CS/ECE/ISyE 524 — Introduction to Optimization — Spring 2021

Final Course Project: Due 5/2/21, 12:05pm

Optimal Cup

Jaskaran Bakshi (jsbakshi@wisc.edu)

Table of Contents

1. Introduction

This project aims to optimize the everyday Cup to achieve an individual's daily intake of water of 3.4 L in a day by creating a Cup that I would have to drink 3 times throughout the day. The motivation for the 3 was to allow people to drink the cup one for every meal to make it easier to remember to drink water. In addition to this, I want to consider the material cost for creating the optimal cup using different types of materials like plastic, stainless steel, to measure the environmental impacts of such a cup.

- Currently 8.25 million tons of plastic waste enter the ocean each year.
- PLASTIC LIDS PRODUCE AN ESTIMATE OF OVER 20,000 TONS OF WASTE EACH YEAR.
  In today's environment, big impacts happen through small changes over a large scale.
- We use an estimated 16 billion disposable coffee cups each year.
- Plastic lids often do not get recycled and, in turn, pollute our oceans. Straws have previously garnered widespread attention due to their impact on ocean pollution, but in fact, plastic lids make up to 8 times more of ocean waste than straws do.

Above are various statistical facts of cups in today's world. Humans fail to consider how much the effect of buying disposable cups affects the overall environment. Because of this, I was encouraged to re-think how our everyday lives are structured because I believe that there needs to be a change that makes a big impact on the overall way we think about cups. For more information about these facts please take a look at the citations at the bottom of this block.

The problem I am trying to solve in this project is the issue of wasted plastic cups, along with regular college students not drinking their required amount of daily water within a day. With this in mind, I set out to create a multi-purpose mug that is not only eco-friendly because of the current climate change scenario but has a purpose for each individual to keep their water intake a constant balance. These 2 impacts, make it clear to me that it is important to improve upon the everyday mug and make it an object that considers these two impacts rather than simply the cost of the cup. The main way I am going about comparing these cups is by making sure the usage of each cup exceeds the overall cost of the cup. This means if I buy a reusable cup I am only required to purchase it one time, but for a reusable cup, the cost of using it constantly increases the more cups an individual buys. And the more cups bought, the more that will be disposed.

The data that is being used in this project has been collected over the internet by looking at various numbers of coffee mugs and cups in stores and collecting the dimensions of multiple mugs to ensure that our mug has the same proportions. In particular, I found this blog post online that I took into consideration where the individual collected different data points of many types of mugs and cups and averaged the dimensions over all of the items. This gave me a good base to set up the cup I am creating by using the ratio between the height and radius of the cup, the ratio between the height of the cup to the major height of the half-torus, and the ratio between the major radius and minor radius of the half-torus(more information on torrus). The other information I had to collect was the different materials that cups can be made with along with the average cost of buying the particular material depending on the surface area.

The rest of the notebook is as follows:

1. Mathematical model: Here I dive into the different equations I used and manipulated to have my optimized cup including the equation of a Cylinder and Half-Torus. I also dive into how I cleaned the data collected over the internet so that I could use it using my model at a base format of cm for the measurements of each side of the cup, cm^2 is used for the surface area of the cup, and cm^3 for the overall volume of the cup
2. Solution: This section dives into the actual code used to optimize the problem described above. I describe the type of optimizer I am using along with all the collected data. In the end, we can see an optimal cup that has been created with the various variables printed out.
3. Result and Discussion: This part of the notebook will highlight the results from the solution. I dive into the different costs of each cup that can be created using the different materials I am considering and explaining the importance of these prices by comparing the re-usable materials cost with the common disposable cup. There is also a subsection where I highlight some extra python code I wrote using the Blender 3D modeling software to generate the actual cup from the optimization solution in section 3.

Citation:

1) foodprint

2) greenmatch

3) the-standard-coffee-mug-dimensions%3A%2082mm,Handle%20height%20(d)%3A%2073mm)

2. Mathematical model

The mathematical model described in this section assumes a few things, these assumptions come mainly from physics and how the dimensions of a cup are measured:

1. The standard unit for this model will be in cm for all measurements
2. The other assumption is that we are trying to maintain the same ratios of a normal cup to this optimized cup

Daily water intake for a male is:

- 3.7 L = 3700 ml

Cylinder equations:

- Volume =πr^2h
- Surface Area =2πrh+2πr^2

Half-Torrus equations:

- Volume =(πr^2)(πR)
- Surface Area =(πR)(2πr)

The equations above are the main way we are going to measure the outcome of the optimal cup. The volumes of the objects will be used to ensure that we are reaching the total water intake of 1,233.33... (3700/3). The surface area equations will be used with the cost of the material.

