Numerical Methods Project 1: Calculating Pi

Jon Cohen & Rachel Sassella

Feb. 14, 2017

Monte Carlo Method

The Monte Carlo Method for estimating pi consists of throwing a number of darts (real or simulated) at a 1x1 square target inscribed with a unit circle, then multiplying the ratio of darts in the circle to total darts by four. ($\pi \simeq \frac{\textit{darts in circle}}{\textit{total darts}} * 4.0$)

To tackle this problem in Python, we used np.random.random() to generate random (x,y) coordinates for darts thrown at the unit square. Large numbers of darts were needed to calculate pi to the specified precision. To avoid problems of storage, each dart was checked upon generation to see whether or not it was in the unit circle. This allowed us to store only two numbers (number of darts in unit circle, total number of darts thrown) rather than a massive array of darts. At the end, the ratio of darts in the circle to total darts was multiplied by 4 to acquire an estimate of pi.

A run of 1 billion darts reliably provided four digits of accuracy. (See Appendix A.)

Riemann Sum

By discretizing the integral $\pi = 4\int_0^1 \sqrt{1-x^2}dx$ we can run a series of simulations that will estimate the value for pi. This is done by approximating the integral with a Riemann sum, which is of the form $\int_0^1 f(x)dx \approx \Delta x \sum_{i=1}^{n} f(x_i)$, where $\Delta x = 1/n$ and $x_i = \Delta x \cdot i$. Thus, we must implement an algorithm that computed the Riemann sum $\Delta x \sum_{i=1}^{n} 4\sqrt{1-x_i^2}$.

To implement the sum as an algorithm, we created a function intpi(n). A loop calculates the values $x_i = \Delta x \cdot i$ and $4\sqrt{1-x_i{}^2}$ for i values from 1 to n. The calculated values of $4\sqrt{1-x_i{}^2}$ are summed after each iteration, and a final estimate is obtained when the iterations are completed. The intpi(n) function was tested for 10, 50, 100, 500, 1000, and 5000 iterations.

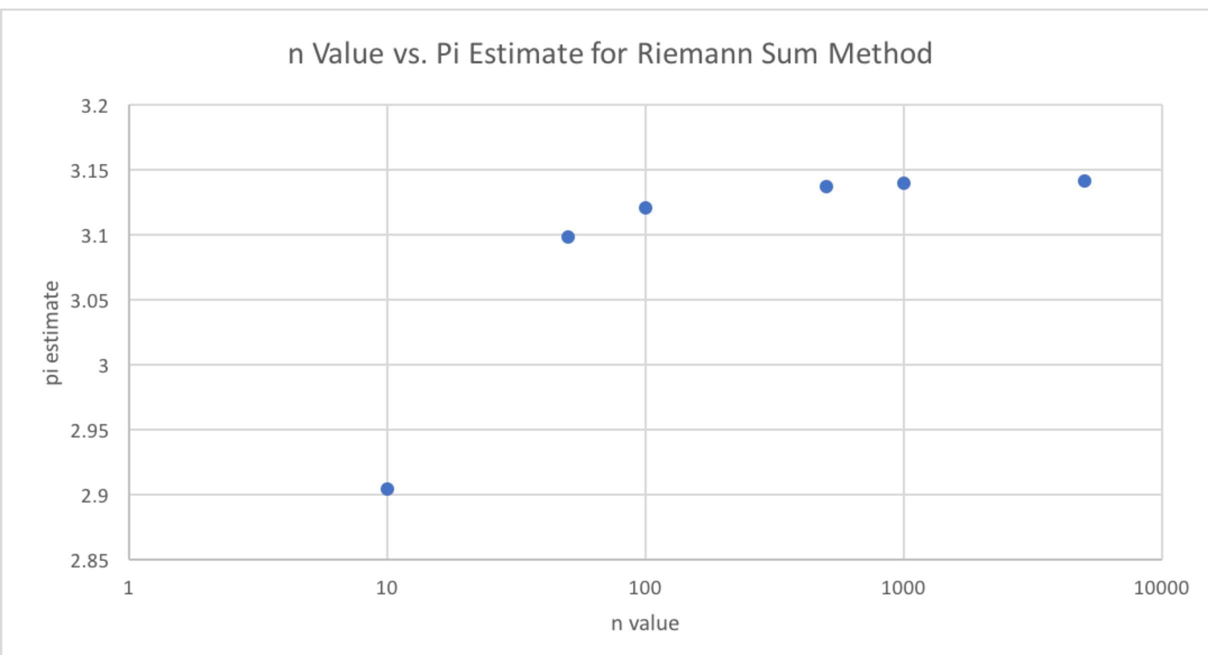
The true error and true percent relative errors for each estimate of pi was calculated after each run of the algorithm, using the formulas $E_t = |\pi - \textit{approximation}|$ and $\varepsilon_t = \frac{E_t}{\textit{true value}} \times 100\%$. After the true relative error was calculated for the first and second run of the function, the approximate relative error could be calculated. This was done using the equation $\varepsilon_a = \frac{\textit{present approximation} - \textit{previous approximation}}{\textit{previous approximation}} \times 100\%$. The approximate relative percent error was computed for every n value after the first.

