Aufgabe6

November 1, 2018

1 Aufgabe 6

2 a)

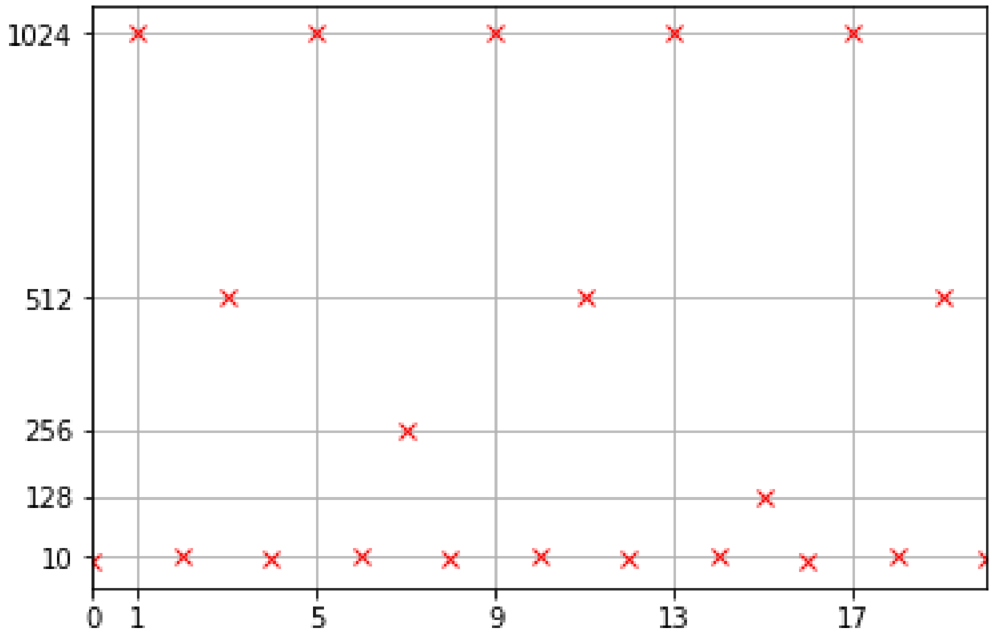
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

In [2]:
```
def lcg(a,b,m,x):
    return ((a*x+b)%m)
```

In [3]:
```
def randGen(a,b,m,x0):
    i = 1
    x=x0
    u=x0/m
    list=[u]
    while(True):
        x = lcg(a,b,m,x)
        u = x/m
        if(u in list):
            break
        list.append(u)
        i +=1
    #print("Elemente in der Liste mit a=",a," : ",i)
    return i
```

In [4]:
```
x0 = 42
L=[]
for a in range(1024):
    L.append(randGen(a,3,1024,x0))
```

In [5]:
```
a=range(1024)
plt.plot(a,L,'rx')
plt.xlim(0,20)
plt.grid()
plt.yticks([10,128,256,512,1024])
plt.xticks([0,1,5,9,13,17])
plt.show()
```



Um eine maximale Periodenlänge zu erhalten müssen folgende Bedingungen gelten:
-$c \neq 0$
-b und m müssen teilerfremd sein
-$(a-1)$ muss durch die Primfaktoren von m teilbar sein
-$(a-1)$ muss durch 4 teilbar sein, wenn m durch 4 teilbar ist

Weil in diesem Fall $m = 1024$ und $b = 3$ vorliegen, ist schnell zu erkennen, dass diese teilerfremd sind. Da $m = 1024 = 2^{10}$ ist, ist 2 der einzige Primfaktor von m. Also muss $(a-1)$ durch 4 und 2 teilbar sein für eine maximale Periodenlänge.

Wie im Plot zu sehen sind die maximalen Ausschläge an den Punkten $a = 1, 5, 9, 13, 17, ...$. Diese folgen genau den vorher gennanten Regeln.

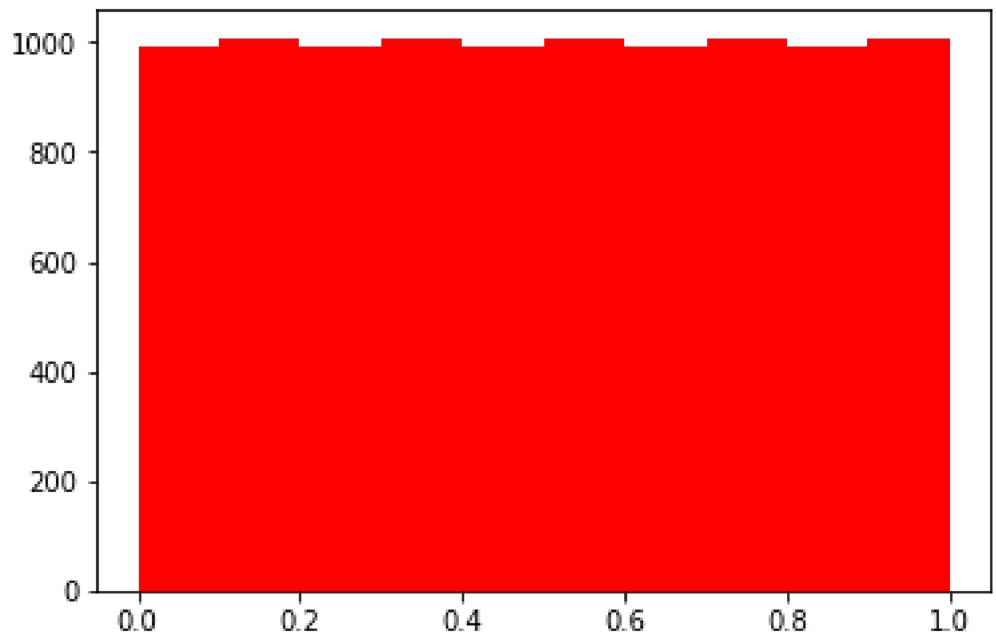
3 b)

```
In [6]: def randGen2(a,b,m,x0):
            x=x0
            list=[x0/m]
            for y in range(10000):
                x = lcg(a,b,m,x)
                u = x/m
                list.append(u)
            return list
```

```
In [7]: m = 10000
        a = 1601
        b = 3456
```

```
In [8]: rnd_nums = randGen2(a,b,m,x0)
```

```
In [9]: from scipy import stats
        binnedData = np.histogram(rnd_nums, bins=10, range=(0,1))[0]
        pValue = stats.chisquare(binnedData)[1] #Berechung des p-Werts mit chi-quadrat Test.
        plt.hist(rnd_nums,bins=10,range=(0,1),color='r')
        plt.show()
        print('pValue =',pValue)
```



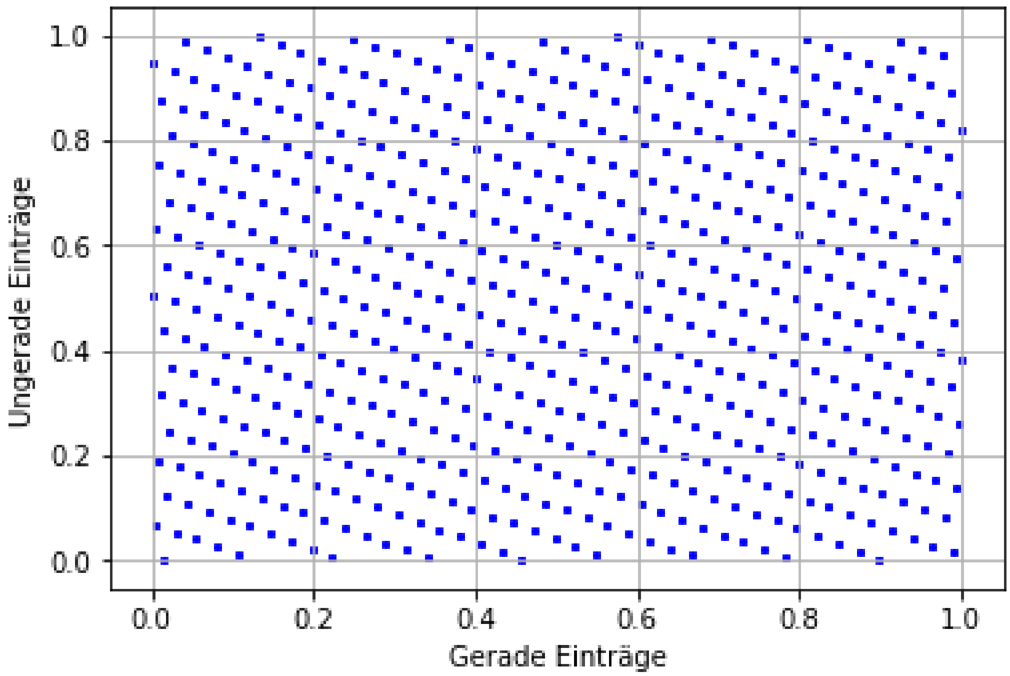

```
pValue = 0.999921124599
```

Es ist klar zu sehen, dass die Verteilung unabhängig vom Startwert ist. Der p-Wert ist sehr nah an 1, also ist es eine sehr gute Gleichverteilung.

4 c)

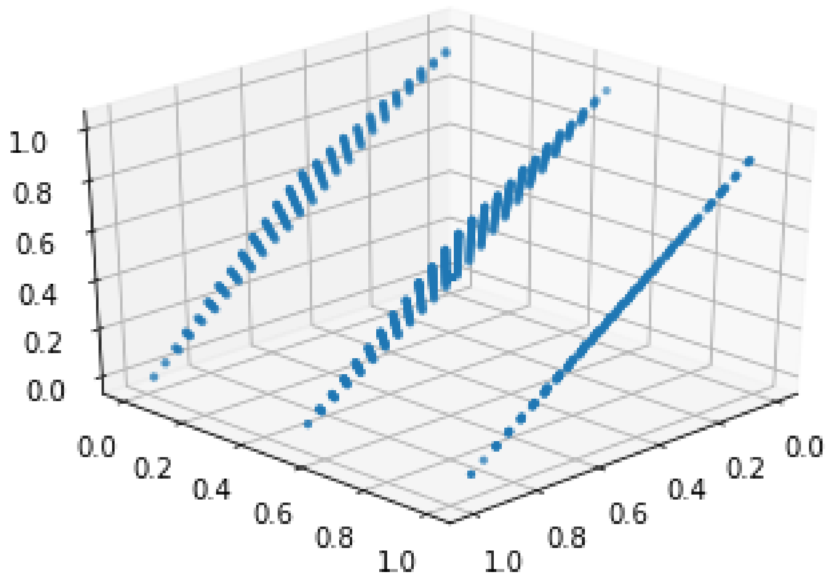
```
In [10]: rnd_nums_1 = np.zeros(5000)
         rnd_nums_2 = np.zeros(5000)
         x = 0
         y = 0
         while x in range(10000) and y in range(10000):
             rnd_nums_1[y] = rnd_nums[x] #Aufteilung der Zahlen in zwei Arrays
             rnd_nums_2[y] = rnd_nums[x+1]
             x += 2
             y += 1
```

```
In [11]: plt.scatter(rnd_nums_1,rnd_nums_2,color='b' ,s=2)
         plt.xlabel('Gerade Einträge')
         plt.ylabel('Ungerade Einträge')
         plt.grid()
         plt.show()
```




```
In [12]: rnd_nums_3 = np.zeros(3333)
         rnd_nums_4 = np.zeros(3333)
         rnd_nums_5 = np.zeros(3333)
         x = 0
         y = 0
         while x in range(3333) and y in range(3333):
             rnd_nums_3[y] = rnd_nums[x]
             rnd_nums_4[y] = rnd_nums[x+1]
             rnd_nums_5[y] = rnd_nums[x+2]
             x += 3
             y += 1
```

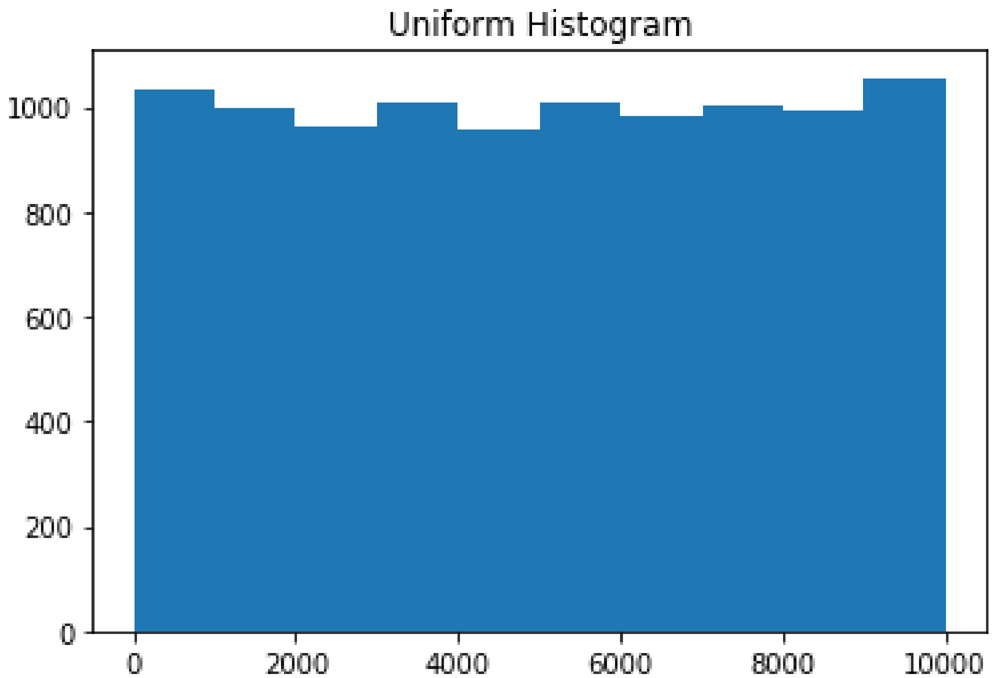
```
In [13]: fig = plt.figure()
         ax = fig.add_subplot(111,projection='3d')
         ax.scatter(rnd_nums_3,rnd_nums_4,rnd_nums_5, lw=0, s=10)
         ax.view_init(30,45)
         plt.show()
```



In den Scatter-Plos ist deutlich zu sehen, dass die Werte an diskreten Stellen gehäuft vorkommen. Somit gibt es auch viele nicht gefüllte Lücken. Somit ist es nur möglich eine begrenzte Anzahl an Zufallszahlen zu erhalten, bevor sich nach einer Periodenlänge die Zahlen wiederholen. Auch liegen die Wertepaare auf parallelen Geraden und im 3D-Plot auf Ebenen. Somit ist das Verhältnis zwischen den Paaren dort gleich.

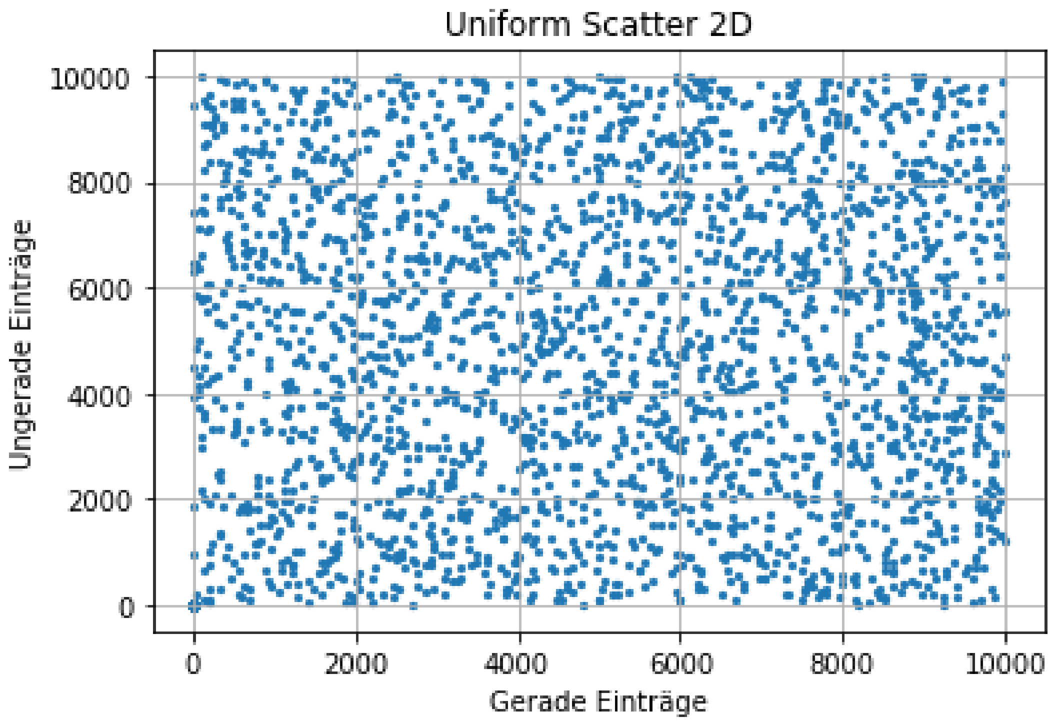
5 d)

```
In [14]: x_u = np.random.uniform(0,10000,10000)
         bin_len = len(x_u)
         plt.hist(x_u, histtype ='stepfilled')
         plt.title('Uniform Histogram')
         plt.show()
```



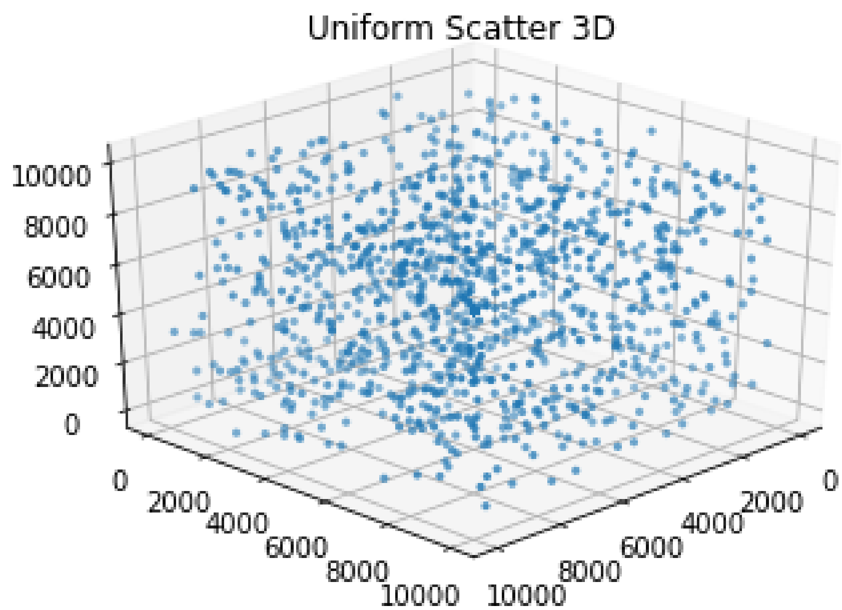

```
In [15]: x_u1 = np.zeros(5000)
         x_u2 = np.zeros(5000)

         x = 0
         y = 0
         while x in range(5000) and y in range(5000):
             x_u1[y] = x_u[x]
             x_u2[y] = x_u[x+1]
             x += 2
             y += 1
```

```
In [16]: plt.scatter(x_u1,x_u2, s=5)
         plt.xlabel('Gerade Einträge')
         plt.ylabel('Ungerade Einträge')
         plt.title('Uniform Scatter 2D')
         plt.grid()
         plt.show()
```



In [17]:
```
x_u3 = np.zeros(3333)
x_u4 = np.zeros(3333)
x_u5 = np.zeros(3333)
x = 0
y = 0
while x in range(3333) and y in range(3333):
    x_u3[y] = x_u[x]
    x_u4[y] = x_u[x+1]
    x_u5[y] = x_u[x+2]
    x += 3
    y += 1
```

In [18]:
```
fig = plt.figure()
ax = fig.add_subplot(111,projection='3d')
ax.scatter(x_u3,x_u4,x_u5, lw=0, s=10)
ax.view_init(30,45)
plt.title('Uniform Scatter 3D')
plt.show()
```



Es ist ein deutlicher Unterschied zu den vorherigen Plots zu erkennen. Die mit uniform geplotteten haben eine wesentlich gröSSeren Zufall inne. Es ist keine diskrete Verteilung zu erkennen, da die Punkte sehr gestreut vorkommen.

6 e)

```
In [19]: def randGen3(a,b,m,x0):
             i = 1
             x=x0
             u=x0/m
             list=[u]
             while(True):
                 x = lcg(a,b,m,x)
                 u = x/m
                 if(u in list):
                     break
                 list.append(u)
                 i +=1
             return list
```

```
In [20]: L2=[]
         for a in range(1024):
             L2.append(randGen3(a,3,1024,x0))
```

```
In [21]: A=[]
         for a in range(1024):
                 if 0.5 in L2[a]:
```

```
            A.append(a)
    for x in range(len(A)): #Überprüfung ob alle Einträge den Regeln für max Perioden geh
        A[x] = (A[x]-1)%4
```

0.5 in L2 mit folgenden a: [1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113,

117, 121, 125, 129, 133, 137, 141, 145, 149,..., 1001, 1005, 1009, 1013, 1017, 1021]

Überprüfung ob das a den Regeln für lange Periodizität entspricht: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,..., 0, 0, 0, 0, 0, 0]

Da $a\%4 = 0$ ist es für alle a erfüllt.

Das sind die "a's" mit denen eine maximale Periodenlänge erreicht werden konnte, aufgrund der oben im Abschnitt a) genannten Regeln.

Mögliche Startwerte um $\frac{1}{2}$ zu erhalten müssen folgende Bedingung erfüllen:

$$\frac{(ax_0 + b)\%m}{m} = 0.5(ax_0 + b)\%m = 0.5 \cdot m(ax_0 + b) = 0.5 \cdot mx_0 = \frac{0.5 \cdot m - b}{a} \tag{1}$$

Weil m und b fest gewählt sind, hängt unser Startwert von a ab.

Für $a = 1$ wäre der Startwert bei $m = 1024$ und $b = 3$, $x_0 = 509$ bei dem $\frac{1}{2}$ garantiert vorkommt.

Es gibt keinen Startwert bei dem für alle "a's" $\frac{1}{2}$ vorkommt, weil x_0 von a abhängt.