% COMP 1012 Winter 2020 -- Assignment 3 and Assignment 4

COMP 1012 Summer 2020 -- Assignment 3

Instructors: Sheikh Jubair (A01)

Due on **Wednesday, August 12th at 11:30 pm**.

Outcomes

- Numpy
- Functions

Description

In this assignment, you are going to calculate the length and surface area of a solid generated by a function.

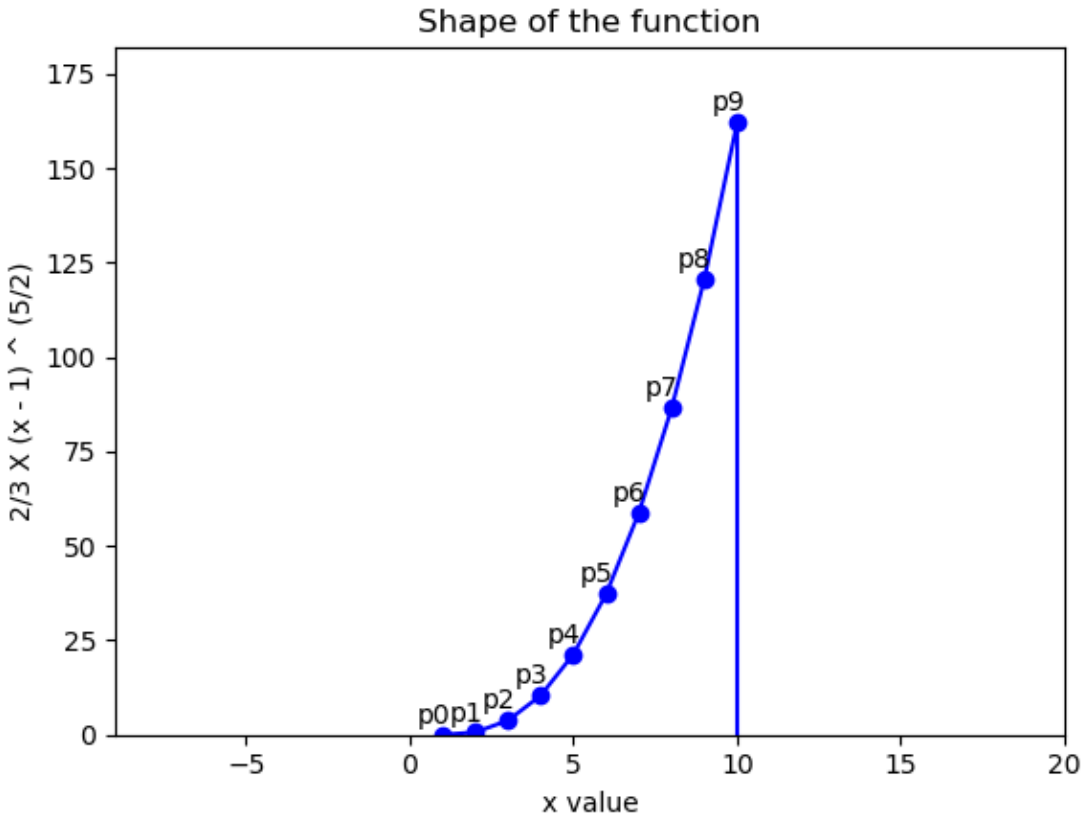
You should use numpy where possible to do all the calculation. You also need to write the code in a numpy way (vectorization) which means you should not use loops if it is possible to write the program without using loops with numpy and you should use numpy arrays instead of a lists.

You need to follow the programming standards for this assignment. You will find more about programming standards on umlearn. Please go to the course page >> Content >> Additional Files >> Programming Standards.pdf

You **can't use** any libraries except **time, numpy**.

Question 1: Required

In this assignment, we are going to find the length and area under a curve. The curve that you will consider in this assignment is given by the (single variable) function $y = f(x) = \frac{2}{3}(x-1)^{\frac{5}{2}}$. So for each value of x, we will get a corresponding y value. For example, If we have 10 values of x, we will get 10 values for y. Now, if we plot these values, we will get a figure like the one we showed here.



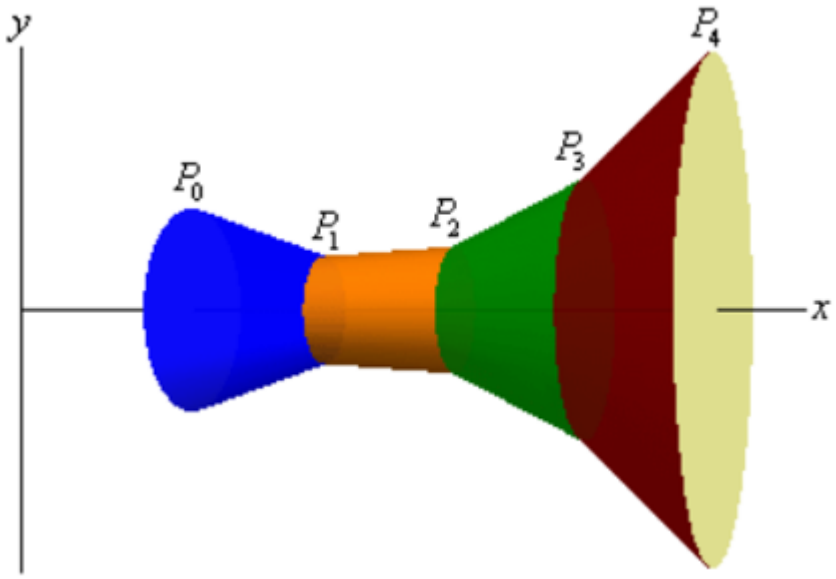
To approximate the length of the curve, we can consider the curve between two neighboring points is a straight line. For example, we can consider $p0$ and $p1$ a straight line and then $p1$ and $p2$ a straight line and so on. As the two neighboring points are straight line, we need to calculate the Euclidean distance between the two neighboring points and then take the summation of those distances. For example, suppose the Euclidean distance between $p0$ and $p1$ is $d0$, p1 and $p2$ is $d1$, ... and $p8$ and $p9$ is $d8$. The length will be $length = d0 + d1 + \ldots + d8$. So, if we have n points, the length of the curve can be approximated by:

$$length = \sum_{i=1}^{n-1} |p_{i+1}, pi|$$

Where $|p_{i+1}, p_i|$ indicates the Euclidean distance between two points. The Euclidean distance can be calculated using the following formula:

$$|p_{i+1}, p_i| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

The surface area of the solid obtained when you revolve the curve $y = f(x)$ around the x-axis can be approximated in a way that is similar to the way we approximated the length of the curve. We can approximate the surface area of a small part of the solid and then take the summation of those small parts to finally get the approximate surface area. This can be compared to [frustums] [tutorial.math.lamar.edu/Classes/CalcII/SurfaceArea.aspx]. Frustums look like the figure below.