The results from the full set of simulations are listed in Appendix B.

The true relative error vs the step size is plotted below:

The linear appearance of the plot of true error and step size on a log-log scale implies that there is an exponential relationship between the two variables. However, the slope seems pretty close to one, suggesting that the relationship is close to a genuine linear function. A decrease in step size corresponds to a proportional decrease in true error.

A higher number of iterations will also decrease the error. As the number of iterations increases, though, each subsequent change in the estimate is smaller. In the following figure, the estimates asymptotically approach the true value of pi as number of iterations increases.



Notice that when we view the increase in our pi estimates with respect to the n value used to the algorithm, we see a saturation trend in the graph. As we continue to increase n, the benefit for getting our estimate close to the actual value of pi will continually become more minute.

Taylor Series

Another way to calculate pi uses the fact that $\pi = 4 * arctan(1)$. However, we don't want to include any trig functions in our code. We can approximate the arctan of 1 using the Taylor series for an arctangent: $arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} \ldots$

The function tsp1(n) uses this approximation. Starting with the x term, the function calculates the sign, denominator, and power of the current term, then sums it with all the terms found so far. Each additional term improves the accuracy of the estimate of pi. (See Appendix C for output for 1 to 20 terms.)

The Machin formula is given by $\pi = 16 * arctan(\frac{1}{5}) - 4 * arctan(\frac{1}{239})$. The function tsp2(n) used essentially the same method of calculation as the above arctan function, but with two Taylor series used to generate estimates. (See Appendix D for output for 1 to 20 terms.) This was much more effective than the first arctan method used in tspl(n). As can be seen in the output, relative error reaches zero by 20 iterations. In this case, the Machin formula computes pi with 14 decimal places of accuracy. This increased accuracy is due to the small fractions involved in Machin. While the first term will converge slower than the second term, both fractions in the arctans will increase rapidly to very small and accurate values. (The arctan(1) method converges much more slowly in part because powers of 1 are always 1.) This leads to much more accuracy in computing the Taylor series with a finite number of terms, and helps curtail truncation error.

Appendix A: Output for Monte Carlo Method

Output of Monte Carlo Method for calculating pi (lab1_montecarlo.py)
Ten iterations, with 4 digits of precision
(Real value of pi 3.14159 rounds to 3.142)

Iteration 1
Full estimate 3.14170634
Rounded estimate 3.142

Iteration 2
Full estimate 3.141585616
Rounded estimate 3.142

Iteration 3
Full estimate 3.141619024
Rounded estimate 3.142

Iteration 4
Full estimate 3.141536372
Rounded estimate 3.142

Iteration 5
Full estimate 3.141642284
Rounded estimate 3.142

Iteration 6
Full estimate 3.141603904
Rounded estimate 3.142

Iteration 7
Full estimate 3.1415775
Rounded estimate 3.142

Iteration 8
Full estimate 3.141545868
Rounded estimate 3.142

Iteration 9
Full estimate 3.141635076
Rounded estimate 3.142

Iteration 10

Full estimate 3.1416339
Rounded estimate 3.142

Appendix B: Output for Riemann Sum Method

n = 10
delta_x = 0.1
pi estimate = 2.90451832625
true error = 0.237074327341
true relative error = 7.54631021532 %

n = 50
delta_x = 0.02
pi estimate = 3.0982685111
true error = 0.0433241424913
true relative error = 1.37905028654 %
approximate relative error = 6.2534988222 %

n = 100
delta_x = 0.01
pi estimate = 3.12041703178
true error = 0.0211756218107
true relative error = 0.674040976845 %
approximate relative error = 0.709793609475 %

n = 500
delta_x = 0.002
pi estimate = 3.137487477
true error = 0.00410517658766
true relative error = 0.130671829238 %
approximate relative error = 0.544080107035 %

n = 1000
delta_x = 0.001
pi estimate = 3.13955546691
true error = 0.00203718667877
true relative error = 0.0648456659855 %
approximate relative error = 0.0658688763642 %

n = 5000
delta_x = 0.0002
pi estimate = 3.14118932743
true error = 0.000403326159208

true relative error = 0.0128382703832 %
approximate relative error = 0.0520140733096 %

Appendix C: Output for arctan() Method (*tsp1(n)*)

Results for arctan method for finding pi. Number of terms refers to the number of Maclaurin series terms used to get the estimate.

