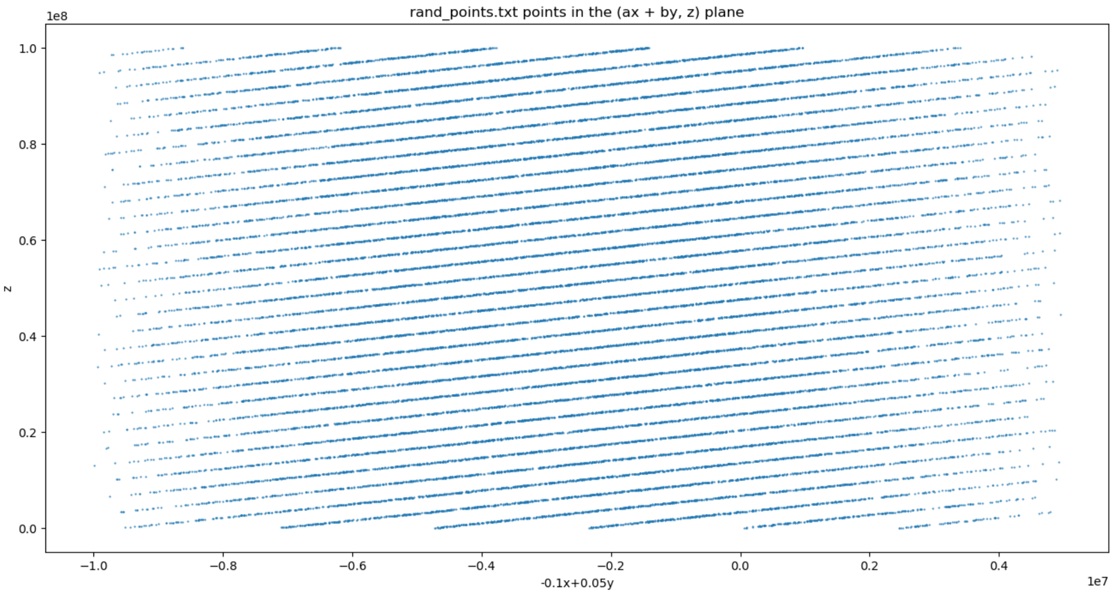
Assignment_7_code

November 21, 2022

Problem 1

[1]:

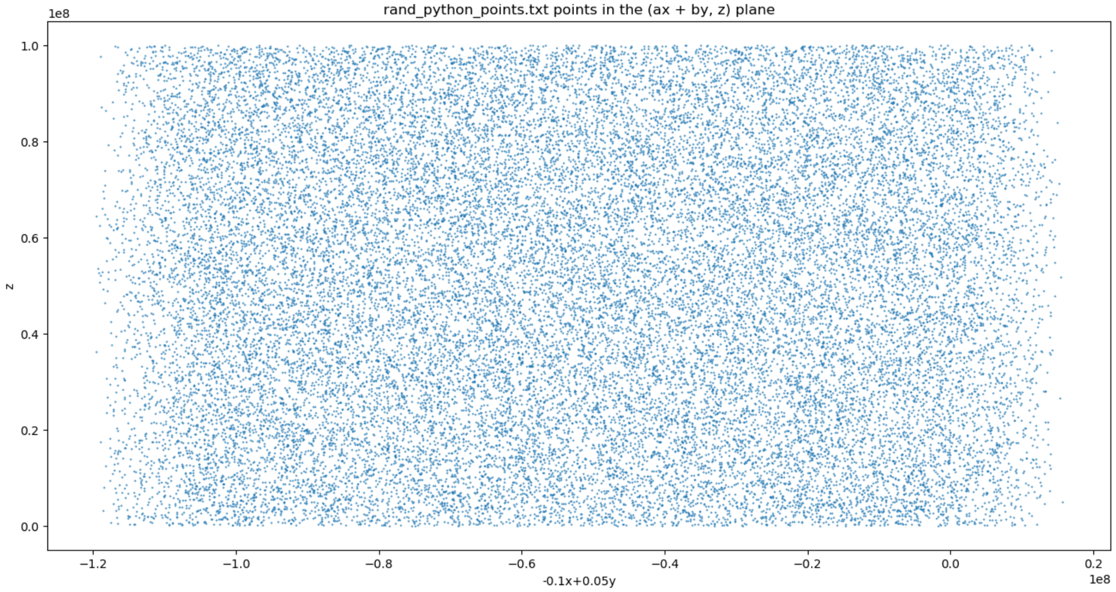
```
import matplotlib.pyplot as plt
import numpy as np
import random as rd


###### We get the data from rand_points.txt and plot it in 3D

f = open("/Users/louis/Desktop/McGill/FALL 2022/PHYS 512/Assignment/7/
 ↪phys512-2022/Assignment7/rand_points.txt", 'r')

xyz = f.readlines()

x = []
y = []
z = []

for line in xyz:
    xyzn = line.split("\n")
    x_y_z = xyzn[0].split(" ")
    x.append(int(x_y_z[0]))
    y.append(int(x_y_z[1]))
    z.append(int(x_y_z[2]))

x = np.array(x)
y = np.array(y)
z = np.array(z)

###We plot the data in (ax + by, z) plane
fig = plt.figure(figsize=(16,8))


plt.plot(-0.1*x + 0.05*y, z,".",markersize=1)
plt.xlabel("-0.1x+0.05y")
plt.ylabel("z")
plt.title("rand_points.txt points in the (ax + by, z) plane")
plt.show()
```



[2]:

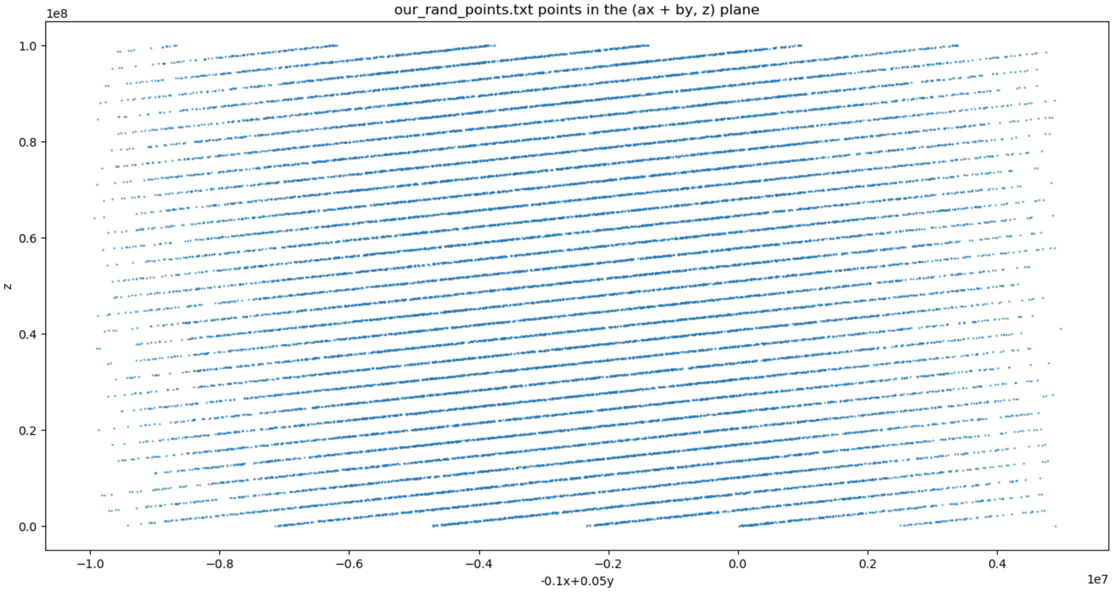
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
import random as rd






###Data making with random
"""
n=30253
x_y_z_new =np.empty(n*3,dtype='int32')
rd.seed(1)
for i in range (n*3):
    x_y_z_new[i] = rd.randint(0, 1e8)

vv=np.reshape(x_y_z_new,[n,3])
maxval=1e8
vmax=np.max(vv,axis=1)
vv2=vv[vmax<maxval,:]
x = []
y = []
z = []


f=open('./Assignment7/rand_python_points.txt','a')
```

```
for i in range(vv2.shape[0]):
    myline=repr(vv2[i,0])+' '+repr(vv2[i,1])+' '+ repr(vv2[i,2])+'\n'
    f.write(myline)
    x.append(vv2[i,0])
    y.append(vv2[i,1])
    z.append(vv2[i,2])
f.close()
"""

### We get the data we made and plot it in 3D
f=open('./Assignment7/rand_python_points.txt','r')

xyz = f.readlines()

x = []
y = []
z = []

for line in xyz:
    xyzn = line.split("\n")
    x_y_z = xyzn[0].split(" ")
    x.append(int(x_y_z[0]))
    y.append(int(x_y_z[1]))
    z.append(int(x_y_z[2]))

x = np.array(x)
y = np.array(y)
z = np.array(z)

###We plot the data in (ax + by, z) plane
fig = plt.figure(figsize=(16,8))


plt.plot(0.16*x - 1.2*y, z,".",markersize=1)
plt.xlabel("-0.1x+0.05y")
plt.ylabel("z")
plt.title("rand_python_points.txt points in the (ax + by, z) plane")
plt.show()
```



[3]:

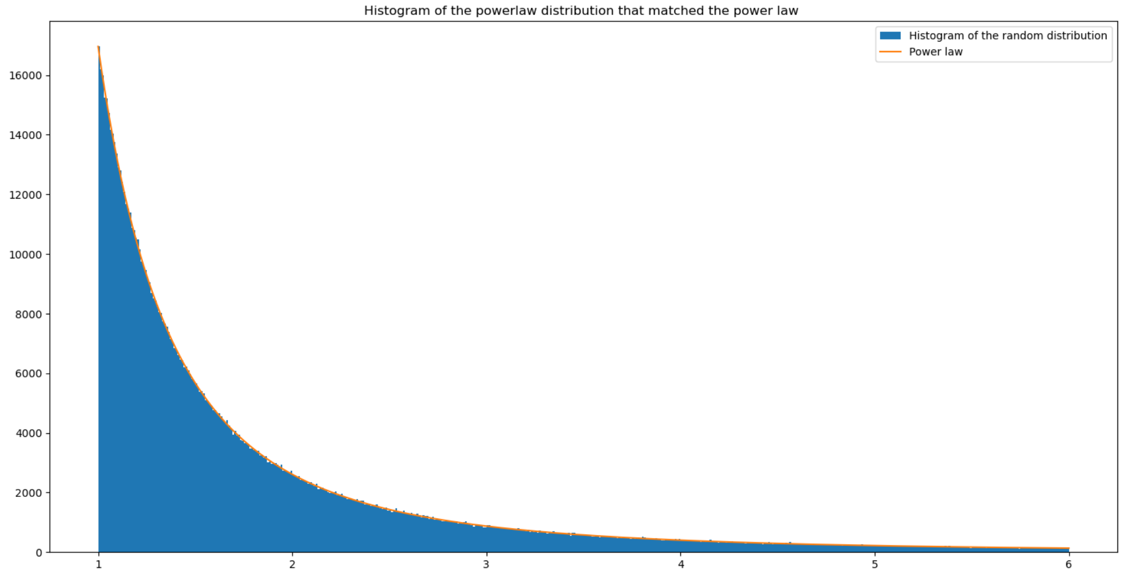
```
import numpy as np
import numba as nb
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
import random as rd


