

Projekt Ray Launcher w moim pokoju

Konrad Kotlarz 147974

1. Wstęp teoretyczny.

Ray tracing w dziedzinie fal radiowych jest podobny do ray tracingu w grafice komputerowej, ale jest używany do modelowania propagacji fal radiowych w różnych środowiskach. Polega on na symulowaniu i analizowaniu tras, po których rozchodzą się fale radiowe w danym środowisku. Ray tracing jest powszechnie stosowany w projektowaniu anten i systemów komunikacyjnych, pomagając przewidywać, jak fale radiowe będą się rozchodzić w różnych środowiskach i jak będą one wpływać na jakość sygnału.

Ray tracing będę używał do symulacji rozchodzenia się fali radiowej w pokoju.

Wzór Friisa na tłumienie ścieżki:

$$\frac{P_r}{P_t} = G_T G_R \left| \frac{\lambda}{4\pi d} \right|^2 \quad (1)$$

gdzie:

d - odległość nadajnika od odbiornika

$G_T G_R$ - zysk anteny

λ - długość fali w próżni [m]

P_r - moc odbiorcza

P_t - moc nadawca

Założenia parametryczne:

Przyjąłem rozmiar pokoju według proporcji tzw. złotego podziału, tzn. boki pokoju mają długość a , b , gdzie $(a+b)/a = a/b$. Na środku krótszej ściany (o długości a) zamontowałem punkt dostępowy transmitujący dla częstotliwości 5 GHz. Przyjąłem charakterystykę anteny nadawczej i odbiorczej jako dookólną o zysku 1 (w skali liniowej). Minimalna wartość boku a to 12 m.

Stałe:

$c = 299792458$ m/s

$f = 5$ GHz

$G_T G_R = 1$

$a = 13$

$b = 21$

gdzie:

c - prędkość fali w próżni

f - częstotliwość fali nośnej

a - długość krótszej ściany

b - długość dłuższej ściany

$$\lambda = \frac{c}{f} = 5.996 \text{ cm}$$

Poniżej są założenia parametryczne zaimplementowane w coolab.

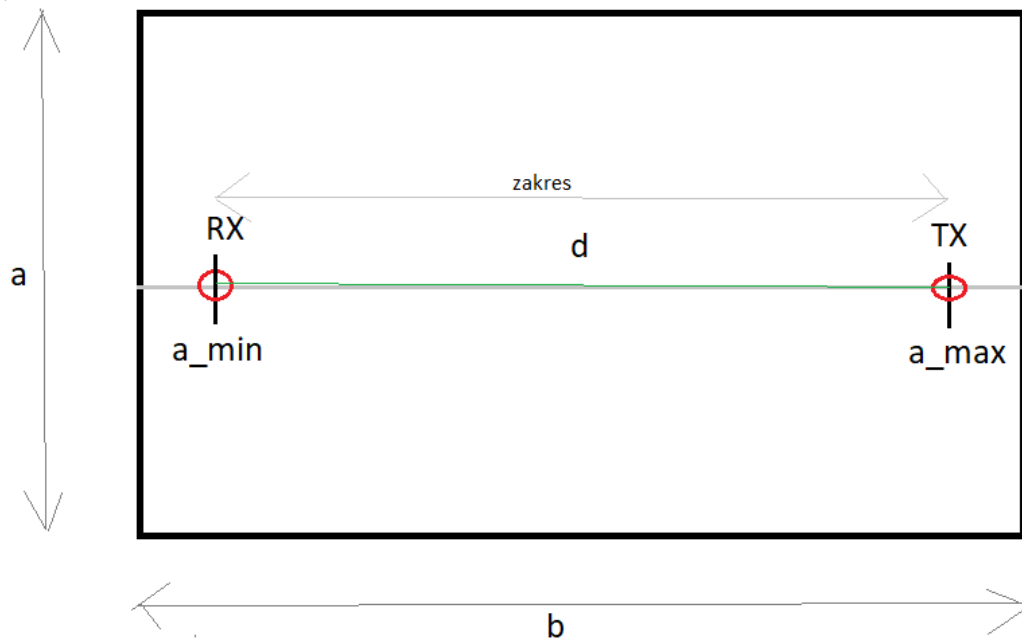
```
import numpy as np
import math
import matplotlib.pyplot as plt
```

```
#parametry
a = 13 # krótsza ściana
b = 21 # dłuższa ściana
f = 5 *10**9 # częstotliwość fali radiowej w GHz
lp = 500 # liczba punktów na wykresie
c = 299792458 #m/s
```

A.1 Wyznaczyłem wykres tłumienia ścieżki dla odbiornika poruszającego się w osi pokoju (osi równoległej do dłuższej ściany, przecinającej punkt położenia anteny)

```
k = 2 # odległość anteny nadawczej i odbiorczej od krótszej ściany w metrach (stała)
```

```
dmin = k #zakres odleglosci anten od (do zadania 1 i 2)
dmax = b-k # zakres odległości anten do (do zadania 1 i 2)
```



Powyżej schemat zbudowanego pokoju dla punktu A1.

Do wyznaczenia tłumienia ścieżki skorzystałam ze wzoru Frissa.

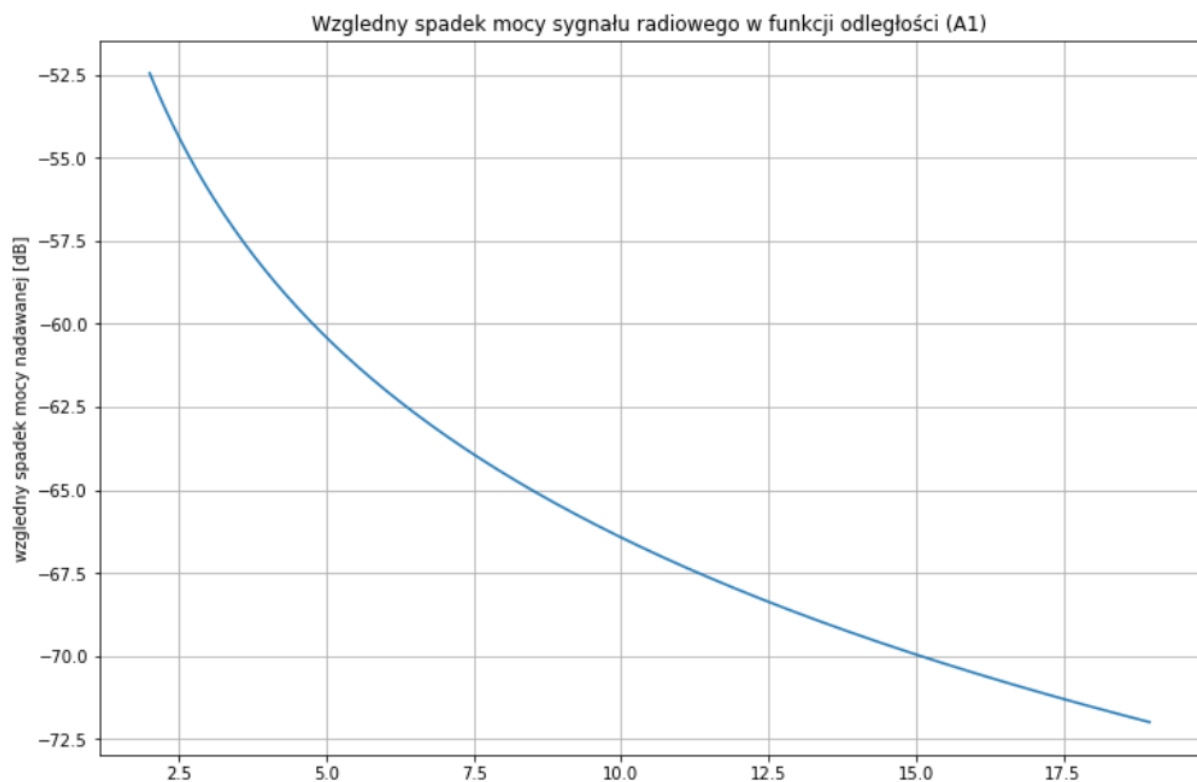
$$\frac{P_r}{P_t} = G_T G_R \left| \frac{\lambda}{4\pi d} \right|^2$$

```
#code A1
lamb = c/f # obliczenie długości fali w próżni
zakres=dmax - dmin #zakres ruchu odbiornika w osi pokoju
#
d=np.arange(dmin, dmax, zakres/(lp-1),dtype=float)#wektor ruchu
odbiornika
```

```
#
PrPt = 1*1*((lamb/(4*np.pi*d))**2) #wzór Frissa na tłumienie ścieżki

PrPt = np.abs(PrPt) # obliczenie modułu mocy
PrPt = 10*np.log10(PrPt) # zamiana na [dB]
```

```
#plot A1
plt.figure(figsize=(12,8)) # rozmiar okna wykresu
plt.title("Względny spadek mocy sygnału radiowego w funkcji odległości (A1)")
plt.xlabel("odległość [m]")
plt.ylabel("względny spadek mocy nadawanej [dB]")
plt.plot(d,PrPt) # wykres
plt.grid()
```



A.2 Wyznacz wykres tłumienia ścieżki, gdy istnieją także ścieżki odbite:

Do wyznaczenia współczynników odbicia „a_{wall}” skorzystałem z rekomendacji ITU-R P.2040-2.

Rec. ITU-R P.2040-2

23

TABLE 3
Material properties

Material class	Real part of relative permittivity		Conductivity S/m		Frequency range
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	GHz
Vacuum (\approx air)	1	0	0	0	0.001-100
Concrete	5.24	0	0.0462	0.7822	1-100
Brick	3.91	0	0.0238	0.16	1-40
Plasterboard	2.73	0	0.0085	0.9395	1-100
Wood	1.99	0	0.0047	1.0718	0.001-100
Glass	6.31	0	0.0036	1.3394	0.1-100
Glass	5.79	0	0.0004	1.658	220-450
Ceiling board	1.48	0	0.0011	1.0750	1-100
Ceiling board	1.52	0	0.0029	1.029	220-450
Chipboard	2.58	0	0.0217	0.7800	1-100
Plywood	2.71	0	0.33	0	1-40
Marble	7.074	0	0.0055	0.9262	1-60
Floorboard	3.66	0	0.0044	1.3515	50-100
Metal	1	0	10^7	0	1-100
Very dry ground	3	0	0.00015	2.52	1-10 only
Medium dry ground	15	-0.1	0.035	1.63	1-10 only
Wet ground	30	-0.4	0.15	1.30	1-10 only

Powyżej w tabeli przedstawione są współczynniki przenikalności elektrycznej dla różnych materiałów.

$$R_{ETM} = \frac{\eta \cos \theta - \sqrt{\eta - \sin^2 \theta}}{\eta \cos \theta + \sqrt{\eta - \sin^2 \theta}} \quad (2)$$

- 1D – od przeciwległej ściany

Ponieważ fala pada na przeciwległą ścianę pod kątem 90 stopni wzór do wyznaczenia współczynnika odbicia uprasza się odpowiednio do:

$$ReTM = \frac{(er_{wall} \cos(0) - \sqrt{er_{wall} - \sin(0)^2})}{er_{wall} \cos(0) + \sqrt{er_{wall} - \sin(0)^2}} = a_{wall} \quad (3)$$

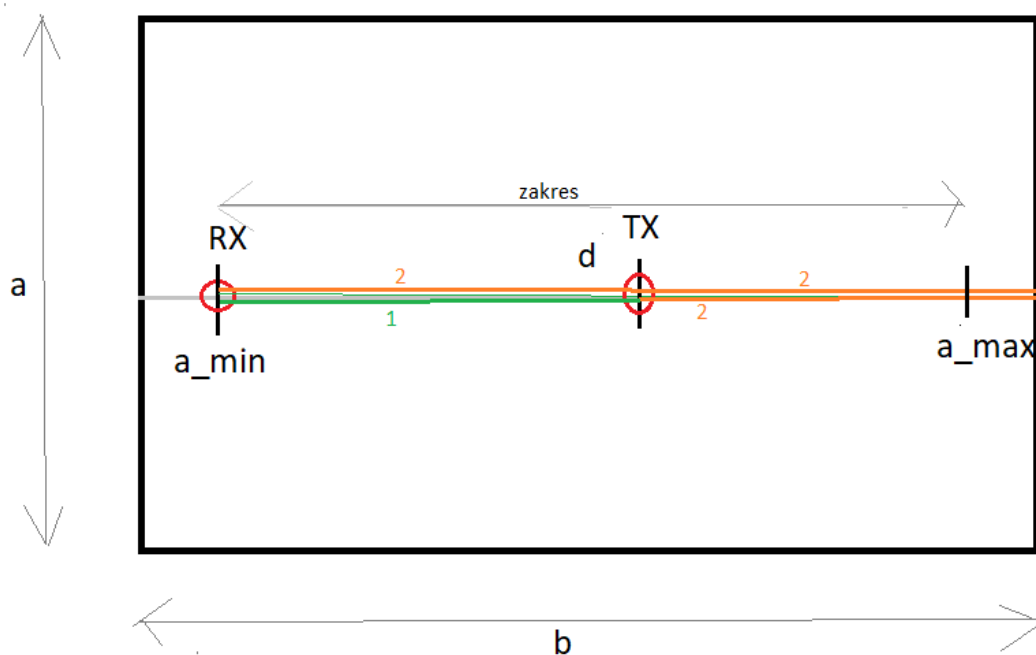
Wyprowadzenia wzorów:

$$\frac{P_R}{P_t} = \left| \frac{1}{d} e^{-j\varphi_1} + \frac{a_{wall}}{2b-d} e^{-j\varphi_2} \right|^2$$

