Cse13s Pie

David Shonack

January 30, 2023

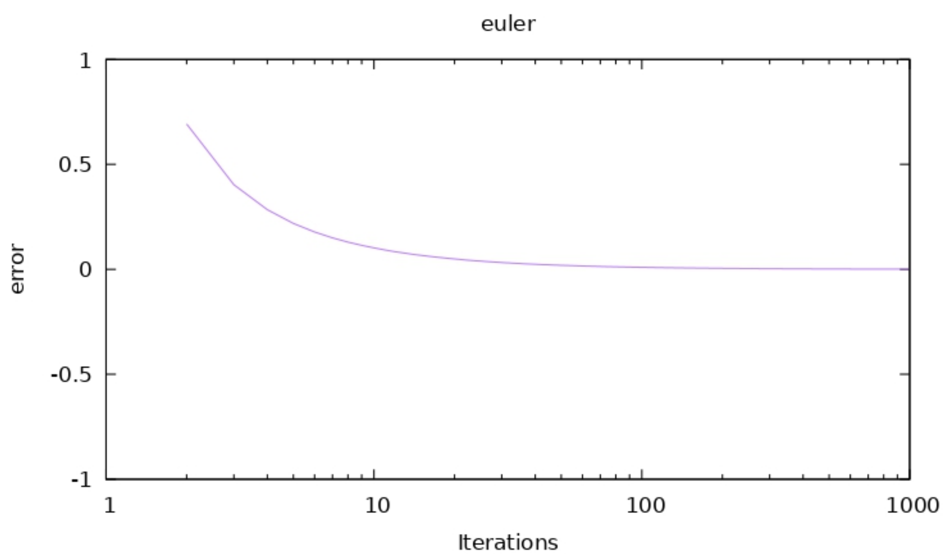
1 π approximations

There are many ways to approximate π, I will be using Euler's solution, Madhava's series, Bailey-Borwein-Plouffe's formula and Viete's formula to calculate π and take that difference to determine the efficiency of each formula and how accurate it is compared to the value in the math library.

2 Euler's Solution

I used this formula to calculate π until the change between each iteration was less then 10^{-14}.

$$\sqrt{6 \sum_{n=1}^{\infty} \frac{1}{n^2}}$$

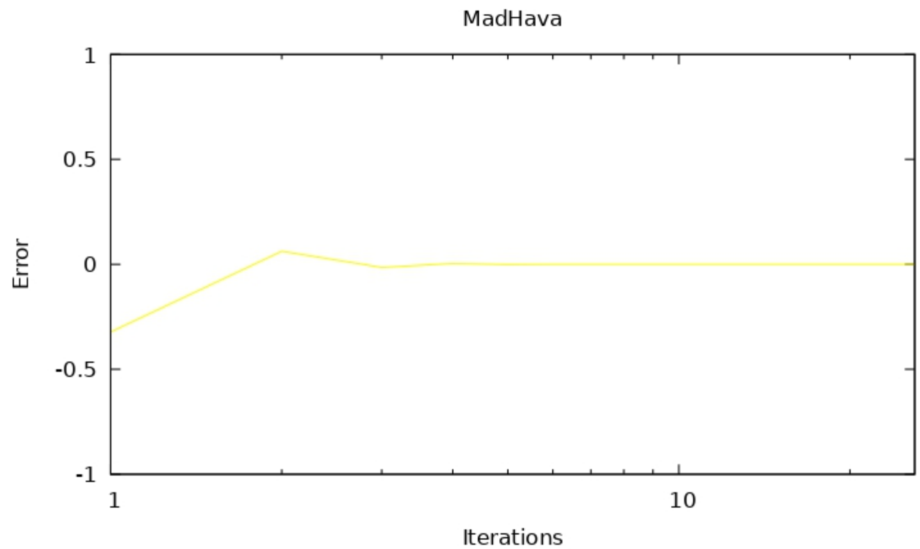


In this figure I found the difference between the math.h library π and the π I calculated using the formula above (y axis). And the amount of iterations taken is 1000 as the x axis. It is clear as the amount of iterations increase, our error decreases. When I calculated to find pi to the nearest 10^{-14} It took my compiler around 1 million iterations until termination. To find π this equation isn't the most efficient formula to use in time and in the amount of iterations.

3 Madhava Series

This is the Madhava Series:

$$\sqrt{12} \sum_{n=1}^{\infty} \frac{-3^{-n}}{2k+1}$$

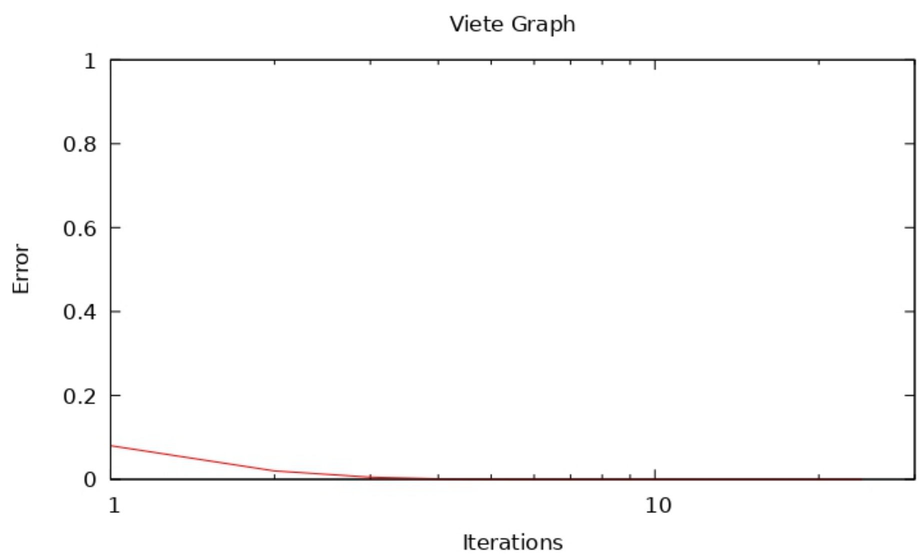


On the x axis is the number of iterations computed and on the y axis is the difference between the real value of π and Madhavas series values. The Madhava series only computed a total of 26 times which is already many times less then Euler's formula. The difference from the math librarys π and Madhava 26th term π is -0.00000000000000666134. The Madhava Series being a lot quicker then Euler's Series is accurate to the 10^{-14} place. Which is very accurate unless you need more precision then that.

4 Viete Formula

This is Vietes Formula which is used to calculate π

$$2\prod_{i=a}^{b}\frac{2}{\sqrt{2+a}}$$

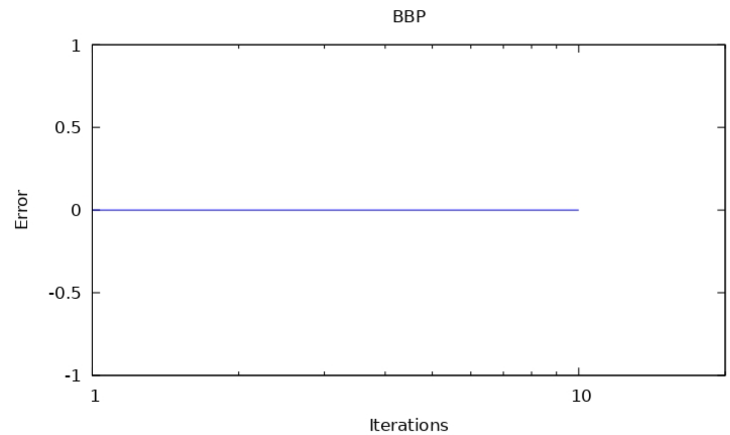


This isnt the nicest graph in the world but it does tell us that vietes formula approaches pi very quickly. The error is our x axis and our y axis is our iterations. After computing the Viete's formula I got 29 iterations with a value differenece of -0.00000000000000133227 all of these series are computed until the difference between value is less then 10^{-14}. Although Viete's series has more iterations it is still as accurate as Madhava's series. They have a difference in 3 terms so they both calculate π relatively well. If you wanted to calculate π to 10^{-50} Madhava would be a little quicker.

5 Bailey-Borwein-Plouffe's formula

Here is the Bailey-Borwein-Plouffe's formula:

$$p(n) = \sum_{k=0}^{n} 16^{-k} \times \frac{(k(120k+151)+47)}{k(k(k(512k+1024)+712)+194)+15}.$$

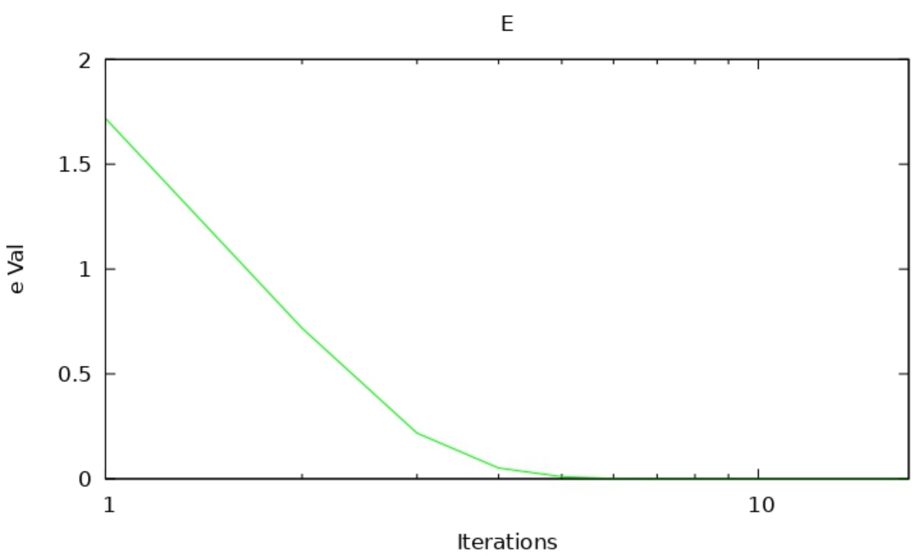


As seen above it looks like we have a constant error of 0. The difference of π's is 0 up to the 10^{-14} place. With only 11 terms being calculated. This is the fastest method and most accurate. Taking the least amount of terms and being the most accurate.

6 e approximation

To calculate e I used Leonhard Euler's series to calcualte this.

$$\frac{x^k}{k!} = \frac{x^{k-1}}{(k-1)!} \times \frac{x}{k}.$$



To calculate with 100 percent accuracy(up to the 10^{-14} place) it took 17 iterations. X axis is the difference between math library and the e value I calculated.

7 Conclusion

Overall if we are looking for the most efficiency way of approximating π we would use Bailey-Borwein-Plouffe's formula since it took the least of terms to accuracy. I dont know is this is relevant but this project really helped me understand how to create a makefile as well as familiarize myself with moving around different files using a head as kind of a navigation point. What I find amazing is that people were able to calculate π to such accuracy using pencil and paper.