2CM4 Design Project

Owen Bruce

Contents

1. Research

I am going to design a rocket which would be able to transport a load from the surface of the moon to a height of 3000 km and minimize the amount of fuel required by optimizing the design shape of the thruster and the duration of the burn. The amount of fuel needs to be minimized because all fuel either needs to be transported to or produced on the moon which are both expensive.

Minimizing the amount of fuel required for any maneuver in space is important as the fuel currently must be transported from the earth into space to be used by the rocket. Currently, the cheapest way to transport anything to orbit per kilogram is through SpaceX on their Falcon Heavy. The Falcon Heavy is able to transport 63,800 kg to Low Earth orbit for a price of $90M resulting in a per kilogram cost of approximately $1,400/kg. [1] A goal then for this optimization is to attempt to get minimum amount of fuel required to bring the rocket to 3000 km above the surface of the moon to below the 63.800 kg that the Falcon Heavy is able to transport.

To fulfill the goal of this design, the rocket needs to reach a height around the orbit of the moon. The lunar gateway which NASA is planning to orbit the moon will be orbiting in a near-rectilinear halo orbit, NRHO. This orbit means that the periapsis, the closest point in the orbit, will be around 3000 km away from the surface. [2] The goal of this rocket will then be to reach a height of 3000 km.

As nothing for NASA's upcoming Artemis mission has been completed yet, I am going to design the rocket to be able to hold and transport the lunar lander from the Apollo 11 mission. This means that the rocket must be able to transport an object which has a mass of 15,103 kg and is 22 ft. 11 in. (approx. 7m) tall by 14 ft. 1 in. (approx. 4.3m) wide. [3], [4] To me, this mass seems reasonable as being similar to the mass and size required to transport either humans or equipment down and up from the lunar surface for the Artemis mission, so I have gone with this as the payload mass.

Due to complications with orbital mechanics, I am simply going to be optimizing for a rocket which is able to reach an elevation of 3000 km above the surface of the moon. This problem could be extended to optimize a burn which would result in the rocket being placed into an orbit of 3000 km above the surface of the moon. However, I am not going to deal with this problem as it would require much greater knowledge of orbital mechanics and the error obtained during the simulation and optimization at the level I am capable of would likely be too high to consider the problem correctly optimized.

2. Physics Background

In order to optimize the rocket, we need to take into account all of the forces acting on the rocket throughout its trip. The only external forces acting on the rocket when it is in the atmosphere are the thrust from the rocket engine, gravity from the moon, and air resistance.

a. Thrust

The first force to consider is the thrust from the rockets. The thrust given by the rockets depends on the mass flow rate of the fuel through the throat of the nozzle, $\dot{m}$, the exit velocity of the fuel,V_e, the exit pressure p_e, the free stream pressure at the nozzle exit p_0, and the area ratio from the throat of the nozzle to the exit of the nozzle A_e. The throat of the nozzle is the smallest part of the nozzle. Figure 1. from NASA shows a simplified diagram of a rocket nozzle design. [5]

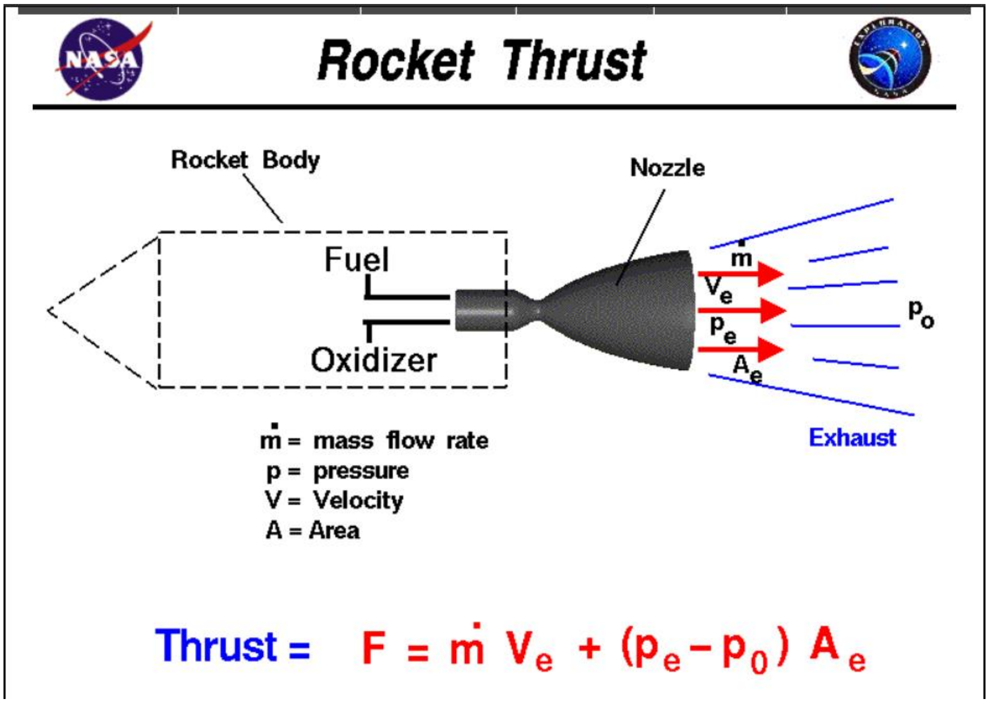

Figure 1. Nozzle diagram.

The equation needed to calculate the force is the following:

$$F_{Thrust} = \dot{m}V_e + (p_e - p_0)A_e$$

This formula comes from the two different interactions happening within the nozzle of the rocket. The first part of the formula, $\dot{m}V_e$, comes directly from Newton's Third Law. As the fuel is forced out of the rocket, the fuel exerts a force back onto the rocket, pushing it forwards. The second part of the formula, $(p_e - p_0)A_e$, causes thrust through the pressure difference between the exit of the nozzle and the outside air. When the pressure inside of the nozzle is greater than the pressure outside of the nozzle, the fuel gets pulled out of the nozzle by the pressure difference creating a force which propels the rocket forwards.

Next, we need to find the formula for all of the parameters which go into the thrust formula. This all depends on the following parameters: the total pressure which is the pressure in the combustion chamber, p_t, the total temperature in the combustion chamber, T_t, the specific heat ratio of the exhaust, γ, the Gas constant, R, the Mach number at the exit, M_e, the area of the throat, A*, and the area of the exit, A_e. [6] The following equations allow us to get the values for the thrust force equation:

$$\text{Mass Flow Rate: } \dot{m} = \frac{A^* p_t}{\sqrt{T_t}} \sqrt{\frac{\gamma}{R}} \left(\frac{\gamma+1}{2}\right)^{-\frac{\gamma+1}{2(\gamma-1)}}$$

$$\text{Exit Mach: } \frac{A_e}{A^*} = \left(\frac{\gamma+1}{2}\right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{\left(1+\frac{\gamma-1}{2} M_e^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}}}{M_e}$$

$$\text{Exit Temperature: } \frac{T_e}{T_t} = \left(1 + \frac{\gamma-1}{2} M_e^2\right)^{-1}$$

$$\text{Exit Pressure: } \frac{p_e}{p_t} = \left(1 + \frac{\gamma-1}{2} M_e^2\right)^{-\frac{\gamma}{\gamma-1}}$$

$$\text{Exit Velocity: } V_e = M_e \sqrt{\gamma R T_e}$$

All of the above equations rely on the Mach number being 1 at the throat of the nozzle. This is what gives many of the constant proportions that are seen in the above equations to calculate the required parameters with this constraint. Combining all of these equations, we can solve for the force produced by the thrust.

i. Mass Flow Rate

The equation for the mass flow rate come from the amount of fuel which is able to pass through the throat of the nozzle. This can be seen as it is proportional to the area of the throat which allows for more air to escape and proportional to the pressure at the throat which forces more air out.

ii. Exit Mach Number

The equation for the exit Mach comes from the relationship between the area at the exit of the nozzle and the area at the throat of the nozzle. Since the Mach number must be 1 at the throat of the nozzle, the Mach number at the exit of the nozzle only depends on the ratio of the two sizes and the properties of the gas within the nozzle.

