Statistical Pi Simulation by Java

Katherine Lim
Eleanor Murray Fallon Middle School
3601 Kohnen Way, Dublin, CA 94568
katherinelim65@gmail.com

Abstract

Pi, the ratio of a circle's circumference to its diameter, is an important but irrational number in math. It is a frequently used number which can be used to find the circumference and area of a circle, as well as the surface area and volumes of spheres and cylinders. The purpose of this project to find pi with higher accuracy. A Java Monte Carlo simulation method is developed to generate random points in the coordinate. By counting the number of points inside and outside the circle, we can calculate the Pi. The sample size can be controlled in the Java program. For each sample size, it was repeated for twenty times see how the relationship between accuracy and the sample size. In the end, the Monte Carlo Method was revised to further improve the Pi accuracy.

Keywords
Monte Carlo Method, Java

1. Introduction

The first thing that was done when starting this project was to find why the accuracy of pi is so important. An example of this would be in the very place that humans all live on, earth. This planet's radius is about 3959 miles. 2 different estimates of pi were used on different accuracy levels(3.14 and 3.1416) to calculate an approximation of the earth's circumference. The two results were 24862.52(about 39 miles difference) and 24875.1888(26 miles difference). Just by 4 digits of pi, the accuracy of the circumference is increased.

2. Objectives

When starting this project, 3 questions were asked. One, is simulating pi by using java possible? Two, what was the relationship between the sample size value in the simulation and the result? Three, how many digits of pi do people really need? Later on as the project was continued, another goal was added- to shorten the code while increasing the accuracy.

3. Methods

3.1 Overview

To achieve the goal of using Java to simulate pi, the first step was to find the most effective Method to implement into Java while still getting accurate results. After researching, I narrowed my choices down to 2. The first one was the traditional Method. This Method involved cutting up a circle into many small(near) triangle pieces. After rearranging, the user is left with a(near) trapezoid. Then one side of the trapezoid was divided by the other, leaving the result with pi(this ratio was close to that of the circumference of a circle to the diameter). The second Method was the Monte Carlo Method, which was later picked because it was easier and faster to implement into java. Later on, 2 other Methods, the sphere and upper triangle, were devised to attempt increasing the accuracy of the result while shortening the code.

3.2 Monte Carlo Method

The Monte Carlo Method is a Method that generates an accurate model of responses based on randomly generated values. In the Java code, a preselected sample size of dots was randomly generated on a quarter of a circle encased in a square as indicated in Figure 1. This is because the area of a circle to the area of a square is π/4 as shown in Equation 1. The area of the circle is represented by the total number of red dots inside the circle in Figure 1, which is the variable "N_c". The area of the square is represented by the total number of dots inside the square (red and blue), which is represented by "M_s", which is the preselected sample size. After all the values are registered into the variables, the values are then plugged in to the Equation 1 below, where the output is named "pi".A Java program code was written to run the above calculation using the Monte Carlo Method. The code was run the 20 times before changing the sample size.

Four values of the varying sample sizes were usedto run the Java code; 10,000,000, 1,000,000, 10,000, and 100. For each of the sample size, twenty outcomes were generated. After running the code four times for different sample sizes, total 80 values of "pi" were generated, and the result is summarized in Section 4, Table 1. As indicated in Table 1, the decimals of pi increases with the sample size. The assessment of this result is discussed in Section 4.

$$\pi = \frac{4\times Area\ of\ circle}{Area\ of\ Square} = \frac{4\times(\pi\times r^2)}{(2\times r)^2} = \frac{4\times N_c}{M_s} \qquad \text{(Equation 1)}$$

