Programming Assignment 2
ENGG1003 Introduction to Procedural Programming

hongfu.sun@newcastle.edu.au

Due date/time:Monday Week 13 | 02/06/2025 | 10:00 am
Submission: Upload individual Python files to Canvas

Grading: Week 13| Lab sessions
Assessment Type: Individual
Weighting: 15%

University of Newcastle, Callaghan, Australia

Question 1

5 marks

(a) (3 marks)

Consider the function

$$f(x) = x^2 \sin(x) + 2x - 3.$$

(i) By writing a Python script to plot the function $f(x)$ over the interval $[0, 2]$, visually confirm that $f(x)$ has a single root in this interval.

(ii) Using the $bisection()$ function in the $rootfinding$ module, estimate the root of $f(x)$ in the interval $[0, 2]$. Display the root to 5 decimal place precision. Check the correctness of your answer by confirming that $f(x) = 0$ is satisfied to at least 5 decimal places, where x is the solution computed numerically by bisection().

(iii) Using the $secant()$ function in the $rootfinding$ module, estimate the root of $f(x)$ in the interval $[0, 2]$. Display the root to 5 decimal place precision. Check the correctness of your answer by confirming that $f(x) = 0$ is satisfied to at least 5 decimal places, where x is the solution computed numerically by secant().

Rubric:

- Correct Plot: 1 mark
- Partially Correct Plot: 0.5 mark
- 1 mark for correct calculation of root to 5dp precision for part ii and iii
- 1 mark for correctly checking accuracy of root by confirming f (x) = 0 for part ii and iii

(b) (2 marks)

Water is discharged from a tank through a long pipe, as shown in Figure 1.

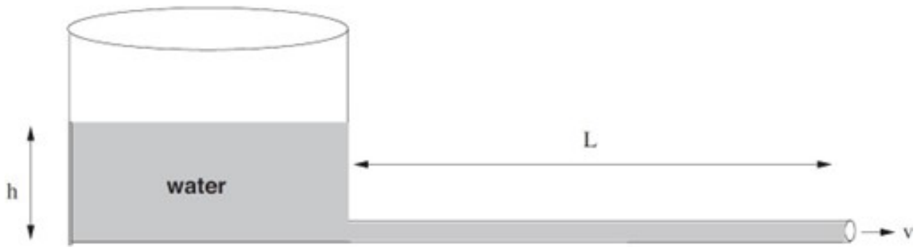

Figure 1: Water height for Question 1

The velocity v [units: m/s] of the water flowing from the pipe is given by

$$v = \sqrt{2gh} \times \tanh(\frac{2}{L}\sqrt{2gh}).$$

where:

- h is the height of the water in the tank [m]
- L is the length of the pipe [m]
- $g = 9.81$ is the acceleration due to gravity [m/s^2]

Write a Python script to calculate the value of h necessary to achieve a velocity of $v = 4$ m/s, when $L = 5$ m. Display your answer to 2 decimal place precision.

Hints:

i. To solve an equation $f(h) = v$, find a root of the equation $f(h) - v = 0$.

ii. The hyperbolic tangent function $\tanh()$ is available in the numpy library as $numpy.tanh()$.

Rubric:

- 1 mark for correct definition of Python function which evaluates $f(h) - v$ for suitable choices of $f(h)$ and v
- 0.5 mark for partially correct definition of Python function
- 1 mark for correctly calculating the root of the equation $f(h) - v = 0$ and displaying the result $(ie : waterheighth)$ to $2dp$ precision
- 0.5 mark for partially correct calculation and display of water height h

Question 2

5 marks

(a) (3 marks)

Write a Python script to evaluate each of the following expressions. Use either the $trapezoidal()$ or $simpson()$ functions in the $integration$ module to numerically evaluate the integral in each expression.

In each case, the exact value of the expression is provided for reference. Determine how many sub-intervals are needed before the results of numerical integration agree with the exact value to five (5) decimal places.

It is not necessary to determine the precise number of sub-intervals; simply determine the required number of sub-intervals to the nearest 100, e.g., 100, 200, 300, etc. Nor is it necessary to code a solution which determines the required number of sub-intervals. Using trial and error to determine the required number of sub-intervals (to the nearest 100) is quite sufficient.

(i)

$$\int_0^1 (7 + 14x^6)\, dx$$

Exact value: 9

(ii)

$$\int_0^1 \frac{2}{1 + x^2}\, dx$$

Exact value: $\pi/2$

(iii)

$$\frac{3\sqrt{3}}{4} + 24 \int_0^{1/4} \sqrt{x - x^2}\, dx$$

Exact value: π

Hint: the $trapezoidal()$ and $simpson()$ functions in the $integration$ module each perform numerical integration using $n = 100$ sub-intervals by default. Overriding the default number of sub-intervals can be done as follows:

```
import integration as integ
T = integ.trapezoidal(f, a, b, n=20) # 20 sub-intervals over [a, b]
S = integ.simpson(f, a, b, n=200)    # 200 sub-intervals over [a, b]
```

Rubric:
For each of parts i, ii and iii:

- 0.5 mark for correct evaluation and display of expression to 5dp precision
- 0.5 mark for correct evaluation of number of sub-intervals (to the nearest) 100

(b)(2 marks)

For large N, we expect the fraction of random numbers X drawn from the standard normal distribution that fall in the interval $[a, b]$ to be close to the exact result:

$$\Pr(a \leq X \leq b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2}\, dx. \tag{1}$$

(i) Write a Python script which uses either the $trapezoidal()$ or $simpson()$ functions in the $integration$ module with $n = 1000$ sub-intervals to evaluate the expression on the right-hand side of the equation (1) when $a = -1$ and $b = 3$. Display the result to a precision of 3 decimal places.

(ii) Extend your script to create a numpy array X of $N = 10^6$ random numbers drawn from the standard normal distribution. Initialise the random number generator using `numpy.random.seed(1)`. Compute the fraction of elements of array X which fall in the interval $[-1, 3]$. Confirm that your result matches the result from part (i) to at least 3 decimal places.

Rubric:
For each of parts i, ii and iii:

- 1 mark for correct evaluation and display of expression in (1) to 3 dp precision
- 0.5 mark for partially correct evaluation and display

Question 3

5 marks

(a) (3 marks)

i. Write a Python script which loads a small dataset as follows:

```
import numpy as np
t = np.linspace(0, 100, num=11)
y = np.array([132, 138, 189, 229, 317, 304, 435, 435, 506, 545, 629])
```

Plot the y-data versus t-data using red circle linemarkers.

ii. Use the $interp1d()$ function in the $scipy.interpolate$ module to linearly interpolate the data in the arrays t and y. Plot the linearly interpolated data on the same set of axes as the plot in part i. Your plot should appear as shown in Figure 2.

iii. Extend your script to display the value of the linearly interpolated data at time t = 55.

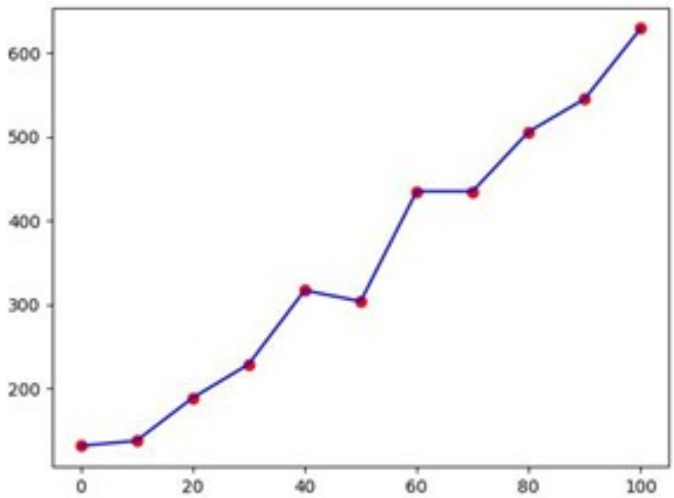

Figure 2: Linearly interpolated data for Question 3(a)

Rubric:

- 1 mark for correct plot
- 0.5 mark for partially correct plot
- 1 mark for correct interpolation and plot
- 0.5 mark for partially correct interpolation and plot
- 1 mark for correct interpolation and display to 1dp precision
- 0.5 mark for partially correct interpolation and display

(b)(2 marks)

The water volume in a tank of capacity 400 litres (L) is measured at 1-minute intervals. The measurement times and water volumes are available in a CSV file $level.csv$, which is available in Canvas.

1. Write a Python script which uses the $pandas$ module to open $level.csv$, read the measurement time data into a numpy array t, and the water volume into a numpy array y. Plot the y-data versus t-data using red circle linemarkers.
2. Use the $curve\_fit()$ function in the $scipy.optimize$ module to fit a straight line to the (t, y) data, and plot the straight-line fit on the same set of axes as the data plot in part (i). Show that if the tank continues to fill at the rate predicted by the straight-line fit, then the tank volume will reach its capacity of 400 L at $t = 87.7$ minutes.

Rubric:

- 1 mark for correct use of pandas and plot of data
- 0.5 mark for partially correct use of pandas plot of data
- 1 mark for correct straight-line fit, data plot and calculation of time t = 87.7 minutes when water volume is 400 L
- 0.5 mark for partially correct straight-line fit, plot and time calculation