Analysis – A Little Slice of π

Brian Nguyen

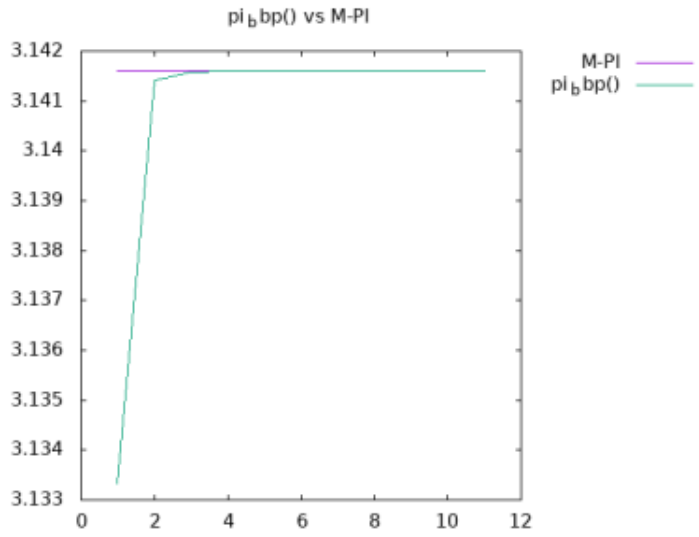
October 2021

1 Introduction

In this math library, I created 6 math functions that try to replicate the constants or operations found in the built-in "math.h" library found in C. Various traditional methods and formulas of series and summations from Euler's Solution to Viete's formula were used to try and accurately approximate the constants or operations, thus providing slight differences in some programs. In this writeup, I will analyze and try to explain these differences between my program and the constants or operations: M_E, M_PI, and sqrt().

2 Analysis of Differences

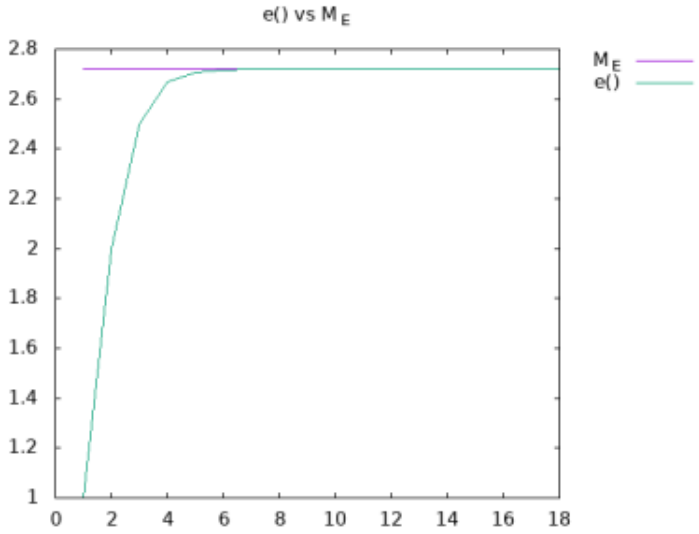
2.1 bbp.c



The output for this program exactly matches the constant M_PI due to being able to handle the formula or summation very well using a for loop. It also reached epsilon fairly fast using only 11 iterations and converged very fast, as shown by the graph, meaning not many small numbers

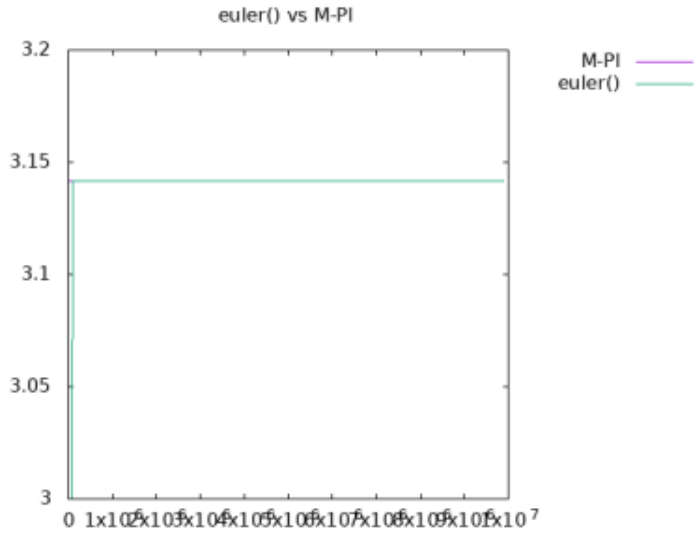
are dealt with until the later iterations when it is clear the difference is negligible. It also did not require other inaccurate programs in my library, such as sqrt_newton.