iii. Exit Temperature and Pressure

Both the exit temperature and the exit pressure are inversely proportional to the Mach number which means that they are both also inversely proportional to the ratio of the exit area vs the throat area. This makes qualitative sense as the wider the nozzle gets, the lower the pressure and temperature become, but if the nozzle stays the same width, then pressure and temperature will not change much between the throat and the nozzle.

iv. Exit Velocity

Finally, the exit velocity is proportional to the Mach number at the exit of the nozzle and the square root of the temperature of the gas at the exit. Since the exit velocity is proportional to the Mach number, this means that the exit velocity of the gas will increase as the nozzle gets wider at the exit. This creates a balance between the two parts of the thrust equation. As the first part of the equation gets larger due to a greater ratio between the throat and exit areas, the second part of the equation gets smaller due to the ratio between the throat and exit areas.

b. Gravity

Next, we need to find the force produced from gravity on the rocket. As the moon can be approximated as a sphere, Newton's equation for gravity does not need to be adapted to account for the fact that the moon is a spatial object.[7] Newton's equation for gravity is the following:

$$F_g = G\frac{m_1 m_2}{r^2}$$

c. Air Resistance

Finally, there is the force from the air resistance on the moon; however, as the molecular density of the earth's atmosphere is approximately 10^{13} times that of the moon, the air resistance is negligible and can be fully ignored.[8] This leads us with only two forces, the thrust force and the gravitational force to balance in order to get our rocket into orbit.

3. FEM Incorporation

For the first test case, I am going to take the parameters from the space shuttle. The pressure inside the pressure chamber is 226 bar, the value of Ae/A* is 45, the temperature within the combustion chamber is 3,315.6°C, the total mass of the space shuttle with the external tank is 104,535 kg, and the total mass of the fuel carried is 729,465 kg. [9]–[11] For the constants, the Rydberg constant is 8.314, the gravitational constant is 6.67×10^{-11}, gamma is unknown for the moon's atmosphere, so 1.4 will be used as the value for air on the earth, the mean radius of the moon is 1737.4 km, the mass of the moon is 7.346×10^{22} kg, the average temperature of the moon is 243K, and the atmospheric pressure on the moon is 3×10^{-15} bar. [12]–[15]

Most of the equations can be put directly into FlexPDE in order to simulate the rocket; however, we can solve for the Mach number using Maple analytically and then using the solve function to find the answer. Attempting to solve for M_e in Maple, we get the following:

0.01286135869, 5.768384710, 1.808303718 + 6.407425544*I, -4.698926753 + 3.991489277*I, -4.698926753 - 3.991489277*I, 1.808303718 - 6.407425544*I

The correct Mach Number for the rocket is the second number in the list, as can be verified by consulting a solver on NASA's website which gives the same answer when a ratio of 45 is entered for Ae/A* as can be seen in Figure 2. below.

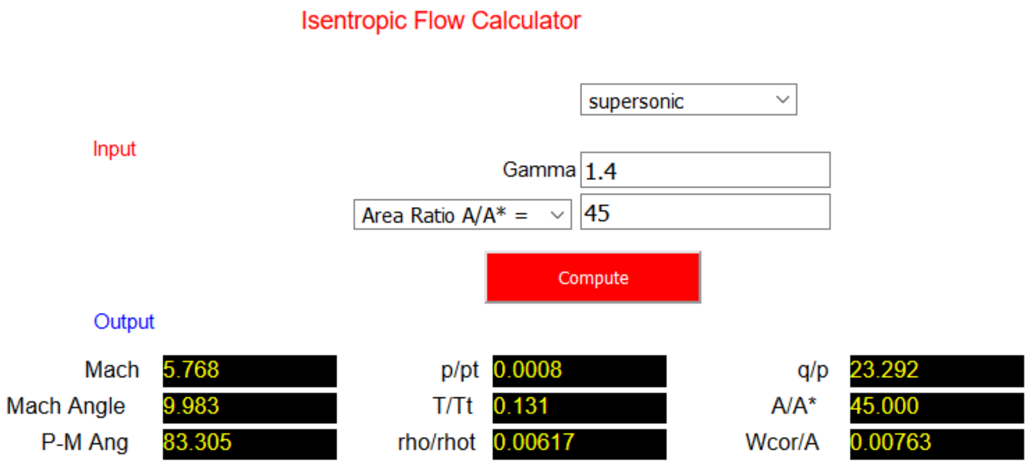

Figure 2.

When running this in FlexPDE, the rocket is able to reach the desired altitude of 3000 km in 4.7842s. This shows that a model of the space shuttle's engine would meet the design requirements; however, it is very fuel inefficient, resulting in 406614.0 kg of fuel being consumed. In terms of the goal of the project, this amount of fuel would require 7 launches of the Falcon Heavy from earth, solely to bring the fuel consumed by this rocket in its takeoff from the moon. Additionally, the amount of time that the rocket takes to reach the 3000 km is around 4.5 s. This means that the rocket is travelling at an average speed of 667 km/s. This rocket can obviously be more efficient as the amount of fuel consumed during the burn to 3000 km isn't even the total amount of fuel onboard the ship. Based on this, it is obvious that this problem can be heavily optimized by modifying the parameters to reduce the amount of fuel required. The trajectory of the rocket vs. time and thrust vs. time can be seen in Figure 3. and 4.

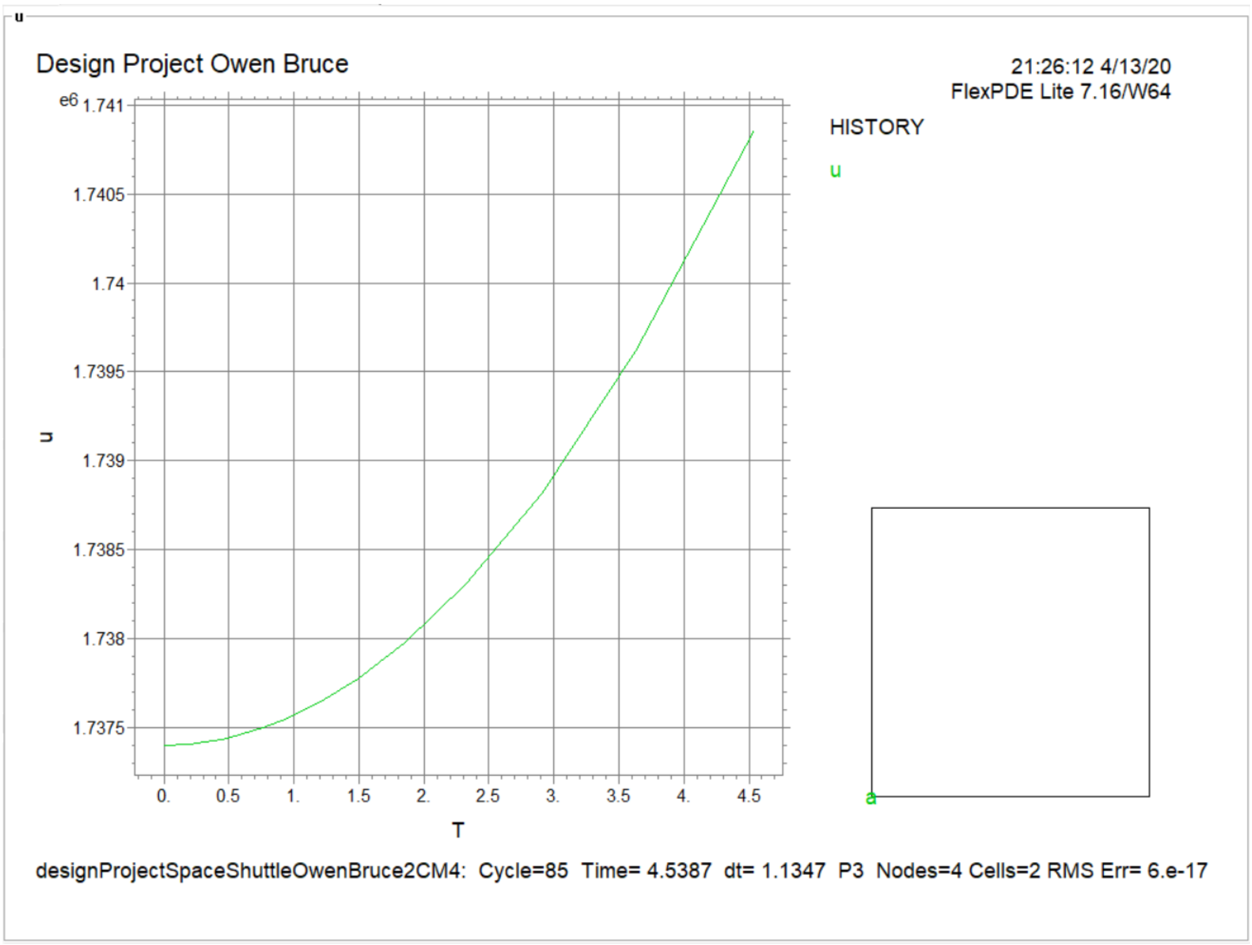
u
Design Project Owen Bruce
21:26:12 4/13/20
FlexPDE Lite 7.16/W64
HISTORY
u
e6 1.741
1.7405
1.74
1.7395
1.739
1.7385
1.738
1.7375
u
0.
0.5
1.
1.5
2.
2.5
3.
3.5
4.
4.5
T
a
designProjectSpaceShuttleOwenBruce2CM4: Cycle=85 Time= 4.5387 dt= 1.1347 P3 Nodes=4 Cells=2 RMS Err= 6.e-17

