University of Amsterdam

Computing the Area of the Mandelbrot Set

Mandelbrot Set Simulations

November 22, 2021

Students:
Ankur Satya
13832824

Julius Wantenaar
11339055

Group:
8

Lecturer:
Dr. G. (Gábor) Závodszky

Course:
Stochastic Simulations

1 Abstract

The repeating pattern and the weirdly appealing shape of the Mandelbrot set is equally fascinating to mathematicians and non-mathematicians. But mathematicians are more interested in calculating the area of the Mandelbrot set. Since there is no known analytical method for calculating it, scientists use numerical methods. To calculate its area, a sampling of the points in the space is to be done. Here we will discuss 2 of very well known methods for sampling: Monte Carlo method and Latin Hypercube method. Apart from these, we will also discuss a modified version(faster) of orthogonal sampling. Lastly, a method called antithetic will also be discussed. We found that Monte Carlo method is a good enough method when computation is expensive whereas the modified orthogonal method is better when accuracy is more importance.

2 Introduction

Mathematics grasps the attention of humans the most if it can accurately describe patterns found in nature. Nature does not exists of smooth components, but instead consists of many recurring rugged components. These components are best described by mathematical sets called fractals. This term was first coined by Mandelbrot in the 70's and has since been used in many different fields studying nature (Kigami, 2001).

The Mandelbrot set is the set of complex numbers c for which the function

$$f_c(z) = z^2 + c$$

does not diverge to infinity when iterated from z = 0 i.e., for which the sequence $f_c(0)$, $f_c(f_c(0))$, etc. remains bounded in absolute value. A more formal definition is: it is the set of values of c in the complex plane for which the orbit of the critical point z = 0 under iteration of the quadratic map

$$z_{n+1} = {z_n}^2 + c \tag{1}$$

remains bounded. Thus, a complex number c is a member of the Mandelbrot set if, when starting with $z_0 = 0$ and applying the iteration repeatedly, the absolute value of z_n remains bounded for all $n > 0$.

For example, for $c = 1$, the sequence is 0, 1, 2, 5, 26, ..., which tends to infinity, so 1 is not an element of the Mandelbrot set. On the other hand, for $c = 1$, the sequence is $0, 1, 0, 1, 0, ...$, which is bounded, so 1 does belong to the set (Branner, 1989). Although the set appears to be self-similar, in reality every mini-Mandelbrot set has it's own set of external decorations.

These characteristics make it that no one has yet derived the exact area of the Mandelbrot set (Bittner et al., 2017). Only an upper bound of 1.69 has been established. An earlier pixel counting study by Mitchell, 2001 has found the area to be between 1.506480 to 1.506488 with a confidence interval of 95%. In this paper, we try different Monte Carlo simulation techniques to get an estimate of the area of the Mandelbrot set. We start with a naive Monte Carlo, which will then be improved using various variance reduction techniques.

3 Methods

The basic method used to find the Mandelbrot set and its area is discussed below:

- Decide the number of points n_p for which an iteration will be performed using some algorithm(this will be discussed in the subsequent sections).
- Decide the number of iterations n_i.
- For every point in n_p, run n_i iterations. If at any point the magnitude of L.H.S in equation 1 is greater than 2, then that point is not considered as part of the Mandelbrot set.
- If after all the iterations the value does not go above 2, then that point is considered a part of the Mandelbrot set.
- The area is calculated by evaluating the fraction of points in the Mandelbrot set. This fraction is then multiplied by the area of the total space in consideration.

Now, there are different methods to choose the points for checking the divergence which are discussed below.

3.1 Monte Carlo Sampling

Here n_p x-coordinates and n_p y-coordinates are chosen using a random number generator. The ranges of x and y are given and random sampling is performed.

3.2 Latin Hypercube Sampling

For each sample point, a row and a column(randomly) is chosen. Here we need to remember where the previously sampled points are, unlike Monte Carlo sampling, which means all the points chosen will be distinct. This ensures that every dimension of the sampling space is evenly(approximately) sampled.

3.3 Modified Orthogonal Sampling

In the basic orthogonal sampling, the sample space is divided into equally probable subspaces. All sample points are then chosen simultaneously making sure that the total set of sample points is a Latin hypercube sample and that each subspace is sampled with the same density.

We use a modified version of orthogonal sampling to decrease the computational time. We create two grids of $n_p^{0.5}$ x $n_p^{0.5}$, one for x-coordinates and another for y-coordinates. Both the grids are filled with increasing order of integers(from 1 to n_p), the grid for x-coordinates is filled row-wise(meaning, one row is filled with integers before moving on to the next row) and the other grid(for y-coordinates) is filled column-wise(meaning, one column is filled with integers before moving on to the next column). Then for the x-coordinates grid, the values in every row are shuffled and for the y-coordinates grid, the values in every column are shuffled.

3.4 Antithetic Sampling

The main idea of variance reduction using antithetic variables is to decrease variance by introducing a second random variable which has negative co-variance with the first random variable (Blanc et al., 2012). This way of variance assumes the investigated function to be monotonic, which the Mandelbrot set is not. We implemented these variables keeping this in mind. We sampled a set of x values and a set of y values from a uniform distribution, for which we then also took it's negatively correlated counterpart. Therefore, we only needed to sample half the points we ended up with. These values represent the real and imaginary part of the c values of the Mandelbrot set. We only took positive values for the imaginary part of the set to make the set more like a monotonic function. This meant we had to multiply the final result by 2.

The area of the Mandelbrot set calculated using the methods described above depends upon n_i and n_p. Some points diverge after a large number of iterations. So, if we keep the value of n_i too low then such points will be considered a part of the Mandelbrot set and therefore polluting the value of the area. On the other hand, if we take a very high value of n_i the computation expense will increase. So, a balance must be stricken.

The four different sampling methods discussed above try to sample points evenly throughout the space so as to represent the population in an appropriate manner. The orthogonal sampling does a good job at that but still we need to have a sufficient number of points to evaluate the area more precisely. Since, sampling process time is directly related to n_p, we can't take a very high value of n_p and that is why we tried to come up with a better way of sampling discussed in 3.4

S.No.	Sampling method	Time complexity(of sampling)	Time complexity(of simulations only)
1	Monte Carlo	n_p	$n_p * n_i$
2	Latin Hypercube	n_p	$n_p * n_i$
3	Modified Orthogonal	$3 * n_p$	$n_p * n_i$
4	Antithetic	n_p	$n_p * n_i$

Table 1: Time complexity of simulations for different sampling methods.

4 Results

4.1 Plotting the Mandelbrot set and Fractal

The figure 1 shows a pictorial representation of the Mandelbrot set. We can see multiple enclosures in the figure represented by different colors. The lighter the color is, the faster it diverges away and the darker the color is, more number of iterations are needed for the divergence. If we could(somehow) zoom in at the boundaries of various bulbs and the main cardioid, we would see a fractal emerging. This is what makes the Mandelbrot set so fascinating and aesthetically pleasing.

4.2 Monte Carlo Sampling

This sampling method is very easy to implement as it depends only on random number generation and it is also computationally less expensive as being shown by the time complexity of the sampling process in the table 3.4. But it has a major drawback.

It is probable that the result of this sampling method might return more points in one part of the space as compared to the other which pollutes the area evaluation process of the Mandelbrot set. If we look at the left sub-figure in the figure 2, we can see that the estimated area converges after around 200 iterations. The final converged value is 1.55 as compared to the generally accepted value of the area of the Mandelbrot set: 1.50. This helps us conclude that this sampling method has some issues and not the computational power of the system because the estimated value of the area does not change significantly after 200 iterations.

