The stochastic approximation of the Mandelbrot area

Jasper den Duijf

Koen Greuell

Abstract

The Mandelbrot set is a set of complex numbers with an infinite complicated boundaries. In the set are all complex numbers c that $z_n = z_{n-1}^2 + c$ with $z_0 = 0$ do not diverge after a number of iterations n. Since its discovery in 1978 the properties has been calculated in more and more detail, although certain properties have only been caught in (confidence) intervals. Since there is no good deterministic way to calculate the area, because integration of the set is extremely difficult, mathematicians came up with the stochastic Monte Carlo approach; taking a high number of random points, check whether they are in the Mandelbrot set or not and calculate the area from this ratio.

But a different number of possibilities to do so exist, mainly differing on the way the random points are chosen. In this paper we'll compare pure random sampling, with latin hyper cube sampling, orthogonal sampling and importance sampling. We will investigate the effects of the number of samples and iterations. Eventually importance sampling turns out to cause the lowest variation.

Contents

1 Introduction **4**

2 Theory **4**

2.1 The Monte Carlo integration 4

2.2 The Mandelbrot set 5

2.3 The Central Limit Theorem 5

3 Methods **5**

3.1 Simulation 6

3.2 Iterations 6

3.3 Sample selection 7

3.3.1 Pure random sampling 7

3.3.2 Latin hypercube sampling 8

3.3.3 Orthogonal sampling 9

3.3.4 Importance sampling 10

3.4 Convergence 11

4 Results **11**

4.1 Introduction 11

4.2 Iterations 11

4.3 Pure random sampling 12

4.4 Latin hypercube sampling 13

4.5 Orthogonal sampling 14

4.6 Importance sampling 15

4.7 Comparing results 16

5 Discussion **17**

1 Introduction

The Mandelbrot set was first defined and drawn in 1978 by Robert W. Brooks and Peter Matelski[1]. It is a fractal, an infinitely self-similar mathematical set in the complex space. This means that the form keeps repeating itself on the edges when zoomed in. Mathematicians have been drawn towards it because of its aesthetic beauty when plotted, but also because of its complex properties. Since the borders are erratically changing whenever zoomed in, it has been difficult to determine the exact area. With a method called pixel counting and a lot of computing power mathematicians have narrowed the area down to a very small interval. However there are still no methods developed to calculate this number numerically.

This is not the first time in history mathematicians have tried to narrow an infinite number down: the search of the exact value of π has been a challenge for centuries. In the effort to approximate the exact value of π Georges-Louis Leclerc, Comte de Buffon was the first to describe Monte Carlo integration buffon. Monte Carlo integration can determine the area of a circle and is thus able to calculate π. This stochastic approach can also be used on a fractal by determining whether random points are in the fractal or not. Different ways to choose these random points exist. All of these sampling methods are expected to return area estimations converging to the true area of the fractal as the amount of samples increases. This we will show in this paper which sampling methods can best be used to estimate the Mandelbrot area with the highest accuracy and precision.

We will start explaining the theory behind the Monte Carlo integration, the Mandelbrot set and the Central Limit Theorem in chapter 2. In chapter 3 we will discuss the different sampling methods in greater detail, as well explain the simulation. In chapter 4 we'll show the results of our simulations for all the different sampling methods and compare them. In chapter 5 we'll discuss out findings.

2 Theory

2.1 The Monte Carlo integration

Monte Carlo integration is a non-deterministic approach to calculate the area of any object. It is one of the uses of the Monte Carlos methods, which tend to follow the same pattern: firstly a domain for the inputs is set, then a number of random inputs is created in this domain and eventually in a deterministic way those inputs are used to calculate a result. It is broadly used for computations that are impossible or computationally too expensive to do deterministic, like predicting or optimizing complicated models. In the case of calculating the area of an object by Monte Carlo integration, the sampling domain will be a two-dimensional square with an area

A. The edges of this square will completely surround the object. Random coordinates between the square's limits are created. For all those coordinates we calculate whether those points are on the object's area or not, receiving a ratio between those numbers. By multiplying this ratio with area A, we approximate the area of the object[4].

2.2 The Mandelbrot set

The Mandelbrot set is a set of complex numbers c for which the series $z_{n+1} = z_n^2 + c$, with $z_0 = 0$ does not diverge for large n. 'n' is called the number of iterations. Numbers outside of this set are unbounded as they to go to ∞ or $-\infty$. When we only check the real numbers $c \in \mathbb{R}$, we see that the Mandelbrot set exist on interval $[-2, 0, 25]$. However for $c \in \mathbb{C}$ this interval is more difficult to describe.
One of the Mandelbrot set's most well known features is that its borders almost endlessly repeat itself as we zoom in. This aesthetic effect is what made it so popular by mathematicians. Exactly determining the border of the set turns out to be difficult, therefore the exact area of the Mandelbrot has not been calculated yet.
When z_n's absolute value becomes larger than 2, we know that the sequence will diverge. But even when the number of iterations grows, and c seems to be in the set after the iterations, there will always be an error ϵ such that $c + \epsilon$ will diverge.
Kerry Mitchell[5] calculated in 2011 a confidence interval for the Mandelbrot set, saying it lies with a 95% certainty in the interval $[1.506480, 1.506488]$. Ever since then this interval has narrowed down to even width of 10^{-8}.

