Санкт-Петербургский Государственный Электротехнический Университет

«ЛЭТИ»

кафедра физики

Задание №1 по дисциплине

«Физические основы информационных технологий» Название: Искревление луча в оптическом канале.

Фамилия И.О.: Токун Г.С. группа: 1303

Преподователь: Альтмарк А.М. Итоговый балл:

Крайник срок сдачи: 22.10.2023

Санкт-Петербург 2023

# Цель работы

Изучит распространение света в оптоволоконном канале.

# Задание

Найти длину траектории светового луча S в прямолинейном дисперсионном оптоволокном канале, с показателем преломления *n*1. Оптоволокно окружено средой с показателем преломления *n*2. Функция расрпеделения показателья пре- ломление:

*n*1(*y, ω*) = *f*1

(*y*)(1 − (0*.*35·1014 )2),

где *y* - поперечная координата, *ω* - циклическая частота светового луча.

*ω*

*R* = 0*.*6

*n*2 = 1

√

*f*1(*y*) = 1*.*4 + 0*.*6 − *y*2

*Zf* (*y*) = 28 + 3 ∗ *sin*[17*.*951958020513104 ∗ *y*] *ω* = 3*.*4 ∗ 1014

*y*0 = −0*.*2

*α* = −20◦

# Основные теоритеческие положения Закон преломления света

1. Падающий и преломленный лучи и перпендикуляр, проведенный к границе раздела двух сред в точке падения луча, лежат в одной плоскости.
2. Отношение синуса угла падения к синусу угла преломления есть величина постоянная для двух сред, равная относительному показателю преломле- ния:

*sinα* = *n*2

*sinβ*

*n*1

# Полное внутреннее отражение света

(1)

Полное внутреннее отражение — это явление, при котором вся падающая вол- на отражается от границы раздела двух сред, меняет направление своего рас- пространения так, что не проникает во вторую среду, а возвращается в первую или скользит по границе раздела двух сред.

Предельный угол полного внутреннего отражения (*α*0) — угол падения, ко- торому соответствует угол преломления 90◦, если *α < α*0, то происходит пре- ломление и падающий луч проникает в менее плотную среду;если *α* = *α*0, то угол преломления *γ* = 90◦, падающий луч скользит вдоль границы раздела двух сред;если *α > α*0, то наблюдается полное внутреннее отражение.

# Выполнение работы

Была разработана программа для построения траектории луча, торца волно- вода и нахождения длины траектории луча.

Сначала происходит задание констант и функций, данных по условию:

R = 0.6

N\_2 = 1

OMEGA = 3.4 \* 10\*\*14 Y\_0 = -0.2

ALPHA\_0 = radians(-20) COUNT\_LAYERS = 1000000

def f1(y):

return 1.4 + sqrt(0.6 - y\*\*2)

def Zf(y):

return 28 + 3\*sin(17.951958020513104 \* y)

def n(y):

return f1(y) \* (1 - ((0.35 \* 10 \*\* 14) / OMEGA) \*\* 2)

def n1(y):

return f1(y) \* (1-(0.35 \* 10\*\*14) \*\* 2)

Основная идея следующая:

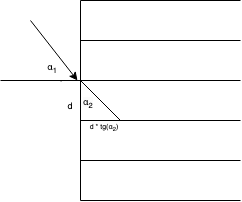


Рис. 1: Пример подсчета

Итеративно с помощью функции:

def new\_angle(n1, n2, sin\_1): return (n1/n2) \* sin\_1

Будем подсчитывать каждый раз новый угол и, используя новый угол, будем считать новое смещение по *x*, смещение по *y* всегда будет одинаковым, ширина слоя, меняться будет только направление (вверх или вниз).

То идем мы вверх или вниз определяет if:

if flag\_ref or abs(abs(y) - R) < 0.0005:

alpha\_2 = alpha\_1 dy \*= - 1

Он срабатывает либо в случае полное отражение луча, либо дохождения до края волновода (R или -R). Полное отражения определяется так:

Длину волны определяем через теорему пифагора, каждый раз прибавляя длину гипотенузы в переменную суммы:

length += sqrt(dx\*\*2+dy\*\*2)

Условие выхода из функции то, что *x* вышел за пределы волновода:

while abs(Zf(y) - x) >= 0.05:

Построение торца волновода происходит следующим образом:

wave\_x = [] wave\_y = [] x = -R

while x < R:

x += 0.001

wave\_x.append(Zf(x)) wave\_y.append(x)

return wave\_x, wave\_y

И конце строим график:

def make\_chart(way\_x, way\_y, wave\_x, wave\_y): plt.xlabel("Z")

plt.ylabel("Y") plt.title("Траектория волны") plt.ylim(-R, R) plt.plot(way\_x, way\_y) plt.plot(wave\_x, wave\_y) plt.show()

# Тестирование

Получившийся результат при 1000000 слоях, длина волны 26.09765052901833:

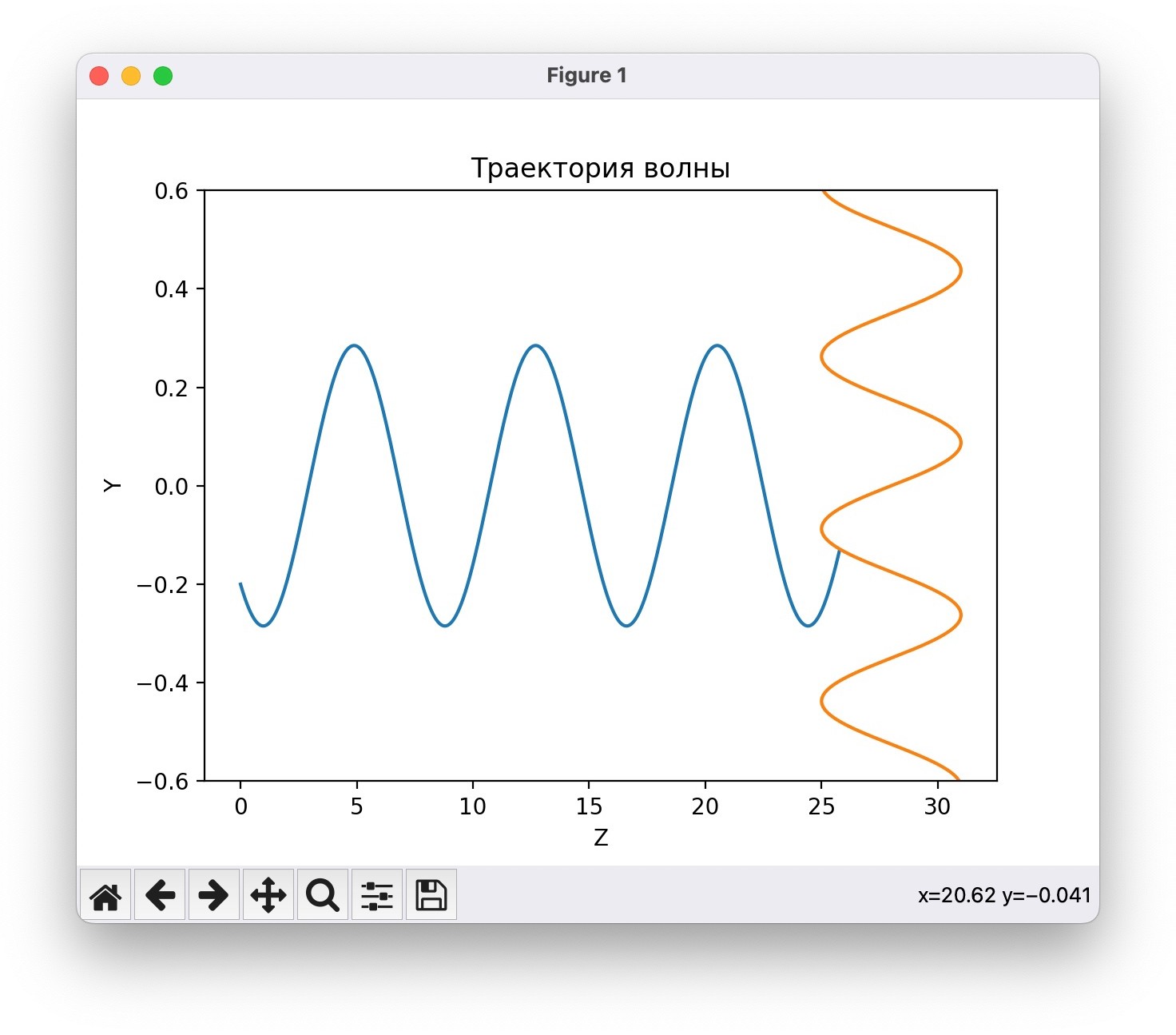


Рис. 2: Результат

# Вывод

Изучено распространие света в оптоволоконном канале.

**Весь код:**

from math import sqrt, sin, radians, pi, asin, tan  
import matplotlib.pyplot as plt  
  
R = 0.6  
N\_2 = 1  
OMEGA = 3.4 \* 10\*\*14  
Y\_0 = -0.2  
ALPHA\_0 = radians(-20)  
COUNT\_LAYERS = 1000000  
  
  
def f1(y):  
 return 1.4 + sqrt(0.6 - y\*\*2)  
  
  
def Zf(y):  
 return 28 + 3\*sin(17.951958020513104 \* y)  
  
  
def n(y):  