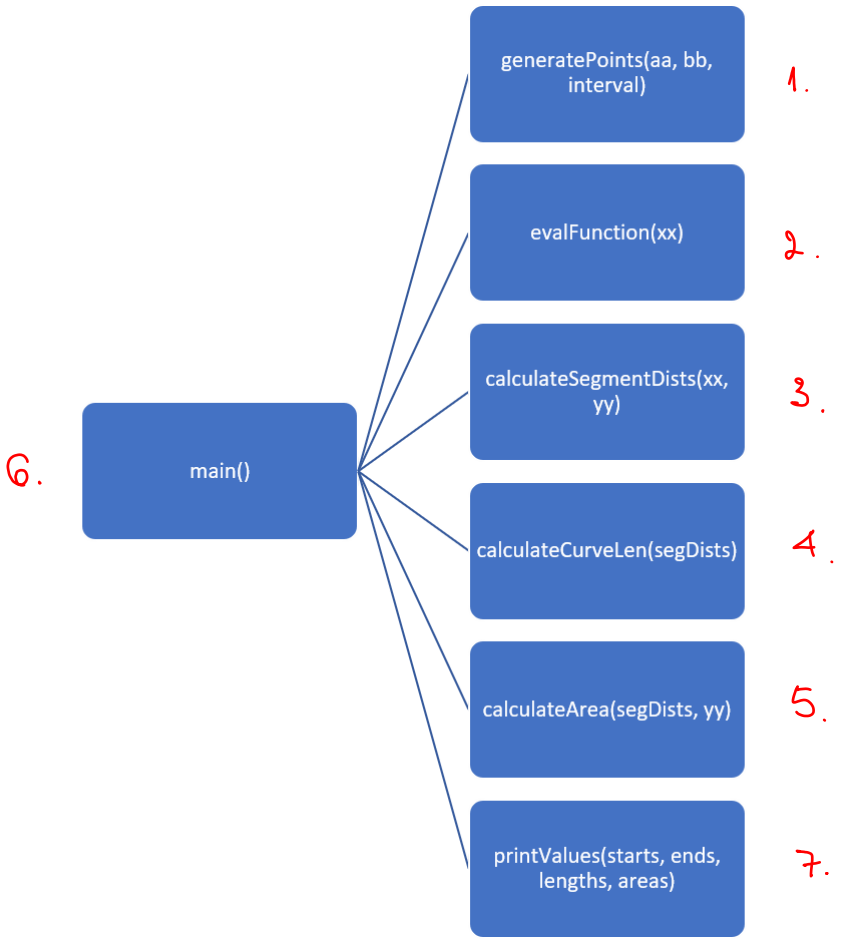


The surface area of a frustum: $\text{surface area} = 2\pi rl$. This equation can be rewritten as:

$area = \sum_{i=1}^{n-1} 2\pi(\frac{f(x_i)+f(x_i+1))}{2})|p_i, p_{i+1}|$

Here, p_is are the points that we used when calculating the length.

What to do:



In your program, you need to implement the following functions.

1. **generatePoints($aa, bb, interval$):**

   This function will generate points using **arange** function where aa is the start and bb is the end and $interval$ is the step.

2. **evalFunction(xx):**

   This function takes one parameter as input. The parameter xx is an array of numbers. It implements the curve function $y = f(x)$ mentioned in the description and returns the array.

3. **calculateSegmentDists(xx, yy):**

   This function takes two numpy array as input. The values in the xx array indicate x axis and values in yy array indicates y axis. It then returns the distances between each segment as an array. Here, the segment means the Euclidean distance between the neighboring points.

4. **calculateCurveLen(segDists):**

   This function takes segment distances as input and returns the curve length.

5. **calculateSurfaceArea(segDists, y):**

   This function takes segment distances $segDists$ and functional values given by y as input and return the surface area for the solid.

6. **main():**

   The program will begin from this function. In this function, you need to use a for loop that starts at 1 and ends at 100 with step 10. Inside the for loop, you need to call the **generatePoints** function. The first parameter of **generatePoints** will be the value of for loop variable. For example, if we write:

```
for i in range(1, 100, 10):
    ...
    ...
```

   here, i is the for loop variable. The second parameter of **generatePoints** function will be the value of for loop variable, i + 10 and you can set the third parameter, *interval* = *1*. Then you need to call other functions inside the for loop to get surface areas and length of the curve for each iteration. For example, if the for loop iterates 10 times, you will get 10 lengths and areas. See sample output for clarification. You also need to print the values.

7. **printValues(starts, ends, lengths, areas):**

   This function takes a list of start points, end points, lengths and areas as input and show them in the output screen as given below.

Sample output:

```
start/end  |length (m)|area (m^2)
-----------+----------+----------
 1 ...  10 |1.629E+02 |8.250E+04
-----------+----------+----------
11 ...  20 |8.383E+02 |3.318E+06
-----------+----------+----------
```

```
21 ...  30 |1.827E+03 |2.417E+07
-----------+----------+----------
31 ...  40 |3.046E+03 |9.205E+07
-----------+----------+----------
41 ...  50 |4.458E+03 |2.514E+08
-----------+----------+----------
51 ...  60 |6.040E+03 |5.619E+08
-----------+----------+----------
61 ...  70 |7.775E+03 |1.098E+09
-----------+----------+----------
71 ...  80 |9.650E+03 |1.950E+09
-----------+----------+----------
81 ...  90 |1.166E+04 |3.222E+09
-----------+----------+----------
91 ... 100 |1.378E+04 |5.034E+09
-----------+----------+----------

Programmed By Instructors
University of Manitoba
Computer Science
```

"Programmed by..."

Use the following code snippet at the **end** of your script to print out the time and end of processing message:

```
import time
print ("\nProgrammed by ******YOU******")
print ("Date: " + time.ctime())
print ("End of processing")
```

Handing in

An assignment submission area has been created for Assignment 3. Click on the "Assessments" menu, then click on "Assignments".

The only thing you're required to turn in for this assignment is the `.py` script that you create. Please name your script file with the following convention:

```
FirstnameLastnameA3Q1.py
```

[tutorial.math.lamar.edu/Classes/CalcII/SurfaceArea.aspx]: