

Excercise 1

MARKUS KORPINEN

Basics if Monte Carlo -simulations
Department of Physics, University of Helsinki

January 20, 2012

Problem 1

$$P_{hit} = P_1 * P_2,$$

where P_1 is the probability that center of needle is less than $l/2$ away from line and P_2 is the the probability that the needle is in a right angle to cross the line. Probabilities P_1 and P_2 are depending on each other so they have to be multiplied. Because P_1 can fall to eather side of line

$$P_1 = \frac{l/2 + l/2}{d} = \frac{l}{d}.$$

Probability that needle crosses the line when center falls x away is a function of x . The probability P_2 when needle drops somewhere can be obtained by integrating and setting $l = 2$ (unit circle)

$$P_2 = 4 \int_0^1 \frac{\cos^{-1}(x)}{2\pi} dx = \frac{2}{\pi}.$$

Therefore

$$P_{hit} = P_1 * P_2 = \frac{2l}{\pi d}.$$

a)

a)

Problem 3

Problem 4

Problem 6

Following block of code can be found from the source code of Mersenne Twister:

```
/* initializes mt[N] with a seed */
void init_genrand( unsigned long s)
{
    mt[0]= s & 0xffffffffUL;
    for (mti=1; mti<N; mti++) {
        mt[mti] =
            (1812433253UL * (mt[mti-1] ^ (mt[mti-1] >> 30)) + mti);
        /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier.
        /* In the previous versions, MSBs of the seed affect */
        /* only MSBs of the array mt[]. */
        /* 2002/01/09 modified by Makoto Matsumoto
        mt[mti] ^= 0xffffffffUL;
```

```
        /* for >32 bit machines */  
    }  
}
```

Figure 1: When you throw only 10 times the predicted value for π is bad.

