## Introduction

Your bitcoin play has coming off massively, you are now a billionaire. Congratulations. You have decided that you would like to send horse to space. A world’s first. You have figured out the life support systems and all welfare issues. Everyone agrees this is a good idea and the horse will enjoy it. To start you have chosen a sub orbital vertical flight as you like the horse to be back for teatime, to give it sugar lumps and brush its hair.

You own a selection of horses but haven’t decided what horse you would like to send. Data relating to your shortlist of candidates are stored in horseyOptions.txt

|  |  |  |  |
| --- | --- | --- | --- |
|  | B.F. Rocket | Flubber | Elsie |
| Mass | 1000 kg | 500 kg | 75 kg |
| Length | 3 m | 2.5m | 2 m |
| Favourite Food | Hay | Hay | Hay |

You talk to a rocket manufacturer and they suggest the rocket most suitable for the comfort each horse. Data on these rockets are stored in horseyRocketOptions.txt

|  |  |  |  |
| --- | --- | --- | --- |
|  | B.F. Rocket | Flubber | Elsie |
| Horse Payload: | B.F. Horse | Derpy | Nozomi |
| Mass Empty: | 2,000 kg | 1,500kg | 1,500kg |
| Mass Fuel: | 10,000 kg | 8,000 kg | 7,000 kg |
| Thrust: | 170 kN | 150 kN | 130 kN |
| Burn Time | 40 seconds | 50 seconds | 60 seconds |
| Diameter | 3 m | 2.5 m | 2 m |

Starting with sample script simpleRocket.m provided to you for the vertical flight of a rocket improve it in the following ways.

1. Write a function which calculates the length of the flight (when rocket returns to 0 metres)
2. Adapt the code to end the simulation at this time.
3. Write a function (*maxHeight.m*) to calculate the max height reached by the rocket. Add to the code it outputs the value in command window.
4. Write a function to calculate velocity at impact (*kaboomSpeed.m*). Add to the code it outputs the value in command window.
5. Write a function (*findDensity.m*) which calculates density as a function of height above sea-level. To help use the data provided in the file (DenistyData.xlsx) obtained from https://ccmc.gsfc.nasa.gov/modelweb/models/nrlmsise00.php
6. Write a function (findDrag.m) which calculates the drag at each time step as a function of velocity and density.
7. Write a function (findWeight.m) which calculates the weight of the rocket at each time step. Assume to fuel mass decrease linearly from the launch time to the end of the burn.

\*\*Some of the slides shown in class have more help on creating the varying functions in 5-7\*\*

1. Update the sample code so it now uses the functions above to calculate the time varying value for drag and weight and that these values are used in the equation on motion.
2. Make the following plots for the Nozomi/Elsie horse/rocket combo:

* Accelerations vs Time
* Velocity vs Time
* Position vs Time
* Fuel mass vs Time
* Density vs Time
* Density vs Position
* Drag vs Time

1. Once you are happy with the changes to the code. Add the functionality to read in the data in from horseOptions.txt and rocketOptions.txt and make the same plots as in (9) comparing the 3 horses on one plot. (Each horse has its own rocket)

Bonus: You decide a parachute is a good idea for the health of the horse. You buy one in an army surplus shop. At an altitude of 10km (on the way down, hint: velocity negative) you deploy the parachute. It has the following properties.

Area = 10m2

Drag Coefficient = 1.75

For the 3 horses, plot how this effects the velocity over time.

You might find this link useful <https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/rktvrecv.html>