

# Basics of Monte Carlo Simulations 2021 Report

Jake Muff 25/01/21

## Question 1

Please see attached scan, detailing my answer. The scan is also in the archive.

## Question 2

The code is attached in the same folder as this document under `buffon.cpp`. Note that in this I did not use the standard `rand()` function given in C/C++, because as discussed in class and in Numerical Recipes it is not that great. There is a better version in the `random.h` header file included in the current versions of C++ (not sure about C) which uses the Mersenne Twister RNG.

The averages errors are shown in the `dat1.txt` text file and plotted below. As we can see the average error goes below 0.0001 some point after  $10^7$  needles. I did not calculate this explicitly because I did not know how to calculate it relatively quickly whilst still maintaining precision. My initial plan was to loop through the numbers 1 to  $10^7$ , however I quickly realised this would take a long time.

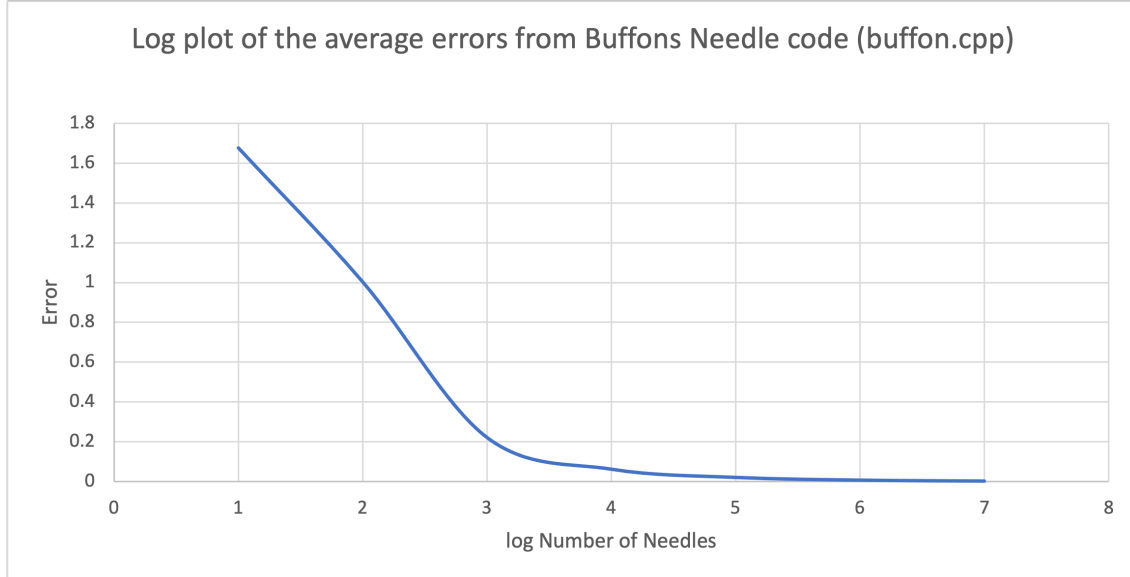


Figure 1: Log plot of Averages errors as a function of the number of needles

### Question 3

This was fairly simple, but was not without hiccups. In particular the interval of the Mersenne twister was not explicitly calculated due to its know and incredibly high period. The other two intervals were calculated and are what I expected given the conditions inputted.

For this question the header file `rng.h` was made which includes all the relevant functions.

### Question 4

The plots for the LCG and Mersenne Twister are below. The plot for the Park Miller was not included because there is a bug somewhere in my code that is stopping it from writing new values to file. By inspection of the text files you can see this error immediately. I did not complete the 2D distribution of numbers between -1 and +1.

From the distributions we can see that Mersenne Twister method produces the noisiest plots with the LCG seemingly repeating.

If I was to use LCG for my Buffon's needle simulation it would produce results with greater error, however I ran out of time to implement this.

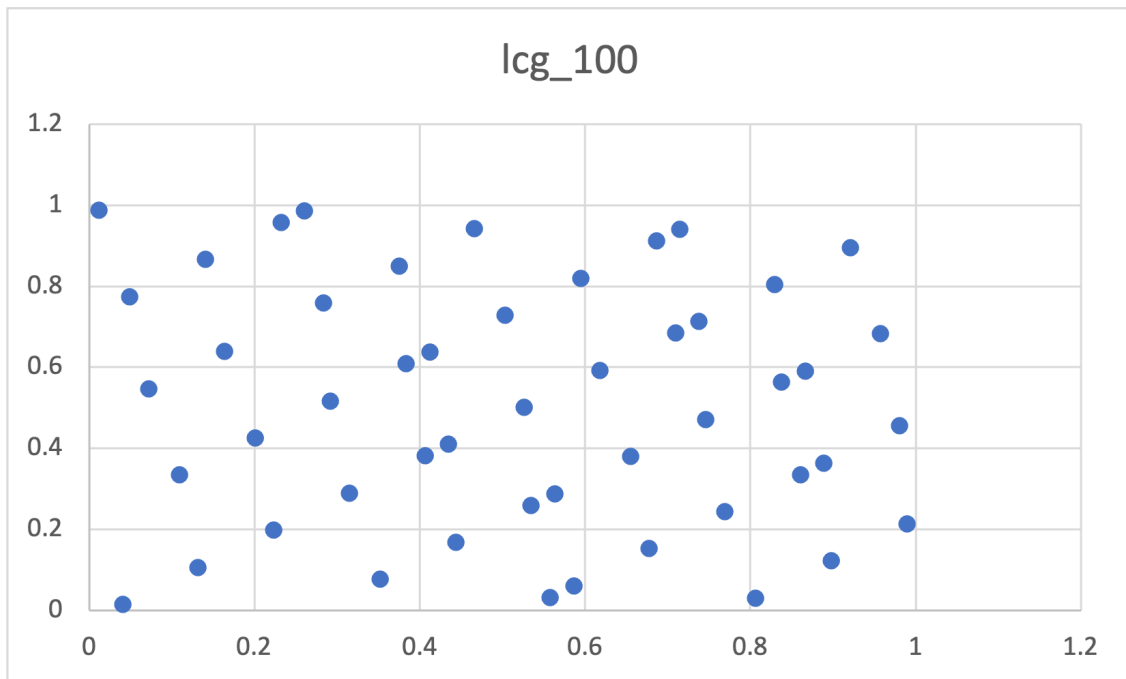


Figure 2: 2D points plot for LCG with 100 points

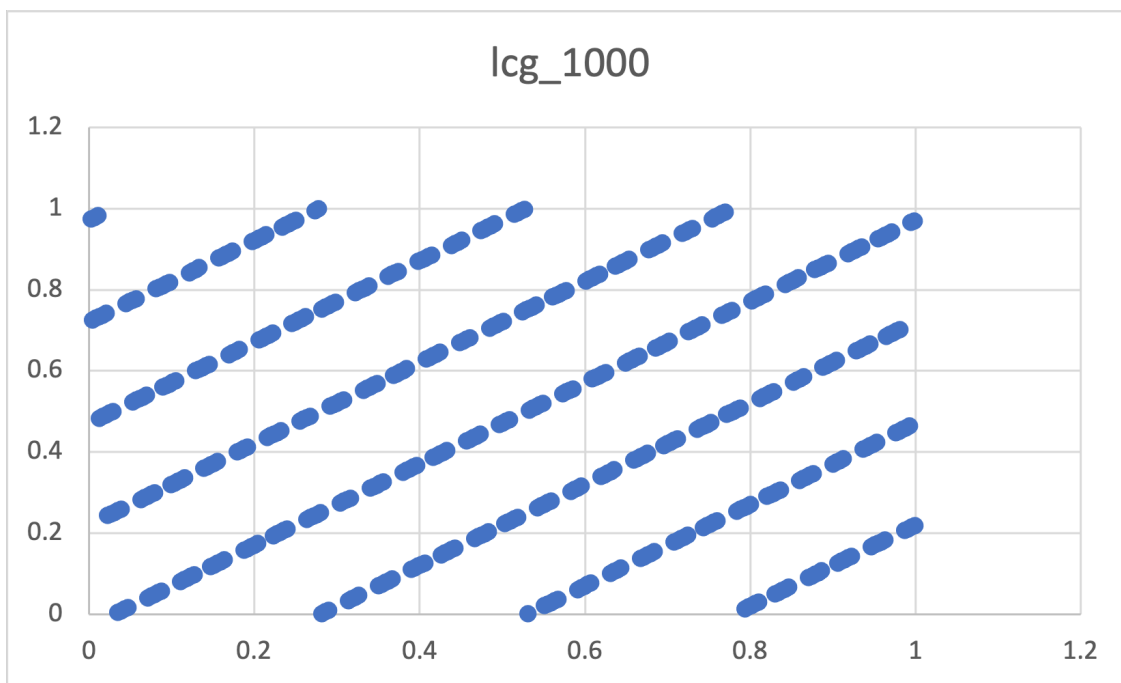


Figure 3: 2D points plot for LCG with 1000 points

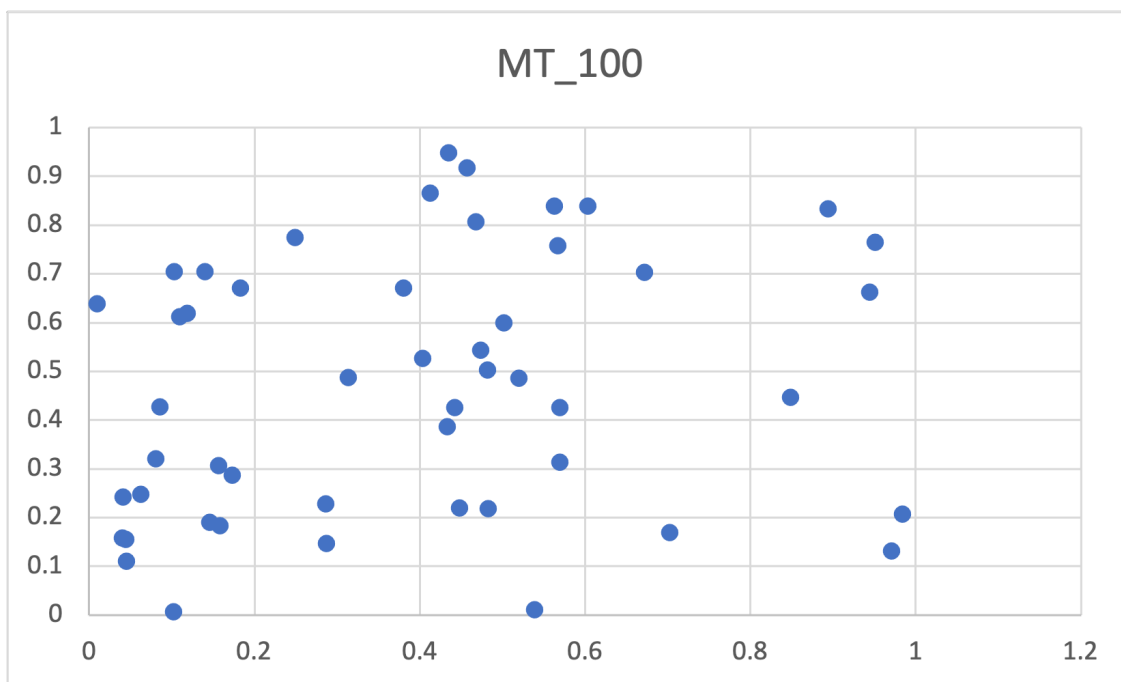


Figure 4: 2D points plot for MT with 100 points

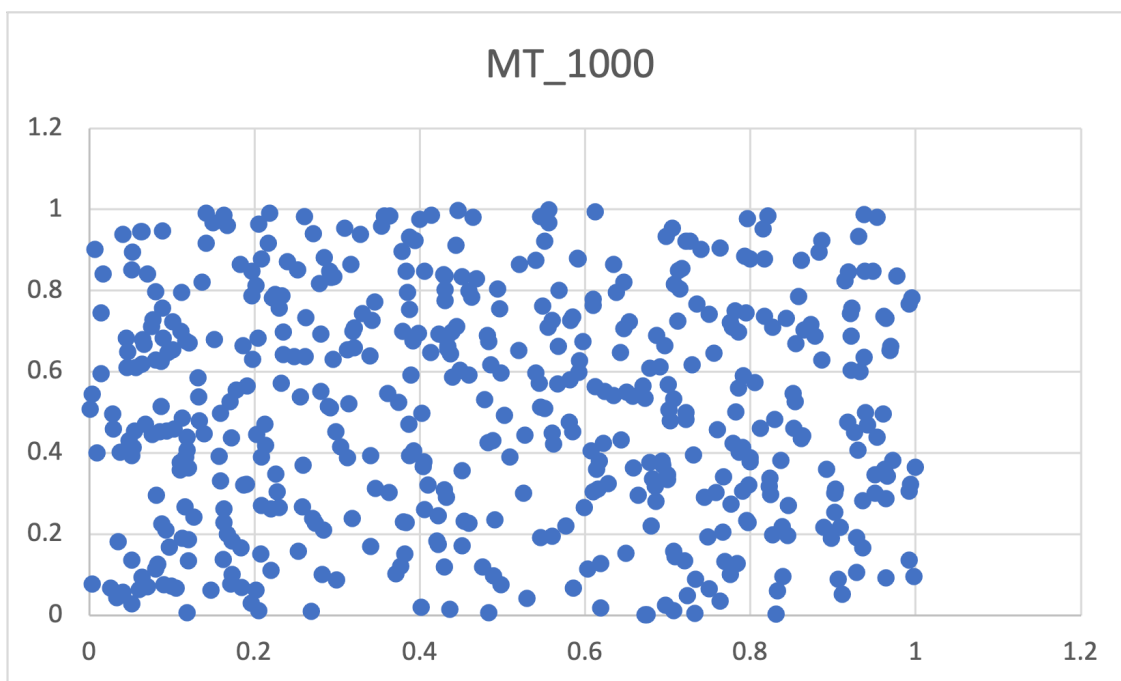


Figure 5: 2D points plot for MT with 1000 points