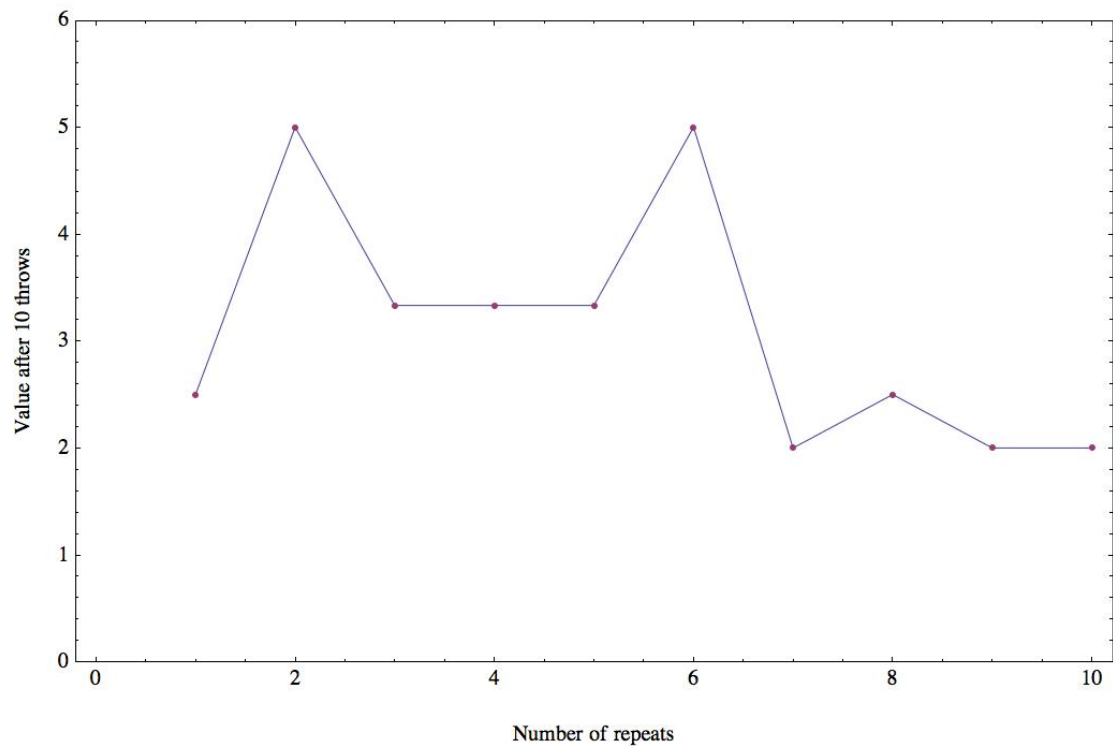


Figure 2: After approximately 5000 throws result starts to converge.

