UNIVERSITY OF AMSTERDAM

Report

Finding the Surface Area of the Mandelbrot Set

November 18, 2024

Students:
Michael MacFarlane Glasow
1231721
Rinske Oskamp
15817997
Ruben Lanjouw
12415049

Lecturer:
dr. A. Tabi
Course:
Stochastic Simulation
Course code:
5284STSI6Y

Contents

1 Introduction

This investigation seeks to explore the effect of different sampling techniques on the efficiency of determining the area of a given graph, which in our case will be the Mandelbrot set. This unique set of values, which is computed by determining whether the computed values given by the complex quadratic function remain bounded as the iterations continue. Across the field of mathematics, the debate is still ongoing as to what the exact area of the Mandelbrot set is (1), arguably making this precise set and engaging plot to study varying sampling techniques on.

We will be integrating the area of the Mandelbrot set firstly through use of the Monte Carlo sampling technique, a well-studied and tested method which particularly when given a large number of samples, can reliably predict the given area of a plot(2). However, as the computational power that is required to compute these areas can often be prohibitive, it is of great importance to study how one can lessen the computational burden while still producing accurate results.

This paper will therefore investigate varying sampling techniques from pure randomness, which we will take as the control group, to applying Latin hypercube and orthogonal sampling techniques in a comparative manner. Furthermore we will propose a further sampling technique that eliminates given sub-areas of the plot to increase the rate of conversion to the desired result. We hypothesize that optimising the rate at which the relevant areas are sampled will ultimately results in a faster rate of convergence to the area of the Mandelbrot set.

Initially, we will explore computing and plotting the Mandelbrot area, and display this peculiar and intriguing graph to the reader, while elaborating what effect varying the number of iterations has on the shape and area of the set. We will then explore the how the number of samples of the Monte Carlo simulation impacts the final approximation of the investigated area. We will then proceed to introduce the Latin hypercube and orthogonal sampling techniques, and explore how varying the sampling technique while keeping the number of samples constant can impact the computed area. We will then proceed to introduce our sub-area elimination technique (SAET), and vary both the iterations of the Mandelbrot set and the number of samples drawn.

2 Theory

2.1 Mandelbrot

The Mandelbrot set is a very famous fractal. A Mandelbrot marks the points on the complex plane for which the quadratic map $z_{n+1} = z_n^2 + c$ with $z_0 = c$, is bounded. If the map does not go to infinity for $c \in \mathbb{C}$, c is part of the Mandelbrot set (3). Beyond being renowned as a beautiful image that allows for infinite zooming, the Mandelbrot set frequently appears in mathematic applications like chaos theory as well. The complexity of the resulting area makes it a fit subject to be studied under through the varying of sampling techniques.

2.2 Monte Carlo Simulation

The Monte Carlo method is a statistical technique that uses stochastic sampling to find solutions to problems that cannot be calculated manually(4). The name comes from the famous Monte Carlo casino in Monaco, as the method is based on randomness just like gambling(5). The Monte Carlo method was developed in the 1940s for the Manhattan project, which was used to develop the first nuclear bomb. With the rise of computational power, the method became more popular because random sampling was easier. The method of the sampling can be varied, but the pure random sampling approach is arguably the most common and easily implemented approach. Regardless of the method used, the result of the estimation is dependent on the number of samples drawn. However, as the number of samples can theoretically approach infinity, the question of computational efficiency comes into play. The following pseudo-algorithm highlights how the Monte Carlo simulation calculates a given area.

2.3 Pure Random Sampling

The traditional method of sampling employed in Monte Carlo simulations is that of pure random sampling, meaning that across the area of $X \in [-2, 1.5]$ and $Y \in [-1.5, 1.5]$, a random $[X, Y]$ coordinate is selected. Although this method is reliable, and easy to implement, it does require a very large number of samples to arrive at a good estimation of the Mandelbrot area. Therefore the two following sessions will explore two more efficient sampling techniques.

2.4 Latin Hypercube Technique

The Latin hypercube sampling method differs from pure random sampling in that it implements a spatial dimension to how samples are drawn. The given bounded area is divided into equally sized hyperplanes(6). As this investigation is looking to integrate a two dimensional area, the representative Latin hypercube will therefore be divided into $n*n$ number of equally sized rectangles across the bounded area of the Mandelbrot set, where n represents the number of samples drawn. The technique ensures that one gains an effective spread of samples with greater accuracy. My making each subspace equally likely to be drawn, Latin hypercube sampling ensures that clustering is avoided and that each interval is uniformly covered.

