UNIVERSITY OF AMSTERDAM

Report

Monte Carlo Convergence on the Mandelbrot Set

Monday 9th December, 2024 14:28

Student:
Lucas Keijzer
lucaskeijzer@gmail.com
14041073
Maarten Stork
MaartenAStork@gmail.com
15761770
Paul Jungnickel
Paul.Jungnickel@student.uva.nl
15716554

Lecturer:
Giulia Pederzani
Andrea Tabi

Course:
Stochastic Simulation

Course code:
5284STSI6Y

Abstract

The Mandelbrot set is a fractal subset of the complex plane. This report compares Monte Carlo methods for estimating its area, using pure random, orthogonal, Latin Hypercube, and adaptive sampling. With a scalable PyTorch implementation, our best estimate is $A_M = 1.5069 \pm 0.0055$. We tested the hypothesis that adaptive sampling improves convergence compared to orthogonal sampling and found no significant evidence to reject the null hypothesis.

1 Introduction

This report delves into numerical methods for approximating the area of the Mandelbrot set through Monte Carlo integration. We explore and test the efficiency of various sampling techniques, focusing particularly on improving the convergence rate of area estimates.

1.1 Background Theory

1.1.1 Estimating the Mandelbrot Set

The Mandelbrot Set M is a subset of the complex plane $\mathbb{C}$ that occurs in complex dynamics. It is defined using the maps

$$f_c : z \mapsto z^2 + c$$

containing all points $c \in \mathbb{C}$ where f_c does not diverge to infinity when starting from $z_0 = 0$.

Due to its fractal shape, no analytical solution for the area of the Mandelbrot set is known. A rough lower bound of $7\pi/16 \approx 1.3744$ and an upper bound of 1.7274 is provided by Ewing and Schober (Ewing & Schober, 1992). The upper bound is further refined to 1.68288 in (Bittner, Cheong, Gates, & Nguyen, 2014). One standard result about the Mandelbrot set is that it does not contain values whose magnitude is larger than 2, because f_c increases exponentially for these values. From this also follows that f_c always diverges if $|z| > 2$. Thus, we can test if a given point c lies on M by iterating f_c and rejecting it as soon as the magnitude of z_j grows above 2. Methods based on sampling points from the complex plane and testing for divergence yield area estimations of $A_M \approx 1.52$ (Ewing & Schober, 1992), 1.5052 (Andreadis & Karakasidis, 2015), and 1.5065918849 (Förstemann, 2012). This discrepancy between the theoretical bounds and the empirical estimates could be explained either by very slow convergence of the numerical methods used for the theoretical upper bounds, or by the fractal boundary having a nonzero Lebesgue measure (Bittner et al., 2014).

1.1.2 Monte Carlo Methods

The Monte Carlo method is a stochastic approach to numerically approximate solutions to mathematical problems by using random sampling (Rubinstein & Kroese, 2016). This technique leverages randomness to estimate the properties of complex systems or to compute integrals.

The core principle behind the method is the law of large numbers, asserting that as the number of trials increases, the average of the results obtained from random samples will converge to the expected value (Hammersley & Handscomb, 1964). Mathematically, if $f(x)$ is a function we want to integrate, and x_i are uniformly distributed random samples, then the Monte Carlo estimate of $\int f(x)\,dx$ can be approximated as:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(x_i),$$

where N is the number of samples.

This method's accuracy improves with the number of samples, and its effectiveness is inherently tied to the distribution and quality of the random numbers generated (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). The Monte Carlo method is often used when exact solutions are challenging or impossible to derive analytically.

1.1.3 Different Sampling Methods

The accuracy and convergence rate of Monte Carlo integration are heavily influenced by the sampling method employed (Rubinstein & Kroese, 2016). Different techniques aim to enhance the uniformity of sample distribution and reduce variance in the estimation process. We will explore four sampling methods: pure random sampling, orthogonal sampling, Latin Hypercube Sampling, and adaptive sampling.