Decision for material: the type of cups that are are used on a day to day basis are the following:

Material	$\$/ft^2$	$\$/cm^2$
ceramic	0.50	0.0005381957526
glass	25.00	0.02690978763
stainless steel	18.25	0.01965490888
Paper	0.60	0.0006458349031
styrofoam	0.44	0.0004736122623
plastic	6.20	0.006673627332

Most manufacturers for drink wear use 304-grade stainless steel which is the cost in the above table. As we can see the data collected online was in $\frac{cost}{ft^2}$ which is unrealistic for a cup, so I had to convert the data into $\frac{cost}{cm^2}$ to ensure consistency. The conversion from $\frac{cost}{ft^2}$ to $\frac{cost}{cm^2}$ is the following:

$$\frac{x}{ft^2} * \frac{1ft^2}{929.03cm^2} = \frac{x}{929.03cm^2}$$

Used the blog information as a guide for my cup to determine simple ratios between the height and the width of a standard mug because otherwise when we try to generate a cup based on the equations of a Cylinder and a Half-Torus model will not create a realistic cup that is drinkable. The following aspect ratios were used when creating this cup.

Material	Ratio
Height to Width	$\frac{41}{95}$
Height to Handle Height	$\frac{73}{95}$
Inner Handle Radius to Handle Height	$\frac{14}{45}$

Decision Variables:

- h - This is the overall height of the optimal cup or the height of the cylinder.
- r - This is the radius of the cylinder for the optimal cup.
- r_{major} - This variable will be the major radius of the overall Half-Torus.
- r_{minor} - This variable will be the minor radius of the overall Half-Torus.
- S - This variable is used as the min-max variable to maximize the surface area of the cup while minimizing cost.

Constraints:

The linear constraints in this model will be the different aspect ratios I am taking into consideration to make the cup look "normal". these ratios can be expressed into linear equations for the different variables it is comparing. For the linear constraints in this model we used the following matrix:

$$A = \begin{bmatrix} -\frac{41}{95} & 1 & 0 & 0 \\ \frac{41}{95} & -1 & 0 & 0 \\ -\frac{73}{190} & 0 & 1 & 0 \\ \frac{73}{190} & 0 & -1 & 0 \\ -\frac{14}{45} & 0 & 0 & 1 \\ \frac{14}{45} & 0 & 0 & -1 \end{bmatrix}$$

$$c = \begin{bmatrix} h \\ r \\ r_{major} \\ r_{minor} \end{bmatrix}$$

Thus, using the linear combinations from above we can create the base constraint for all the aspect ratios using the equation:

$$A * c^T \le 0$$

The next constraints are all Non-Linear constraints because they are measuring the surface area and volume of the combined objects:

$$\pi^2 * r_{major} * r_{minor}^2 + \pi * r_{minor}^2 * h - \frac{3700ml}{3} \le 0$$
$$\pi^2 * r_{major} * r_{minor}^2 + \pi * r_{minor}^2 * h + \frac{3700ml}{3} \le 0$$
$$S - (2 * \pi * r^2 + 2 * \pi * r * h + 2 * \pi^2 * r_{major} * r_{minor}) \le 0$$

The first 2 equations are measuring the combined volume of the cylinder and half-torus and making sure the value is less than or equal to $\frac{3700ml}{3}$.

The last equation is where I am calculating the surface area of the combined object of the cylinder and half-torus. we are trying to maximize the decision variable of S so that we can use the maximized surface area of the cup to check the cost of each material using the dimensions found.

Objective Function:

$$\text{minimize} \sum_{i=1}^{n} X_i$$

$$X_i = S * a_i$$

$$a = \begin{bmatrix} 0.0005381957526 & 0.02690978763 & 0.01965490888 & 0.0006458349031 & 0.0004736122623 & 0.006673627332 \end{bmatrix}$$

The object equations above show how we are using the maxed out surface area of our cup, S, and the different cost of each material we found in the table above to figure the minimized total cost for all the cups. The reason I do this is that we want to find the minimized cost for each cup, so we can take the sum over all the different cups and minimize that as a whole to find out the object to our problem.

