AST4320 – Oblig 2

Jonas Gahr Sturtzel Lunde (jonassl)

October 14, 2019

EXERCISE 1

$$W(x) = \begin{cases} 1, & |x| < R \\ 0, & \text{else} \end{cases}$$

$$\tilde{W}(x) = \int_{-\infty}^{\infty} W(x) e^{-ikx} dx$$

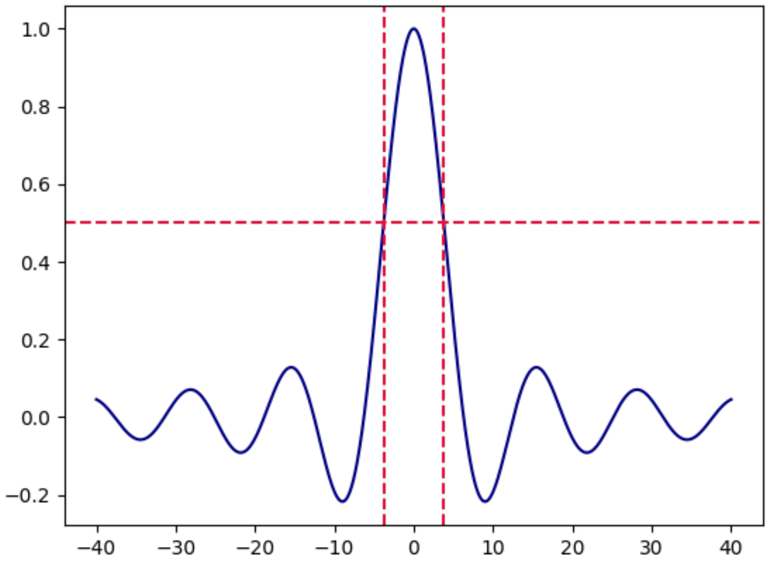
$$= \int_{-\infty}^{-R} 0 \cdot e^{-ikx} dx + \int_{-R}^{R} 1 \cdot e^{-ikx} dx + \int_{R}^{\infty} 0 \cdot e^{-ikx}$$

$$= \int_{-R}^{R} e^{-ikx} dx = \left[\frac{i}{k} e^{-ikx}\right]_{-R}^{R} = \frac{i}{k}\left(e^{-ikR} - e^{ikR}\right)$$

$$= \frac{2}{k} \sin(Rk)$$

For small k, we have that $\sin(Rk) \approx k$, $k << 1$, meaning that

$$\tilde{W}(0) = 2R$$


1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−40
−30
−20
−10
0
10
20
30
40

Exercise 2

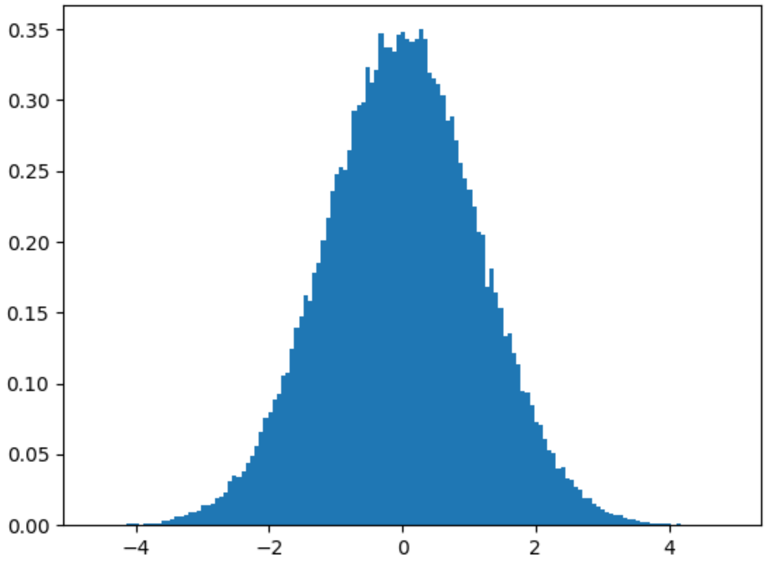


Figure 1: Histogram of delta values after random walks.

