Trans-Lunar Injection Rocket Configuration Problem

Overview:
You have been asked to determine the ***best*** configuration for a single rocket that can perform the following tasks:

1. Achieve orbit by achieving a velocity of at least 7,800 m/s and an altitude of 200 km. From this point a final rocket stage will be used to place a payload of at least 25,000 kg into a Trans-Lunar Injection Trajectory defined by a terminal velocity of 10,900 m/s.
2. Achieve terminal velocity with a tolerance of +/-100 m/s.
3. Launch with a maximum dynamic pressure of no more than 33% of atmospheric pressure (33,440 Pa).
4. The maximum acceleration seen by the payload on the rocket should be no more than 8 g's (78.48 m/s^2).
5. In addition to achieving a feasible configuration, a design that minimizes the risk of failure is desired. To determine this, probability, the probability of failure of each engine component is provided in ME8710-RocketsFinal.xlsx.

A simulation written in C++ has been developed to allow you to evaluate your designs (RocketSim.exe). The source code is available to you to examine, but the file RocketSim.exe will be used to determine feasibility of your proposed design. The general operation of the program is described subsequently in the problem description. The simulation generates a set of telemetry data (Telemetry.txt). This file can be loaded into an excel file (Telemetry.xlsx) for viewing and analysis or can be processed to find critical data points and times using any number of software packages. In addition, key values from the simulation are documented in SimulationSummaryData.txt.

Your rocket configuration includes the following considerations:

- Number of stages (1-5)
- Number of boosters (0+)
- Type of booster (32 choices)
- Type of First Stage Engines (46 choices of Engine)
- Number of First Stage Engines (up to 50 Engines)
- Type of Second, Third, Fourth and Fifth Stage Engines (40 choices of Engine)
- Number of Second, Third Fourth and Fifth Stage Engines
- Throttle Profile of the First Stage (and Boosters)
- Burn Durations of the Second, Third, Fourth and Fifth Stages
- Diameter of the Main Rocket Core
- Mass of the Payload

From these parameters, the masses of the rocket stages and boosters are determined by the program.

Program Description:
RocketSim.exe relies on several files for proper operation. These files include:

- *Batch.txt* – Allows for batch runs of RocketSim.exe to be run. These runs will append results of each individual model to the end of the previous file.
- *log.txt* – Stores information about each run of RocketSim.exe. This file is rewritten each time RocketSim.exe is executed.
- *Rocket.txt* – Stores information about the configuration of each rocket proposed. These values can be pasted into the Variable Definition sheet within Telemetry.xlsx to show the meaning of each value. This file is rewritten each time RocketSim.exe is executed.
- *RocketData.txt* – This is a file which is read by RocketSim.exe which contains the data necessary to define the rocket based on the parameters entered in RocketSim.exe. The data is shown in a more readable format in ME8710-RocketsFinal.xlsx. DO NOT MODIFY THIS FILE. I have made it read-only to protect it.
- *Telemetry.txt* – This file stores the results of RocketSim.exe. The data within it can be copied into the Telemetry sheet in Telemetry.xlsx to analyze the performance of a design. Graphs of the resulting performance are provided in the Telemetry Plots sheet.
- *SimulationSummaryData.txt* – This file stores five key values from the simulation along with the time in the simulation where they occur. These values are the altitude of the rocket when it should be in orbit (km), the velocity in orbit (m/s), the velocity for translunar injection (m/s), the maximum acceleration of the rocket (m/s^2) and the maximum dynamic pressure (Pa).

