Investigating stochastic sampling techniques by estimating the area of the Mandelbrot set

Aron Golombek, 14064103

November 2021

Abstract

This report investigates the convergence behaviour of three distinct sampling techniques, namely pure monte carlo sampling, latin hypercube sampling and orthogonal sampling. This is done by means of approximating the area of the Mandelbrot set. The latter two methods introduce specific constraints on the way sample points are distributed in order to decrease the bias of a sample. Additionally control variates are tested on all sampling methods to study their effectiveness in reducing variance. The central aim is too investigate how accuracy can be improved and variance reduced without drastically increasing computational effort. The test used to ensure statistical significance are Welch's t-test for equality of mean and Levene's test for equality of variance. The paper finds that the results produced by all methods decrease significantly in variance as the sample size is increased. It can not be said conclusively that this convergence happens at different rates for different methods. Additionally it will be shown that the area estimate approaches the *true value* for all methods when the number of Mandelbrot iterations are increased. Regrading this it was also shown that statistically significant changes occur over a larger range of increasing Mandelbrot function iterations for latin hypercube sampling compared to pure monte carlo sampling and an even larger range for orthogonal sampling. Comparing the individual methods it can also be concluded that orthogonal sampling produces results with statistically significant difference in variance compared to both other methods over a large range of experiment parameters while the variances of latin hypercube and pure monter carlo sampling become less distinguishable as the parameters of Mandelbrot iterations and sample size are increased. Finally no statistically significant improvements could be measured when introducing control variates for any method. However, the results indicate that a different variable with a higher covariance to the area approximation of the Mandelbrot set could have this effect.

Introduction

Stochastic simulation relies heavily on the analysis of large amounts of data. This data is often generated using Monte Carlo simulation which chooses random points within the sample space. The simplicity of this method along with the computational power of modern computers make it a popular approach, often yielding acceptable results. However, more accurate results can be achieved by making relatively small adjustments to this method, resulting in less samples being required to obtain an equally accurate answer or achieving a more accurate answer with an equally large sample size. Essentially these methods are expected to converge to an accurate solution faster than a purely random generation of samples.

This report aims to analyze the convergence behaviour of three algorithms containing potential improvements to the Monte Carlo sampling method. These algorithms are built based on the principle of variance reduction. All algorithms will aim to estimate the area of the Mandelbrot set, and compared based on their results for varying algorithm parameters.

This report consists of three sections. Section 1 will introduce some general theory regarding the Mandelbrot set as well as providing an overview of the distinct sampling techniques and how these have been implemented. In section 2 the results are provided and their relevance discussed in detail. Finally, section 3 elaborates on how these results can be interpreted and what conclusions can be drawn from them.

1 Theory and Method

This section will introduce the Mandelbrot set and how its area can be estimated using stochastic simulation. Section 1.1 explains what condition needs to be met by a point in order for it to be part of the Mandelbrot set and how this can be evaluated using python. Section 1.2 explains how the area of the Mandelbrot will be approximated using stochastic methods in this report. Section 1.3 elaborates on the sampling methods investigated in this report. Pen-ultimately section 1.4 introduces a variance reduction technique which can be applied to all sampling methods and finally section 1.5 introduces the statistical significance tests used in this report.

1.1 The Mandelbrot Set

The Mandelbrot set is a specific set of complex numbers c for which the function displayed in eq. (1) does not diverge to infinity if evaluated iteratively starting at $z = 0$.

$$f(z) = z^2 + c \tag{1}$$

Hence, to test specific value of c, it is initially added to 0 becoming the new z value. Subsequently this z value is squared and c is added again leading to the new z value. This process is then repeated, if the function does not diverge, point c is concluded to be part of the Mandelbrot set.

This can be implemented in python by creating a function which preforms the process described above for a given number of iterations or until the magnitude of z exceeds two, as it can be mathematically proven that the point will no longer converge after fulfilling this criterion once. If after the given number of iterations this criterion has not been fulfilled it is concluded that the point is part of the Mandelbrot set. Figure 1 is the set generated with the algorithm described above. The shade of green indicates how many iterations were necessary for a point to diverge.

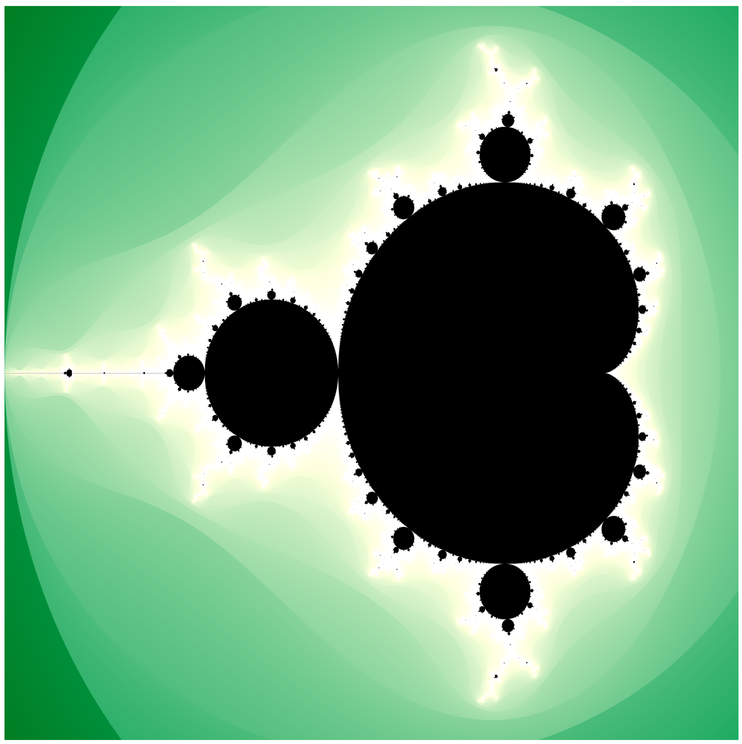


Figure 1: Mandelbrot set, 2000 divisions per side, number of iterations: $I = 1000$

