SMCEF P2
Reentry
v1

André Wemans

June 5, 2024

Contents

1 Introduction

The second SMCEF project is to use a model of a space module reentering Earth atmosphere to obtain the reentry parameters that ensure the trajectory to be inside some constrains.

The project is to be implemented by groups of 2 or 3 students, but notice that bigger groups are expected to deliver more and better work than groups of two students.

The objectives of the project are:

- Implement the model as described in these guidelines
- Simulate several reentry parameters using a forward method
- From the simulations obtain the reentry parameters that ensure reentry restrictions
- Use an implicit method to repeat the simulation
- Compare results
- Although not explicitly requested code and execution time optimization are expected

The reentry process can have two very different objectives, for the case of decommissioned satellites or *space junk* it can be used to incinerate them in the reentry and by this way unclutter the Earth orbit.

On the other hand, it is also used to remove most of the kinetic energy from a module allowing it to land with safe velocities. This method has been used due the prohibitive costs of safely bring a module from orbit to Earth surface relying on thrusters, since that would imply to launch the craft with at least the double of fuel used at lift off.

The reentry parameters relevant for the reentry conditions are velocity and downward angle with the horizon at the reentry interface, i.e. a fixed altitude from Earth surface. For having an estimation where the module will land (or ocean landing) it is also important to know the above Earth position and direction of flight, but we will not consider that matter.

The reentry process is a very sensitive phase since the kinetic energy removed from the module will produce large amounts of energy that can cause damage to the module structure or even destroy it, as sadly it happened with the Space Shuttle Colombia in 2003. The modules use heat shields designed to protect the vessel by absorbing thermal energy while their layers are evaporate, and the vessel design is also important to divert most of the thermal energy from the vessel walls. We also will not be considering the thermal problem, since it would need other sub models to be able to capture that problem.

The three aspects that our model will have in consideration for regarding a successful reentry trajectory are the maximum level of gs sustained by the vessel and any astronaut inside of the module, touchdown velocity and distance from reentry interface to landing position as projected in the Earth surface.

The vessel is designed to lose velocity through air drag, to also create some lift force that helps maintain a trajectory, allowing more time in the descendant phase and in this way to remove more kinetic energy since will have a longer trajectory. These are the main aspects that our model will try to capture and use for the simulations.

2 Model

The space reentry model to be used is a very simplified one, only considering the air resistance force seen in the classes, and also some of the Newton laws, including the gravitational Newton law.

The air resistance empiric law to be used is:

$$F_d = -\frac{1}{2}\rho A C_d v^2$$

with ρ the air density, A the surface considered for the drag coefficient, C_d the drag coefficient that depends on the geometry of the body (it also has some dependence on the velocity but we will consider it constant) and v the velocity.

We will consider two reentry stages, before and after the opening of the parachutes.

Before the parachutes deploying the space module will have a drag resistance force that will depend on the total velocity of the vessel, will be always aligned and of contrary direction of the velocity vector. This is one of the differences between what has been seen in the classes, where the body was falling vertically and all motion was in one direction.

The space module will also have a lift force, i.e. upward on the y axis, also obtained from the air resistance empiric law described before but with a different drag coefficient.

After the parachutes deploying the space module will have the same drag resistance as before, but no more lift component, added to the drag resistance from the parachutes, that will have different parameters A and C_d. This last one will also always be aligned with the velocity vector and in opposite direction.

In a different aspect from the problems seen in the classes the gravitational acceleration will not be constant and must be computed for each vertical position in the trajectory.

2.1 Air density

Like the gravitational force, the air density will be different for each given altitude above Earth surface. But contrary to the gravitational force there is not a single and simple equation that allows us to compute its value for a given altitude.

This is a very usual problem when a model is being build, it needs information that is not simply available. It would be possible to attempt to build another model to address that aspect, but it is complex since depends on many factors like air temperature that also depends on how solar radiation interacts with the atomic and molecular air constituents.

Searching for air density values it is possible to find the information of values of air density for some altitudes, as it is the case in The Engineering Toolbox, and can be seen in Table 1, the students can find this table in clip.

Since we need values between the data given in Table 1, and also outside the range in the table we need to infer them as best as possible.

First we plot the data points to have some idea on the general behavior, see Figure 1.

It would be possible to use a linear interpolation between any two points, and use the line obtained from the last two points to extrapolate the values outside. For relationships approximately linear or as a non precise estimation could be a solution. The problem with non linear relationships is lack of precision and non defined derivatives when from one interval to the other.

2.1.1 Function fit

Another possible solution would be to encounter a function that would fit well the data. Observing the Figure 1 one candidate function is an exponential function.

Using the *optimize curve_fit* from *scipy* to fit the data to

Table 1: Air density by altitude above sea level, taken from The Engineering Toolbox

Altitude above sea level / m	Density / kgm^{-3}
-1000	1.347
0	1.225
1000	1.112
2000	1.007
3000	0.9093
4000	0.8194
5000	0.7364
6000	0.6601
7000	0.5900
8000	0.5258
9000	0.4671
10000	0.4135
15000	0.1948
20000	0.08891
25000	0.04008
30000	0.01841
40000	0.003996
50000	0.001027
60000	0.0003097
70000	0.00008283
80000	0.00001846

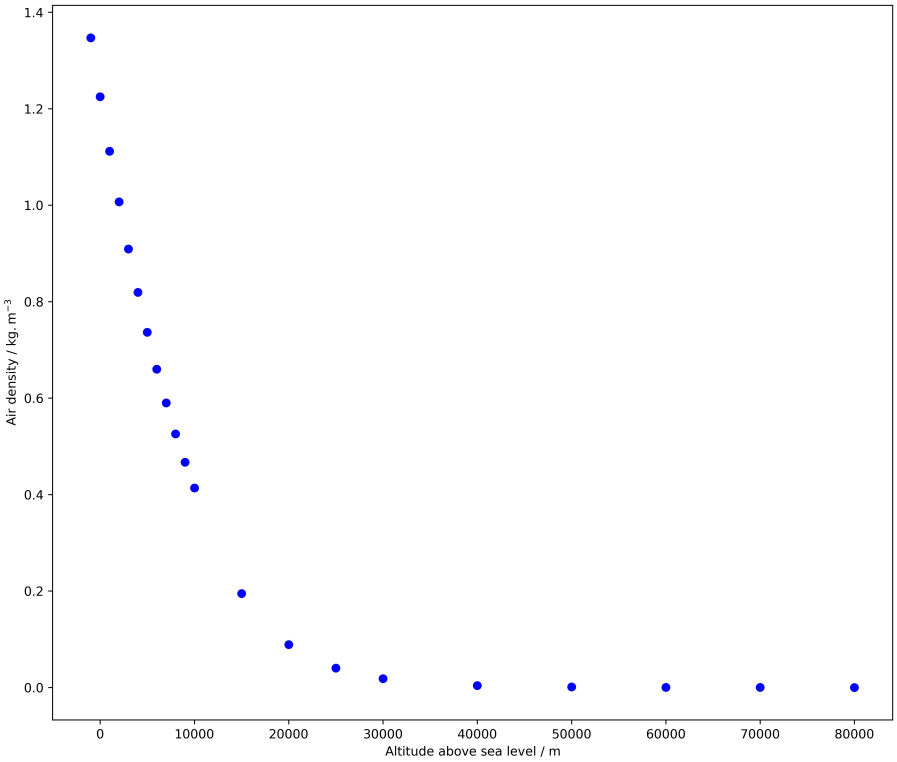

Figure 1: Air density vs altitude above sea level

$$f(y) = A \times \mathrm{e}^{B.y}$$

to obtain the A and B exponential function parameters.
The result exponential curve and the data can be seen in the Figure 2