Figure 3.

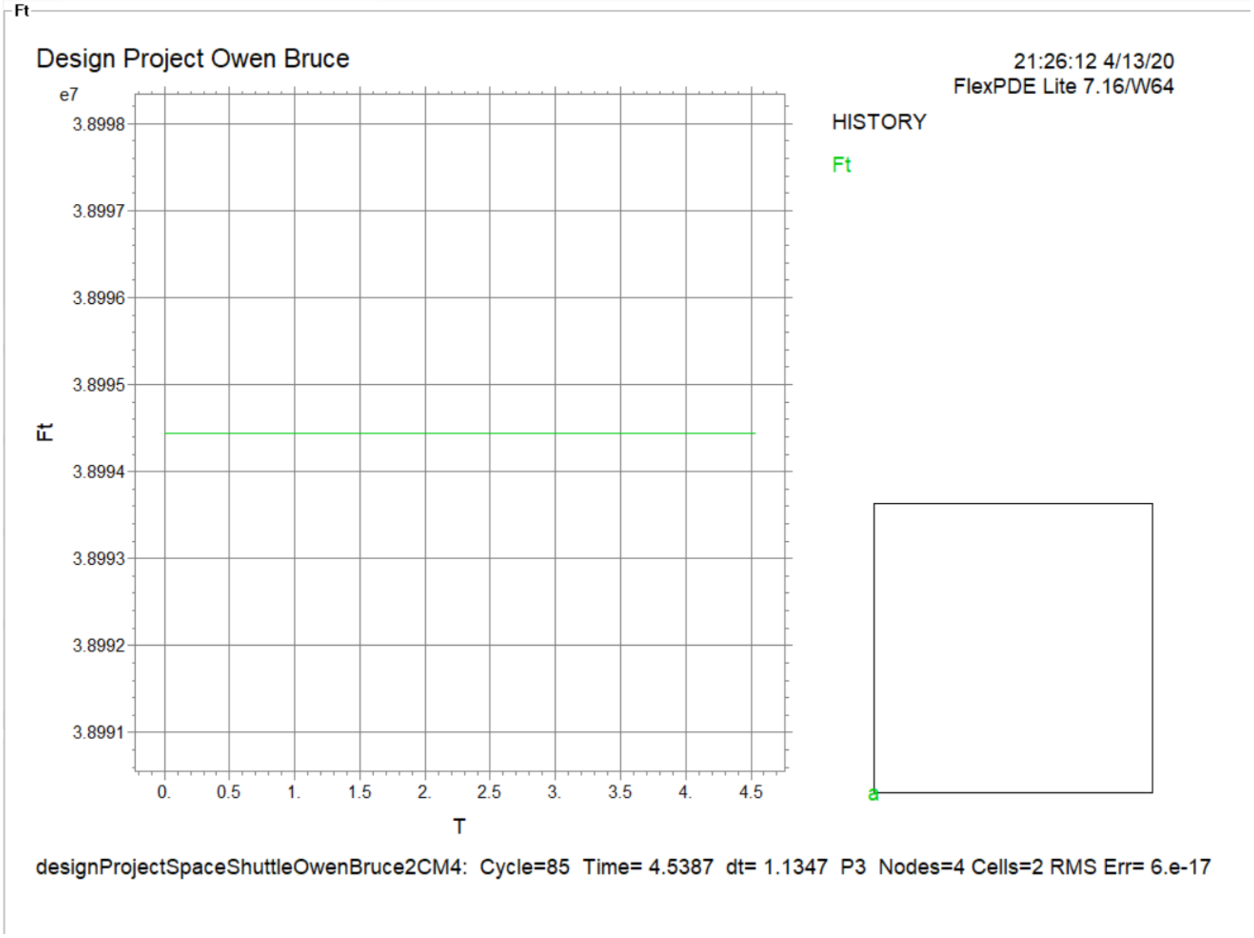

Figure 4.

4. <u>Scripting</u>

To optimize this rocket now, I am going to use a form of gradient descent. Gradient descent is a technique which allows for optimizing multiple parameters while minimizing the computational power required for the optimization. Gradient descent assumes that there is a continuous multi-dimensional function which is solvable for the value that you want to minimize. A starting value is chosen for each parameter, then they are all varied by a small amount. Then, if new point is lower, this becomes the new starting point and another point is chosen in the same direction.[16]

If the difference between the third and second point is larger than the difference between the first two points, then the step size will be slowly increased until the difference becomes lower. If at any time the value at the next point is larger than the previous, the new point will still be taken, but the direction of the search will be reversed, and the step size will be shrunk. This will then continue until a minimum value along this line is reached within the accuracy desired.

Then, a new direction will by chosen by varying the direction of one parameter. For example, if we are trying to find the minimal value of $z = f(x,y)$, the search will start along the $\delta x = \delta y$ line (eg. both x and y are increases or both are decreasing with each search). Then once a minimum point is found within the desired accuracy, one of these parameters will be changed, so now we are searching along the $\delta x = -\delta y$ line. This will then continue until a minimum value is found where changing any parameter will not result in lowering the value within accuracy.

For this case, we want to minimize the amount of fuel required for the rocket to reach 3000 km. The values which we are able to change are the combustion pressure, the combustion temperature, the area of the nozzle exit, and the area of the nozzle throat. Due to the nature of this problem however, an extra step must be taken after a new value is reached along each line. After a new value is reached, the amount of fuel that the rocket is carrying must also be reduced to the total amount of fuel and the simulation run again to truly minimize the amount of fuel that the rocket carries as the fuel can add a significant amount of mass to the rocket.

Another issue which must be considered is the boundary conditions for this optimization. As it is possible that the minimum amount of fuel required involves one or more parameters reaching unrealistic or infinite values, maximums and minimums for each parameter must be set such that the end result is one that is realistic. The values which are going to be taken as the minimum and maximum possible values for each parameter either come from the physical minimum or maximum possible, for example the combustion temperature, or the minimum and maximum used on rockets which have been produced. These values taken as the minimum/maximum will be taken to be slightly smaller or larger than the values which are already used if physically possible.

The minimum combustion temperature is 2,985 K [17], the maximum combustion temperature is 10,000 K [18], the minimum combustion pressure is 2 MPa, the maximum combustion pressure is 25 MPa [19], the minimum nozzle throat area is 0.1 m^2, the maximum nozzle throat area is 1 m^2, the minimum nozzle exit area is 1 m^2, and the maximum nozzle area is 15 m^2. Additionally, the maximum amount of time to reach 3 km will be set to 1 hour so that the rocket does not take too long to reach the maximum height.

To script the FlexPDE simulation, I am going to be using python. The primary part of the program will be to take the current input parameters, obtain the value of output, and then create new input parameters until the accuracy target is reached. This is formed in a nested while loop with the inner while loop optimizing along the line and the out while loop optimizing along each line until the initial change in the amount of fuel consumed is less than the accuracy desired. Before this is done however, the Mach value at the exit of the nozzle must be calculated in python. The Mach value is calculated using the solve function in sympy which is able to solve for the value of Me given the values of Ae and Astar.

Each time that the simulation runs, the program identifies whether the rocket made it to the altitude within the time limit set, used more fuel, or used less fuel. If the rocket did not make it to the altitude within the time limit, the value is not updated, and instead the step size is decreased by 10% and the simulation is run with the new parameters. If the rocket uses more fuel, then the value is updated and steps back with a step size reduced by 10%. If the rocket uses less fuel and the change in the amount of fuel is larger than the previous change, than the step size is increased by 10%. Finally, if the rocket uses less fuel and the change in the amount of fuel is smaller than the previous change, than the step size is decreased by 10%. Once the change in fuel becomes lower than the accuracy desired, then the inner loop stops running and one of the values is changed to be negative, iterating through each possible line.

Along the first line of descent, the simulation requires 49 simulations to get within an accuracy of 1 kg of fuel. This gives a total mass of fuel of 56043.666333 kg after the first optimization. Eventually, the simulation reaches a minimal fuel mass of 55581.23 kg. Figure 5. and 6. show the altitude of the rocket vs. time and the thrust force vs. time for the optimal parameters.

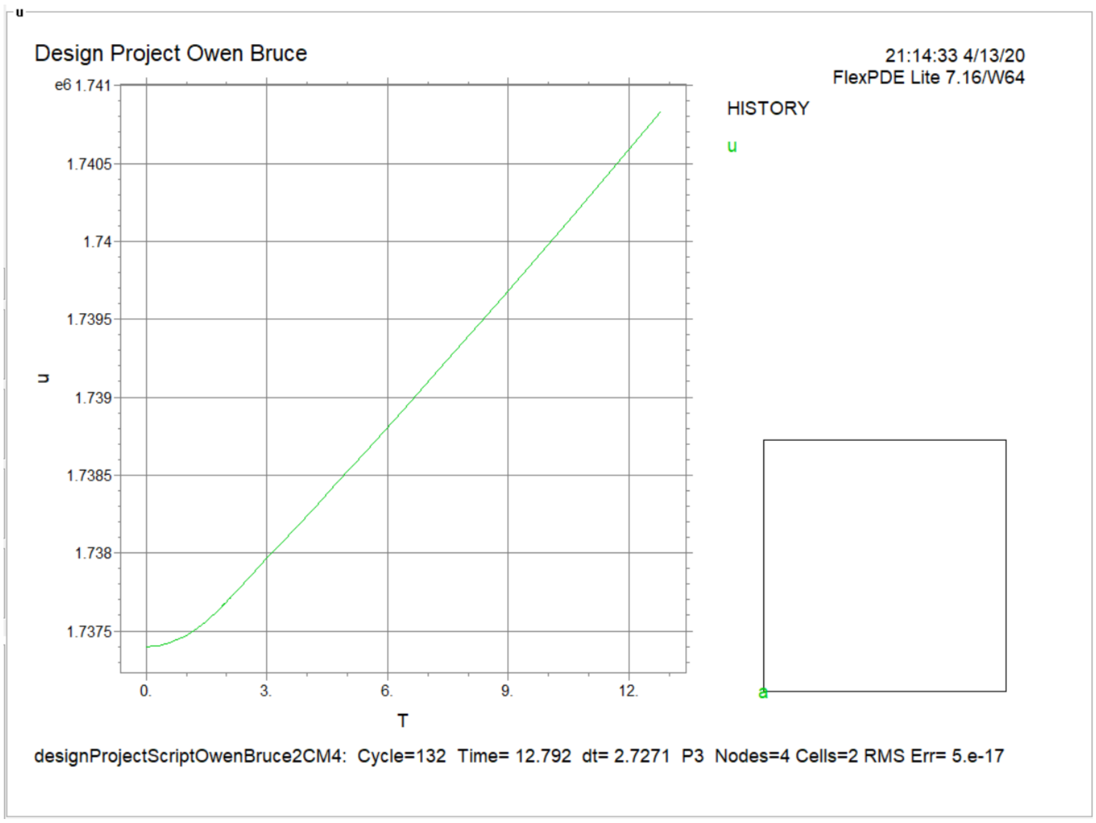

Figure 5.

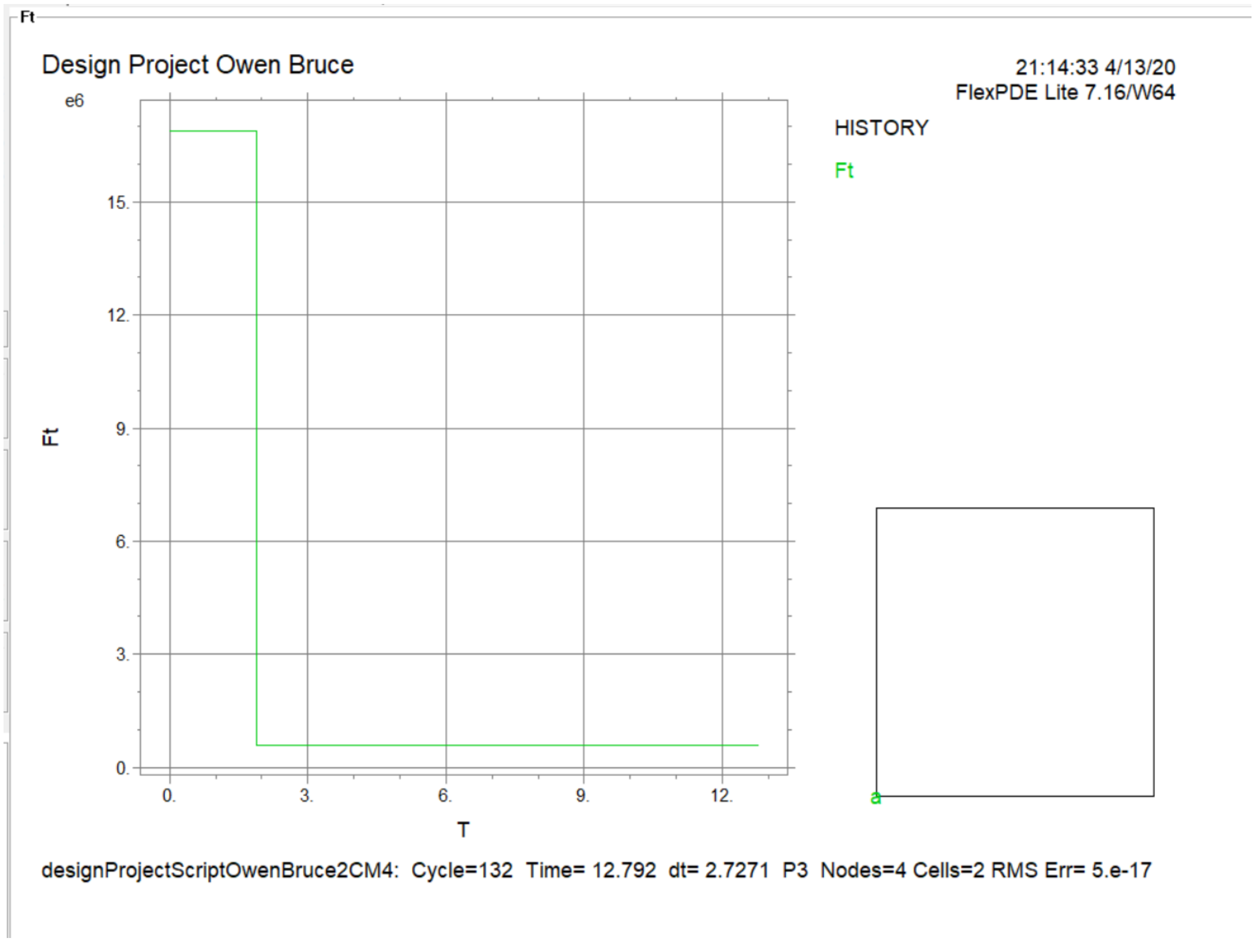

Figure 6.