The code files for RocketSim.exe are located in a subdirectory (SourceCode) and include the following files:

- **Cexcept.h, Cmatrix.h and CVector.h** – These are class files used to define the data structures used in the program. Basically, all you may really need to know is the dVector<TYPE> defines a vector of type TYPE, and dMatrix<TYPE> defines a matrix of type TYPE.
- **Constants.h** – this header file defines constants needed throughout the program and defines necessary header files that are needed to define functionality in the program.
- **FlightModels.h** and **FlightModels.cpp** – these files are the header (.h) and source (.cpp) files that define several relevant functions for the simulation of the rocket flight. These models include:
    - Density() which calculates the atmospheric pressure (Pa) , temperature (K) and density (kg/m^3) as a function of altitude;
    - WaveDrag() which calculates the effective drag coefficient of the rocket as a function of altitude and velocity and incorporates approximations for the effects of transiting the sound barrier;
    - Thrust() determines the thrust (N) produced by the rocket at a time, t;

   - Throttle() determines the throttle settings (%) of the rocket based on the user defined throttle profile;
   - GrossVehicleMass() uses the data provided by the user to define the rocket configuration to determine the GrossVehicleMass at Launch (kg);
   - Fdrag() calculates the Drag Force (N) felt by the rocket;
   - Gravity() calculates the effective acceleration due to gravity (m/s^2) as a function of altitude;
   - MassChange() calculates the change in mass of the rocket during a simulation time period due to both fuel consumption and separation of boosters and/or stages during flight.
- **Menu.h** and **Menu.cpp** – these files define the Menus used by the user to enter data into the program. These files are not used in Batch file mode.
- **RocketSim.h** and **RocketSim.cpp** – these are the main files of the program and they contain the simulation loop that simulates the flight of the rocket.

Boosters are managed independently of the rest of the rocket and all their parameters are predefined. The boosters are the elements shown in Figure 1.

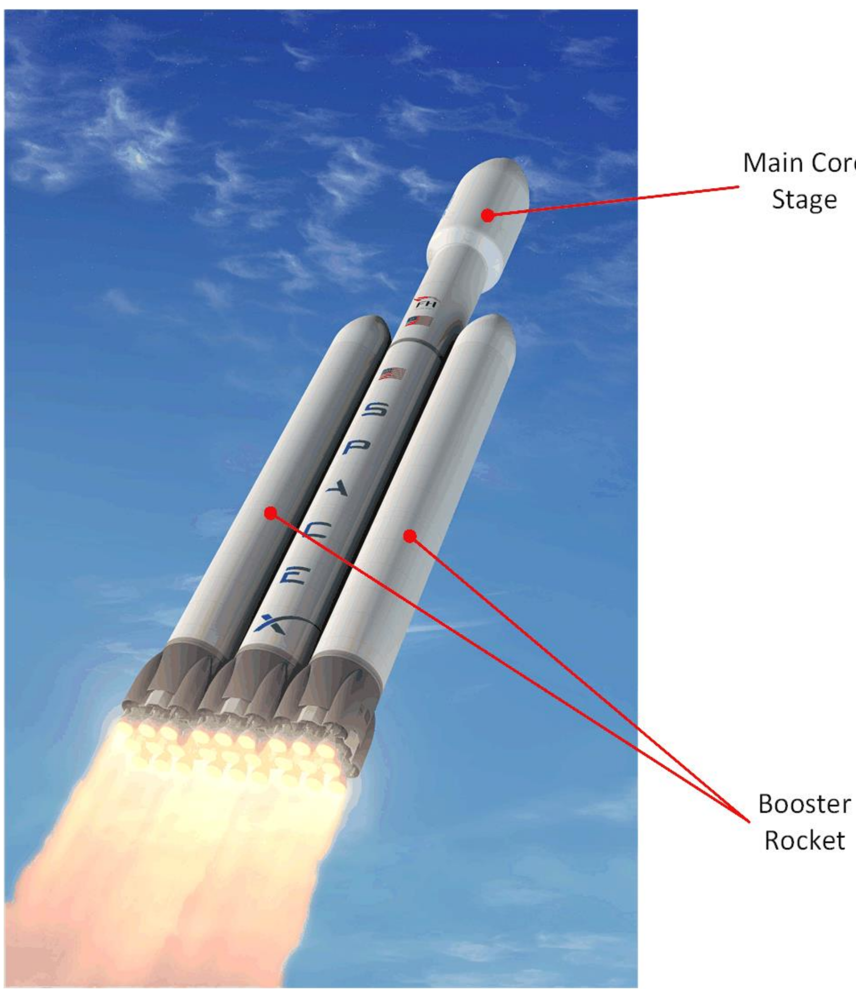


Figure 1. Booster versus Core Stage.