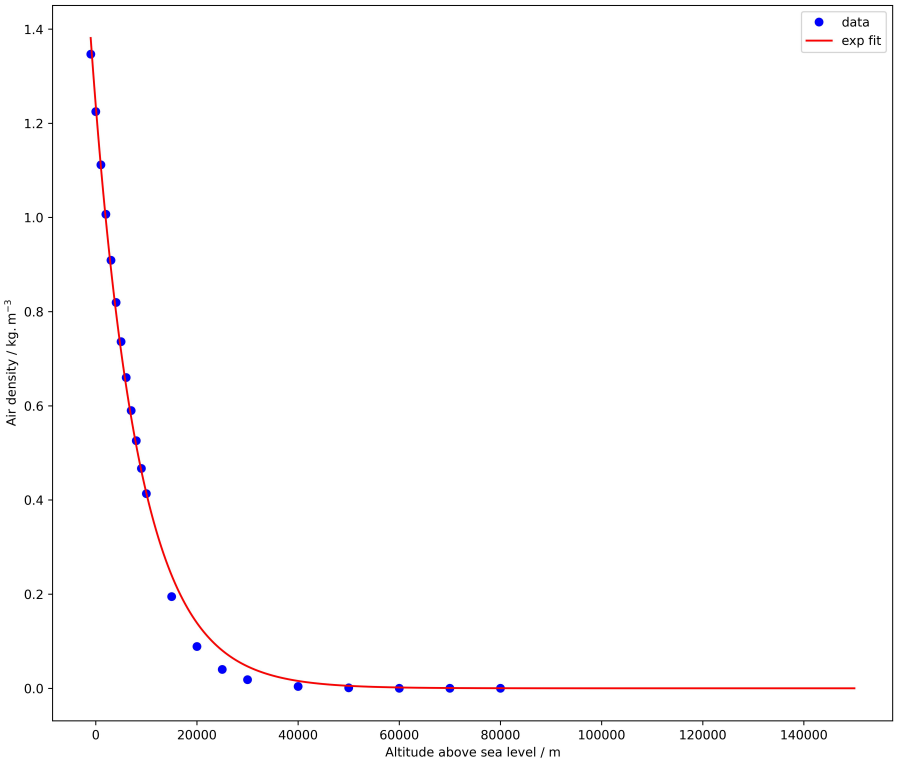

Figure 2: Air density vs altitude above sea level - data and exponential fit

For most of the range seems to give a good approximation except on the most non linear part of the data. On a side note, don't be satisfy with *seems*, some analysis of the difference between data and estimation needs to be done to know if it is a good fit for the intended propose or not.

The good side of this solution is it is always derivative defined and when altitude goes for infinite the air density goes to 0. But makes an overestimation of air density values for part of the range of interest.

2.1.2 Cubic spline

Another solution to be considered is to use a cubic spline. Like the linear interpolation it will compute several cubic equations, but will be done so it ensures the overall cubic piecewise will have defined first and second derivatives. It can also be used to extrapolate values outside the data range.

It is possible to use *interpolation.CubicSpline* of *scipy* to obtain the cubic spline function, even to extrapolate if that option is chosen. The function obtain is an object that has the method derivative, which allows to obtain the derivative function of the cubic spline, that it will be useful for the implicit method.

The function obtained will have the same exact values on the data, but depending on the data can introduce artifacts, and is always important to observe the data obtained in this way.

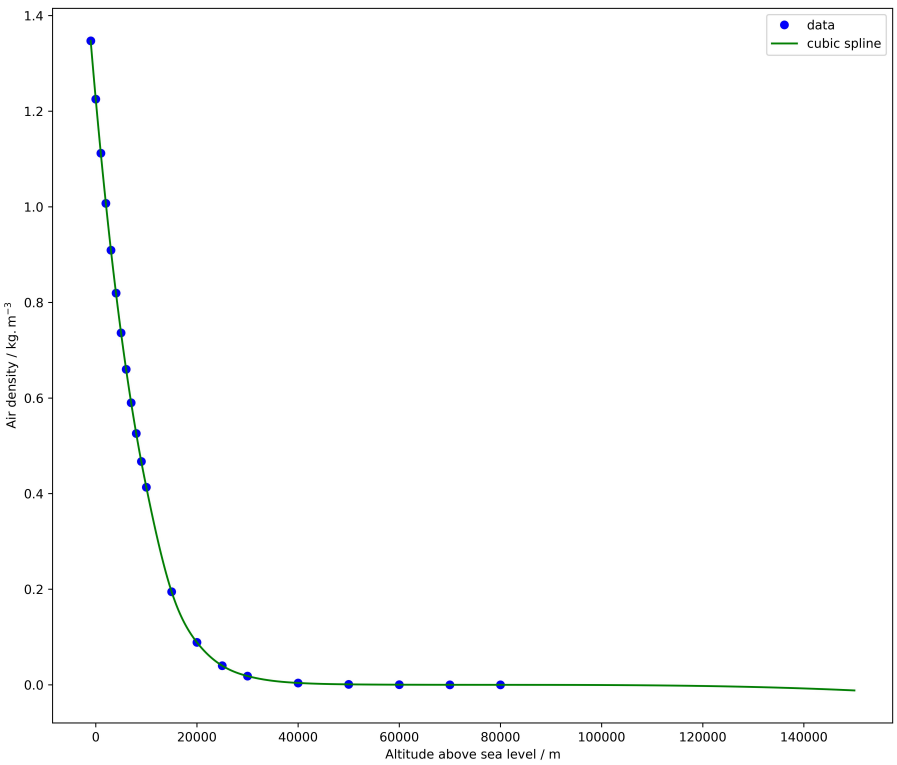

Figure 3: Air density vs altitude above sea level - data and cubic spline

Now the function obtained *seems* to follow the data more closely but it will give

negative air density values that is not possible.

A possible solution is to replace any negative output obtained by a zero value.

2.1.3 Optimal solution

Usually there is not an optimal solution, and it is up to the persons building the model to decide how to solve the missing data problems.

For this model and simulations both air density options showed obtained approximately the same results, but the exponential fit produces lower gs values than the cubic spline solution.

The students can also try to find more complete data or other source of computing air density with the altitude.

3 Simulations

The first objective of the project is using a forward method of the chose of the students to obtain the window of v_0 and α (downward angle with the horizon) initial values at the reentry interface that will allow a reentry within the define restrictions.