Pure Random Sampling

Pure random sampling involves selecting sample points independently and uniformly at random over the domain of interest (Hammersley & Handscomb, 1964). N random complex numbers c_i are generated by sampling the real and imaginary parts uniformly within the specified ranges (typically $[-2, 2]$ for both the real and the imaginary part). The area of the Mandelbrot set A_M can then be estimated by:

$$\hat{A}_M = \frac{\sum_{i=1}^{N} f(c_i)}{N} \times \text{Area of } D,$$

where $f(c_i)$ is the indicator function that equals 1 if $c_i \in M$ and 0 otherwise, and D is the area of the sampling domain. While simple and unbiased, pure random sampling may converge slowly due to clustering and gaps in the sample distribution (Metropolis et al., 1953).

Orthogonal Sampling

Orthogonal sampling improves upon pure random sampling by ensuring that samples are more evenly distributed across the domain (Press, Teukolsky, Vetterling, & Flannery, 2007). Domain D can be divided into a grid of equally sized, non-overlapping cells, forming an $n \times n$ grid where $n = \sqrt{N}$ and N is adjusted to be a perfect square. Within each cell, a single sample point can be selected by adding a random offset to the cell's starting coordinates:

$$x = x_{\min} + (i + r_{\text{real}})\,\frac{\Delta x}{n}, \quad y = y_{\min} + (j + r_{\text{imag}})\,\frac{\Delta y}{n},$$

where $i, j = 0, 1, \ldots, n-1$, r_{real} and r_{imag} are independent random numbers in $[0, 1)$, and $\Delta x = x_{\max} - x_{\min}$, $\Delta y = y_{\max} - y_{\min}$. This ensures that each cell contains exactly one sample, reducing variance by preventing clustering of samples and promoting uniform coverage of the domain (Press et al., 2007).

Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) is a stratified sampling technique that divides the domain into N intervals along each dimension and ensures that each interval is sampled exactly once along each axis (McKay, Beckman, & Conover, 1979). In this method, we generate N sample points by systematically selecting values such that for each sample $c_i = (x_i, y_i)$, the components x_i and y_i are chosen from shuffled partitions of their respective axes. Specifically, the samples are constructed as:

$$x_i = x_{\min} + \left(\frac{P_i + r_i}{N} \right) \Delta x, \quad y_i = y_{\min} + \left(\frac{Q_i + s_i}{N} \right) \Delta y,$$

where P_i and Q_i are random permutations of the integers $0, 1, \ldots, N-1$, r_i and s_i are independent random numbers in $[0, 1)$, and $\Delta x = x_{\max} - x_{\min}$, $\Delta y = y_{\max} - y_{\min}$. This approach maintains uniform marginal distributions and provides better coverage of the domain than pure random sampling, potentially reducing the variance of the estimator (McKay et al., 1979).

Adaptive Sampling

Adaptive sampling adjusts the sampling strategy based on information gathered during the estimation process (Robert & Casella, 2013). The domain is partitioned into multiple regions, and the sampling effort is iteratively refined by allocating more samples to regions with higher variance in the indicator function. At each iteration k, the area estimate is updated as:

$$\hat{A}_M^{(k)} = \sum_{r=1}^{R} \hat{A}_r^{(k)},$$

where $\hat{A}_r^{(k)}$ is the area estimate for region r after the k-th iteration. By focusing computational resources on the most critical regions—those contributing most to the estimator's variance—adaptive sampling enhances the efficiency and accuracy of the area estimation compared to methods that distribute samples uniformly across the entire domain (Robert & Casella, 2013).

Each of these sampling methods offers a trade-off between implementation complexity and statistical efficiency. By selecting an appropriate sampling technique, we can achieve more accurate results with fewer samples compared to pure random sampling.

1.2 Related Work

The following papers provide theoretical frameworks and insights necessary for analyzing Monte Carlo convergence on the Mandelbrot set. Other valuable but more specific sources will be cited in the references but are not discussed in detail here.