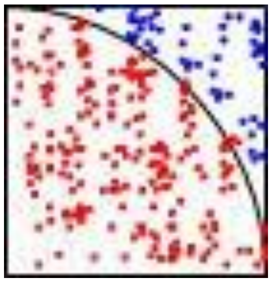
Figure 1. Monte Carlo Method

3.3 Sphere Method

It was concluded that for the Monte Carlo Method, the accuracy was unsatisfactory and the code length was too long. After brainstorming multiple ideas to improve the accuracy of resulting pi, the used Method was later named "Sphere Method". In the previous Method, only two dimensions were used to generate pi. This Method simply added a new third dimension by generating a "z" coordinate, thus "Sphere". The number used to multiply the Sphere was determined using the same Method. Refer to equation 2 below, the ratio of the Area of a Sphere to the Area of a Cube is π/6. Therefore, the number that is used to multiply the equation is 6. The volume of the sphere is the represented by number of dots within the sphere, which is variable "N_s". The volume of the cube is represented by the total amount of dots in the cube, which is variable "M_c". This is the preselected sample size. After all the values are stored into the variables, the values are then put in to the Equation 2 below, where the output is named "pi".

The previous Java program code was modified to run the Sphere Method. The code was running for 20 times for each of the four sample sizes (10,000,000, 1,000,000, 10,000, and 100).We were left with, like previously, 80 resulting digits of pi. These results were recorded in Table 3 then analyzed through SPSS in Section 4.

$$\pi = \frac{6 \times Volume\ of\ Sphere}{Volume\ of\ Cube} = \frac{6 \times \left(\frac{3}{4}\pi r^3\right)}{(2r)^3} = \frac{6 \times N_s}{M_c} \qquad \text{(Equation 2)}$$

3.4 Upper Triangle Method

In order to achieve both goals, another adaptation of the original Monte Carlo Method was devised. It was later named the Upper Triangle Method. This Method pushed to coordinates up into the upper triangular half of the space, hence the name "Upper Triangle Method". The dots in the lower right triangle are not counted because they are away from the circumference of the circle. This also improved the accuracy of the outcome, because there is the same sample size in a smaller sample area, which resulted in a higher concentration, and therefore in better accuracy.

A new Java code was written to run the Upper Triangle Method as described above. Same as Monte Carlo Method, the code was running 20 times for each of the four sample sizes. Four values of the varying sample sizes were used to run the Java code; 10,000,000, 1,000,000, 10,000, and 100.After generating the eighty values with the same four varying sample sizes, the results were put into SPSS and analyzed.Table 4 in section 4 contains the analysis results.

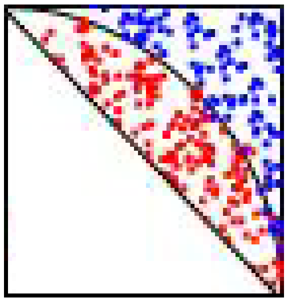

Figure 2. Upper Triangle Method.

4. Results

4.1 Monte Carlo Method

As described in 3.2 above, Table 1 below summarizes the result of "Pi" using the Monte Carlo Method. The decimals of pi in Table 1 increases with the sample size, and it can be concluded that the sample size impacts the accuracy of the output.

Table 1. Monte Carlo Method: Results

M= 10,000,000	**M= 1,000,000**	**M= 10,000**	**M=100**
3.1420304	**3.144392**	**3.1176**	**3.24**
3.142086	3.14192	3.1252	3.32
3.141056	3.140932	3.1352	2.96
3.1419224	3.140056	3.1228	3.08
3.1414736	3.141332	3.1396	3.0
3.1425504	3.140756	3.1484	3.12
3.1415148	3.139512	3.1476	3.24
3.1419804	3.138656	3.1416	3.08
3.141442	3.141964	3.1192	2.88
3.1414804	3.143	3.124	3.28
3.1419784	3.142388	3.1516	3.44
3.1426356	3.141192	3.1716	3.08
3.140472	3.145368	3.1392	2.88
3.1423584	3.141288	3.1444	2.92
3.1414756	3.139404	3.124	3.08
3.1416132	3.140104	3.1464	2.96
3.1409932	3.143172	3.1388	3.2
3.1419644	3.141808	3.114	3.32
3.14113	3.13922	3.1256	3.0
3.14229	3.1411968	3.1212	3.12

The SPSS (Statistical Package for Social Sciences) program was used in order to conduct further analysis. The first Method of analysis that was conducted was the Box and Whiskers plot. The results of the Monte Carlo Method was put into SPSS, and the values were used to make Figure 3 as shown below.

