**Univerzitet u Sarajevu**

**Elektrotehnički fakultet**

**Multimedijalni Sistemi**

# Laboratorijska vježba 6. - Fourierova Transformacija

**Izrada laboratorijske vježbe vrši se u ovom *Jupyter Notebook*-u. Isti je potrebno konvertovati u PDF dokument i predati na Zamger.**

**Ime i prezime studenta, broj indeksa:**

**Amila Laković**

**Datum izrade izvještaja:**

**06.05.2021.**

---

## Zadatak 1.

**Potrebno je implementirati funkciju `fourier_transform` koja vrši izračunavanje Fourierove transformaciju za funkciju proslijeđenu kao parametar. Osim funkcije, kao parametar se prosljeđuje i segment na kojem je potrebno izračunati vrijednost Fourierove transformacije (početak, kraj segmenta i veličina koraka). Ova funkcija kao rezultat treba vratiti tri vrijednosti: W, F i Theta.**

**Fourierova transformacija izračunava se prema sljedećoj formuli:**

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt.$$

**Potrebno je voditi računa o sljedećim stvarima:**

- `t` **je varijabla po kojoj se vrši integriranje. Ova varijabla predstavlja x varijablu za koju se vrši izračunavanje vrijednosti funkcije f prije prelaska u frekvencijski domen. Njoj nije potrebno dodijeliti vrijednost, već je definisati kao ulaznu varijablu lambda izraza pri integriranju.**
- `w` **je varijabla koja predstavlja x frekvencijske domene. Ovoj varijabli potrebno je dodijeliti sve x vrijednosti iz segmenta (od početka do kraja segmenta sa pomijeranjem za veličinu koraka).**
- `j` **je imaginarni dio kompleksnog broja. Funkcija** `quad` **iz prethodne vježbe ne može odjednom vršiti integriranje i za realni, i za imaginarni dio kompleksnog broja. Iz tog razloga potrebno je razdvojiti izraz na dva integrala sa realnim i imaginarnim dijelom.**
- `F` **i** `Theta` **su vrijednosti koje se dobivaju po sljedećim formulama (Fr i Fi su realni i imaginarni dio rezultne vrijednosti dobivene integriranjem):**

$$|F(\omega)| = \sqrt{F_r^2 + F_i^2},$$

$$\theta(\omega) = \arctan \frac{F_i}{F_r}.$$

**Detaljne informacije o Fourierovoj transformaciji moguće je pronaći u Poglavlju 3 materijala za rad na predmetu, s početkom na str. 69.**

**U nastavku je data funkcija za koju je potrebno izvršiti Fourierovu transformaciju:**

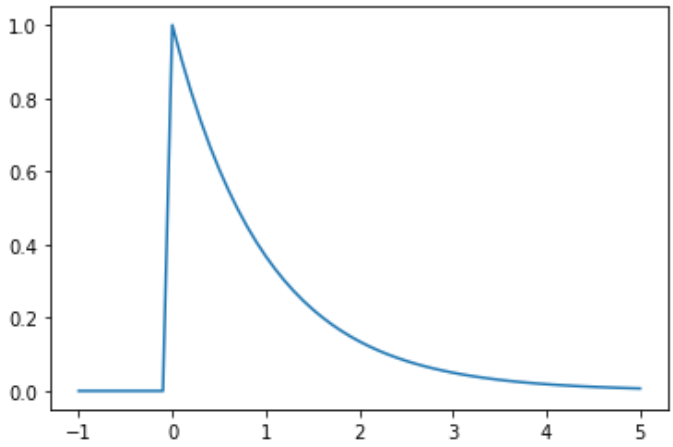
$$f(t) = \begin{cases} e^{-t} & \text{za } t \geq 0, \\ 0 & \text{za } t < 0. \end{cases}$$

In [1]:

```
import math
import matplotlib.pyplot as plt

def f(x):
    if x < -0.01:
        return 0
    else:
        return math.exp(-x)

x = []
y = []
i = -1

while i < 5:
    x.append(i)
    y.append(f(i))
    i += 0.1

plt.plot(x, y)
```

