Project 2

ME-328-102: ME Analysis

Abstract

Multiple-Truncation Trapezoidal Rule to find the volume of a 3D object and its center of buoyancy

Justin Dyer

University of South Alabama student, Author

Landon Freeman

University of South Alabama student, Program Discussion

Index

2-1 Problem Description:

The problem is derived from a 3D object. To calculate the volume of a 3D object, one can integrate the cross-sectional area along the x-axis as follows:

$$V = \int_0^L A(x)dx \qquad \text{(Eqn 2.1)}$$

To calculate the location of the center of buoyancy, one would need to find the x, y, and z components of the center of buoyancy as follows:

$$x_{CB} = \frac{1}{V}\int_0^L x \times A(x)dx \qquad \text{(Eqn 2.2)}$$

$$y_{CB} = \frac{1}{V}\int_0^L y \times A(x)dx \qquad \text{(Eqn 2.3)}$$

$$z_{CB} = \frac{1}{V}\int_0^L z \times A(x)dx \qquad \text{(Eqn 2.4)}$$

where x, y, z are the coordinates of the cross-sectional area's center.

For our project, we are going to use given values for x, y, z, and A(x) and input them into our Python file. Our Python File will use these values and compute x_{CB}, y_{CB}, z_{CB}, and V using the Multiple-Truncation Trapezoidal Rule.

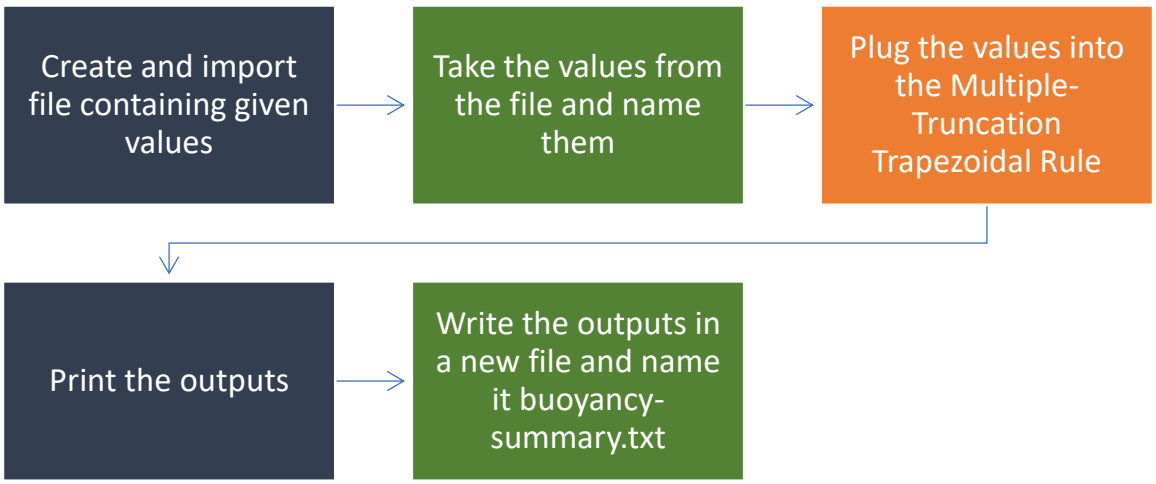
Our given values for x, y, z, A(x) are as follows:

0.0, 0.0, 1.5, 24.0
8.0, 0.0, 1.5, 24.0
16.0, 0.0, 1.6, 25.0
24.0, 0.0, 1.7, 22.0
32.0, 0.0, 2.0, 18.0
40.0, 0.0, 2.5, 10.0
48.0, 0.0, 2.8, 3.0

2-2 Flowchart:

To begin this problem, we need to derive a flowchart to guide us while writing the program. We found that the following flowchart was best suited to guide us:

(Figure 2-1: Flowchart)



Using this flowchart, we have a layout of the steps needed to take to complete what is asked. First, create and import file containing given values. Second, take the values from the file and name them. Third, plug the values into the Multiple-Truncation Trapezoidal Rule. Fourth, print the outputs. Finally, write the outputs in a new file and name it buoyancy-summary.txt.

Following this layout should guide us and let us successfully create the program needed.

2-3 Python Source Codes:

For this project, we used the following Python source codes:

```
"""
Created on Mon Feb 22 15:34:21 2022
ME 328 Project 2
by Justin Dyer
J00655685
"""
x = []
y = []
z = []
A = []
numOfDataPointsRead = 0
inputfile = open('cross-section-input.txt', 'r')
for line in inputfile:
    numOfDataPointsRead += 1
    x.append(eval(line.split(',' or '')[0]))
    y.append(eval(line.split(',' or '')[1]))
    z.append(eval(line.split(',' or '')[2]))
    A.append(eval(line.split(',' or '')[3]))
```

```
inputfile.close()


print('numOfDataPointsRead = ', numOfDataPointsRead)

print('x = ',x)

print('y = ',y)

print('z = ',z)

print('A = ',A)


h = x[1]-x[0]

print('h = ',h)


V = A[0]

xCB = x[0]*A[0]

yCB = y[0]*A[0]

zCB = z[0]*A[0]


for i in range(1,(numOfDataPointsRead),1):
    c = 2.0
    if i == numOfDataPointsRead - 1:
        c = 1.0
```

```
    V = V + c*A[i]
    xCB = xCB + c*x[i]*A[i]
    yCB = yCB + c*y[i]*A[i]
    zCB = zCB + c*z[i]*A[i]

V = V * h *0.5
xCB = xCB * h * 0.5
yCB = yCB * h * 0.5
zCB = zCB * h * 0.5
xCB = xCB/V
yCB = yCB/V
zCB = zCB/V

print('x_CB = ', xCB)
print('y_CB = ', yCB)
print('z_CB = ', zCB)
print('V = ', V)

outputfile = open('buoyancy-summary.txt', 'w')
outputfile.write('{:10.4f}, {:10.4f}, {:10.4f}, {:10.4f}' .format(xCB, yCB, zCB, V))
outputfile.close()
```

2-4 Results and Discussion:

The following is the output we got from our Python Code:

numOfDataPointsRead = 7

x = [0.0, 8.0, 16.0, 24.0, 32.0, 40.0, 48.0]

y = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

z = [1.5, 1.5, 1.6, 1.7, 2.0, 2.5, 2.8]

A = [24.0, 24.0, 25.0, 22.0, 18.0, 10.0, 3.0]

h = 8.0

x_CB = 19.27111111111111

y_CB = 0.0

z_CB = 1.7475555555555555

V = 900.0

Analyzing the data, the program tells you all the given values for x, y, z, A. It also calculates the step value, h, between the x inputs. Finally, it calulates x_{CB}, y_{CB}, z_{CB}, and V using the Multiple-Truncation Trapezoidal Rule and displays their values.

The program then creates a file named buoyancy-summary.txt and writes the values calculated for x_{CB}, y_{CB}, z_{CB}, and V in the file.

2-5 Conclusions:

We have learned to use the Multiple-Truncation Trapezoidal Rule for solving for the center of buoyancy in previous classes. For each value given, with our knowledge of Python we wrote a code to calculate x_{CB}, y_{CB}, z_{CB}, and V using the Multiple-Truncation Trapezoidal Rule.

We seven inputs for each variable x_{CB}, y_{CB}, z_{CB}, and V. Our python program successfully took those values, and calculated calculate x_{CB}, y_{CB}, z_{CB}, and V using the Multiple-Truncation Trapezoidal Rule. To improve this project, we could have more data input points and more accurate inputs; however, to our knowledge, the outputs we received are exact.