For the right sub-figure, the number of iterations were fixed at 100 and number of points were varied. We see that when the number of points are low, the estimated area fluctuates a lot. As the number of points increase, the estimation becomes more stable which is expected as a greater number of sample points represent the space in a better way. But the final estimated

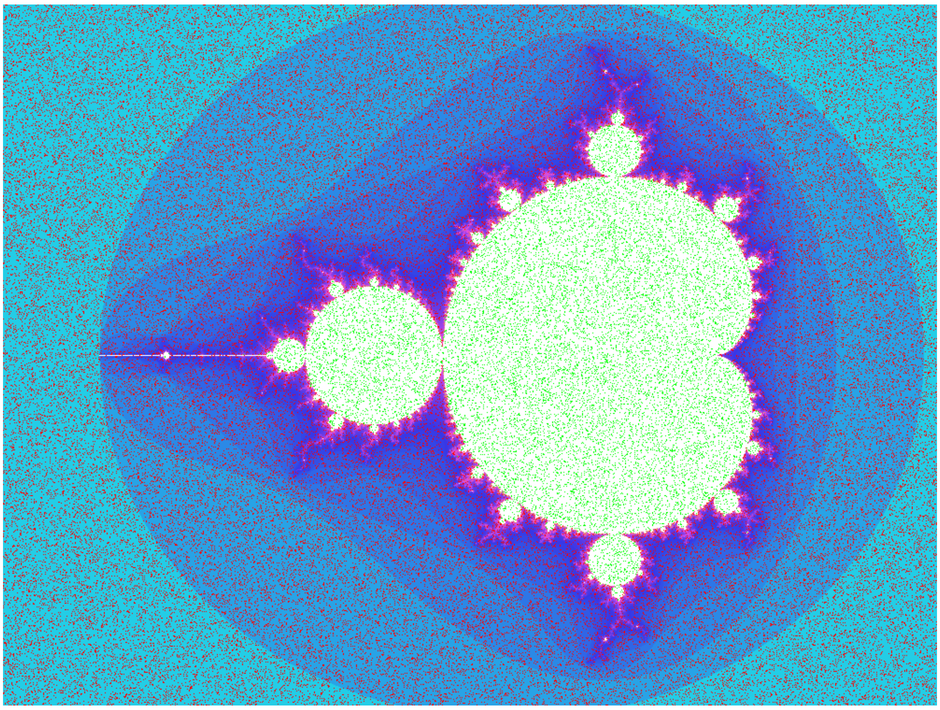

Figure 1: The Mandelbrot set(uncolored) within a continuously colored environment. It shows the main cardioid and multiple bulbs.

