Julian Lafond

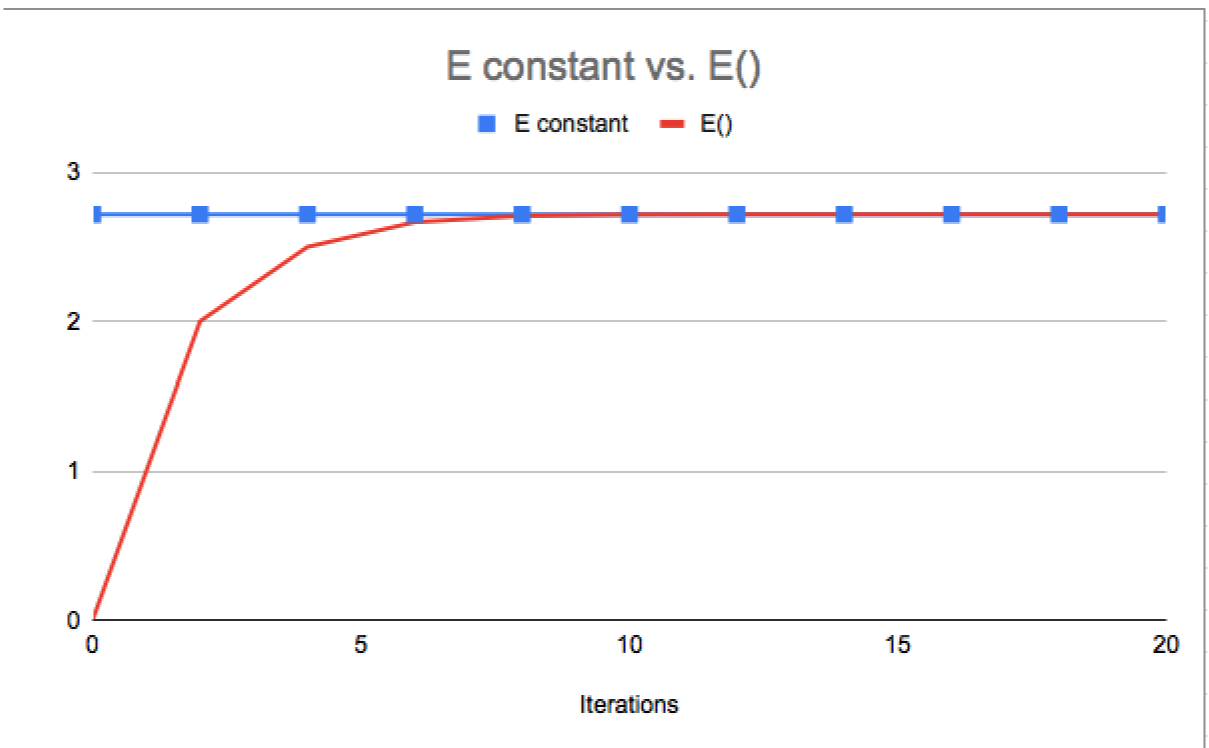
Writeup: A Little Slice of Pi

In this assignment, we are implementing various mathematical formulas in order to find the best approximation closest to pi and e. We then are comparing our approximations of pi and e values to the actual values in the C math library.

For our first function of E, we implemented the following formula:

$$e = \sum_{k=0}^{\infty} \frac{1}{k!} = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \frac{1}{720} + \frac{1}{5040} + \frac{1}{40320} + \frac{1}{362880} + \frac{1}{3628800} + \cdots$$

After implementing the algorithm, here is my approximation in red compared to the actual value in the math.h library in blue.



We can see from that graph how close my approximation was to the actual value. Not only was it a good approximation but we see how rapidly my approximation grew. It only took around 5 iterations until my approximation was relatively the same to the actual e constant value.

In the second function of euler's solution, we are now iterating through the following formula

