Final Project Report

Salar Shoaib i200830 Section Y
Ammar Haider i200513 Section Y

Objective

The objective of this project was to automate solving a mathematical issue. The issue at hand was to create a propane tank in such a way that manufacturing its sides were of a different cost than manufacturing its middle part, while achieving a certain volume, and also while keeping the developing cost to its minimum.

Flow Chart/Code Flow

Added into the folder by the name:

i200513_Ammar_i200830_Salar_Y_Final_Project_Code_Flow.pdf

Matlab Commands

1. solve()
   Used to solve an equation, whether to get a numeric answer or to get an expression in terms of a variable
2. diff()
   Used to find the derivative of a function
3. abs()
   Used to get the absolute value
4. imag()
   Returns the imaginary part of the value
5. eps
   Gives the distance from 1.0 to the next largest double-precision number.
6. real()
   Returns the real part of an imaginary number
7. vpa()
   Converts the rational expressions into numeric (decimal) form
8. fplot()
   Plots the graph of a function
9. legend() / xlabel() / ylabel() / zlabel() / title()
   Provides legend on the plot, gives labels to the axis and gives a heading
10. grid on
    Turns on the grid on the graph

11. xlim([]) / ylim([])
    Gives the limits on which the graph is to be plotted

12. hold on / hold off
    Makes sure the previous plot remains on the graph, makes sure the previous plot no longer remains the next time the figure is made
13. fimplicit3()
    Produces the 3d shape formed by a 3d implicit function
14. MeshDensity
    A scalar value which decides the resolution of the function being plotted

How to Run the Program

A detailed example is given in the manual:

i200513_Ammar_i200830_Salar_Y_Final_Project_Manual.pdf

Conclusion

It is much easier to do a specified problem by hand, especially when you would need to hard code it. But going into the trouble of making a program which can handle user input values of costs and desired volume, and to be able to produce a visual representation of the tank created is a very futuristic approach.

Contributions

Salar Shoaib:

1. Handled the logical part. He explained what the program should do/produce in order to make it usable by using his problem solving skills.
2. Handled the logistics of producing the program – he made sure the resources for researching were present at all times.
3. He also did the debugging of the program, at least 20 times to produce a program which is deprived of errors

Ammar Haider:

1. Wrote down the code.

Difficulties

The objective of the problem was set as such that multiple fields were involved.

1. The task to make the program visually appealing on matlab was especially difficult because matlab is not really for visually pleasing program forming. To overcome this problem, we made sure our plots were interesting and useful enough that the user did not focus on anything else.

2. Once the program became so large with so many functions, it was difficult to keep track of where the bugs were occurring. For this, we commented the functions and parts of the program we knew were not giving errors and ran the program in the "Live Editor" which basically provides a real time run of the program
3. Once the goal was set to make the program general based, which could solve any problem, the way to tackle the issue became more difficult. Happily, the matlab documentation provides sufficient examples and functions which were used to resolve the issues.

