Part 6: Arriving at the Planet

Modeling the Atmosphere

When landing on a planet it is important to take the atmosphere of the planet into account. The lander will experience a strong friction force which (1) will slow down the space craft and (2) may rip the space craft apart if it is too strong. In order to reduce the speed and thereby the frictional forces (as well as the landing velocity to avoid crash), you need a parachute which size you need to decide.

In this part you will make a model for the atmosphere which you will need in part 7 to calculate the orbit of the lander. In order to be able to model the atmosphere, you need to know which molecules it consists of. You need to do this by solving exercise 1D.7 in part 1D. Note that for the project, you **only** need to identify **which** elements are present in the atmosphere. You do **not** need the temperature of the different gases (although you will need to calculate these for the fitting procedure and you also need that information to decide which lines are real) and you do **not** need to sketch the layers of the atmosphere where you find these gases. In the project we will assume that these gases are present in equal relative amounts everywhere in the atmosphere (but the total density changes as a function of height).

In order to get a model for the density of your atmosphere as a function of height h above the surface, you will need to first **read through** exercise 1E.1 from part 1E in detail. Then you need to model the atmosphere in the following way:

1. First you need to find mean molecular μ weight of your atmosphere using information from 1D.7. How to do this is described in the first point of exercise 1E.1.

2. You now need the surface temperature T_0 that you calculated for your destination planet in part 3.

3. Assume the atmosphere to be adiabatic (see exercise 1E.1) up to the height where $T = T_0/2$ and isothermal for heights above the point where $T = T_0/2$ (**note:** T_0 in Kelvin!)

4. Solve for the density profile (and simultaneously the temperature profile) of the planet's atmosphere using the equation of hydrostatic equilibrium and the assumption of ideal gas. **Hint:** you may choose to do this analytically (different expressions in two different zones) or numerically. Ideally you want to include a varying gravitational constant if you do it numerically. You may find more hints/ideas in exercise 1E.1.

Some hints for finding the correct gas particles in your target atmosphere.

1. The text file containing the spectrum your satellite measures when orbiting your target planet can be found on the course webpage, there is a link in exercise 1D.7. The first column is wavelengths in nm and the second column is normalized flux.

2. NB! There are 10 million data points in the file, this should give two numpy arrays (λ and flux) of approx 100 MB each.

3. Use χ^2 fitting (see part 1A) to fit lines corresponding to different gases to your data. Because of noise, some of the lines might not be visible to the naked eye.

4. To reduce computing time, just compare a couple of hundred data points around the line center you are testing instead of the whole array. No need to check millions of times if the flux far outside of the line is flat.

5. We assume the atmosphere consists of equal parts of each gas line you find. Whenever you find an absorption line, the gas corresponding to that line should be included.

6. You will not always find all lines from the same gas, and all the gases are not neccessarily the same temperature, but different lines from the same gas will be at the same temperature.

Air Resistance and Terminal Velocity

When the satellite enters the atmosphere it will experience an air resistance. We will model this force using the drag equation:

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

where F_D is the drag force, ρ is the density, A is the cross sectional area of the satellite (or lander, with or without parachute), v is the velocity relative to the atmosphere and C_D is a dimensionless drag coefficient that we will set equal to 1 for simplicity.

When (in the next part) the lander tavels through the atmosphere it will slow down until it reaches its terminal velocity, this is the velocity at which the gravitational and drag forces are equal and sum up to zero, giving a constant velocity.

Exercise 6.1: Show that the terminal velocity **close to the surface of the planet** is given by:

$$v_t = \sqrt{\frac{2GMm}{R^2\rho_0 A}},$$

where M is the mass of the planet, m is the mass of the satellite and R is the radius of the planet.

Exercise 6.2: Calculate the parachute area needed to achieve a terminal velocity of exactly 3 m/s at the surface.

Scouting out the Planet

In this part the goal is for you to get into a very close orbit to the planet and to take a series of pictures of the surface. You are then to decide where you want the lander module to land.