Figure 2. SpaceX Falcon9 Rocket Launching

Example Program:
Let's do an example based on the SpaceX Falcon9 rocket, shown in Figure 2. This rocket is a two-stage rocket with 9 Merlin 1D engines on the first stage and a single Merlin 1D-Vacuum engine on the second stage. I've approximated the throttle settings and stage durations to get something that works. The model is not perfect, but it does produce results that are good enough for what we are trying to accomplish in this class.

Here is what we would enter in response to each menu entry.

Enter the number of stages in the Rocket (>0 and <=5): 2

Enter the number of boosters attached to the first stage of the Rocket (>=0): 0

Select the type of Rocket Engine for Stage 1 from the list:
144) SpaceX Merlin 1D

Enter the number of Rocket Engines for Stage 1 of the Rocket (>=0 and <= 50): 9

Select the type of Rocket Engine for Stage 2 from the list:
238) SpaceX Merlin 1DV

Enter the number of Rocket Engines for Stage 2 of the Rocket (>=0 and <= 50): 1

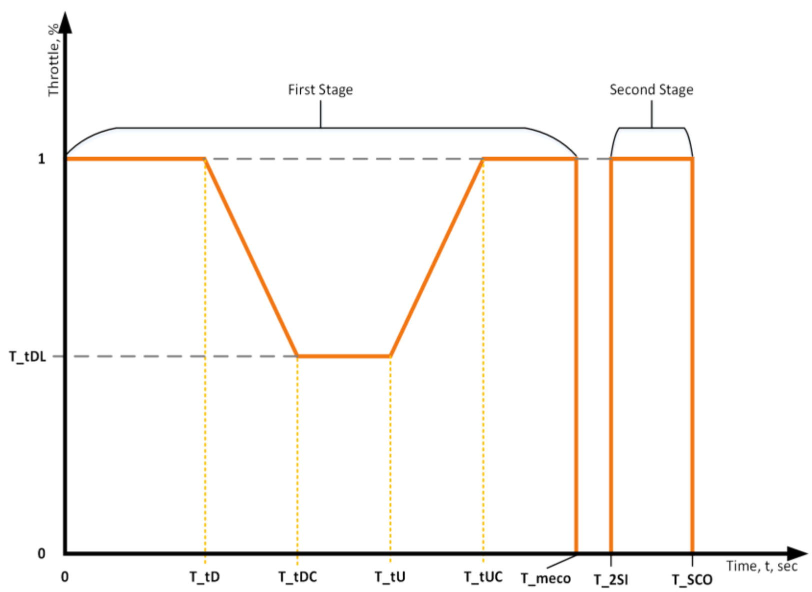


Figure 3. Throttle Profile Diagram.

At this point, the menus now will ask for the throttle profile for the rocket. This is important as the throttle profile allows us to reduce the dynamic pressure experienced by the rocket. The maximum dynamic pressure level is known as MaxQ, and represents the maximum load placed on a rocket during flight. Typically, rockets can be designed for a maximum dynamic pressure of about 1/3rd of an atmosphere. All first stages (and the boosters) will obey the throttle profile. The profile looks like the following diagram shown in Figure 3.

You do not have full freedom to define the profile, and the booster rockets if any have predefined burn durations. Thus, boosters have the profile applied to them, but may burnout (and be discarded) during the First Stage thrust profile. Boosters will however throttle down just like the main engines. Note that all times are given as integer seconds. Note also that while the menus will not allow illegal times, you can define illegal times (i.e. T_meco before T_tUC) – please don't do that.