Careful that these reentry parameters are not independent one of the other, so the students need to obtain pair of valid values. Ideally would be to obtain a plot of one in function of the other and the approximate curves that bound valid reentry solutions. See Figure 2 on *Apollo 11 Reentry Report.pdf* also in clip. For our case we will only consider fewer restrictions.

The reentry interface is defined at 130 000 m of altitude and a successful reentry solution will be the ones that ensure the vessel and crew don't sustain more than 15g (150 $\mathrm{m.s^{-2}}$), the space module touches ocean surface at no more than 25 $\mathrm{m.s^{-1}}$ and the *horizontal* distance projected on the Earth surface will be between 2500 km and 4500 km.

The parachutes are deployed if the altitude is less or equal to 1 000 km and the speed is less or equal to 100 $\mathrm{m.s^{-1}}$. As previously stated, when the parachutes are deployed the lift component cease to affect the movement and the vessel is subject to the drag resistance of itself plus the one from the parachutes.

For the v_0 and α values it is recommended to explore values between 0 and 15 000 $\mathrm{m.s^{-1}}$ for v_0 and between 0^{o} and 15^{o} for α. These are very broad ranges and the set of valid values pair should be inside those ranges but being more restricted.

The space module model parameters are $m = 12000\,\mathrm{kg}$, $A = 4\pi\,\mathrm{m^2}$, drag coefficient $C_d = 1.2$, lift coefficient $C_l = 1.0$.

The parachutes total model parameters are $A_p = 301\,\mathrm{m^2}$ and $C_{d_p} = 1.0$.

3.1 g values

The g values are the total acceleration the vessel and crew are subjected to, less the gravity acceleration and usually measured in g, i.e. multiples of the gravity acceleration on the sea level. Don't forget that the gravity acceleration is only in one direction while the total acceleration can have components on the vertical, like gravity, and on horizontal direction. For this measurement the students can use for the standard gravity acceleration at sea level the value of $10\,\mathrm{m.s^{-2}}$.

3.2 Horizontal distance projected on the Earth surface

This value pretends to measure the distance from the point on the Earth surface vertically below the crossing the reentry interface by the space module to the point it touches down. We can not use the x values of the trajectory because that movement is compose of arc of circles at different heights from the Earth center, which would increase the values obtained.

An approximation would be to consider that at each step we consider as if each step in the forward method was taken at constant height (which is sometimes a rough approximation) and compute the θ angle measured from the center of the Earth that step movement had made. Considering it at constant height and equal to the initial height of the step, we would have:

$$\theta_i = \frac{x_i}{R_{Earth} + y_i}$$

The sum of all θ_i will give the total angle from interface reentry crossing to touchdown as measured from the center of the Earth, and now it is possible to compute the distance as measured at the Earth surface by:

$$distance = R_{Earth} \times \theta$$

3.3 Next project phase

After obtaining the information of good reentry parameters for obtaining reentry solutions, apply a backward method and run the simulation for good and not good solutions encountered with the forward method, comparing the results and time used in the computation.

In the backward method use the Newton method developed by the students in the group for obtaining the root of equation obtained. You can use the Backward Euler method or the Trapezoid method, that is a 2nd order method, but probably wouldn't be advisable at this stage try to implement higher order backward methods.

4 Deliverables

For this project each group must send a zip file containing:

- The code used for obtaining good reentry solutions using a forward method
- The code used using a backward method
- A group report on the work

All files must be named XXXXX-YYYYY(-ZZZZZ)-SMCEF-2024-P1-file_type, with:

- XXXXX, YYYYYY and eventually ZZZZZ the student numbers
- file_type as code-v0, code-v1, report

It does not represent the file extensions but please let them stay in the file names. The code files should be .py (not jupyter notebooks) and the report in pdf format.

The report should have a short introduction, about one page, reporting what is the objective of the simulation and what physical phenomena are we trying to simulate, with a concise description.

A section explaining the code structure, how it works (what information it needs and it's given to the code), and any decision or considerations you had to do implementing the code.

Another section with the reasoning used to chose the forward and backward methods, the meta-parameters of the methods chosen in the final version. Results of the simulations, remember that graphs are a very concise way to transmit information.

In the conclusions besides analyzing the results, there should also be a short information on what were the main difficulties of the project, and what could eventually be improved. You may include considerations on how to improve the model and simulations.

5 Grading

The project grading is divided in different categories, namely:

- Model implementation correctly done - 8 values
- Code readability and comments - 2 values
- Simulation code - 3 values
- Quality of final results - 2 values
- Report - 5 values divide as:

- Readability and presentation of results - 1 values
- Reasoning for the implementation decisions - 3 values
- Introduction and conclusions - 1 value

6 Project Deliver

The project should be sent by an email to ajw@fct.unl.pt, with subject "[SMCEF] P2 deliver of students (number of the students"), no later than 23:59 of July 8th.