PHYS512_HW7

November 18, 2022

[1]:
```python
import numpy as np
import math
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib as mpl
mpl.style.use('seaborn')
import random
```

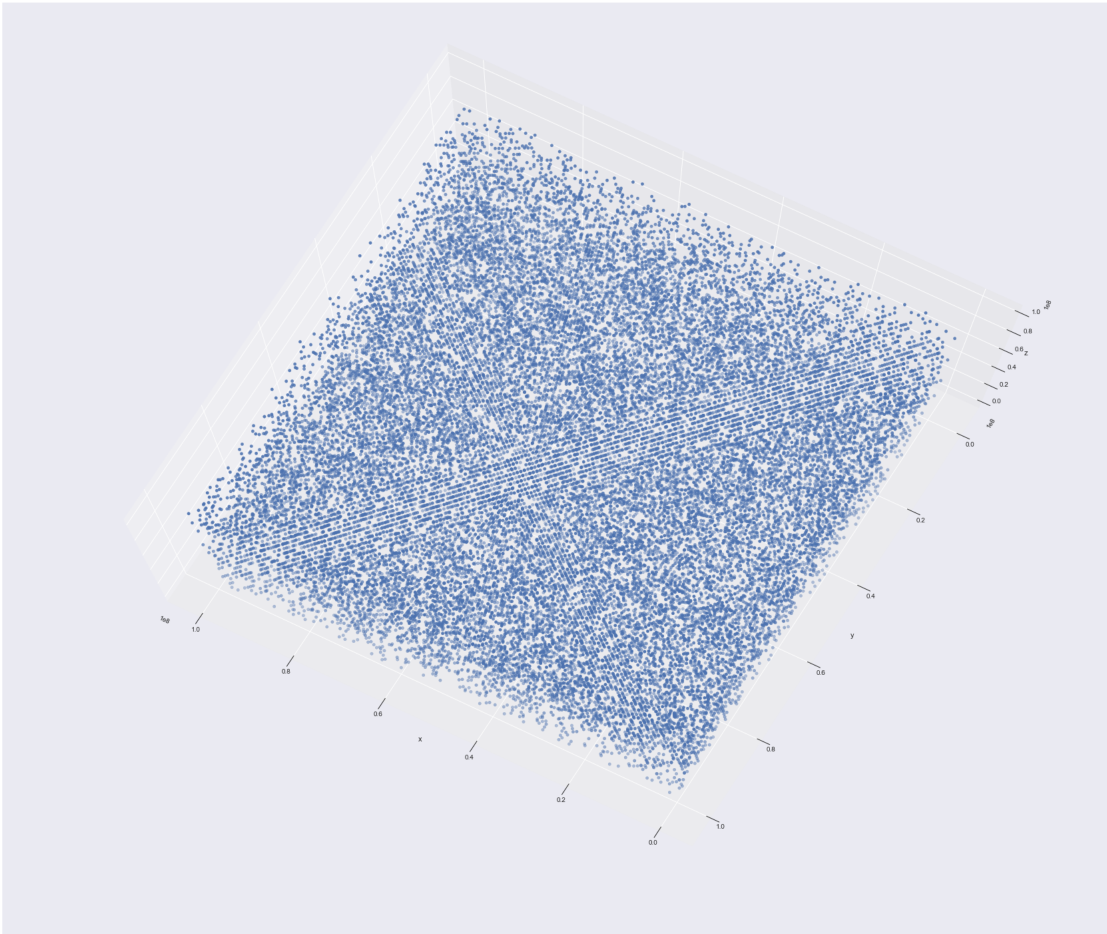
1 Problem 1

[2]:
```python
data = np.loadtxt('/Users/junalexsugiyama/Desktop/PHYS512/repo/phys512-2022/
 ↪problem_sets/rand_points.txt')

x = data[:, 0]
y = data[:, 1]
z = data[:, 2]
```

[ ]:

[60]:
```python
fig = plt.figure(figsize = (30, 26))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# # All of those angles show some sort of structure existing in the "random"␣
 ↪coordinates
elev_l = [-42.0, -42.0,-40, 60.0, 90 ]
azim_l = [105.5, 106.5, 100.5, 100.5, 100]
elev = 80
azim = 119
ax.view_init(elev, azim)

plt.show()
```

```
print('ax.azim {}'.format(ax.azim))
print('ax.elev {}'.format(ax.elev))
```



```
ax.azim 119
ax.elev 80
```

