Project 03: Numerical Estimation of π

Author: Sean McMeans

Date: 12 February 2026

Course: CS4130 - HPC

1. Optimization of Sample Size (N)

Objective: Determine the optimal value of N that balances accuracy with computational cost.

- **Observation:** As N increases, the change in the estimated value of π decreases.
- **Optimal Value:** Based on the results, the optimal N is approximately 200,000,000. At this point, the error is within acceptable bounds (<10−6) without unnecessary processing time.

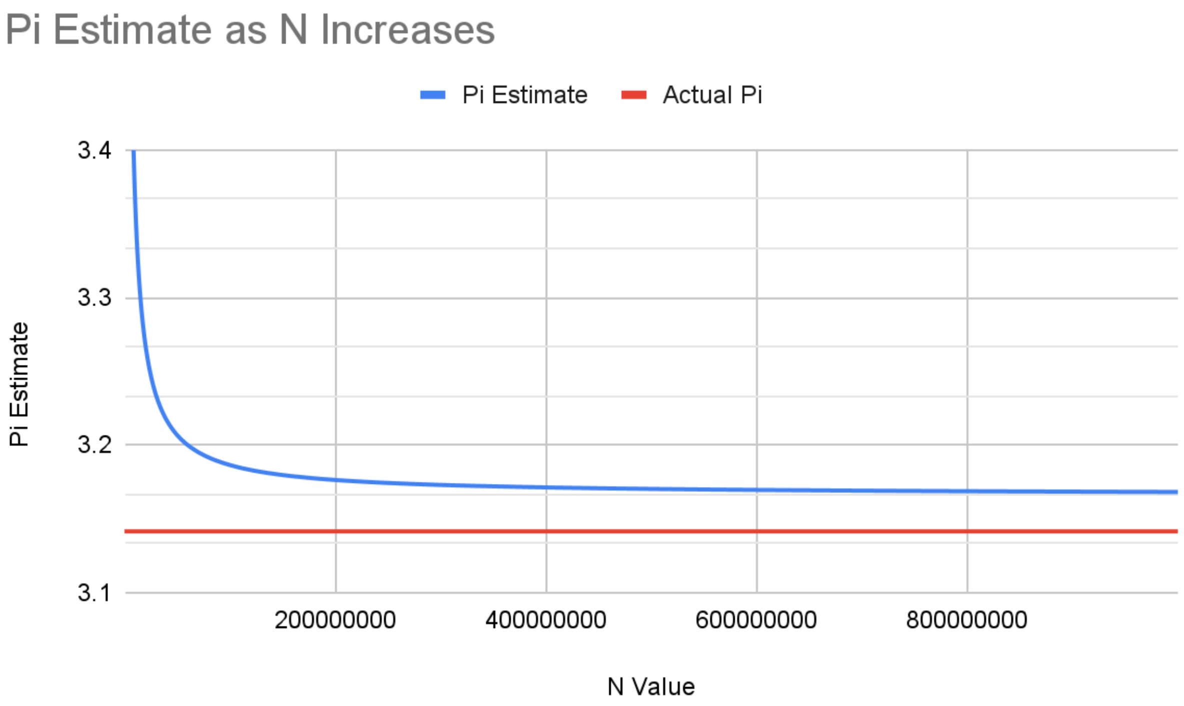

Figure 1: *Convergence of π as a function of N. The horizontal line represents π.*

2. Convergence Thresholds (ε)

Objective: Analyze the lower bounds of the tolerance parameter ε before convergence fails or becomes computationally prohibitive.

- **Minimum ε:** The algorithm successfully converges for ε values as small as **10^{-12}**.
- **Behavior at Limit:** Below this threshold, the time to run exceeds 90 minutes using 4 processors.