Model used: NLP while solving a min-max problem

$$\begin{aligned} \underset{h,r,r_{major},r_{minor},S}{\text{minimize}} \quad & \sum_{i=1}^{n} X_i \\ \text{subject to:} \quad & A * c^T \le 0 \\ & -c \le 0 \\ & \pi^2 * r_{major} * r_{minor}^2 + \pi * r_{minor}^2 * h - \frac{3700ml}{3} \le 0 \\ & \pi^2 * r_{major} * r_{minor}^2 + \pi * r_{minor}^2 * h + \frac{3700ml}{3} \le 0 \\ & S - (2 * \pi * r^2 + 2 * \pi * r * h + 2 * \pi^2 * r_{major} * r_{minor}) \le 0 \\ \text{With NL expressions:} \quad & X_i = S * a_i \end{aligned}$$

$$\text{Where:} \quad A = \begin{bmatrix} -\frac{41}{95} & 1 & 0 & 0 \\ \frac{41}{95} & -1 & 0 & 0 \\ -\frac{73}{190} & 0 & 1 & 0 \\ \frac{73}{190} & 0 & -1 & 0 \\ -\frac{14}{45} & 0 & 0 & 1 \\ \frac{14}{45} & 0 & 0 & -1 \end{bmatrix}$$

$$c = \begin{bmatrix} h \\ r \\ r_{major} \\ r_{minor} \end{bmatrix}$$

$$a = [0.0005381957526 \quad 0.02690978763 \quad 0.01965490888 \quad 0.0006458349031 \quad 0.0004736122623 \quad 0.006673$$
$$n = 6$$

3. Solution

This is the setup code, a lot of the research done for this project is written in this block of code where we have the different materials with their respective average cost per cm^2. The aspect ratios that I found online for different types of mugs are included in this as well. The daily water intake of 3.7 L (3700 mL) is included in this as well.

In [8]:
```
# cup is measured in centimeters becasue cm^3 = 1 ml
material = ["ceramic", "glass", "stainless steel", "Paper", "styrofoam", "plastic"]
# stainless steel avg - 0.1614587258
# lower cost - 0.08611132041
avg_cost = [0.0005381957526 , 0.02690978763, 0.01965490888 , 0.0006458349031, 0.00047361226223, 0.006673627332] #per sqr foot convert to per sqr centimeter
# enviormental cost of producing each of these materials

# make a handle of the cup with an empty volume using the equations of a half torus
ration_hieght_to_handle_hieght = 73/95
radius_to_length_ratio = 14/45
hight_to_width_ratio = 41/95 # height to width raio of a mug must stay constant when creating this mug

daily_water_intake = 3700
```

Out[8]: 3700

For this optimization problem, I used the Ipopt optimizer because many non-linear constraints need to be set up as part of this problem. I have also set up the radius as r, height as h, the minor radius of the torus as a handle, and the major radius of the torus as R. The last variable surf will be used as a min-max variable to maximize the surface area of the overall object while

of all the cups is minimized which is minimizing all the cups at once. This way we know the price of each material cup.