$$\varphi_1 = 2\pi \frac{d}{c}$$

$$\varphi_2 = 2\pi \frac{2b-d}{c}$$

$$\frac{P_R}{P_t} [dB] = 20 \log \left| \frac{P_R}{P_t} \right|$$



Powyżej schemat dla punktu 1D

Poniżej implementacja kodu:

```
#1D code
ReTM = (er_wall*np.cos(0) - np.sqrt(er_wall -
(np.sin(0)**2)))/(er_wall*np.cos(0) + np.sqrt(er_wall- (np.sin(0)**2)))
# obliczenie współczynnika odbicia od ściany
```

```

a_wall = ReTM

d0=np.arange(dmin, dmax+k, zakres/(lp-1),dtype=float) # nowy wektor
ruchu odbiornika po całej długości ściany b

fi1 = np.pi*(-2)*d0/c #faza bez odbicia
fi2 = np.pi*(-2)*(2*b-d0)/c # faza z odbiciem od przeciwległej ściany

np.seterr(divide='ignore', invalid='ignore') #usuwa błędy związane z
dzieleniem przez zero
# dalsze wyznaczenie fazy
e1=np.exp(1j*fi1)
e2=np.exp(1j*fi2)

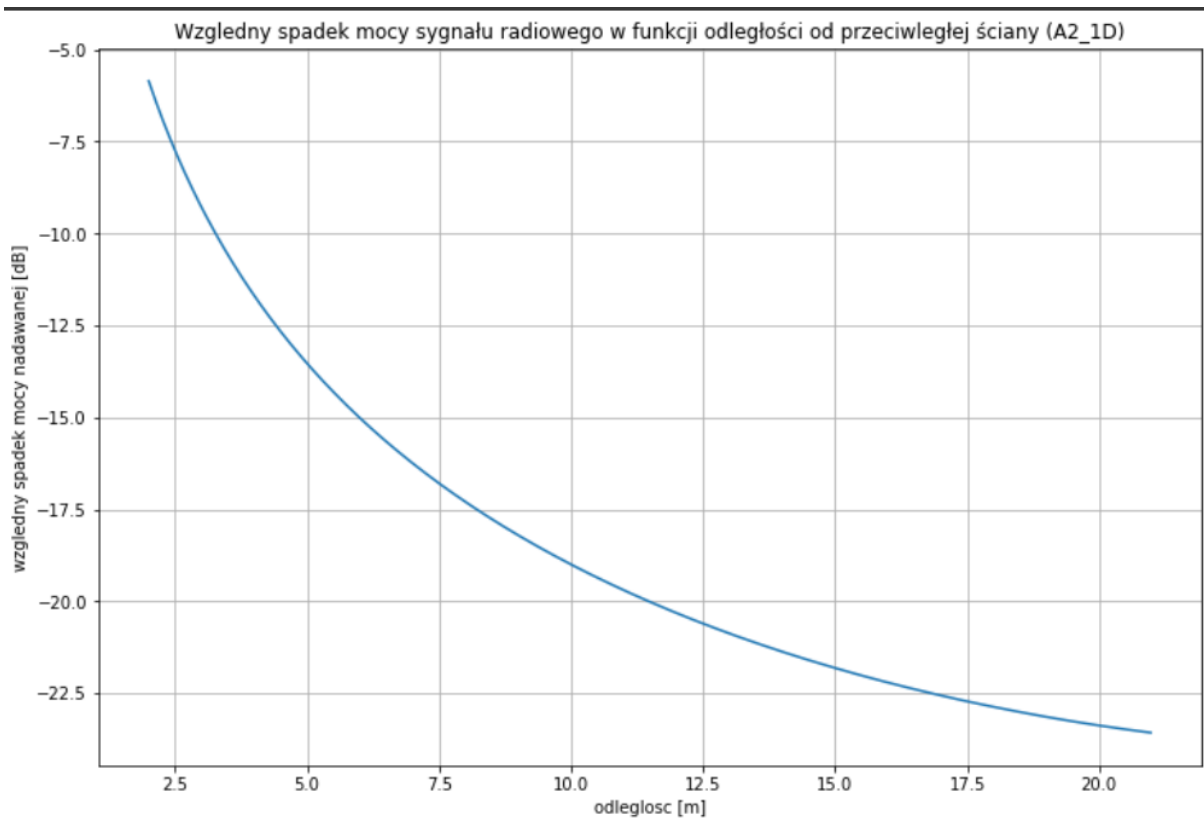
PrPt1 = np.abs((1/d0*e1 + a_wall/(2*b-d0)*e2)) # wyznaczenie tłumienie
ścieżki przy dwóch drogach
PrPt1 = 20*np.log10(PrPt1) # tłumienie wyrażone w [dB]

```

```

#plot 1D
plt.figure(figsize=(12,8))
plt.title("Względny spadek mocy sygnału radiowego w funkcji odległości
od przeciwległej ściany (A2_1D)")
plt.xlabel("odleglosc [m]")
plt.ylabel("względny spadek mocy nadawanej [dB]")
plt.plot(d0,PrPt1)
plt.grid()

```



- **2D – od przeciwległej ściany i bocznych ścian**

Skorzystałem ze wzoru na wielodrogowość:

$$\frac{P_R}{P_0} = \left| \frac{1}{d} e^{-j\varphi_1} + \frac{a_{wall1}}{2b-d} e^{-j\varphi_2} + \frac{a_{wall1}}{d_3} e^{-j\varphi_3} + \frac{a_{wall2}}{d_4} e^{-j\varphi_4} \right|^2$$

Obliczenie fazy:

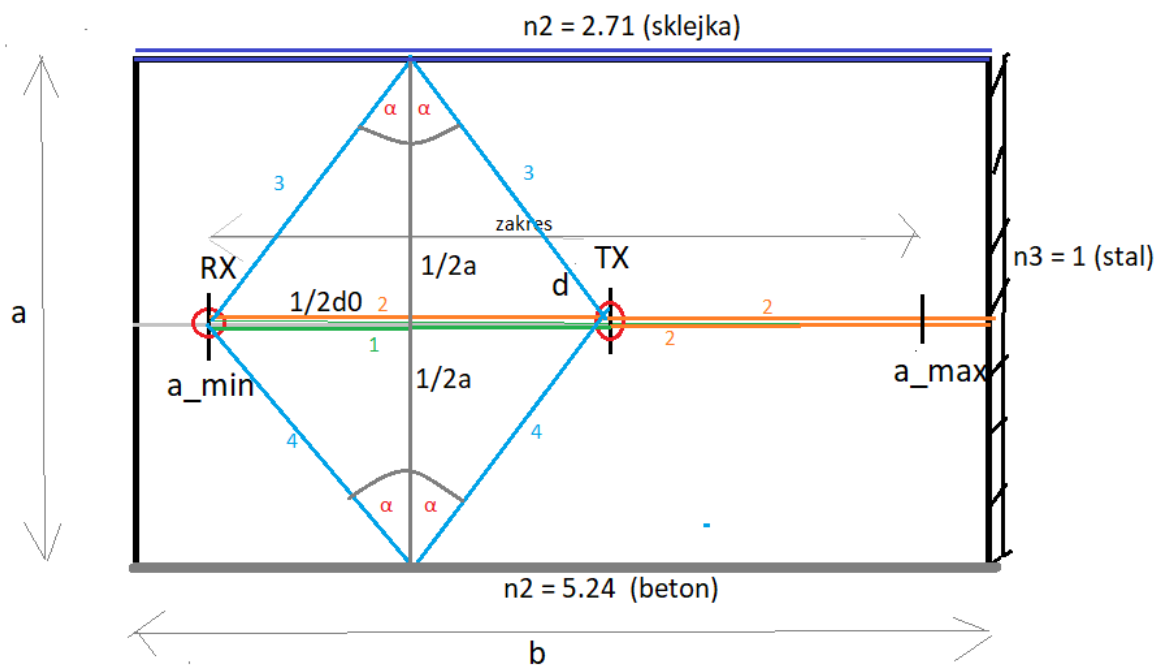
$$\varphi_1 = 2\pi \frac{d0}{c}$$

$$\varphi_2 = 2\pi \frac{2b-d1}{c}$$

$$\varphi_1 = 2\pi \frac{d2}{c}$$

$$\varphi_1 = 2\pi \frac{d2}{c}$$

$$\frac{Pr}{Pt} [dB] = 10 \log \left| \frac{Pr}{Pt} \right|$$



Powyżej schemat dla punktu 2.2D

```
#PARAMETRY 2D

er_wall1 = 5.24#plasterboard

er_wall2 = 2.71#plywood

er_wall3 = 1    #metal

# prawa i lewa ściana są z różnych materiałów i przeciwległa też

# cztery ścieżki bezpośrednio ,od przeciwległej ściany, od dwóch
bocznych ścian

d0=np.arange(dmin, dmax+k, zakres/(lp-1),dtype=float) #wektor do
poruszania odbiornikiem

h1 = a*np.ones(len(d0)) #wektor odbicia od przeciwległej ściany

h2 = b*np.ones(len(d0)) # wektor odbicia od bocznej ściany

d1=np.sqrt(np.power(h1,2)+(d0**2)/4)    # wektor ścieżki dla
przeciwległej ściany

d2=np.sqrt(np.power(h2,2)+(d0**2)/4)    # wektor ścieżki dla bocznych
ścian
```

```

sin_alp = d0/d2 # z tw. sinusów

cos_alp = np.sqrt(1-sin_alp**2)

ReTM1 = (er_wall1*cos_alp - np.sqrt(er_wall1 -
(sin_alp**2)))/(er_wall1*cos_alp + np.sqrt(er_wall1- (sin_alp**2)))
#kąt zależy od d0

ReTM2 = (er_wall2*cos_alp - np.sqrt(er_wall2 -
(sin_alp**2)))/(er_wall2*cos_alp + np.sqrt(er_wall2- (sin_alp**2)))
#kąt zależy od d0

ReTM3 = (er_wall3 - np.sqrt(er_wall3))/(er_wall3 + np.sqrt(er_wall3)) #
przeciwległa ściana jest z metalu i kat to 90 stopni

a_wall1 = ReTM1 #obliczenie współczynnika odbicia od ściany
a_wall2 = ReTM2
a_wall3 = ReTM3

```

```

#2D code

d2=2*np.sqrt(np.power(h2,2)+(d0**2)/4) #

fi1 = np.pi*(-2)*d0/c #faza bez odbicia

fi2 = np.pi*(-2)*(2*b-d1)/c # faza z odbiciem od przeciwległej ściany

fi3 = -2*np.pi*f*d2/c # faza odbicia od ściany bocznej plasterboard

fi4 = -2*np.pi*f*d2/c # faza odbicia od ściany bocznej plywood

e1=np.exp(1j*fi1)

e2=np.exp(1j*fi2)

e3=np.exp(1j*fi3)

e4=np.exp(1j*fi3)

#print(d0)