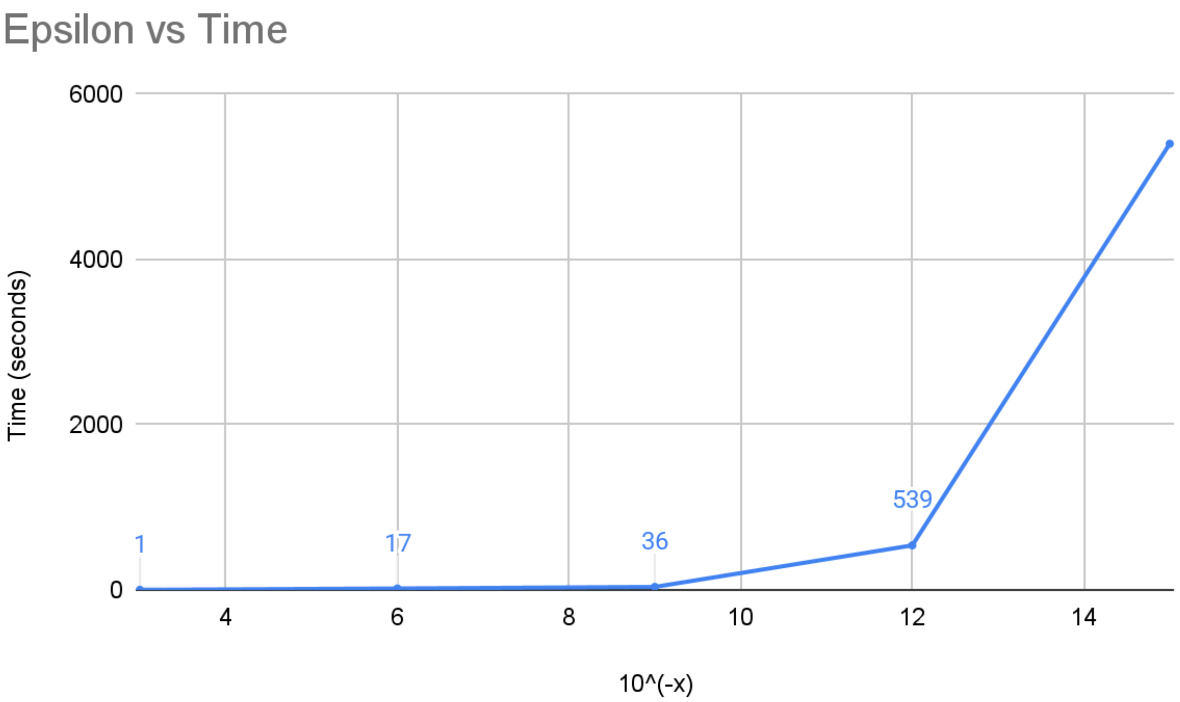

Figure 2: *The time to complete the calculation with each ε. The x axis represents the negative power of ε, so an x of 6 represents ε being 10^{-6}. Note the exponential increase in computational time as the exponent x increases (representing a smaller ε).*

3. Divergence Analysis

Question: Does the approximation of π ever converge initially and then diverge ("blow up")?

- **Observation:** No, the method remained stable.

4. Parallelization (ncpu)

Objective: Assess the impact of core count on performance.

Core count	Time Taken (Seconds)	Speed Up Factor
2	23	1.0x
4	15	1.53x
8	15	1.53x
16	11	2.09x

Figure 3: *How long it takes each number of cores to complete the calculation with an ε of 10^{-6}.*

- **Reasonable Core Count:** The best number of cores appears to be **4** cores. Beyond this, the overhead of putting all the calculations on 1 core outweighed the benefits.

5. Non-Convergence

Investigation: Reasons why the code might fail to converge to the exact value of π.

- **Observation:** The code consistently converged with a value slightly higher than π. See figure 1.
- **Reason:** Without changing the method to calculate M and P, the if statement does not include 1 as being within the circle, despite it being on the border of the circle. Additionally, the algorithm relies on randomness to achieve a value, which will become more precise as the algorithm runs, with current runs not being long enough to get an exact answer.