Enter the Throttle Down Time, T_tD where T_tD > 0 sec: 60

Enter the Throttle Down Complete Time, T_tDC where T_tDC > 60 sec: 70

Enter the Throttle Down Level, T_tDL where 0.2 <= T_tDL < 1.0: 0.8

Enter the Throttle Up Time, T_tU where T_tU > 70 sec: 90

Enter the Throttle Up Complete time, T_tUC where T_tUC > 90 sec: 100

Enter the MECO time (Main Engine Cut Off), T_meco where T_meco > 100 sec: 240

Enter the Stage 2 Ignition time (2SI), T_2SI > 241 sec: 245

Enter the Stage 2 Cut-Off time (2SCO), T_2SCO > 246 sec: 485

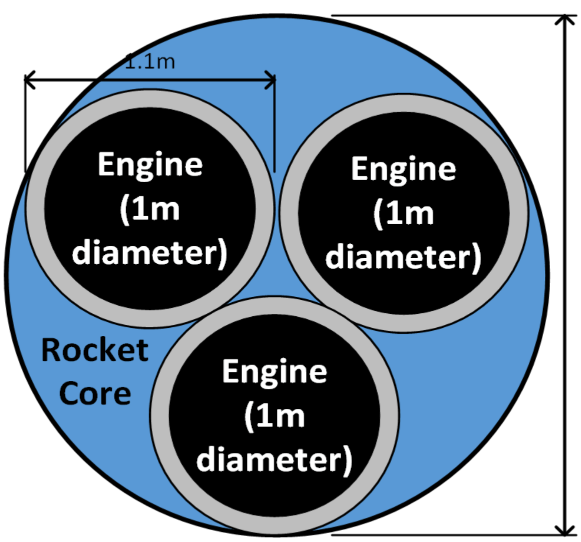


Figure 4. Rocket Core Diameter.

Only a few more parameters need to be defined through the menus. The next question is the size of the main stage. (This is assumed to be constant for the entire length of the main rocket.) Boosters are accommodated separately. You will need to determine this value externally, based on how you configure the engines. Every engine has a diameter provided. Engines can be packed so that a circle of 110% of that diameter may touch. The main stage diameter then is the circle diameter necessary to enclose all those engines (in meters). See Figure 4 for an example using 3 rocket engines. Each engine is 1m in diameter, and by arranging them in a triangle, I get a minimum diameter of the main core of the rocket will be about 2.55m (or larger). You are responsible for determining this value for all stages that you may use. In addition to the core diameter, you need to define the payload mass at this time.

Determine the Main Stage Diameter which should be sufficient to account for the diameters of all rocket nozzles used on the main stage. Booster Engines are not included in this calculation. Enter the Diameter in meters of the main stage: 3

Specify the mass of the payload to be launched in kg: 1000

The remainder of the calculations for the rocket are done automatically. The result is a rocket defined by the following parameters as seen in Figure 5.