Out[1]:

```
[<matplotlib.lines.Line2D at 0x7ff7d24c3eb0>]
```




**Rješenje:**

In [2]:

```
import scipy.integrate as integrate
```

```
import numpy as np

def fourier_transform(f, pocetak_segmenta, kraj_segmenta, step):
    w = []
    F = []
    Theta = []
    integrand = lambda trig_fun, w: lambda t: f(t) * trig_fun(w * t)

    for x in np.arange(pocetak_segmenta, kraj_segmenta, step):
        w.append(x)

        #realni dio - fcosdt
        realni_dio = integrate.quad(integrand(np.cos, x), -np.Inf, np.Inf)[0]

        #imaginarni dio - fsindt
        imaginarni_dio = -integrate.quad(integrand(np.sin, x), -np.Inf, np.Inf)[0]

        #formula za F(w)
        F.append(np.sqrt(realni_dio * realni_dio + imaginarni_dio * imaginarni_dio))

        #formula za Theta
        value = imaginarni_dio/realni_dio
        Theta_rjesenje = np.arctan(value)
        Theta.append(Theta_rjesenje)

    return w, F, Theta
```

**Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je dodati još primjera kako bi se implementacija adekvatno testirala.**

In [3]:

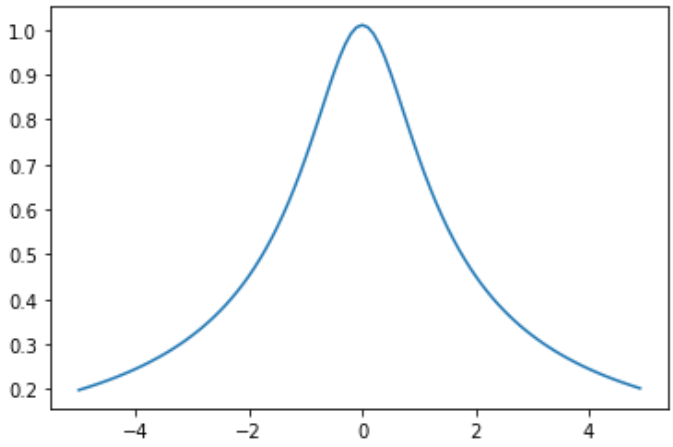
```
[W, F, Theta] = fourier_transform(f, -5, 5, 0.1)
```

In [4]:

```
plt.plot(W, F)
```

Out[4]:

```
[<matplotlib.lines.Line2D at 0x7ff7d2b9e4f0>]
```

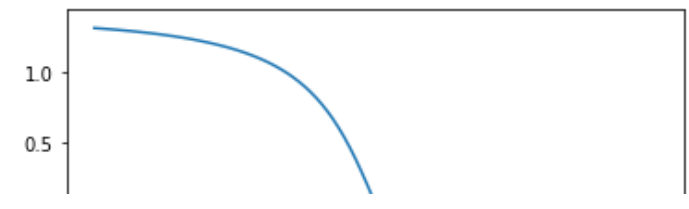


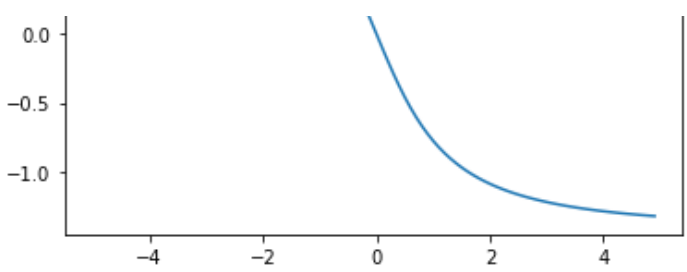

In [5]:

```
plt.plot(W, Theta)
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x7ff7d2c050a0>]
```