1. *Estimating the Area of the Mandelbrot Set* by Kostelich and Schreiber (1991) examines stochastic sampling techniques for estimating the Mandelbrot set's area using Monte Carlo methods, focusing on convergence rates and statistical accuracy. Their work addresses the challenges of applying stochastic processes to fractal structures, informing our experimental simulations (Kostelich & Schreiber, 1991).

2. *Monte Carlo Integration and Fractal Dimension Computation* by Tishby and Biham (1992) explores computing the fractal dimension of the Mandelbrot set using Monte Carlo integration, analyzing the convergence properties of stochastic methods. This study informs our exploration of convergence behaviors in fractal geometries (Tishby & Biham, 1992).

3. *Monte Carlo Methods in Fractal Geometry* by Barnsley (1988) discusses applying Monte Carlo methods in fractal geometry, including the Mandelbrot set, and examines convergence characteristics of stochastic methods. Barnsley's approach guides our implementation and assessment of convergence in fractal contexts (Barnsley, 1988).

These texts collectively provide a framework for applying Monte Carlo methods to the Mandelbrot set and analyzing their convergence properties, shaping our experimental design and analytical framework.

1.3 Research Question

This study is guided by the research question:

Does adaptive sampling improve the convergence rate of Monte Carlo methods in estimating the area of the Mandelbrot set?

To address this, we compare traditional random sampling with adaptive sampling to evaluate whether adaptive methods achieve faster convergence and greater accuracy using fewer samples. Additionally, other sampling methods are evaluated to provide a comprehensive analysis. The following hypotheses guide our study:

$\mathbf{H_1}$**:** *Adaptive sampling significantly enhances the convergence rate, resulting in lower estimation errors compared to orthogonal sampling.*
$\mathbf{H_0}$**:** *Adaptive sampling does not improve the convergence rate, showing no significant error reduction compared to orthogonal sampling.*

1.4 Report Structure

This report is structured to provide an analysis of Monte Carlo convergence on the Mandelbrot set and to test whether certain sampling methods can further improve the convergence rate of the Monte Carlo approach. The *Introduction* establishes the context and outlines the problem, including a brief overview of relevant literature. The *Methods* section presents methods to implement our experiments and the parameters used in the investigation. The *Results* section objectively presents the findings, focusing on the performance of different sampling methods and their impact on convergence rates. The *Discussion* interprets the results, assesses the effectiveness of the sampling methods, and suggests directions for future research.

2 Methods

2.1 Scalable Implementation of the Mandelbrot iteration

Given a point $c \in \mathbb{C}$, we test if it is inside the Mandelbrot set by iterating the map f_c a large number of times. Our implementation computes the Mandelbrot iteration on all samples in parallel, taking advantage of Single Instruction / Multiple Data (SIMD) operations. For this, the PyTorch library (Paszke et al., 2019), is used to generate scalable code with python-like structure. This gives us a significant performance advantage over regular python code, allowing us to test much larger sample sizes and iteration counts.

2.2 Mandelbrot Visualization

To better understand the Mandelbrot set we first visualize it. The visualization is essential for understanding the fractal structure, and understanding the problems regarding the area estimation. To make this more clear we will also look into the boundaries of the fractal to show its complexity. We do this by by generating a 2D grid of complex numbers, iteratively computing divergence using $Z = Z^p + C$, and tracking the number of iterations before divergence. The results are displayed as a color-mapped image, where the color intensity represents divergence steps.

2.3 Mandelbrot Set area estimation

To estimate the area of the Mandelbrot set, we employed the Monte Carlo method with the four different sampling techniques: pure random sampling, orthogonal sampling, Latin Hypercube Sampling, and adaptive sampling. The complex plane was sampled within specific real and imaginary ranges, and the points were iterated to determine if they diverged.

The resulting area estimates were compared across the four methods, highlighting differences in accuracy and convergence rate. By analyzing the mean absolute errors and the convergence patterns of each approach, we aimed to determine the most effective sampling method for estimating the area of the Mandelbrot set.

2.4 Quality of our approximation

Since we are approximating the area numerically, our estimation is limited by three factors: The finite number of sampled points in $\mathbb{C}$, the finite number of iterations and the numeric error caused by finite-digit representation of all values in memory. We refer to these errors as the sampling error ε_s, iteration error ε_i and numerical error ε_{num}, respectively. Both the sampling error and the iteration error can be minimized by increasing the sample/iteration count due to laws of large numbers while the numerical error might be decreased by choosing a larger floating point representation.

To bound the sampling error, we can use a concentration inequality, which expresses how much a random variable is expected to deviate from its mean. For a set of bounded random variables $X_i \in [a, b]$, Hoeffding's inequality specifies the likelihood that the sum $S_n = \frac{1}{n}\sum_{i=0}^{n} X_i$ is more than ε away from its expected value:

$$P(|S_n - E[S_n]| \geq \varepsilon) \leq 2exp\left(\frac{-2\varepsilon^2 n}{(b-a)^2}\right)$$

Specifying a confidence level $1-\delta$, we can rearrange this inequality to get an expression for the error ε depending only on the number of samples:

$$\begin{aligned} 2exp\left(\frac{-2\varepsilon^2 n}{(b-a)^2}\right) &\leq \delta \\ \frac{2\varepsilon^2 n}{(b-a)^2} &\leq -log(\delta/2) \\ \varepsilon &\leq (b-a)\sqrt{-\frac{1}{2n}log(\delta/2)} \end{aligned} \tag{1}$$

We treat the iteration error by applying as many iterations as feasible and measuring the convergence rate of the estimates at all previous iterations towards the value at the final iteration. The area estimate of a fixed sample can never increase since each iteration can only eliminate more points the Mandelbrot set. If we observe a good convergence behavior, we can find an iteration count that leads to an ε_i that is comparable to the other expected errors.

2.5 Statistical Tests

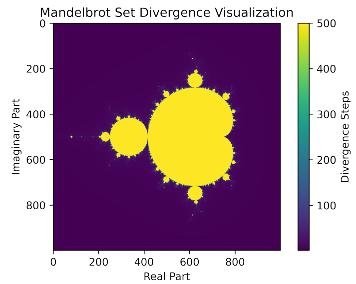
To evaluate whether adaptive sampling significantly improves convergence compared to orthogonal sampling, we employed an independent t-test under the assumptions of (1) independence, (2) normality, and (3) equal variances. The Shapiro-Wilk test indicated no significant deviation from normality (p-values: 0.12 and 0.15 for orthogonal and adaptive sampling, respectively). Furthermore, visual inspection using histograms and Q-Q plots further supported normality for both datasets. Levene's test confirmed equal variances (p-value: 0.271). Given these findings, we proceeded with the independent t-test.

3 Results

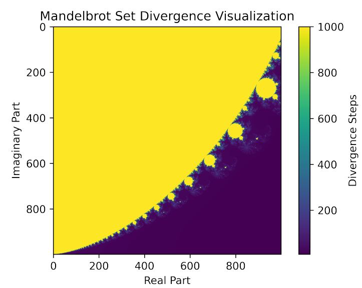
In our computational experiments, we investigate the convergence rates of the Monte Carlo estimates with different sampling methods. We first examine the iteration, sample and numerical errors to verify the quality of our approximation. Then we test if the convergence rate of adaptive sampling is better than that of the best performing static sampling method. We also use our implementation of the Mandelbrot iteration to visualize different sub-areas of the Mandelbrot set.

3.1 Visualization

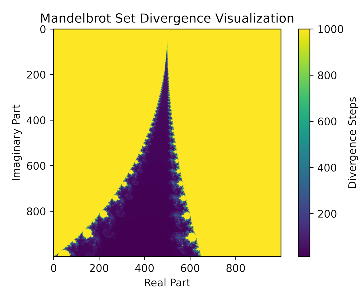
In figure 1 the visualizations of the Mandelbrot set are shown. In figure 1a you can see the Mandelbrot set as a whole. The other sub figures are zoomed in at the boundaries of the set. These boundaries show the intricate fractal structure, which bring the challenges regarding the area estimation.



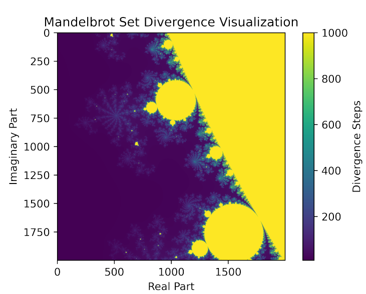
(a) Mandelbrot set



(b) Elephants Valley



(c) Seahorse Valley



(d) Spiral Region

Figure 1: Visualizations of the Mandelbrot set

3.2 Convergence of the Area Estimate

Figure 2 shows the convergence of the area estimate with increasing number of iterations. Since we do not have the true value of A_M, we choose a very large number of iterations i $=500$ 000 and measure the difference between the area estimate between iteration j and i for all $j < i$. The samples are uniformly distributed on the complex rectangle $[-2, 2]^2$ with different sample counts s.
The vertical line at the end of each curve indicates the iteration j at which the area estimate stops changing. We observe that, until they stop changing, the trajectories of all curves are very similar, with the estimate difference decreasing by an order of magnitude for each order of magnitude increase of the iteration count. We conclude that the distance $|A_{j,s} - A_{i,s}|$ is inversely proportional to the number of iterations. At 10 000 iterations, the distance from the 500 000 iteration estimate is smaller than 10^{-3}.

Figure 3 shows the convergence of the area estimate with increasing sample counts. We compare four different methods for generating uniform random distributed points on the complex rectangle: pure random sampling, Latin Hypercube sampling, orthogonal sampling sampling. We also plot the error bound with a 95% confidence using Hoeffding's inequality that is valid for any Monte Carlo estimation with independent samples. All methods we considered converge much faster than the absolute bound. Pure random and Latin Hypercube sampling show similar convergence rates, while orthogonal sampling outperforms the other sampling methods on almost all tested sample sizes. Compared to the convergence from iteration count, increasing the sample size by a factor of 10 decreases the error much less than by a factor of 10. With the largest sample count, the estimates differ by about 10^{-3}.

We also test the impact of increasing the size of the numerical representation from the standard 64-bit representation of complex numbers to 128 and 256 bit representations. For this, we iterate 100 000 times over 100 000 uniformly sampled points. The 128 and 256 bit computations generate the same result, which differs from the 64 bit one by only 2 points.

3.3 Statistical Tests

In Figure 5, we provide a statistical summary of Orthogonal and Adaptive sampling methods, comparing their mean, standard deviation, t-statistic, and p-value. The comparison is based on 29 iterations, with sample sizes logarithmically spaced from 1,000 to 1,000,000.

Statistic	Orthogonal Sampling	Adaptive Sampling
Mean	0.0087	0.0114
Standard Deviation	0.0092	0.016
t-Statistic	-0.7856	
p-Value	0.4354	

Figure 5: Statistical Summary of Sampling Methods over 29 iterations, with sample sizes logarithmically spaced from 1,000 to 1,000,000.

4 Discussion

4.1 Accuracy of our Area Estimation

If we assume that the iteration ε_i error is indeed inversely proportional to the iteration count as suggested by our data, we can estimate it from the distance to a finite iteration count with a telescoping sum:

$$\begin{aligned}
\varepsilon_i(j) &= \lim_{i\to\infty} A_{j,s} - A_{i,s} \\
&= \lim_{i\to\infty} A_{j,s} - A_{100j,s} + A_{100j,s} - A_{i,s} \\
&\approx \lim_{i\to\infty} A_{j,s} - A_{100j,s} + \frac{1}{100} A_{j,s} - A_{i,s} \\
&= (A_{j,s} - A_{100j,s}) \sum_{k=0}^{\infty} \frac{1}{100} \\
&= 1.\overline{01}\,(A_{j,s} - A_{100j,s})
\end{aligned}$$