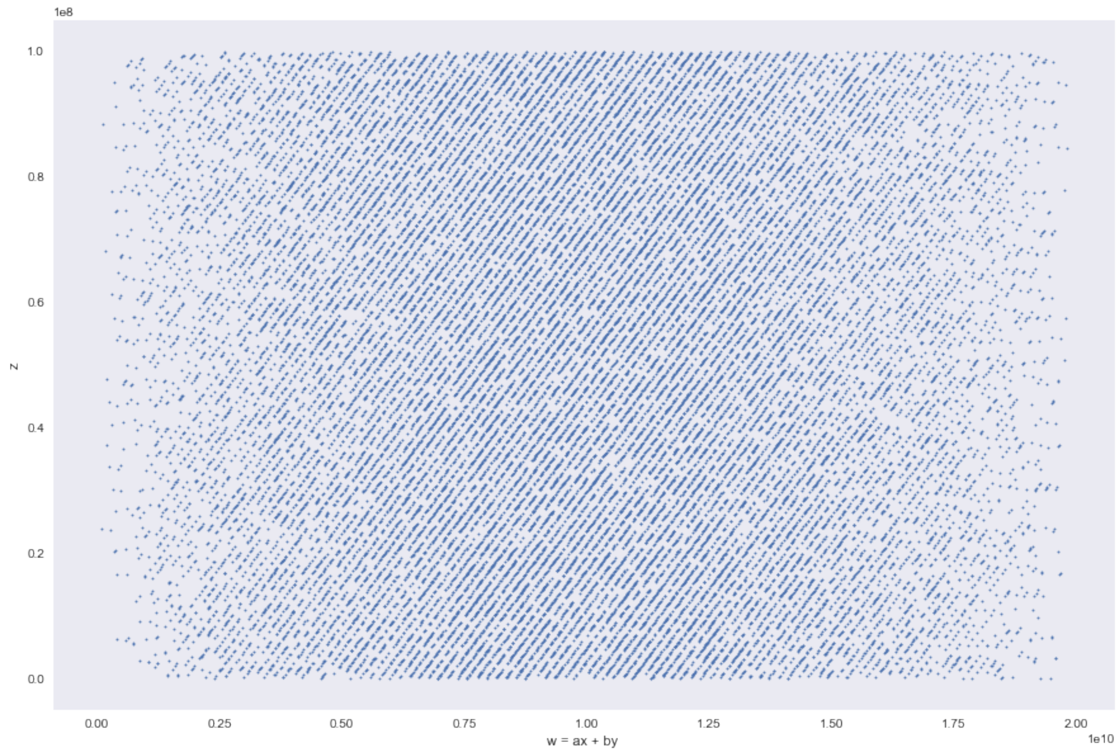
[ ]:

```

```

[77]:

```
# I just kept plugging in random numbers until i saw the pattern appear.
#However, i took and inspiration from elevation and azimuth values in 3D plot
a = 80
b = 119

w = a * x + b * y

plt.figure(figsize = (15, 10))
plt.scatter(w, z, s=2)
plt.xlabel('w = ax + by ')
```

```
plt.ylabel('z')
plt.grid(b=None)
```



[ ]:

```

```

[ ]:

```

```

2 Problem 2

[23]:

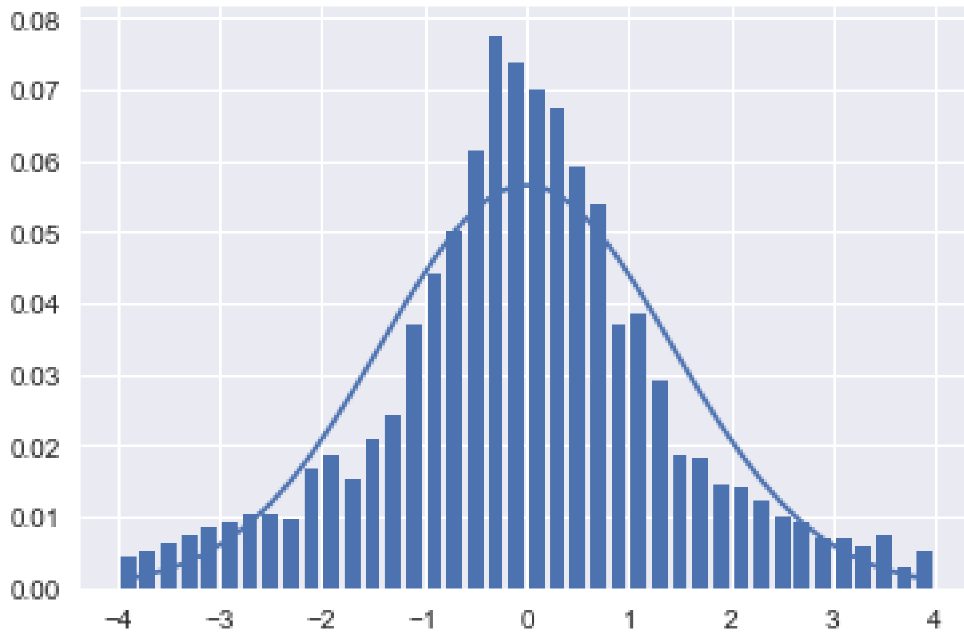
```
def gauss(x):
    return 1/np.sqrt(2 * np.pi) * np.exp(-0.5 * x**2/2)

def lorentz(x):
    return 1/(np.pi * (1 + x**2))

def power(x):
    return x**(-1)

n=10000
yy=np.pi*(np.random.rand(n)-0.5)

M = 0.1
```

```
u1=np.pi * (np.random.rand(n)*3 - 0.5)
u2=np.pi * (np.random.rand(n) - 0.5)

idx = np.zeros([3, len(u1)])
funcs = ['gauss', 'lorentz', 'power']


idx_g, = np.where(u2<=gauss(u1)/max(gauss(u1)))
idx_l, = np.where(u2<=lorentz(u1)/max(lorentz(u1)))
idx_p, = np.where(u2<=power(u1)/max(power(u1)))

f_g = u1[idx_g]
f_l = u1[idx_l]
f_p = u1[idx_p]

ef_g = f_g.size/u1.size
ef_l = f_l.size/u1.size
ef_p = f_p.size/u1.size

print('Efficiency of Gaussian as bounding distribution: ', ef_g)
print('Efficiency of Lorentzian as bounding distribution: ', ef_l)
print('Efficiency of Power Law as bounding distribution: ',ef_p)
```

```
Efficiency of Gaussian as bounding distribution:  0.6054
Efficiency of Lorentzian as bounding distribution:  0.581
Efficiency of Power Law as bounding distribution:  0.4963
```