2.3 The Central Limit Theorem

One of the most important rules in stochastic simulation is the Central Limit Theorem. It say that the sample mean of every distribution has a normal distribution. The stochastic $\bar{x}$ is the mean of the sample X, μ is the expected value of X $E[X]$ and σ^2 is the variation of X. Now we know that $\bar{x}$ follows the distribution of $Y \sim N(\mu, \sigma^2)$.
The central limit theorem suggests that the amount of samples is negatively correlated to the variation in the surface area estimation. As numbers are considered to be in the Mandelbrot set when |real| ¡ 2 after i iterations and |imaginary| ¡ 2 after i iterations, there should be a negative correlation between the amount of iterations and samples categorized as Mandelbrot numbers.

3 Methods

First we investigated the effect of varying the amount of iterations on the accuracy and precision of the estimation of the Mandelbrot set area. Next we investigated the effect of varying the sample sizes and finally applied four different stochastic sampling techniques at different sample sizes.

3.1 Simulation

Unless otherwise specified the position of the samples was determined by combing a pseudo random real number with a pseudo random imaginary number. Pseudo random numbers were generated using the random package in Python which generates numbers using Mersenne Twister [3]. In the case of latin hypercube sampling and orthogonal sampling rows and columns were randomly combined using the Same Mersenne Twister technique.

3.2 Iterations

We can increase the number of iterations in every simulation, which will calculate a higher value of z_n. Especially a low number of iterations can cause a problem, as these iterations decide how precise the estimations of the real area is. To explain this, we refer to figure 1.

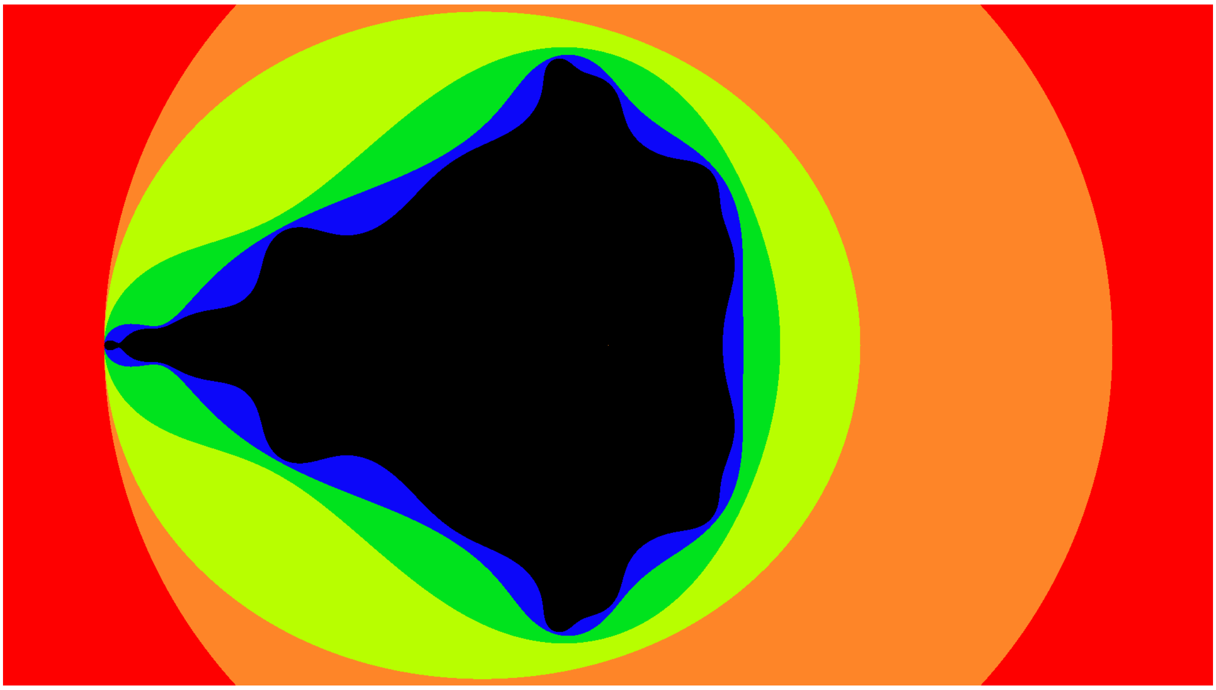

Figure 1: The number of iterations is negatively correlated with the area of the Mandelbrot set. In orange 1 iteration, yellow 2 iterations, etc. The more iterations, the smaller the Mandelbrot area becomes. So with a finite number of iterations, we will always have a slightly higher area.

So a high number of iterations is necessary to give an unbiased estimation. In chapter 4 we will check how the results change given a certain number of iterations. However, to compare the sampling methods, we will choose a fixed number of iterations. We know by doing so that we are actually estimating an area slightly bigger than the real area.

3.3 Sample selection

Every calculation of the Mandelbrot area requires N samples. The samples taken were selected by four different techniques: Pure random sampling, Latin hypercube sampling, Orthogonal sampling and importance sampling. Mandelbrot set numbers were presumed to only exist in the space of real range [-2, 2] and imaginary range[-2, 2] [6]. For each sample selection technique different amounts of n samples were tried.

3.3.1 Pure random sampling

Pure random sampling requires two random numbers both between 0 and 1 for every sample. This number is multiplied by 4 and subtracted with 2. One of the numbers is multiplied with i. Then summed up together these points imaginary part of the complex number. Every point is independent from all others. We see an example in figure 2 with 7000 samples.

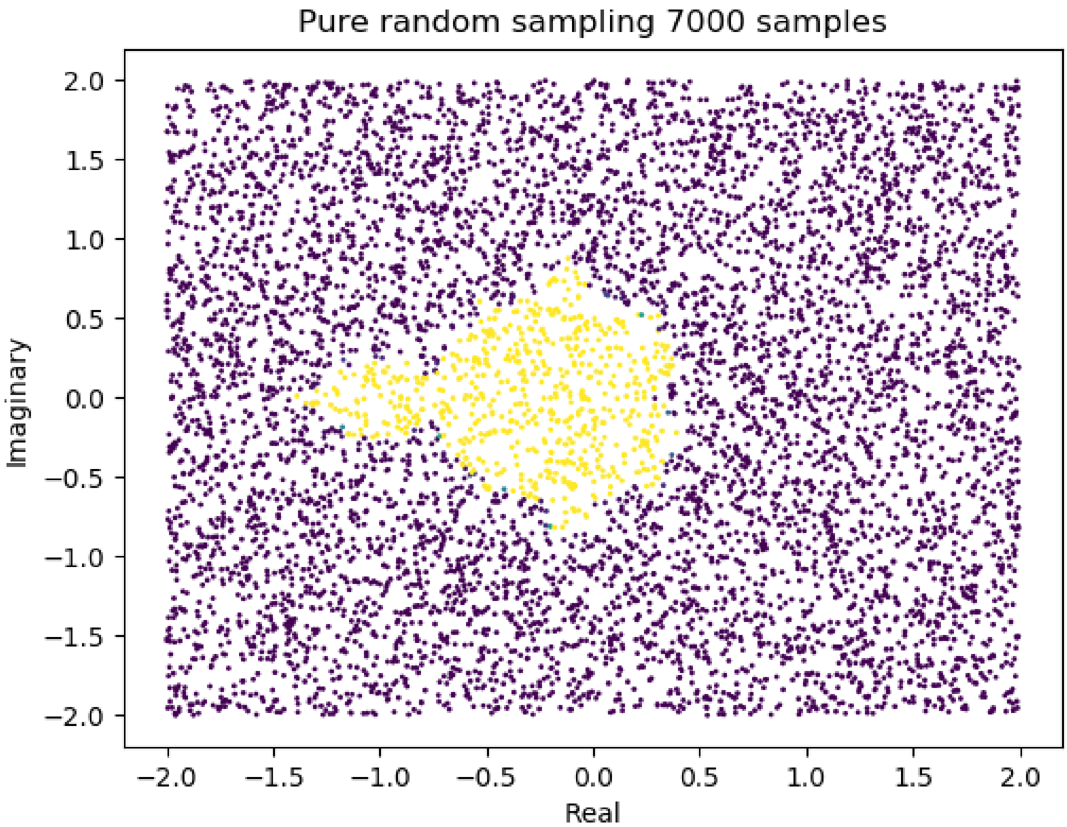

Figure 2: To illustrate random sampling and Monte Carlo integration there are only 7000 points in this figure (Pure random sampling).

3.3.2 Latin hypercube sampling

In order to apply latin hypercube sampling we need to create a grid over the area. This divides the space into N^2 squares of even size. Then an algorithm picks at random a row and column, so that in every row and column a square was picked exactly once. Within each of these selected squares a random position was evaluated as described in pure random sampling, only with different borders. See Figure 3 for an example with a low number of points.

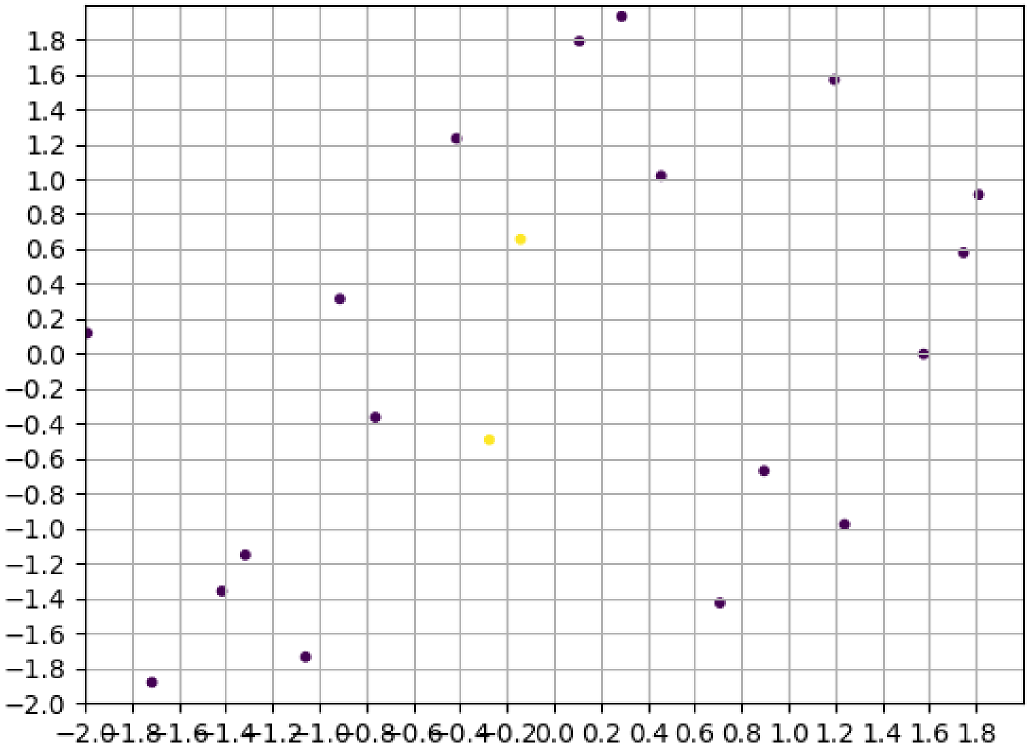

Figure 3: To illustrate latin hypercube sampling there are only 400 squares in the grid. In this figure we can see how each point appears to be completely random, yet in every row and every column only one point is placed.

This method can be used to lower the variance of the Monte Carlo integration, as no longer all points are chosen completely random, but in a fixed domain and thus more controlled. However this method doesn't change anything how unbiased the estimator is. These result are discussed in chapter 5

3.3.3 Orthogonal sampling

Orthogonal sampling divided the space into n cubes of even size and selected a randomly combined a cube row and cube column similarly to the technique done in latin hypercube sampling. Then the space within each cube was divided similarly to the technique in which the cubes were made and each unique space within a cube was assigned to one cube.

3.3.4 Importance sampling

Importance sampling is a method developed to decrease the variance of the sample mean by focusing on the important domains. If we consider the Mandelbrot set, we see that the most important parts near the border. The further we move from the border, the more likely we can predict the outcome of the sample. So we would like to remove this part of the variance by increasing the samples near the border.

To do so, we have to define the borders. To do so we will create a grid and take a first round of samples in every square. With this information we can determine whether a square in the grid is outside of inside the Mandelbrot set: if the variance in a grid is 0 and thus all of the samples result the same, we give those squares a low priority status. We then do a second round of sampling, but focus on the grids with a variance other than 0, since those grids have points in and out of the figure. For a graphical explanation, see figure 4.

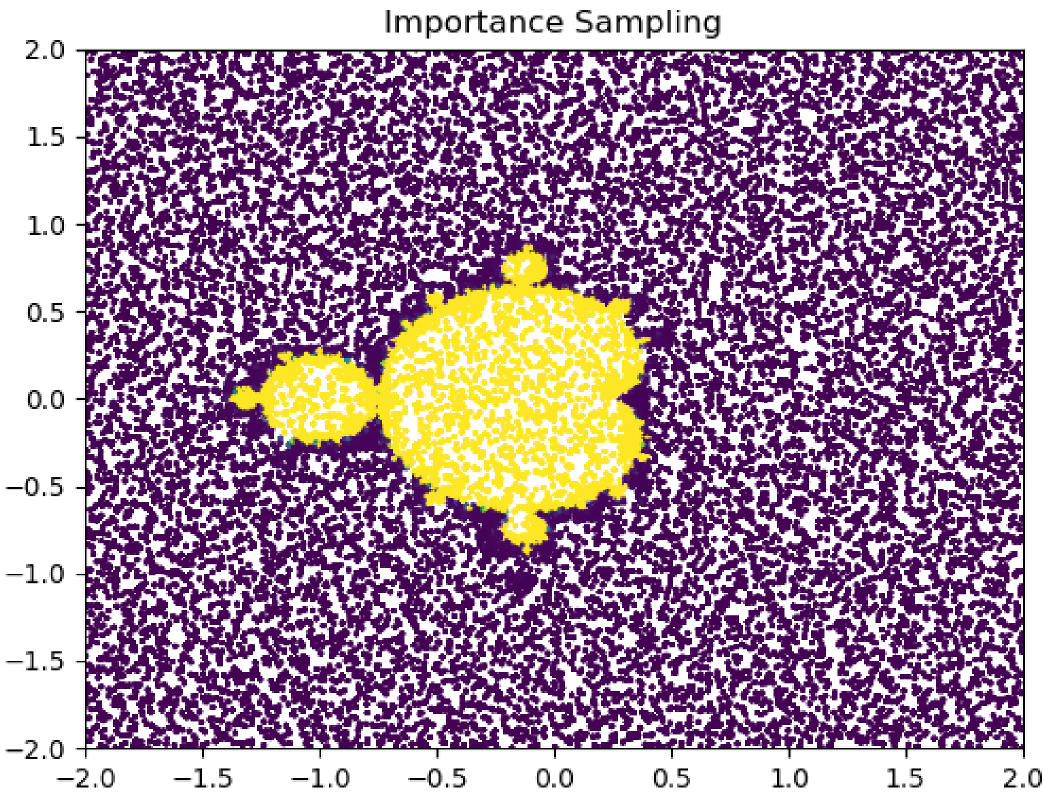

Figure 4: To illustrate importance sampling we have plotted in 1600 points. We see that the density of the points is higher near the borders of the Mandelbrot set.

3.4 Convergence

We defined the precision of the estimates to be the standard deviation of the estimates. We defined the accuracy of the estimations to be $|\mathrm{mean}(\mathrm{A}_{\mathrm{i,s}}) - \mathrm{A}_{\mathrm{M}}|$.

4 Results

4.1 Introduction

In this chapter we'll analyze the results for every different sampling method and finally compare the different methods to define the best approach to calculate the Mandelbrot set area.

4.2 Iterations

The amount of iterations is not related to the accuracy or precision in determining the area of the Mandelbrot set (figure 5). Increasing the amount of iterations does not decrease the variation in area estimation and only for very small amounts of iterations the area seems to be slightly overestimated. Since the variation is clearly larger than this error, we will set the number of iterations to 1000 and try to minimize the variance in this chapter.

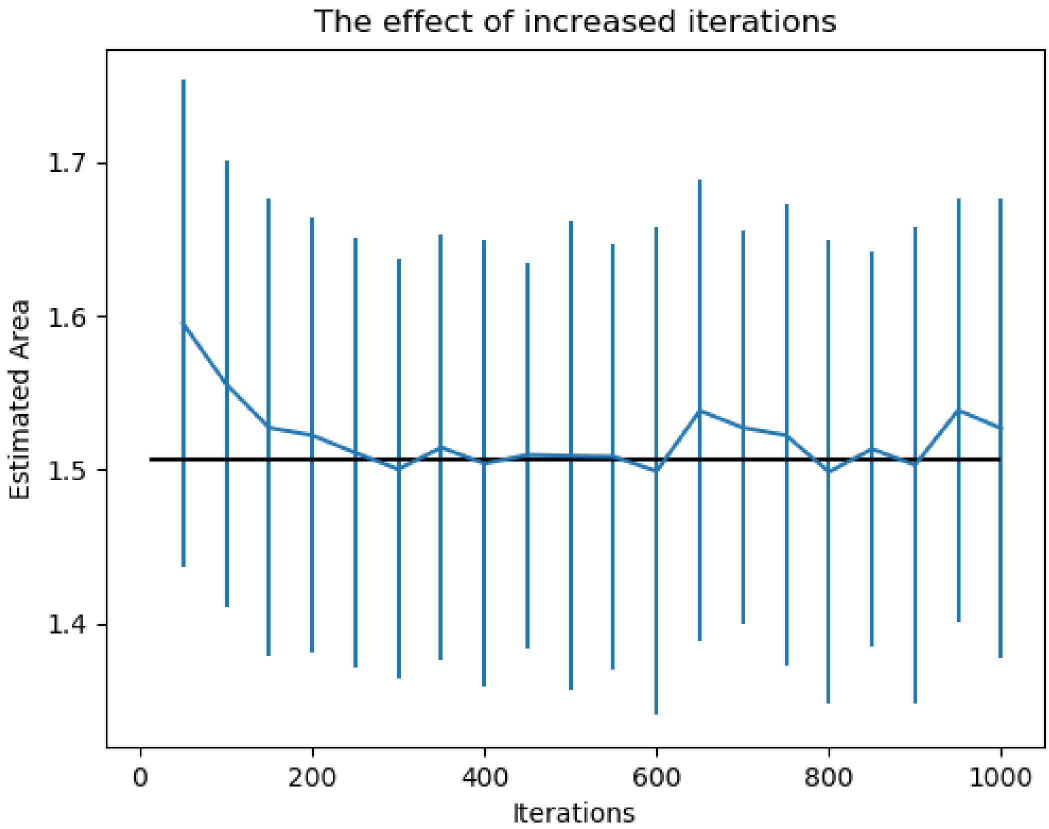

Figure 5: Mean and standard deviation of the estimated area at different amount of iterations. Per amount of iterations a 1000 samples were taken and this process was repeated 100 times. The pure random sampling method was used.

4.3 Pure random sampling

We have ran the simulation for pure random sampling multiple times. We continue to keep the number of iterations steady on 1000, but we multiply the number of samples each time with 10. We calculate in this way the area of the Mandelbrot a thousand times and calculate the mean and standard deviation of our simulations. The given results are shown in figure 6.
We can clearly see the effects of increasing the number of samples. The real given area of the Mandelbrot set is plotted with the vertical black line. Every time this value is in in the interval of $[\mu - \sigma, \mu + \sigma]$, but we more importantly see that the standard deviation decreases as the number of samples increasing. It converges to the real value.

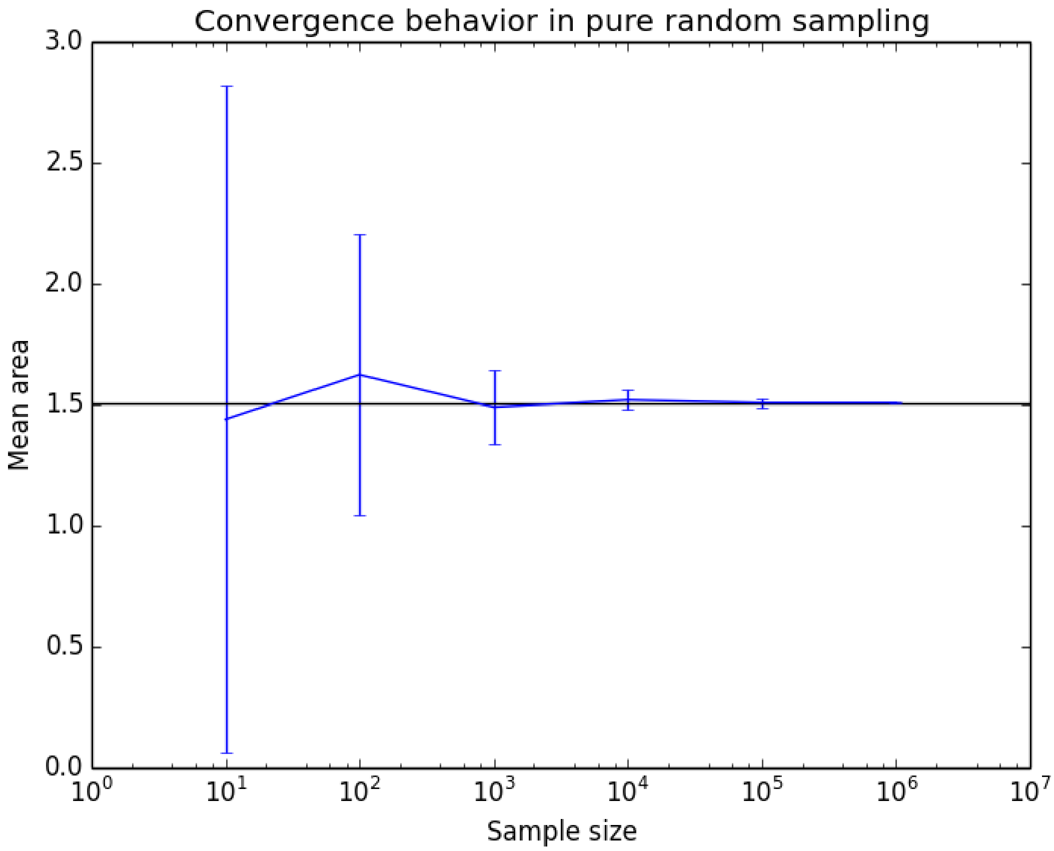

Figure 6: The average of the Mandelbrot size does not change with the increase of samples (since this is unbiased), but the variance clearly decreases. Black is given the real value. With 10^6 samples the standard deviation is a mere $4.29 * 10^{-3}$.

4.4 Latin hypercube sampling

We have ran the simulation for latin hypercube sampling multiple times. We continue to keep the number of iterations steady on 1000, but we multiply the number of samples each time with 10 (and thus the number of squares in the grid with 10^2). We calculate in this way the area of the Mandelbrot a thousand times and calculate the mean and standard deviation of our simulations. The given results are shown in figure 7 We can clearly see the effects of increasing the number of samples, just like pure random sampling. The real given area of the Mandelbrot set is plotted with the vertical black line. Every time this value is in in the interval of $[\mu - \sigma, \mu + \sigma]$, but we more importantly see that the standard deviation decreases as the number of samples increasing.

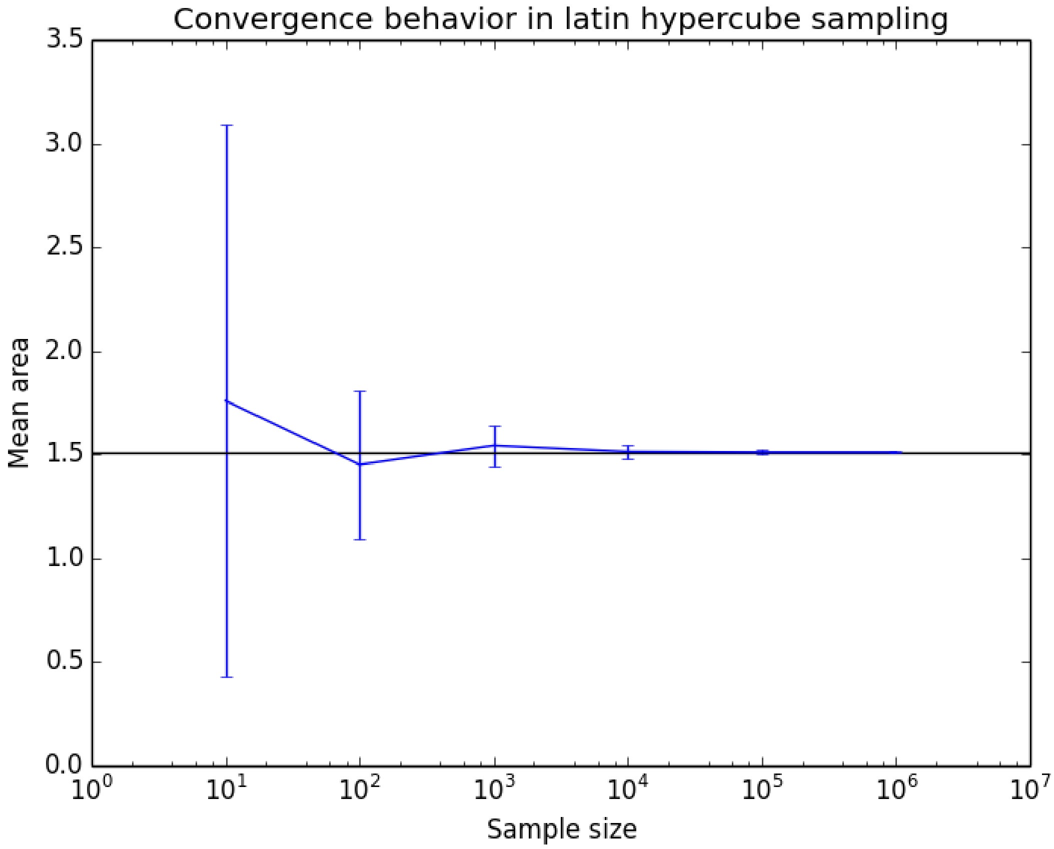

Figure 7: The average of the Mandelbrot size does not change with the increase of samples (since this is unbiased), but the variance clearly decreases. It converges to the real value, represented with the black line. With 10^6 samples the standard deviation is a mere $3.29 * 10^{-3}$.

4.5 Orthogonal sampling

We have ran the simulation for orthogonal sampling multiple times. We continue to keep the number of iterations steady on 1000, but we multiply the number of samples each time with 10 (and thus the number of squares in the grid with 10^2). We calculate in this way the area of the Mandelbrot a thousand times and calculate the mean and standard deviation of our simulations. The given results are shown in figure 8.

We can clearly see the effects of increasing the number of samples, just like pure random sampling. The real given area of the Mandelbrot set is plotted with the vertical black line. Every time this value is in in the interval of $[\mu - \sigma, \mu + \sigma]$, but we more importantly see that the standard deviation decreases as the number of samples increasing.

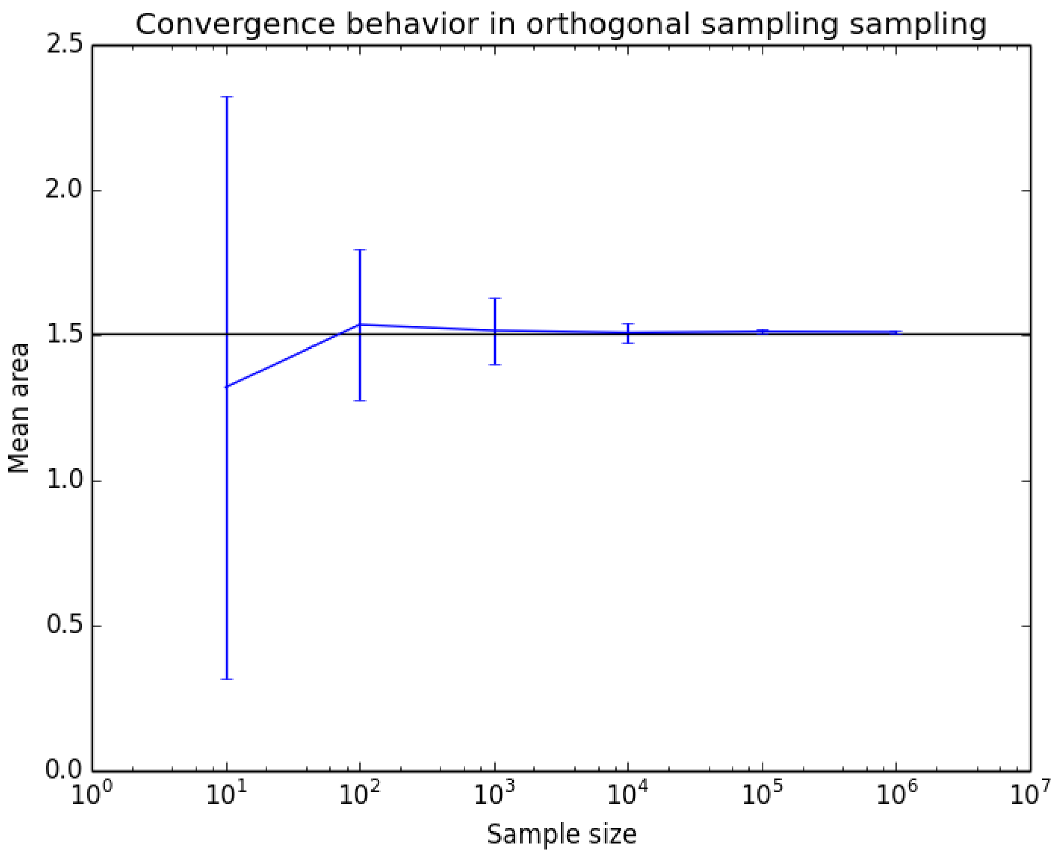

Figure 8: The average of the estimated Mandelbrot area does not change with the increase of samples (since this is unbiased), but the variance clearly decreases. It converges to the real value, represented with the black line. With 10^6 samples the standard deviation is a mere $4.09 * 10^3$.

4.6 Importance sampling

We have ran the simulation for Importance sampling multiple times. We continue to keep the number of iterations steady on 1000, but we multiply the number of samples each time with 10 (and thus the number of squares in the grid with 10^2). We calculate in this way the area of the Mandelbrot a thousand times and calculate the mean and standard deviation of our simulations. The given results are shown in figure 9.

We can clearly see the effects of increasing the number of samples, just like pure random sampling. The real given area of the Mandelbrot set is plotted with the vertical black line. Every time this value is in in the interval of $[\mu - \sigma, \mu + \sigma]$, but we more importantly see that the standard deviation decreases as the number of samples increasing.