We conclude that our iteration error with 10 000 iterations is likely smaller than 10^{-3}. The sampling error ε_s is probabilistically bounded by Hoeffding's inequality. With a confidence of 95% and our maximum sample count of 10 million, inequality (1) evaluates to $\varepsilon_s \leq 4.52 * 10^{-3}$. Since our preliminary tests did not find a significant difference in estimates due to the numerical accuracy, we neglect the numerical error, as it is much smaller than ε_s and ε_i.
Using orthogonal sampling, which appeared to have the best convergence in our experiments, we arrive at a final area estimate of $A_M = 1.5069 \pm 0.0055$. We note that the convergence of our sampling methods is likely much better in practice than our upper bound given by inequality (1). Compared to the best pixel-counting estimate with 10^{13} samples and 10^8 iterations, our area estimate is only off by 0.00033.

4.2 Comparison of the Static Sampling Methods

All the sampling methods lead to estimates whose quality is better than the Hoeffding bound, as shown in Figure 3. Due to our method of first generating samples and then iterating over them, the sample generation takes up a relatively small part of the overall computation time. The points generated with orthogonal sampling appear to be more evenly distributed such that they are better able to capture the fractal shape of M, while Latin Hypercube sampling does not seem to be worth the (slight) increase in sampling cost over pure random sampling.

4.3 Statistical Analysis

The statistical analysis, as summarized in Figure 5, indicates that adaptive sampling does not significantly outperform orthogonal sampling in terms of convergence rate for estimating the area of the Mandelbrot set. Specifically, the mean error for orthogonal sampling (0.0087) was slightly lower compared to adaptive sampling (0.0114), and the corresponding standard deviations suggest that adaptive sampling showed greater variability. The t-statistic value of -0.7856 and p-value of 0.4354 indicate that there is no statistically significant difference between the errors produced by the two sampling methods. Thus, we fail to reject the null hypothesis ($\mathbf{H_0}$), implying that adaptive sampling does not significantly enhance convergence compared to orthogonal sampling under the conditions tested. Figure 4 visually reinforces this finding, as the convergence lines for both methods overlap significantly, showing similar trends in convergence rates.

In our previous comparison between random sampling, orthogonal sampling, and Latin Hypercube Sampling, orthogonal sampling consistently demonstrated superior performance, achieving lower estimation errors and better convergence rates. Given this context, orthogonal sampling emerged as the most effective traditional sampling method among those evaluated. The current results further reinforce orthogonal sampling as a robust approach, showing no significant benefit of using adaptive sampling over orthogonal under the examined conditions.

However, while adaptive sampling did not surpass orthogonal sampling, it is worth noting that adaptive sampling might still have a potential advantage over normal (random) sampling. Adaptive sampling aims to refine sample placement based on variance estimates, which theoretically could lead to more efficient use of samples compared to the purely random approach. To conclusively determine whether adaptive sampling provides a significant improvement over normal sampling, further testing and statistical analysis would be required.

4.4 Possible Improvements and Future Directions

Our sampling error is directly proportional to the total area from which points are sampled. Due to ease of implementation, we only sampled points from a four by four rectangle, while rejecting all points outside of a circle with a radius of two. This leads us to immediately reject $(4-\pi)/4 \approx 21\%$ of our sampled points. The sampling area can be further reduced by observing that the Mandelbrot set is symmetric about the real axis. Thus, sampling only points with positive imaginary values and doubling the estimate would again halve the sampling error.

Additionally, we did not have the computational resources or time to perform adaptive sampling with 10 million samples, which limited our ability to fully evaluate its performance at higher resolutions. Implementing adaptive sampling at larger scales could further enhance accuracy, especially for capturing complex boundary details. Future experiments should focus on capturing a larger sample set for more comprehensive analysis.

4.5 Conclusion

In this report, we showed that the area of the Mandelbrot set can be accurately estimated using Monte Carlo sampling methods. While the area of the Mandelbrot set is mostly of academic interest, the methods we explored are universal tools that can be applied on many kinds of approximation problems. We also found that computationally cheap adaptations to the sample selection can have a large impact on the approximation quality. While we could not conclusively determine that adaptive sampling outperforms orthogonal sampling, our exploration highlighted the benefits of focused sampling techniques, suggesting that further investigation into these approaches is worthwhile.

5 Task distribution

The code for our experiments in this paper is available in the GitHub repository: `https://github.com/PaulJungnickel/Stochastic_Simulation_Assignment_1`

Table 1: Git-Fame of the repository

Author	loc	coms	fils	distribution
Lucas-Keijzer	78519	8	14	95.4/26.7/51.9
Maarten Stork	1985	17	5	2.4/56.7/18.5
PaulJungnickel	1802	5	8	2.2/16.7/29.6

This assignment was collaboratively created in Google Colab, allowing us to work simultaneously on the same files. We used this collaborative approach to facilitate our work, but as a result, the division of code contributions as shown by Git-fame (depicted in *Table 1*) might not accurately reflect individual contributions since much of the work was done in Colab before transferring to this repository.

1. *Maarten*: Wrote the introduction and text on the Mandelbrot set. Covered sampling methods, created various visualizations. Implemented adaptive sampling and compared it to orthogonal sampling. Conducted statistical tests and discussed the findings.

2. *Paul*: Implemented Mandelbrot iteration in PyTorch. Contributed to Methods, Results, and Discussion sections for Tasks 1.2 and 1.3. Provided background information in the introduction.

3. *Lucas*: Created Mandelbrot visualizations and generated some plots. Contributed to parts of the Methods and Results sections. Documented random sampling and assisted in other tasks.

References

Andreadis, I., & Karakasidis, T. E. (2015, April). On a numerical approximation of the boundary structure and of the area of the Mandelbrot set. *Nonlinear Dynamics*, *80*(1), 929–935. Retrieved 2024-11-13, from `https://doi.org/10.1007/s11071-015-1917-4` doi: 10.1007/s11071-015-1917-4

Barnsley, M. F. (1988). *Fractals everywhere* (1st ed.). Academic Press.

Bittner, D., Cheong, W., Gates, D., & Nguyen, H. (2014, October). New Approximations for the Area of the Mandelbrot Set. *Involve, a Journal of Mathematics*, *10*. doi: 10.2140/involve.2017.10.555

Ewing, J. H., & Schober, G. (1992, December). The area of the Mandelbrot set. *Numerische Mathematik*, *61*(1), 59–72. Retrieved 2024-11-13, from `https://doi.org/10.1007/BF01385497` doi: 10.1007/BF01385497

Förstemann, T. (2012). Numerical estimation of the area of the Mandelbrot set. Retrieved 2024-11-13, from `https://www.foerstemann.name/labor/area/Mset_area.pdf`

Hammersley, J. M., & Handscomb, D. C. (1964). *Monte carlo methods*. Methuen.

Kostelich, E. J., & Schreiber, T. (1991). Estimating the area of the mandelbrot set. *Physica D: Nonlinear Phenomena*, *50*(2-3), 321–326.

McKay, M. D., Beckman, W. A., & Conover, J. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239-245. doi: 10.1080/00401706.1979.10489061

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087-1092. doi: 10.1063/1.1699113

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019, December). PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 8026–8037). Red Hook, NY, USA: Curran Associates Inc.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge University Press.

Robert, C. P., & Casella, G. (2013). *Monte carlo statistical methods* (3rd ed.). Springer.

Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the monte carlo method* (4th ed.). John Wiley & Sons.

Tishby, J. M., & Biham, O. (1992). Monte carlo integration and fractal dimension computation. *Journal of Statistical Physics*, *56*(3-4), 911–923.