2.5 Orthogonal Sampling

Orthogonal sampling involves dividing the given area into equally probably subspaces, with each subspace having the same number of samples drawn into. The orthogonal sampling still follows the logic of the Latin hypercube, meaning that there are no overlapping hyperspaces which have more than one sample(7). This makes the sampling stratified and free of bias due to correlations between the dimensions.

3 Method

3.1 The Mandelbrot set

As stated in Section 2.1 the Mandelbrot set is the set of complex numbers $c \in \mathbb{C}$ for which $z_{n+1} = z_n^2 + c$, with $z_0 = c$, is bounded. To create a visual representation of the Mandelbrot set, we test if a map of a complex number c is still in scope after a given number of iterations. For the most accurate representation of the Mandelbrot set, this number should theoretically approach infinity. However, it has been found that setting this number to 100 already provides an accurate representation. If after 100 iterations the quadratic map is still within the bounds of the Mandelbrot set, we assume that it is part of the set. Figure 1 shows a visual representation of the Mandelbrot set. In this image, the colors are based on the number of iterations preformed before the map falls out of bound.

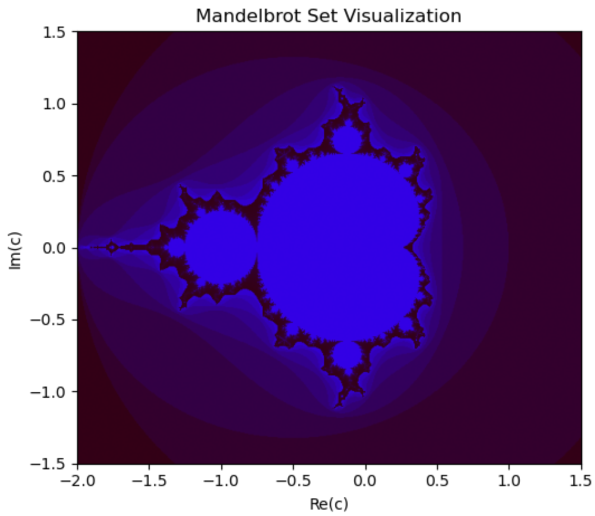

Figure 1: A visualization of the Mandelbrot set, with coloring based on the number of iterations needed

3.2 Estimating the area of the Mandelbrot Set

This investigating will be using the Monte Carlo integration to estimate the total area of the Mandelbrot set. The area of the Mandelbrot set, denoted by A_M, will be approximated by an estimate $A_{i,s}$, where i refers to the maximum number of iterations of the quadratic map and s refers to the number of samples drawn. The algorithm is simple; we take a uniform random sample of coordinates in the complex plane and test each point for the conditions of the Mandelbrot. We then calculate which part of the samples was part of the set, and so estimate the total surface area of the Mandelbrot. Figures 2 and 3 show this estimation visually.

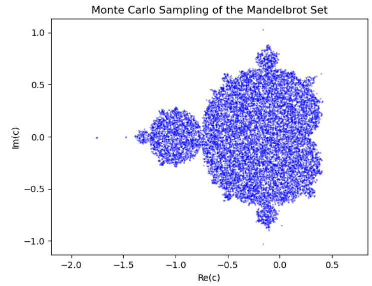

Figure 2: Monte Carlo sampling of the Mandelbrot set with a maximum of 100 iterations and 100,000 samples.

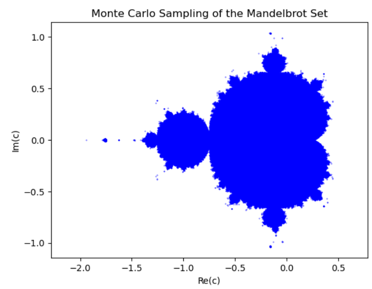

Figure 3: Monte Carlo sampling of the Mandelbrot set with a maximum of 100 iterations and 2,000,000 samples.

