Buffon’s Needle Report

Task 1

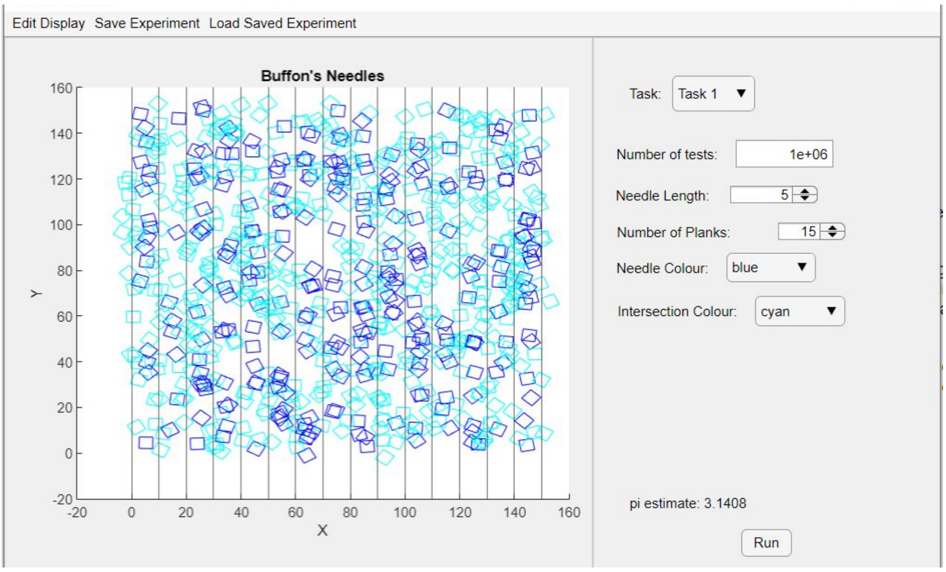
This task required a modification, to Buffon’s needle experiment, that dropped 4 line boundaries - i.e. a square - instead of a singular needle, to calculate pi.

To create these squares I first generated a random coordinate as the starting point for the first line. Then using this as the base, I was able to generate 3 additional points - a distance of “L” apart, “L” being the needle length decided upon by the user. Each square was rotated by a random angle, just like the needles in the original experiment.

I opted to only allow the app to display the first 750 squares in each experiment, I did this because otherwise the display can appear very dense and no shapes could easily be identified. It also meant that the app would run slightly more efficiently when doing larger numbers of tests as only a small proportion of the results would need to be displayed.

I used the equation piEstimate = 2 * numTests / crossed; to calculate pi using the squares as opposed to the normal equation as the squares made it more likely for each test to hit a crack.

Below is an example of the results generated by task 1.



Task: Task 1

Number of tests: 1e+06

Needle Length: 5

Number of Planks: 15

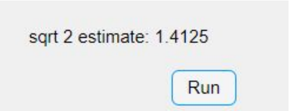
Needle Colour: blue

Intersection Colour: cyan

pi estimate: 3.1408

Task 2

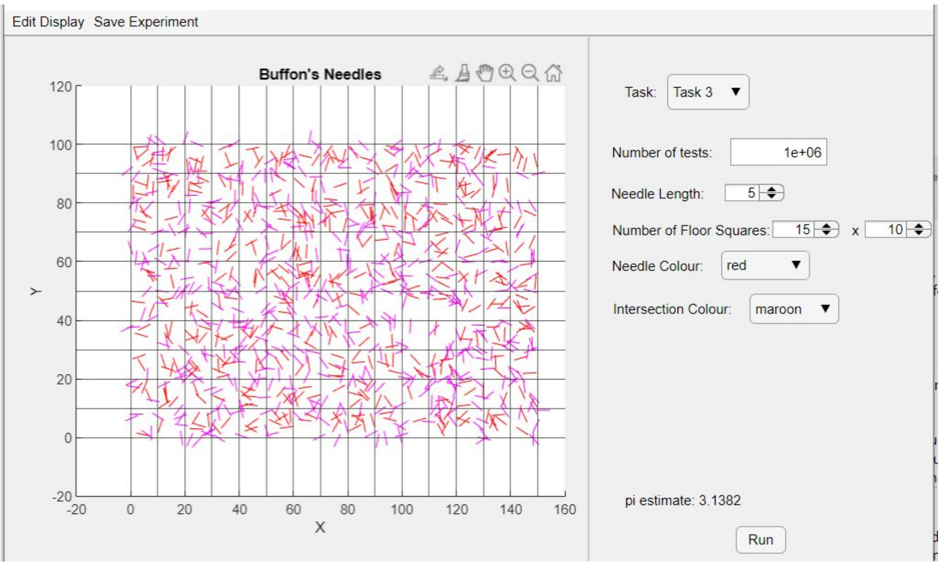
This task needed to estimate the square root of 2, for this an equation was given to estimate pi and it needed to be rearranged to estimate root 2. This task utilised the squares used in task 1 and presents them as above to the user, the one difference in the output is instead of reading “pi estimate:” and the estimate, it reads “sqrt 2 estimate”. Shown here.



sqrt 2 estimate: 1.4125

Task 3

This task required a further modification to the experiment so that it was dropping needles, as is standard for Buffon’s needles, but onto square tiles as opposed to planks. This required a new equation for calculating pi. An example of this task in action is shown below.



Task: Task 3

Number of tests: 1e+06

Needle Length: 5

Number of Floor Squares: 15 x 10

Needle Colour: red

Intersection Colour: maroon

pi estimate: 3.1382

The equations I used to calculate pi in this task were: P = crossed/numTests; and piEstimate = (4*L*D - (L*L)) / (P*(D*D)); Where P is the proportion of tests that hit a line, L is the length of the needles, and D is the width of the square tiles.

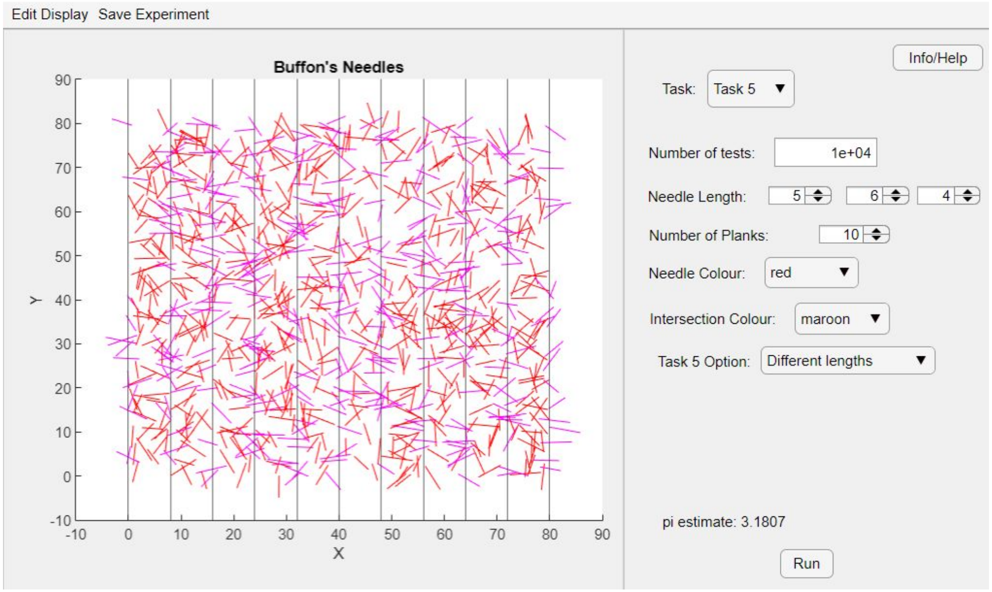
In all of these tasks, the user is able to adjust the number of tests, the needle length, the number of planks/squares, the needle colour when it isn't intersecting any cracks, and the needle colour when it is intersecting cracks. The user is also able to edit the colour and size of the label for the estimate.

Task 5

This task required some additional novel extensions to be made to the experiment. As an additional feature in task 5, not a novel extension, I also added an info/help button to explain the app and it's functionality.

The first novel extension I opted for was to make alternate versions to estimate pi. For this I decided to create the option to run the simulation with triangles, squares (again), or irregular pentagons.

The second novel extension I attempted was the ability to save the simulation, however I ran into some issues with saving the UIAxes, so after managing to save the simulation as an image or .fig file, I was unable to reload it into the program. As a result I decided to leave the save function at this but also add an extension that allows for different sized needles to be dropped. This extension allows the user to select 3 different lengths for the needles, the simulation then runs and randomly decides which length of the 3 options, each needle is going to be. It also uses the 3 options to calculate the spacing of the planks. Again, shown below.



Task: Task 5

Number of tests: 1e+04

Needle Length: 5 6 4

Number of Planks: 10

Needle Colour: red

Intersection Colour: maroon

Task 5 Option: Different lengths

pi estimate: 3.1807