In [5]:
```
@NLexpression(m, cost[i in 1:6], surf*avg_cost[i])
@NLobjective(m, Min, sum(cost[i] for i in 1:6))

optimize!(m)
```

```
******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Eclipse Public License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.13.2, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:       25
Number of nonzeros in Lagrangian Hessian.............:       18

Total number of variables............................:        5
                     variables with only lower bounds:        4
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        9
        inequality constraints with only lower bounds:        5
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        4

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  0.0000000e+00 1.23e+03 1.19e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1r 0.0000000e+00 1.23e+03 9.99e+02   3.1 0.00e+00    -  0.00e+00 2.51e-07R  6
   2r 5.1400400e-01 9.03e+02 3.78e+04   3.1 1.22e+03    -  8.02e-03 8.13e-03f  1
   3r 5.1477599e-01 8.41e+02 3.88e+04   2.4 5.20e+00   4.0 1.35e-01 6.86e-02h  1
   4r 5.1753717e-01 4.45e+02 2.72e+04   2.4 5.55e+00   3.5 4.90e-01 4.66e-01h  1
   5  1.0632520e+00 4.37e+02 2.87e+00  -1.0 6.80e+02    -  5.13e-01 1.75e-02h  1
   6  3.0441659e+01 2.32e+02 1.30e+00  -1.0 1.22e+03    -  5.77e-01 6.31e-01h  1
   7  3.9079638e+01 5.10e+01 7.28e-01  -1.0 4.51e+02    -  6.05e-01 6.25e-01h  1
   8  3.8877113e+01 4.24e+01 6.55e-01  -1.0 1.69e+02    -  3.93e-01 1.83e-01h  1
   9  3.8385727e+01 3.98e+01 5.90e-01  -1.0 2.04e+02    -  3.39e-02 4.40e-02h  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10  4.0734603e+01 8.26e+00 2.40e-01  -1.0 1.32e+02    -  1.00e+00 8.31e-01h  1
  11  4.0653019e+01 6.16e+00 4.50e+00  -1.0 2.23e+01    -  1.00e+00 2.71e-01h  1
  12  4.0816709e+01 1.39e+00 1.73e+00  -1.0 1.62e+01    -  1.00e+00 7.89e-01h  1
  13  4.0819638e+01 7.17e-01 1.02e+01  -1.0 3.41e+00    -  1.00e+00 5.28e-01h  1
  14  4.0834362e+01 3.40e-01 2.06e+01  -1.0 1.59e+00    -  1.00e+00 6.09e-01h  1
  15  4.0834659e+01 1.83e-01 5.46e+01  -1.0 5.91e-01    -  1.00e+00 5.76e-01h  1
  16  4.0836399e+01 9.19e-02 1.28e+02  -1.0 2.12e-01    -  1.00e+00 5.90e-01h  1
  17  4.0835928e+01 4.01e-02 3.13e+02  -1.0 6.15e-02    -  1.00e+00 5.84e-01h  1
  18  4.0835956e+01 3.41e-02 1.55e+03  -1.0 2.36e-02    -  1.00e+00 1.47e-01f  3
  19  4.0835992e+01 7.76e-03 7.64e+02  -1.0 1.99e-02    -  1.00e+00 7.69e-01h  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  20  4.0835982e+01 6.74e-03 7.25e+03  -1.0 4.66e-03    -  1.00e+00 1.34e-01f  3
  21  4.0835950e+01 1.61e-03 3.78e+03  -1.0 3.96e-03    -  1.00e+00 7.64e-01h  1
  22  4.0835952e+01 1.39e-03 3.54e+04  -1.0 9.53e-04    -  1.00e+00 1.33e-01f  3
  23  4.0835953e+01 3.23e-04 1.83e+04  -1.0 8.09e-04    -  1.00e+00 7.65e-01h  1
  24  4.0835953e+01 2.80e-04 1.70e+05  -1.0 1.95e-04    -  1.00e+00 1.34e-01f  3
  25  4.0835951e+01 6.44e-05 8.40e+04  -1.0 1.65e-04    -  1.00e+00 7.74e-01h  1
  26  4.0835951e+01 5.53e-05 7.72e+05  -1.0 3.83e-05    -  1.00e+00 1.39e-01f  3
  27  4.0835951e+01 9.83e-06 2.95e+05  -1.0 3.22e-05    -  1.00e+00 8.18e-01h  1
  28  4.0835951e+01 6.69e-06 2.16e+06  -1.0 6.25e-06    -  1.00e+00 3.31e-01f  2
  29  4.0835951e+01 5.44e-08 5.49e+04  -1.0 3.87e-06    -  1.00e+00 9.88e-01h  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  30  4.0835951e+01 6.94e-08 3.20e+06  -1.0 6.98e-07    -  1.00e+00 2.67e-02f  3
  31  4.0835951e+01 3.20e-09 1.00e-06  -1.0 6.96e-08    -  1.00e+00 1.00e+00h  1
  32  4.0735955e+01 0.00e+00 2.39e-07  -8.6 1.82e+00    -  1.00e+00 1.00e+00f  1
  33  4.0735953e+01 0.00e+00 9.74e+01  -8.6 3.32e-05    -  9.98e-01 1.00e+00h  1
  34  4.0735953e+01 8.27e-16 2.82e+00  -8.6 3.82e-07    -  8.25e-01 1.00e+00h  1
  35  4.0735953e+01 0.00e+00 3.93e-01  -8.6 2.52e-07    -  8.41e-01 1.00e+00f  1
  36  4.0735953e+01 0.00e+00 3.52e-14  -8.6 1.91e-07    -  1.00e+00 1.00e+00h  1

Number of Iterations....: 36

                                   (scaled)                 (unscaled)
Objective...............:   4.0735953053714667e+01    4.0735953053714667e+01
Dual infeasibility......:   3.5202365248052163e-14    3.5202365248052163e-14
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   3.5020045025860792e-09    3.5020045025860792e-09
Overall NLP error.......:   3.5020045025860792e-09    3.5020045025860792e-09


Number of objective function evaluations             = 56
Number of objective gradient evaluations             = 35
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 56
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 38
Number of Lagrangian Hessian evaluations             = 36
Total CPU secs in IPOPT (w/o function evaluations)   =      2.395
Total CPU secs in NLP function evaluations           =      0.620

EXIT: Optimal Solution Found.
```