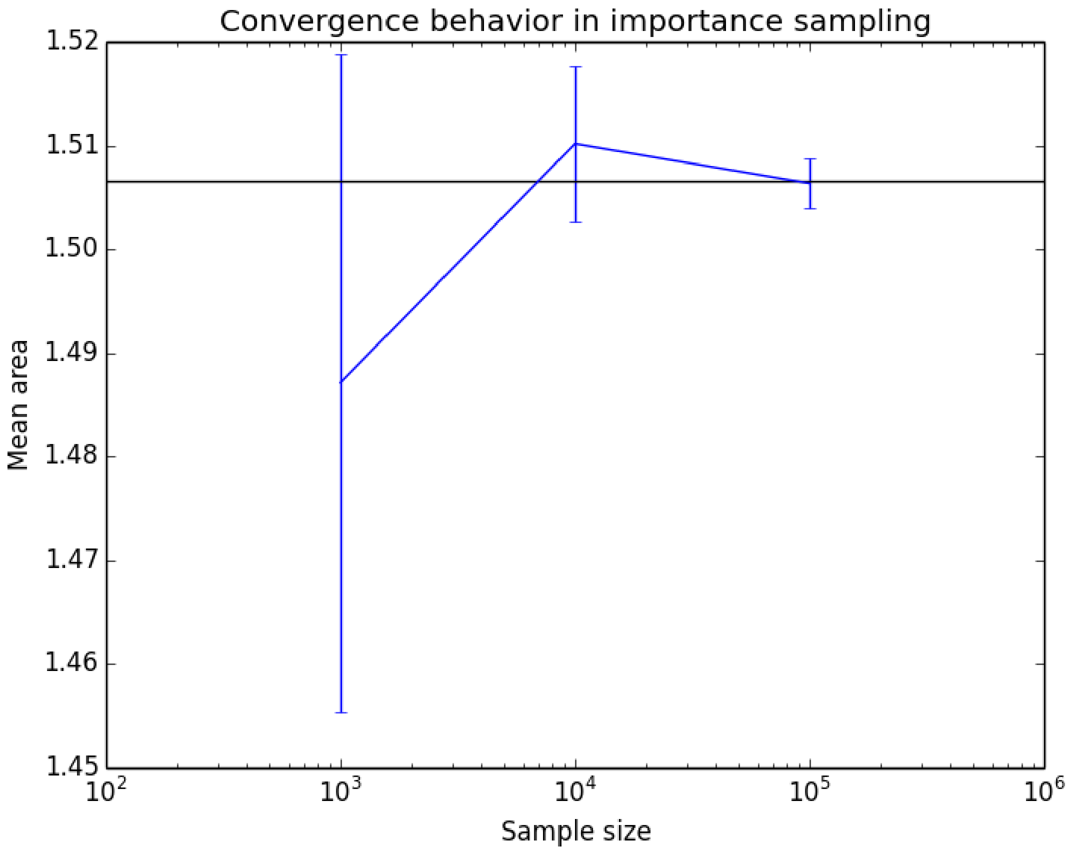

Figure 9: The average of the estimated Mandelbrot area does not change with the increase of samples (since this is unbiased), but the variance clearly decreases.It converges to the real value, represented with the black line. With 10^6 samples the standard deviation is a mere $5.35 * 10^{-4}$.

4.7 Comparing results

We have seen in previous parts that with every sampling method the average estimate Mandelbrot area converges with the increase of samples. However, we can compare their variances and thus conclude which one converges the fastest. The easiest version, pure random sampling, has also the highest variance given a certain number iterations and samples. The lowest variance is found when importance sampling is used (see table 1). We note that the standard deviation is of course also stochastic, but the difference is quite large, so we didn't investigate this property further.

Sampling type	Mean (10^5 samples)	Std (10^5 samples)	Std (10^6 samples)
Pure random sampling	1,508404	0,017797	0,004290
Latin hypercube sampling	1,510724	0,011574	0.003291
Ortohonormal sampling	1,5105792	0.006727	0.003651
Importance sampling	1,506353	0.000535	-

Table 1: The results of the simulations, as shown in previous figures. We can compare the standard deviation (std) for the different sampling methods and see that even if we have 10 times less samples, importance sampling is more accurate.

5 Discussion

We would like to light out certain aspects of our paper. Firstly we'd like to discuss some problems we found during this research.

The estimation of the area of the Mandelbrot set converges to the true area of the Mandelbrot set as the amount of samples increases but does not converge as the amount of iterations increases above 50. The amount of iterations does not influence the amount accuracy or precision of the area estimation within the range investigated. All sampling methods show increasing convergence with increasing sample sizes. Importance sampling seems to achieve obtain the best results when computational costs must be kept low. We consider this unfortunate, however we think it was necessary to compare the different sampling methods.More advanced Mandelbrot area analysis techniques should focus on preventing wasting computation time on iterating within the Mandelbrot set as the program will go through all iterations.

The amount of samples was indeed negatively correlated to the variation in the surface area estimation. However only for negative correlation between the amount of iterations and samples categorized as Mandelbrot numbers.

The true area of the Mandelbrot set as suggested by Kerry Mitchell [5] was always within the standard deviation of our estimation, suggesting our estimation methods were never inaccurate. A potential weakness of Monte Carlo integration is the inefficiency of sampling areas well within or well outside of the figure. Intensifying sampling at the border or the Mandelbrot set proved to improve the cost to accuracy and cost to precision ratio.

We defined samples to be out of the Mandelbrot range when the absolute or imaginary part became greater than 2 while Wolfram suggests it should be radius 2. While this doesn't affect our results, it was done because it was easier to check during our iterations and thus computationally faster.

We also have had passed the change for a couple of other improvements.
It was possible to speed up the process. For example is the Mandelbrot set completely symmetrically in the Real axis. It would have been possible to only study the non-negative imaginary part and multiply that estimated area by two. However this improvement will nearly matter when the computation time gets higher.
Secondly we now chose a square domain for all points. However the smaller this domain would have been, how better our results. Not only could we have picked a rectangle domain by removing a strip on the right side of the square. But we could also have created a circle around the Mandelbrot with a radius of 2.
Eventually, another often referred possibility was the use of antithetic values. If we can create an artificial negative correlation between the sample point, we lower the variance of the estimate area. However we ran out of time during this research to properly implement this.

References

[1] Robert Brooks and Peter Matelski, The dynamics of 2-generator subgroups of PSL(2,C), in Irwin Kra (1 May 1981). Irwin Kra, ed. Riemann Surfaces and Related Topics: Proceedings of the 1978 Stony Brook Conference. Bernard Maskit. Princeton University Press. pp. 65–71. ISBN 0-691-08267-7.

[2] de l'Acad. Roy. des. Sciences (1733), 43–45; naturelle, générale et particulière Supplément 4 (1777), p. 46

[3] Matsumoto, M., Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1), 3-30.

[4] Sheldon M. Ross, Simulation, Fourth Edition, Academic Press, Inc., Orlando, FL, 2006

[5] A Statistical Investigation of the Area of the Mandelbrot Set, Kerry Mitchell, 2011, http://www.kerrymitchellart.com/articles/area/mandelbrot-area.html

[6] Mandelbrot Set at http://mathworld.wolfram.com/MandelbrotSet.html, consulted on 16 november 2018