

Part 7: The Landing

This is the final part of the project. All your hard work has led to this! You have announced the coordinates where you want to land, now it is time to launch the lander module from your satellite, and make it to the surface.

You will continue the simulation in the same environment as in part 6, only you now have three new commands. You can launch the lander using the command "launchLander", and you can specify the strength of the landing engine, as well as the height of deployment, using the "landing" command. The lander can also deploy a parachute to increase the drag force, this is done with the command "parachute". You do not have to think about fuel in this part.

After the lander is launched you do not have to model the rest of the satellite, but only the lander itself. The lander cannot use the boost command, but has a landing engine that will engage at a specific height above the ground and give a constant force in the direction radially upwards until the satellite reaches ground. In part 6 you found an approximate expression for the terminal velocity of the lander. We will call a landing safe if you reach the ground with less than 3 m/s **radial** velocity. You need to make sure that the frictional forces from the atmosphere at no point during the landing exceed 250000 N as this will rip the satellite apart.

Launch the lander from the satellite with the correct velocity relative to the satellite, and deploy the parachute and landing engine such that you land safely at the place you set out to land.

Exercise 7.1: If the terminal velocity, v_t , is not slow enough for the satellite to land safely, we will want to engage a landing engine once we get close to the ground in order to slow down the satellite to a safe speed. Lets say that the landing engine supplies a constant force F_L pointing radially upwards, what should this force be? Show that in order to slow the satellite down to a safe speed v_{safe} , the force must be given by:

$$F_L = \frac{1}{2}\rho_0 A (v_t^2 - v_{\text{safe}}^2).$$

(Hints: Equate the total force slowing down the satellite with the gravitational force: $F_D + F_L = F_G$, then rewrite the gravitational force in terms of the terminal velocity.)

Your task is therefore,

1. To do simulations to find (1) the point in orbit from which you want to launch your lander, (2) the velocity (amplitude and direction) with which you want to launch the lander, (3) the size and time for deploying the parachute, and (4) (if you need it) the constant force and starting time of the landing engine.
2. Make sure that the frictional force from the atmosphere never exceeds the limit of 250000 N at any point during the simulation of your descent. Staying below 25000 N is recommended. The area of the lander before the parachute is deployed is 0.3 m^2 .
3. Make sure the radial velocity when you reach the surface in your simulation does not exceed 3m/s.

4. If you manage to adjust these parameters in the simulations such that the lander goes several times around the planet at steadily decreasing altitude, you will get a much cooler landing video. The lower the radial velocity, and the longer you manage to stay in the air at lower altitude, the better the landing video. Once the lander “hits” the atmosphere though, it will probably slow down its tangential velocity.
5. Make sure the time step in your simulated landing is sufficiently small. As always you can check this by rerunning the simulation with half the time step and see if the changes to the landing orbit are small.
6. Take into account the rotation of the planet: try to avoid landing on the night side of the planet, then you won’t see very much.
7. Now do the landing for ‘real’. Do you manage to land safely or does the lander crash or burn?
8. If it crashes, remember that the largest uncertainty in your simulation is the mean molecular weight μ which was very hard to estimate from the spectra. Try to slow down even more in your simulations to take into account the possibility that the atmosphere is thinner or thicker than you thought. If your lander is ripped apart before landing it probably means a higher atmospheric density, if it crashes at landing it probably means the atmosphere is thinner than you thought. But it could also be an error in your simulation.

Hints :

- Be careful when setting the force for the landing engine, the problem becomes very unstable if this force is too strong.
- If you deploy the landing engine very close to the ground, make sure that you use a very small time step in your simulation, in order to model this force well.
- The `landOnPlanet` 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` : Same as in part 6.
 - (ii) `boost` : Same as in part 6.
 - (iii) `picture` : Same as in part 6.
 - (iv) `video` : Same as in part 6.
 - (v) `launchLander` : Launches the lander at a time given. Also needs a launch velocity v_x, v_y, v_z relative to the satellite.
 - (vi) `parachute` : Deploys parachute. Needs a time, and optionally a surface area in m^2 (defaults to 42m^2).
 - (vii) `landing` : Sets a time after the `init` when the lander turns off (you do not need to give the exact time of the landing, but you should insert a time where you are sure the landing has already happened. See example.). Also sets the height of deployment for the landing engine (in meters), and the strength of the engine (in Newtons). The last two arguments can be 0 for no landing engine.

An example input file might look like this (please note that this is an *example* and you will most likely have to use more precision (more significant digits) in an *actual* command file):

```

1  init
2  boost      0.500e3  0.0  -3.0e3  0.0
3  launchlander 0.600e4  0.0  4.6e3  0.0
4  parachute  0.650e4  35
5  picture    0.940e4  1.5  -3.142  0.0  0.0  1.0
6  landing    1.000e4  30  30000

```

where

```

1  init

```

means we are initializing the landing module. Note that no input is required (and indeed, no input is allowed), since this data is taken directly from the end result of the sendSatellite module. The next line,

```

2  boost      0.500e3  0.0  -3.0e3  0.0

```

is just a boost, done in the same way as in part 4, except now we need three input coordinates since we are working in 3D. The next line,

```

3  launchlander 0.600e4  0.0  4.6e3  0.0

```

means we are launching the lander at a time $t = 0.6 \cdot 10^4$ s after the initial time. The lander is given an initial velocity boost of $\Delta \mathbf{v} = (0, 4.6 \cdot 10^3, 0)$ m/s. The next line,

```

4  parachute  0.650e4  35

```

means we are deploying a parachute at time $t = 0.65 \cdot 10^4$ s, with surface area of 35 m^2 . The next line after this,

```

5  picture    0.940e4  1.5  -3.142  0.0  0.0  1.0

```

means we are taking a picture at time $t = 0.94 \cdot 10^4$ s. This is done in the same way as in part 4. The next line,

```

6  landing    1.000e4  30  30000

```

means the landing engine is ready to deploy and starts once the satellite is 30 m above the surface of the target planet. The strength of the landing engine's thrust is given in newtons, here 30000 N. The first argument, $1.000 \cdot 10^4$ s means the actual landing should happen *before* this point in time.

Your final goals for this section and the whole project

You need to give the correct commands to the lander in order to land safely and as close as possible to your chosen landing site. You need to make a video of your landing. It will be very difficult to land precisely on (or even close to) your chosen landing site, so we will be **very** tolerant here.

Hints :

- You can expand your command file from part 6 to do part 7.
- When taking a landing video, you will get *much* better results if you instead of focusing directly at the planet, tilt your camera along the horizon to see a sunrise, moonrise, clouds, plants, etc.
- If your lander burns due to air friction, it probably reached a too high velocity before hitting the atmosphere. Try orienting the lander before it burns and reduce the orbital height of the satellite before launching the lander again (yes, we have access to time machines in case of errors like these...)