This last piece of the optimization is printing out the optimized values in a neat fashion and the overall optimized model is below this block.

In [6]:
```
println("radius of the optimal cup is: ", value.(r))
println("hieght of the optimal cup is: ", value.(h))
println("maximum surface area of the cup is: " ,value.(surf))
println("the minimum cost of the each material cup: ", value.(cost))
println("the radius of the hollow handle is: ", value.(handle))
println("the overall radius of the cupe handle: ", value.(R))
println("the overall volume of the cup is: ", π^2 *value.(R) * value.(handle)^2  + π*value.(r)^2*value.(h))
println("price of all cups: ", objective_value(m))
```

```
radius of the optimal cup is: 5.372516525942034
hieght of the optimal cup is: 12.448513882574991
maximum surface area of the cup is: 742.0573032588791
the minimum cost of the each material cup: [0.39937208879973884, 19.968604439986944, 14.585068679291794, 0.4792465065448455, 0.3514474381526749, 4.952213900938668]
the radius of the hollow handle is: 1.487997787724594
the overall radius of the cupe handle: 4.782850061977391
the overall volume of the cup is: 1233.3333333328949
price of all cups: 40.73595305371467
```

In [7]:
```
m
```

Out[7]:

$$\begin{aligned} \min \quad & subexpression_1 + subexpression_2 + subexpression_3 + subexpression_4 + subexpression_5 + subexpression_6 \\ \text{Subject to} \quad & -r + 0.43157894736842106h \ge 0.0 \\ & 0.38421052631578945h - R \ge 0.0 \\ & -handle + 0.3111111111111111R \ge 0.0 \\ & -r + 0.43157894736842106h \le 0.0 \\ & 0.38421052631578945h - R \le 0.0 \\ & -handle + 0.3111111111111111R \le 0.0 \\ & r \ge 0.0 \\ & h \ge 0.0 \\ & handle \ge 0.0 \\ & R \ge 0.0 \\ & (3.141592653589793^{2.0} * R * handle^{2.0} + 3.141592653589793 * r^{2.0} * h) - 3700.0/3.0 \ge 0 \\ & (3.141592653589793^{2.0} * R * handle^{2.0} + 3.141592653589793 * r^{2.0} * h) - 3700.0/3.0 \le 0 \\ & surf - (2.0 * 3.141592653589793 * r^{2.0} + 2.0 * 3.141592653589793 * r * h + 2.0 * 3.141592653589793^{2.0} \\ \text{With NL expressions} \quad & subexpression_1 : surf * 0.0005381957526 \\ & subexpression_2 : surf * 0.02690978763 \\ & subexpression_3 : surf * 0.01965490888 \\ & subexpression_4 : surf * 0.0006458349031 \\ & subexpression_5 : surf * 0.0004736122623 \\ & subexpression_6 : surf * 0.006673627332 \end{aligned}$$

4. Results and discussion

Cup Dimensions:

Radius	5.372516525942034 cm
Hieght	12.448513882574991 cm
Minor Radius of Handle	1.487997787724594 cm
Major Radius of Handle	4.782850061977391 cm
Surface Area	742.0573032588791 cm^2
Volume	1233.3333333328949 cm^2

The cup dimensions from above are interesting because it maintains such an elegant ratio between all the different dimensions, while the volume of 1233.3333333328949 * 3 = 3,699.9999999987 which is approximately the required amount of water 3700 mL. To visualize this cup I modeled the output of my optimization problem. I used Blender python to generate a 3-D plot of the cups using the dimensions above and created a video. Look at the sub-section for more information.