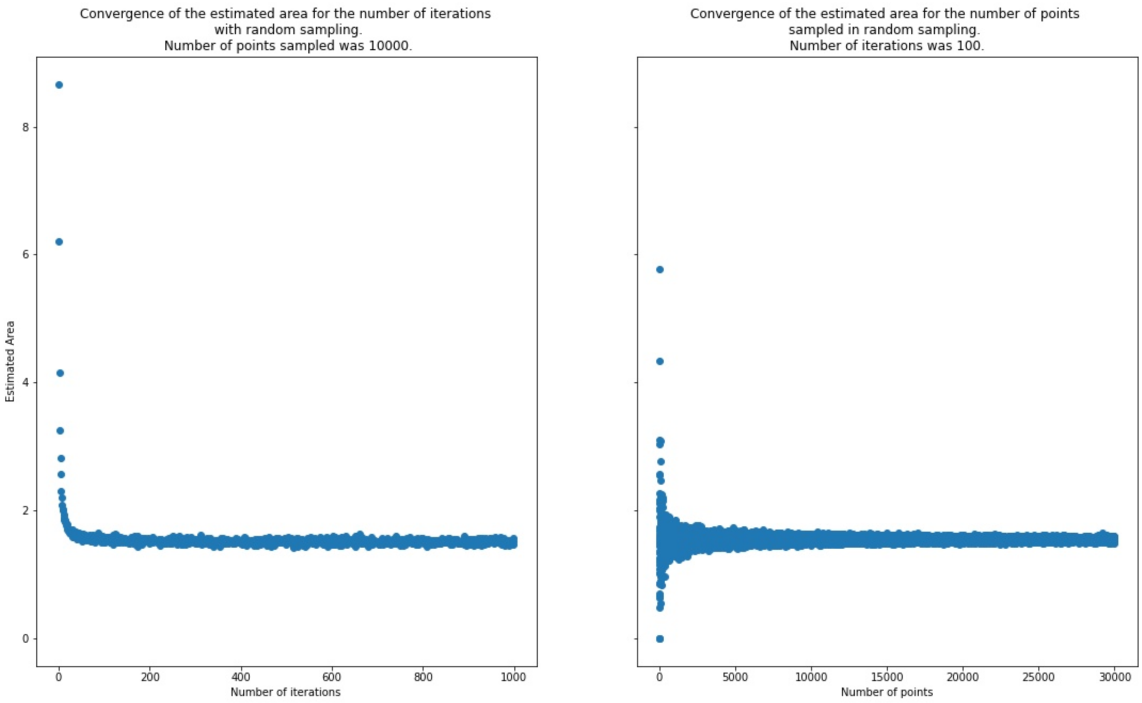

Figure 2: Convergence of estimated area with number of iterations and number of points for random sampling

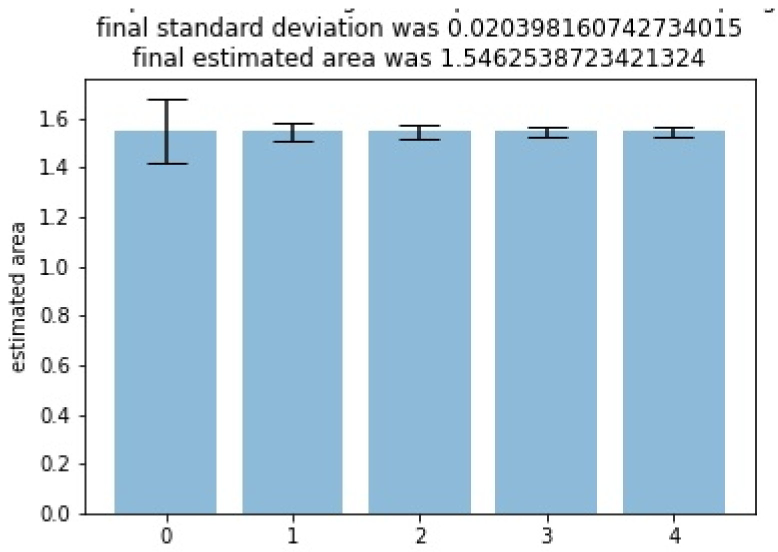

Figure 3: The y-axis shows the estimated area while the x-axis shows the bins created by clubbing the number of iterations.