```

```
PrPt2 = np.abs((1/d0*e1 + a_wall3/(2*b-d1)*e2 + a_wall1/d2*e3) +
a_wall2/d2*e4)

PrPt2 = 20*np.log10(PrPt2)
```

```
#plot 2D

plt.figure(figsize=(12,8))

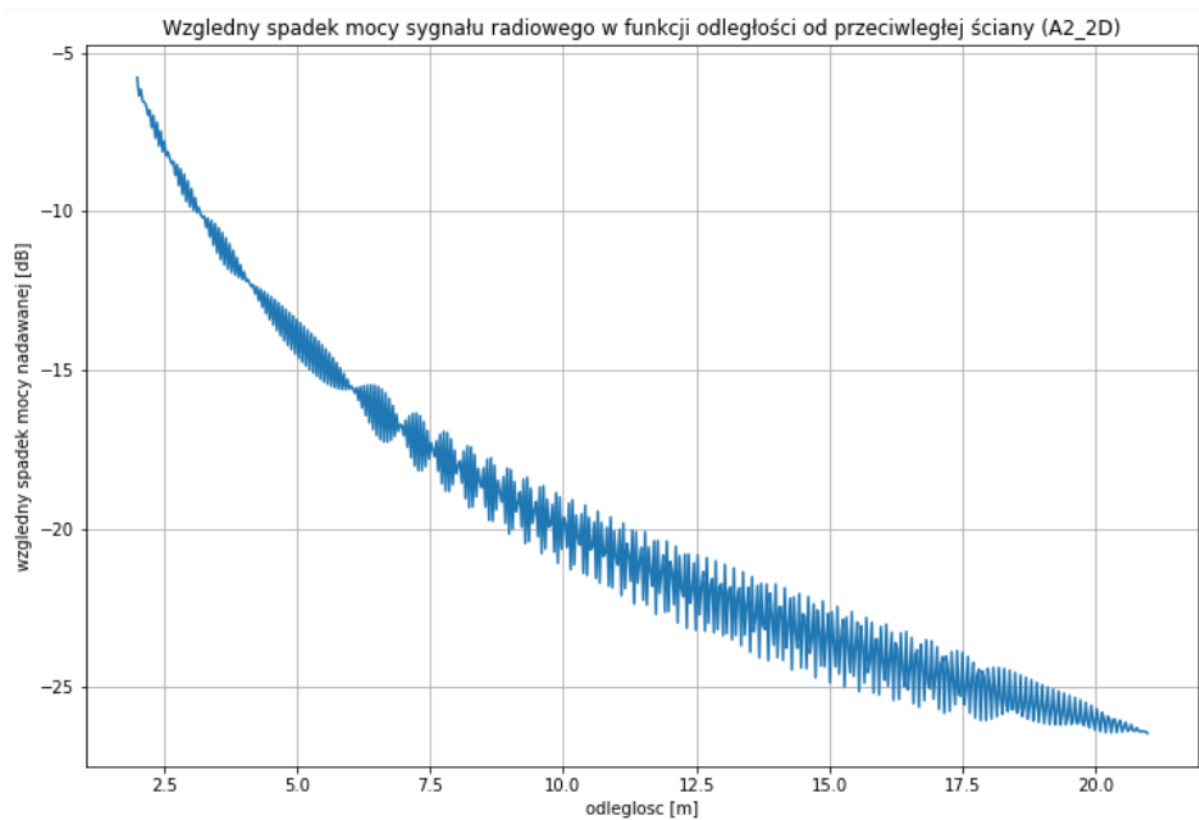
plt.title("Względny spadek mocy sygnału radiowego w funkcji odległości
od przeciwległej ściany (A2_2D)")

plt.xlabel("odległość [m]")

plt.ylabel("względny spadek mocy nadawanej [dB]")

plt.plot(d0,PrPt2)

plt.grid()
```



A.3 Rozbudowałem pokój o część o bokach b, a . Odbiornik porusza się w osi tej części. Wyznaczyłem stratę dyfrakcji według metody Deygout. Przyjmij występowanie dyfrakcji wyłącznie, gdy odbiornik nie ma bezpośredniej widoczności z nadajnikiem.

- **1D – wyznacz wartość straty dyfrakcji w osi dobudowanej części**

Skorzystałem z poniższych wzorów.

$$S_1 + S_2 = \sqrt{\left(a - \frac{1}{2}b\right)^2 + \left(d - \frac{1}{2}b\right)^2}$$

$$r_1 = \text{const.} = \sqrt{\left(\frac{1}{2}b\right)^2 + (a - b)^2}$$

$$r_2 = \text{const.} = \sqrt{\left(\frac{1}{2}b\right)^2 + (d - b)^2}$$

Z twierdzenia cosinusów:

$$r_2^2 = r_1^2 + (s_1 + s_2)^2 - 2r_1(s_1 + s_2)\cos\alpha$$

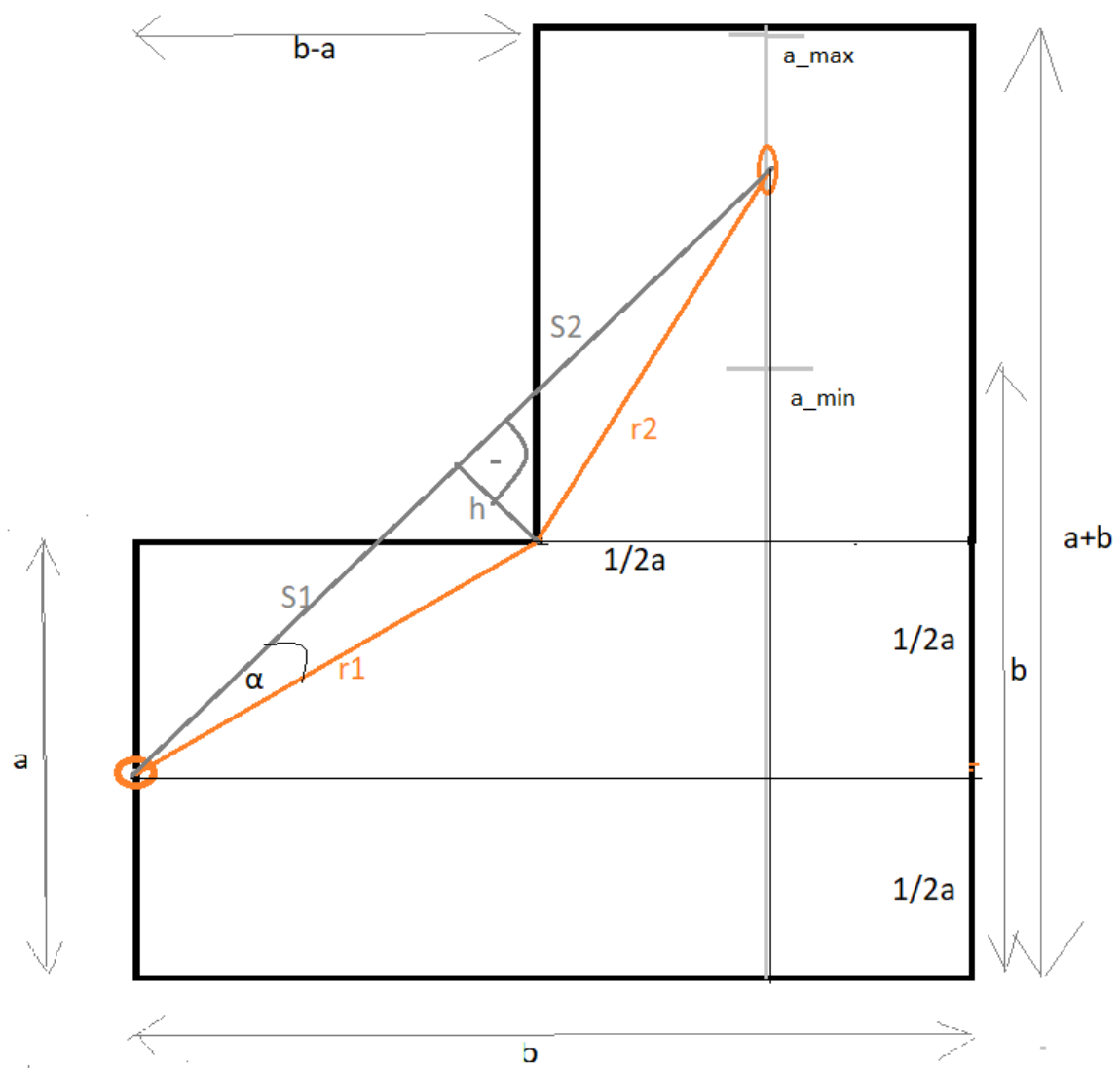
$$\alpha = \arccos\left(\frac{r_2^2 - r_1^2 - (s_1 + s_2)^2}{-2r_1(s_1 + s_2)}\right)$$

$$h = r_1 \frac{n}{r_1} = r_1 \sin\alpha$$

Obliczenie straty dyfrakcji C:

$$v = h \sqrt{\frac{2}{\lambda} \frac{s_1 + s_2}{r_1 r_2}}$$

$$C = 6,9 + 20 \log(\sqrt{(v - 0,1)^2 + 1} + v - 0,1)$$



```
#parametry A3
```

```
dmin = b
```

```
dmax = a+b
```

```

#D1 code

d= np.arange(dmin, dmax, zakres/(lp-1),dtype=float)

#DYFRAKCJA (ugięcie fali na krawędzi)

S = np.sqrt((b-1/2*a)**2+(d-1/2*a)**2)

# tw. Cosinusów

r1 = np.sqrt((b-a)**2 + (1/2*a)**2) #stałe (przeciwprostokątna trójkąta r1)

r2 = np.sqrt((1/2*a)**2 + (a-d)**2) # zmienia w zależności od d
(przeciwprostokątna trójkąta r2)

alpha = np.arccos((r2**2 - r1**2 - S**2)/(-2*r1*S)) # wyznaczenie kata
alpha

# Wyznaczanie straty dyfrakcji w osi dobudowanej pokoju (na rogu
ściany)

h = r1*np.sin(alpha) # obliczenie wysokości trójkąta z tw.cosinusów

v = h*np.sqrt(2/lamb * S/(r1*r2)) #wyznaczenie v

# straty dyfrakcji

C = 6.9 + 20*np.log10(np.sqrt((v-0.1)**2 + 1) + v-0.1)

