Modeling π and e

Through Approximation and Application

Adam Mahmoud and Jack Galetti

π: Five Methods of Approximation

Monte Carlo

Using random numbers within a bounded region, we can use the ratio of points that lie on a circle’s circumference to diameter to estimate pi.

Perimeter

Simultaneously increasing and summing the sides of an n-gon inscribed within a circle of radius one half will eventually yield the true value of pi.

Leibniz

Takes an infinite series of fractions with odd denominators and alternating signs to approximate the convergence of arctangent.

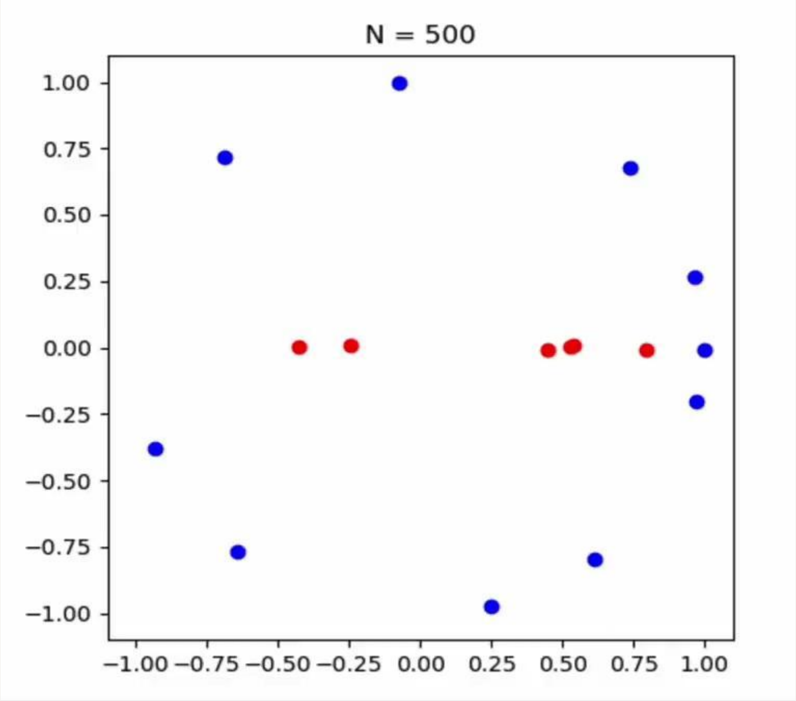
Wallace Product

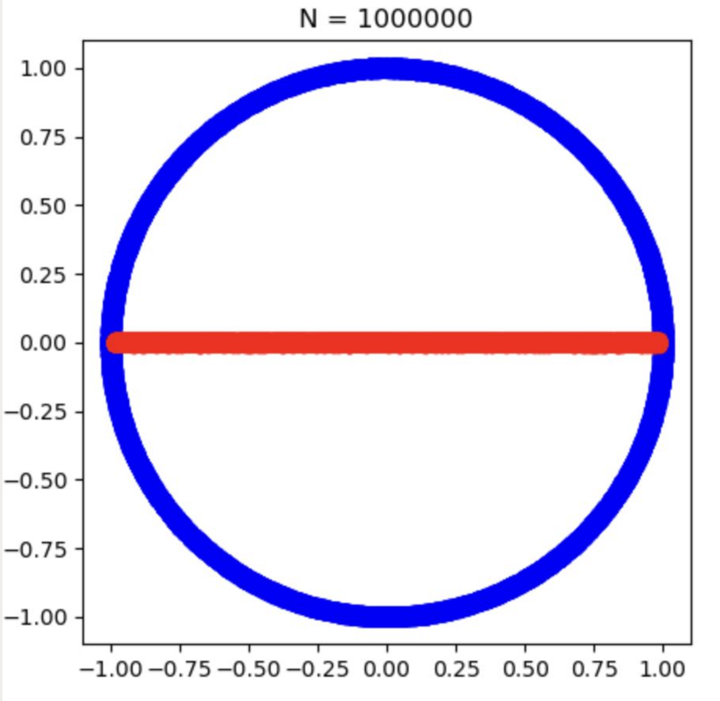
Multiplies a sequence of even and odd integer pairs and divides them by the product of the preceding odd integer to achieve a close value of pi.

Buffon Needle

Counts the number of needles dropped randomly over a bounded region that fall on top of its upper and lower bounds, which has a probability of pi.