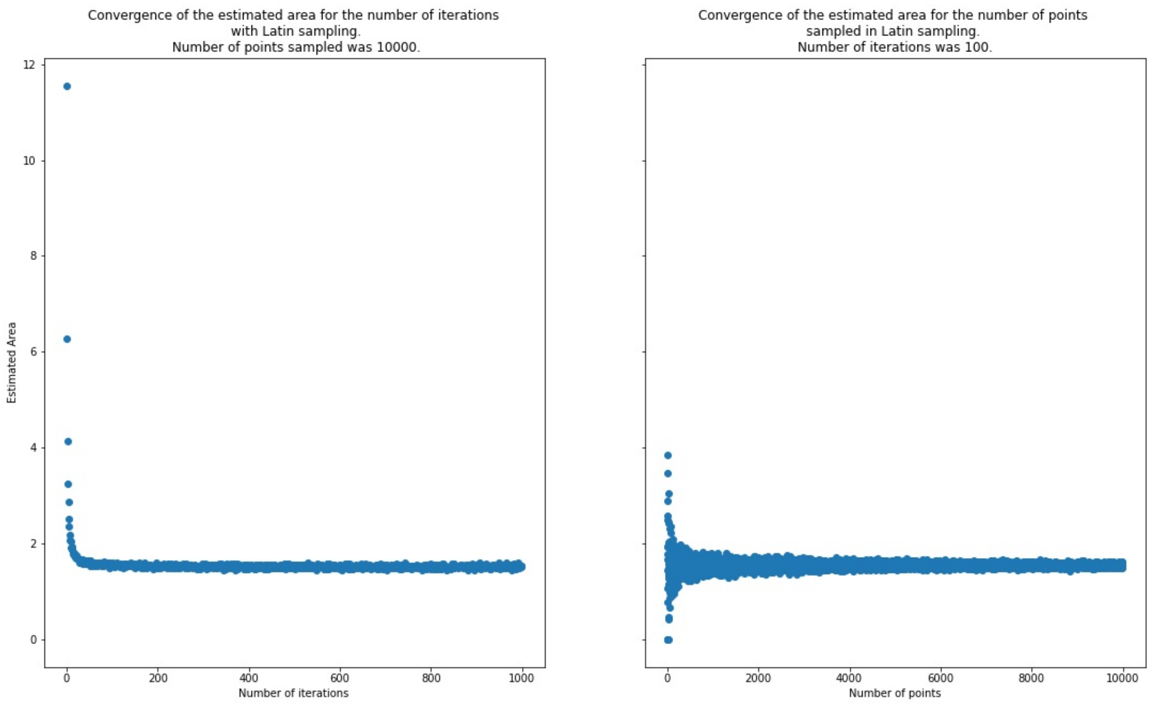

Figure 4: Convergence of estimated area with number of iterations and number of points for Latin Hypercube sampling method.

value of the area(1.56) is worse than what we evaluated in the left sub-figure. It is because 100 iterations are not enough for proper convergence of the area estimation.

The figure 3 clearly shows that the standard deviation of the estimated area decreases significantly with increasing number of iterations.

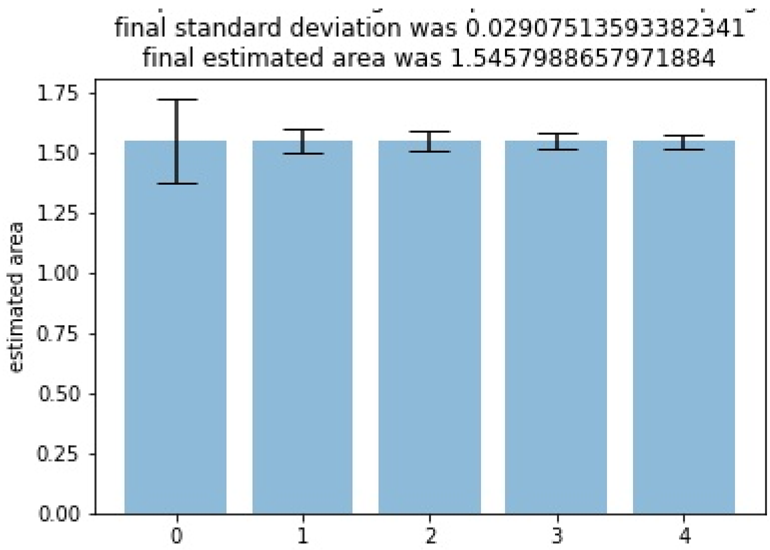

Figure 5: The y-axis shows the estimated area while the x-axis shows the bins created by clubbing the number of iterations.

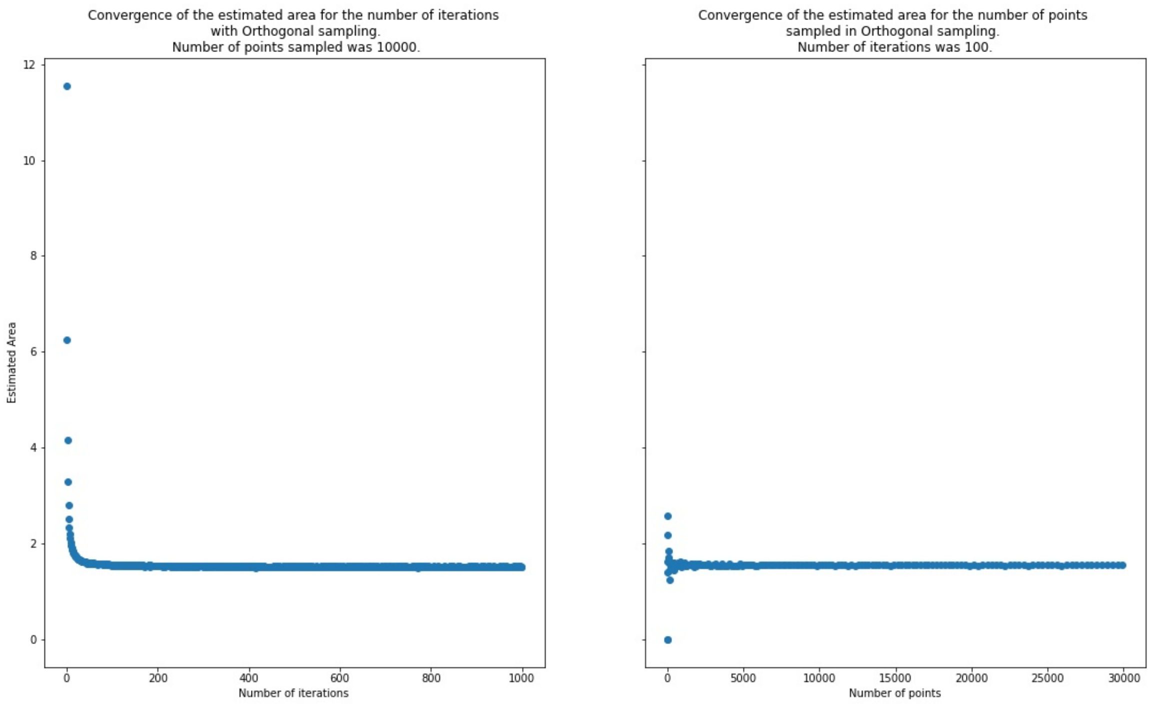

Figure 6: Convergence of estimated area with number of iterations and number of points for Modified Orthogonal sampling method.

4.3 Latin Hypercube Sampling

This sampling method is an improvement over the Monte Carlo sampling method. It tries to create sample points evenly from the complete space unlike the Monte Carlo method. Also the

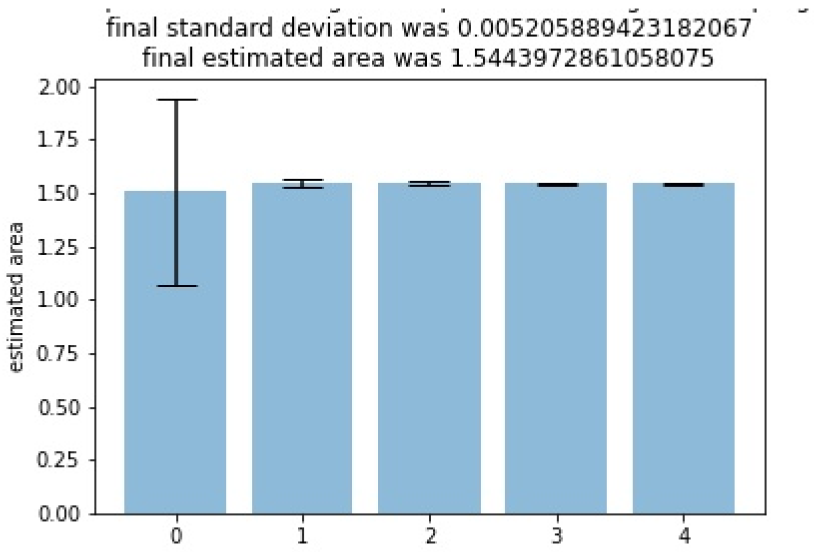

Figure 7: The y-axis shows the estimated area while the x-axis shows the bins created by clubbing the number of iterations.

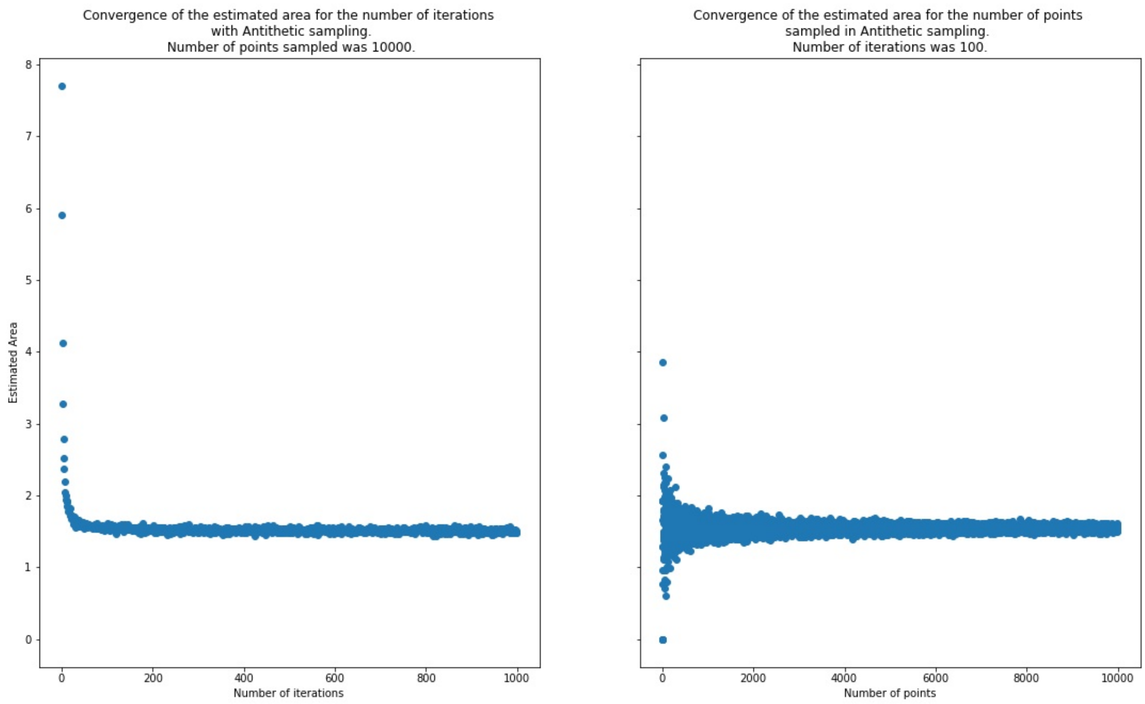

Figure 8: Convergence of estimated area with number of iterations and number of points for Antithetic sampling method.

time complexity of sampling these points is same as the Monte Carlo method as shown in the table 3.4.

Looking at the left sub-figure in the figure 4, we can conclude that the estimated area converges similar to the Monte Carlo method but the big difference is the converged value. It con-

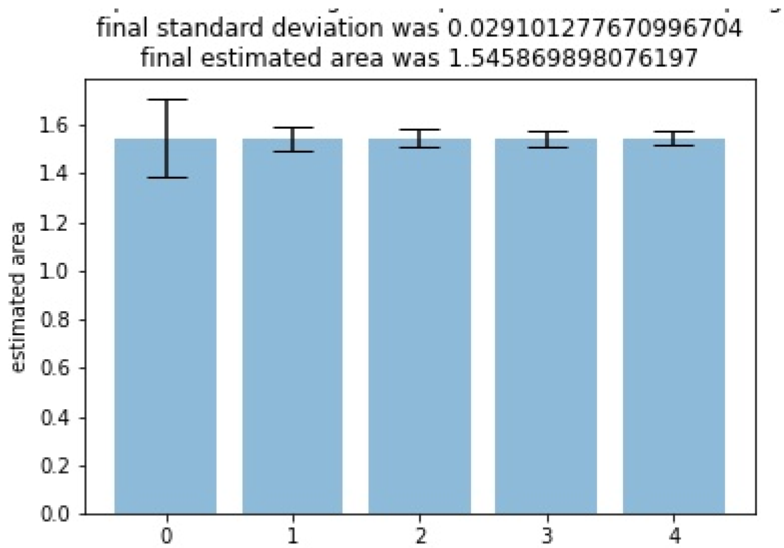

Figure 9: The y-axis shows the estimated area while the x-axis shows the bins created by clubbing the number of iterations.

verges to 1.495 which is quite close to the accepted value: 1.506. The area estimation using this sampling method by varying the number of points shows the same pattern as the Monte Carlo sampling method.But the final estimated value of the area(1.493) is a bit worse than what we evaluated in the left sub-figure. It is because of low number of iterations(100).

The figure 5 shows that the standard deviation of area estimation is quite significant in the beginning(even more than the Monte Carlo method). It is because this method samples the points more evenly which in-turn fluctuates the area estimation in the beginning. But it soon converges to a good approximate value. In conclusion, area estimation using this sampling method is better than the Monte Carlo method both in terms of time complexity and final estimated area.

4.4 Modified Orthogonal Sampling

This sampling method is an improvement over the Latin Hypercube sampling method in terms of evenly sampling of the points by further dividing the grid created in the Latin Hypercube sampling method. But the time complexity of this method is greater than both of the previous cases as shown in the table 3.4 which makes it quite computationally expensive. This pushes us to think of a better method which is discussed in the next sub-section.

Looking at the left sub-figure in the figure 6, we can conclude that the estimated area converges similar to the Latin Hypercube sampling method but the converged value(1.505) is closer to the accepted value: 1.506. The area estimation using this sampling method by varying the number of points shows the same pattern as the Latin Hypercube sampling method. But the final estimated value of the area(1.45) is quite worse than what we evaluated in the left sub-figure. The reason, yet again, is the low number of iterations.

The figure 7 shows that the standard deviation of area estimation is quite significant in the beginning(even more than the Latin Hypercube method). It is because this method samples the points even more evenly than the Latin Hypercube method which in-turn fluctuates the area estimation in the beginning. But it soon converges to a good approximate value with very low standard deviation compared to the previous two methods. In conclusion, area estimation using this sampling method is better than the Latin Hypercube method in terms of the final estimated area but worse in terms of the computation time. So, while choosing a method between these two, one needs to go through the trade-off between accuracy and computational resources.

4.5 Antithetic Sampling

While this technique is a theoretical improvement over random sampling, we do not see faster convergence using this method. Final standard deviation was about the same as for random sampling, namely 0.02 with an equal amount of points sampled. Also, this point of variance was not reached faster. The final estimated area with this technique was a bit higher than the reference area of 1.506480 to 1.506488, which is to be expected at low depths.

5 Discussion

All in all, the area of the Mandelbrot set can be estimated reasonably well with Monte Carlo simulation. We have seen a large variance reduction using orthogonal sampling techniques, we did not see such reductions for antithetic sampling. This difference might be explained by the violation of the assumption of the sampled function being monotonic. We therefore conclude that further decreasing the variance in the estimate of the area of the Mandelbrot set might should focus on further improving the orthogonal technique. This might be achieved by focusing the orthogonal sampling to smaller frames around the Mandelbrot set.

References

Bittner, D., Cheong, L., Gates, D., & Nguyen, H. (2017). New approximations for the area of the mandelbrot set. *Involve, a Journal of Mathematics*, *10*(4), 555–572.

Blanc, X., Costaouec, R., Le Bris, C., & Legoll, F. (2012). Variance reduction in stochastic homogenization using antithetic variables. *Markov Processes and Related Fields*, *18*(1), 31–66.

Branner, B. (1989). *The mandelbrot set* (Vol. 39).

Kigami, J. (2001). *Analysis on fractals*. Cambridge University Press.

Mitchell, K. (2001). A statistical investigation of the area of the mandelbrot set.