1.2 Area Estimation

In order to be able to accurately estimate the area of the Mandelbrot set, a large number of points with an initial magnitude of less than two needs to be tested for divergence. Thus, the first step is to determine the sample size n, the number of points to be tested. Subsequently these points need to be generated using a specific sampling method, three possible methods are described in section 1.3. Next m the number of iterations, the number of times each point is evaluated for divergence for must be specified. Subsequently eq. (1) is tested for divergence for each point in the sample. The fraction of points which did not cause the function to diverge after m iterations multiplied by the area of the sample space the samples have been drawn from then gives an estimate for the area of the Mandelbrot set.

The method used for this paper stores the value 1 for each point for which eq. (1) did not diverge after m iterations and the value 0 if the eq. (1) diverged. The sum of this list divided by the number of points results in the fraction of tested points which are in the Mandelbrot set. Multiplying this value by 2.75×2.5 results in the initial area estimate as the real component of the samples are drawn from the half open interval $[-2, 0.75)$ with the imaginary component being drawn from the half open interval $[-1.25, 1.25)$.

1.3 Sampling Methods

Sampling Method refers to the method used to generate the sample of points which are tested for divergence. While these points can be generated completely randomly, there are ways to increase the accuracy of the result while keeping the number of samples and iterations the same. Therefore a better result can be achieved without requiring more computational effort. This can be achieved by reducing the variance of the simulation estimate. This section will introduce the four sampling techniques which will be investigated in this report.

1.3.1 Monte-Carlo Sampling

This sampling method for finding the sample points is purely random. It was implemented by using the *numpy.random.uniform()* function to generate two random floats from a uniform distribution. The values are generated in the half open interval $[-2, 0.75)$ and $[-1.25, 1.25)$ representing the real and imaginary component of c respectively.

1.3.2 Latin Hyper-cube Sampling

Latin Hyper-cube sampling is a variance reduction method based on stratified sampling. The method used to generate the sample has been adapted from the one described in Simulation by Sheldon Ross [4]. If n samples are required, the sample space is split into an $n \times n$ gird. Exactly one sample is then placed in each row and each column of the grid. In which combination columns and rows are combined remains to be determined by a random process, as well as the exact placement of the point within it's square of the grid. The exact method used to generate the samples using this technique is detailed below.

1. Generate 2 vectors with n components, $U_1 = (0, 1, ..., n-2, n-1)$ and $U_2 = (0, 1, ..., n-2, n-1)$
2. Generate a random permutation of each vector using *numpy.random.permutation*.

3. To create the real component of the first sample generate a random float between 0 and 1 using *numpy.random.random*. Subsequently add the first element of the permutation of U_1 and divide by n. Scale this result to fit the sample region by multiplying with the size of the domain and subtracting the magnitude of the most negative value in the domain.

4. To create the imaginary component generate a new random float and apply the same process as described for the real component but using the permutation of U_2 and scaling using the range rather than the domain.

5. Repeat steps three and four n times, generating a new random float every time and using the successive element in the permutations of U_1 and U_2.

1.3.3 Orthogonal Sampling

The orthogonal sampling technique [5]is an improvement upon the Latin Hypercube sampling method described above. It aims to reduce correlation by defining $\sqrt{n}$ equally probable sub-domains and making sure that the number of samples in each is equal. The only difference in implementation is that the permutation of vectors U_1 and U_2 occurs not entirely randomly but according to a specific pattern. The precise method is explained below.

1. Generate 2 vectors with n components, $U_1 = (0, 1, ..., n-2, n-1)$ and $U_2 = (0, 1, ..., n-2, n-1)$

2. U_1 is split into $\sqrt{n}$ individual vectors the first containing the $\sqrt{n}$ elements of U_1, the second containing elements $\sqrt{n}+1$ to $2\sqrt{n}$ and so on.

3. The permutations of these vectors are then found and combined back into a large vector U_{1p}.

4. The permutation of U_2 follows a different pattern, however the first step is identical to step 1 described above.

5. Now a new set of $\sqrt{n}$ vectors is defined, each consisting of exactly one randomly chosen element from each vector created in step 3.

6. A random permutation of each of these vectors is generated and these are then combined into vector U_{2p}

7. To create the real component of the first sample generate a random float between 0 and 1, add the first element U_{1p} and divide by n. Scale this result by multiplying with the size of the domain and subtracting the magnitude of the most negative value in the domain.

8. To create the imaginary component repeat step 7 but using the elements U_{2p} and scaling using the range rather than the domain.

9. Repeat steps six and seven n times, generating a new random float every time and using the successive element in the permutations of U_1 and U_2.

1.4 Control Variates

In order to further improve the convergence of the estimated value of the Mandelbrots set area, a method using control variates was implemented as a variance reduction technique. As stated before, the goal of the Monte Carlo simulation is to estimate the area of the Mandelbrot set. Instead of simply evaluating the expected value of the area, in this section it will be shown that we can obtain the same value using a different random variable. If $A_{i,s} = E[X]$, and for a variable Y the value of $E[Y] = \mu_Y$ is known, then we can define Z as:

$$Z = X + c \cdot (Y - \mu_Y), \tag{2}$$

Using the properties of the expected value we can say that:

$$E[Z] = E[X].$$

A specific value of c can be derived in order to minimize the variance of the variable Z, this is shown below[4]:

$$c = \frac{Cov(X, Y)}{Var[Y]} \tag{3}$$

Essentially if a variable Y is positively correlated with X, the variance of Z will be reduced.

For this application, the variable Y is defined as the estimation of the area of a circle centered at $(0, 0)$ and with radius $r = 0.5$. Therefore the value μ_Y is the area of that circle:$\mu_Y = \frac{\pi r^2}{2}$.

A positive correlation is plausible between this variables because if the point is inside the circle, then there is a high probability that it will also be in the set as the circle covers a big part of the set.

1.5 Hypothesis Testing

In order to be able to analyze how the different methods introduced above compare to one another significance testing will be used to ensure that the results obtained carry statistical relevance. Two central quantities will be important for the analysis of how the methods perform over a range of parameters and against each other, namely the mean and the variance.

Welch's t-test will be used to test for equality of mean, hence if two set's of data produce a mean which is likely to have been sampled from the same underlying distribution. This test has been chosen instead of a classical t-test as it accounts for the samples to have different variances[6]. It has been implemented using the python library *scipy*, specifically *scipy.stats.ttest_ind* with the argument *equal_var* set to *False* to ensure that Welch's t-test is performed rather than a classical t-test. This test return a p-value which can be interpreted in the following way: a p-value of 0.05 means that the confidence level that the two means are from different underlying distributions is 95%. A p-value of 0.01 means that the confidence level is 99%, hence the lower the p-value the higher the level of certainty that two samples stem from different underlying distributions. An often used and widely accepted level to show statistical relevance is a p-value of 0.05 or lower.

In order to assess the equality of variances for two samples Levene's test was employed [3][1]. It can be interpreted in the same way as described for Welch's test above and was implemented using *scipy.stats.levene* with the argument *center* being set equal to *'mean'* centering the samples around the mean, this was how the original test was proposed and remains to be recommended for samples from a normal distribution[3].

2 Results and Discussion

This Chapter consists of three main sections. Section 2.1 investigates the effect of increasing the sample size while keeping the number of Mandelbrot function iterations the same. Subsequently section 2.2 analyzes the effect of increasing number of Mandelbrot function iteration performed while keeping the sample size constant. Lastly section 2.3 explores the impact of using control variates in order to reduce variance for the distinct sampling techniques. Although the exact area of the Mandelbrot set is not know and can only be estimated there will be references to the *true area* throughout this chapter. This value, specifically $A_M = 1.506484$, will be used as a reference value as it has been computed using significantly more Mandelbrot function iterations and sample sizes which are several orders of magnitude higher [2]. Each plot in this section shows the mean value and standard deviation of 50 experiment repetitions for a given combination of Mandelbrot function iterations and sample size.

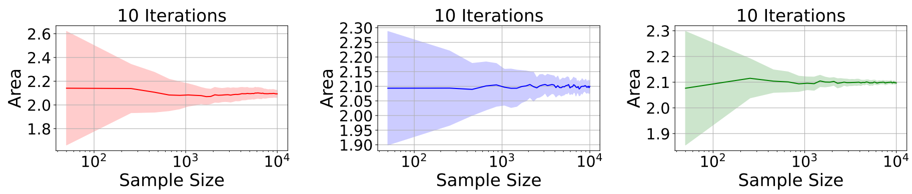
2.1 Investigating Effect of Increasing Sample Size

This section explores the effect of increasing the sample size while maintaining the same number of Mandelbrot function iterations. The approach can be summarized as follows.

1. Fix the number of Mandelbrot function iterations to I.
2. Fix the maximum number of samples to $S = 10^4$.
3. Calculate the area estimate $A_{s,I}$ as $s \to S$.
4. Logarithmically increase I and repeat steps 1 to 3.

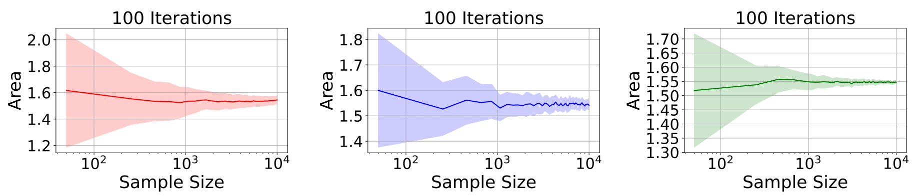
The results of these experiments have been plotted in fig. 2 to fig. 5. It is evident that regardless of the number of Mandelbrot function iterations preformed the variance of all methods decreases as the sample size is increased. However, this does not imply that the methods converge to the right value as using too little Mandelbrot iterations leads to an overestimation of the Area. This is intuitive as more points which in reality diverge after more iteration are considered to be part of the Mandelbrot set.