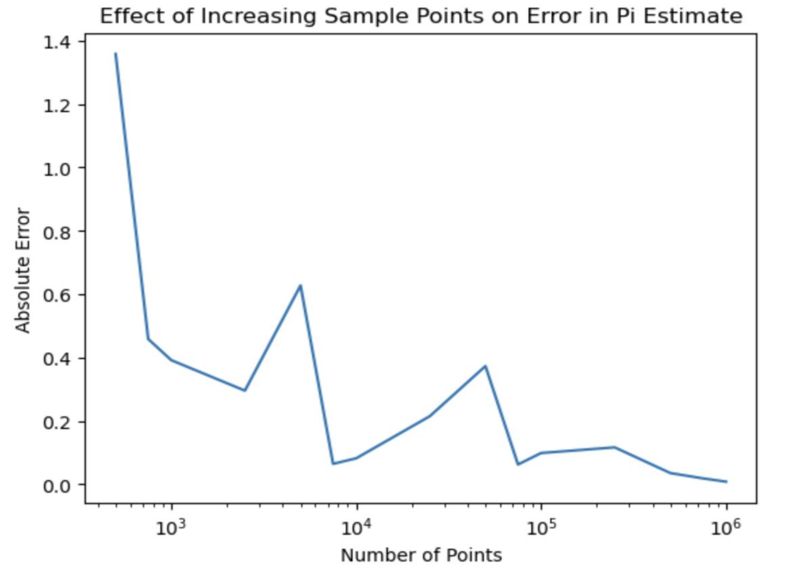
Monte Carlo


N = 500
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


N = 1000000
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

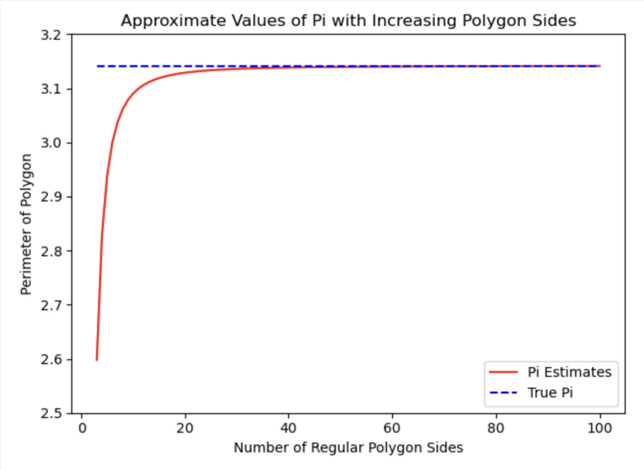
Monte Carlo (cont.)

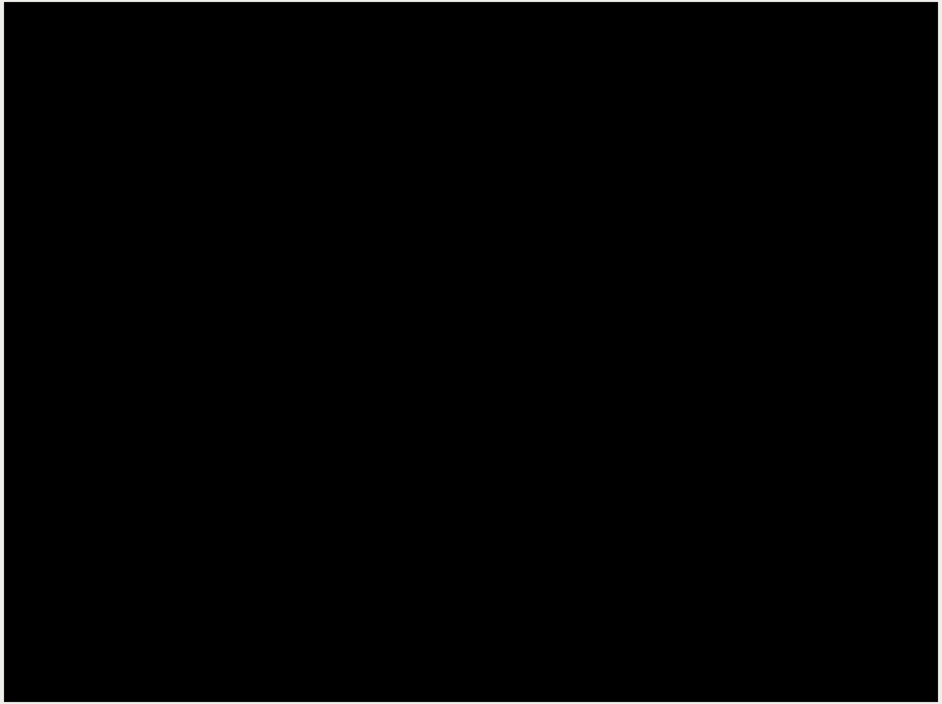
N:	Estimated Pi:	Absolute Error:
500	6.3333	3.1917
750	7.0	3.8584
1000	6.4	3.2584
2500	4.2083	1.0667
5000	3.48	0.33841
7500	3.1194	0.02219
10000	3.2614	0.11977
25000	2.9097	0.23185
50000	3.1316	0.010014
75000	3.1085	0.033109
100000	3.179	0.0374
250000	3.2001	0.05849
500000	3.0751	0.066447
750000	3.1524	0.010759
1000000	3.199	0.057415



Above: graph of absolute error vs number of points shows the increasing level of accuracy in our approximation as n increases. **Left**: number of random points dropped in comparison with the approximation of pi it achieved and its absolute error.

Polygon Perimeter



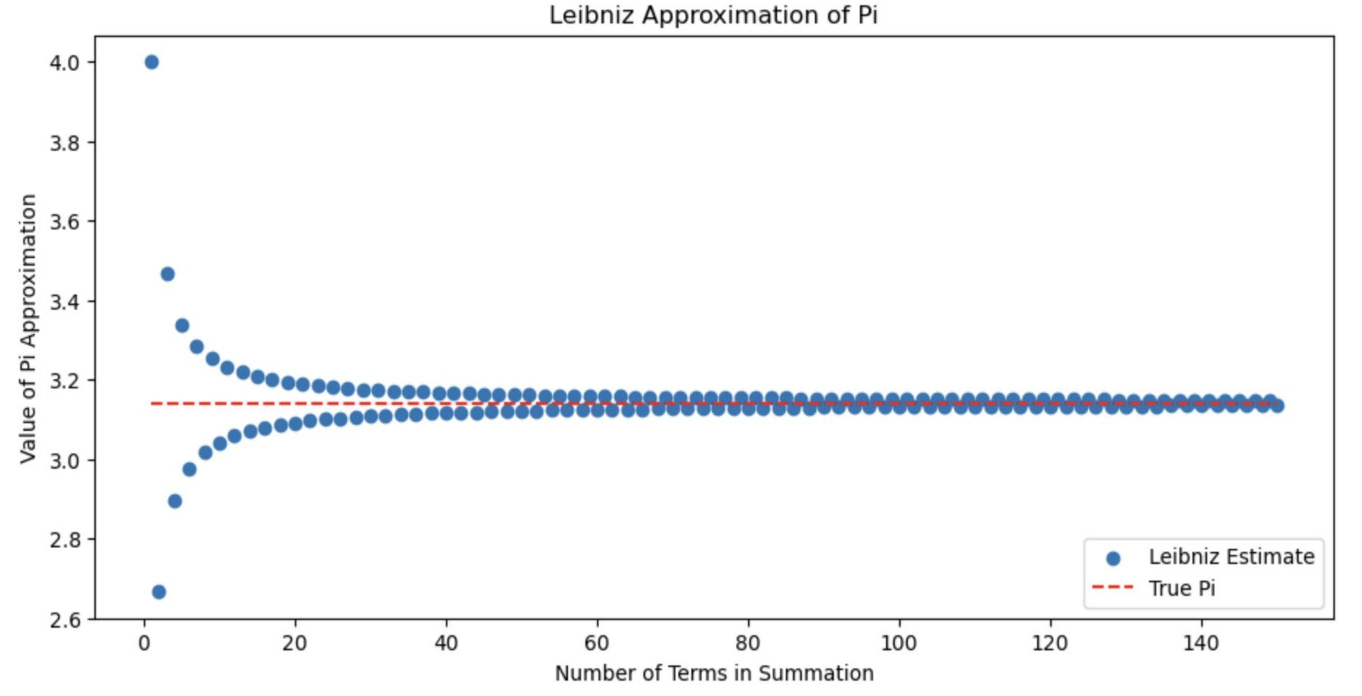


Above: this graph illustrates the rapid convergence and high accuracy that this method yields in relation to the value of true pi.

Right: as the number of sides increases, the polygon’s perimeter approaches that of a circle’s. Because a circle’s perimeter is described by 2πr, by setting r to one half (as we did), as the number of sides increases to infinity, the perimeter will tend to pi.

Leibniz Formula

$$\frac{\pi}{4} = \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$$

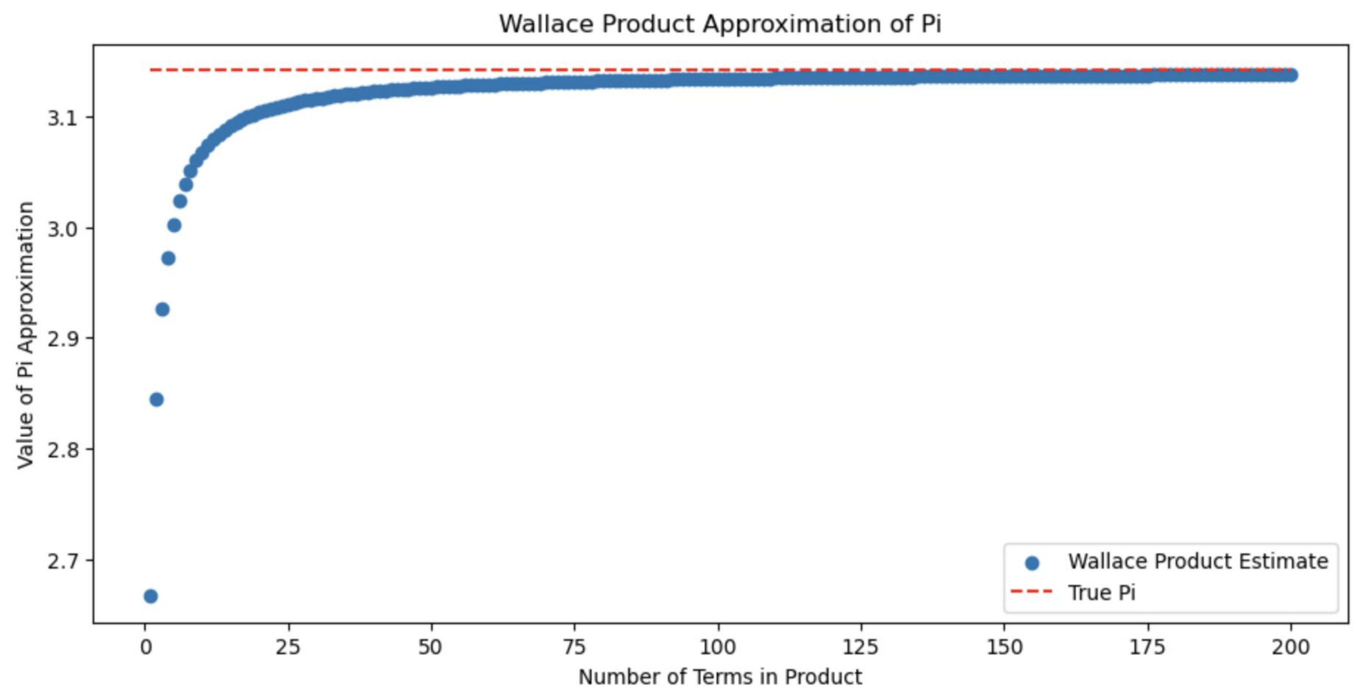


Left: this graph shows the summation that yields pi. The points lie both above and below true pi because the terms contain alternating signs that waver around the true value. In our code, we multiplied our raw function for the Leibniz formula by four since the formula itself approximates the convergence of arctangent, which is pi divided by four.