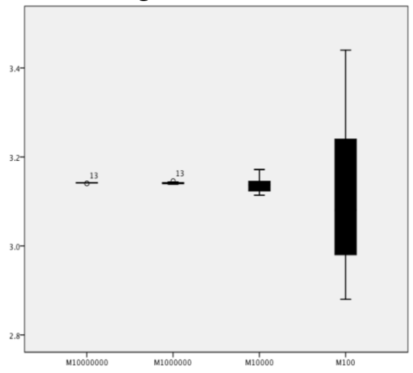

Figure 3. Box and Whiskers Plot

As displayed in Figure 3, the distribution in M100 (sample size set at 100) was wider and therefore, more varied. The plot for M10000 was considerably smaller. This change was not as significant in M10000 to M1000000, and even less visible in the change from M1000000 to M10000000. This reinforces the concept that the accuracy increases with the sample size.

Table 2. Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
M10000000	20	3.140472	3.142636	3.14172236	.000557683
M1000000	20	3.138656	3.145368	3.14138304	.001719461
M10000	20	3.114000	3.171600	3.13490000	.014559280
M100	20	2.880000	3.440000	3.11000000	.159406795
Valid N (listwise)	20				

Table 2 above summarizes the descriptive statistics for different four sample sizes. The range (the difference between Minimum and Maximum value) for M10000000 is smallest, while the range of M100 is the biggest. The difference between the mean and pi decreases with the sample size, and the Standard Deviation increases with the sample size. All this indicates that the Sample Size impact the accuracy of the result.

4.2 Sphere Method

As described in section 3.3, the results of the SPSS analysis is recorded in Table 3. The results of the Monte Carlo Method and the Sphere Method were compared to find out which Method was more accurate. After comparing Table 1 with Table 3, it was noticed that the value of standard deviation that calculated by using the Monte Carlo Method is consistently lower than the value calculated by using the Sphere Method for the same sample size. Therefore, it was concluded that the accuracy of the Sphere Method is worse than the accuracy of the Monte Carlo Method.

Table 3. Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
M10000000	20	3.1397766	3.1436292	3.141558570	.0010176806
M1000000	20	3.1392240	3.1485060	3.143016300	.0023638176
M10000	20	3.0942000	3.1938000	3.145500000	.0301728704
M100	20	2.7600000	3.6600000	3.129000000	.2410044769
Valid N (listwise)	20				

In addition, the Sphere Method's range (the difference between Minimum and Maximum value) is smaller than that of the Monte Carlo Method, the Mean was further away from the actual value of pi, and the Standard Deviation was bigger. This was a setback, especially because higher accuracy was one of the goals. However, the code was shortened. One of the goals was achieved from using this Method.

4.3 Upper Triangle Method

As mentioned in the 3.4, the results were put into SPSS and analyzed. The analysis results are put into Table 4 below. Comparing to descriptive statistics value calculated for Monte Carlo Method (Table 2) and Sphere Method (Table 4), the value of the standard deviation that calculated by using this Method is significantly lower than the value of standard deviation that calculated by using Monte Carlo Method and Sphere Method. Therefore, it can be concluded that the Upper Triangle Method is more accurate compared to the other two Methods.

Table 4. Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
M10000000	20	3.141102226	3.141770130	3.141368284	.0001953283
M1000000	20	3.139085343	3.142782931	3.141015955	.0009628689
M10000	20	3.125231911	3.153061224	3.136382189	.0074296649
M100	20	3.061224490	3.283858998	3.149350649	.0645152218
Valid N (listwise)	20				

Table 4 above also shows that the mean is closer to pi than the results of using the Sphere and Monte Carlo Methods. The range is smaller, and the Standard Deviation is considerably smaller than the other two results. From this, it is concluded that this Method gathers the most accurate results.