Furthermore, one can observe that on average the standard deviation achieved using pure monte carlo sampling is higher than the observed standard deviation of latin hypercube sampling while the standard deviation of orthogonal sampling is even lower.



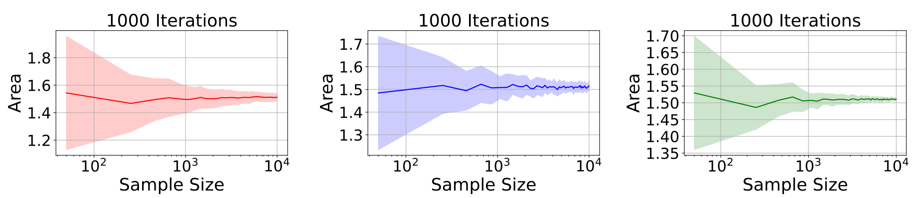
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 2: Method Convergence; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10 Mandelbrot Function Iterations, 50 experiment repetitions



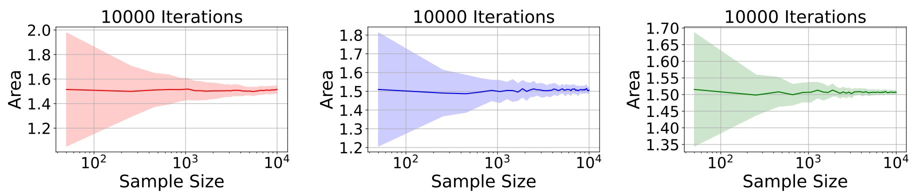
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 3: Method Convergence; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^2 Mandelbrot Function Iterations, 50 experiment repetitions



(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 4: Method Convergence; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^3 Mandelbrot Function Iterations, 50 experiment repetitions



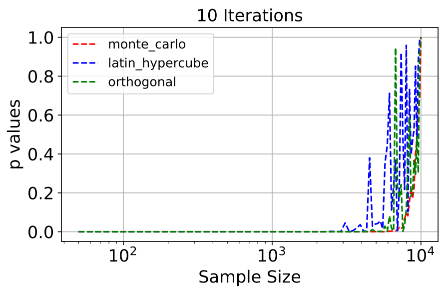
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 5: Method Convergence; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^4 Mandelbrot Function Iterations, 50 experiment repetitions

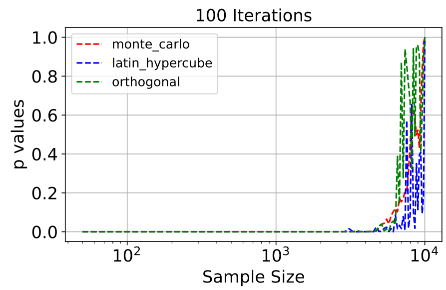
In order to confirm that the statements made above carry statistical relevance, significance testing needs to be performed. In order to analyze whether the mean value shows significant change as the sample size is increased Welch's test was used to compare the final data point (i.e the the largest sample size) to all previous points. The results of this were ambiguous with the resulting p-value showing major fluctuations and no clear patterns, this is consistent with what can be observed in fig. 2 to fig. 5 as we see that the mean value for the area estimation doesn't change drastically.

What does change drastically however are the standard deviations which decrease observably with increasing sample size. In order to test the significance of this the Levene's test was employed also comparing the final data points to all previous ones, the results are shown in fig. 6. Although there are some fluctuations close to the final value it is evident that the values with lower sample size, especially in terms of orders of magnitude differ significantly. To be precise the confidence level that the results obtained using sample sizes of different orders of magnitude differ in a statistically relevant way is above 99%. While this can only be verified on the graph for the order of magnitude $S = 10^4$, it has also been verified for the other orders of magnitude during testing.

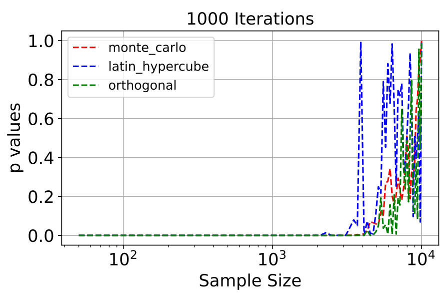
This testing did not yield conclusive results on the speed of convergence of the different methods as we can see that there are large fluctuations and that different methods seem to converge faster than others depending on the test parameters.