2.2 e.c



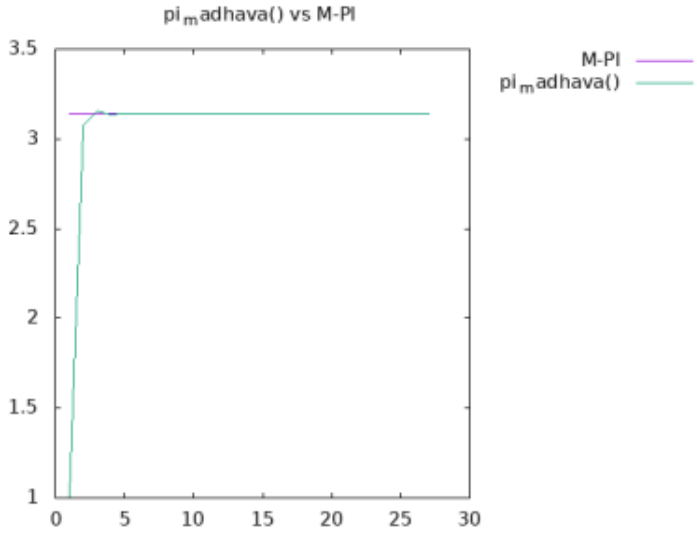
The output for this program exactly matches the constant M_E due to being able to handle the formula or summation very well using a for loop. It also reached epsilon relatively fast and converged fast, as shown by the graph using only 18 iterations, meaning not many small numbers were dealt with until the later iterations when it is clear the difference is negligible. It also did not require other inaccurate programs in my library, such as sqrt_newton.

2.3 euler.c



The output for this program varied slightly (1e-8) with the constant M_PI due to the formula using a summation with only the power of 2 to scale. It took around 10 million iterations to finally reach epsilon meaning those small numbers may eventually converge even later on. Because of the formula rising relatively slowly compared to others that used factorials and such, it consistently handled tiny numbers in later iterations that actually made a small but noticeable difference that may require even smaller numbers to get even more accurate epsilon. With this being said, the 1e-8 difference is very small, all things considered, but it is still a difference.

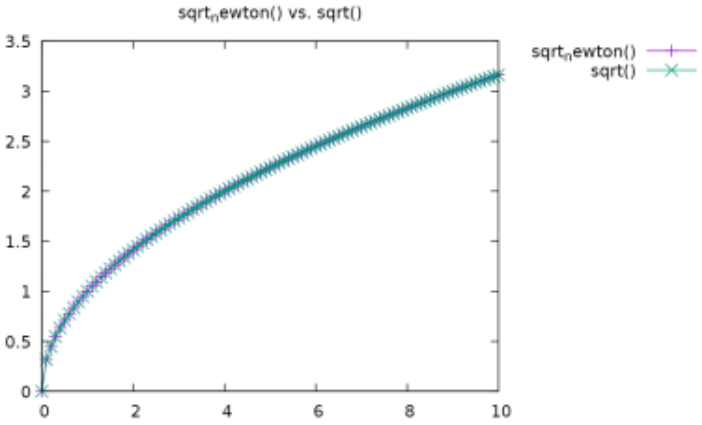
2.4 madhava.c



The output for this program was nearly identical, with the constant having only a 1e-14 difference from M_PI due to being able to handle the formula or summation very well using a for

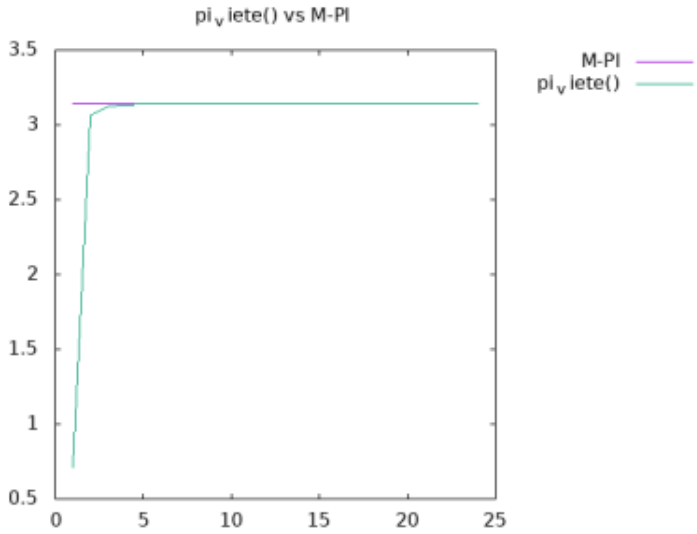
loop. It also reached epsilon quickly, but not as fast as other methods that did it below 20, using 27 iterations, meaning only a few small numbers were dealt with when it is clear the difference is negligible. However, it did converge as fast as other methods, as shown by the graph. It did require sqrt_newton, a program of mine that is not 100 percent accurate due to it being approximations ultimately, which most likely explains the difference shown.

2.5 newton.c



The output for this program was identical with the sqrt() operation except for calculating the sqrt of 0. It could be inaccurate in the later decimal places, which I strongly assume because the formula essentially approximates using tangent lines to guess where the sqrt is. This could mean it is very possible for there to be slight differences that affect other programs of mine that use it, such as madhava and viete.

2.6 viete.c



The output for this program was nearly identical, with the constant having only a 1e-14 difference from M_PI due to being able to handle the formula or summation very well using a for loop. It also reached epsilon quickly, but not as fast as other methods that did it below 20, using 24 iterations, meaning only a few small numbers were dealt with when it is clear the difference is negligible. However, it did converge as fast as other methods, as shown by the graph. It did require sqrt_newton, a program of mine that is not 100 percent accurate due to it being approximations ultimately, which most likely explains the difference shown.

3 Conclusion

From the results of the graphs and outputs, it can be concluded that these formulas or summations that approximate the constants M_E and M_PI are exceedingly accurate, but do, in the end, have some very slight variance since they are approximations at the end of the day. The sqrt_newton operation also was exceedingly accurate but still most likely also had some slight variance which caused other math programs to be inaccurate on top of their own formulas and approximations.