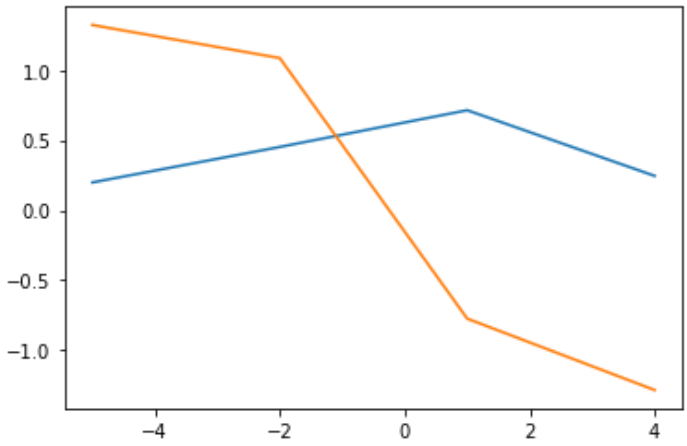



In [6]:

```
[W, F, Theta] = fourier_transform(f, -5, 5, 3)
plt.plot(W, F)
plt.plot(W, Theta)
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x7ff7d2d691c0>]
```

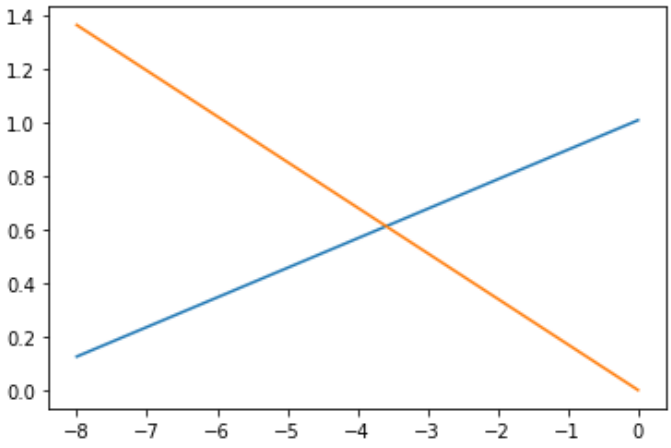



In [7]:

```
[W, F, Theta] = fourier_transform(f, -8, 8, 8)
plt.plot(W, F)
plt.plot(W, Theta)
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x7ff7d2e4a4c0>]
```



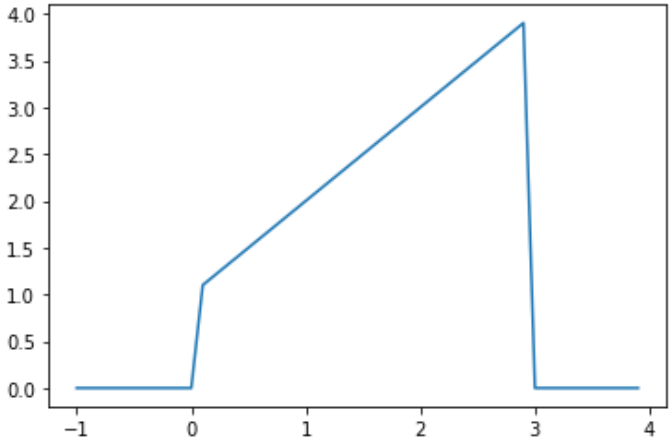

In [8]:

```
import math
import matplotlib.pyplot as plt

def f2(x):
    if x <= 0:
        return 0
    elif x < 3 :
        return x + 1
    else:
```

```
        return 0

x = []
y = []
i = -1

while i < 4:
    x.append(i)
    y.append(f2(i))
    i += 0.1

plt.plot(x, y)
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x7ff7d2ed4e50>]
```




In [9]:

```
[W, F, Theta] = fourier_transform(f2, -4, 4, 0.1)
plt.plot(W, F)
plt.plot(W, Theta)
```

Out[9]:

```
[<matplotlib.lines.Line2D at 0x7ff7d301f130>]
```