5. Conclusion

In conclusion, I think that this optimization was successful; however, there is further room for improvement throughout the entire process. The amount of fuel required to lift this rocket above 3000 km is around 7 times smaller than that of the space shuttle simulation, but it is still a lot of fuel compared that required for missions that have gone to the moon in the past. One major reason for this is that the total mass of the rocket with the payload being simulated is around 10 times larger than that of the payload alone. Since the amount of fuel that the lunar module contained for ascent was approximately 5,200 lb (2,400 kg) and the dry weight was approximately 4,700 lb (2,100 kg), [3] the total amount of fuel required may be much higher but the ratio of dry weight to fuel is much higher. The ratio of dry weight to fuel for the lunar module was 0.875, while the ratio of dry weight to fuel for the optimized rocket was 2.079.

In addition, it is possible that the gradient descent became trapped inside a local minimum. It is apparent that there are local minima by looking through the change in fuel for each simulation along the first line. It could be the case that there is a lower amount of fuel possible for the rocket with different design parameters, but the method of optimization is unable to obtain these results in its current form. One way to improve this would be to run the optimization again but start on different legs of the optimization each time. This would result in either the same values being obtained indicating that it is likely a true minimum for the equation has been reached, or different values being obtained which would indicate that at least one or potentially all of the different gradient descents have become trapped in local minima. If this were the case, another method may have to be used in order to escape the local minima and to find the global minimum.

Overall, I am satisfied with the result of this project. I think that the rocket design obtained at the end of the optimization is not quite the design which I was anticipating, but I think that this design is very valuable. It shows that it is possible to create a highly efficient rocket to take off and land on the moon. By comparing the result from the space shuttle simulation to the optimized result, it also shows that the takeoff conditions on the moon and on the earth are very different, resulting in takeoffs from the moon being much more efficient than the on the earth.

6. References

[1] spacexcmsadmin, "Capabilities & Services," *SpaceX*, Nov. 27, 2012. https://www.spacex.com/about/capabilities (accessed Apr. 13, 2020).

[2] "Angelic halo orbit chosen for humankind's first lunar outpost." https://www.esa.int/Enabling_Support/Operations/Angelic_halo_orbit_chosen_for_humankind_s_first_lunar_outpost (accessed Apr. 11, 2020).

[3] "LM04_Lunar_Module_ppLV1-17.pdf." Accessed: Apr. 09, 2020. [Online]. Available: https://www.hq.nasa.gov/alsj/LM04_Lunar_Module_ppLV1-17.pdf.

[4] "NASA - NSSDCA - Spacecraft - Details." https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1969-059C (accessed Apr. 09, 2020).

[5] "Rocket Thrust Equation." https://www.grc.nasa.gov/WWW/K-12/rocket/rockth.html (accessed Apr. 11, 2020).

[6] "Rocket Thrust Equations." https://www.grc.nasa.gov/WWW/K-12/rocket/rktthsum.html (accessed Apr. 11, 2020).

[7] "Newton's law of universal gravitation," *Wikipedia*. Apr. 06, 2020, Accessed: Apr. 11, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Newton%27s_law_of_universal_gravitation&oldid=949443145.

[8] N. Administrator, "Is There an Atmosphere on the Moon?," *NASA*, Jun. 07, 2013. http://www.nasa.gov/mission_pages/LADEE/news/lunar-atmosphere.html (accessed Apr. 11, 2020).

[9] F. A. da S. Mota *et al.*, "Modeling and Analysis of a LOX/Ethanol Liquid Rocket Engine," *J. Aerosp. Technol. Manag.*, vol. 10, 2018, doi: 10.5028/jatm.v10.914.

[10] J. Wilson, "NASA - Space Shuttle Main Engines." https://www.nasa.gov/returntoflight/system/system_SSME.html (accessed Apr. 12, 2020).

[11] "Shuttle technical facts." https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Space_Shuttle/Shuttle_technical_facts (accessed Apr. 12, 2020).

[12] "Gas constant," *Wikipedia*. Mar. 17, 2020, Accessed: Apr. 12, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gas_constant&oldid=946002765.

[13] "Gravitational constant," *Wikipedia*. Apr. 09, 2020, Accessed: Apr. 12, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gravitational_constant&oldid=949873874.

[14] "Specific Heats." https://www.grc.nasa.gov/WWW/Wright/airplane/specheat.html (accessed Apr. 12, 2020).

[15] "Moon Fact Sheet." https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html (accessed Apr. 12, 2020).

[16] P. Pandey, "Understanding the Mathematics behind Gradient Descent.," *Medium*, Jan. 17, 2020. https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e (accessed Apr. 13, 2020).

[17] "Lox/LH2." http://www.astronautix.com/l/loxlh2.html (accessed Apr. 13, 2020).

[18] M. Faisal, Z. Quadri, P. Kulkarni, U. Guven, S. Bhat, and K. Sundarraj, "CFD SIMULATION OF A LIQUID ROCKET PROPELLANT (LH 2 /LOx) COMBUSTION CHAMBER," Aug. 2013.

[19] "Cryogenic rocket engine," *Wikipedia*. Mar. 02, 2020, Accessed: Apr. 13, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cryogenic_rocket_engine&oldid=943537768.

7. Appendix

a. Space Shuttle Simulation Code

```
{ Fill in the following sections (removing comment marks ! if necessary),
  and delete those that are unused.}
TITLE 'Design Project Owen Bruce'     { the problem identification }
COORDINATES cartesian2  { coordinate system, 1D,2D,3D, etc }
VARIABLES        { system variables }
  u (threshold=1e-3)
  v (threshold=1e-3)
SELECT         { method controls }
	ngrid = 1
DEFINITIONS    { parameter definitions }
	MoonRad = 1737.4e3


	Mfuel = 729465
	Mr = 104535
    Mp = 11e3


	Astar = 1 !Rocket Parameters
    Pt = 226*100e3
    Tt = 3315.6+273.15
    gamma = 1.4
    Rg = 8.314
    p0 = 3*(10e-15)*100e3
    Ae = 45
    G = 6.67e-11


    Mm = 7.346e22
```

```
  Me = 5.768

     mdot = Astar*pt/sqrt(Tt)*sqrt(gamma/Rg)*((gamma+1)/2)^(-(gamma+1)/(2*(gamma-1)))   !Thrust Equations
  Te = Tt*(1+(gamma-1)/2*Me^2)^(-1)
  pe = pt*(1+(gamma-1)/2*Me^2)^(-gamma/(gamma-1))
  ve = Me*sqrt(gamma*Rg*Te)

  Tfuel = Mfuel/mdot

  M = if t<tfuel then Mfuel + Mr + Mp - tfuel*mdot else Mr+Mp

     Ft = if t<tfuel then mdot*Ve+(pe-p0)*Ae else (pe-p0)*Ae   !Force Equations
  Fg = -G*M*mm/u^2
  Fnet = Ft + Fg
  a = Fnet/M
  r = u
INITIAL VALUES
u = MoonRad
EQUATIONS        { PDE's, one for each variable }
     u: dt(u) = v
  v: dt(v) = a
! CONSTRAINTS    { Integral constraints }
BOUNDARIES       { The domain definition }
 REGION 1       { For each material region }
  START(0,0)   { Walk the domain boundary }
  LINE TO (1,0) TO (1,1) TO (0,1) TO CLOSE
TIME 0 TO 3600 halt(u-MoonRad>3e3 or u<(MoonRad-1e3))    { if time dependent }
```

```
MONITORS        { show progress }
PLOTS           { save result displays }
for time = endtime
  history(u) at (0,0)
  history(Fg) at (0,0)
  history(Ft) at (0,0)
Summary
        report(if val(u,0,0)<(MoonRad+2.9e3) then 0 else if t<tfuel then tintegral(mdot) else mdot*tfuel) as 'Total mass of fuel consumed'
END
```

