Analysis – A Little Slice of π

Daniel Chang

Fall 2021

1 Introduction

In this assignment we experiment with different converging series to calculate mathematical constants. The three constants we are looking at are e, π, and square root. For each converging series, we terminate the series when the term goes below epsilon, which is set to 10^{-14}. In this paper, we will analyze the different rates at which the converging series terminate based on how many terms it takes to reach epsilon. We will look at outliers in the data and seek to explain why these outliers exist based on the series.

2 Code

Below are the functions that will be used in Analysis.

```
double pi_euler(void) {
    /*
    Calculation of pi using the Bailey-Borwein-Plouffe formula.
    */
    double sum = 0, term = 1;
    int terms3 = 1;
    while (term > EPSILON) {
        double t = (double) terms3;
        term = 1 / t / t;
        sum += term;
        terms3 += 1;
    }
    terms3 -= 1;
    return sqrt_newton(6 * sum);
}
double sqrt_newton(double x) {
    /*
    Calculation of square root using Newton's method.
```

```
    Credit for this code goes to Darrell Long
    */
    int iters = 0;
    double z = 0, y = 1;
    while (absolute(y - z) > EPSILON) {
        z = y;
        y = 0.5 * (z + x / z);
        iters += 1;
    }
    return y;
}
```

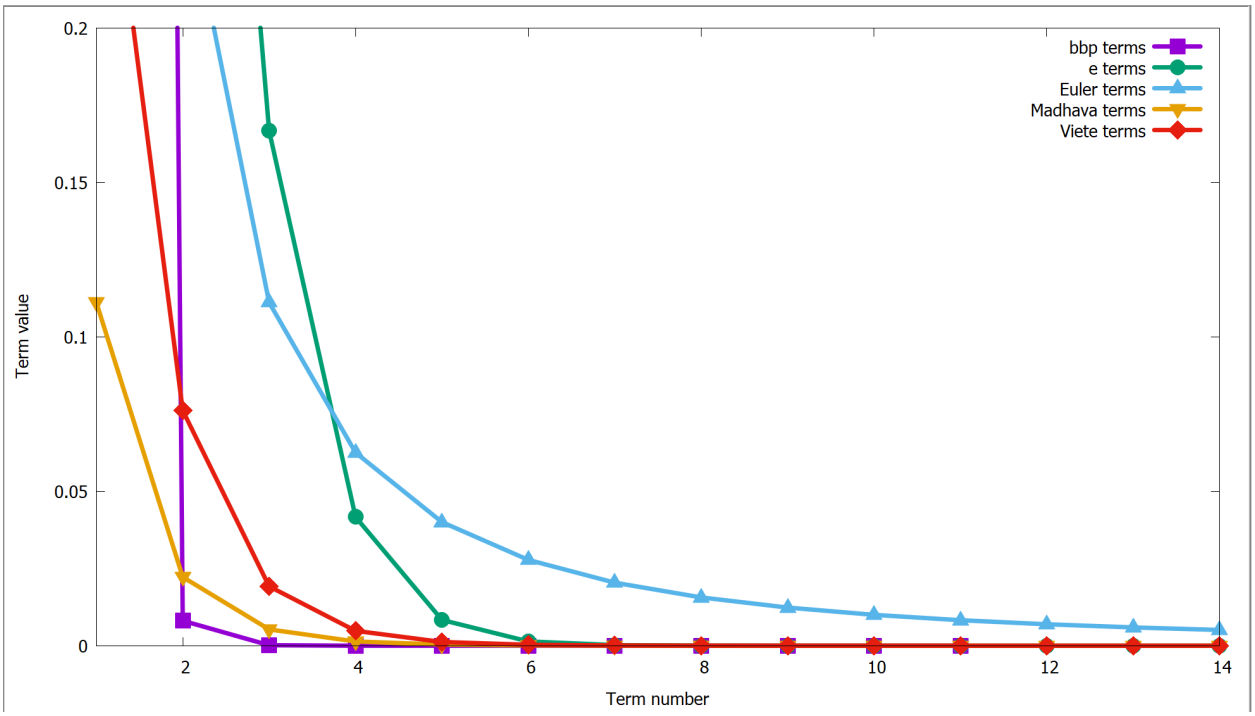
3 Analysis

Let's look at how quickly each series converges based on the number of terms:

Table 1: Number of terms for each series to converge until epsilon

Convergent Series	Number of terms
e()	18
pi_euler()	10000000
pi_bbp()	11
pi_madhava()	27
pi_viete()	23
sqrt_newton(0)	47
sqrt_newton(0.1)	7
sqrt_newton(0.2)	7

The most notable entries are pi_euler() and sqrt_newton(0). Compared to the other π convergent series, pi_euler() converges much more slowly. Below we can visualize the convergence by looking at the first few terms of each series:



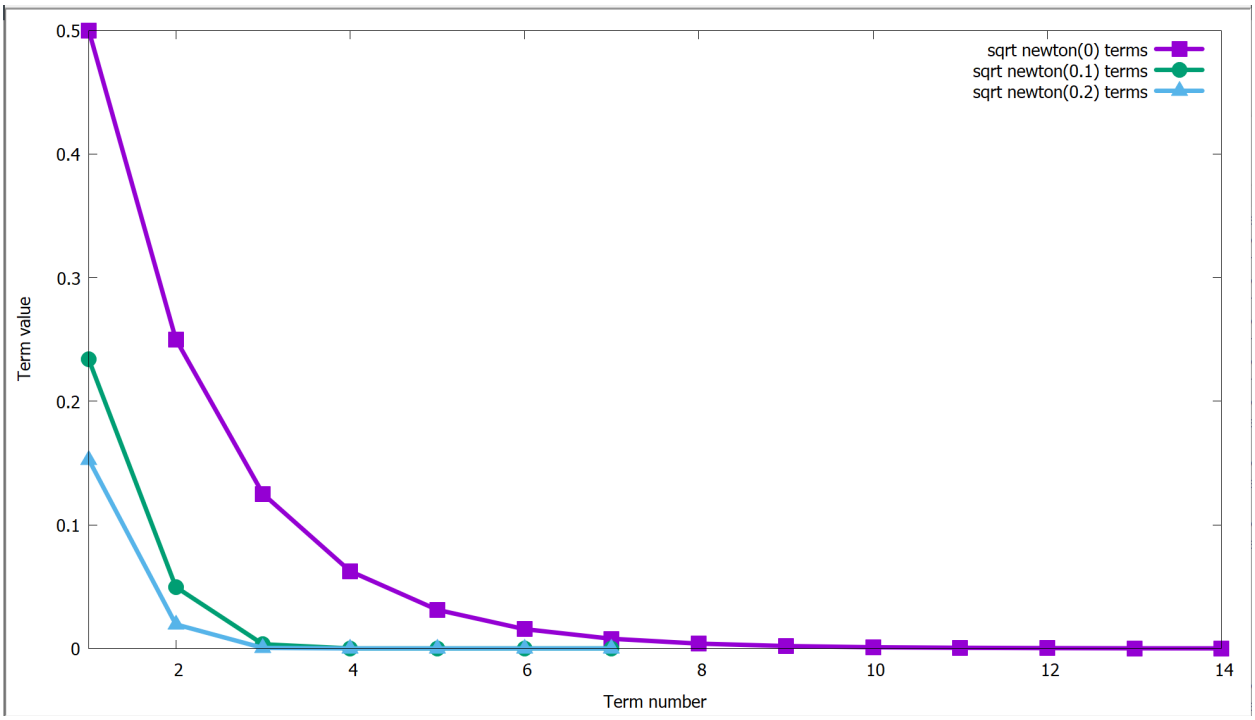
Note: The above graph uses absolute value of Madhava terms and (1 - term) for Viete terms to make the data easier to visualize.

Compared to the other series, the Euler series converges much more slowly. This can be explained by looking at Euler's equation:

$$\frac{\pi^2}{6} = \sum_{k=1}^{\infty} \frac{1}{k^2}$$

Clearly, the nth term of the summation is simply $\frac{1}{n^2}$. Since the summation terminates when the term is less than Epsilon, which is defined as 10^{-14}, then it makes sense that the summation would have exactly 10^7 terms, as the 10^7th term would be 10^{-14}.

Next, we consider why sqrt_newton(0) takes so long to terminate. In the square root approximation test we test the sqrt_newton() for values 0 to 10, and 0 is an outlier with 47 terms to terminate compared to the average of 5 to 7 terms. First, we visualize the convergence:



Compared to 0.1 and 0.2, the sqrt_newton() function calculates the square root of 0 much more slowly. This can be explained by the Newton iterate equation:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

The Newton iterate uses the slope of the line to guide the next guess. Reading further, page 10 says the function we are using is calculating the inverse of x^2 to find $\sqrt{x}$. Because the slope of the function x^2 is close to zero around $x = 0$, it makes sense that Newton's iterate will converge slower at $x = 0$.

4 Conclusion

In this paper we analyzed different converging series based on the rate at which the series converged. This was done quantitatively by terminating each series when the terms went under epsilon, set to 10^{-14}. The data shows that out of the 5 π converging series, pi_euler() converged much more slowly due to the nature of the terms of Euler's solution. Another series we analyzed was sqrt_newton(0.0), which took 47 terms to converge compared to the average of 5 to 7 terms. Again, due to the nature of Newton's iterate, the series converged much more slowly at $x = 0$. The data suggests that the best alternatives to the <math.h> library in calculating these constants is $e()$, pi_bbp(), and sqrt_newton(x) with a special case when $x = 0$.