Position	RocketDesign (Rocket.txt)	From Rocket.txt
0	Number of Stages	2
1	Number of Boosters	0
2	Type of Boosters	0
3	1st Stage Exists (0 = "No", 1 = "Yes")	1
4	1st Stage Engine Type	144
5	Number of 1st Stage Engines	9
6	2nd Stage Exists (0 = "No", 1 = "Yes")	1
7	2nd Stage Engine Type	238
8	Number of 2nd Stage Engines	1
9	3rd Stage Exists (0 = "No", 1 = "Yes")	0
10	3rd Stage Engine Type	0
11	Number of 3rd Stage Engines	0
12	4th Stage Exists (0 = "No", 1 = "Yes")	0
13	4th Stage Engine Type	0
14	Number of 4th Stage Engines	0
15	5th Stage Exists (0 = "No", 1 = "Yes")	0
16	5th Stage Engine Type	0
17	Number of 5th Stage Engines	0
18	Main Engine Ignition @ T=0 (FIXED = 0.0)	0
19	Main Engine Throttle @ T=0 (FIXED = 1.0)	1
20	Main Engine Throttle Down time (T_tD)	60
21	Main Engine Throttle @ T=T_tD (FIXED = 1.0)	1
22	Main Engine Throttle Down Complete time (T_tDC)	70
23	Main Engine Throttle @ T=T_tDC	0.8
24	Main Engine Throttle Up time (T_tU)	90
25	Main Engine Throttle @ T=T_tU (FIXED = Throttle @ T=t_DC)	0.8
26	Main Engine Throttle Up Complete time (T_tUC)	100
27	Main Engine Throttle @ T=T_tUC (FIXED = 1.0)	1
28	Main Engine Cut Off - MECO - time (T_meco)	240
29	Main Engine Throttle @ T=T_meco (FIXED = 0.0)	0
30	Second Stage Igntion @T>=T_meco+1 (T_2SI)	245
31	Second Stage Engine Throttle @ T=T_2SI (FIXED = 1.0 if present)	1
32	Second Stage Cut Off @T>=T_2SI+1 (T_2ECO)	485
33	Second Stage Engine Throttle @ T=T_2ECO (FIXED = 0.0)	0
34	Third Stage Igntion @T>=T_2ECO+1 (T_3SI)	486
35	Third Stage Engine Throttle @ T=T_3SI (FIXED = 1.0 if present)	0
36	Third Stage Cut Off @T>=T_3SI+1 (T_3ECO)	487
37	Third Stage Engine Throttle @ T=T_3ECO (FIXED = 0.0)	0
38	Fourth Stage Igntion @T>=T_3ECO+1 (T_4SI)	488
39	Fourth Stage Engine Throttle @ T=T_4SI (FIXED = 1.0 if present)	0
40	Fourth Stage Cut Off @T>=T_4SI+1 (T_4ECO)	489
41	Fourth Stage Engine Throttle @ T=T_4ECO (FIXED = 0.0)	0
42	Fifth Stage Igntion @T>=T_4ECO+1 (T_5SI)	490
43	Fifth Stage Engine Throttle @ T=T_5SI (FIXED = 1.0 if present)	0
44	Fifth Stage Cut Off @T>=T_5SI+1 (T_5ECO)	491
45	Fifth Stage Engine Throttle @ T=T_5ECO (FIXED = 0.0)	0
46	Diameter of Main Stage (m)	3
47	Payload (kg)	1000
48	Booster Wet Mass (kg)	0
49	Booster Dry Mass (kg)	0
50	First Stage Wet Mass (kg)	6.34E+05
51	First Stage Dry Mass (kg)	50899
52	Second Stage Wet Mass (kg)	74860.4
53	Second Stage Dry Mass (kg)	5980.4
54	Third Stage Wet Mass (kg)	0
55	Third Stage Dry Mass (kg)	0
56	Fourth Stage Wet Mass (kg)	0
57	Fourth Stage Dry Mass (kg)	0
58	Fifth Stage Wet Mass (kg)	0
59	Fifth Stage Dry Mass (kg)	0
60	Booster Seperation Time (sec)	0
61	First Stage Seperation Time (sec)	240.5
62	Second Stage Seperation Time (sec)	485.5
63	Third Stage Seperation Time (sec)	0
64	Fourth Stage Seperation Time (sec)	0
65	Fifth Stage Seperation Time (sec)	0