###Data making with C
"""
mylib=ctypes.cdll.LoadLibrary('libc.dylib')

rand=mylib.rand
rand.argtypes=[]
rand.restype=ctypes.c_int


@nb.njit
def get_rands_nb(vals):
    n=len(vals)
    for i in range(n):
        vals[i]=rand()
    return vals

def get_rands(n):
    vec=np.empty(n,dtype='int32')
    get_rands_nb(vec)
    return vec


n=300000000
vec=get_rands(n*3)
#vv=vec&(2**16-1)

vv=np.reshape(vec,[n,3])
vmax=np.max(vv,axis=1)

maxval=1e8
vv2=vv[vmax<maxval,:]

f=open('./Assignment7/our_rand_points.txt','a')
for i in range(vv2.shape[0]):
    myline=repr(vv2[i,0])+' '+repr(vv2[i,1])+' '+ repr(vv2[i,2])+'\n'
    f.write(myline)
f.close()
"""

###### We get the data from rand_points.txt and plot it in 3D

f = open("./Assignment7/our_rand_points.txt", 'r')

xyz = f.readlines()

x = []
y = []
z = []

for line in xyz:
    xyzn = line.split("\n")
    x_y_z = xyzn[0].split(" ")
    x.append(int(x_y_z[0]))
    y.append(int(x_y_z[1]))
    z.append(int(x_y_z[2]))

x = np.array(x)
y = np.array(y)
z = np.array(z)


###We plot the data in (ax + by, z) plane
fig = plt.figure(figsize=(16,8))


plt.plot(-0.1*x + 0.05*y, z,".",markersize=1)
plt.xlabel("-0.1x+0.05y")
plt.ylabel("z")
plt.title("our_rand_points.txt points in the (ax + by, z) plane")
plt.show()
```

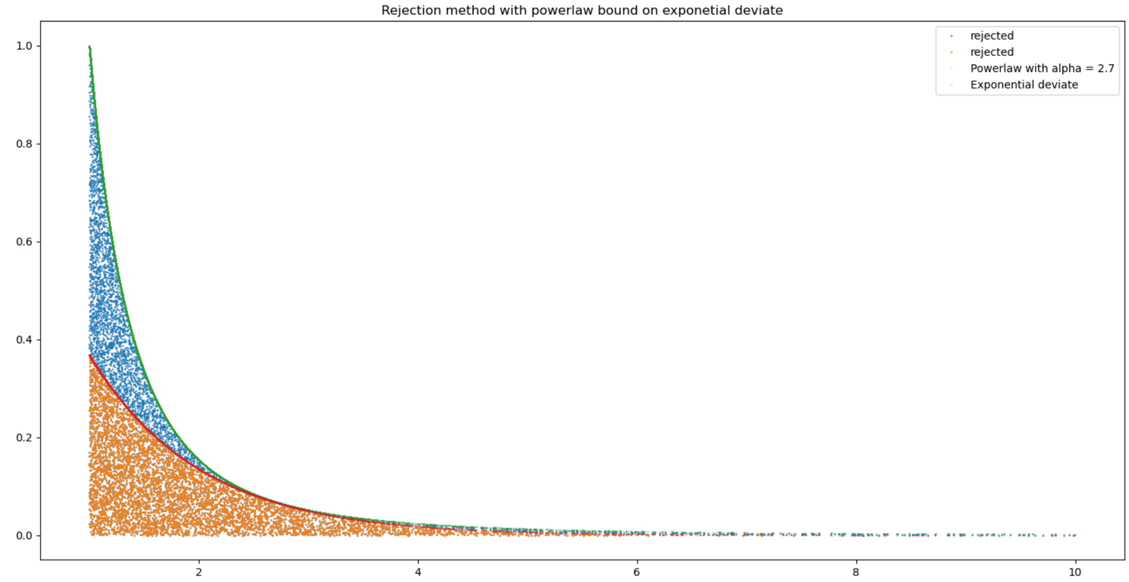


Problem 2

[4]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sc


def power_law_cdf(x, alpha):
    return x**(1/(1-alpha))
def power_law(x, alpha):
    return x**(-alpha)
def exp_ (x):
    return np.e**(-x)
x = np.random.rand(1000000)
s = power_law_cdf(x, 2.7)
bins=np.linspace(1,6,501)

plt.figure(figsize=(18,9))
plt.title("Histogram of the powerlaw distribution that matched the power law ")
n, bin, patches = plt.hist(s, bins, label="Histogram of the random␣
 ↪distribution")
plt.plot(bins, n.max()*power_law(bins, 2.7), label="Power law")
plt.legend()
plt.show()
s_x = power_law_cdf(np.random.rand(10000), 2.7)
random_power_law_set = s_x[s_x<=10]

random_values_under_exp = []
random_values_over_exp = []
random_values = np.random.rand(len(random_power_law_set))
exp_random = exp_(random_power_law_set)
power_random = power_law(random_power_law_set, 2.7)
index_accepted = np.where(random_values * power_random<= exp_random)

random_values_under_exp =  exp_random[index_accepted]
plt.figure(figsize=(18,9))
plt.title("Rejection method with powerlaw bound on exponetial deviate")
plt.plot(random_power_law_set, random_values * power_random, '.', markersize=1.
 ↪3, label="rejected")
plt.plot(random_power_law_set[index_accepted], (random_values *␣
 ↪power_random)[index_accepted], '.', markersize=1.3, label="rejected")
plt.plot(random_power_law_set, power_random, '.', markersize=.3,␣
 ↪label="Powerlaw with alpha = 2.7" )
plt.plot(random_power_law_set, exp_random, '.', markersize=.3,␣
 ↪label="Exponential deviate" )
plt.legend()
plt.show()


percentage = len(index_accepted[0])*100/random_values.size
print("The precentage of accepted data is " + str(percentage) + "%")
```





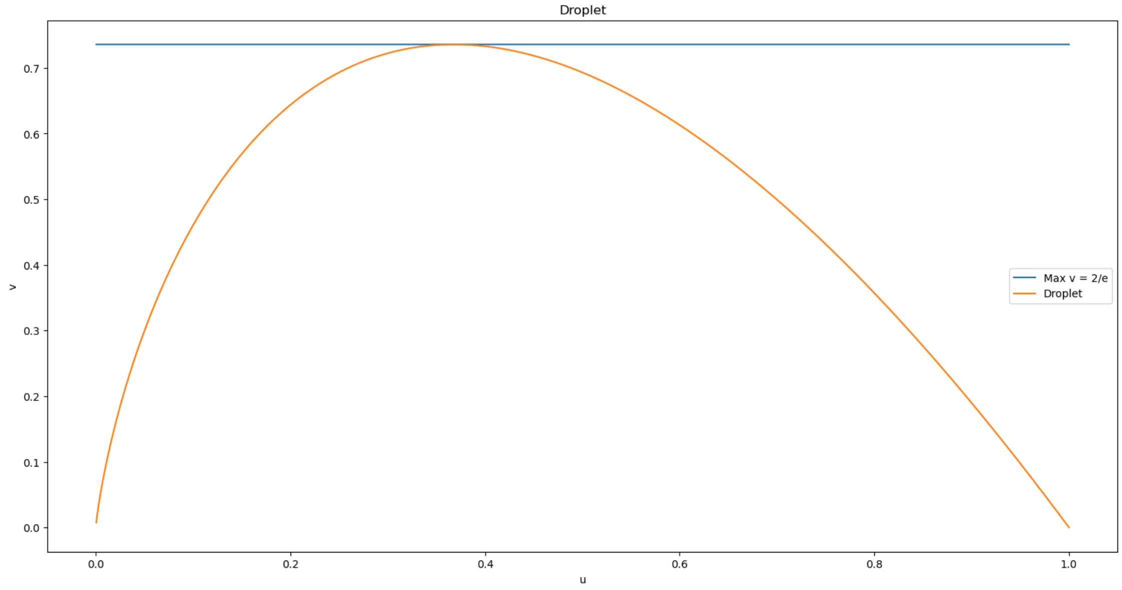
```
The precentage of accepted data is 63.46193100632782%
```

