# Assignment 2 - AST4320

Maren Rasmussen

October 3, 2018

## Exercise 1

If we consider a top-hat smoothing function

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

the Fourier conjugate can be given by

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

Since $W(x)$ is only non-zero when $|x| < R$, we can change the limits in the integral, insert for $W$ and get that

$$\begin{aligned}
\tilde{W}(k) &= \int_{-R}^{R} \exp(-ikx) \cdot 1 \cdot dx \\
&= -\frac{1}{ik} \Big[ \exp(-ikx) \Big]_{-R}^{R} \\
&= -\frac{1}{ik} \exp(-ikR) + \frac{1}{ik} \exp(ikR) \\
&= \frac{2}{k} \left[ \frac{1}{2i} \left( e^{ikR} - e^{-ikR} \right) \right] \\
&= \frac{2}{k} \sin(kR)
\end{aligned}$$

Figure 1 shows the top-hat smoothing function and figure 2 shows the Fourier conjugate of the top-hat smoothing function both with $R = 1$. The full width at half maximum (FWHM) of the Fourier conjugate is 3.8, meaning that the width of the peak when it is at half its maximum is 3.8. The program code making the plots and calculates the FWHM can be found in appendix A.1

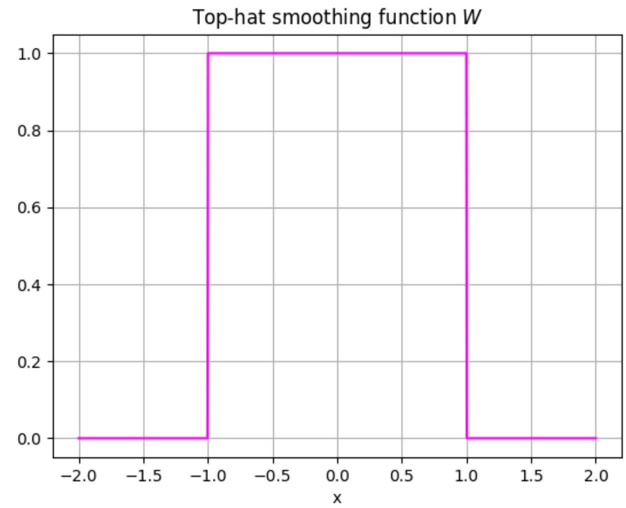


Figure 1: Plot of the top-hat smoothing function $W$ for $R = 1$ with respect to $x$.

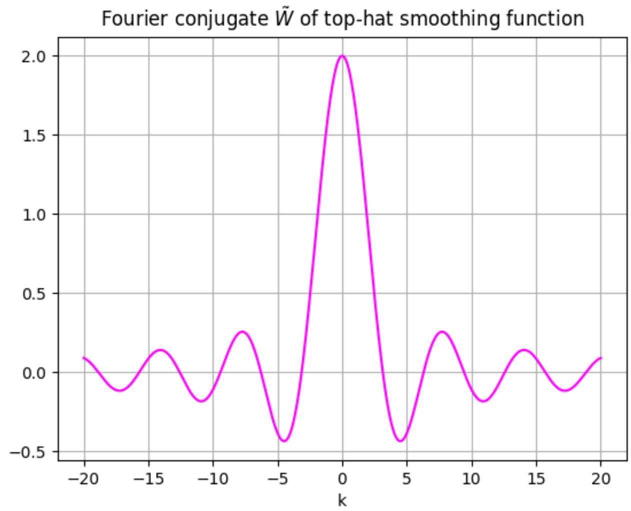


Figure 2: Plot of the Fourier conjugate $\tilde{W}$ of the top-hat smoothing function for $R = 1$ with respect to $j$.

## Exercise 2

In this exercise random walk is studied. We have assumed that we have a power spectrum on the form $P(k) = k$, a smoothing scale of $S_c = 2\pi/k$ and a variance given as a function of the smoothing scale $\sigma^2(S_c) = \pi/S_c^4$. We start by requiring that the first variance as function of the first scale $S_1$ is $\sigma^2(S_1) < 10^{-4}$, so I chose $\sigma^2(S_1) = 5 \cdot 10^{-5}$. The first point in your random walk is then $(S_1, \delta)$. The random walk algorithm is done in the following way:

- Reduce $S_c$ with a factor of $\epsilon$ and compute the new variance $\sigma^2(S_2)$
- Draw a random number $\beta$ from a Gaussian PDF with $\sigma_{12}^2 = \sigma^2(S_2) - \sigma^2(S_1)$. Then your new $\delta_2 = \delta + \beta$, and your next point is $(S_2, \delta_2)$.
- Continue to reduce $S_c$ with a factor of $\epsilon$ until you reach $S_c = 1.0$.

For a Gaussian random field, the overdensity $\delta$, smoothed with scale $S_c$ corresponding to mass scale $M$, is Gaussian distributed with variance $\sigma^2(M)$:

$$P(\delta|M) = \frac{1}{\sqrt{2\pi\sigma^2(M)}} \exp\left(-\frac{\delta^2}{2\sigma^2(M)}\right) \tag{1}$$

Figure 3 shows this Gaussian function plotted together with the histogram of $\delta$-values gotten from the random walk for $S_c = 1.0$. The random walk was then done $N = 10^5$ times with an $\epsilon = 0.01$.

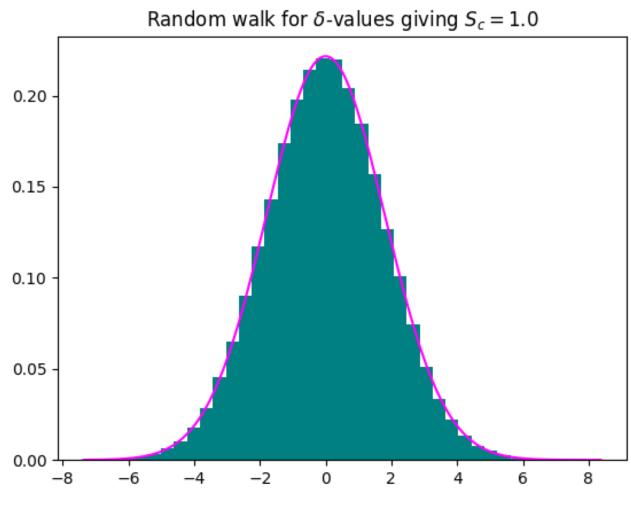


Figure 3: Histogram of $\delta$-values gotten from random walk for $S_c = 1.0$ plotted together with the Gaussian distribution given in equation 1

If we only consider the $\delta$-values never crossing the threshold $\delta_{\text{crit}} = 1.0$, we should get a Gaussian function on the form

$$
\begin{aligned}
P_{\rm nc}(\delta|M) &= P(\delta|M) - P([2\delta_{\rm crit} - \delta]|M) \\
&= \frac{1}{\sqrt{2\pi}\sigma(M)}\left(\exp\left(-\frac{\delta^2}{2\sigma^2(M)}\right) - \exp\left(-\frac{(2\delta_{\rm crit}-\delta)^2}{2\sigma^2(M)}\right)\right)
\end{aligned} \tag{2}
$$

Figure 4 shows the histogram of the $\delta$-values that never crosses the threshold $\delta_{\rm crit} = 1$. I am not sure why the Gaussian curve not matches the histogram of the $\delta$-values. The program code making these plots can by found in appendix A.2

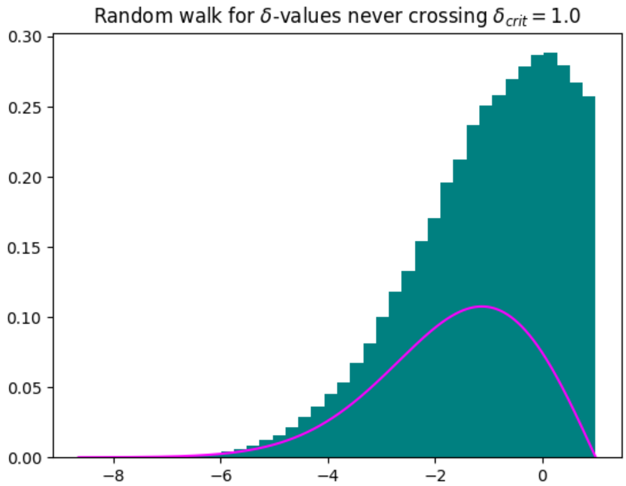


Figure 4: Histogram of $\delta$-values never crossing the treshold $\delta_{\rm crit} = 1.0$ gotten from random walk for $S_c = 1.0$ plotted together with the Gaussian distribution given in equation 2

3.