Wallis Product

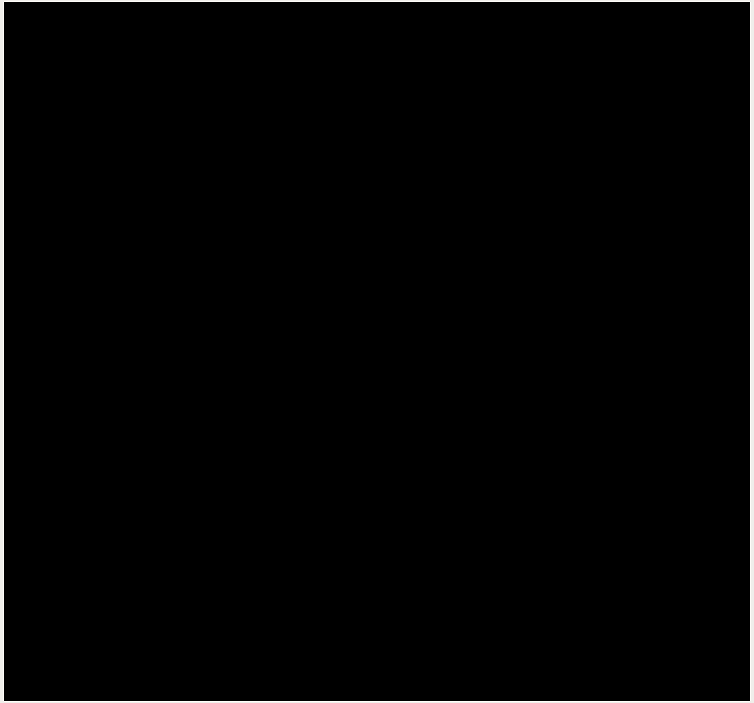
$$\pi = 2 \prod_{i=1}^{\infty} \frac{(2i)^2}{(2i-1)(2i+1)}$$

Right: this method involves an infinite product of ratios of squares of certain integers to other integers one unit above and below the numerator. This method converges quickly and yields a highly accurate estimation of pi that can be computed relatively efficiently. Similar to Leibniz, this formula extends loosely from arctangent.

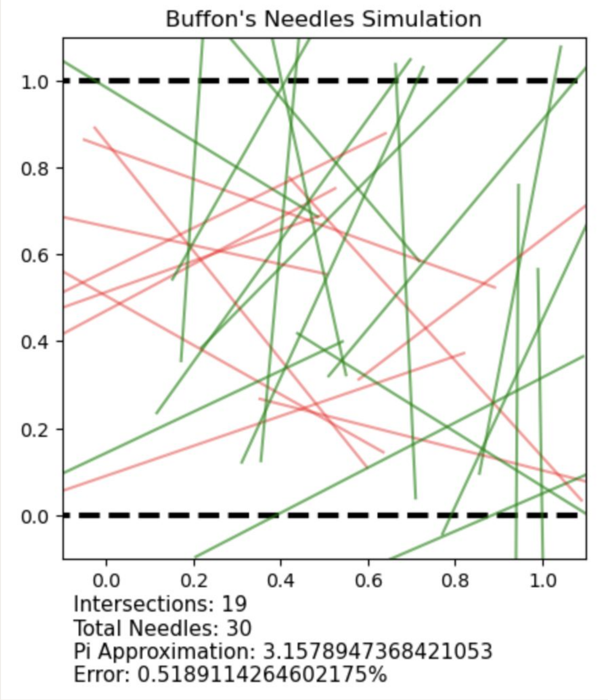


Buffon Needle

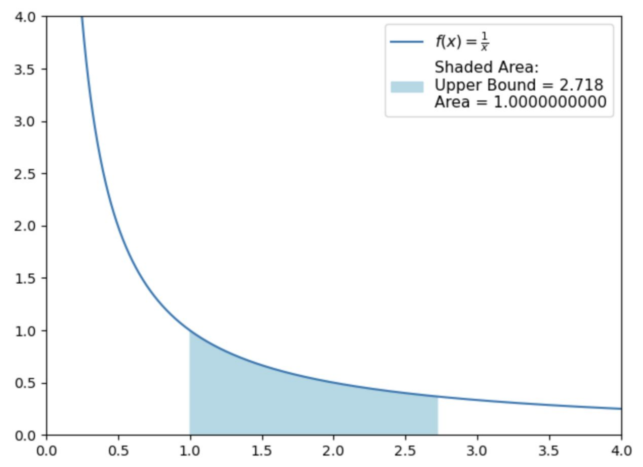
$\pi \approx (2L/D)(\#\text{ of needles}/\#\text{ of hits})$