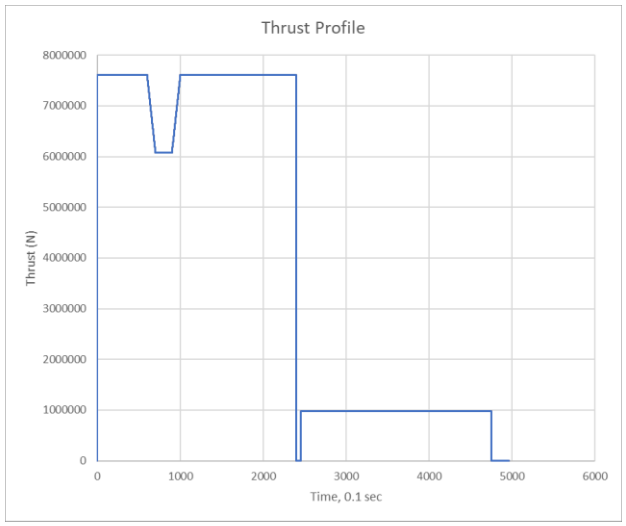
Automatically entered for stages 3-5

Automatically found for stages 3-5

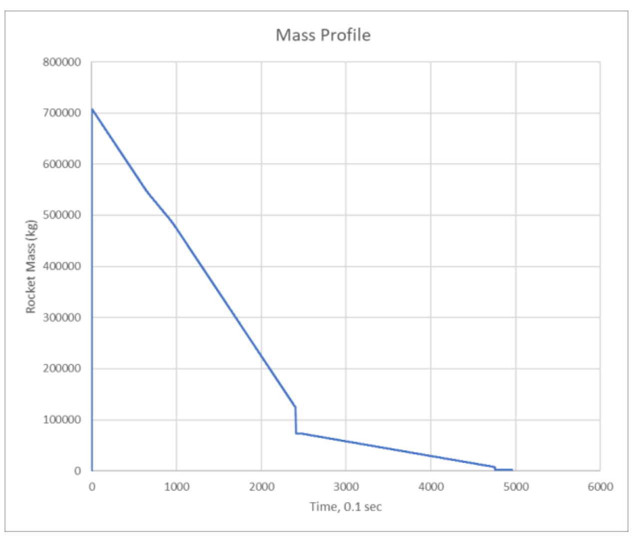
Calculated automatically by the simulation using RocketData.txt

Figure 5. Rocket Definition Vector.

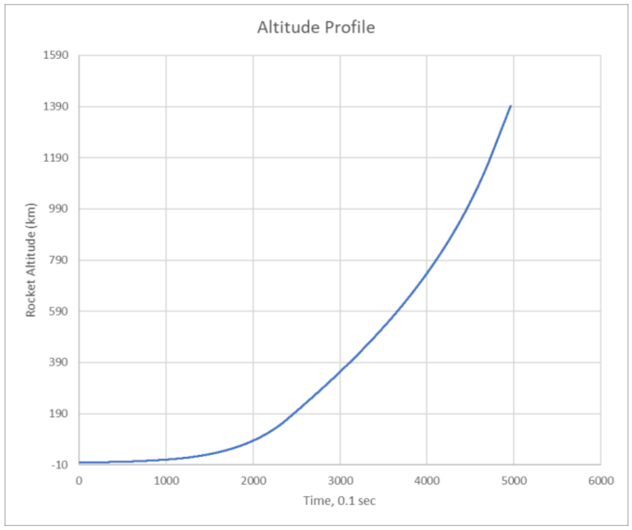
Example Results:
The results of the simulation can be seen by loading the data from Telemetry.txt into the Excel File Telemetry.xlsx. The file automatically generates a number of graphs. Let's take a look at some of the results. (Note, the graphs shown here are for a 1700kg payload Falcon 9 with a slightly different thrust profile.)



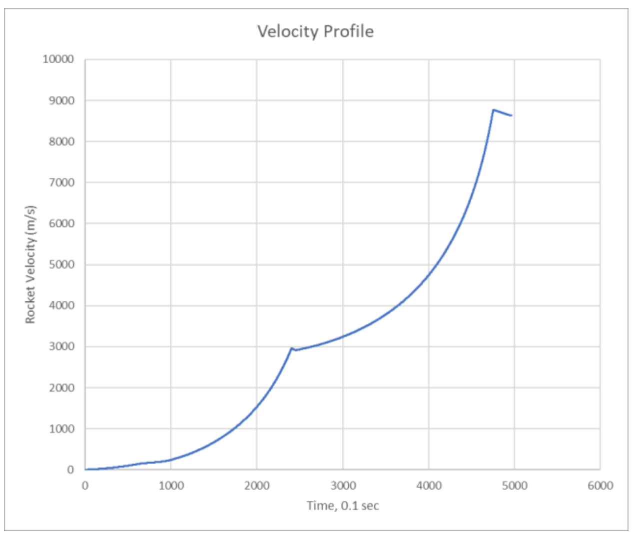
The Thrust Profile looks pretty much as one might expect from out previous discussions. The first stage with its nine engines puts out a lot more thrust than the upper stage, although the upper stage is vacuum optimized.