We examine the change in absolute error of the Monte Carlo method for varying number of samples. To do this, we set the true area to our best estimate. This estimate is composed by taking the average value of five runs of the Monte Carlo method, with 2 million samples and a maximum of 1000 iterations. This results in an area of $A_{true} = 1.5083628$. We observed that the estimate does not change significantly after further increasing these values. Next, we investigate the effect of varying the maximum iterations while keeping the sample size fixed.

3.3 Implementing Latin hypercube and Orthogonal sampling

As part of the mission of this paper is to investigate more efficient methods of sampling, the Latin hypercube and orthogonal methods will be implemented and compared to the pure random sampling technique. For the pure random and Latin hypercube sampling 100,000 samples will be drawn, whereas the orthogonal sampling, due to computational limitations, will be drawn 10,000 times. We will then compare both the absolute error from the true Mandelbrot area and also be comparing the standard deviation between the three methods. To calculate this, we will vary the number of samples drawn while keeping the number of iterations constant.

We expect to see the greatest rate of convergence to the true area of the Mandelbrot area to be that of the orthogonal sampling method, as it improves on the already enhanced methodology of the Latin hypercube to further improve the sampling technique. We expect the standard deviation to indicate the precision of the estimation, and therefore expect the orthogonal sampling to have the lowest standard deviation, and the traditional pure sampling the highest.

3.4 Improving the Monte Carlo algorithm

There are several ways and methods to improve the Monte Carlo method to estimate the surface of the Mandelbrot set. This can be achieved, as previously described, by improving the sampling method or by applying filters to reduce the total area over which the Monte Carlo sampling is applied, making it more efficient. Given that all points of the Mandelbrot lie inside the subset of $x \in [-2, 1.5]$ and $y \in [-1.5, -1.5]$, we will divide this entire area into 25 sub-regions, evenly distributed over the x and y-axis, for our improved algorithm. Figure 4 shows this separation.

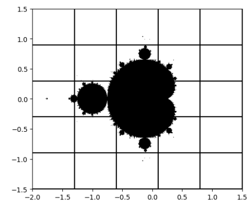

Figure 4: Visualization of planes in Mandelbrot set for improving Monte Carlo.

All these sub-regions are then tested for the first fifty percent of the total amount of iterations. The samples are evenly distributed over the regions meaning that all regions get $\frac{N}{50}$ of the iterations. Then the results are evaluated by looking at the percentage of hits. If all samples in the individual sub-regions are returned positively, we it assumed that the region is completely filled and and will therefore exclude it from further sampling. If the sub-region has no positive samples drawn, the assumption is made that is empty, and will therefore also be excluded from further testing. The next 50% of the iterations is then distributed over the remaining sub-regions, resulting in higher density in these areas. This should improve the quality of estimation in these regions, because we then assume that these sub-regions are those that lie on the contour of the Mandelbrot area, allowing for their complexity to be captured more accurately. The way of sampling which is used in this algorithm can also differ but for simplicity reasons the random sampling method is used. The implementation of this algorithm can be described by this pseudo-algorithm. We expect to find that this technique will allow us to greatly increase the precision with which the estimation of the area is computed, as we can safely exclude a large amount of the bounded area.

Algorithm 2 Improved Monte Carlo Algorithm on 2d Mandelbrot

```
Define the borders of the planes c
Define the total amount of iterations N
for plane in planes do
    for i in range(0.5N/25) do
        Calculate the hit ratio
    end for
    Define amount of planes which are 0 or 1
    Calculate the new amount of iterations for left-over planes
    for i in range(0.5N * Amount of non 0 or 1 planes) do
        Calculate percentage of area
    end for
    Define the total area ratio for plane
end for
Add all area ratio's and multiply with size of planes =0
```

4 Results & Discussion

4.1 Monte Carlo Random Sampling

Figure 5 shows the absolute error in relation to the true area. This plot shows that increasing the maximum number of iterations to 1000 has little effect on the precision of the area computation.

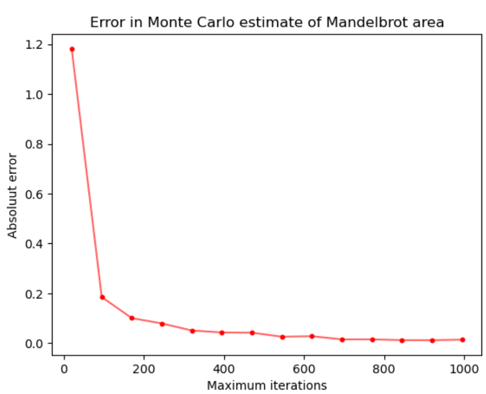

Figure 5: The absolute Error of the Monte Carlo method with 2 million samples and varying maximum of iterations

4.2 Different Sampling Methods

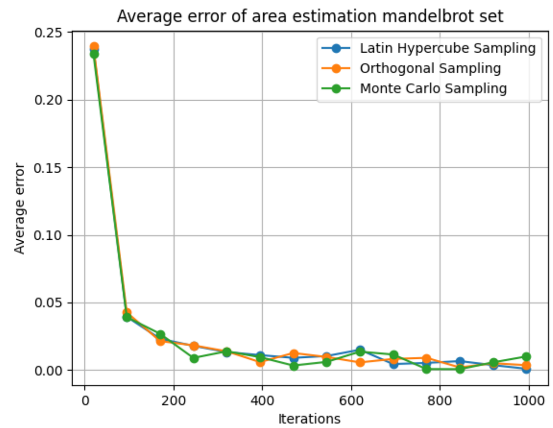

Figure 6: Different Monte Carlo Sampling methods. S= 100,000 (orthogonal =10,000), n=5

We find in Figure 6 that the three sampling techniques follow comparable trajectories over the increased iterations. We can conclude that while keeping the number of samples equal, increasing the number of iterations results in a similar convergence to the true area of the Mandelbrot set. As we want to gain more granular insight into the difference in results between the sampling techniques, let us now turn to the results of varying the number of samples drawn.

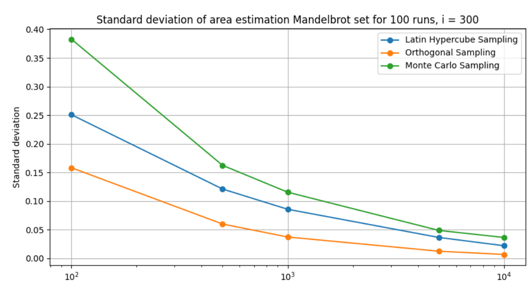

Figure 7: Standard deviation of area estimation of Mandelbrot set for 100 runs, with i=300

We observe in Figure 7 that the difference between the sampling techniques becomes more apparent when studying the standard deviation from the area estimation when increasing the sampling size. As predicted, we find that the orthogonal sampling technique does in fact, have the lowest standard deviation out of all three techniques. This indicates the orthogonal sampling is the most precise and effective method for reaching a good estimate of a given integration problem. It can be argued that Latin hypercube, but in particular Orthogonal sampling, greatly improve the rate of convergence to the true area. The real-life implication of this is that using this method can significantly lower the computing power needed to reach a desired outcome, as the sampling technique using the principle of orthogonality allows for more precise measurement of the given area. Going forward, it could therefore be argued that one should encourage the use of this method as opposed to pure random sampling, especially when dealing with large and complex areas.

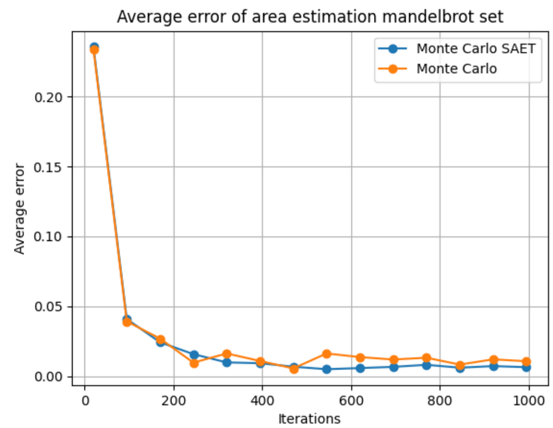

Figure 8: Absolute error of area estimation between pure random sampling and SAET

When comparing the average error found in Figure 8 between the two techniques, we find a similar result to that in the in 6, namely that both sampling techniques display a similar trajectory as the iterations increase to 1000. This once more confirms our finding that both sampling techniques are effective at arriving at the true area of the Mandelbrot set. Let us however turn once more to the difference in precision between the two techniques.

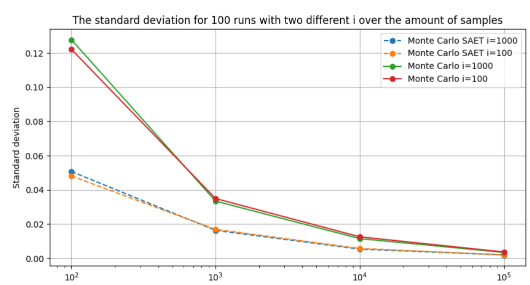

Figure 9: .....

4.3 Improved Monte Carlo SAET

Now let us turn to the results that we found when applying our newly devised sub-area elimination technique (SAET). We compared the traditional sampling technique of pure random sampling within Monte Carlo simulations to our improved sampling techniques.

In Figure 9 we observe even when varying the number of iterations used in the Mandelbrot algorithm, we can clearly distinguish the precision of the SAET method to that of the pure random sampling. The standard deviation produced by the SAET method is below half of that of the pure random sampling technique, even after 1000 samples. This allows us to conclude that the SAET method allows for a considerably more efficient way of sampling the area of a graph.

Sample Size	error normal	error SAET	σ normal	σ SAET
100	0.00447	0.00844	0.13	0.05
1000	0.00669	0.00490	0.03	0.02
10000	0.00325	0.00451	0.01	0.01
100000	0.00426	0.00339	0.00	0.00

Table 1: Differences in absolute error and standard deviation with i=1000 for different sample sizes.

From 1 we can additionally glean that while the errors remain in a comparable range as the number of samples increases, we can observe the stark difference in the standard deviation. A possible weakness that we can observe here is that for small samples the error can be quite large, however as the sample size increases the error returns to lower than the traditional sampling method.
As we concluded with the LHS and orthogonal techniques, we find that the SAET method considerably outperforms the results of the pure random sampling techniques. We therefore conclude the using the SAET technique should be preferred over that of pure randomness, as it would conserve considerable computational power when applied to larger problem sets. For smaller samples, we advise caution with the SAET, as the error can be quite large for smaller sample sizes.

5 Conclusion

This paper set out to investigate how varying sampling techniques could improve the precision and efficiency of calculating the area of the Mandelbrot set. With using pure random sampling as the baseline, we compared how changing the sampling technique to incorporate the principles of the Latin hypercube method and orthogonality, we found that we were able to considerable reduce the standard deviation of the area estimation quite significantly, which allowed us to conclude that the two latter techniques are more precise in estimating the set. To further improve upon the Monte Carlo technique, we introduced our sub-area elimination technique, which allowed us to omit given sub-areas of the bounded area based on whether samples had been returned from them. We found that compared to the pure random sampling technique, we were also able to considerably increase the precision of the result. However, we did also find a possible shortcoming of the technique, in that it performs poorly for small sample sets. In both cases however we were able to encourage researchers to opt for a selection of the improved sampling techniques, as these are able to more efficiently calculate the area of the Mandelbrot set. We expect these techniques to considerably lower the computational burden of achieving precise results going forward. For further research, we are curious to expand upon on our SAET method by one the one hand increasing the number of sub-regions. Furthermore, we are curious to study the effect of lowering the 50 percent threshold from when the sub-area elimination occurs. In addition, we are also curious to change the elimination method from our current Boolean approach, where we eliminate areas only at the extremes, to a continuous weighting. This would involve creating a weighting of areas after the threshold, and weighting the number of samples based on the prior results.

References

[1] Ioannis Andreadis and Theodoros E. Karakasidis. On numerical approximations of the area of the generalized mandelbrot sets. *Applied Mathematics and Computation*, 219(23):10974–10982, 2013. URL: `https://www.sciencedirect.com/science/article/pii/S0096300313004876`, `doi:10.1016/j.amc.2013.04.052`.

[2] R. L. Harrison. Introduction to monte carlo simulation. *AIP Conf. Proc.*, 1204:17–21, 2010. Jan 5. `doi:10.1063/1.3295638`.

[3] Wolfram MathWorld. Mandelbrot set, 2024. Accessed: 14 November 2024. URL: `https://mathworld.wolfram.com/MandelbrotSet.html`.

[4] Daniel D McCracken. The monte carlo method. *Scientific American*, 192(5):90–97, 1955.

[5] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[6] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. `doi:10.2307/1268522`.

[7] B. Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993. `doi:10.1080/01621459.1993.10476423`.