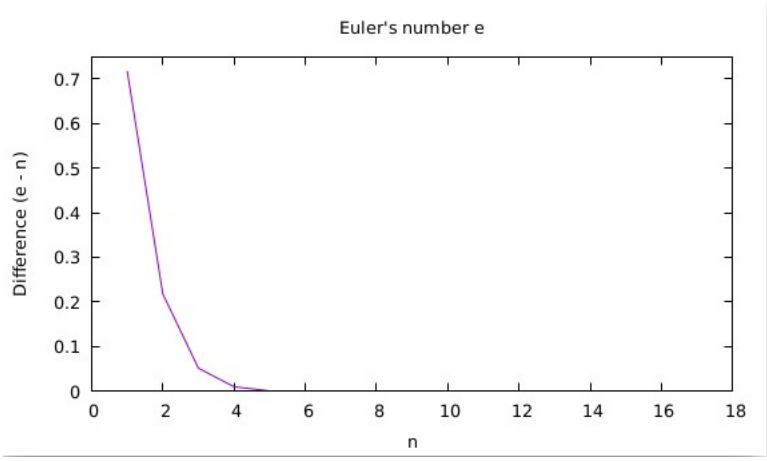
Assignment 2 Writeup

Caitlin Smith

January 29, 2023

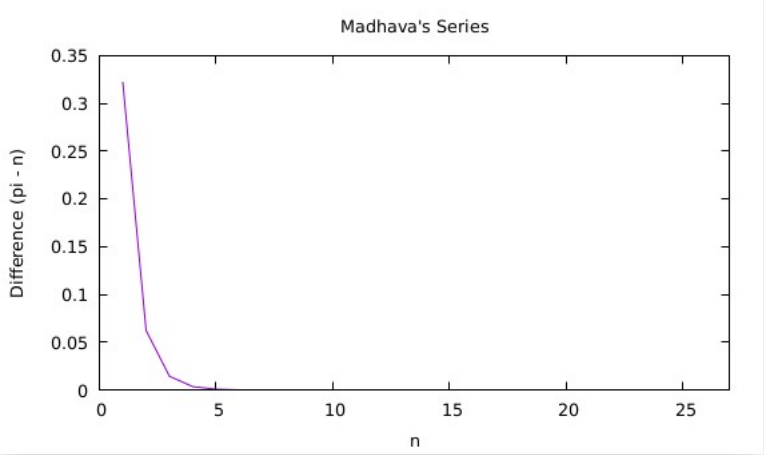
For all of the graphs below, I was able to slightly alter print statements in my functions to collect the correct data. I created a bash script that ran ./mathlib-test, putting data into a .dat file, and plotted the graphs using gnuplot.

1 e.c



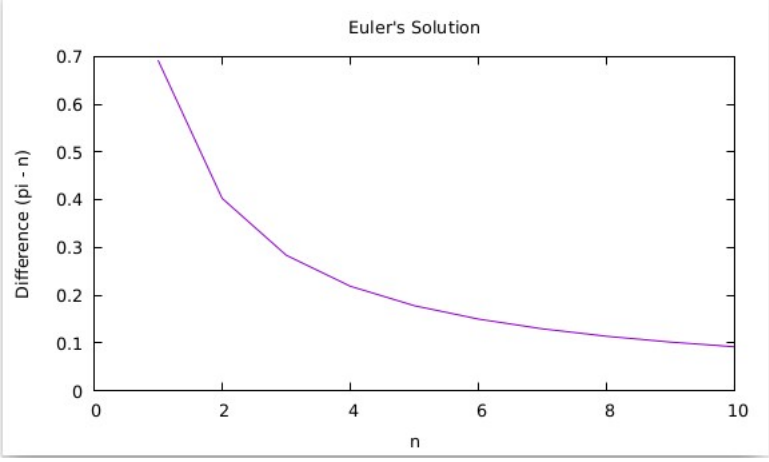
The graph above displays data from my function's approximation of Euler's number compared to the math library. The x-axis is the term and the y-axis is the difference. By the fourth term, the variation from the library's e value is already very close to 0. It then takes until 18 terms for the function to reach "equivalence" using epsilon.