4.4 Overall

In Figure 4 below illustrate the comparison of error bar for the same sample size (M= 10,000,000) and all three Methods. It is visible that the Upper Triangle has the smallest Error. The graph shows how the dramatic the accuracy of the outcome was improved by using the Upper Triangle Method. The Sphere Method generated the highest error bar and is the least accurate Method. The original Method(Circle) is the second most accurate.

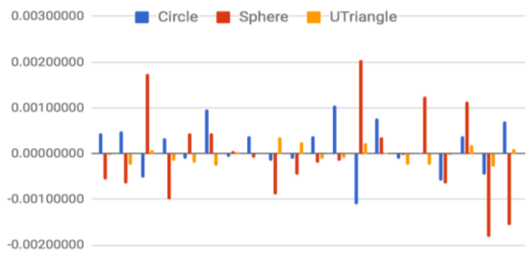

Figure 4. Error Bar (M= 10,000,000)

After all of the results were generated, Figure 5 below was made to find out how many digits of pi is actually needed. This graph was generated by subtracting various accuracies of pi from the results of the 3 Methods. The x-axis displays the number of digits of pi and the y-axis displays the amount of error. In Figure 3, at 6 digits of pi, there is a saturation point. From this, it is concluded that only 6 digits of pi are needed.

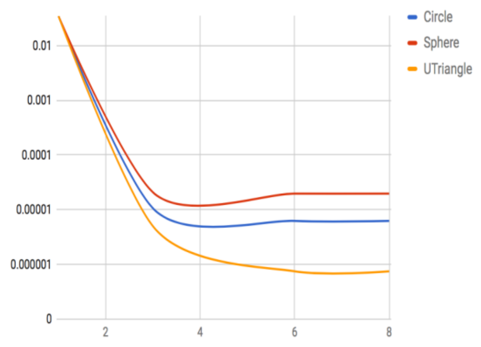

Figure 5. Pi Digits Impact on Accuracy

5. Conclusion

When the project was completed, all goals were achieved. Java was used to successfully simulate pi, it was concluded that the sample size impacted the result, determined that only needed 6 digits of pi, and the code was shortened even though the accuracy improved. For the future, another goal will be added, which is to measure the computation time for each of the codes, then compare them.

The Sphere Method did not work as well as the other two Methods. After some investigating, the root cause of this was found. First, the Monte Carlo Method with the Upper Triangle Method were compared. The concentration of the dots was higher within the same sample size for the Upper Triangle Method. However, that wasn't the only reason. The further away the dots were from the circumference of the circle, the less accurate the value of the data was. The Upper Triangle Method took care of these outliers by limiting the location to the upper half. This concludes that the accuracy is also dependant on how the data is to the circumference.

The results from the same sample sizes in the Monte Carlo Method and the Sphere Method were compared. The Monte Carlo Method did have the outliers, but the error was exponentially increased after using the Sphere Method. It is concluded from these observations that because the dot distribution in the Sphere Method is less concentrated than in the Monte Carlo Method, therefore results for Sphere Method are the least accurate.

In the end, the first few digits of pi were consistently outputted. In the future, I plan to expand the range of numbers and increase the consistency of the accurate results by changing the variables that show correlation to accuracy.

References

Blatner, D., The joy of pi, Penguin, 1997.
Mooney, C., Monte carlo simulation, Sage Publications, 1999.
Landau, E., On Pi Day, How Scientists Use This Number
, NASA JPL, Available: https://www.nasa.gov/jpl/on-pi-day-how-scientists-use-this-number, March 12, 2015.

Biographies

Katherine Lim is a student going to 8th grade at Eleanor Murray Fallon in Dublin. She is the co-author of the statistical STEM paper "Exam Cheating Pattern". In March 2017, she presented the poster "Exam Cheating Pattern" during the JMP Discovery Summit conference in Prague. In her spare time she like to read fantasy, nonfiction, and classic books and play games. Katherine also enjoys traveling around the world with her family. Some places she has explored include Singapore, Canada, China, Malaysia, Japan, Italy etc.