$$P_{nc}(\delta|M) = P(\delta|M) - P([2\delta_{crit} - \delta]|M)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2(M)}}\left(\exp\left[\frac{-\delta^2}{2\sigma^2(M)}\right] - \exp\left[-\frac{(2\delta_{crit}-\delta)^2}{2\sigma^2(M)}\right]\right)$$

a) $P(>M) = P(\delta > \delta_{crit}|M) = 1 - P(\delta < \delta_{crit}|M)$

$= 1 - P_{nc}(\delta|M)$

The probability can then be found by

$$P(>M) = 1 - \int_{-\infty}^{\delta_{crit}} d\delta\, P_{nc}(\delta|M).$$

b) $P(>M) = 1 - \int_{-\infty}^{\delta_{crit}} P_{nc}(\delta|M)\, d\delta$

$$= \int_{-\infty}^{\delta_{crit}} \frac{1}{\sqrt{2\pi\sigma^2}}\left[\exp\left(-\frac{\delta^2}{2\sigma^2}\right) - \exp\left(-\frac{(2\delta_{crit}-\delta)^2}{2\sigma^2}\right)\right]d\delta$$

To make the calculations easier, we can calculate the integrals one by one:

$$\int_{-\infty}^{\delta_{crit}} \exp\left(-\frac{\delta^2}{2\sigma^2}\right) d\delta$$

$$= \int_{-\infty}^{0} \exp\left(-\frac{\delta^2}{2\sigma^2}\right) d\delta + \int_{0}^{\delta_{crit}} \exp\left(-\frac{\delta^2}{2\sigma^2}\right) d\delta$$

Substitute: $\delta_* = \delta/\sqrt{2}\sigma$

$$\frac{d\delta_*}{d\delta} = \frac{1}{\sqrt{2}\sigma} \Rightarrow d\delta = \sqrt{2}\sigma\, d\delta$$

$$= \int_{-\infty}^{0} \sqrt{2}\,\sigma \exp(-\delta_*^2)\, d\delta_* + \int_{0}^{\delta_{crit}/\sqrt{2}\sigma} \sqrt{2}\,\sigma \exp(-\delta_*^2)\, d\delta_*$$

We know that

$$\int_{-\infty}^{0} e^{-x^2}\, dx = \frac{\sqrt{\pi}}{2}$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

giving:

$$= \sqrt{2}\,\sigma \frac{\sqrt{\pi}}{2} + \sqrt{2}\,\sigma \frac{\sqrt{\pi}}{2} \text{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\,\sigma}\right)$$

$$= \underline{\frac{\sigma\sqrt{\pi}}{\sqrt{2}} \left(1 + \text{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\,\sigma}\right)\right)}$$

The next integral can be calculated similarly:

$$\int_{-\infty}^{\delta_{crit}} \exp\left(-\frac{(2\delta_{crit} - \delta)^2}{2\sigma^2}\right) d\delta$$

$$= \int_{-\infty}^{0} \exp\left(-\frac{(2\delta_{crit} - \delta)^2}{2\sigma^2}\right) d\delta + \int_{0}^{\delta_{crit}} \exp\left(-\frac{(2\delta_{crit} - \delta)^2}{2\sigma^2}\right) d\delta$$

Substitute: $\delta_* = \dfrac{2\delta_{crit} - \delta}{\sqrt{2}\sigma}$

$$\frac{d\delta_*}{d\delta} = -\frac{1}{\sqrt{2}\sigma} \Rightarrow d\delta = -\sqrt{2}\,\sigma\, d\delta_*$$

$$= \int_{\frac{2\delta_{crit} - (-\infty)}{\sqrt{2}\sigma}}^{0} \exp(-\delta_*^2)(-\sqrt{2}\sigma)\, d\delta_* + \int_{0}^{\delta_{crit}/\sqrt{2}\sigma} (-\sqrt{2}\sigma)\exp(-\delta_*^2)\, d\delta_*$$

$$= \int_{\infty}^{0} \exp(-\delta_*^2)(-\sqrt{2}\sigma)\, d\delta_* + \int_{0}^{\delta_{crit}/\sqrt{2}\sigma} (-\sqrt{2}\sigma)\exp(-\delta_*^2)\, d\delta_*$$

$$= -\int_{0}^{\infty} \exp(-\delta_*^2)(-\sqrt{\pi}\sigma)\, d\delta_* - \sqrt{2}\sigma \frac{\sqrt{\pi}}{2} \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right)$$

$$= \sqrt{2}\sigma \frac{\sqrt{\pi}}{2} - \sqrt{\pi}\sigma \frac{\sqrt{\pi}}{2} \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right)$$

$$= \underline{\frac{\sigma\sqrt{\pi}}{\sqrt{2}}\left(1 - \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right)\right)}$$

Inserting back to the original expression we get:

$$P(>M) = 1 - \frac{1}{\sqrt{2\pi}\sigma} \frac{\sigma\sqrt{\pi}}{\sqrt{2}} \left[1 + \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right) - 1 + \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right)\right]$$

$$= 1 - \frac{2}{2} \operatorname{erf}\left(\frac{\delta_{crit}}{\sqrt{2}\sigma}\right)$$

Defining $\nu = \delta_{crit}/\sigma$ :

$$P(>M) = 1 - \operatorname{erf}\left(\frac{\nu}{\sqrt{2}}\right)$$

From the lectures we know that

$$P(\delta > \delta_{crit} \mid M) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\nu}{\sqrt{2}}\right)\right)$$

And therefore we can now see that we need the factor of 2.