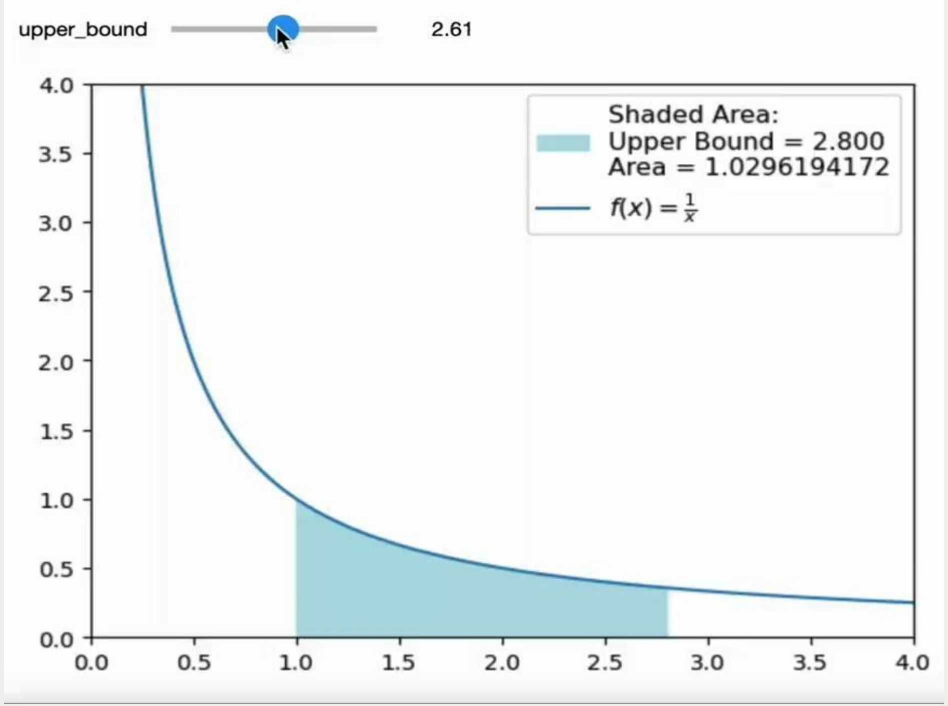
Left: this illustrates the increased accuracy of pi that comes with an increased number of needles.
Right: needles that cross the upper and lower bounds of the enclosed region are represented in green, and those that do not are red. The ratio between the green needles and the total is a probability that is actually an approximation of pi.



e: Approximation by Integration Visualized



Above: this illustrates what we modelled our approximation off of — the integration of 1/x from 1 to e which yields an area of 1.
Right: this interactive slider we created essentially shows our process to approximate e. We kept the left hand bound at 1 and varied our right hand bound until we got an area of (or very close to) 1, at which point that bound is an incredibly close approximation of e.



e: Approximation by Integration

Process:

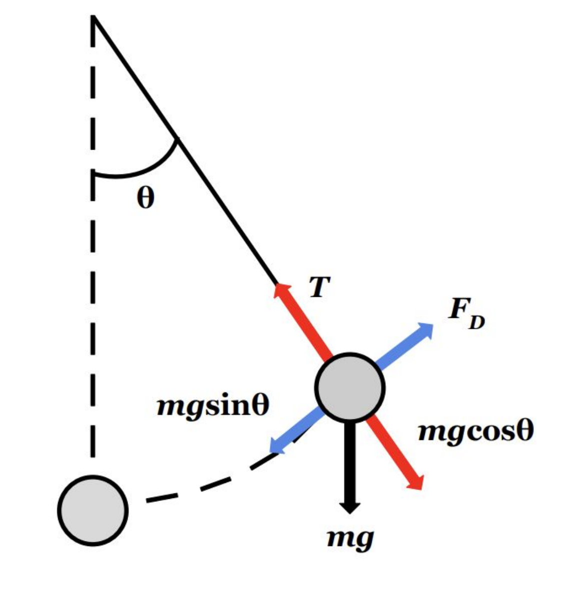
- We took advantage of the fact that the **definite integral of 1/x from 1 to e has an area of 1**
- Using right hand, left hand, and trapezoidal integration methods with 1 million "steps" within the range of 1 to 3, we determined **at what point the area was equal to 1 and at what x value that occurred**
- For each method, we had three values that were within 0.0001% of an area of 1, our final result for each method was the **average of these values**

Results	
True value	2.718281828459045
Left Hand	**2.7182**77718277718
Right Hand	**2.718281**718281718
Trapezoid	**2.7182**79718279718
Most Accurate	Surprisingly, it was our **right hand** integration of 1/x that yielded the closest value of e — accurate to 6 decimal places.