 return f1(y) \* (1 - ((0.35 \* 10 \*\* 14) / OMEGA) \*\* 2)  
  
  
def n1(y):  
 return f1(y) \* (1-(0.35 \* 10\*\*14) \*\* 2)  
  
  
def new\_angle(n1, n2, sin\_1):  
 return (n1/n2) \* sin\_1  
  
  
def make\_chart(way\_x, way\_y, wave\_x, wave\_y):  
 plt.xlabel("Z")  
 plt.ylabel("Y")  
 plt.title("Траектория волны")  
 plt.ylim(-R, R)  
 plt.plot(way\_x, way\_y)  
 plt.plot(wave\_x, wave\_y)  
 plt.show()  
  
  
def make\_wave():  
 wave\_x = []  
 wave\_y = []  
 x = -R  
 while x < R:  
 x += 0.001  
 wave\_x.append(Zf(x))  
 wave\_y.append(x)  
 return wave\_x, wave\_y  
  
  
def solve():  
 dy = R\*2/COUNT\_LAYERS  
 x = 0  
 y = Y\_0  
 way\_x = [x]  
 way\_y = [y]  
 alpha\_1 = ALPHA\_0  
 n1 = 1  
 n2 = n(y)  
 alpha\_2 = pi/2 - asin(new\_angle(n1, n2, abs(sin(alpha\_1))))  
 length = 0  
 flag\_ref = False  
 if ALPHA\_0 < 0:  
 dy \*= -1  
 while abs(Zf(y) - x) >= 0.05:  
 if flag\_ref or abs(abs(y) - R) < 0.0005:  
 alpha\_2 = alpha\_1  
 dy \*= - 1  
 dx = abs(dy) \* abs(tan(alpha\_2))  
 x += dx  
 y += dy  
 length += sqrt(dx\*\*2+dy\*\*2)  
 way\_x.append(x)  
 way\_y.append(y)  
 n1 = n2  
 alpha\_1 = alpha\_2  
 n2 = n(y)  
 if new\_angle(n1, n2, sin(alpha\_1)) >= 1:  
 flag\_ref = True  
 continue  
 alpha\_2 = asin(new\_angle(n1, n2, sin(alpha\_1)))  
 flag\_ref = False  
 return way\_x, way\_y, length  
  
  
way\_x, way\_y, length = solve()  
wave\_x, wave\_y = make\_wave()  
print(length)  
make\_chart(way\_x, way\_y, wave\_x, wave\_y)