# Appendix

## A.1

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

R = 1.0
i = np.sqrt(-1+0j)

def W(x):
    """
    Top-hat smoothing function.
    """
    return (abs(x) < R)*1.0

def W_tilde(k):
    """
    Fourier transformation of W.
    """
    #return -1.0/(i*k) * np.exp(-i*k*R) + 1.0/(i*k) * np.exp
        (i*k*R)
    return 2.0/k * np.sin(k*R)


def FWHM(x_values, f_values):
    """
    Calculates the full width at half maximum.
    """
    half_max = 0.5*np.max(f_values)
    x_fwhm = 0
    for i in range(len(x_values)):
        if f_values[i] < half_max:
            x_fwhm = x_values[i]
        else:
            break
    return (abs(x_fwhm)*2)


N = 1000
x = np.linspace(-2,2, N)
k = np.linspace(-20,20, N)

print("Full width at half maximum: %.2f" %(FWHM(k,W_tilde(k)
    )))

plt.figure()
plt.plot(x, W(x), color='magenta')
plt.title(r'Top-hat smoothing function $W$')
plt.xlabel('x')
plt.grid()

plt.figure()
plt.plot(k, W_tilde(k), color='magenta')
```

```
plt.title(r'Fourier conjugate $\tilde{W}$ of top-hat
    smoothing function')
plt.xlabel('k')
plt.grid()
plt.show()
```

### A.2

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


def random_walk():
    """
    Function for doing random walk by reducing S_c with a
    small number epsilon for each step.
    """
    # Creating initial values
    var = 0.5*1e-4
    S_c = [(np.pi/var)**(1.0/4)]
    delta = [np.random.normal(scale=np.sqrt(var))]
    eps = 0.1

    # Doing the random walk untill S_c reaches 1.
    i = 1
    while S_c[-1] >= 1.0:
        S_c.append(S_c[i-1] - eps)
        var_new = np.pi/S_c[i]**4
        beta = np.random.normal(scale=(np.sqrt(var_new-var))
            )
        delta.append(delta[i-1] + beta)
        var = var_new
        i+=1

    return delta, S_c


def gauss_dist(delta, S_c):
    """
    Gaussian distribution for overdensity smoothed with
    scale S_c, corresponding to mass M, for variance
    sigma^2(M).
    """
    var = (np.pi/(S_c**4))
    return 1.0/(np.sqrt(2*np.pi*var))*np.exp(-delta**2/(2*
        var))


def gauss_dist_nc(delta, S_c):
    """
    Gaussian distribution for overdensity smoothed with
    scale S_c, corresponding to mass M, for variance
```

```
    sigma^2(M), when the chains never cross delta_crit=1.0.
    """
    var = (np.pi/(S_c**4))
    delta_crit = 1.0
    return 1.0/(np.sqrt(2*np.pi*var))*(np.exp(-delta**2/(2*
        var)) - \
    np.exp(- (2*delta_crit - delta)**2/(2*var)))



deltas = []
S_cs = []
deltas_nc = []
S_cs_nc = []

# Doing the random walk 10^5 times and saving the last
    values
N = int(1e5)
for j in range(N):
    delta, S_c = random_walk()
    deltas.append(delta[-1])
    S_cs.append(S_c[-1])
    if delta[-1] < 1:
        deltas_nc.append(delta[-1])
        S_cs_nc.append(S_c[-1])


# Creating array with delta values between the max and
# min delta from the random walk.
delta_array = np.linspace(np.min(deltas),np.max(deltas),len(
    deltas))
delta_nc_array = np.linspace(np.min(deltas_nc),np.max(
    deltas_nc),len(deltas_nc))


plt.figure()
plt.title(r'Random walk for $\delta$-values giving $S_c =
    1.0$')
plt.hist(deltas, bins=40, density = True, color='teal')
plt.plot(delta_array, gauss_dist(delta_array, np.array(S_cs)
    ), color='magenta')

plt.figure()
plt.title(r'Random walk for $\delta$-values never crossing $
    \delta_{crit} = 1.0$')
plt.hist(deltas_nc, bins=40, density = True, color='teal')
plt.plot(delta_nc_array, gauss_dist_nc(delta_nc_array, np.
    array(S_cs_nc)), color='magenta')
plt.show()
```