b. Python Optimization Code

```
import subprocess
import scipy as sp
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time
import sympy as sym

startTime = time.time()

FlexCode = """TITLE 'Design Project Owen Bruce'    { the problem identification }
COORDINATES cartesian2  { coordinate system, 1D,2D,3D, etc }
VARIABLES       { system variables }
  u (threshold=1e-3)
  v (threshold=1e-3)
SELECT       { method controls }
        ngrid = 1
```

```
DEFINITIONS    { parameter definitions }
        MoonRad = 1737.4e3   !Constants
   Mm = 7.346e22          !Moon Mass
   Mr = 104535   !Rocket Mass
   Mp = 11e3     !Payload Mass
   gamma = 1.4
   Rg = 8.314
   p0 = 3*(10e-15)*100e3          !External Pressure
   G = 6.67e-11



        Astar = %s !Rocket Parameters
   Ae = %s
   Pt = %s
   Tt = %s
   Mfuel =%s
   Me = %s



        mdot = Astar*pt/sqrt(Tt)*sqrt(gamma/Rg)*((gamma+1)/2)^(-(gamma+1)/(2*(gamma-1)))
!Thrust Equations
   Te = Tt*(1+(gamma-1)/2*Me^2)^(-1)
   pe = pt*(1+(gamma-1)/2*Me^2)^(-gamma/(gamma-1))
   ve = Me*sqrt(gamma*Rg*Te)

   Tfuel = Mfuel/mdot

   M = if t<tfuel then Mfuel + Mr + Mp - tfuel*mdot else Mr+Mp
```

```
        Ft = if t<tfuel then mdot*Ve+(pe-p0)*Ae else (pe-p0)*Ae   !Force Equations
  Fg = -G*M*mm/u^2
  Fnet = Ft + Fg
  a = Fnet/M
INITIAL VALUES
u = MoonRad
EQUATIONS        { PDE's, one for each variable }
        u: dt(u) = v
  v: dt(v) = a
! CONSTRAINTS    { Integral constraints }
BOUNDARIES       { The domain definition }
 REGION 1       { For each material region }
  START(0,0)   { Walk the domain boundary }
  LINE TO (1,0) TO (1,1) TO (0,1) TO CLOSE
TIME 0 TO 3600 halt(u-MoonRad>3e3 or u<(MoonRad-1e3))    { if time dependent }
MONITORS         { show progress }
PLOTS            { save result displays }
for time = endtime
 history(u) at (0,0)
 history(Fg) at (0,0)
 history(Ft) at (0,0)
 history(if val(u,0,0)<MoonRad+2.9e3 then 0 else if t<tfuel then tintegral(mdot) else mdot*tfuel)
        Export Format '#t#b#1' file = 'FuelConsumed.txt'
Summary
        report(if t<tfuel then tintegral(mdot) else mdot*tfuel) as 'Total mass of fuel consumed'
END
"""
```

```
FlexFileName = "D:\\EngPhys\\2CM4\\DesignProject\\designProjectScriptOwenBruce2CM4.pde"
#Change to whatever folder and file you want


outcome = 0 #Variable equals 0 if the next value is lower, variable equals 1 if the next value is higher, variable equals 2 if the rocket fails to reach the required height, variable equals 3 if the simulation crashes


#Initial Values for Parameters

Astar = 1

Ae = 10

Pt = 10*10**6

Tt = 5000

Mfuel = 780000


#Constants

gamma = 1.4


#Paramters

initDeltaFuel = 10

deltaFuel = 10

deltaFuelPrevious = 0

fuelArray = np.array([])

fuelConsumedNew = 0

fuelConsumedPrevious = 235364

accuracy = 1

maxAstar = 1

minAstar = 0.1

deltaAstar = 0.01

maxAe = 15

minAe = 1
```

```
deltaAe = 0.01

maxPt = 25*10**6

minPt = 2*10**6

deltaPt = 5*10**4

minTt = 2985

maxTt = 10000

deltaTt = 50

simulation = 0

line = 0


while initDeltaFuel > accuracy:


  line += 1


  while deltaFuel > accuracy:    #Continue to change the same amount while the fuel amount is changing

    simulation += 1

    #First, we need to solve for Me based on Ae and Astar


    w = sym.symbols('w')


    f = sym.Eq(Ae/Astar,((gamma+1)/2)**(-(gamma+1)/(2*(gamma-1)))*(1+(gamma-
1)*w**2/2)**((gamma+1)/(2*(gamma-1)))/w)

    a = sym.solve(f,w)


    Me = a[1]




    with open(FlexFileName, "w") as f:

      print(FlexCode%(Astar,Ae,Pt,Tt,Mfuel,Me), file=f)
```

```
    #completed = subprocess.run(["ls"], shell=True)

    completed =subprocess.Popen(["FlexPDE7n","-S", FlexFileName])   #Change to FlexPDE6s if that's
what you are running

    time.sleep(0.2)

    print('Started')

    completed.wait()


    try:

      with
open("D:\\EngPhys\\2CM4\\DesignProject\\designProjectScriptOwenBruce2CM4_output\\FuelConsum
ed.txt") as f:   #Change to whatever folder the output is in

        data=sp.loadtxt(f, skiprows=8)

      fuelArray=data[:,1]

      if(fuelArray[-1]==0):

        outcome = 2

        print("Simulation #",simulation," on line #",line," failed to reach the required height.")

      elif(fuelArray[-1] >= fuelConsumedPrevious):

        outcome = 1

        fuelConsumedNew = fuelArray[-1]

        print("Simulation #",simulation," on line #",line," used more fuel.")

      else:

        outcome = 0

        fuelConsumedNew = fuelArray[-1]

        print("Simulation #",simulation," on line #",line," used less fuel.")


    except:

      print("Failed")

      outcome = 3
```

```
Mfuel = fuelConsumedNew
deltaFuel = abs(fuelConsumedNew-fuelConsumedPrevious)

if(outcome==2):
    Ae -= deltaAe
    Astar -= deltaAstar
    Pt -= deltaPt
    Tt -= deltaTt
    deltaAe *= 0.9
    deltaAstar *= 0.9
    deltaPt *= 0.9
    deltaTt *= 0.9

elif(outcome==1):
    deltaAe *= -0.9
    deltaAstar *= -0.9
    deltaPt *= -0.9
    deltaTt *= -0.9

elif(outcome==0):

    if(deltaFuel>=deltaFuelPrevious*0.9):
        deltaAe *= 1.1
        deltaAstar *= 1.1
        deltaPt *= 1.1
        deltaTt *= 1.1

    else:
        deltaAe *= 0.9
```

```
        deltaAstar *= 0.9
        deltaPt *= 0.9
        deltaTt *= 0.9


    if(Ae+deltaAe<maxAe and Ae+deltaAe>minAe):
      Ae+=deltaAe

    if(Astar+deltaAstar<maxAstar and Astar+deltaAstar>minAstar):
      Astar+=deltaAstar

    if(Pt+deltaPt<maxPt and Pt+deltaPt>minPt):
      Pt+=deltaPt

    if(Tt+deltaTt<maxTt and Tt+deltaTt>minTt):
      Tt+=deltaTt

    if(simulation == 1):
      initDeltaFuel = deltaFuel

    deltaFuelPrevious = deltaFuel
    fuelConsumedPrevious = fuelConsumedNew

    print("Ae =", Ae)
    print("Astar =", Astar)
    print("Pt =", Pt)
    print("Tt =", Tt)
    print("Mfuel =", Mfuel)
    print("Delta fuel =",deltaFuel)
```

```
if(line==5):
   line -= 4

if(line==1):
   deltaAe *= -2
   deltaAstar *= 2
   deltaPt *= 2
   deltaTt *= 2

elif(line==2):
   deltaAe *= 2
   deltaAstar *= -2
   deltaPt *= 2
   deltaTt *= 2

elif(line==3):
   deltaAe *= 2
   deltaAstar *= 2
   deltaPt *= -2
   deltaTt *= 2

elif(line==4):
   deltaAe *= 2
   deltaAstar *= 2
   deltaPt *= 2
   deltaTt *= -2
```

```
    if(Ae+deltaAe<maxAe and Ae+deltaAe>minAe):
        Ae+=deltaAe

    if(Astar+deltaAstar<maxAstar and Astar+deltaAstar>minAstar):
        Astar+=deltaAstar

    if(Pt+deltaPt<maxPt and Pt+deltaPt>minPt):
        Pt+=deltaPt

    if(Tt+deltaTt<maxTt and Tt+deltaTt>minTt):
        Tt+=deltaTt

    simulation = 0
    deltaFuel = 10



print("Minimized Fuel Consumed: ", fuelConsumedNew)


print("Full simulation takes",time.time()-startTime,"s")
```

c. Full Python Output

```
Simulation # 1  on line # 1  used more fuel.
Ae = 9.991
Astar = 0.991
Pt = 9955000.0
Tt = 4955.0
Mfuel = 240970.75786
Delta fuel = 5606.757860000012
Started
```

Simulation # 2 on line # 1 used less fuel.