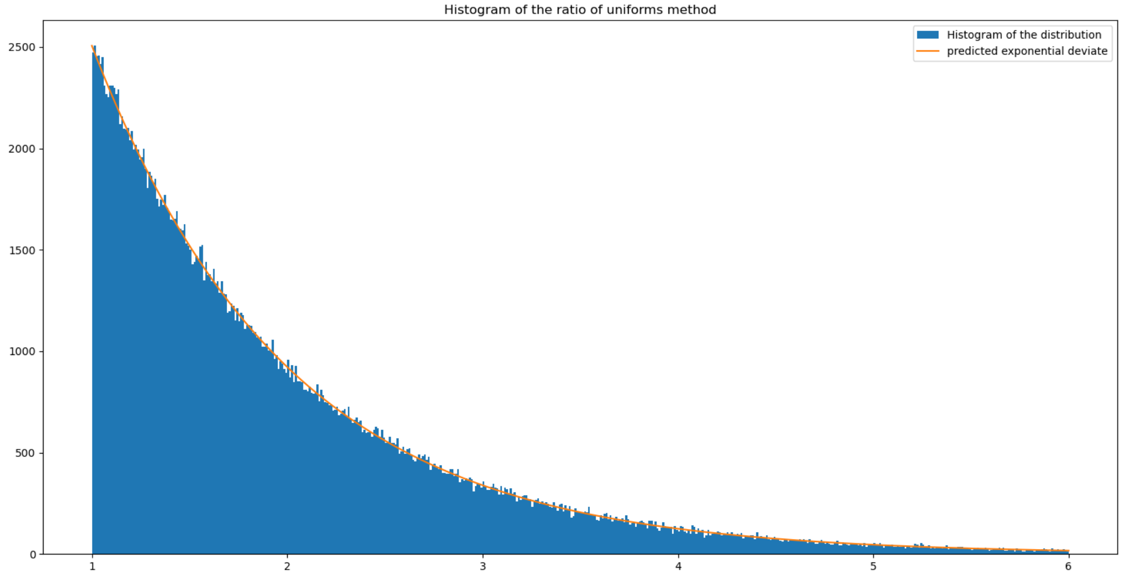
Problem 3

[5]:

```python
#code from class

import numpy as np
from matplotlib import pyplot as plt

u=np.linspace(0,1,2001)
u=u[1:]
v=2*u*(-np.log(u))
plt.figure(figsize=(18,9))
plt.title("Droplet")
plt.plot(u, np.zeros(2000)+2/np.e, label="Max v = 2/e")
plt.plot(u,v, label="Droplet")
plt.xlabel("u")
plt.ylabel("v")
plt.legend()
plt.show()

N=1000000
u=np.random.rand(N)
v=np.random.rand(N)*0.73
r=v/u
exponential_accepted=np.where(v<= 2*u*(-np.log(u)))
```

```
bins=np.linspace(1,6,501)
plt.figure(figsize=(18,9))
n, bin, patches = plt.hist(r[exponential_accepted], bins, label="Histogram of the distribution")
pred=np.exp(-bins)*n.max()*np.e
plt.plot(bins,pred, label="predicted exponential deviate")
plt.title("Histogram of the ratio of uniforms method")
plt.legend()
plt.show()
```


Droplet
v
u
Max v = 2/e
Droplet


Histogram of the ratio of uniforms method
Histogram of the distribution
predicted exponential deviate