$$p(n) = \sqrt{6 \sum_{k=1}^{n} \frac{1}{k^2}}$$

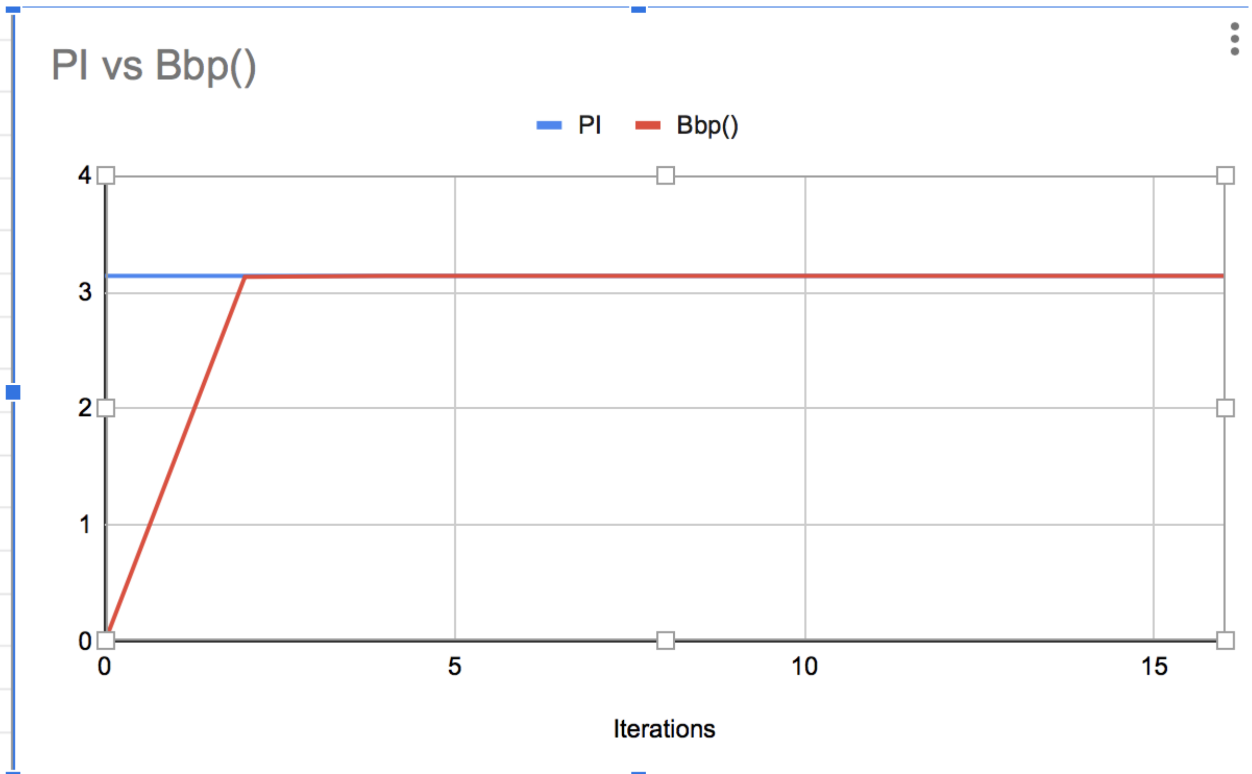
to find the closest approximation to the value of Pi. The following graph supports my findings:

This graph in red shows the constant value of Pi. The line in orange shows my approximation. This graph displays the first iteration to the 1800th iteration. We notice our approximation is very similar to the actual PI value although there is a significant drop in the line around the middle of our iterations. This could be the reason due to the large difference between our approximation and the actual value.

The next formula was bbp. Implementing the formula for bbp

$$p(n) = \sum_{k=0}^{n} 16^{-k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

to find the closest approximation to PI. Here were my findings:

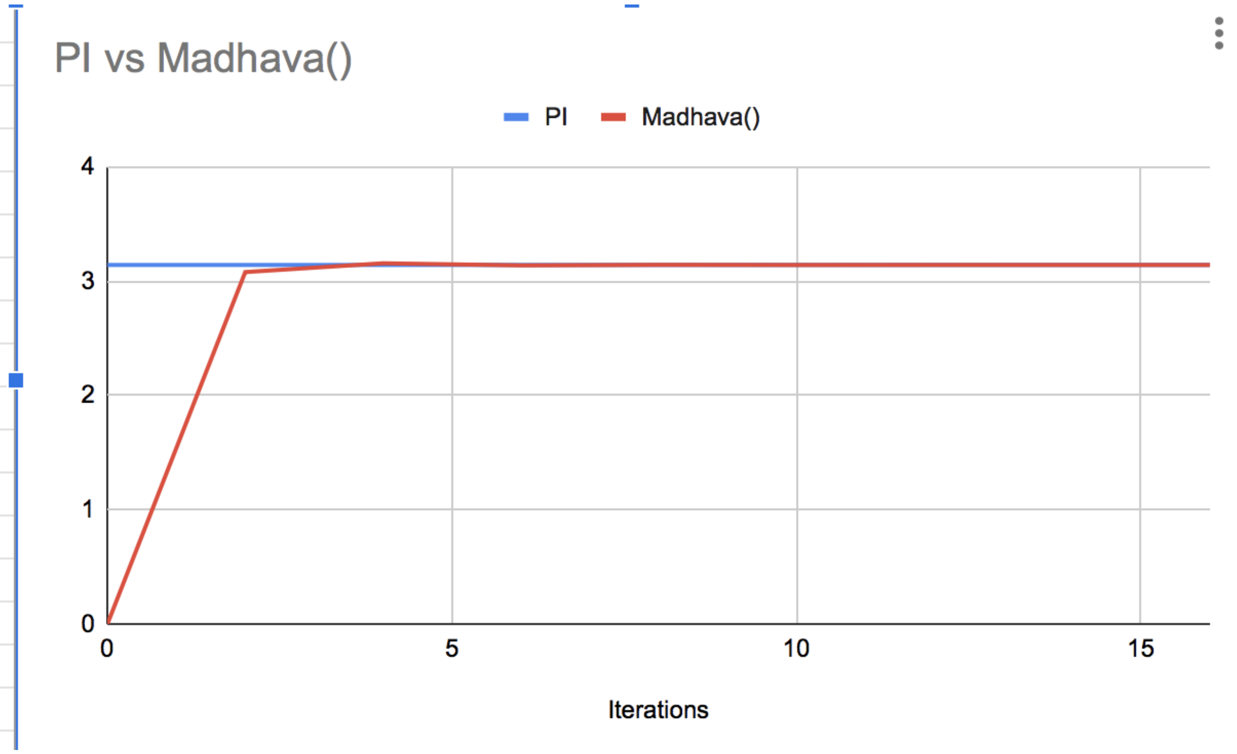


We see my approximation was very close to the actual value of PI. In fact it was exactly the same and we see that it did not take many iterations until we found the same value to PI. This was by far the best approximation I had and I believe it was due to the efficiency of my program.

In the madhava formula we were required to implement the following formula

$$p(n) = \sqrt{12} \sum_{k=0}^{n} \frac{(-3)^{-k}}{2k+1}$$

to find the closest approximation to the value of PI.