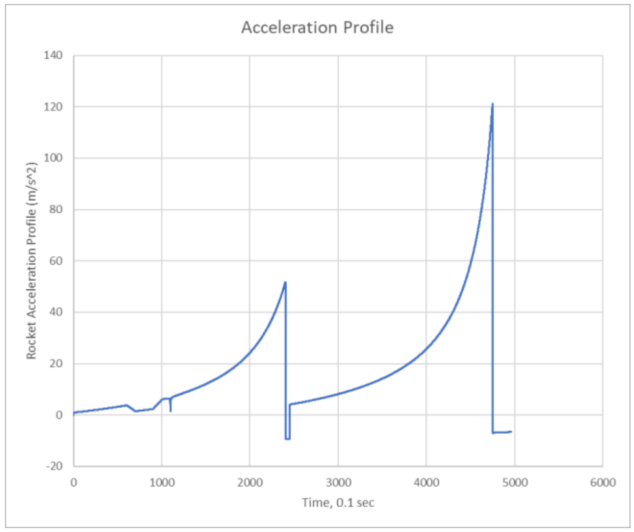
The mass profile is similarly as expected, with a rapid decrease in mass as the rocket burns through the first stage, a jump when the first stage separates, and then a slower rate of decrease for the second stage, followed by another jump as that stage is discarded.



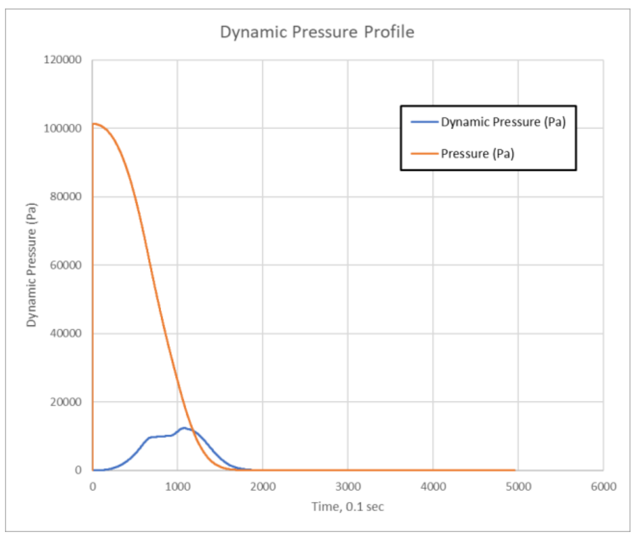
The altitude profile shows that the rocket reaches 1390 km – much more than the real thing, but in this simulation, I do not model the roll of the rocket that is used to boost orbital velocity. (Next version!)



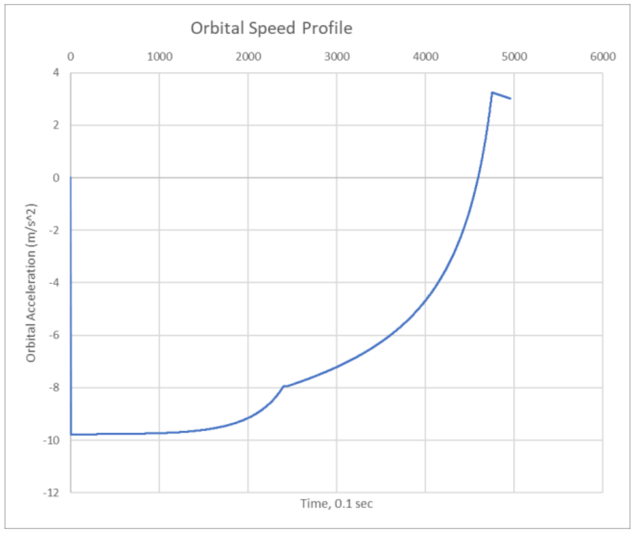
Similarly, you see there is a break in the velocity curve between T_meco and T_2SI. And again, you see the same effect after T_2CO. This is a partial artifact of the simulation model used.



