/2)}$$

For the minimum, simply substitute  $\alpha_\phi/2$  with  $-\alpha_\phi/2$  (because then  $\phi - \phi_0 = -\alpha_\phi/2$ ) in the above expression. This gives:

$$x_{\text{picture},\min} = -\frac{2 \sin(\alpha_\phi/2)}{1 + \cos(\alpha_\phi/2)}$$

Using that  $\sin(-\theta) = -\sin(\theta)$  and  $\cos(-\theta) = \cos(\theta)$ .

We can find the equivalent expressions for  $y_{\text{picture,max}}$ , by letting  $\phi = \phi_0$ . Note, however, that the  $y$ -axis is flipped (low  $y$ -values at the top and high values at the bottom). It follows from this that the max (largest)  $y$  is located at  $\theta - \theta_0 = \theta - \pi/2 = -\alpha_\theta/2 + \pi/2$ . From this it follows that  $\theta = -\alpha_\theta/2 + \pi/2$ . Inserting gives:

$$y_{\text{picture,max}} = \frac{2 \sin \pi/2 \cos(-\alpha_\theta/2 + \pi/2) - 2 \cos(\pi/2) \sin(-\alpha_\theta/2 + \pi/2) \cos(\phi_0 - \phi_0)}{1 + \cos(\pi/2) \cos(-\alpha_\theta/2 + \pi/2) + \sin(\pi/2) \sin(-\alpha_\theta/2 + \pi/2) \cos(\phi_0 - \phi_0)}$$

To simplify this expression, I use the sum formula for the sine and the cosine, specifically that:

$$\cos(a + b) = \cos(a) \cos(b) - \sin(a) \sin(b)$$

$$\sin(a + b) = \sin(a) \cos(b) + \cos(a) \sin(b)$$

This gives after some elementary algebra:

$$y_{\text{picture,max}} = \frac{2 \sin(\alpha_\theta/2)}{1 + \cos(\alpha_\theta/2)}$$

If we now exchange  $-\alpha_\theta/2$  with  $\alpha_\theta/2$ , we get:

$$y_{\text{picture,min}} = -\frac{2 \sin(\alpha_\theta/2)}{1 + \cos(\alpha_\theta/2)}$$

So that finally:

$$x_{\text{picture,max/min}} = \pm \frac{2 \sin(\alpha_\phi/2)}{1 + \cos(\alpha_\phi/2)} \quad (71)$$

$$y_{\text{picture,max/min}} = \pm \frac{2 \sin(\alpha_\theta/2)}{1 + \cos(\alpha_\theta/2)} \quad (72)$$

Which are the equations given in the main text.

### B.3 Solving the triangulation equation

Here we complete the calculations from section 2.4.2.

I will begin by solving the equation 38 and 39, and then simply check which of the resulting points fit equation 40. Writing equations 38 and 39 out gives:

$$x^2 - 2ax + a^2 + y^2 - 2yb + b^2 - r_1^2 = 0$$

$$x^2 - 2xc + c^2 + y^2 - 2yd + d^2 - r_2^2 = 0$$

Subtracting:

$$2x(c - a) + 2y(d - b) + a^2 + b^2 - c^2 - d^2 + r_2^2 - r_1^2 = 0$$

This equation can now be explicitly solved for  $x$ , giving:

$$x = \frac{r_1^2 + c^2 + d^2 + 2y(b - d) - a^2 - b^2 - r_2^2}{2(c - a)}$$

Let us define:

$$\gamma = \frac{r_1^2 + c^2 + d^2 - a^2 - b^2 - r_2^2}{2(c - a)} - a$$

$$\lambda = \frac{b - d}{c - a}$$

Then:

$$(x - a) = \lambda y + \gamma$$

Which gives, when inserted:

$$(\lambda y + \gamma)^2 + (y - b)^2 - r_1^2 = 0$$

Writing this out gives:

$$(\lambda y)^2 + 2\gamma\lambda y + \gamma^2 + y^2 - 2yb + b^2 - r_1^2 = 0$$

Which can be rearranged to give:

$$(\lambda^2 + 1)y^2 + 2(\gamma\lambda - b)y + \gamma^2 + b^2 - r_1^2 = 0$$

This is a quadratic equation, which can be solved to give:

$$y_{1,2} = \frac{-2(\gamma\lambda - b) \pm \sqrt{4(\gamma\lambda - b)^2 - 4(\lambda^2 + 1)(\gamma^2 + b^2 - r_1^2)}}{2(\lambda^2 + 1)}$$