1 terms
estimate of pi: 4.0

2 terms
true relative error: 15.1173636843 %
approximate relative error: 50.0 %
estimate of pi: 2.66666666667

3 terms
true relative error: 10.3474272104 %
approximate relative error: 23.0769230769 %
estimate of pi: 3.46666666667

4 terms
true relative error: 7.84170914298 %
approximate relative error: 19.7368421053 %
estimate of pi: 2.89523809524

5 terms
true relative error: 6.30539690963 %
approximate relative error: 13.3079847909 %
estimate of pi: 3.33968253968

6 terms
true relative error: 5.2695080425 %
approximate relative error: 12.2187742436 %
estimate of pi: 2.97604617605

7 terms

true relative error: 4.52464230161 %
approximate relative error: 9.37018307688 %
estimate of pi: 3.28373848374

8 terms
true relative error: 3.96362132995 %
approximate relative error: 8.83859194726 %
estimate of pi: 3.01707181707

9 terms
true relative error: 3.52602305084 %
approximate relative error: 7.2345523957 %
estimate of pi: 3.25236593472

10 terms
true relative error: 3.17523771092 %
approximate relative error: 6.92101958563 %
estimate of pi: 3.04183961893

11 terms
true relative error: 2.8878077402 %
approximate relative error: 5.89287067563 %
estimate of pi: 3.23231580941

12 terms
true relative error: 2.64801636735 %
approximate relative error: 5.68640093502 %
estimate of pi: 3.05840276593

13 terms
true relative error: 2.44494181159 %
approximate relative error: 4.9714100949 %
estimate of pi: 3.21840276593

14 terms
true relative error: 2.27076020594 %

approximate relative error: 4.82527238263 %
estimate of pi: 3.07025461778

15 terms
true relative error: 2.1197209828 %
approximate relative error: 4.29934702767 %
estimate of pi: 3.20818565226

16 terms
true relative error: 1.98750335506 %
approximate relative error: 4.19051088223 %
estimate of pi: 3.0791533942

17 terms
true relative error: 1.87079829565 %
approximate relative error: 3.78744617227 %
estimate of pi: 3.20036551541

18 terms
true relative error: 1.76702897502 %
approximate relative error: 3.7032650369 %
estimate of pi: 3.08607980112

19 terms
true relative error: 1.67415898373 %
approximate relative error: 3.38452561904 %
estimate of pi: 3.19418790923

20 terms
true relative error: 1.59055779765 %
approximate relative error: 3.31748327021 %
estimate of pi: 3.09162380667

Appendix D: Machin Formula Output (tspi2(n))

n = 1

pi estimate = 3.18326359833
true relative error = 1.32642736763 %

n = 2
pi estimate = 3.14059702933
true relative error = 0.0316917046071 %
approximate relative error = 1.35854961977 %

n = 3
pi estimate = 3.14162102933
true relative error = 0.00090322770551 %
approximate relative error = 0.0325946379088 %

n = 4
pi estimate = 3.14159177218
true relative error = 2.80560757861e-05 %
approximate relative error = 0.000931284042578 %

n = 5
pi estimate = 3.1415926824
true relative error = 9.17197403797e-07 %
approximate relative error = 2.89732729242e-05 %

n = 6
pi estimate = 3.14159265262
true relative error = 3.10188048086e-08 %
approximate relative error = 9.48216208899e-07 %

n = 7
pi estimate = 3.14159265362
true relative error = 1.07467408737e-09 %
approximate relative error = 3.20934788957e-08 %

n = 8
pi estimate = 3.14159265359
true relative error = 3.78980760045e-11 %
approximate relative error = 1.11257216337e-09 %

n = 9
pi estimate = 3.14159265359
true relative error = 1.37117246268e-12 %
approximate relative error = 3.92692484671e-11 %

n = 10
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 1.39944405984e-12 %

n = 11
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 5.65431943371e-14 %

n = 12
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 13
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 14
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 15
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 16
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 17
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 18
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %

approximate relative error = 0.0 %

n = 19
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %

n = 20
pi estimate = 3.14159265359
true relative error = 2.82715971686e-14 %
approximate relative error = 0.0 %