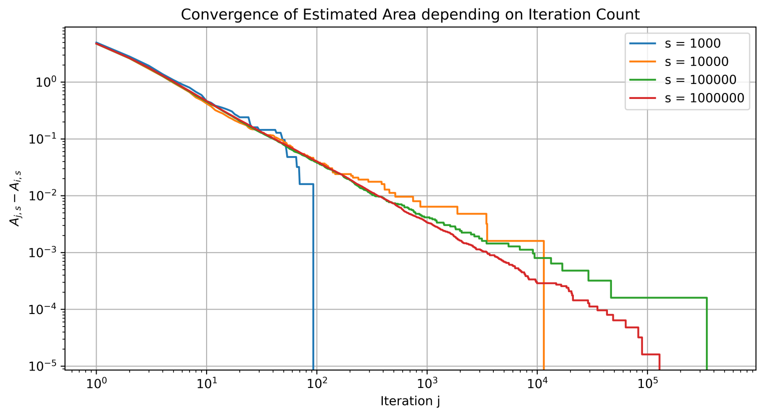


Figure 2: Difference of area estimates at each iteration from the final estimation for different sample sizes with uniform random sampling. The maximum number of iterations i is 500 000.

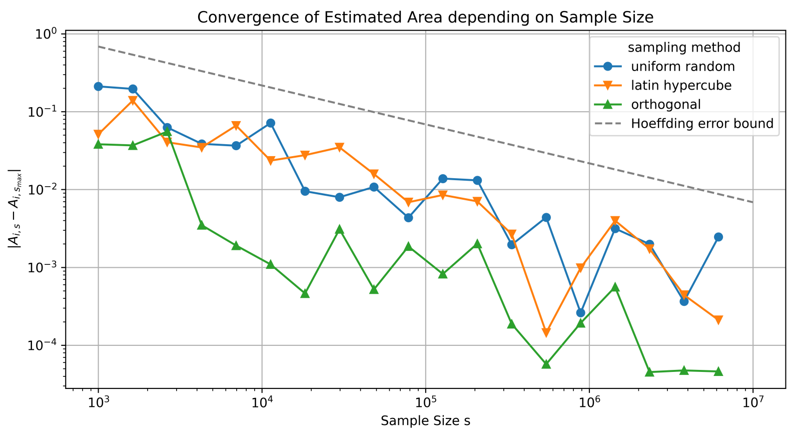


Figure 3: Convergence of the area estimates at different sample sizes towards the final estimation with 10 million samples and 10 000 Mandelbrot iterations. Four methods for generating samples are compared.

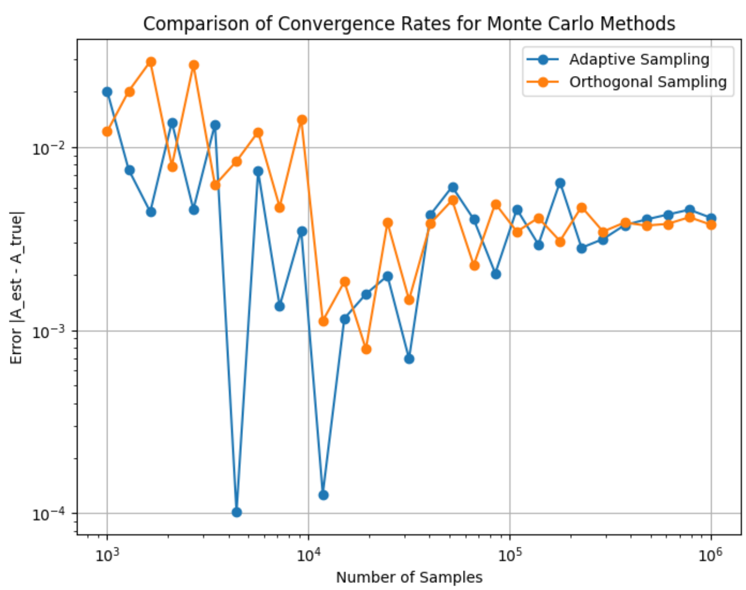


Figure 4: Comparison of convergence between orthogonal- and adaptive- sampling for different number of samples.