Each of these  $y_s$  can be inserted into the  $x$  equation above, to also give a possible  $x$ -position. These two possible positions can be compared to the final equation to give the correct position.

## B.4 Finding the velocity of the satellite

Here we will show how to arrive at equation 44.

Assume that we know the angle to our reference stars, as seen from our star. Let the angle to the first reference star be denoted by  $\phi_1$ , and the angle to the

second reference star by  $\phi_2$ . Unit vectors in the direction of these stars,  $\vec{u}_1$  and  $\vec{u}_2$ , can then easily be defined as:

$$\vec{u}_1 = \begin{pmatrix} \cos \phi_1 \\ \sin \phi_1 \end{pmatrix} \quad \vec{u}_2 = \begin{pmatrix} \cos \phi_2 \\ \sin \phi_2 \end{pmatrix}$$

This follows from simple trigonometry. Now, by a standard theorem in linear algebra, these vectors are linearly independent if and only if the determinant of the matrix with these vectors as columns is nonzero. This matrix is given by:

$$\mathbf{A} = \begin{pmatrix} \cos \phi_1 & \cos \phi_2 \\ \sin \phi_1 & \sin \phi_2 \end{pmatrix} \quad (73)$$

With determinant:

$$\det(\mathbf{A}) = \cos \phi_1 \sin \phi_2 - \cos \phi_2 \sin \phi_1 = \sin(\phi_1 - \phi_2)$$

This determinant is zero if and only if  $\phi_1$  and  $\phi_2$  differ by an integral multiple of  $\pi$ . If this is the case, then the stars are collinear, and we cannot find a basis with which to compute the velocity of the satellite.

If this is not the case, then we can expand the velocity of the satellite in the standard basis in terms of the basis vectors  $\vec{u}_1$  and  $\vec{u}_2$ . This will give us the change of coordinate matrix specified by  $\mathbf{A}$  above. However, we want the inverse of this matrix - we have the velocity of the satellite with respect to the basis  $\vec{u}_1$  and  $\vec{u}_2$  and wish to find the velocity with respect to the standard basis. A standard theorem in linear algebra states that the inverse of a  $2 \times 2$  is given by the following relation:

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \implies \mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Employing this formula to the matrix  $A$  in equation 73 gives:

$$\mathbf{A}^{-1} = \frac{1}{\sin(\phi_1 - \phi_2)} \begin{pmatrix} \sin \phi_2 & -\cos \phi_2 \\ -\sin \phi_1 & \cos \phi_1 \end{pmatrix}$$

Which is precisely the change of coordinate matrix given in equation 44.

## B.5 Atmosphere calculations

Here we show how to arrive at equation, 55, 56 and 57.

As stated in the main text, we require two important equations, the ideal gas equation:

$$P = \frac{\rho k T}{\mu u}$$

And the equation of hydrostatic equilibrium:

$$\frac{dP}{dr} = -G\rho(r) \frac{M}{r^2}$$

We will first investigate the adiabatic model.

### B.5.1 Adiabatic model

As explained in section 2.5.3, adiabatic implies that the following equation holds:

$$P^{1-\gamma} T^\gamma = A$$

Where  $A$  is constant. Let us, as is done by [Fitzpatrick, 2006], implicitly differentiate the above equation with respect to temperature. This gives:

$$\frac{d}{dT} (P^{1-\gamma} T^\gamma) = 0$$

Which can be written out as:

$$(1 - \gamma) P^{-\gamma} T^\gamma \frac{dP}{dT} + \gamma T^{\gamma-1} P^{1-\gamma} = 0$$

Rearranging:

$$\frac{P}{T} = \frac{\gamma}{\gamma - 1} \frac{dP}{dT}$$

Solving the ideal gas equation for the density gives:

$$\rho = \frac{\mu u}{k} \frac{P}{T}$$

Inserting for  $P/T$ :

$$\rho = \frac{\gamma - 1}{\gamma} \frac{\mu u}{k} \frac{dP}{dT}$$

I can now insert this relation into the hydrostatic equilibrium to "cancel out" the differential  $dP$ <sup>20</sup>, giving:

$$\frac{dT}{dr} = -\frac{\gamma - 1}{\gamma} \frac{\mu u}{k} \frac{GM}{r^2}$$

---

<sup>20</sup>Mathematicians, please look away

This is a straightforward separable differential equation:

$$dT = -\frac{\gamma-1}{\gamma} \frac{GM\mu u}{k} \frac{1}{r^2} dr$$

Integrating gives:

$$T = \frac{\gamma-1}{\gamma} \frac{GM\mu u}{k} \frac{1}{r} + C_1$$

This is the first of equation 53. The other can be readily found, simply by inserting into the adiabatic atmosphere equation, to give:

$$P = AT^{\frac{\gamma}{\gamma-1}}$$

The density can now be found by inserting into the ideal gas equation.  $P/T$  is given by:

$$\frac{P}{T} = AT^{\frac{1}{\gamma-1}}$$

Giving the density as:

$$\rho = \frac{\mu u}{k} AT^{\frac{\gamma}{\gamma-1}}$$

Which thus gives all of equation 53.

### B.5.2 Isothermal model

Once the radius is above  $r_{T_0/2}$ , we assume an isothermal atmosphere, i.e. constant temperature. This makes the equation much easier, as the hydrostatic equilibrium now reads:

$$\frac{dP}{dr} = -\frac{GM}{r^2} \frac{\mu u}{kT} P$$

Which is a separable differential equation, which can be solved as:

$$\frac{1}{P} dP = -\frac{GM}{r^2} \frac{\mu u}{kT} dr$$

Integrating gives:

$$\ln P = \frac{GM}{r} \frac{\mu u}{kT} + C_2$$

Where  $C_2$  is a constant. Rewriting:

$$P = C_2 \exp \left( \frac{GM}{r} \frac{\mu u}{kT} \right)$$

Realizing that  $T = T_0/2$  and inserting in the ideal gas equation gives:

$$\rho = \frac{2\mu u}{kT_0} C_2 \exp \left( 2GM\mu u / rkT_0 \right)$$

Which thus gives all of equation 54.

### B.5.3 Implementing the boundary conditions

The constant can be fixed by boundary conditions, as explained in section 2.5.3.

To fix  $C_1$ , realize that the temperature must be  $T_0$  at the surface of the planet, where  $r = R_p$ , the radius of the planet. Inserting gives:

$$T_0 = \frac{\gamma - 1}{\gamma} \frac{GM\mu u}{k} \frac{1}{R_p} + C_1$$

Which immediately implies that:

$$C_1 = T_0 + \frac{1 - \gamma}{\gamma} \frac{GM\mu u}{k} \frac{1}{R_p}$$

Which gives the explicit dependence of temperature upon radius as:

$$T(r) = \frac{\gamma - 1}{\gamma} \frac{GM\mu u}{k} \left( \frac{1}{r} - \frac{1}{R_p} \right) + T_0$$