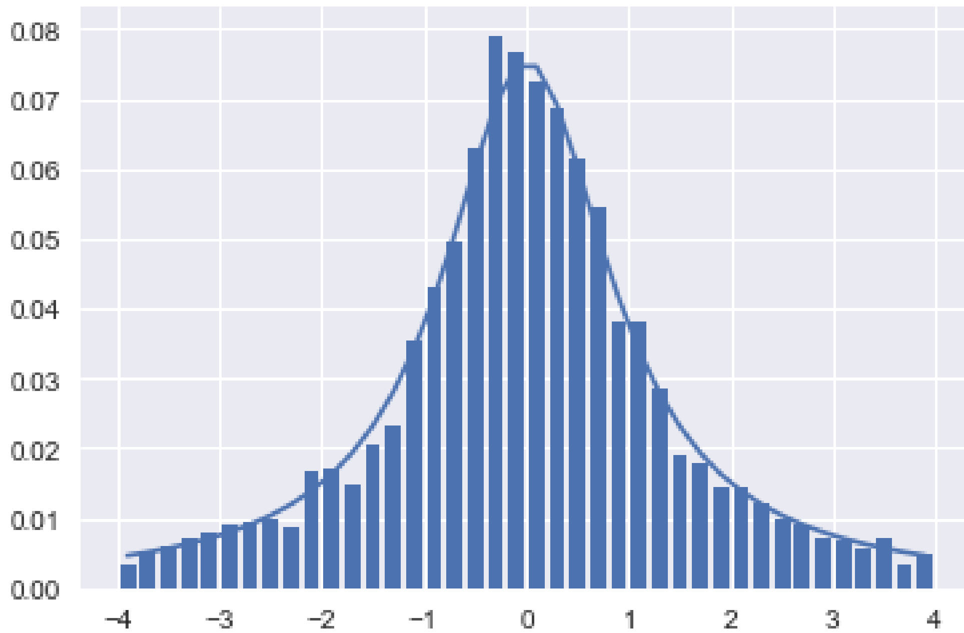
[24]:

```
#Gaussian
y_use=f_g
x_use=np.tan(y_use)
aa,bb=np.histogram(x_use,np.linspace(-4,4,41))
b_cent=0.5*(bb[1:]+bb[:-1])
pred=gauss(b_cent)
pred=pred/pred.sum()
aa=aa/aa.sum()
plt.figure(2)
plt.clf()
plt.plot(b_cent,pred)
plt.bar(b_cent,aa,0.15)
plt.show()
```



[25]:

```
#Lorentzian
y_use=f_l
x_use=np.tan(y_use)
aa,bb=np.histogram(x_use,np.linspace(-4,4,41))
b_cent=0.5*(bb[1:]+bb[:-1])
pred=lorentz(b_cent)
pred=pred/pred.sum()
aa=aa/aa.sum()
plt.figure(2)
plt.clf()
plt.plot(b_cent,pred)
plt.bar(b_cent,aa,0.15)
plt.show()
```


0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−4
−3
−2
−1
0
1
2
3
4

[ ]: