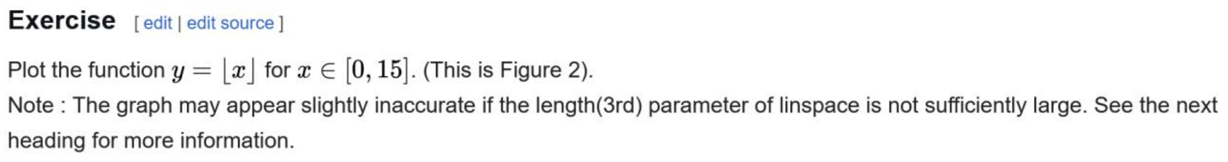
Rachel Hussmann

Learning how to program in MATLAB/Octave

As part of our Project 2 for Probability and Applied Statistics, we were tasked with learning a new programming language to write our Plotter, Salter, and Smoother program. The programming language we were tasked with learning was MATLAB/Octave. MATLAB was the main language to learn, but Octave is extremely similar and is open source for us to use.

The first tutorial I used was from Wikibooks, called Octave Programming Tutorial/Getting started. Link to tutorial: https://en.wikibooks.org/wiki/Octave_Programming_Tutorial/Getting_started

I followed through the tutorial beginning with learning how to plot graphs. I tried the first exercise which was:

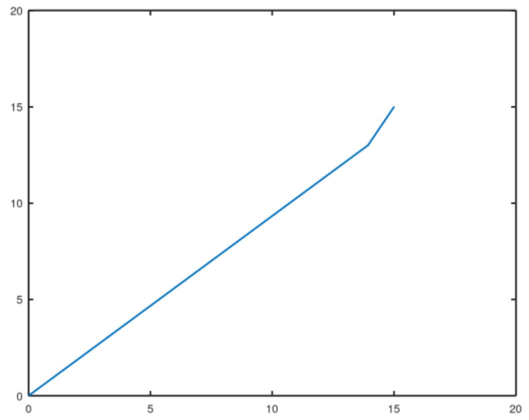


Exercise [edit | edit source]

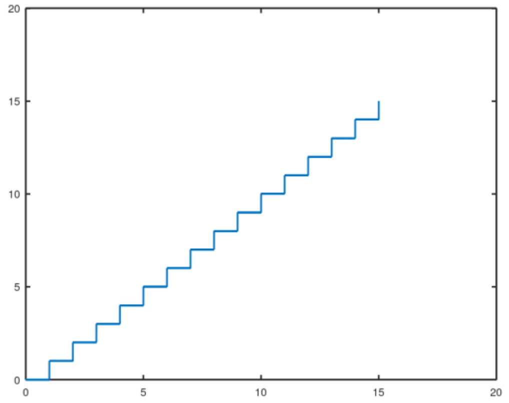
Plot the function $y = \lfloor x \rfloor$ for $x \in [0, 15]$. (This is Figure 2).

Note : The graph may appear slightly inaccurate if the length(3rd) parameter of linspace is not sufficiently large. See the next heading for more information.

At first my function did not look correct:



However, after reading more about the linspace function, I learned that if there is not enough space between the start and the final value given, the graph may look a little strange. After giving the vector more space in between the values (10000 instead of 15), it looked correct:

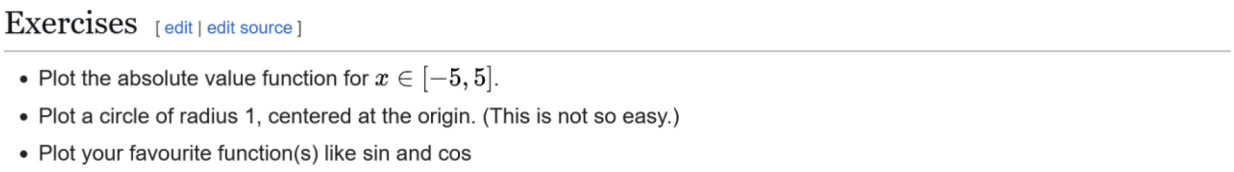


I realized later on that the reason why the first graph looked strange is because I did not give the vector enough data points. Giving a function less data points will cause it to look strange, because we do not have enough data to see the trend. When I gave the function more data points to use, it gave me the graph I was looking for because it was able to capture more data and graph the trend properly.