Application: Damped Pendulum

$$\frac{d^2\theta}{dt^2} + \frac{b}{m}\frac{d\theta}{dt} + \frac{g}{L}\sin\theta = 0$$

$$\theta(t, c_1, c_2) = Ae^{-\gamma t}\left(c_1\cos(\alpha t) + c_2\sin(\alpha t)\right) \text{ where } \alpha = \omega\sqrt{1-\gamma^2},\ \omega = \sqrt{\frac{g}{L}},\text{ and } \gamma = \frac{b}{2m}$$



Objectives:

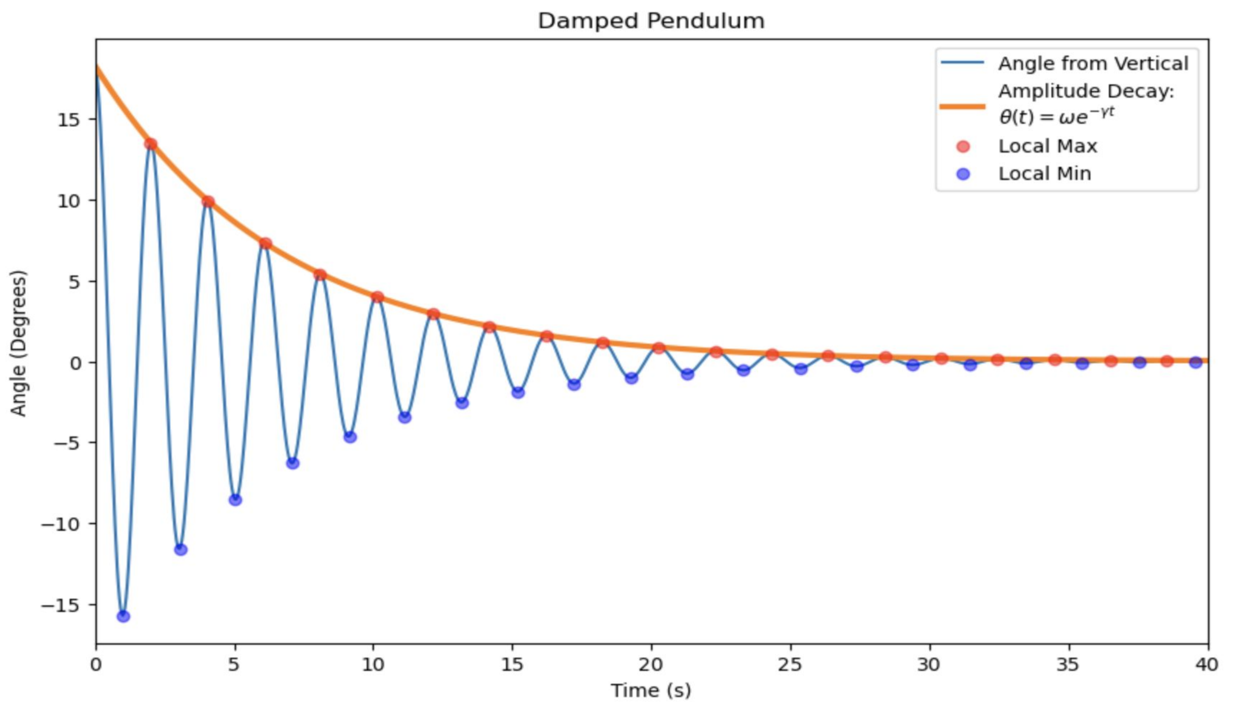
- Model damping oscillation of pendulum's angle for underdamped parameters
- Show declining arc length subtended by the pendulum with each period

Pendulum Properties:

- Length = 1 m
- Mass = 1 kg
- Damping Coefficient = 0.3 N*s / m
- Gravity = 9.81 m / s^2

NOTE: we used our right hand approximation of e and our polygon-perimeter pi value to model our pendulum

Application: Damped Pendulum



Sources

- “Matplotlib.animation.” *Matplotlib*, https://matplotlib.org/stable/api/animation_api.html.
- “WolframAlpha.” WolframAlpha Math Input, Wolfram, https://www.wolframalpha.com/.