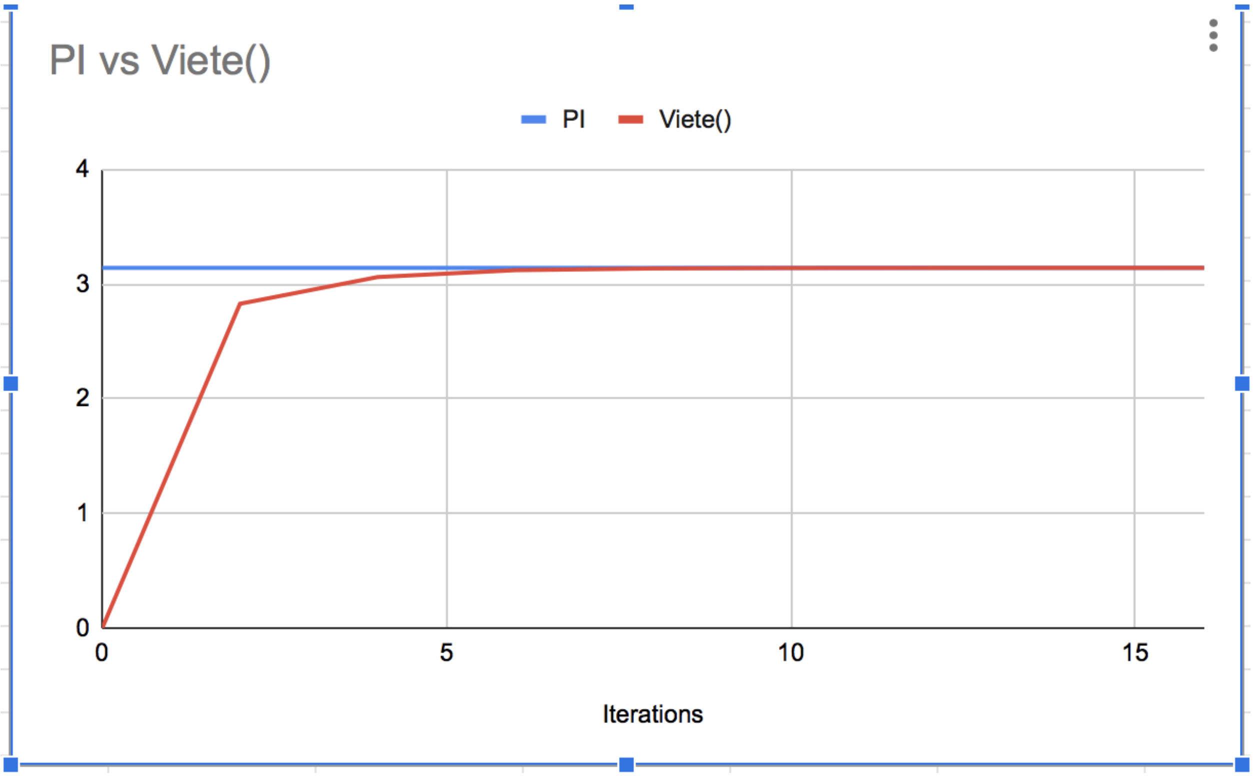


Similar to the bbp function, I also had a very close approximation to the PI value. It also did not take many iterations to reach a similar value to PI.

In the viete function we were required to implement the following formula

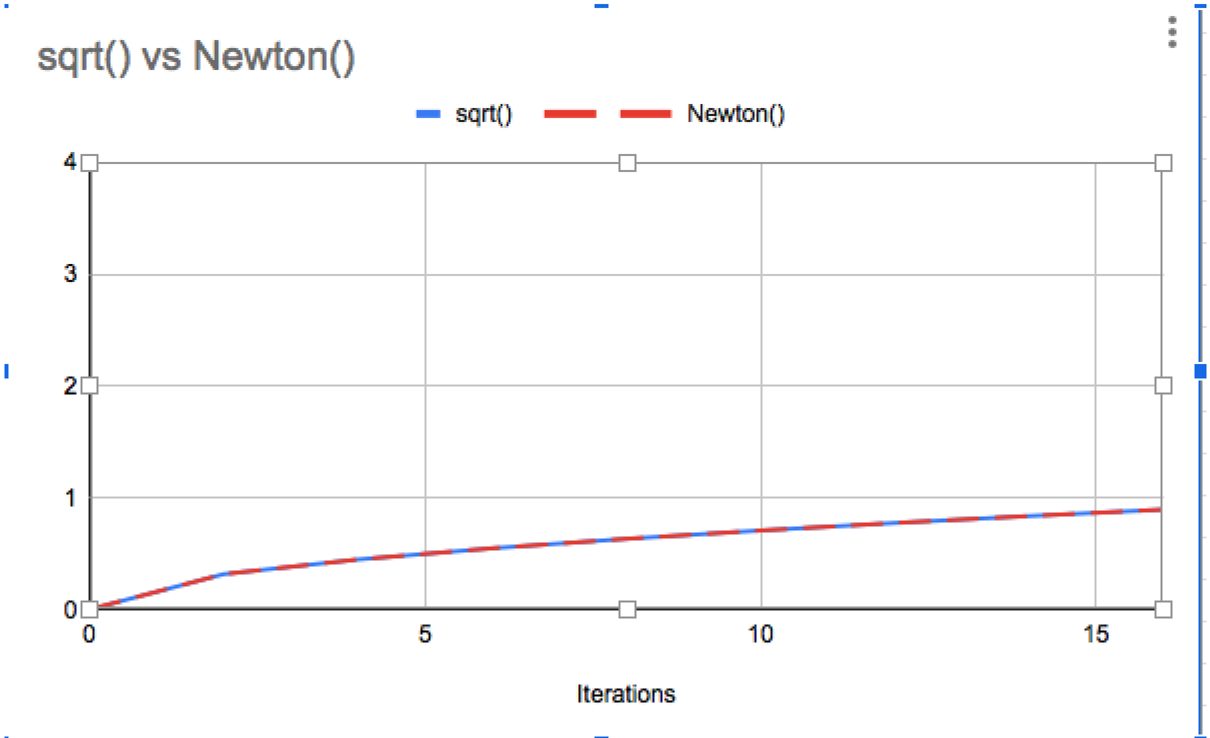
$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \times \frac{\sqrt{2+\sqrt{2}}}{2} \times \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdots$$

.



In this function we also see that we had a close approximation to the actual value of PI. Although this approximation took a little longer compared to the bbp and madhava, we still were able to find a solid approximation.

In the final function of Newton, we see that the approximation is the same exact as the actual value of the square root. This was our strongest approximation since it was not too difficult calculating the square root of a number.



In conclusion, many of my approximations were accurate to the actual value in the math library. I would say the only approximation that could be improved was the euler function. This comes to show that many of the actual values in the math library are close to the epsilon value which is $1e^{-14}$. This assignment demonstrated the importance of writing our own formulas to find a close approximation which is completely visible.