Cost of each material cup:

Material	Cost of Cup
ceramic	0.40
glass	19.97
stainless steel	14.57
Paper	0.48
styrofoam	0.35
plastic	4.95

- From the table, we would need to buy 42 disposable styrofoam cups before reaching the cost-effectivity of a stainless steel cup.
- From the table, we would need to buy 31 disposable paper cups before reaching the cost-effectivity of a stainless steel cup.
- From the table, we would need to buy 42 disposable paper cups before reaching the cost-effectivity of a glass cup.
- From the table, we would need to buy 58 disposable styrofoam cups before reaching the cost-effectivity of a glass cup.

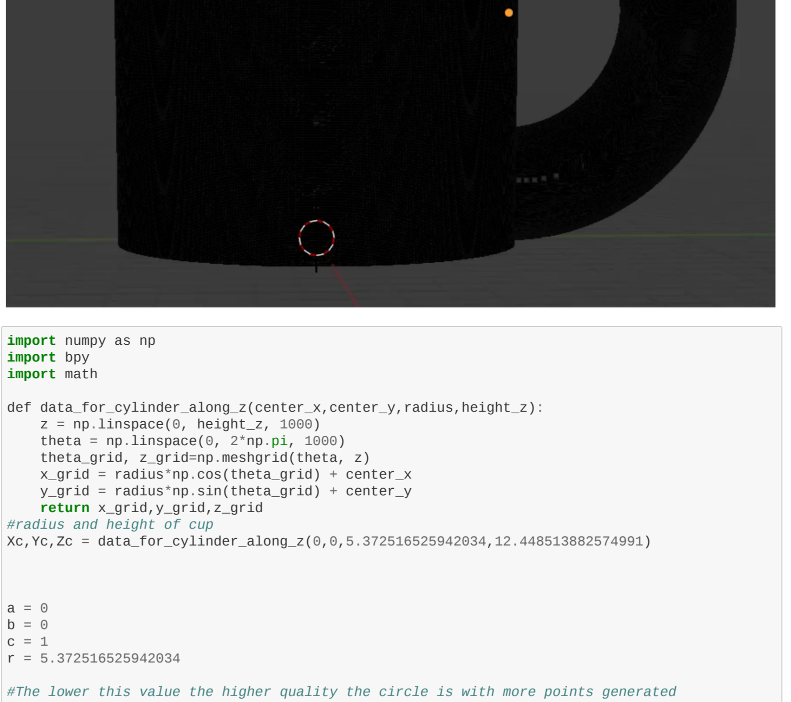
Overall we can see that the proportions of the cups have been met and the prices of the cups are relatively accurate. Comparing to the prices online for similar-sized cups results in the prices we see above which is expected. What is interesting is that the overall volume of the cup has been diversified because of the hollow handle that is created to hold more liquid than a regular mug that normally has a solid handle. The result of this optimization problem shows us that with the same cup using different material, even when the cost is lower for disposable material like paper, styrofoam but we end up finding out there is a higher environmental cost because we have to buy so many of the 1-use cups throughout the day, whereas if we buy a re-usable cup-like glass, stainless steel or plastic we end up saving money at a certain number of disposable cups bought. This is an interesting find because it would assume that if this optimized cup is created then we can expect to use a re-usable material like stainless steel, and people have the added feature of being able to know exactly how much they are drinking for their daily intake.

limitations

- how I went about forming the actual cups with the aspect ratios because it constricts the cup so that there are no multiple versions of the cup so losing this constrain may result in a wide array of cups that could be made.
- I am never actually trying to form an object in 3D space, rather I use the different variables like height or ratio to make a representation of a cup, but it could be better to consider optimizing the points instead.
- When making this cup the empty half-torus will be hard to create because it is an empty handle that is meant to hold liquid but the physical manifestation of such an object is beyond my scope.
- When optimizing this cup we dont consider the thickness of the material into our equation because for the purposes of a design this cup can simply be a base design to add padding depending on the side.