To find A, I require that  $\rho = \rho_0$  when  $r = R_p$ , which gives:

$$\rho_0 = \frac{\mu u}{k} A T_0^{1/(\gamma-1)}$$

I.e:

$$A = \frac{\rho_0 k}{\mu u T_0^{1/(\gamma-1)}}$$

Which gives the adiabatic pressure as:

$$P(r) = \frac{\rho_0 k}{\mu u T_0^{1/(\gamma-1)}} T^{\frac{\gamma}{\gamma-1}}$$

And the adiabatic density as:

$$\rho = \rho_0 \left( \frac{T}{T_0} \right)^{1/(\gamma-1)}$$

We will model the atmosphere as transitioning to an isothermal atmosphere when  $T = T_0/2$ . This happens for  $r = r_{T_0/2}$ , which can be found from:

$$-\frac{T_0}{2} = \frac{\gamma - 1}{\gamma} \frac{GM\mu u}{k} \left( \frac{1}{r_{T_0/2}} - \frac{1}{R_p} \right)$$

Which gives:

$$-\frac{k}{GM\mu u} \frac{T_0 \gamma}{2(\gamma - 1)} = \frac{1}{r_{T_0/2}} - \frac{1}{R_p}$$

I.e:

$$\frac{1}{r_{T_0/2}} = \frac{1}{R_p} - \frac{T_0}{2} \frac{\gamma}{\gamma - 1} \frac{k}{GM\mu u}$$

Now the constant  $C_2$  can be easily fixed, by requiring that  $\rho = \rho_{T_0/2}$  (from the adiabatic model) when  $r = r_{T_0/2}$ , which gives the isothermal density as:

$$\rho_{T_0/2} = \frac{2\mu u}{kT_0} C_2 \exp\left(\frac{2GM\mu u}{r_{T_0/2}kT_0}\right)$$

Which gives:

$$C_2 = \frac{kT_0\rho_{T_0/2}}{2\mu u} \exp\left(-\frac{2GM\mu u}{r_{T_0/2}kT_0}\right)$$

From which it follows that:

$$\rho = \rho_{T_0/2} \exp\left(\frac{2GM\mu m_H}{kT_0} \left(\frac{1}{r} - \frac{1}{r_{T_0/2}}\right)\right)$$

Which thus give equations 55, 56 and 57. in the main text.

## C Pertinent parameters

Table 7: Data about our solar system

Planet	Planet Name	Mass, $M_p [M_\odot]$	Radius, $R_p [\text{km}]$	Surface density, $\rho_0 [\text{kg m}^{-3}]$
0	Byappo	$1.005 \times 10^{-5}$	9503	1.053
1	Domin	$1.679 \times 10^{-6}$	5398	8.579
2	Munnmon	0.000644	64503	22.502
3	Pjeng	$3.852 \times 10^{-8}$	1526	1.200
4	Plaging	0.000622	67444	24.613
5	Psiwe	$3.333 \times 10^{-7}$	2913	1.097
6	Hiffre	$9.463 \times 10^{-8}$	2058	1.426

Table 8: Further data about our solar system

Planet	Rotational period, $T$ [earth days]	Semi-major axis, $a$ [AU]	Eccentricity of orbit, $e$
0	0.944	3.16	0.003
1	1.707	4.19	0.002
2	0.531	28.62	0.062
3	12.831	5.58	0.015
4	0.374	16.25	0.102
5	27.990	11.80	0.046
6	13.151	7.45	0.024

Table 9: Data about our star

Star surface temperature, $T_*$ [K]	7969
Star radius, $R_*$ [km]	1638325
Star mass, $M_*$ [ $M_\odot$ ]	2.561

Table 10: Parameters of our satellite

Mass of satellite, $M_{\text{sat}}$ [kg]	1100
Mass of lander module, $M_{\text{lander}}$ [kg]	90
Surface area of satellite, $A_{\text{sat}}$ [ $\text{m}^2$ ]	6
Surface area of lander module, $A_{\text{lander}}$ [ $\text{m}^2$ ]	0.3

Table 11: Parameters used in the simulation of a single cell of a rocket engine

Length of box, $L$ [m]	$10^{-6}$
Number of particles	$10^5$
Temperature, $T$ [K]	$10^4$
Mass of satellite, $M_{\text{sat}}$ [kg]	1100