Here is the script that gave me the result above:

```
x = linspace(0,15,10000);

y = floor(x);

plot(x,y);
```

Continuing through the tutorial led to the exercises section:

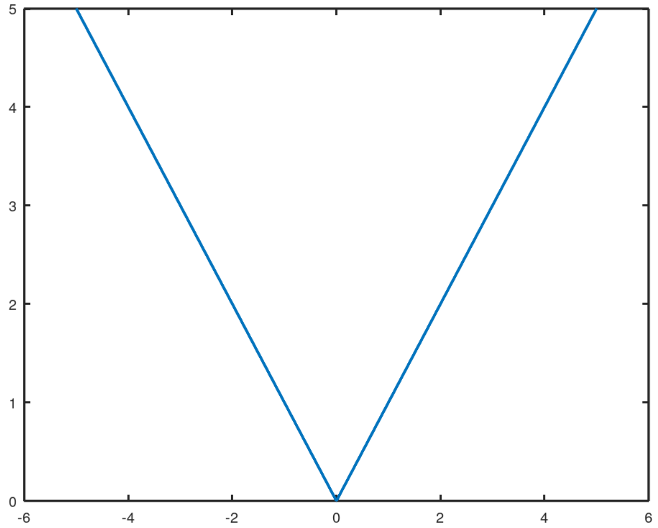


Exercises

- Plot the absolute value function for $x \in [-5, 5]$.
- Plot a circle of radius 1, centered at the origin. (This is not so easy.)
- Plot your favourite function(s) like sin and cos

Plot the absolute value function for -5 to 5

My output



My code

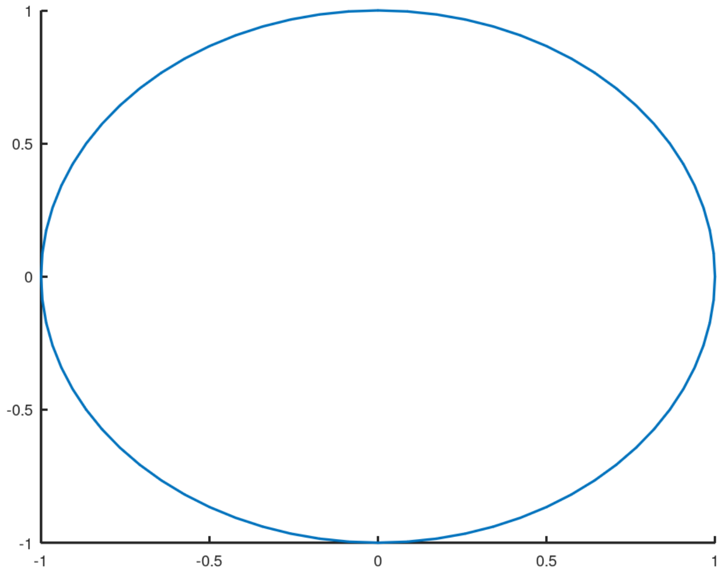
```
x = linspace(-5,5, 10000);

y = abs(x);

plot(x,y);
```

Plot a circle with a radius of one, centered at the origin

This part was more difficult than I expected, as I tried many different times on my own with no success. I soon realized that I did not have enough experience with the software to figure this out on my own, so I looked for another tutorial to help me figure out what to do. The second tutorial that I used was titled “How to draw a circle in GNU Octave” by the YouTube channel Quancept. Link (https://www.youtube.com/watch?v=An7GJnpuJcI)

This tutorial taught me that Octave has the ability to import packages, much like Java and Python can, to add more functionality to the software.

My output



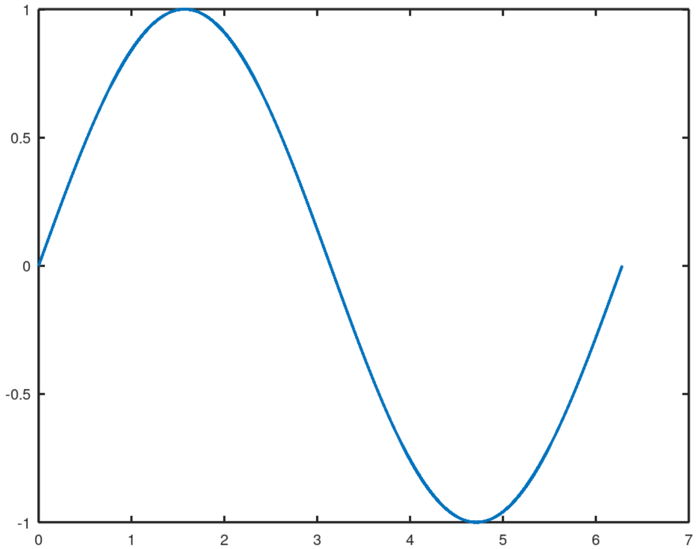
My code

```
pkg load geometry

x = 0;

y = 0;

radius = 1;

c = drawCircle(x,y,radius);
```

This package made it much easier to solve this problem. It also saved time since someone else already wrote the code to create the circle.

Plot your favorite function

My output



My code

```
x = linspace(0, 2*pi, 100000);

y = sin(x);

plot(x,y);
```

After completing tutorial one, I learned a lot of information about the language. The main takeaways that I learned were:

- Syntax – It was similar to languages that I have used before but still has some differences to watch out for.
- Declaration of variables – Unlike Java, Octave does not need to know what the type of the variable will be to store it. Octave is much like Python in this respect, as it is able to figure out what the type of variable will be at runtime.
- Documentation – As I was running through the tutorial, I felt like I didn't really understand what was happening, so I resorted to the documentation for Octave for help. This gave me a better understanding of what was happening behind the scenes and also showed me new methods that I could use in my Plotter, Salter, and Smoother program.