Ae = 9.982899999999999

Astar = 0.9829

Pt = 9914500.0

Tt = 4914.5

Mfuel = 240403.03251

Delta fuel = 567.7253500000224

Started

Simulation # 3 on line # 1 used less fuel.

Ae = 9.973989999999999

Astar = 0.97399

Pt = 9869950.0

Tt = 4869.95

Mfuel = 239891.25702

Delta fuel = 511.77549

Started

Simulation # 4 on line # 1 used less fuel.

Ae = 9.964189

Astar = 0.964189

Pt = 9820945.0

Tt = 4820.945

Mfuel = 239327.39966

Delta fuel = 563.8573599999945

Started

Simulation # 5 on line # 1 used less fuel.

Ae = 9.953407899999998

Astar = 0.9534079

Pt = 9767039.5

Tt = 4767.0395

Mfuel = 238706.06366

Delta fuel = 621.3359999999811

Started

Simulation # 6 on line # 1 used less fuel.

Ae = 9.941548689999998

Astar = 0.94154869

Pt = 9707743.45

Tt = 4707.74345

Mfuel = 238021.27103

Delta fuel = 684.7926300000108

Started

Simulation # 7 on line # 1 used less fuel.

Ae = 9.928503558999997

Astar = 0.9285035589999999

Pt = 9642517.795

Tt = 4642.517795

Mfuel = 237266.39793

Delta fuel = 754.8730999999971

Started

Simulation # 8 on line # 1 used less fuel.

Ae = 9.914153914899998

Astar = 0.9141539148999999

Pt = 9570769.5745

Tt = 4570.7695745

Mfuel = 236434.10041

Delta fuel = 832.2975199999928

Started

Simulation # 9 on line # 1 used less fuel.

Ae = 9.898369306389998

Astar = 0.8983693063899999

Pt = 9491846.53195

Tt = 4491.84653195

Mfuel = 235516.23096

Delta fuel = 917.8694500000274

Started

Simulation # 10 on line # 1 used less fuel.

Ae = 9.881006237028998

Astar = 0.8810062370289999

Pt = 9405031.185145

Tt = 4405.031185145

Mfuel = 234503.7449

Delta fuel = 1012.4860599999956

Started

Simulation # 11 on line # 1 used less fuel.

Ae = 9.861906860731898

Astar = 0.8619068607318999

Pt = 9309534.3036595

Tt = 4309.5343036594995

Mfuel = 233386.59543

Delta fuel = 1117.149470000004

Started

Simulation # 12 on line # 1 used less fuel.

Ae = 9.840897546805088

Astar = 0.8408975468050899

Pt = 9204487.73402545

Tt = 4204.48773402545

Mfuel = 232153.61663

Delta fuel = 1232.9787999999826

Started

Simulation # 13 on line # 1 used less fuel.

Ae = 9.817787301485597

Astar = 0.8177873014855989

Pt = 9088936.507427996

Tt = 4088.9365074279945

Mfuel = 230792.39386

Delta fuel = 1361.222769999993

Started

Simulation # 14 on line # 1 used less fuel.

Ae = 9.792366031634156

Astar = 0.7923660316341588

Pt = 8961830.158170795

Tt = 3961.8301581707938

Mfuel = 229289.12215

Delta fuel = 1503.271710000001

Started

Simulation # 15 on line # 1 used less fuel.

Ae = 9.764402634797571

Astar = 0.7644026347975746

Pt = 8822013.173987875

Tt = 3822.013173987873

Mfuel = 227628.455

Delta fuel = 1660.667150000023

Started

Simulation # 16 on line # 1 used less fuel.

Ae = 9.73364289827733

Astar = 0.7336428982773321

Pt = 8668214.491386661

Tt = 3668.21449138666

Mfuel = 225793.35023

Delta fuel = 1835.1047699999763

Started

Simulation # 17 on line # 1 used less fuel.

Ae = 9.699807188105062

Astar = 0.6998071881050653

Pt = 8499035.940525327

Tt = 3499.0359405253257

Mfuel = 223764.92689

Delta fuel = 2028.4233400000085

Started

Simulation # 18 on line # 1 used less fuel.

Ae = 9.662587906915569

Astar = 0.6625879069155718

Pt = 8312939.53457786

Tt = 3312.939534577858

Mfuel = 221522.36321

Delta fuel = 2242.563679999992

Started

Simulation # 19 on line # 1 used less fuel.

Ae = 9.621646697607126

Astar = 0.6216466976071289

Pt = 8108233.488035645

Tt = 3108.233488035644

Mfuel = 219042.89697

Delta fuel = 2479.466240000009

Started

Simulation # 20 on line # 1 used less fuel.

Ae = 9.576611367367839

Astar = 0.5766113673678417

Pt = 7883056.836839209

Tt = 3108.233488035644

Mfuel = 216302.05744

Delta fuel = 2740.8395299999975

Started

Simulation # 21 on line # 1 used less fuel.

Ae = 9.527072504104623

Astar = 0.5270725041046258

Pt = 7635362.52052313

Tt = 3108.233488035644

Mfuel = 210333.52268

Delta fuel = 5968.53476000001

Started

Simulation # 22 on line # 1 used less fuel.

Ae = 9.472579754515086

Astar = 0.4725797545150883

Pt = 7362898.772575443

Tt = 3108.233488035644

Mfuel = 197947.0724

Delta fuel = 12386.45027999999

Started

Simulation # 23 on line # 1 used less fuel.

Ae = 9.412637729966596

Astar = 0.4126377299665971

Pt = 7063188.649832986

Tt = 3108.233488035644

Mfuel = 164717.09152

Delta fuel = 33229.98088000002

Started

Simulation # 24 on line # 1 used less fuel.

Ae = 9.358689907872954

Astar = 0.358689907872955

Pt = 6793449.539364776

Tt = 3108.233488035644

Mfuel = 150930.64706

Delta fuel = 13786.444459999999

Started

Simulation # 25 on line # 1 used less fuel.

Ae = 9.310136867988676

Astar = 0.3101368679886771

Pt = 6550684.339943387

Tt = 3108.233488035644

Mfuel = 142884.40803

Delta fuel = 8046.239029999997

Started

Simulation # 26 on line # 1 used less fuel.

Ae = 9.266439132092826

Astar = 0.26643913209282705

Pt = 6332195.660464136

Tt = 3108.233488035644

Mfuel = 139880.00647

Delta fuel = 3004.4015599999984

Started

Simulation # 27 on line # 1 used less fuel.

Ae = 9.21837162260739

Astar = 0.21837162260739193

Pt = 6091858.11303696

Tt = 3108.233488035644

Mfuel = 131044.32312

Delta fuel = 8835.683349999992

Started

Simulation # 28 on line # 1 used less fuel.

Ae = 9.165497362173411

Astar = 0.16549736217341332

Pt = 5827486.810867067

Tt = 3108.233488035644

Mfuel = 112491.95807

Delta fuel = 18552.365050000008

Started

Simulation # 29 on line # 1 used less fuel.

Ae = 9.117910527782831

Astar = 0.11791052778283256

Pt = 5589552.638914163

Tt = 3108.233488035644

Mfuel = 97140.510367

Delta fuel = 15351.447702999998

Started

Simulation # 30 on line # 1 used less fuel.

Ae = 9.065565009953191

Astar = 0.11791052778283256

Pt = 5327825.049765969

Tt = 3108.233488035644

Mfuel = 76972.784718

Delta fuel = 20167.725649

Started

Simulation # 31 on line # 1 used less fuel.

Ae = 9.018454043906516

Astar = 0.11791052778283256

Pt = 5092270.219532594

Tt = 3108.233488035644

Mfuel = 75516.158647

Delta fuel = 1456.6260709999915

Started

Simulation # 32 on line # 1 used less fuel.

Ae = 8.966631981255173

Astar = 0.11791052778283256

Pt = 4833159.906275881

Tt = 3108.233488035644

Mfuel = 73127.498793

Delta fuel = 2388.6598539999977

Started

Simulation # 33 on line # 1 used less fuel.

Ae = 8.919992124868964

Astar = 0.11791052778283256

Pt = 4599960.624344841

Tt = 3108.233488035644

Mfuel = 71667.682482

Delta fuel = 1459.8163110000023

Started

Simulation # 34 on line # 1 used less fuel.

Ae = 8.868688282844134

Astar = 0.11791052778283256

Pt = 4343441.414220695

Tt = 3108.233488035644

Mfuel = 70198.837679

Delta fuel = 1468.844803

Started

Simulation # 35 on line # 1 used less fuel.

Ae = 8.812254056616823

Astar = 0.11791052778283256

Pt = 4061270.2830841355

Tt = 3108.233488035644

Mfuel = 68721.862744

Delta fuel = 1476.9749350000056

Started

Simulation # 36 on line # 1 used less fuel.

Ae = 8.761463253012241

Astar = 0.11791052778283256

Pt = 3807316.265061232

Tt = 3108.233488035644

Mfuel = 67678.2509

Delta fuel = 1043.611843999999

Started

Simulation # 37 on line # 1 used less fuel.

Ae = 8.705593369047202

Astar = 0.11791052778283256

Pt = 3527966.845236038

Tt = 3108.233488035644

Mfuel = 65426.50413

Delta fuel = 2251.746769999998

Started

Simulation # 38 on line # 1 used less fuel.

Ae = 8.655310473478668

Astar = 0.11791052778283256

Pt = 3276552.3673933633

Tt = 3108.233488035644

Mfuel = 63999.588857

Delta fuel = 1426.9152729999987

Started

Simulation # 39 on line # 1 used less fuel.

Ae = 8.610055867466986

Astar = 0.11791052778283256

Pt = 3050279.337334956

Tt = 3108.233488035644

Mfuel = 63023.288328

Delta fuel = 976.3005290000001

Started

Simulation # 40 on line # 1 used less fuel.

Ae = 8.560275800854136

Astar = 0.11791052778283256

Pt = 2801379.004270708

Tt = 3108.233488035644

Mfuel = 61672.512448

Delta fuel = 1350.775880000001

Started

Simulation # 41 on line # 1 used less fuel.

Ae = 8.505517727580001

Astar = 0.11791052778283256

Pt = 2527588.6379000354

Tt = 3108.233488035644

Mfuel = 59895.749382

Delta fuel = 1776.7630659999995

Started

Simulation # 42 on line # 1 used less fuel.

Ae = 8.445283846978453

Astar = 0.11791052778283256

Pt = 2226419.234892295

Tt = 3108.233488035644

Mfuel = 58142.400587

Delta fuel = 1753.3487950000053

Started

Simulation # 43 on line # 1 used less fuel.

Ae = 8.39107335443706

Astar = 0.11791052778283256

Pt = 2226419.234892295

Tt = 3108.233488035644

Mfuel = 57057.691311

Delta fuel = 1084.7092759999941

Started

Simulation # 44 on line # 1 used less fuel.

Ae = 8.342283911149806

Astar = 0.11791052778283256

Pt = 2226419.234892295

Tt = 3108.233488035644

Mfuel = 56809.323864

Delta fuel = 248.36744700000418

Started

Simulation # 45 on line # 1 used less fuel.

Ae = 8.298373412191278

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56727.972278

Delta fuel = 81.35158599999704

Started

Simulation # 46 on line # 1 used less fuel.

Ae = 8.250071863336897

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56092.502467

Delta fuel = 635.4698110000027

Started

Simulation # 47 on line # 1 used less fuel.

Ae = 8.206600469367954

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56081.843458

Delta fuel = 10.659008999995422

Started

Simulation # 48 on line # 1 used less fuel.

Ae = 8.158781936002116

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56043.797105

Delta fuel = 38.04635300000518

Started

Simulation # 49 on line # 1 used less fuel.

Ae = 8.115745255972863

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56043.666333

Delta fuel = 0.13077199999679578

Started

Simulation # 1 on line # 2 used less fuel.

Ae = 8.29649931209573

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56040.429921

Delta fuel = 3.2364119999983814

Started

Simulation # 2 on line # 2 used less fuel.

Ae = 8.400648077766524

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56034.175931

Delta fuel = 6.2539900000047055

Started

Simulation # 3 on line # 2 used less fuel.

Ae = 8.515211720004398

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56023.221092

Delta fuel = 10.954838999998174

Started

Simulation # 4 on line # 2 used less fuel.

Ae = 8.64123172646606

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 56011.40085

Delta fuel = 11.820242000001599

Started

Simulation # 5 on line # 2 used less fuel.

Ae = 8.779853733573887

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55998.667668

Delta fuel = 12.733181999996305

Started

Simulation # 6 on line # 2 used less fuel.

Ae = 8.932337941392497

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55984.974969

Delta fuel = 13.692698999999266

Started

Simulation # 7 on line # 2 used less fuel.

Ae = 9.100070569992967

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55970.277829

Delta fuel = 14.697140000003856

Started

Simulation # 8 on line # 2 used less fuel.

Ae = 9.284576461453485

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55954.533753

Delta fuel = 15.744075999995403

Started

Simulation # 9 on line # 2 used less fuel.

Ae = 9.487532942060055

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55937.703511

Delta fuel = 16.830242000003636

Started

Simulation # 10 on line # 2 used less fuel.

Ae = 9.710785070727281

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55919.752028

Delta fuel = 17.951482999997097

Started

Simulation # 11 on line # 2 used less fuel.

Ae = 9.956362412261232

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55900.649305

Delta fuel = 19.102723000003607

Started

Simulation # 12 on line # 2 used less fuel.

Ae = 10.226497487948576

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55880.371356

Delta fuel = 20.277948999995715

Started

Simulation # 13 on line # 2 used less fuel.

Ae = 10.523646071204656

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55858.901129

Delta fuel = 21.470227000005252

Started

Simulation # 14 on line # 2 used less fuel.

Ae = 10.850509512786342

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55836.229375

Delta fuel = 22.671753999995417

Started

Simulation # 15 on line # 2 used less fuel.

Ae = 11.210059298526197

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55812.355447

Delta fuel = 23.873928000000888

Started

Simulation # 16 on line # 2 used less fuel.

Ae = 11.605564062840038

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55787.287978

Delta fuel = 25.067469000001438

Started

Simulation # 17 on line # 2 used less fuel.

Ae = 12.040619303585263

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55761.045417

Delta fuel = 26.242560999999114

Started

Simulation # 18 on line # 2 used less fuel.

Ae = 12.519180068405012

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55733.656381

Delta fuel = 27.389036000000488

Started

Simulation # 19 on line # 2 used less fuel.

Ae = 13.045596909706735

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55705.159808

Delta fuel = 28.49657300000399

Started

Simulation # 20 on line # 2 used less fuel.

Ae = 13.62465543513863

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55675.604882

Delta fuel = 29.554925999997067

Started

Simulation # 21 on line # 2 used less fuel.

Ae = 14.261619813113715

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55645.050725

Delta fuel = 30.554156999998668

Started

Simulation # 22 on line # 2 used less fuel.

Ae = 14.962280628886308

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55613.565852

Delta fuel = 31.484873000001244

Started

Simulation # 23 on line # 2 used less fuel.

Ae = 14.962280628886308

Astar = 0.11791052778283256

Pt = 2006866.7400996527

Tt = 3108.233488035644

Mfuel = 55581.227399

Delta fuel = 32.33845299999666

Started

Simulation # 24 on line # 2 used more fuel.

Ae = 14.26862642127144

Astar = 0.8115647353977

Pt = 5475137.778173989

Tt = 6576.504526109982

Mfuel = 55581.227399

Delta fuel = 0.0

Started

Simulation # 1 on line # 3 used more fuel.

Ae = 14.129895579748467

Astar = 0.8115647353977

Pt = 6168791.985788855

Tt = 6576.504526109982

Mfuel = 55581.227399

Delta fuel = 0.0

Minimized Fuel Consumed: 55581.227399

Full simulation takes 176.4969778060913 s