J. Bürki, M. Block and R. Barniol Duran

Homework 6: Solving Equations in Python

- Make sure that you scripts run without error in order to get credit. Do not hesitate to ask for help if needed!
- Check the output of the Autograder for any issue that should be fixed. In case of "Unexpected error", email me so I can take a look as it may be an issue with the Autograder rather than your code.
- Take ownership of your learning! Remember that you are responsible for the work you turn in. Simply copying somebody else's answers, copying from the Internet, using AI to generate your code, sharing your code (or part of your code) in any way, or copying it from someone else will be considered academic dishonesty. Please, contact me if you have any questions about collaborations.

Problem 1

Going back once more to the projectile problem, write a code that uses the bisection method to solve numerically for the x-coordinate of the landing point if the ground is defined by the function

$$y(x) = x^{1/3}. \tag{1}$$

You should reuse the function you defined previously to generate the trajectories and call your function implementing the bisection method `bisection`; it should take three arguments: the two values giving the initial bracket, and the required precision.

Use $\theta_0 = 27^o$, $y_0 = 5\,\mathrm{m/s}$, and solve for $v_0 = 7\,\mathrm{m/s}, 17\,\mathrm{m/s}, 27\,\mathrm{m/s}$. Create a plot that shows all 3 trajectories and plots the ground as a thick, black line. Make sure to interrupt the trajectories when they hit the ground and to include a legend and a symbol with the same color as the trajectory at each landing point.

Print the coordinates of the landing points on the terminal with some text. It might look something like:

```
v0 = 7 m/s, (x,y) = (10.2, 3.82) m
...
```

What to submit: Submit the python script finding the solution and creating the plot, as well as the figure that its creates.

Problem 2

Solve the following equations and systems of equations using the `scipy` module.

In all cases, find all solutions, or at least 5 solutions (whichever is smaller). For parts (a) and (b), limit yourselves to positive (i.e., not including 0) values of x.

For each case, make a plot, adding dots at each solution you have solved for. Print in the terminal the values of the solution(s) you found.

(a)
$$\ln(x) = 3\cos(x)$$

(b)
$$x = 8\cos(x)$$

(c)
$$\begin{cases} x^5 - y^3 + y^2 = \sin(x) \\ y^5 - x^3 = -\frac{1}{4}\left(\cos^2(x) - 0.98\right) \end{cases}$$

Do not try to solve any part of this system of equations analytically. The goal here is to use purely numerical methods.

What to submit: For this problem, create a separate script file for each part and include "`part-a`", ... to each script name. Make sure to also submit figures for each part and use the usual naming convention for each file (e.g. `buerki-hw06-p1-part-a.pdf` for the figure of the first part).

Problem 3 Additional problem for practice (no submission)

Solve the following system of linear equations using the `numpy` module:

$$\begin{cases} 11a + 1 & = b + 3c \\ 9c & = 8 + 4a \\ 2a + 5b & = 7 + 15c \end{cases}$$