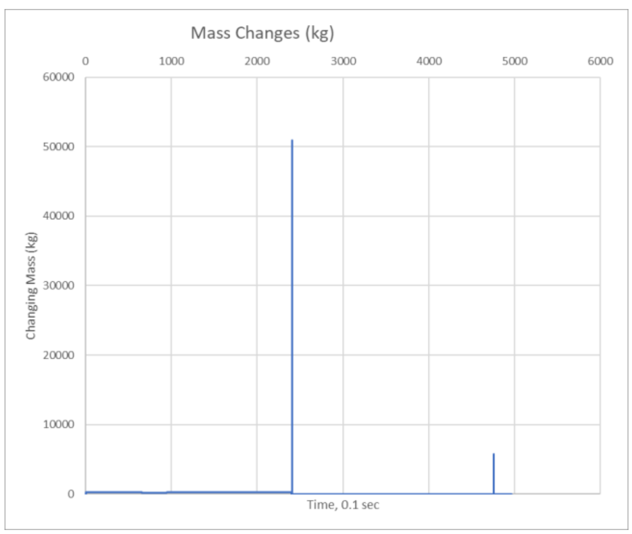
The acceleration profile also shows the stage cut-offs, where the rocket is briefly dominated by gravity. There is also a small blip just after 100 seconds (1000 on the graph) which corresponds to MaxQ – I did not quite time the throttle down to hit MaxQ in this trial. You can see where I throttled down on the graph as well.



You can see MaxQ as the peak of the blue line on this graph. Note how quickly the atmospheric pressure drops as the rocket climbs. MaxQ is determined by the velocity of the rocket and the atmospheric pressure as it climbs. This example has MaxQ well below the threshold.



The orbital speed (or orbital acceleration balance) profile needs to go positive for you to claim that the rocket will reach a stable orbit. This comes from a simplification of the simulation so as to not include the pitch over of the rocket, whereby some of the climbing acceleration goes to a tangential acceleration. Ideally, this value would be close to zero.



The final plot shown here clearly shows where we are discarding the stages of the rocket (the two big spikes. The empty (dry) mass of the stages is estimated from the required fuel needed by the stage and the known engine masses.

So, what do I need to do?
There are three optimization problems in this exam that you need to formulate and solve. They include the following:

1) Determine the minimum diameter of the core of the rocket so as to be able to arrange the rocket engines (for each stage) within this diameter. (**Hint:** *This problem can be formulated similarly to a trim loss problem – or with other approaches*.)
2) Determine the optimal first stage thrust profile so as to allow the rocket to pass through MaxQ without exceeding the dynamic pressure limits.
3) Determine the best configuration of the rocket to achieve the flight objectives of the problem while maximizing the chance of a successful flight. You may assume that the probability of a successful flight is defined by the following equation.

$$P_{success} = \prod_{i=0}^{n_{engines}} \left(1 - P_{failure,i}\right)$$

Submission:
So, what do you turn in? I want a brief writeup – like you would submit to your boss that provides your answer to the problem, as well as an explanation of your approach and justification for your results. Show me that as a result of this course, you have gained an understanding of how to perform an optimization study, select appropriate tools and draw appropriate conclusions from our results. To aid in confirming your results, please submit copies of the following files with the data for you proposed rocket for my analysis.

1) Your writeup as a PDF file. I do not want a massive document – just what is necessary to provide your answers, approach and justification.
2) The rocket.txt file for your proposed rocket.
3) The Telemetry.txt file for your proposed rocket.
4) The signed cover page of the exam with your acknowledgement of the rules and that this is your own work.

I will confirm the analysis by running your design against the master code that I am supplying you to verify that the design is feasible.

Final Comments:
If you find a bug in the program – please contact me ASAP and provide copies of the relevant files so that I can recreate the error. If necessary, I may issue a revised code. While I have tried to preclude user errors in the software, I am confident that you are clever enough to break the code if you try. Note that some rocket configurations are not feasible. Also, you may not add hardware (types of rocket engines) to the simulation. You are stuck with the hardware you have. Similarly, the properties of the launch site are programmed into the simulation. I will be releasing a video of the software in action for you to see how I have been using it.

Good Luck and remember – this exam is your own ***individual*** work.