4.A. 3D Blender Model

Please check out the youtube video for a 3D model that I generated using blender-python and the various information that was provided to me through my optimization equations. The code I used to generate the blender cup object is set up below but will not work in this environment as it requires the addition of Blender as an addon. I used the radius, high, minor/ major radius of the optimization problem to graph the functions in 3D space, this is for a visual representation of the optimal cup.



In [ ]:
```
import numpy as np
import bpy
import math

def data_for_cylinder_along_z(center_x,center_y,radius,height_z):
    z = np.linspace(0, height_z, 1000)
    theta = np.linspace(0, 2*np.pi, 1000)
    theta_grid, z_grid=np.meshgrid(theta, z)
    x_grid = radius*np.cos(theta_grid) + center_x
    y_grid = radius*np.sin(theta_grid) + center_y
    return x_grid,y_grid,z_grid
#radius and height of cup
Xc,Yc,Zc = data_for_cylinder_along_z(0,0,5.372516525942034,12.448513882574991)



a = 0
b = 0
c = 1
r = 5.372516525942034

#The lower this value the higher quality the circle is with more points generated
stepSize = 0.1

n = 1000

# theta: poloidal angle; phi: toroidal angle
theta = np.linspace(0, 2*np.pi, n)
phi   = np.linspace(0, np.pi, n)
theta, phi = np.meshgrid(theta, phi)

# R0: major radius; a: minor radius
R0, a = 4.782850061977391, 1.487997787724594

# torus parametrization
x = (R0 + a*np.cos(theta)) * np.cos(phi)
y = (R0 + a*np.cos(theta)) * np.cos(phi)
z = a * np.sin(theta)

half_torrus_points = []
cylinder_points = []

for i in range(len(x)):
    for j in range(len(x[i])):
        half_torrus_points.append([x[i][j],y[i][j],z[i][j]])
for i in range(len(Xc)):
    for j in range(len(Xc[i])):
        cylinder_points.append([Xc[i][j], Yc[i][j], Zc[i][j]])

circle = []
for x in range(-11,11):
        for y in range(-11,11):
            if math.pow(x,2) + math.pow(y,2) < math.pow(r,2):
                circle.append([x,y,1])

print(half_torrus_points)
vertices = half_torrus_points
edges = []
faces = []
new_mesh = bpy.data.meshes.new('Half Torrus')
new_mesh.from_pydata(vertices, edges, faces)
new_mesh.update()
# make object from mesh
new_object = bpy.data.objects.new('Half Torrus', new_mesh)
# make collection
new_collection = bpy.data.collections.new('torrus_collection')
bpy.context.scene.collection.children.link(new_collection)
# add object to scene collection


cylinder_vertices = cylinder_points
cylinder_edges = []
cylinder_faces = []
cylinder_mesh = bpy.data.meshes.new('Cylinder')
cylinder_mesh.from_pydata(cylinder_vertices, cylinder_edges, cylinder_faces)
cylinder_mesh.update()
# make object from mesh
cylinder_object = bpy.data.objects.new('Cylinder', cylinder_mesh)
# make collection
cylinder_collection = bpy.data.collections.new('Cylinder_collection')
bpy.context.scene.collection.children.link(cylinder_collection)


new_collection.objects.link(new_object)
# add object to scene collection
cylinder_collection.objects.link(cylinder_object)


circle_vertices = circle
circle_edges = []
circle_faces = []
circle_mesh = bpy.data.meshes.new('circle')
circle_mesh.from_pydata(circle_vertices, circle_edges, circle_faces)
circle_mesh.update()
# make object from mesh
circle_object = bpy.data.objects.new('circle', circle_mesh)
# make collection
circle_collection = bpy.data.collections.new('circle_collection')
bpy.context.scene.collection.children.link(circle_collection)

circle_collection.objects.link(circle_object)
```

5. Conclusion

In conclusion, I made an optimal cup for a specific size and now, I can make this sized cup or buy a cup with those dimensions and always know what it represents. I also know the significant impact of each material cup and its overall cost to a consumer and the hidden cost behind single-use cups compared to re-usable cups.

One possible future direction for this study could be to take a dataset of all the mugs and compare their relative sizes to the ones I have created to incorporate better proportions. Because I based my information on a blog post that did an average on a dataset they created, incorporating that dataset within this code for the proportions would be an interesting route to take to see if the cup can be more proportionally sound to everyday cups we use.