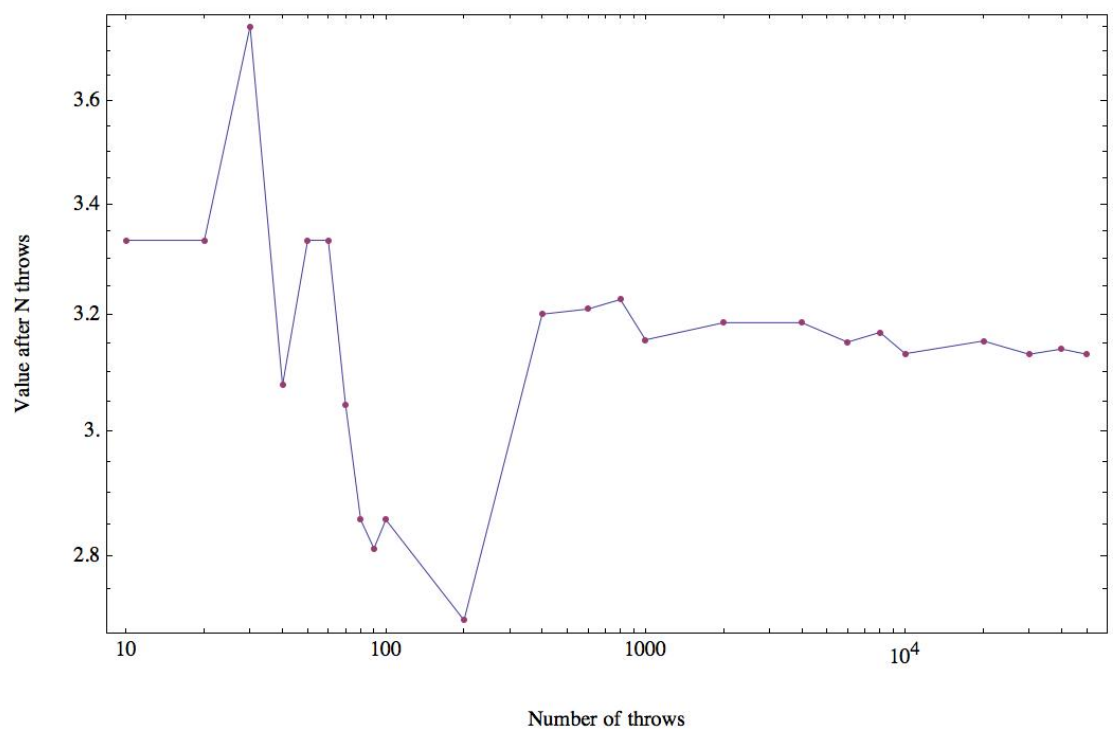


Figure 3: 100 numbers generated with the quick-and-dirty generator

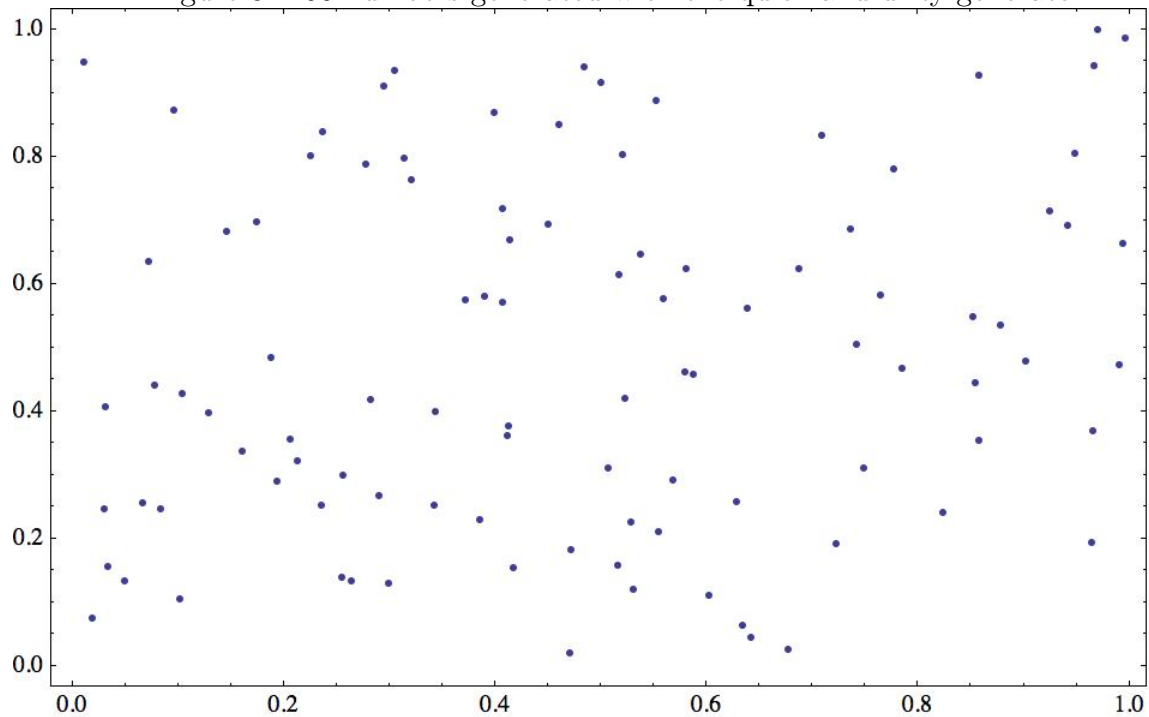


Figure 4: 10000 numbers generated with the quick-and-dirty generator

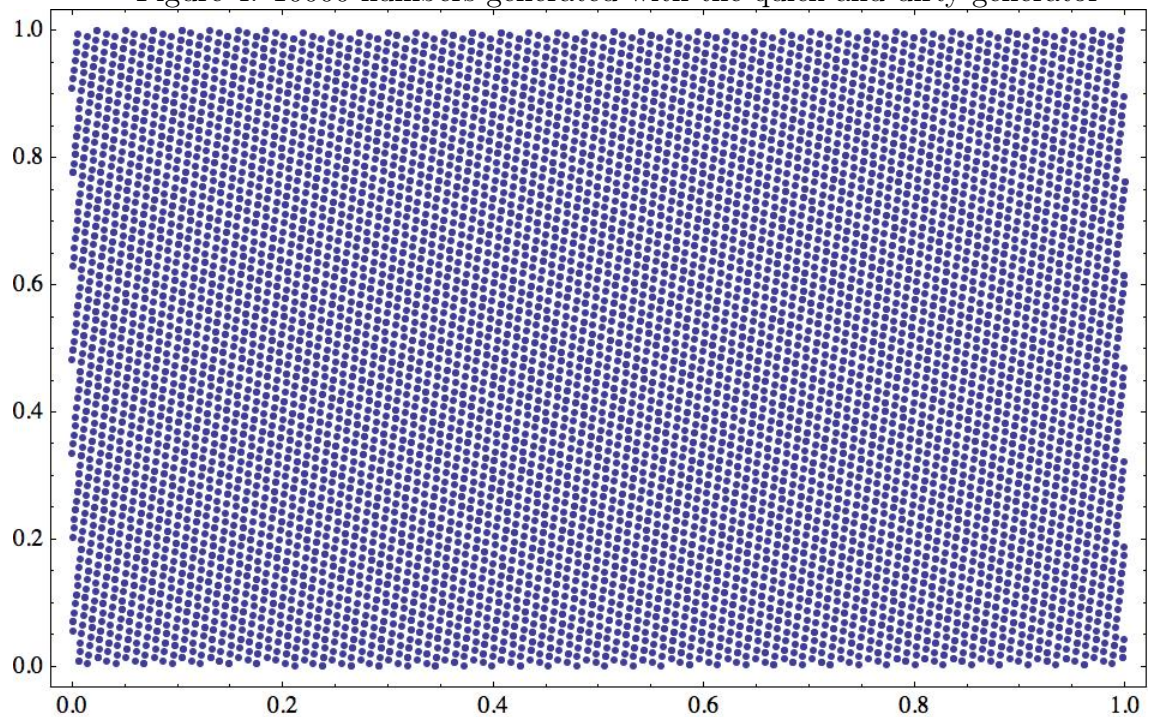


Figure 5: 100 numbers generated with the Park-Miller generator

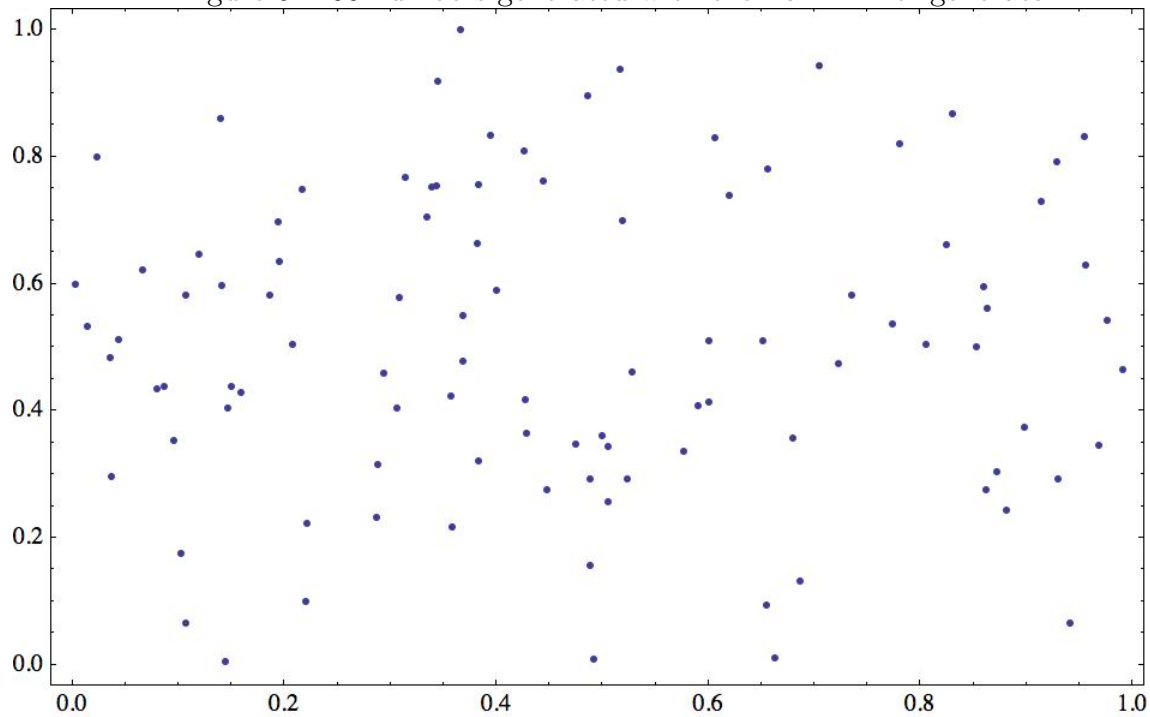


Figure 6: 10000 numbers generated with the Park-Miller generator

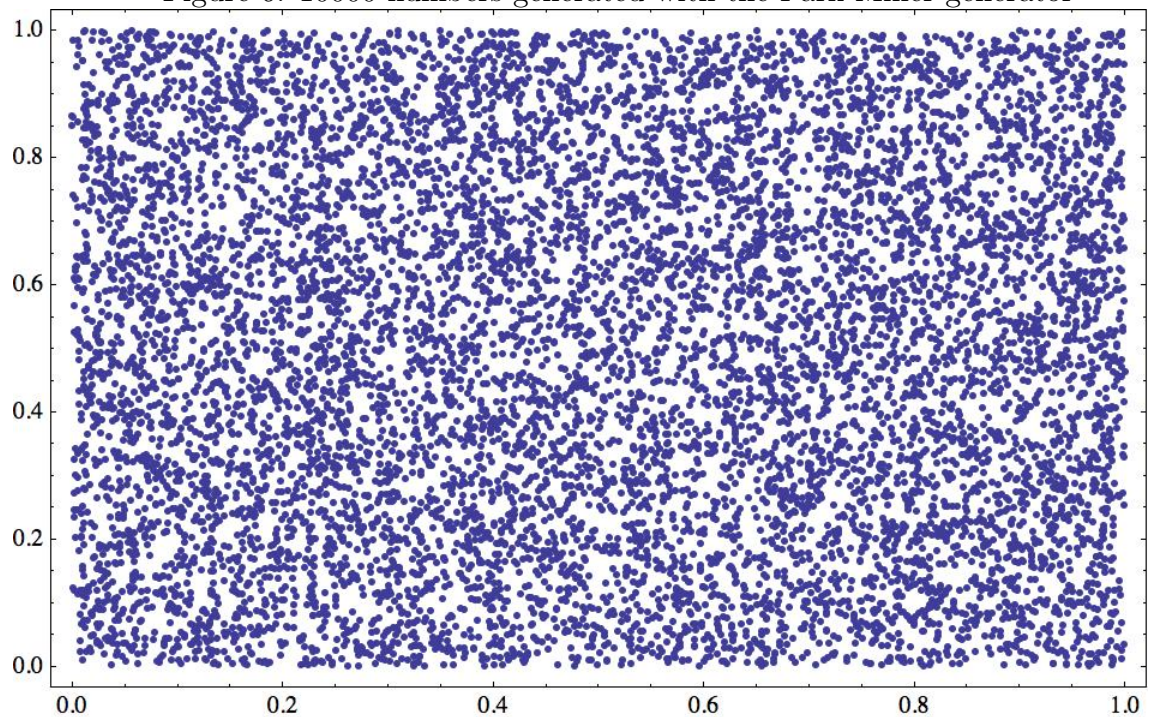


Figure 7: 10 000 points with the quick-and-dirty generator with interval $x=[0, 0.0001]$

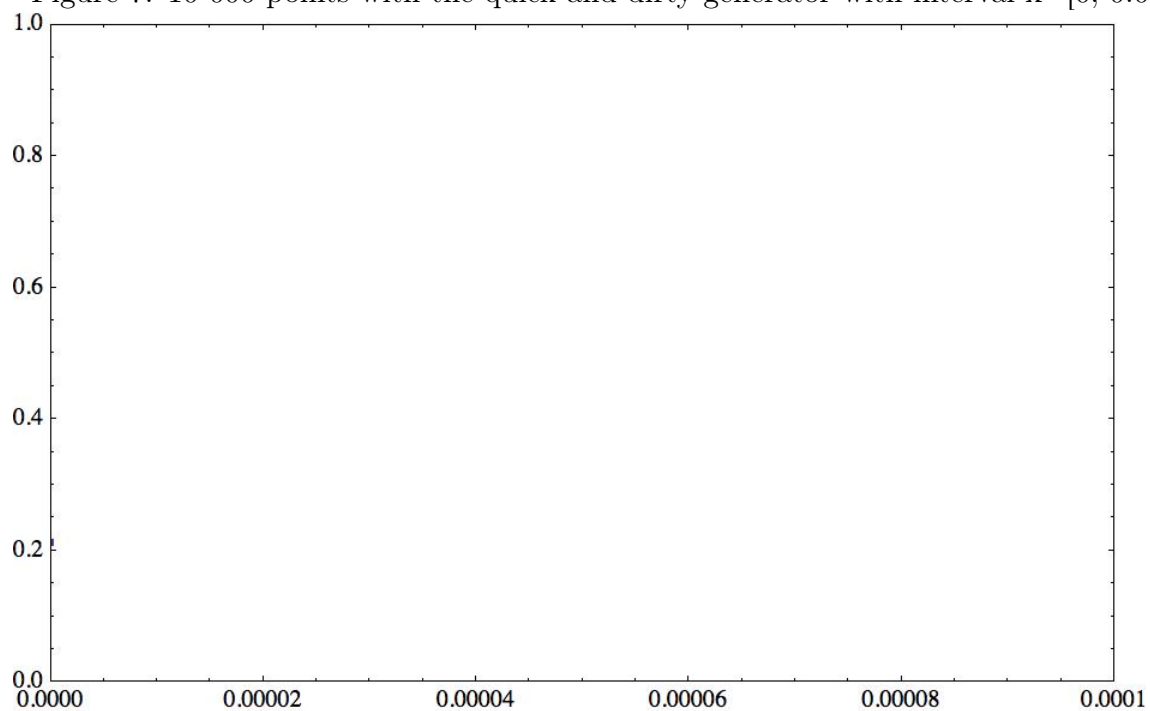


Figure 8: 10 000 points with the quick-and-dirty generator with interval $x=[0, 0.001]$

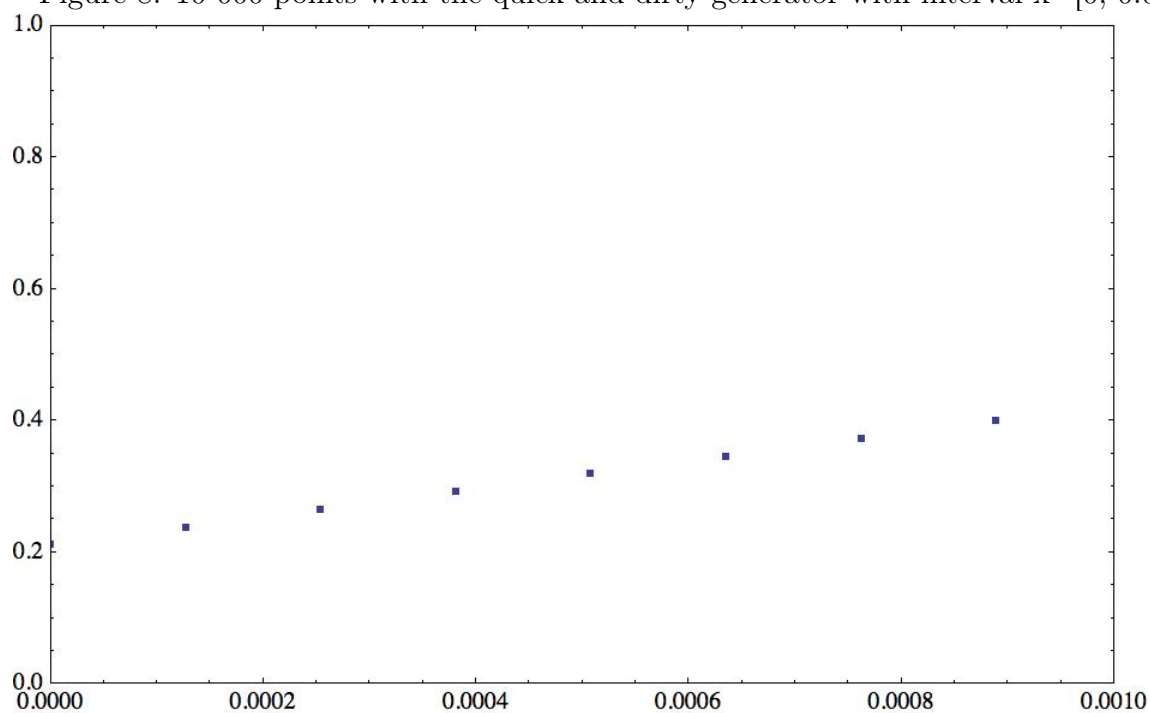


Figure 9: 10 000 points with the Park-Miller generator with interval $x=[0, 0.0001]$

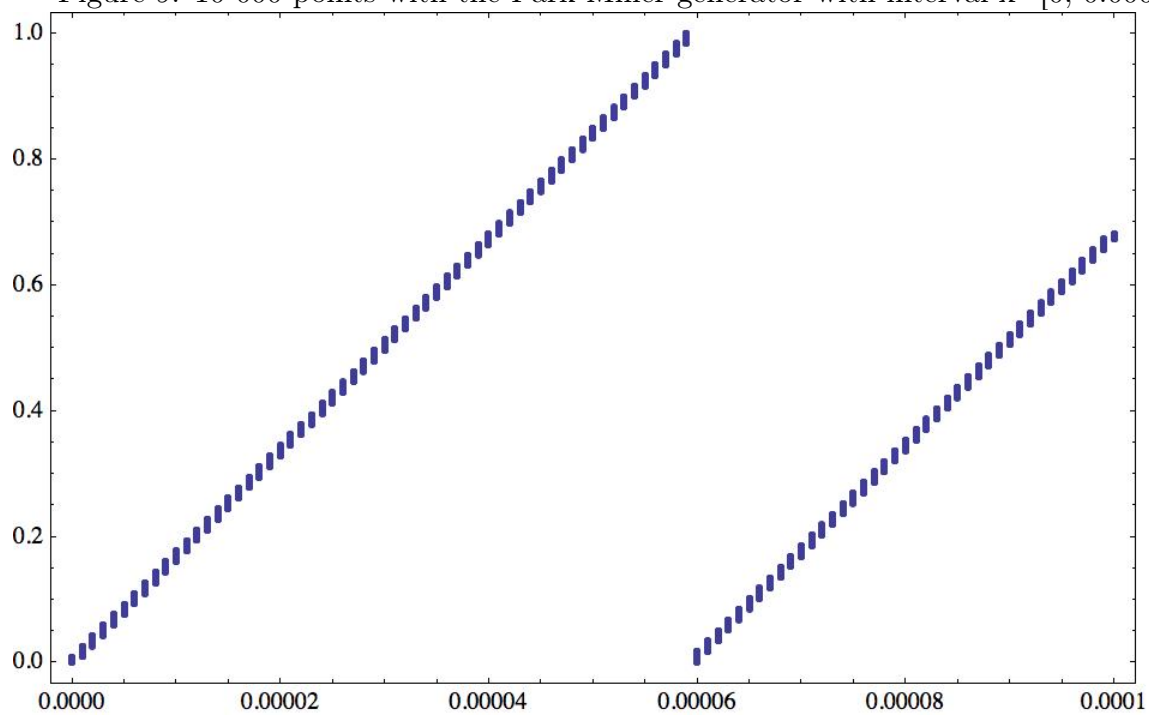


Figure 10: 10 000 points with the Park-Miller generator with interval $x=[0, 0.001]$

