

Laboratorium z podstaw fizyki Wydziału EliT AGH

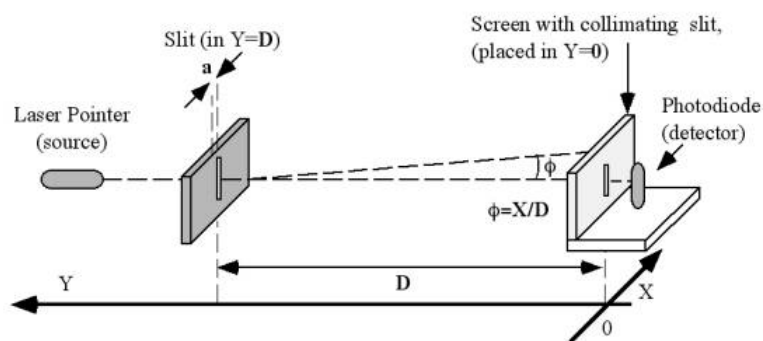
Przykłady obliczeń

© Michał Kołodziej 2016, kolodziej.michal@gmail.com

Laboratorium 11 - Badanie zjawiska dyfrakcji i polaryzacji światła

Ustawienie eksperymentu

Stanowisko jest wyposażone następująco:



[<http://www.lepla.org/en/modules/Activities/m14/m14-setup.htm> (<http://www.lepla.org/en/modules/Activities/m14/m14-setup.htm>)]

Na ekranie otrzymamy podobny obraz dyfrakcyjny jak poniżej:



[<https://www.andrews.edu/phys/wiki/PhysLab/doku.php?id=lab-9> (<https://www.andrews.edu/phys/wiki/PhysLab/doku.php?id=lab-9>)]

Związek obrazu dyfrakcyjnego z intensywnością odczytywaną przez miernik

Laser emituje elektromagnetyczną falę płaską, zgodnie z oznaczeniami osi na rysunku powyżej wzór fali będzie następujący:

$$E(y, t) = E_m \cos(\omega t + ky)$$

E_m to maksimum pola elektrycznego.

Fala w przestrzeni (medium) rozchodzi się według zasady Huygensa (https://pl.wikipedia.org/wiki/Zasada_Huygensa) czyli każdy punkt przestrzeni jest źródłem fali kulistej, a na skutek interferencji obserwujemy na przykład wzór fali płaskiej.

Fala płaska napotyka szczelinę o szerokości porównywalnej lub większej do długości tej fali. Korzystając z zasady Huygensa możemy przewidzieć wzór fali za szczeliną.

Dla pojedynczej szczeliny jasność w funkcji kąta odchylenia od osi przyjmuje postać:

$$I(\theta) = I_0 [\text{sinc}(\alpha)]^2$$

$$\alpha = \frac{\pi d}{\lambda} \sin \theta$$

gdzie:

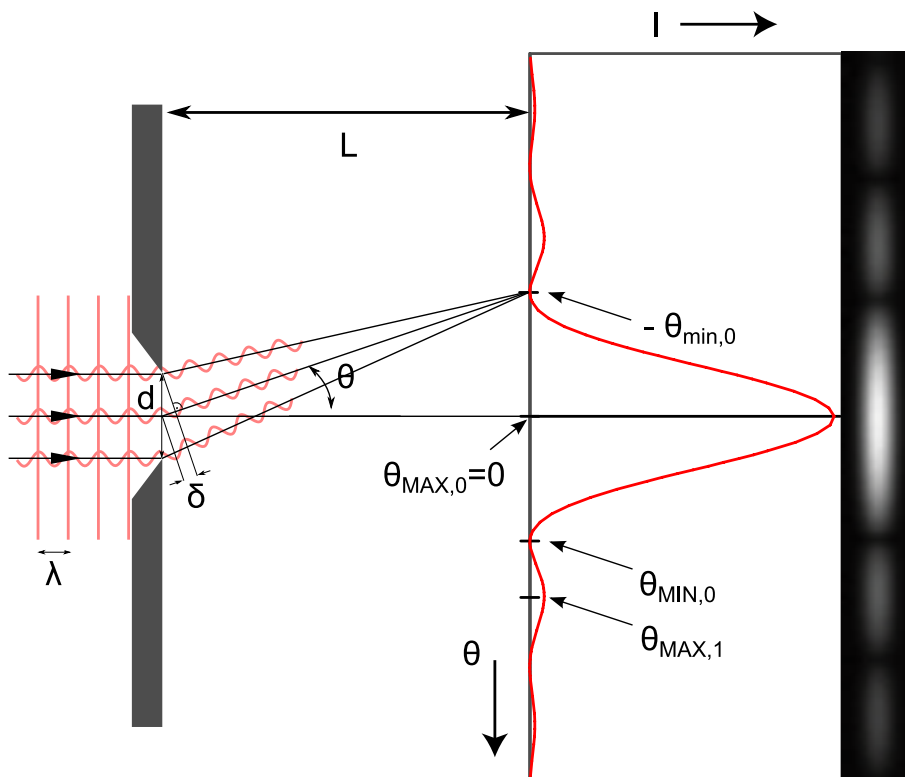
I – intensywność światła,

I_0 – intensywność światła w maksimum, czyli dla kąta równego θ ,

λ – długość fali,

d – szerokość szczeliny,

funkcja $\text{sinc}(x) = \sin(x)/x$.



[https://commons.wikimedia.org/wiki/File:Single_Slit_Diffraction.svg] (https://commons.wikimedia.org/wiki/File:Single_Slit_Diffraction.svg)

Właściwości nieznormalizowanej funkcji sinc

[https://pl.wikipedia.org/wiki/Funkcja_sinc (https://pl.wikipedia.org/wiki/Funkcja_sinc)]

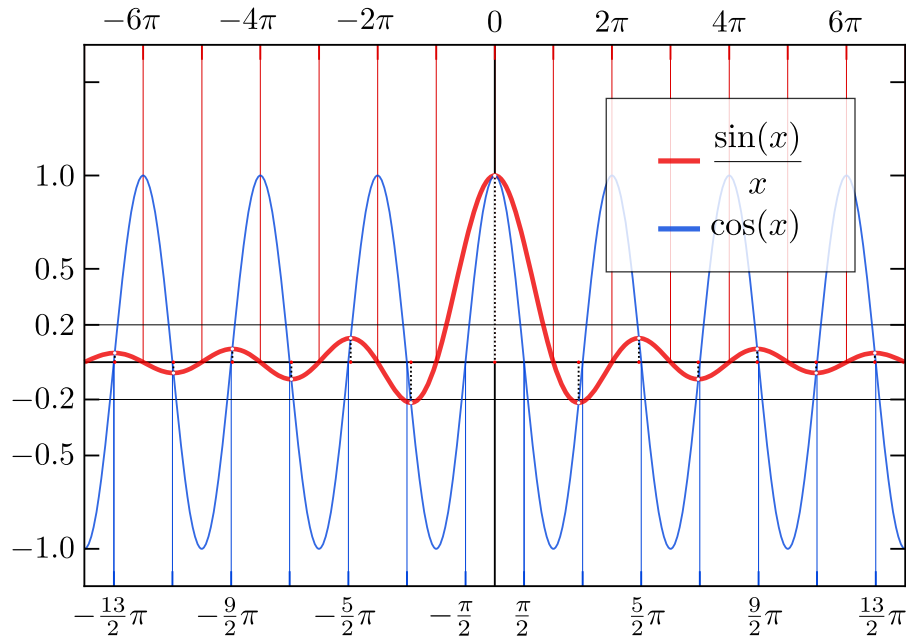
Argument funkcji sinc we wzorze na intensywność fali na ekranie:

$$\frac{\pi d}{\lambda} \sin \theta$$

służy do skalowania ("zwężania", "rozszerzania") funkcji sinc w zależności od szerokości szczeliny. Jest to analogiczna procedura jak na przykład gdy chcemy "rozszerzyć" dwukrotnie funkcję $\cos(\omega t)$ stosujemy $\cos(0.5\omega t)$

Istotną informacją jest wiedza dla jakich argumentów funkcja sinc przyjmuje minimum (lub ma miejsca zerowe).

Miejsca zerowe nieznormalizowana funkcja sinc przyjmuje dla argumentów będących całkowitą niezerową wielokrotnością liczby π .



Pomiar intensywności w funkcji położenia detektora $I(x_i)$

[https://pl.wikipedia.org/wiki/Nat%C4%99C5%BCenie_promieniowania (https://pl.wikipedia.org/wiki/Nat%C4%99C5%BCenie_promieniowania)]

W radiometrii natężenie (intensywność) promieniowania to wielkość fizyczna określona jako strumień promieniowania wysyłany w jednostkowy kąt bryłowy.

$$I = \frac{\Phi}{\Omega}$$

gdzie:

- I - natężenie (intensywność) promieniowania
- Φ - strumień promieniowania
- Ω - kąt bryłowy

W układzie SI jednostką natężenia (intensywności) promieniowania jest wat na steradian (W/sr). (Fotodioda mierzy irradancję - W/m², ale w tym przypadku możemy ją uznać za intensywność)

Przesuwając detektor odczytujemy jego względne położenie ze śruby mikrometrycznej i odpowiadające mu natężenie światła na mierniku uniwersalnym (napięcia/prądu). Zapisujemy pomiar. Następnie normalizujemy położenia przyjmując położenie największego maksimum jako $x=0$.

```
In [77]: polozenia_pomiar = collect(0:1:20)
intensywnosci = [
    0.219,0.233,0.258,0.258,0.234,0.259,0.288,0.292,0.27,0.309,0.35,0.381,0.394,0.392,0.
    .377,0.346,0.308,0.278
]

znormalizuj_polozenia(polozenia, intensywnosci) = polozenia - polozenia[indmax(intensyw
nosci)]
polozenia = znormalizuj_polozenia(polozenia, intensywnosci)

intensywnosc_max = findmax(intensywnosci)[1]

[
    ("polozenia_pomiar [mm]", polozenia_pomiar),
    ("polozenia [mm]", polozenia),
    ("intensywnosci [W/m2]",intensywnosci),
    ("intensywnosc_max [W/m2]",intensywnosc_max)
]

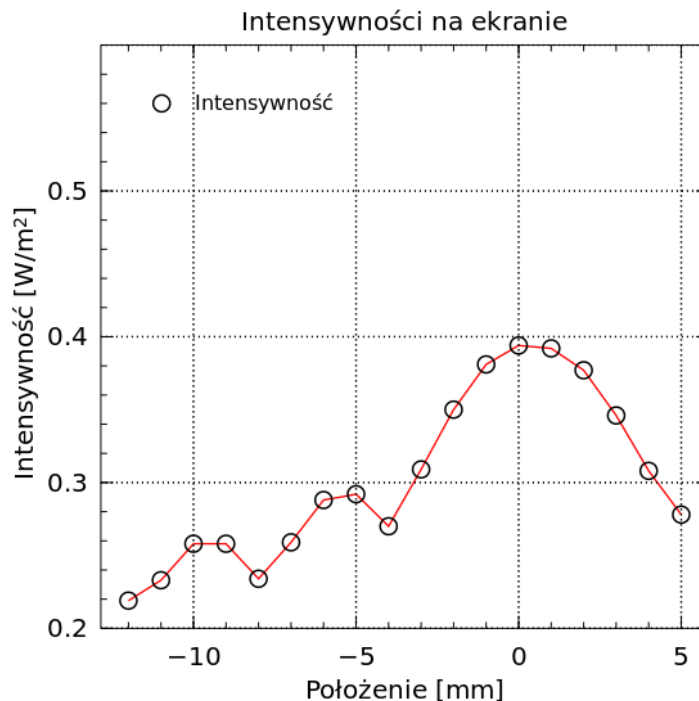
Out[77]: 4-element Array{Tuple{ASCIIString,Any},1}:
 ("polozenia_pomiar [mm]", [0,1,2,3,4,5,6,7,8,9 ... 11,12,13,14,15,16,17,18,19,20])
 ("polozenia [mm]", [-12,-11,-10,-9,-8,-7,-6,-5,-4,-3 ... -1,0,1,2,3,4,5,6,7,8])
 ("intensywnosci [W/m2]", [0.219,0.233,0.258,0.258,0.234,0.259,0.288,0.292,0.27,0.309,0.
35,0.381,0.394,0.392,0.377,0.346,0.308,0.278])
 ("intensywnosc_max [W/m2]", 0.394)
```

Ilustracja graficzna znormalizowanej intensywności $I(x)$

```
In [23]: import Winston
p = Winston.FramedPlot(title = "Intensywności na ekranie", xlabel="Położenie [mm]", ylabel="Intensywność [W/m^2]")
Winston.setattr(p, yrange=(0.2, 0.6))
Winston.setattr(p.frame, draw_grid=true)
a = Winston.Curve(położenia, intensywnosci, color=parse(Winston.Colorant, "red"))
Winston.setattr(a, label="")
b = Winston.Points(położenia, intensywnosci, kind="circle")
Winston.setattr(b, label="Intensywność")

l = Winston.Legend(.1, .9, Any[b])
Winston.add(p, a, b, l)
Winston.savepng(p, "Lab_11_prazki.png", 600, 600)
HTML("""""")
```

Out[23]:



Wyznaczamy znormalizowane położenia minimów intensywności x_{min}

Zapisujemy położenia (wg znormalizowanej skali x) dla których ekstrema intensywności są minimami.

```
In [91]: find_minimums(xs) = find(x -> x[2] <= x[1] && x[1] <= x[3] , zip(xs[1:end-2], xs[2:end-1],
xs[3:end]) |> collect) + 1
find_minimums(intensywnosci)

położenia_minimow = położenia[find_minimums(intensywnosci)] |> reverse #/> abs
```

```
Out[91]: 2-element Array{Int32,1}:
 -4
 -8
```

Dopasowanie funkcji $I(\theta)$ do zmierzonej intensywności = znalezienie szerokości szczeliny d

Funkcja

$$I(\theta) = I_0 [\text{sinc}(\alpha)]^2$$

ma miejsca zerowe dla α będącego wielokrotnościami π , czyli 1 minimum (licząc od maximum) będzie dla $\alpha = \pi$, drugie minimum dla $\alpha = 2\pi$, w związku z czym otrzymujemy równania:

$$\begin{aligned}\pi &= \frac{\pi d}{\lambda} \sin \theta_1 \\ 2\pi &= \frac{\pi d}{\lambda} \sin \theta_2\end{aligned}$$

gdzie:

d – jest nieznaną szerokością szczeliny,
 λ – jest długością fali płaskiej padającej na szczelinę (w przypadku lasera gazowego He-Ne 632,8 nm),
 θ_1 – nieznanym kątem dla którego zmierzono pierwsze minimum intensywności,
 θ_2 – nieznanym kątem dla którego zmierzono drugie minimum intensywności.

Kąt θ możemy wyznaczyć z właściwości geometrycznych stanowiska pomiarowego, tangens tego kąta to stosunek znormalizowanego położenia fotodiody x do odległości pomiędzy szczeliną a fotodiodą L :

$$\theta = \arctg\left(\frac{x}{L}\right)$$

Przekształcając powyższe wzory otrzymujemy wzór na szerokość szczeliny, dla pierwszego minimum:

$$d = \frac{\lambda}{\sin\left(\arctg\left(\frac{x_1}{L}\right)\right)}$$

dla drugiego minimum:

$$d = \frac{2\lambda}{\sin\left(\arctg\left(\frac{x_2}{L}\right)\right)}$$

itd.

Podsumowując, jeżeli znajdziemy d spełniające powyższe równania to wykres funkcji $I(\theta)$ powinien mieć minima dla takich samych x -ów jak punkty pomiarowe intensywności.

```

In [92]: odleglosc_detektora = 80 # [cm]
         λ_lasera = 632.8 # nm
         nano = 1e-9
         mili = 1e-3

         kat_padania(x, L) = atan(x/L)
         θ_minimow = round(map(x-> kat_padania(x, odleglosc_detektora*10), polozenia_minimow), 4)

         szerokosc_szczeliny(rzad_minimum, θ, λ) = rzad_minimum * λ / sin(θ)
         szerokosci_szczeliny = [
             szerokosc_szczeliny(1, θ_minimow[1], λ_lasera * nano),
             szerokosc_szczeliny(2, θ_minimow[2], λ_lasera * nano)
         ]

         szerokosci_szczeliny_srednia = mean(szerokosci_szczeliny) # [m]

         [
             ("odleglosc_detektora [cm]", odleglosc_detektora),
             ("λ_lasera [nm]", λ_lasera),
             ("θ_minimow [rad]", θ_minimow),
             ("szerokosci_szczeliny [mm]", round(szerokosci_szczeliny/mili, 2)),
             ("szerokosci_szczeliny_srednia [mm]", round(szerokosci_szczeliny_srednia/mili, 2))
         ]

Out[92]: 5-element Array{Tuple{AbstractString,Any},1}:
 ("odleglosc_detektora [cm]",80)
 ("λ_lasera [nm]",632.8)
 ("θ_minimow [rad]",[-0.005,-0.01])
 ("szerokosci_szczeliny [mm]",[-0.13,-0.13])
 ("szerokosci_szczeliny_srednia [mm]",-0.13)