The simulations that you will do will be similar to what you have done until now, but will be different in a number of ways:

- We disregard all the other planets but the one we are landing on, you should be close enough to this planet in order for this to be a very good approximation. Just place the planet stationary at the origin of your coordinate system (the x and y axis of the new coordinate system should still point in the same directions as in the previous parts).
- We will include the effect of the atmosphere of the planet, using the model that we have developed.
- We will now use all three dimensions.
- We will use SI units.

The procedure here will be very similar to part 3 and 5, only in three dimensions and including the atmosphere drag. Remember that the atmosphere drag acts in the opposite direction of the velocity.

Goal: Simulate the satellite orbit and figure out what instructions to give in order for it to get into a stable orbit as close to the planet as possible; **Warning:** make sure that your orbit is so high that the atmospheric density (calculated above) is basically zero. If you try to enter an orbit which is too low, the atmospheric friction will make you crash! Staying above $F_G/F_D > 1000$ should be considered safe.

Once you have found a suitable set of instructions, call the `land_on_planet(target_planet, 'commandfile.txt')` function with a command-file, like in part 5. This command file should start with an init command, which loads the position and velocity of the satellite at the end of the `send_satellite` function call. The initial position and velocity is the one you have after the final boost in part 5 (the one that should put you in a nice circular orbit). The initial time is the time of the last boost performed in sendSatellite. The time of the subsequent commands should be given in *seconds after this initial time*. Note that all quantities should now be given in SI units (as opposed to AU and years as before).

Take a picture of the planet from the position that you start in, can you see the planet clearly? Can you see any details? Get into a stable orbit close to the planet and take at least 10 pictures of different parts of the planet, pick a suitable spot to land on and give the coordinates (θ, ϕ) of this point to the group teacher. When finding these coordinates remember to take into account the rotation of the planet since the init-command.

- The `land_on_planet` function takes as input an integer selecting the planet you are landing on, and a text-file containing instructions. All times are in seconds after the init. The possible instructions are as follows.
  (i) `init` : No parameters are required (and indeed, no parameters allowed), since this data is taken directly from the end result of the sendSatellite module. I.e. this module begins where part 5 left off. This line is mandatory.
  (ii) `boost` : Change the velocity vector. A time t; and Δv_x, Δv_y and Δv_z, needs to be given.
  (iii) `orient` : Takes a time. Will return the coordinates and velocity of the satellite, relative to the planet.

(iv) `picture` : A time and direction is needed. The direction is given in terms of a θ and ϕ (spherical polar coordinates) and an up-direction. The up-direction is given as a 3D vector. Inserting your current xyz coordinates as the up direction will result in a picture taken "naturally" along the horizon of the planet.

(v) `video` : Same as in part 5. First option is where your camera is focused on the planet you're landing on. Commands are first line video $time_1$ planetNumber. Second line is video $time_2$ planetNumber. The other option is manual camera rotation. First command line is video $time_1$ θ_1 ϕ_1 and the second is video $time_2$ θ_2 ϕ_2.

The goals of this part

In addition to calculating the atmopsheric density profile and terminal velocity, the goal of this section is to take pictures and videos from a low orbit around the planet, and decide on a landing spot. When calculating the θ and ϕ of the landing spot, remember to include the planet rotation from the time of the init command.

Hints :

- Try to get into successively closer orbits until you are as close as you need.
- Get as close as possible without running into the atmosphere, make sure $F_G/F_D > 1000$.
- When calculating the atmosphere drag, remember that the atmosphere rotates perfectly with the planet.
- Use the period of the planet to calculate the rotational velocity. Planets rotate around the z-axis following the right-hand rule. At time $t = 0$ after `init`, $\phi = 0$ is along the x-axis.
- Take a lot of pictures and videos of the planet as you get closer and closer, since they are pretty cool to look at!
- When taking a video, you can still focus on the center of the planet by choosing your target planet in the `video` command, however, you will probably get much nicer videos if you tilt the camera up manually, to look at the horizon or at the moons of your planet.