M20 - Introduction to Computer Programming with MATLAB
Instructor: Prof. Enrique López Droguett, Ph.D.
Teacher Assistants: M. Fidansoy, G. San Martín, M. Pishahang, V. Vela.
Fall 2023 – UCLA
Student: *Alex Lie*
UCLA ID: *905901892*

HOMEWORK 4

Task 1: Golden Search Method

Introduction

I am using MATLAB to minimize the amount of stainless steel needed to create a cylindrical water tank. This water tank needs to hold 2800 gallons. I will optimize the height and diameter of the water tank to have the smallest surface area by implementing the Golden Search Method, which finds the minimum of a function by iteratively narrowing the range of values the minimum exists in. I will also use different cases of the Golden Search Method with different parameters.

Model and Theory

1) $Surface\ Area\ of\ a\ Cylinder\ = 2\pi(\frac{D}{2})^2 + \pi DH$
2) $Volume\ of\ a\ Cylinder\ = \pi(\frac{D}{2})^2 H$

Methodology

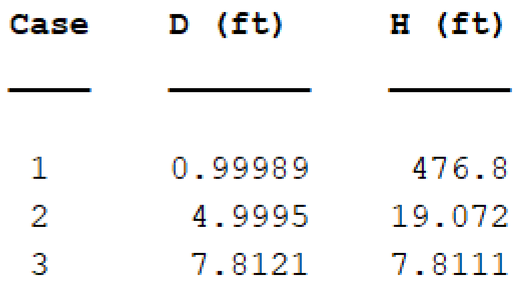
I first cleared the workspace and command window. Then I created a variable for the volume the tank needed to hold (and converted the value to the correct units). Then I created a function that would calculate the surface area of the tank in terms of the diameter and the volume of 2880 gallons by combining equations 1 and 2. Then I defined variables with values that are needed for the Golden Search method given in the problem statement for all three cases. These variables were the lower bound for x (xl), the upper bound for x (xu), the target approximation error (es), and the maximum number of iterations (maxit). I was then able to run the Golden Search method three times by using the goldenSearch() function, using the variables for each case as parameters. The output of the goldenSearch() method is the optimized diameter value that minimizes the surface area (Dmin), the value of the minimized surface area (SAmin), the approximation error at the optimized diameter (ea), and the iteration number at the optimized diameter (iter). I was able to put the three optimized diameters in a vector. I calculated the heights for each optimized diameter from each case and then put them in a vector as well. I then used the table() function to show the optimized diameter and height in each case in a table. I also output the surface area values for each case. Lastly, I added comments to make my code easier to understand for other users.

Calculations and results

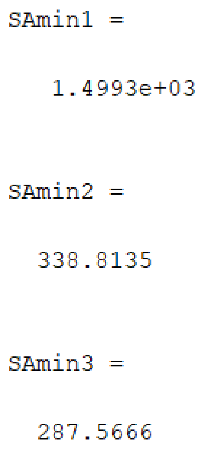
See next page.

M20 - Introduction to Computer Programming with MATLAB
Instructor: Prof. Enrique López Droguett, Ph.D.
Teacher Assistants: M. Fidansoy, G. San Martín, M. Pishahang, V. Vela.
Fall 2023 – UCLA
Student: *Alex Lie*
UCLA ID: *905901892*

3)



```
Case      D (ft)      H (ft)
____    _______    ______

 1     0.99989     476.8
 2      4.9995    19.072
 3      7.8121    7.8111
```



```
SAmin1 =

   1.4993e+03


SAmin2 =

  338.8135


SAmin3 =

  287.5666
```

The table shows the optimized diameter and height to minimize the surface area in cases 1, 2, and 3. The output on the right shows the surface area values for cases 1, 2, and 3.

Discussions and Conclusions

In all three cases, the lower bound, target approximation error, and the maximum number of iterations were all the same. The only thing that differed between the cases was the upper bound. After running all three cases of the Golden Search Method, we are able to fill in the table and see three different values of the optimized diameters and heights. The optimized diameter is always going to be less than or equal to the upper bound. It is never going to be larger than the upper bound because the algorithm cannot test values outside of the range. We see in case 1 and 2, the optimized diameter is really close in value to its upper bound (1 and 5 respectively). This means the minimum of the surface area function is at the upper bound and it could potentially continue to get lower past that bound. Based on the table, as we increase the upper bound, the optimized diameter increases and the optimized height decreases. Using an upper bound greater than 7.8121 will produce the same results in case 3, because case 3 is the minimum value of the surface area.

Case 3's optimized tank diameter and height values are the better design. This is because it has the smallest surface area out of the three cases. Looking at the output (3) the surface area in case 3 is smaller than the surface area in case 1 and 2. This is also because case 3 has a larger range of values starting at 0, compared to case 1 and 2. It happens that the minimized diameter is in the range in case 3, but not case 1 or 2. I also noticed that the diameter and height of the case 3 cylinder are almost the exact same value. After doing research online, it can be concluded that the surface area of a cylinder can be minimized when the diameter and height values are equal, regardless of the volume.

Task 2: Parabolic Interpolation Optimization

Introduction

I am using MATLAB to find the minimum total drag force and the corresponding velocity for an airfoil with a range of different weights. To do this, I am implementing the parabolic interpolation optimization method, which iteratively fits parabolas to a function using three different points on that function until it closely matches the function. I will also create plots for the drag force for a 17,500 lb airfoil and make a scatter plot of the minimum drag forces with the corresponding velocities with different airfoil weights.

Model and Theory

1. $F_D = 0.01\sigma V^2 + \frac{0.95}{\sigma}(\frac{W}{V})^2$

Methodology

I first cleared the workspace and command window. Then, I declared the variables rho and W with the values stated in the problem statement. Afterwards, I created a function for the drag force in terms of the velocity V, the density of the air between the flight altitude and sea level rho, and the weight W (1). Then I defined variables with values that are needed for the parabolic interpolation optimization method. These variables are the target approximation error (es), the maximum number of iterations (maxit), and the velocity vector with the three initial guesses for the minimum (Vguesses). The drag force function is also an input to the parabIntMin() function. Vguesses contains the values of the lower bound, the upper bound, and the average of the two. I had to use 1 instead of 0 for the lower bound because using 0 would result in a division by zero error in the drag force function. Changing the lower bound to 1 does not make a difference in the final output. I then used the parabIntMin() function to output the optimal velocity (Vmin), the minimum drag force (Fdmin), the velocity vector containing the three values used at the last iteration (last3V), the drag force vector corresponding the the velocity vector (last3Fd), the approximation error at the optimal velocity (ea), and the iteration number at the optimal velocity (iter). I then plotted the drag force function in the velocity range from 0 to 1,200 with increments of one using the plot() function. I also plotted the minimum drag force with the optimal velocity on the same graph. I also added labels, limits, a title, and a legend. To calculate the minimum drag force and the corresponding velocity of an airfoil with varying weights, I first declared the range of weights varying from 15,000 lbs to 20,000 lbs with an increment of 500. I then created a matrix with 2 rows and 11 columns (the amount of different weight values) using the zeros() function. Using a for-loop, I iterated through each weight value, and set the corresponding spot on the matrix to the minimized drag force and optimal velocity using the parabIntMin() function with that weight value. Once iterating through all of the weight values I created a scatter plot using the 1st row as values on the x-axis (velocity), and the 2nd row as values on the y-axis (drag force). I also added labels and a title. Lastly, I added comments to make my code easier to understand for other users.

Calculations and results

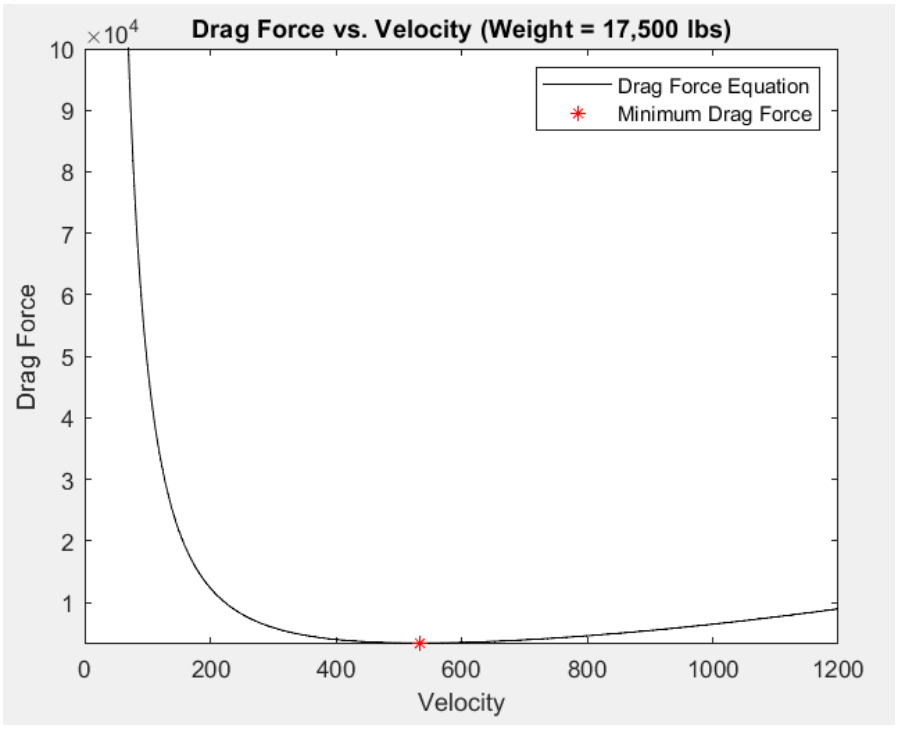
See next page.

M20 - Introduction to Computer Programming with MATLAB
Instructor: Prof. Enrique López Droguett, Ph.D.
Teacher Assistants: M. Fidansoy, G. San Martín, M. Pishahang, V. Vela.
Fall 2023 – UCLA
Student: *Alex Lie*
UCLA ID: *905901892*

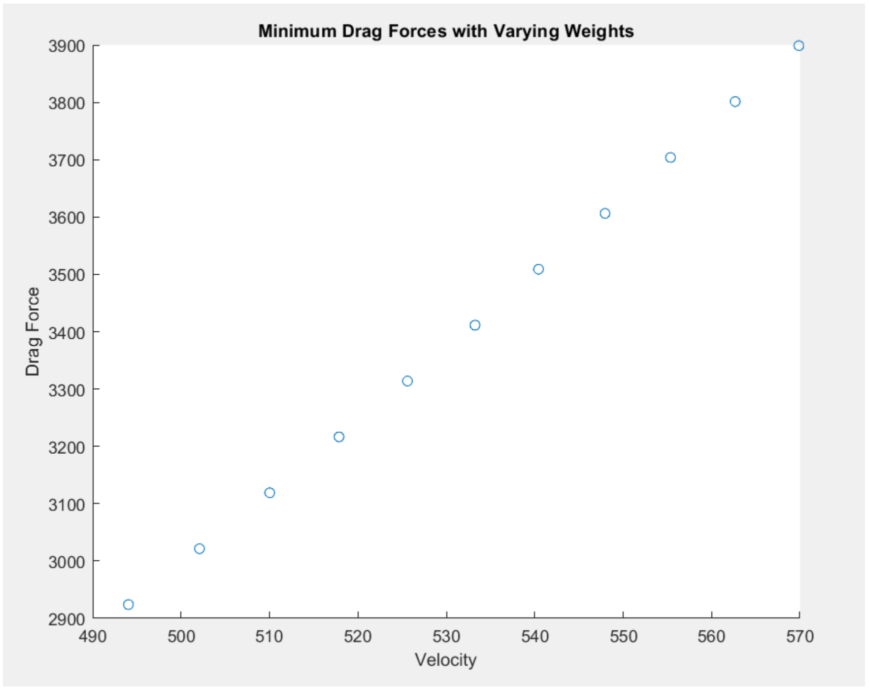
2)

```
Vmin =

   533.2461



Fdmin =

   3.4114e+03
```

3)



4)



The output shows the minimum drag force and the corresponding velocity at that drag force value. The first plot shows the drag force as a function of velocity of a 17,500 lb airfoil, with the red star marked as the minimum (the output). The second plot shows the different minimum drag forces and their corresponding velocities of airfoils of different weights.

<u>Discussions and Conclusions</u>

The optimized velocity and minimum drag force for the airfoil weighing 17,500 lbs is 533.2461 and 3411.4 respectively (2). Figure 3 shows that my results are accurate, since the red point (representing the minimum drag force) lies on the lowest point of the graph (you will need to zoom in to see this).

In figure 4, as the weight increases, the minimum drag force and the optimal velocity both increase. This change seems to be linear. Increasing the weight by 500 lbs seems to increase the optimized velocity by 7-8 and increase the drag force by 97.46. This is what I expected because looking at equation 1, if the weight increases, the drag force also increases. The velocity would need to be adjusted as well to find the new minimum drag force.

I also used the parabInterp() function, which creates a parabola using 3 velocity values and their corresponding drag force values. The inputs for the function are three velocity values and three drag force values. I created a parabola using the last three velocity and drag force values (last3V and last3Fd) obtained from the parabIntMin() function that was used as an approximation to find the minimum drag force and optimized velocity. This graph also runs through the red point. Although I excluded those lines of code from running, feel free to uncomment it and test it out!