Computing the Area of the Mandelbrot Set

Stochastic Simulation – Assignment 1

Daan van Ingen (10345078)
Walter Vianen (11811293)

1 Introduction

Numerical methods are an alternative to exact mathematical analysis for problems where exact descriptions and solutions are not readily available or even impossible to derive. One type of computational algorithms are Monte Carlo (MC) methods. MC methods rely on repeated random sampling to obtain numerical results. In particular, MC integration is a numerical integration technique using random numbers.

In this report the area of the Mandelbrot set is estimated using a hit-and-miss algorithm, which is a type of MC integration technique. The Mandelbrot set is a set of complex numbers of which the boundary shows increasingly finer recursive details when zooming in. The boundary of the set also incorporates smaller versions of the main shape. Hence, this fractal has the property of self-similarity. The exact value of the area of the Mandelbrot set is unknown. Hence, numerical integration techniques are necessary to find approximations of this value.

The outline of the report is as follows. In section 2 a formal definition of the Mandelbrot set is given including a description of its implementation. In the same section the details on the hit-and-miss algorithm are provided and evaluation metrics are defined. Section 3 contains the results of several estimations of the area of the Mandelbrot set along with discussion. Finally, in section 4 key remarks are summarized and conclusions are drawn.

2 Definitions & Methods

2.1 The Mandelbrot Set

The Mandelbrot set is the set of complex numbers c for which the function $f_c(z) = z^2 + c$ does not diverge when iterated from $z = 0$ [1]. We define the Mandelbrot sequence as the numbers resulting from:

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

and noting $z_0 = 0$. Images of the Mandelbrot set can be created by applying a color gradient to the iterations n_i at which the Mandelbrot sequence of a complex number c_i first exceeds a predetermined threshold. Examples of these images are shown in figure 1. The recursive details and self-similarity become increasingly clear when zooming in on the boundary of the set.

2.1.1 Area

An exact expression for the area of the Mandelbrot set A_M exists [2] but contains a sum that converges very slowly. Investigation of this provided the upper limit $A_M \leq 1.7274$ and led the authors to believe that the true value lies between 1.66 and 1.71.

The area of the Mandelbrot set obtained by pixel counting is $1.50659177 \pm 0.00000008$ with a grid size of 819200×409600 (≈ 336 billion) pixels [3]. The estimation is significantly smaller than the estimate from the investigation of the exact expression. This establishes a point of reference for pure computational methods.

2.1.2 Implementation

For the implementation of the Mandelbrot set only the first N iterations are considered for each complex number c (recall the definition of the Mandelbrot sequence in equation 1). At each iteration it is checked whether the value of $|z_n|$ exceeds a predetermined threshold τ. If $|z_n|$ remains bound for all N iterations, c is considered to be part of the Mandelbrot set (zero is returned). Pseudocode for this method is provided in algorithm 1.

Algorithm 1 Pseudocode for testing whether a complex number c belongs to the Mandelbrot set.

```
while n < N do
    if |z| > τ then
        return n
    z = z^2 + c
    n += 1
return 0
```

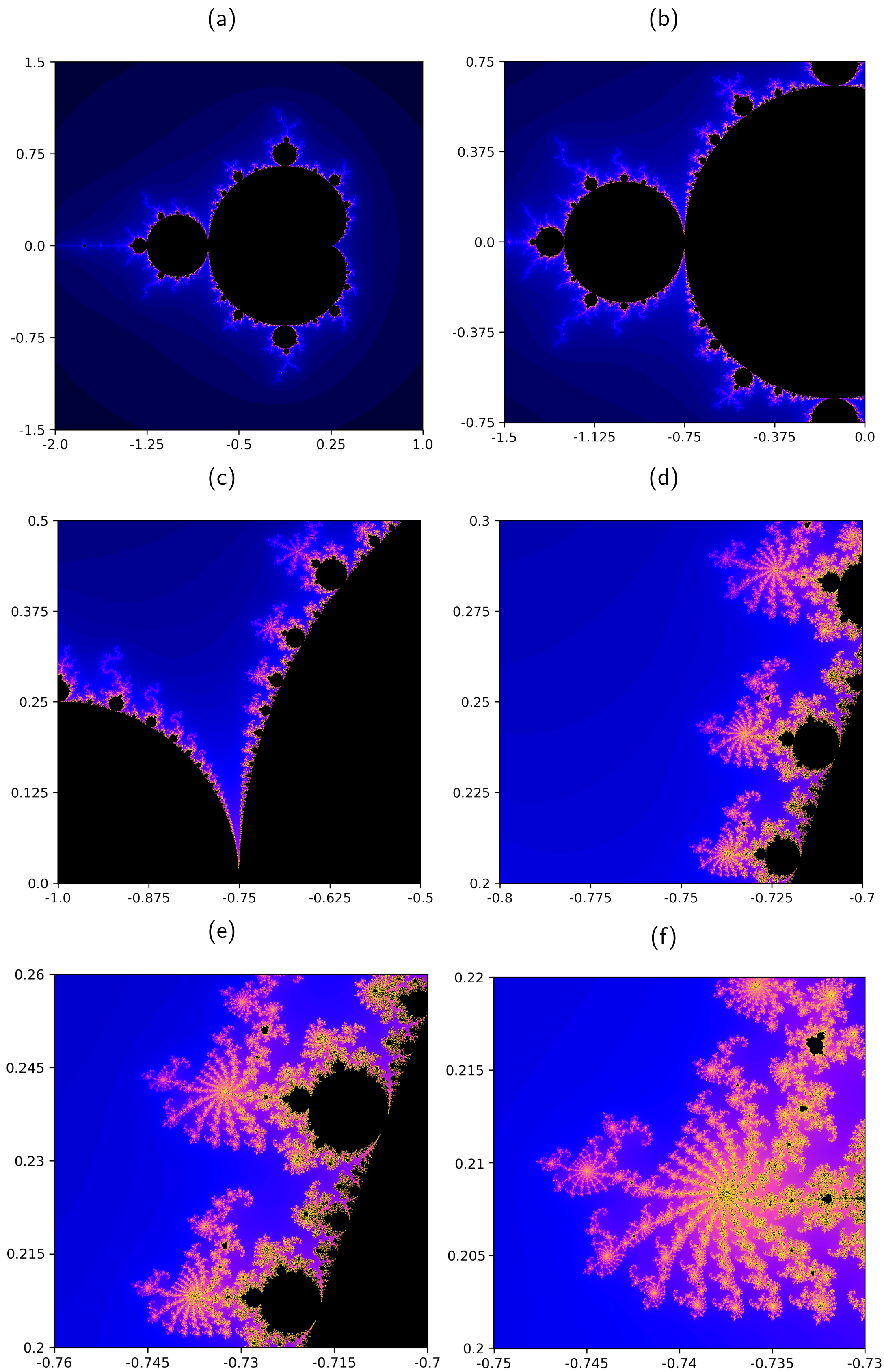

Fig. 1: Visual representation of the Mandelbrot set (black) for different zoom. The color gradient (blue to pink – fast to slow) indicates how quickly the Mandelbrot sequence diverges.

2.2 Hit-and-Miss Method

Hit-and-miss is a Monte Carlo method that relies on random number generation to estimate the area of a shape. A set of random input parameters are generated in a predefined domain, the random numbers are then used in a deterministic computation and the results are assembled for the final answer. The area of the shape is proportional to the fraction of random samples that are inside the shape. A classic example would be the estimation of π by throwing darts at a 2×2 square and find the fraction of darts that land inside a circle with radius 1.

Monte Carlo methods are particularly useful in high dimensional problems where numerical integration might not be feasible due to a combinatorial explosion in function evaluations. For simplicity we will only look at the 2D problem of the Mandelbrot set. The next sections will describe the implementation and evaluation of the hit-and-miss method for the Mandelbrot set and look at sampling methods.

2.2.1 Implementation

The hit-and-miss algorithm is conceptually simple. Random numbers, S in total, are sampled from a predefined interval for the real and imaginary axes to construct complex numbers c. These complex numbers are then tested as to whether or not they belong to the Mandelbrot set. Note that it might be preferred to immediately evaluate c after having sampled it due to memory constraints. The proportion of shots that belong to the set (hits) to those that do not (misses) multiplied by the area spanned by the intervals is an estimate of the area of the Mandelbrot set. Pseudocode of this algorithm is provided in algorithm 2.

In the remainder of this report the intervals used are $[-2, 1]$ for the real axis and $[-1.5, 1.5]$ for the imaginary axis resulting in a square search space with total area of 9. Moreover, the Mandelbrot sequence for a complex number is said to diverge if at some point $|z_n| > 3$ (so $\tau = 3$).

Algorithm 2 Pseudocode for the hit-and-miss algorithm.

```
for all shots do
    Sample random c
    Determine whether c belongs to the Mandelbrot set
    if it belongs to the set then
        Hits += 1
Estimated area = (Hits / S) * 9
```

2.2.2 Sampling Methods

To sample the complex numbers to be evaluated three different methods are distinguished and compared.

1. **Pure random sampling** Both $Re\{c\}$ and $Im\{c\}$ are sampled uniformly from their respective intervals for each c.

2. **Latin hypercube sampling (LHS)** A grid is defined over the search space. Points are sampled such that only one point can occupy a row and column. When sampling S points in total, each dimension is divided into S equally probable intervals. Samples can be taken one at a time. However, this method requires that the position of all previous sample points are remembered in order to satisfy the constraints.

3. **Orthogonal sampling (OS)** This method is an extension of the LHS method. The space is divided into equally probable subspaces comprising a collection of grid points. All points are then sampled simultaneously such that the LHS constraints are satisfied and each subspace is sampled with the same density.

The goal of LHS and OS is to reduce the variance of the outcome by ensuring that the ensemble of random numbers is representative of the real variability whereas traditional random sampling is just a collection of random numbers without any guarantees. When the number of samples drawn becomes large, the variance of traditional random sampling will converge to its true value and therefore the effectiveness of LHS and OS will decrease. But LHS and OS can still be used to decrease the number of samples needed to reach some accuracy, which can be useful when sampling is an expensive process (for instance when sampling is done through some deterministic model).

2.3 Evaluation

In short, the task is to investigate the area A_M of the Mandelbrot set by MC methods. An estimate of this area $A_{N,S}$ depends on the maximum number of iterations computed in the Mandelbrot sequence N and the number of shots taken S.

MC integration methods are inherently stochastic. Hence, confidence intervals are computed for the results where applicable. After N_r number of independent runs, $A_{N,S}$ is expected to be normally distributed as is implied by the central limit theorem. This distribution has unknown mean and variance. The mean result $\bar{A}_{N,S}$ is simply the sample mean and the standard deviation s is the corrected sample standard deviation:

$$s = \sqrt{\frac{1}{N_r - 1}\sum_{i=1}^{N_r}(x_i - \bar{A}_{N,S})^2}. \tag{2}$$

where x_i, of course, refers to a $A_{N,S}$ resulting from one specific independent run. Consequently, the confidence interval is:

$$\left[\bar{A}_{N,S} - t^* \frac{s}{\sqrt{N_r}}, \bar{A}_{N,S} + t^* \frac{s}{\sqrt{N_r}}\right] \tag{3}$$

where t^* is the critical value derived from the Student's t distribution. For a two-sided 95% confidence interval and $N_r = 100$ (where the number of degrees of freedom $\nu = N_r - 1$) this results in $t^* = 1.9842$.

To quantify the convergence of the algorithm the size of the 95% confidence interval ϕ is taken as a measure:

$$\phi = 2t^* \frac{s}{\sqrt{N_r}}. \tag{4}$$

3 Results & Discussion

3.1 Error Estimation

According to the central limit theorem it is expected that $A_{N,S}$ are normally distributed over the N_r independent runs. This is confirmed visually in figure 2. Similar results are obtained for all sampling methods. It is noteworthy that the orthogonal sampling method results in a narrower distribution than the LHS, which is again narrower than the distribution for random sampling. The narrower the distribution, the better the convergence of the algorithm. Moreover, the normality of the distributions is also confirmed when performing Shapiro-Wilk normality tests on the obtained data.

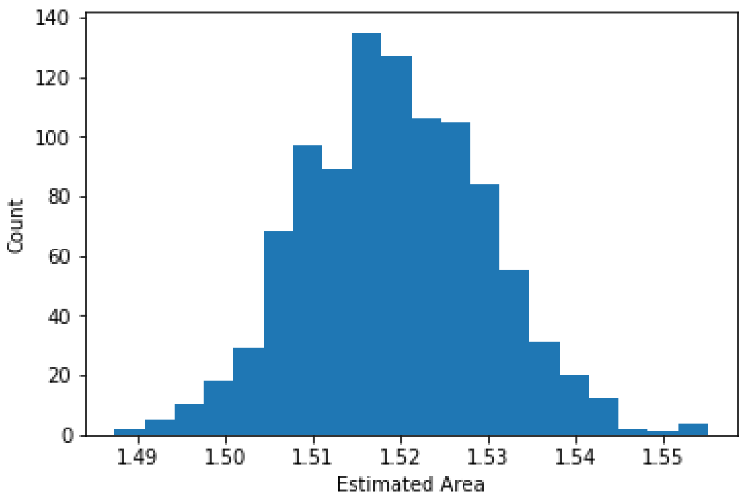

Fig. 2: Histogram of $A_{N,S}$, with $N = 300$, $S = 10^5$ and $N_r = 1000$. The result clearly approximates a normal distribution.

To compare the error caused by the finiteness of the maximum number of iterations in the Mandelbrot sequence N and the number of shots taken S we study $(\bar{A}_{j,S} - \bar{A}_{i,S})$ as a function of j where i is kept constant at $i = 10^6$. The results using the random sampling method are shown in figure 3.[1] If there were to be a difference in the errors caused by the finiteness of N and S then there should be a clear and consistent difference between the performance for different N. This is, however, concluding from figure 3, not the case for the considered

[1] Both $\bar{A}_{j,S}$ and $\bar{A}_{i,S}$ have their own confidence interval. Combining this gives the new convergence measure: $\phi = 2t^*/\sqrt{N_r}(s_{j,S} + s_{i,S})$.

parameter intervals; varying the maximum number of iterations does not seem to have a significant effect on the error $(\bar{A}_{j,S} - \bar{A}_{i,S})$ and its convergence.

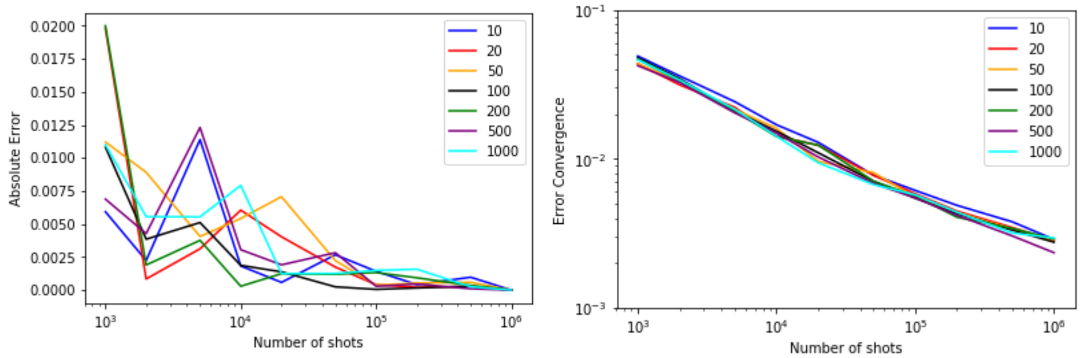

Fig. 3: $(\bar{A}_{j,S} - \bar{A}_{i,S})$ (left) and corresponding convergence (right) as a function of the number of shots. Different colors indicate different values for the maximum number of iterations in the Mandelbrot sequence N. Note that $N_r = 100$.

3.2 Sampling Methods Comparison

Three sampling methods are compared: random sampling, latin hypercube sampling, and orthogonal sampling. Figure 4 shows $\bar{A}_{N,S}$ and its convergence as a function of the number of shots taken S for all sampling methods. All methods converge to a value of $\bar{A}_{300,10^6} \approx 1.519$. This is somewhat higher than the value obtained using pixel counting [3], but the similarity provides confidence in the method. It is, however, still quite far from the optimal area provided by investigation of the exact expression for A_M [2].

From the figure it is concluded that the orthogonal sampling method is more stable in its approximation than the other methods. The convergence is also best for the orthogonal sampling method, followed by the latin hypercube sampling and random sampling respectively.

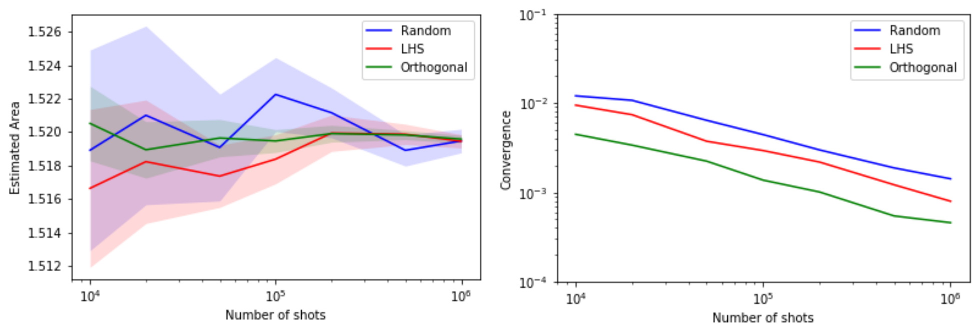

Fig. 4: $\bar{A}_{N,S}$ for all sampling methods (left) with corresponding 95% confidence intervals (shaded areas), and corresponding convergence (right) as a function of the number of shots taken S, and $N = 300$ and $N_r = 100$. Additionally, the number of subcells is 400 for the orthogonal sampling.

3.3 Optimization Orthogonal Sampling

Firstly, the optimal number of subcells is determined. The results for this are shown in figure 5. From the figure it can be concluded that, in general, increasing the number of subcells improves convergence of the algorithm. This, of course, only holds for the considered interval. Higher values for the number of subcells have not been investigated, because creating the sampling slows down for a higher number of subcells. The current optimal amount of 400 subcells is still considered to be computationally feasible.

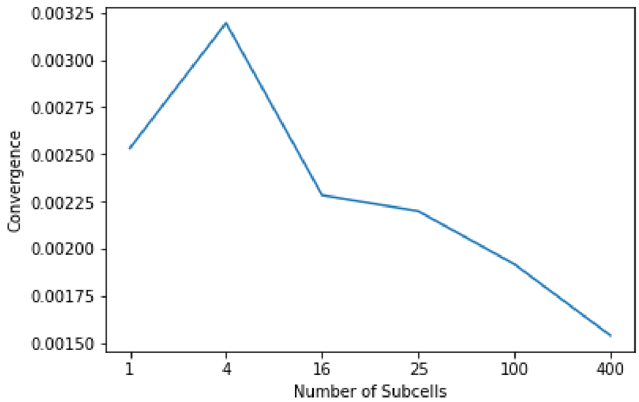

Fig. 5: Convergence for the orthogonal sampling method as a function of the number of subcells. Parameter settings are: $N = 300$, $S = 10^5$, and $N_r = 100$.

Next we describe a method to improve overall convergence for the orthogonal sampling method. The original method is to include all subcells in the hit-and-miss algorithm. However, for some quite large regions, and thus entire subcells, the results are the same: either the subcell entirely belongs to the Mandelbrot set, or the Mandelbrot sequence quickly diverges for all the points within the subcell. Knowing this beforehand it is unnecessary to include these cells in the hit-and-miss algorithm and less shots are needed to achieve similar convergence to the traditional method. Figure 6 shows which subcells can be excluded, because the results are known beforehand.

To be more precise, figure 6 was created by calculating divergence of all points in the usual square area with a limited resolution. When all the points within a subcell belong to the Mandelbrot set or when all of them diverged within two iterations, this was noted. The remaining subcells are the focus of the hit-and-miss algorithm. Out of the 400 subcells, 44 belong to the Mandelbrot set and 173 converge within two iterations. Thus, 217 subcells need not be sampled. It is expected that the improved sampling method can achieve the same convergence as the traditional method with using less than half the amount of shots taken. This is confirmed in figure 7.

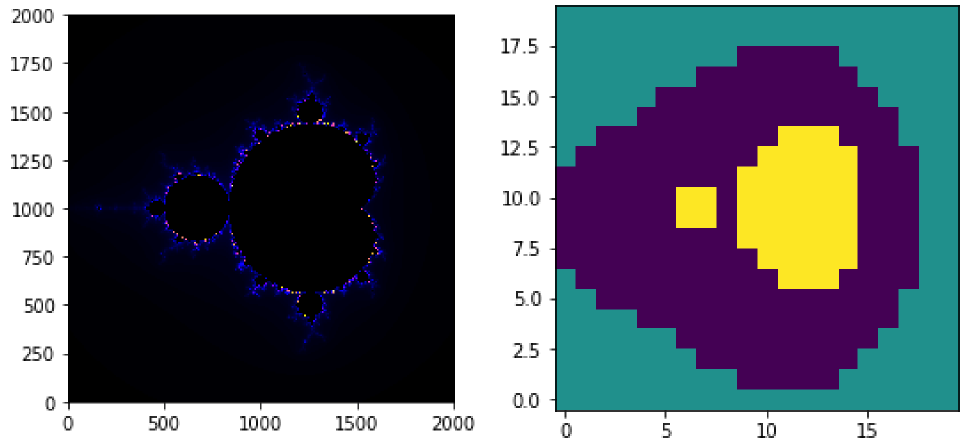

Fig. 6: The Mandelbrot set computed with limited resolution (left) and corresponding subcell division for the orthogonal sampling (right). Entire subcells belonging to the set are marked yellow, quickly diverging subcells are green and the remaining cells are purple.

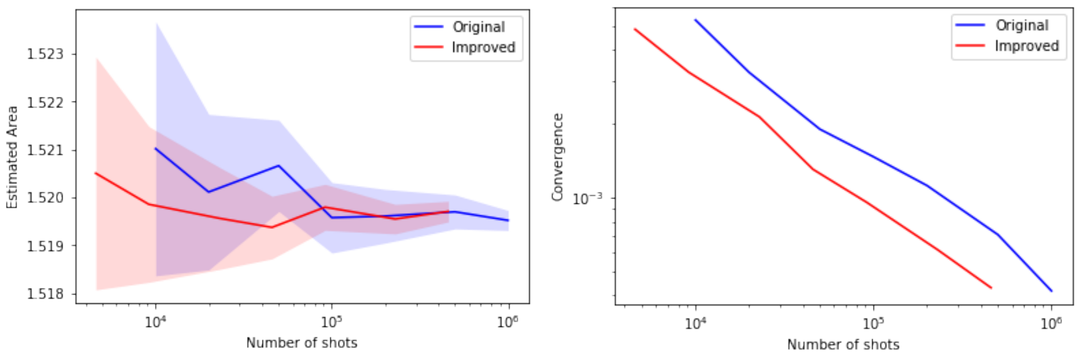

Fig. 7: $\bar{A}_{N,S}$ for the original and improved orthogonal sampling (left) and corresponding convergence (right) as a function of the number of shots taken S. Parameter settings are: $N = 300$ and $N_r = 100$.

4 Conclusion

The hit-and-miss Monte Carlo method was used to estimate the area of the Mandelbrot set. Different sampling methods were used in the form of pure random sampling, latin hypercube sampling and orthogonal sampling. The performed experiments on the Mandelbrot set yield results that are close to the estimate of 1.50659177 obtained by pixel counting [3]. While the average estimated area of the performed experiments is generally a bit higher (figure 2), we can say that it performs well considering the difference in precision.

The error of the estimate becomes smaller as the number of shots is increased, the number of iterations in the Mandelbrot set seems to have a minor impact on the convergence of the error. The different sampling methods had their impact on the variance of the outcome, with orthogonal sampling being the best. Orthogonal sampling was further improved by focusing computation on certain areas of the Mandelbrot set. This proved effective as the improved orthogonal sampling gives the same results as the traditional orthogonal sampling with less than half the amount of shots taken.

References

[1] Benoit B Mandelbrot. Fractal aspects of the iteration of z λz (1-z) for complex λ and z. *Annals of the New York Academy of Sciences*, 357(1):249–259, 1980.

[2] John H Ewing and Glenn Schober. The area of the mandelbrot set. *Numerische Mathematik*, 61(1):59–72, 1992.

[3] Robert P Munafo. Pixel counting. 2012. http://www.mrob.com/pub/muency/pixelcounting.html.