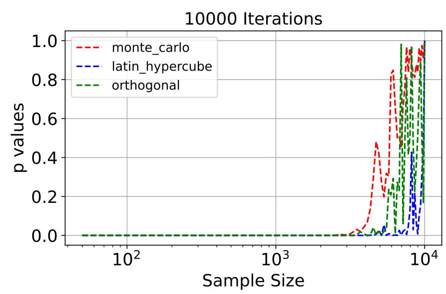
(a) 10^1 Mandelbrot Function Iterations



(b) 10^2 Mandelbrot Function Iterations



(c) 10^3 Mandelbrot Function Iterations

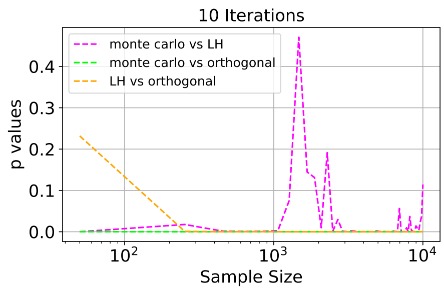


(d) 10^4 Mandelbrot Function Iterations

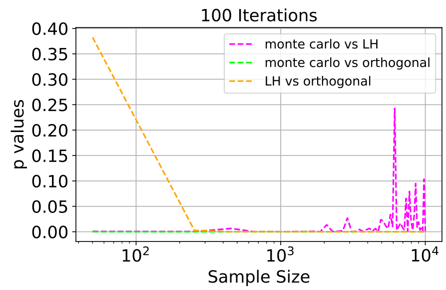
Figure 6: Levene's test p-value comparison; 50 points for Sample Size $[5 \times 10^1, 10^4]$, 50 experiment repetitions

Having analyzed the effect of increasing sample size for a given method, it is also compelling to analyze how the methods compare to one another for a given sample size. First Welch's test was used leading to no clear results. While there were isolated cases exhibiting statistically significant differences, these wee not part of a trend and could not be used to extract any useful conclusions.

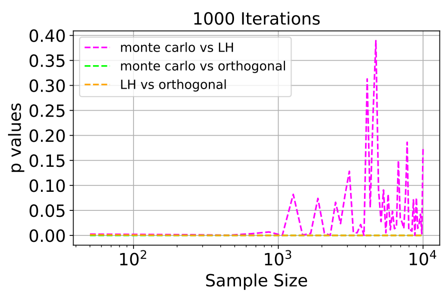
Similarly as before the next step was to employ the Levene's test, the results of which are plotted in fig. 7. It can be seen that for all parameter combinations pure monte carlo sampling is statistically significantly different from orthogonal sampling with an extremely high level of confidence ($\approx 99\%$). This indicates that indeed the standard deviation achieved when using orthogonal sampling is significantly lower. This is highlighted by the fact that for large parts range of sample sizes it also differs significantly from latin hypercube sampling. Finally it can be seen that latin hypercube sampling and pure monte carlo sampling exhibit statistically significant difference is the standard deviation they produce for low sample sizes. It is intuitive that this difference is offset once the sample size reaches a certain threshold where the principles of creating an unbiased sample are offset by the sheer size of the sample.



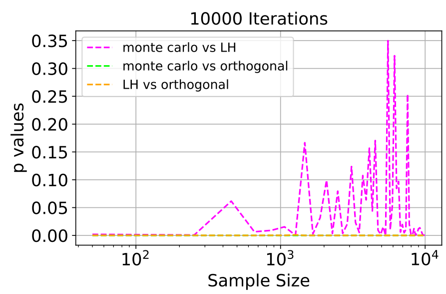
(a) 10^1 Mandelbrot Function Iterations



(b) 10^2 Mandelbrot Function Iterations



(c) 10^3 Mandelbrot Function Iterations



(d) 10^4 Mandelbrot Function Iterations

Figure 7: Levene's test method comparison; 50 points for Sample Size $[5 \times 10^1, 10^4]$, 50 experiment repetitions

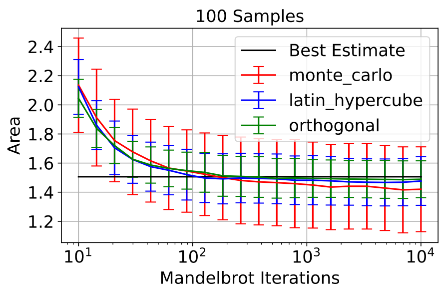
2.2 Investigating Effect of Increasing Mandelbrot Function Iterations

This section explores the effect of increasing the number of Mandelbrot function iterations performed while maintaining an equal sample size. The approach can be summarized as follows.

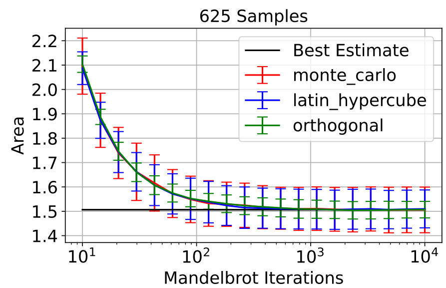
1. Fix the Sample Size S.
2. Fix the maximum number of Mandelbrot function iterations to $I = 10^4$.
3. Calculate the area estimate $A_{S,i}$ as $i \to I$.
4. Logarithmically increase I and repeat steps 1 to 3.

The results of these experiments for five different sample sizes can be seen in fig. 8. It can be observed that increasing the number of Mandelbrot iterations leads to the area estimate approaching its *true value*. The exceptions to this are very small samples where the probability of drawing a biased sample is a lot higher, this can be seen in fig. 8a.

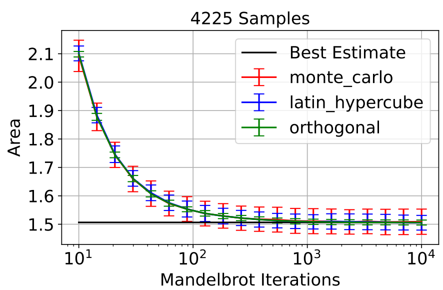
Furthermore, one can observe that as mentioned in the previous section the standard deviation of the area estimate decreases for all methods when the sample size is increased. Additionally, the observation that the standard deviation is consistently smallest for orthogonal sampling, second smallest for latin hypercube sampling, and largest for random monte carlo sampling can be seen more clearly. Lastly, it becomes evident that with increasing sample size the methods become increasingly similar in their performance as the samples become so large that any initial bias becomes negligible.



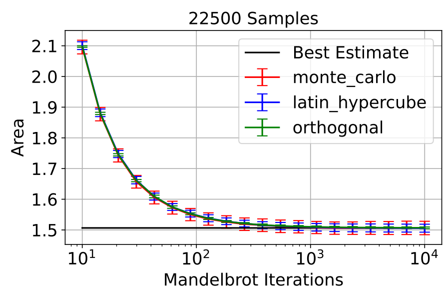
(a) 10^2 Samples



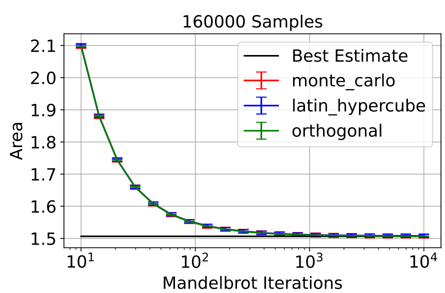
(b) 6.25×10^2 Samples



(c) 4.225×10^3 Samples



(d) 2.25×10^4 Samples



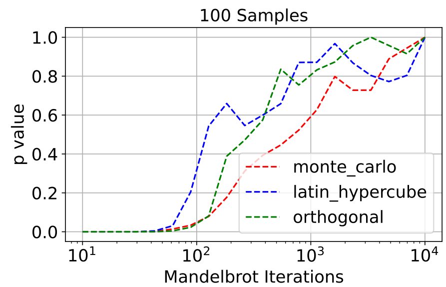
(e) 1.6×10^5 Samples

Figure 8: Method Convergence; 20 points for Mandelbrot Function Iterations $[10^1, 10^4]$, 50 experiment repetitions

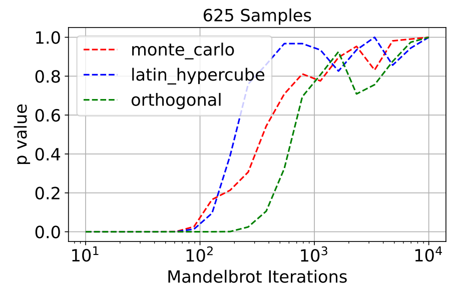
In order to test how significant the differences in estimated area values using different number of Mandelbrot function iterations are Welch's test was applied.

For each method, the final value and hence the most accurate of the simulation was compared to each previous value. The results are shown in fig. 9. Two important observations can be made. Increasing the sample size results in each method increasing the accuracy of its result for a larger range of Mandelbrot function iterations. This can be seen in the fact that for a sample size of 100 the p-values start to increase dramatically around 10^2 Mandelbrot iterations indicating that there are no longer statistically significant differences. This increase of p-values is delayed by a whole order of magnitude when increasing the sample size to 1.6×10^5.

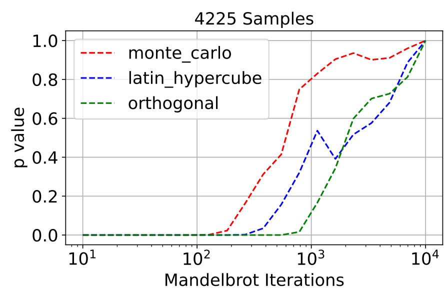
Secondly we see that for all sample sizes except for very small samples (i.e $S = 100$) the p-value for latin hypercube sampling increases at a higher number of Mandelbrot iterations than. The p-values obtained using orthogonal sampling increase at an even later stage. This shows that given a specific sample size, merely increasing the number of Mandelbrot function iterations will not necessarily improve the accuracy of the solution. However, using methods to decrease the bias in the sample results in improvements being visible over a larger range of values. The Levene's test was also applied in the same manner yielding the result that increasing the number of Mandelbrot iterations for a given method and sample size does not decrease the standard deviation of the results. Looking at the results in fig. 8 this was the expected outcome.



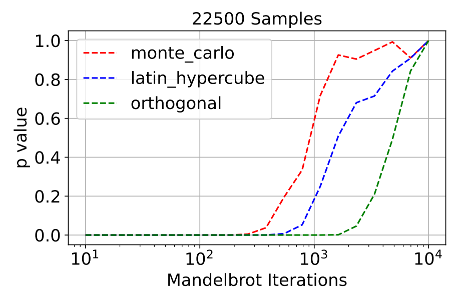
(a) 10^2 Samples



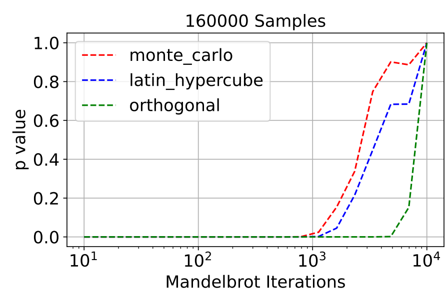
(b) 6.25×10^2 Samples



(c) 4.225×10^3 Samples



(d) 2.25×10^4 Samples



(e) 1.6×10^5 Samples

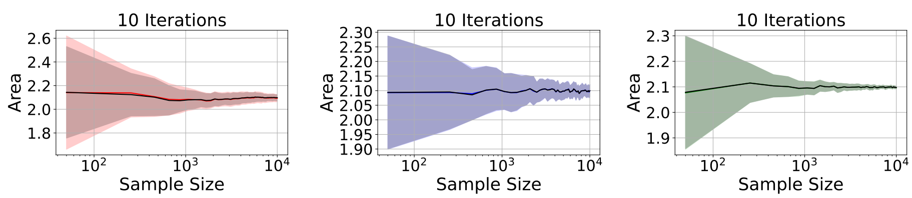
Figure 9: Welch's test p-value comparisson; 20 points for Mandelbrot Function Iterations $[10^1, 10^4]$, 50 experiment repetitions

Furthermore the methods were compared across each other using Welch's test to see if their performance differed significantly from one another. While there were again isolated cases for low sample sizes where a statistically significant change was detected, this was not part of a general trend.

The Levene's test confirmed what was already observed before that the different methods produce results with statistically significantly different standard deviations for most of the parameter combinations. Specifically for increasing sample size the p-values comparing latin hypercube to orthognal sampling became smaller indicating that the benefit of orthogonal sampling is amplified when increasing the sample size. On the other hand the p-values comparing latin hypercube and pure monte carlo sampling grew with increasing sample size indicating that the results of the methods became harder to distinguish. Finally the results produced by orthogonal sampling were always statistically significantly different from pure monte carlo sampling with a p-value close to zero.

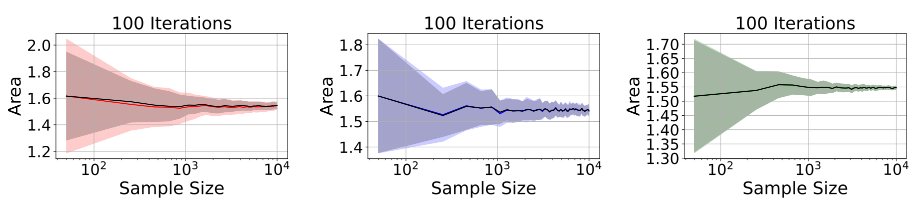
2.3 Investigating Effect of Introducing Control Variates

For the final set of experiments, the effectiveness of using control variates to reduce the variance of the data set as outlined in section 1.4 was investigated. As seen in the previous sections, change of variance can best be analyzed when investigating the behaviour for increasing sample size. The results of the experiments have been visualized in fig. 10 to fig. 13 with the colour representing the results without control variates and the black plot representing the outcome using control variates.



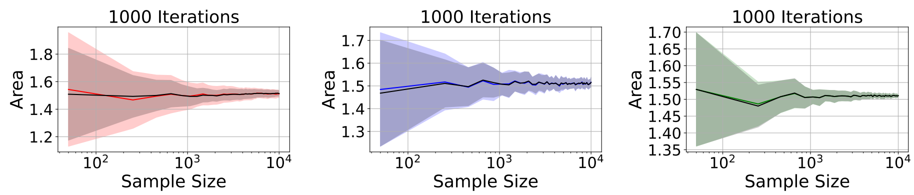
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 10: Testing Control Variates; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10 Mandelbrot Function Iterations, 50 experiment repetitions



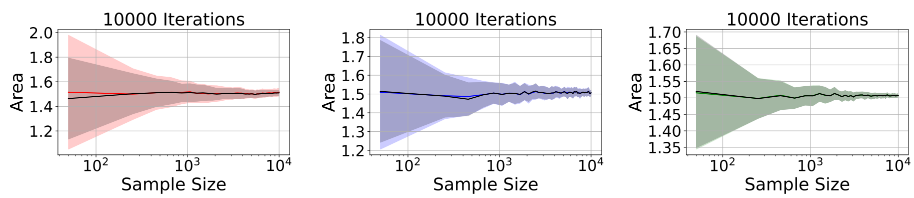
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 11: Testing Control Variates; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^2 Mandelbrot Function Iterations, 50 experiment repetitions



(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

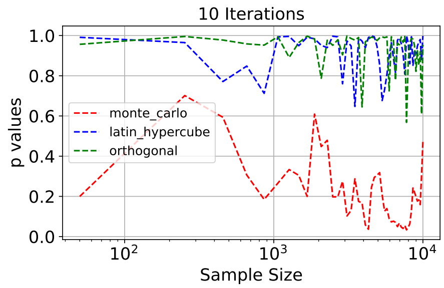
Figure 12: Testing Control Variates; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^3 Mandelbrot Function Iterations, 50 experiment repetitions



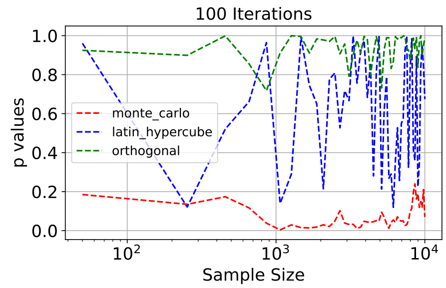
(a) Monte Carlo Sampling (b) LH Sampling (c) Orthogonal Sampling

Figure 13: Testing Control Variates; 50 points for Sample Size $[5 \times 10^1, 10^4]$ and 10^4 Mandelbrot Function Iterations, 50 experiment repetitions

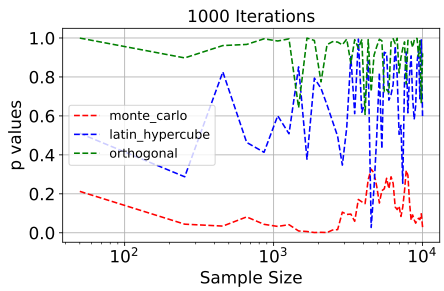
While visually one can observe that the variance is reduced especially for pure monte carlo sampling but also latin hypercube sampling, this needs to be verified using a statistical significance test. While the assumption that the mean won't differ significantly was confirmed using Welch's test, the difference in varaiance between the approaches was anlyzed using a Levene's test for which the resulting p-values have been plotted in fig. 14.



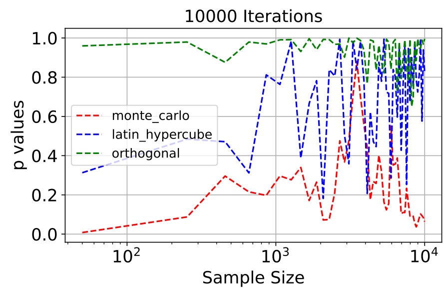
(a) 10^1 Mandelbrot Function Iterations



(b) 10^2 Mandelbrot Function Iterations



(c) 10^3 Mandelbrot Function Iterations



(d) 10^4 Mandelbrot Function Iterations

Figure 14: Levene's test p-value comparison for control variates; 50 points for Sample Size $[5 \times 10^1, 10^4]$, 50 experiment repetitions

While it cannot be said that using control variates leads to significant statistical difference for any method, one can observe that the method has the biggest effect of monte carlo sampling, followed by latin hypercube sampling with almost no effect on orthogonal sampling. Although not significant, there is an observable effect, implying that a different variable with a larger covariance in relation to the area of the mandelbrot could lead to a significant improvement.

3 Conclusion

As shown in this paper the sample size and Mandelbrot function iterations applied can have significant effect on the results obtained for the estimate of the Mandelbrot area. It was shown that while not having a statistically significant impact on the mean value increasing the sample size for any method drastically decreases the variance resulting in a higher level of confidence in the answer. Conversely, increasing the number of Mandelbrot iterations did not show a relevant impact on the variance but it could be shown that the mean area estimate approached the *true area* of the Mandelbrot set. It could additionally be observed that increasing the sample size and Mandelbrot iterations indefinitely did not necessarily yield better results. Instead the combination of parameters as well as the sampling method used determines up to which point statistically relevant differences could be observed.

It was shown that above a small threshold of sample size, orthogonal sampling produced significant difference, in terms of average area estimate, for a larger range of increasing Mandelbrot iterations than the other methods. While latin hypercube sampling out performed pure monte carlo sampling in the aspect, the difference between these methods was smaller than the gap to orthogonal sampling. Furthermore the gap to orthogonal sampling increased as the sample size was increased while the gap between monte carlo and latin hypercube sampling seemed to shrink. This effect is likely due to the difference in variances of the methods, as a lower variance allows significant difference to be detected even though the change in mean is similar.

Furthermore it was shown that increasing the sample size lead to more significant differences between latin hypercube and orthognal sampling in terms of variance, but reduced the relevant difference between latin hypercube and monte carlo sampling. It can therefore be concluded that from the tested sampling methods orthogonal sampling performs the best as it produces significantly lower variances while increasing the accuracy of its average estimate for a larger range of Mandelbrot function iterations. Additionally latin hypercube sampling outperformed pure monte carlo sampling. Lastly the gap between the latter two methods shrunk while increasing the experiment parameters while the gap from both of them to orthogonal sampling grew. However, this effect is expected to be offset past a specific threshold of sample size as such a large proportion of the sample space will be sampled by any method.

While using control variates to lower the variance of the individual sampling techniques could not been proven to have a significant statistical impact, it seems plausible that a different choice of variable, having a larger covariance with the estimated area of the Mandelbrot set could produce the desired results.

This paper has shown that significant differences both between experiment parameters and sampling methods exist. For future research more variance reduction techniques can be explored and other potentially simple changes like exploiting the symmetry of the Mandelbrot set about the real axis could yield more accurate results.

References

[1] Joseph L. Gastwirth, Yulia R. Gel, and Weiwen Miao. The Impact of Levene's Test of Equality of Variances on Statistical Theory and Practice. *Statistical Science*, 24(3):343 – 360, 2009.

[2] Kerry Mitchell. A Statistical Investigation of the Area of the Mandelbrot Set. 2001.

[3] I. Olkin. *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford studies in mathematics and statistics. Stanford University Press, 1960.

[4] Sheldon M. Ross. *Simulation*. Knovel Library. Academic Press, Amsterdam, fifth edition. edition, 2013.

[5] Boxin Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993.

[6] Bernard Lewis Welch. The Generalization of 'Student's' Problem When several different Population Variances are involved. *Biometrika*, 34(1-2):28–35, 1947.