```

Ilustracja funkcji $I(\theta)$ dla wyznaczonej szerokości szczeliny d

$$I(x, L, d, \lambda, I_0) = I_0 \left[\text{sinc} \left(\frac{\pi d}{\lambda} \sin \left(\arctg \left(\frac{x}{L} \right) \right) \right) \right]^2$$

```

In [103]: nsinc(x) = if x == 0 1 else sin(x)/x end
intensywnosc(x, L, d, λ, I_0) = I_0 * nsinc(pi * d / λ * sin(atan(x/L)))^2
centy = 1e-2

xs = linspace(-12, 12, 100)
is = map(x-> intensywnosc(x * mili, odleglosc_detektora * centy, szerokosci_szczeliny_s
rednia, λ_lasera * nano, intensywnosc_max), xs)

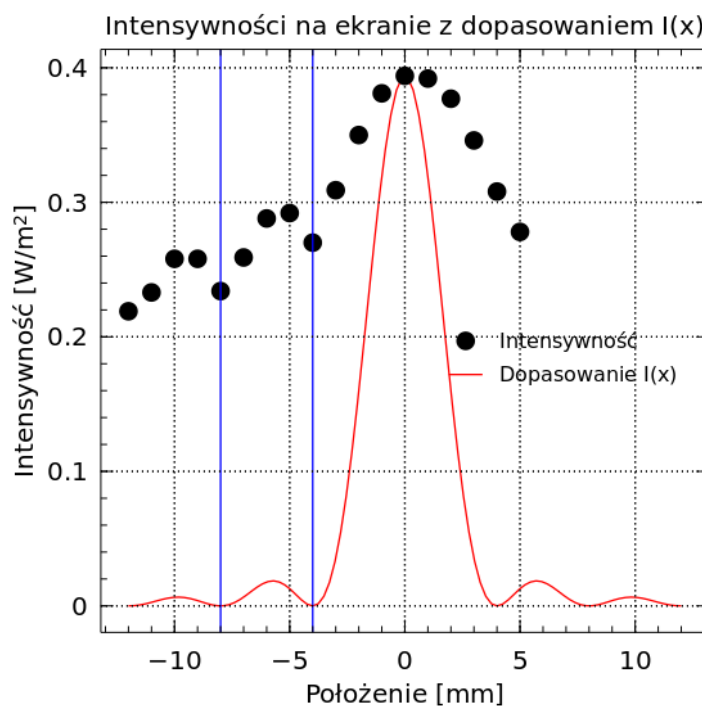
import Winston
p = Winston.FramedPlot(title = "Intensywności na ekranie z dopasowaniem I(x)",
xlabel="Położenie [mm]", ylabel="Intensywność [W/m^2]")

# Winston.setattr(p, yrange=(0.2, 0.6))
Winston.setattr(p.frame, draw_grid=true)
a = Winston.Curve(xs, is, color=parse(Winston.Colorant, "red"))
Winston.setattr(a, label="Dopasowanie I(x)")
b = Winston.Points(polozenia, intensywnosci, kind="filled circle")
Winston.setattr(b, label="Intensywność")
l1 = Winston.LineX(polozenia_minimow[1], color=parse(Winston.Colorant, "blue"))
l2 = Winston.LineX(polozenia_minimow[2], color=parse(Winston.Colorant, "blue"))

l = Winston.Legend(.6, .5, Any[b, a])
Winston.add(p, a, b, l1, l2, l)
Winston.savepng(p, "Lab_11_dopasowanie.png", 600, 600)
HTML("<img src='Lab_11_dopasowanie.png?$(datetime2unix(now()))' alt='Test' width='550' />")

```

Out[103]:



Błąd wyznaczenia szerokości szczeliny Δd dla pierwszego minimum

$$\Delta d = d(x + \Delta x, L + \Delta L, \lambda) - d(x, L, \lambda)$$

$$d(x, L, \lambda) = \frac{\lambda}{\sin(\arctg(\frac{x}{L}))}$$


```
In [107]: szerokosc_szczeliny_x(rzad, x, L, λ) = szerokosc_szczeliny(rzad, kat_padania(x, L), λ)

szerokosc_szczeliny_blad =
    szerokosc_szczeliny_x(1, (polozenia_minimow[1] + 0.5) * mili, (odleglosc_detektora
+ 2) * centy, λ_lasera * nano) -
    szerokosc_szczeliny_x(1, polozenia_minimow[1] * mili, odleglosc_detektora * centy,
λ_lasera * nano)

("szerokosc_szczeliny_blad [mm]", round(szerokosc_szczeliny_blad / mili, 3))

Out[107]: ("szerokosc_szczeliny_blad [mm]", -0.022)
```