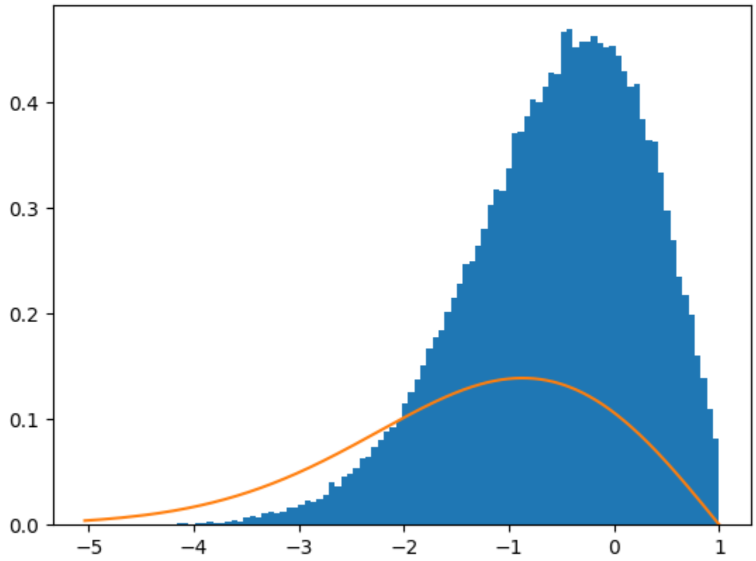


Figure 2: Histogram of delta values after random walk with cutoff for value that exceed 1 at some point. Also shown as orange plot: Analytical plot.

EXERCISE 3

(1) $\int_{-\infty}^{\delta_{crit}} d\delta \, P_{nc}(\delta|M)$ denotes the probability that a mass is not larger than δ_{crit} and therefore not part of a collapsed object.

$1 - \int_{-\infty}^{\delta_{crit}} d\delta \, P_{nc}(\delta|M)$ therefore denotes the probability that a perturbation is part of a collapsed object

(2)

Appendix: Code

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

N = 10000
x = np.linspace(-2, 2, N)
R = 0.5

W = np.where(np.abs(x) < R, 1, 0)

W = W

plt.plot(W)
plt.show()

k = np.linspace(-40, 40, 10000)

W_f = 2/k*np.sin(R*k)
Wmax = 2*R
half_idx = np.argmin(np.abs(Wmax/2 - W_f))

plt.plot(k, W_f, c="navy")
plt.axvline(x=k[half_idx], ls="--", c="crimson")
plt.axvline(x=-k[half_idx], ls="--", c="crimson")
plt.axhline(y=Wmax/2, ls="--", c="crimson")
plt.savefig("tophat.png")
```

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import trange

M = 100000
deltas = np.zeros(M)

for j in trange(M):
    var = 1e-4
    Sc = (np.pi/var)**(1.0/4)
    delta = np.random.normal(0, np.sqrt(var))
    N = 101
    Sc_array = np.linspace(Sc, 1, N)
    epsilon = Sc_array[-1] - Sc_array[-2]
    delta_array = np.zeros(N)
    delta_array[0] = delta

    for i in range(1, N):
        Sc = Sc_array[i]
        var2 = (np.pi/Sc)**(1.0/4)
        beta = np.random.normal(0, np.sqrt(var2 - var))
        delta_array[i] = delta_array[i-1] + beta
        var = var2

    deltas[j] = delta_array[-1]

plt.hist(deltas, bins="auto", density="true")
plt.savefig("hist1.png")
```

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import trange

```

```
M = 100000
deltas = []

for j in trange(M):
    var = 1e-4
    Sc = (np.pi/var)**(1.0/4)
    delta = np.random.normal(0, np.sqrt(var))
    N = 101
    Sc_array = np.linspace(Sc, 1, N)
    epsilon = Sc_array[-1] - Sc_array[-2]
    delta_array = np.zeros(N)
    delta_array[0] = delta

    for i in range(1, N):
        Sc = Sc_array[i]
        var2 = (np.pi/Sc)**(1.0/4)
        beta = np.random.normal(0, np.sqrt(var2 - var))
        delta_array[i] = delta_array[i-1] + beta
        var = var2

    if not (delta_array > 1).any():
        deltas.append(delta_array[-1])

deltas = np.array(deltas)

def Pnc(delta):
    var = np.pi
    return 1/np.sqrt(2*np.pi*var)*(np.exp(-delta**2/(2*var)) - np.exp(-(2*1-delta)**2/(2*v

plt.hist(deltas, bins="auto", density="true")
plt.plot(np.sort(deltas), Pnc(np.sort(deltas)))
plt.savefig("hist2.png")
```