Annexure A

```
disp('Y_i200513_Ammar Y_i200830_Salar')
disp('This Program was developed by Salar&Ammar Advisors inc.')
disp('Hello and welcome to Make Your Propane Tank')
disp('This Program will tell you the best way to make a capsule shaped
propane tank with the least amount of money!')
disp('Simply enter : ')
disp('1 the amount it costs you to make end peices of the tank')
disp('2 the amount it costs you to make the cylindrical part')
disp('3 the volume you want the tank to be')
input('Press enter to continue') %this simply lets the user press enter to
continue to the program
choice='y';
while(choice~='n')      %the code will loop till the user wants to stop
making more tanks
    %unit saves the unit of length being used
    unit=input('What unit are you using? (ft / cm / m) = ','s');
    %hemi saves the value of the cost of making the sides of the tank (which
are hemispherical in this case) input by the user
    ends_cost=input('Enter what it costs you to make the end parts (per unnit
square) = ');
    %cyl saves the value of the cost of making the cylindrical part of tank
    cyl_cost=input('Enter what it costs you to make one of the cylindrical
part (per unit square) = ');
    %volume saves the desired volume needed by user
    volume=input('Enter the volume you want to obtain = ');
    %r and l are symbolic variables used to solve the equations of volume and
surface area
    syms r l
    %volume of cylinder is declared as it is (pi x radius^2 * height)
    volume_of_cylinder=pi*(r^2)*l;
    %volume of spherer is declared as what the volume of a sphere would
produce (4/3 x pi x radius^3). the reason why we are taking volume of sphere
is because the two hemispheres at the end will add up to make one sphere
    volume_of_sphere=(4/3)*pi*r^3;
    %here the expression for total volume is made up
    total_volume=volume_of_cylinder+volume_of_sphere;
    %here, the expression for length is made with respect to r, making h the
subject where the total volume expression formed above is equal to the
desired volume of the user
    length=solve(total_volume==(volume),l);
    %here a function is made of cost and radius ( cost(r) ). the cost depends
on the area of the tank, more specifically, the area of cylinder mulitplied
```

```
by its contribution to cost and the area of the hemispheres multiplied by
their contribution
    cost=ends_cost*(4*pi*(r^2))+cyl_cost*(2*pi*r*length);
    %here, the TRUE magic happens. diff is a function which finds cost'(r)
and stores it in dC_dr
    dC_dr=diff(cost,r);
    %the roots of dC_dr are found so we can find the minimum of the cost. all
the possible (even non-real) roots are stored in roots_for_dC_dr
    roots_for_dC_dr=solve(dC_dr==0,r);
    %here, logical indexing is being done to fetch the real root (whether it
is negative or positive)
    idx=abs(imag(roots_for_dC_dr))<eps;
    real_root=real(roots_for_dC_dr(idx));
    %it is checked using the if statement if the smalest possible value for
radius is negative or not. (it is possibly negative if the cost of
manufacture for the end pieces is set to be smaller than the price of the
cylinderical part)
    %using if, the value for radius is converted to positive if needed
    if(real_root<0)
        solution_for_radius=vpa(real_root*-1);
    else
        solution_for_radius=vpa(real_root);
    end
%length was initially an expression of r. using subs() we substitute the
valye of r with real_root which is the required radius, and use vpa() to form
the rational number into a decimal number, understandable by the user
    solution_for_length=vpa(subs(length,r,solution_for_radius));



    %three symbolic functions are declared x,y,z
    %a cylinder function is created which is a function of x,y,z
    %by nature, the original formula created was to actually form a sphere
    %which is x^2 + y^2 + z^2 = r^2 where r is the radius of sphere
    %this modified equation simply takes the two hemispheres of a sphere
    %and stretches them outward, creating a cylinder in between
    %the modified equation is
    %x^2 + y^2 + 1/4(|z-(L/2)|+|z+(L/2)|-2L)^2 = r^2
    %where, L is the length of the cylindrical bit you want in the middle
    %in the function below, x has been swapped with z for better
    %orientation when displaying the interactive figure of the tank
    %and since matlab does not offer |X| functions
    %the value is first squared and then squarerooted so only positive
    %solutions are presented
    syms x y z
    cylinder= z^2 + y^2 + 1/4*( sqrt((x-(solution_for_length/2))^2) +
sqrt((x+(solution_for_length/2))^2) - 2*(solution_for_length/2) )^2 -
solution_for_radius^2;

%all the outputs are done after here
        %here all the graphs that might be needed by the user are ploted and
will be desplayed with useful information
        figure(1)
        fplot(length)   %plots graph of length with respect to change in
radius
        xlim([0 10])
        title('plot of length of cylinder with respect to radius')
```

```
        xlabel('radius')
        ylabel('length')
        grid on
        figure(2)
        fplot(cost,'r')   %plots cost(r)
        title('Change in cost with respect to radius (in red) and rate of
change of cost with respect to radius (blue)')
        hold on  %hold on so the graph of cost(r) and cost`(r) are on the
same figure
        fplot(dC_dr,'b')    %plots cost'(r) on the same figure
        legend('cost(r)','cost`(r)')
        xlabel('radius')
        ylabel('cost')
        xlim([0 10])
        grid on
        hold off  %hold off so the next time this figure is printed, the
privious plots are not shown
        figure(3)
        fs=fimplicit3(cylinder);   %the cylinder function is made a 3
dimensional implicit function and saved in another variable so changes can be
made into it
        fs.MeshDensity=60;
        title('interactive shape of the tank created')
        xlabel(unit)
        ylabel(unit)
        zlabel(unit)
        %here, the cost is being calculated by substituting the value of
        %radius caluclated into the cost(r) function
        final_cost=vpa(subs(cost,r,solution_for_radius));
        disp('optimum radius in the definded units : ')
        disp(solution_for_radius)   %printing the final optimal solution for
radius
        disp('optimum length of cylinder in the defined units = ')
        disp(solution_for_length)   %printing the final optimal solution for
length of cylinder
        disp('it will cost you: ')
        disp(final_cost)            %printing the final cost of making the
tank


    choice=input('Do you want to create a new tank with different inputs?
(Enter "y" for yes / "n" for no) = ','s');    %user makes choice
    while~(choice=='n'||choice=='y')    %choice validated by this loop
        choice=input('Invalid entry, re-enter (Enter "y" for yes / "n" for
no) = ','s');
    end
end
```

Hand Written Solution

<u>Solution</u> $V = 10\pi \text{ ft}^3$

Volume of sides collectively $= \frac{4\pi r^3}{3}$

Volume of cylinder $= \cancel{4}\pi r^2 \times l$

Area of sides collectively $= 4\pi r^2$

Area of cylinder $= 2\pi r \times l$

Total Volume $= \frac{4}{3}\pi r^3 + \cancel{4}\pi r^2 \times l$

$$\Downarrow$$

$$\boxed{10\pi = \frac{4}{3}\pi r^3 + \pi r^2 l}$$

Total area $= 4\pi r^2 + 2\pi r l$

$$\boxed{\text{Cost}(r) = 2\times(4\pi r^2) + 2\pi r(l)}$$

$$10\pi = 4\pi r^3 + 3\pi r^2 l$$

$$30\pi - 4\pi r^3 \overset{3}{=} 3\pi r^2 l$$

$$l = \frac{30\pi - 4\pi r^3}{3\pi r^2} \qquad \boxed{l = \frac{30 - 4r^3}{3r^2}}$$

$$\text{Cost}(r) = 2\times(4\pi r^2) + 2\pi r\left(\frac{30 - 4r^3}{3r^2}\right)$$

$$\text{Cost}(r) = 2\pi\left(4r^2 + \frac{\overset{10}{\cancel{30}}}{\cancel{3}r} - \frac{4}{3}r^2\right)$$

$$\text{Cost}'(r) = 2\pi\left(8r + \frac{(-1)(10)}{r^2} - \frac{8}{3}r\right)$$

$$\text{Cost}'(r) = 2\pi\left(\left(\frac{-10}{r^2}\right) + \frac{16}{3}r\right) \qquad \text{for } \text{Cost}'(r) = 0$$

$$0 = 2\pi\left(\frac{-10}{r^2} + \frac{16}{3}r\right) = -\frac{10}{r^2} + \frac{16r}{3}$$

$$\frac{10}{r^2} = \frac{16r}{3}$$

$$\sqrt[3]{\frac{30}{16}} = \sqrt[3]{r^3} \qquad \boxed{r = 1.233 \text{ ft}}$$

$$l = \frac{30 - 4r^3}{3r^2}$$

$$l = \frac{30 - 4(1.233)^3}{3(1.233)^2} \qquad \boxed{l = 4.934 \text{ ft}}$$

Forming the propane tank with radius = 1.233 ft and length of cylinder as 4.934 ft will give the smallest production cost.