Universiteit van Amsterdam

Stochastic Simulation

Assignment 1: Computing the area of the Mandelbrot set.

Caterina Buranelli (13169408)[a]
Ignas Krikštaponis (13250868)[b]

[a]caterina.buranelli@student.uva.nl
[b]ignas.krikstaponis@student.uva.nl

17 Nov. 2020

Abstract

The exact value of the area of the Mandelbrot set can be calculated only with an infinite number of iterations on an infinitive number of samples. As a precise approximation has not been found yet, the area of the fractal is still widely studied among mathematicians. Monte Carlo integration was utilised in this study to approximate the area in question. In this paper the optimal value for the number of iterations and for the number of samples was investigated, as well as the best sampling method. Orthogonal sampling was found to be the best method when considering the standard deviation, but it had drawbacks in terms of computational effort. Randomised Quasi-Monte Carlo was the sampling method with the best trade-off between standard deviation - slightly bigger than in orthogonal - and computational time - more than 100 times smaller than orthogonal.

1 Introduction

Since the beginning of the XX century, lots of mathematicians struggled in representing the Mandelbrot set, a fractal which is named after Benoît Mandelbrot, one of the first who managed to visualise it in 1980 (Taylor and Sprott [2008]). The Mandelbrot set is defined as the set of complex numbers c for which the function

$$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$$

remains bounded. (Kilmister [1978])

Even if now, thanks to modern computers, it is easy to plot the Mandelbrot set, the exact area of it is still under research. An infinite number of iteration on an infinite number of points is required in order to determine the area, as understandable from the definition of the function. A plot of the fractal can be found in Figure 1.

The aim of this paper is to investigate the actual value of the area of the fractal via Monte Carlo integration, by finding the best parameters (number of iterations and number of samples) and the best sampling method. Standard deviation (SD) calculated over several simulations will be the main criterion used to compare the results. In addition to this, the average time needed for one simulation - considered as a measure of the computing effort - will be taken into account. Furthermore, the effect of the number of simulations will also be investigated.

The research questions of this paper are as follows: *In order to calculate the area of the Mandelbrot set, what is the best value for the i and s, taking into consideration*

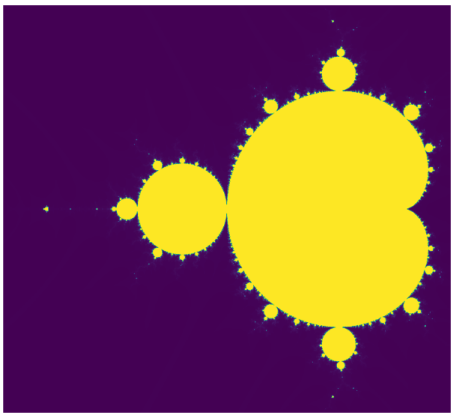

Figure 1: The Mandelbrot set.

the limited computing power at hand and the number of simulations? and *In order to calculate the area of the Mandelbrot set, what is the best sampling method?*.

2 Methods

It is possible to demonstrate that for $|z_n| > 2$ the function diverges, therefore that point will not belong to the set. Besides that, the fractal is bounded by a circle of radius 2, therefore the area is smaller than 4π. Moreover, the fractal can be inscribed in a rectangular: its top and bottom expand from $y = -1.12$ to $y = 1.12$, its right and left sides expand from $x = -2$ to $x = 0.47$. (Mitchell [2001])

The Mandelbrot set can be visualised by setting an iteration value i and dividing the complex plane bounds into evenly spaced samples and calculating the number of iterations reached before the sample was declared to be outside the set. Each point is then assigned a colour corresponding to this value as in Figure 1.

Monte Carlo integration This paper utilises Monte Carlo integration to stochastically estimate the Mandelbrot set's area. First, a set number of samples on a complex plane, bounded as described above, is generated. Each value is then evaluated over a set number of iterations to determine whether it belongs to the Mandelbrot set. The ratio between the number of samples that belongs to the Mandelbrot set and all samples taken multiplied by the search area yields an estimation of the Mandelbrot set's area.

In the paper six sampling techniques were used and compared. Firstly, three

standard samplings were implemented: Random, Latin hypercube and Orthogonal sampling. Secondly, the best of these methods was used as benchmark to test three *improving* methods: Antithetic variables, Poisson disk and Randomised Quasi-Monte Carlo.

Random sampling The real part, x, and the complex part, y, of the complex number z, are randomly drawn respectively from a $U(-2.2, 0.55)$ and $U(-1.2, 1.2)$. The bounds for the Uniform distributions were found through the information said above, plus, the values were rounded up to avoid losing significant samples. This is the method performed during the experiments for estimating the optimal values of s and i.

Latin hypercube sampling The Latin hypercube sampling (Ross [2012]) is a type of stratified sampling where the sample space is divided in the following way:

1. Set s as the number of samples needed
2. Divide the sample space in a grid made of $s \times s$ equal areas
3. Draw one sample from each area; there must be only one sample for row and column

This method ensures good representation of the actual area, but it introduces negative correlation between the points.

Orthogonal sampling The Orthogonal sampling (Owen [1992], Leary et al. [2003]) is a variation of the Latin hypercube sampling and it is built to minimize the correlation in the samples. The only difference is that before drawing the sample, the sample space is divided in equally probable sub-regions, that will contain the same amount of samples. At the end of the sampling, both the sub-regions and the total sample will have the properties of a Latin hypercube. The implementation of orthogonal sampling requires that $\sqrt{s}$ is an integer.

Orthogonal sampling requests high computational effort, and therefore time, to be performed.

Antithetic variables To try to decrease the variance of the results, antithetic variables were used for the sampling. For the real part of z, $s/2$ samples were drawn from a random variable $X' \sim U(-2.2, 0.55)$; the rest $s/2$ of the sample was given by a second random variable defined as $X'' = X' - 2(X' - \mu_{X'})$. By doing this, the two

variables have same distribution and negative covariance, and the sampling needs half of the computational effort as regular sampling. The same thinking was applied to find the complex part of z, with $Y' \sim U(-1.2, 1.2)$.

Poisson disk sampling Poisson disk distribution allows to randomly generate samples that are at least at a distance r between each other. The algorithm from Bridson [2007] was used for this purpose. Once the bounds of the sample are set, the algorithm takes the first random sample x_0, then it chooses the next point from a set of points uniformly generated from the spherical annulus between radius r and $2r$ of x_0.

It is clear that with this method it is not possible to set the total amount of samples, which is always different and depends both on the extent of the sample and on the size of r. An exploratory analysis was carried out in order to decide the value of r, in order to have a number of samples around the value needed.

Randomised Quasi-Monte Carlo Quasi-Monte Carlo (Niederreiter [1992]) differs from the regular Monte Carlo method as it does not use pseudorandom numbers but it works with low-discrepancy sequences instead. This means that it is deterministic, but that the sequence has a behavior similar to the one of an equidistributed sequence: the proportion of points falling into the Mandelbrot area is nearly proportional to the actual area, an useful property for an integration problem. The whole process can be randomised by adding some noise to the sequence. In this paper the Hammersley Sequence was used.

3 Results

3.1 Finding the optimal number of iterations

In order to find the optimal number of iteration i, the area $A_{i,s}$ of the Mandelbrot set was calculated for a fixed sample $s = 10\ 000$ and varying i, which took values from 50 to 10 000 in 500 steps; random sampling was used to select the points. The last area, $A_{\max(i),s}$, was taken as the baseline and the error was given by the difference between the 500 areas and the baseline. 50 simulations were performed, as the value of the area would slightly oscillate due to stochasticity.

The average error over the 50 simulations is plotted in Figure 2. Given a threshold of 0.001, the corresponding value of i was found to be 2582. The threshold was chosen by taking in consideration that an error of 0.001 is significantly equal to 0, as the 95% confidence interval was $(0.008, -0.006)$.

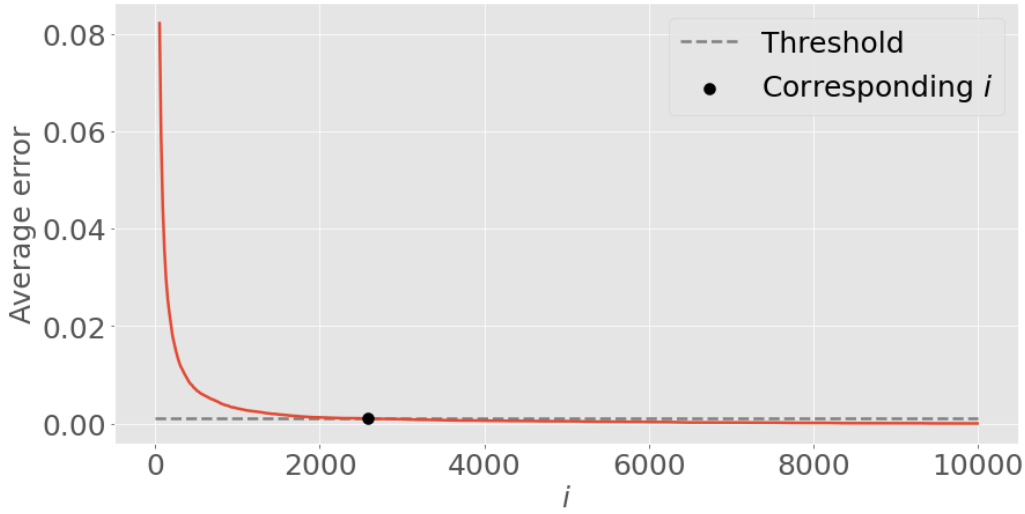

Figure 2: Average error against iterations for 50 simulations; varying i and $s = 10\ 000$.

3.2 Effect of number of samples on the simulation results

The next step was to find the optimal number of samples for the given number of iterations. The area of the set was computed for a value of s that goes from 50 to 10 000, in 500 steps; 50 simulations were performed. The criterion used was to choose s when the standard deviation appeared to be equal to 0.05; this way, the precision of the estimation of the mean area would be $\pm 1.96 \frac{SD}{\sqrt{\text{simulation}}} = \pm 1.96 \frac{0.05}{\sqrt{50}} = \pm 0.01$. This was observed also for Latin hypercube and orthogonal sampling, even though the random sampling was the method used for establishing the optimal value of s.

In Figure 3 the standard deviation for the three types of sampling is plotted. The data were fitted with an exponential curve and the threshold was calculated on the fitted values. For orthogonal sampling, s took values only from 50 to 3000 in 48 steps, due to the more computational effort needed; moreover, the fitted curve shows that the standard deviation does not change much for bigger values of s.

The optimal value of s was chosen looking at the random sampling and was found as $s = 3021$. Because orthogonal sampling needs a number of samples which is an exact square, in order to better compare the methods later in the paper, $s = 3025 = 55^2$ was chosen to be the optimal number of samples.

Beside that, the graphs are interesting in order to compare the three sampling methods. It is clear that Orthogonal is the one with the lowest variability, and to achieve a standard deviation of 0.05 only 400 samples are needed. Latin hypercube shows better results than Random sampling, and the threshold is reached when

Table 1: Summary of simulation results.

Sampling	$\bar{A}$	SD	CI-95%	$\bar{\text{time}}$(s)
Random	1.4999	0.0501	(1.4901, 1.5097)	0.2922
Latin hypercube	1.5076	0.0367	(1.5004, 1.5148)	0.2896
Orthogonal	1.5084	0.0110	(1.5062, 1.5106)	37.0888
Random antithetic	1.5069	0.0499	(1.4971, 1.5167)	0.2893
Poisson disk	1.5100	0.0192	(1.5062, 1.5138)	0.3894
Hammersley	1.5109	0.0147	(1.5080, 1.5138)	0.3146

$s = 1326$.

In conclusion, orthogonal sampling is the best method, as for the same number of samples (3025), it has the lowest variance and therefore the highest precision; for this reason it will be taken as benchmark in the next experiments.

3.3 Investigating the number of simulations

Once the optimal values for i and s are found, it is interesting to investigate how many simulations are needed in order to achieve good precision of the results. 100 simulations were performed for the three methods above and for the improving methods that will be discussed in the next subsection. The radius of the 95% confidence interval (CI) of the average area is considered to be the indicator of precision. In Figure 4 the progression of CI-95% with increasing simulation number is plotted. As we can see, coherently with the previous results, the orthogonal sampling is the one with the narrowest interval radius, and around 30 simulations would give enough precision as the value of CI seems to have converged. Random and Latin hypercube are quite distant from the values of Orthogonal: for an error of $\pm 1\%$, Latin hypercube would need around 60 simulations and Random around 100, while Orthogonal only around 10.

3.4 Improving the results

Three methods were implemented in order to investigate if the results demonstrated in Section 3.3 were improvable.

Applied to random sampling, antithetic variables did not give better results, as the standard deviation remained at around 0.05, as set from the threshold. This can be seen from Table 1 and from Figure 5, where the radius of the respective confidence

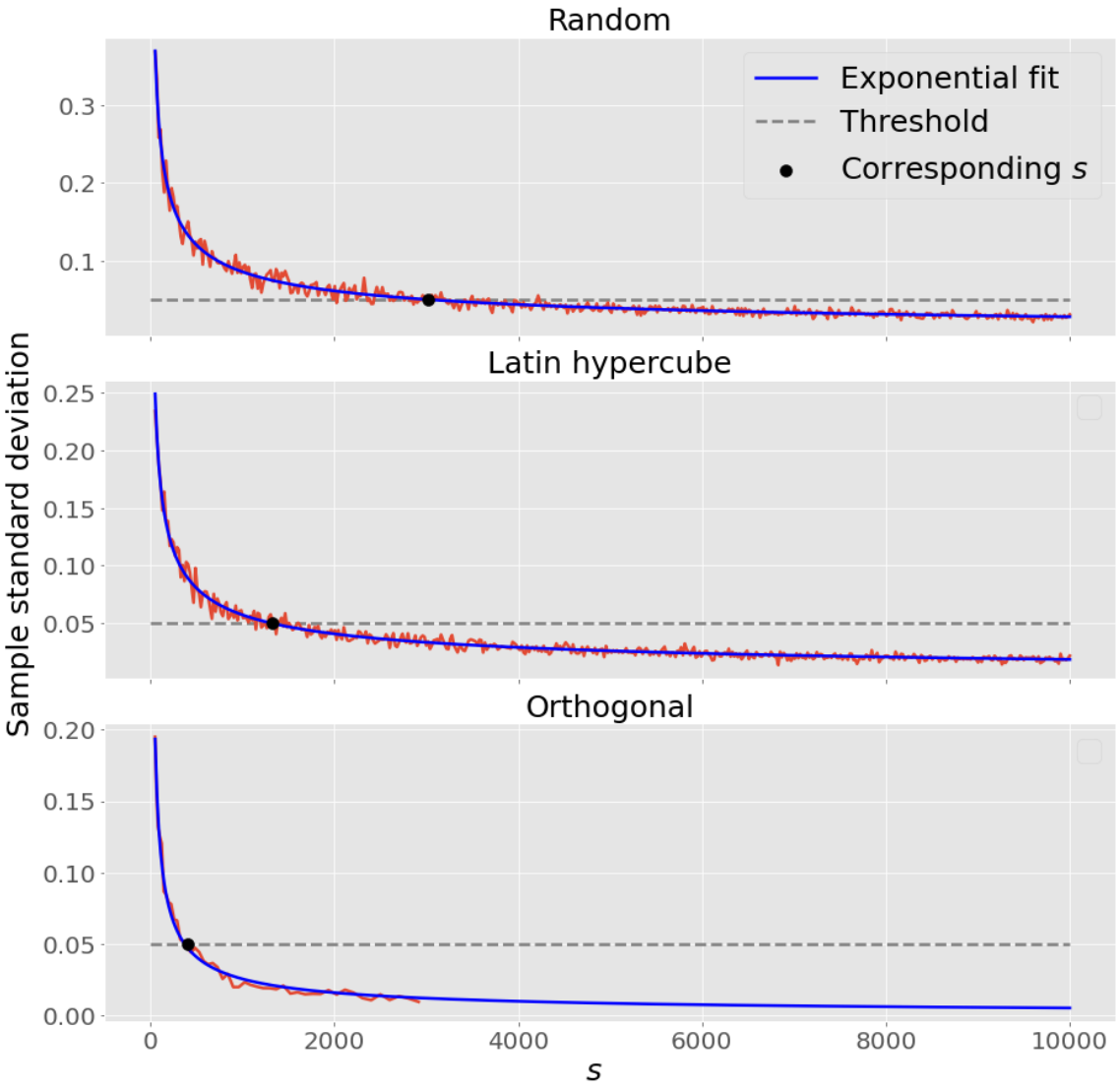

Figure 3: Sample standard deviation for 50 simulations

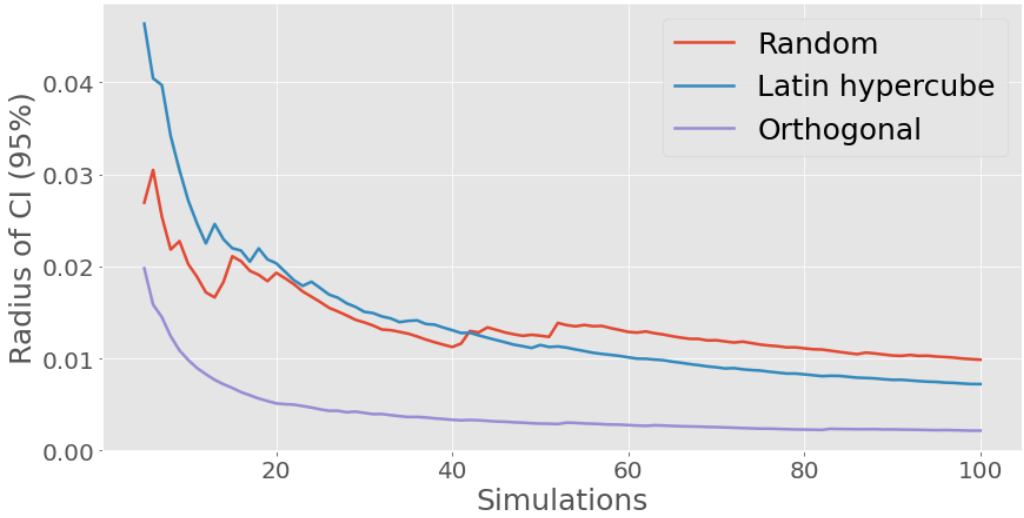

Figure 4: Radius of 95% confidence interval of the average area for the standard methods.

interval shows a behaviour similar (from 20 simulations on) to the the Random regular sampling; therefore antithetic variables do not lead to a faster convergence.

For Poisson disk sampling, the parameter r was set equal to 0.0331 as the number of samples generated was found to be around the number needed, 3025.

Poisson disk sampling produced results similar to, but not better than orthogonal sampling, in terms of standard deviation. From Figure 5, we can see that the radius of the respective confidence interval is really close to the Orthogonal curve; the average area in the two methods is significantly the same, as the t-test performed on the 100 areas resulted in a p-value equal to 0.47 (t-test value= -0.71, with Welch approximation). On the other hand, the algorithm was really quick and took only 0.39 seconds to finish one simulation, while orthogonal took 37.1 seconds. Therefore this is quite a good improvement. The results of the method are summarized in Table 1.

The Quasi-Monte Carlo method with Hammersley sequence was randomised by adding some noise with distribution $U(-0.05, 0.05)$; the results for this method are summarized in Table 1. The standard deviation is lower than in Poisson disk, but it is still bigger than 0.011, the standard deviation found with orthogonal sampling. The average time for one simulation, though, is quite smaller than the average time for Poisson disk. The average area in Quasi-Monte Carlo and in Orthogonal is significantly the same, as the t-test performed on the 100 areas resulted in a p-value equal to 0.1676 (t-test value= -1.385, with Welch approximation). Finally,

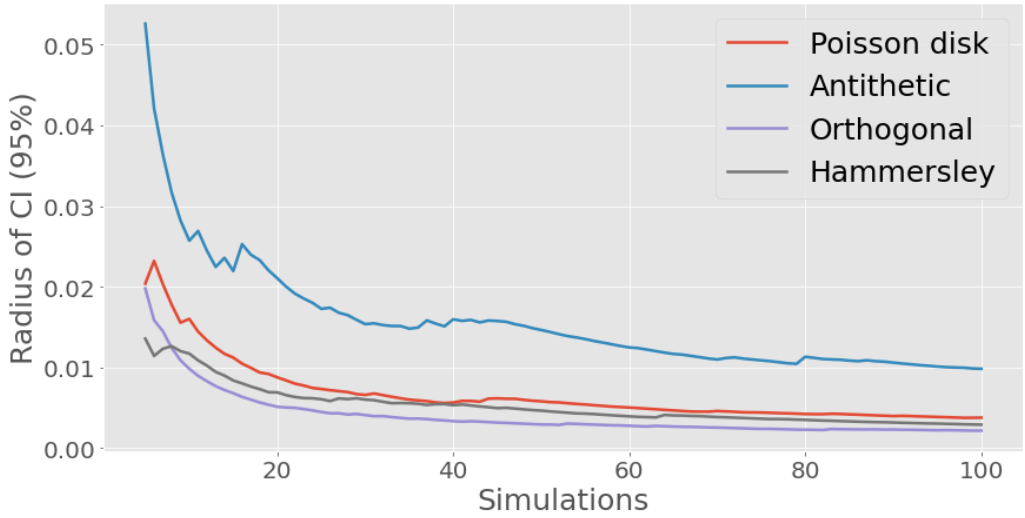

Figure 5: Radius of 95% confidence interval of the average area for the improving methods, with set Orthogonal as the benchmark.

coherently with what was said above, in Figure 5 we see that the radius of the 95% confidence interval is in between Orthogonal and Poisson disk sampling.

4 Discussion

The optimal value for the number of iterations was found by observing the average error between the area calculated with varying number of iterations and the area calculated for the biggest number of iterations; the sample had a fixed size and was randomly generated. The error was considered to be small enough when equal to 0.001, which led to $i = 2582$ as optimal value. To investigate the optimal value for the number of samples, the threshold was set to 0.05 for the standard deviation. The behavior was observed for the three *standard* sampling methods, but the optimal value was established via random sampling; $s = 3025$. By observing how the variance changed with increasing simulations, for different sampling methods, would allow firstly to check whether the number of simulations used during the previous experiments was enough; secondly, would lead to get a criterion in comparing the sampling methods at hand.

The way antithetic variables were used in this analysis did not lead to an improvement of the random sampling method because the Mandelbrot area is not a monotone function. On the other hand, this technique could have been used in order

to halve the computational time needed for the orthogonal sampling, which is the main drawback of this sampling method. Given that the Mandelbrot area is symmetric, another way of halving the computational time could be to work only on half of the set: the sampled area would have 0 as the lower bound of the complex part; after the whole process, the value for the area would be the doubled value of the half area. Finally, by applying both antithetic variables and the halving area method, one fourth of the time could be saved, but the quality of the result should be investigated.

Even though the best sampling method in terms of standard deviation was found to be orthogonal sampling, the quality of the results have to be balanced in terms of computational time utilised. In this case, the randomised Quasi-Monte Carlo method would yield the best results as it provides standard deviation that is comparable to the one observed with orthogonal sampling but with an average simulation time lower by a magnitude of around a 100. However, it must be taken into consideration that the Hammersley sequence is deterministic and the stochasticity was added artificially by introducing a small noise from $U(-0.05, 0.05)$. Therefore, the SD of randomised Quasi-Monte Carlo method is directly proportional to the magnitude of sound introduced and therefore, this method could yield imprecise results. To avoid this, the best option in terms of result precision and computational time would be the Poisson Disk sampling as it is both purely stochastic, fast and yields results significantly the same as orthogonal as found by carrying out the Welch t-test.

The results of this study could be further improved by exploring other methods of sampling, in particularly for randomised Quasi-Monte Carlo method. These could include Korobov Lattice, Sobol sequence or Chebyshev sampling. Different randomisation methods could also be experimented with, such as randomly shifted lattice (L'Ecuyer [2016]).

References

R. P. Taylor and J. C. Sprott. Biophilic fractals and the visual journey of organic screen-savers. *Nonlinear Dynamics, Psychology, and Life Sciences*, 12(1):117–129, January 2008. ISSN 1090-0578.

C. W. Kilmister. *The Mathematical Gazette*, 62(420):130–132, 1978. ISSN 00255572. URL `http://www.jstor.org/stable/3617679`.

Kerry Mitchell. A statistical investigation of the area of the mandelbrot set. 2001.

URL https://www.fractalus.com/kerry/articles/area/mandelbrot-area.html.

Sheldon M. Ross. *Simulation*. Elsevier Science Publishing Co Inc, 2012.

Art B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2(2):439–452, 1992. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24304869.

Stephen Leary, Atul Bhaskar, and Andy Keane. Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics*, 30(5):585–598, 2003. URL https://eprints.soton.ac.uk/22393/.

Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 Sketches*, SIGGRAPH '07, page 22–es, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781450347266. doi: 10.1145/1278780.1278807. URL https://doi.org/10.1145/1278780.1278807.

Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, USA, 1992. ISBN 0898712955.

Pierre L'Ecuyer. Introduction to (randomized) quasi-monte carlo. 2016.