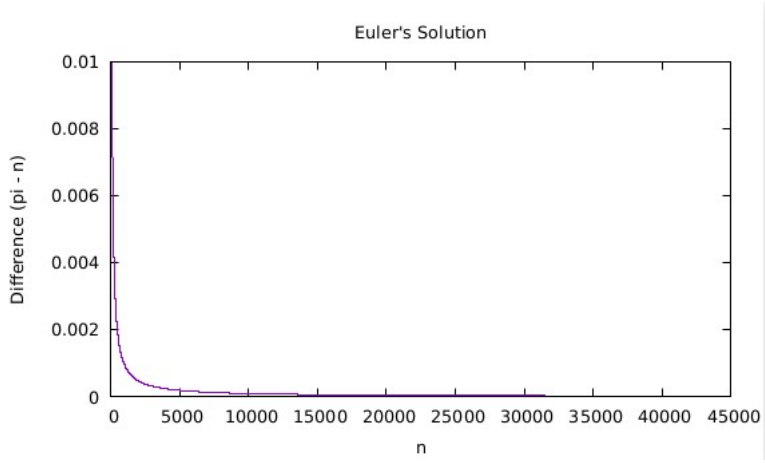
2 madhava.c



The graph of Madhava's series above looks very similar to the graph for Euler's number. At the very beginning, there is a steep drop from 0.3 to almost 0 in difference. It takes until about 5 terms for the difference to be extremely close to 0. The formula then runs for about 20 more iterations.

3 euler.c

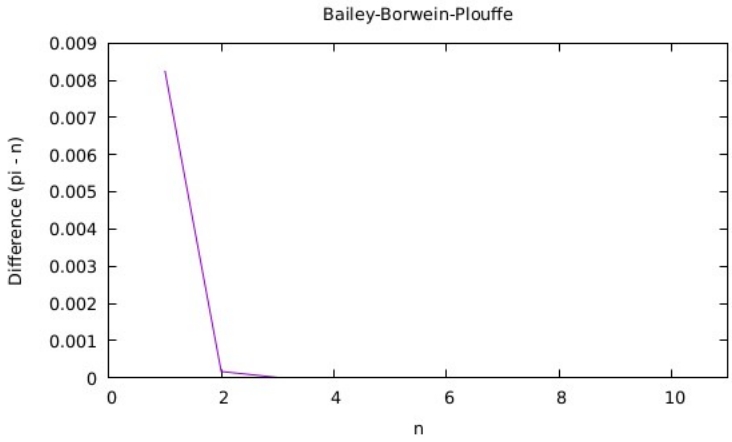




The graph of Euler's solution is very interesting. From the terms 0 to 10, it drops from 0.7 to 0.1 in difference. This is shown in the first graph above. Very soon after, the difference is down to below 0.01. This is shown in the second graph. It continues to drop drastically until about 15,000 terms where it is almost on the axis. The line fully disappears on the graph slightly after 30,000.

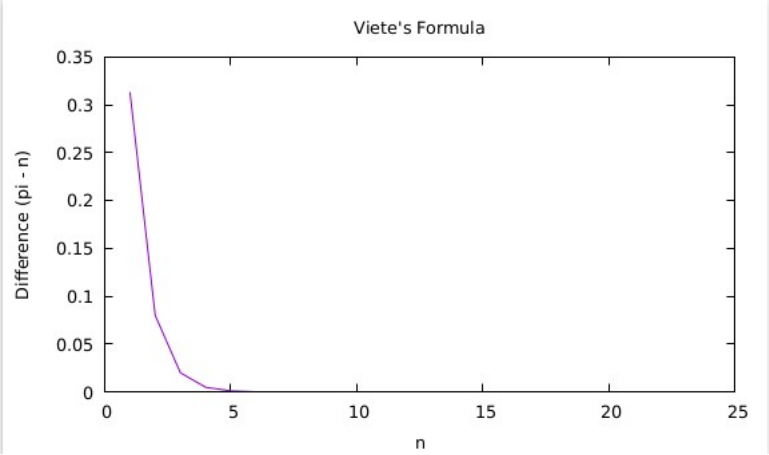
My function for this series was not as close to the true value of pi as I had hoped. The terms got so small so quickly that my while loop terminated at around 46,000 terms. Given more time, I would probably change the function so that the condition of my while loop was comparing the difference between the previous and current term to epsilon instead of just the current term.

4 bbp.c



The graph above displays the BBP results from my function and the math library's. The x-axis is the term and the y-axis is the difference from M_PI. There is a sharp and smooth drop from 1 to 2 terms. After that, it quickly flattens to be on the x-axis. The largest difference was slightly over 0.008. Because of that, this formula only took a little more than 10 iterations to find an essentially "equivalent" value using epsilon.

5 viete.c



The graph for Viete's formula is very similar to the first two graphs in shape. It quickly drops from slightly above 0.3 to being on the x-axis in 5 terms. The line is not visible after that. This formula then quickly finds the approximate value of pi.