```

```

# plot D1

plt.figure(figsize=(12,8))

plt.title("wartość straty dyfrakcji w osi dobudowanej części D1")

plt.xlabel("odległość [m]")

plt.ylabel("strata dyfrakcji [dB]")

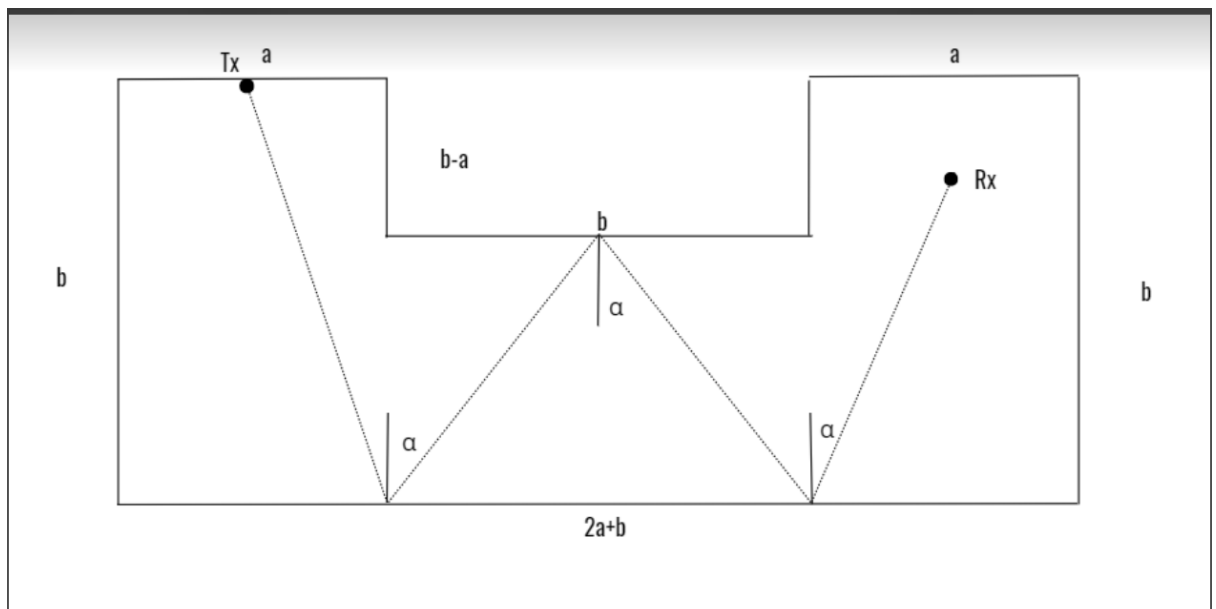
plt.plot(d, C)

plt.grid()

```

A.4 Kolejna rozbudowa, tak, aby pokój stał się symetryczną podkową. Wyznacz wartość tłumienia ścieżki dla trzech wariantów radiowych:

A.4.3 Sygnał ulega kilkukrotnym odbiciom wynikającym z zasad geometrii.



Powyżej schemat dla punktu 3.4D

```
# code A.4.1

d_min = 0

d_max = b-a

tlum_scian = 5 #(light wall)

zakres = d_max - d_min

d=np.arange(d_min, d_max, zakres/(lp-1),dtype=float)

h1 = b*np.ones(len(d))
```

```

h2 = a*np.ones(len(d))

h3 = a*np.ones(len(d))

h4 = (b-d)*np.ones(len(d))

d1 = np.sqrt((1/2*a)**2 + h1**2)

d2 = np.sqrt((1/2*b)**2 + h2**2)

d3 = np.sqrt((1/2*b)**2 + h2**2)

d4 = np.sqrt((1/2*a)**2 + h4**2)

ReTM = (er_wall*np.cos(0) - np.sqrt(er_wall -
(np.sin(0)**2)))/(er_wall*np.cos(0) + np.sqrt(er_wall- (np.sin(0)**2)))
# obliczenie współczynnika odbicia od ściany

a_wall = ReTM


fi1 = np.pi*(-2)*f*d1/c #faza bez odbicia

fi2 = np.pi*(-2)*f*d2/c # faza z odbiciem od przeciwległej ściany

fi3 = np.pi*(-2)*f*d3/c # faza odbicia od ściany bocznej plasterboard

e1=np.exp(1j*fi1)

e2=np.exp(1j*fi2)

e3=np.exp(1j*fi3)


PrPt4 = np.abs(a_wall/d1*e1 + a_wall/d2*e2 + a_wall/d3*e2 +
a_wall/d4*e3)

PrPt4 = 20*np.log10(PrPt4)

plt.title('Sygnał ulega kilkukrotnym odbiciom wynikającym z zasad
geometrii ')

plt.xlabel('Odległość RX-TX [m]')

plt.ylabel('Tłumienie ścieżki [dB] ')

```



```
plt.plot(d, PrPt4)

plt.grid()
```

Bibliografia:

- Wykłady “Podstawy Radiokomunikacji” - *dr inż. Krzysztof Cichoń*
- Ćwiczenia z przedmiotu “Podstawy Radiokomunikacji” - *dr inż. Krzysztof Cichoń*
- “Projektowanie i obliczenia w radiokomunikacji”, Poznań 2005 - *prof. Hanna Bogucka*