Санкт-Петербургский Государственный Электротехнический

Университет "ЛЭТИ"

кафедра физики

Задание №1 по дисциплине

"Физические основы информационных технологий"

Название: Искривление луча в оптическом канале

|  |  |
| --- | --- |
| Фамилия И.О.: | Кузнецов Н.А. |
| группа: | 1303 |
| Преподаватель: | Альтмарк А. М. |
| Итоговый балл: |  |
| Крайний срок сдачи: | 22.10.23 |

Санкт-Петербург 2023

Условие задания

Найти длину траектории светового луча *S* в прямолинейном дисперсионном оптоволоконном канале, Рис.1, c показателем преломления n1. Оптоволокно окружено средой с показателем преломления n2. Функцию распределения показателя преломления n1(y, ω) можно представить как:

,

где y – поперечная координата, ω – циклическая частота светового луча.

Функцию , функцию Zf(y), описывающую координату z выходного торца волновода, начальный угол ввода луча α в волновод, координату ввода луча в волновод *y*=y0, радиус канала R можно взять в файле FOIT\_IDZ1.xlsx. Все геометрические размеры даются в безразмерных координатах.

Необходимо построить график траектории луча, а также записать ответ *S* в текстовый файл IDZ1\IDZ1.txt. Помимо текстового файла IDZ1.txt в папке IDZ1 должен находиться Word-файл (Pdf-файл) с отчетом, а также файл с кодом (Python, Mathcad, Mathematica).

n2

Y

R

Zf(y)

y0

n1

Z

α

0

n2

-R

n2

Рисунок.1

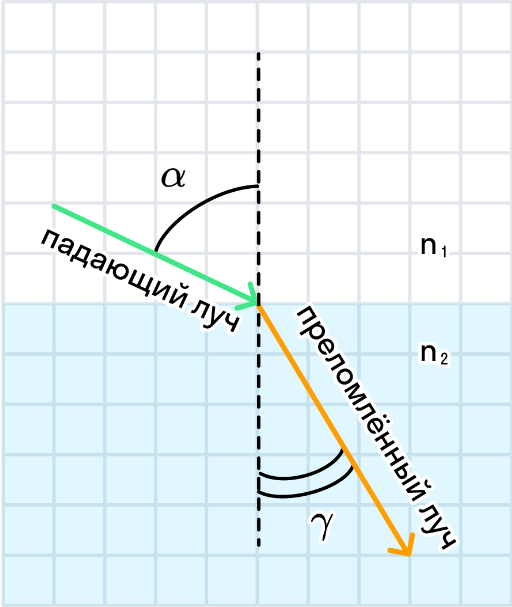
**Таблица с исходными данными**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вар | R | n2 | f1(y) | Zf(y) | ω\*10^14, рад/с | y0 | α, град |
| 6 | 1.4 | 1 | 1.3 - 0.15\*Cos[4\*y] | 12 + 3\*Sin[17.951958020513104\*y] | 3.5 | -0.8 | -20 |

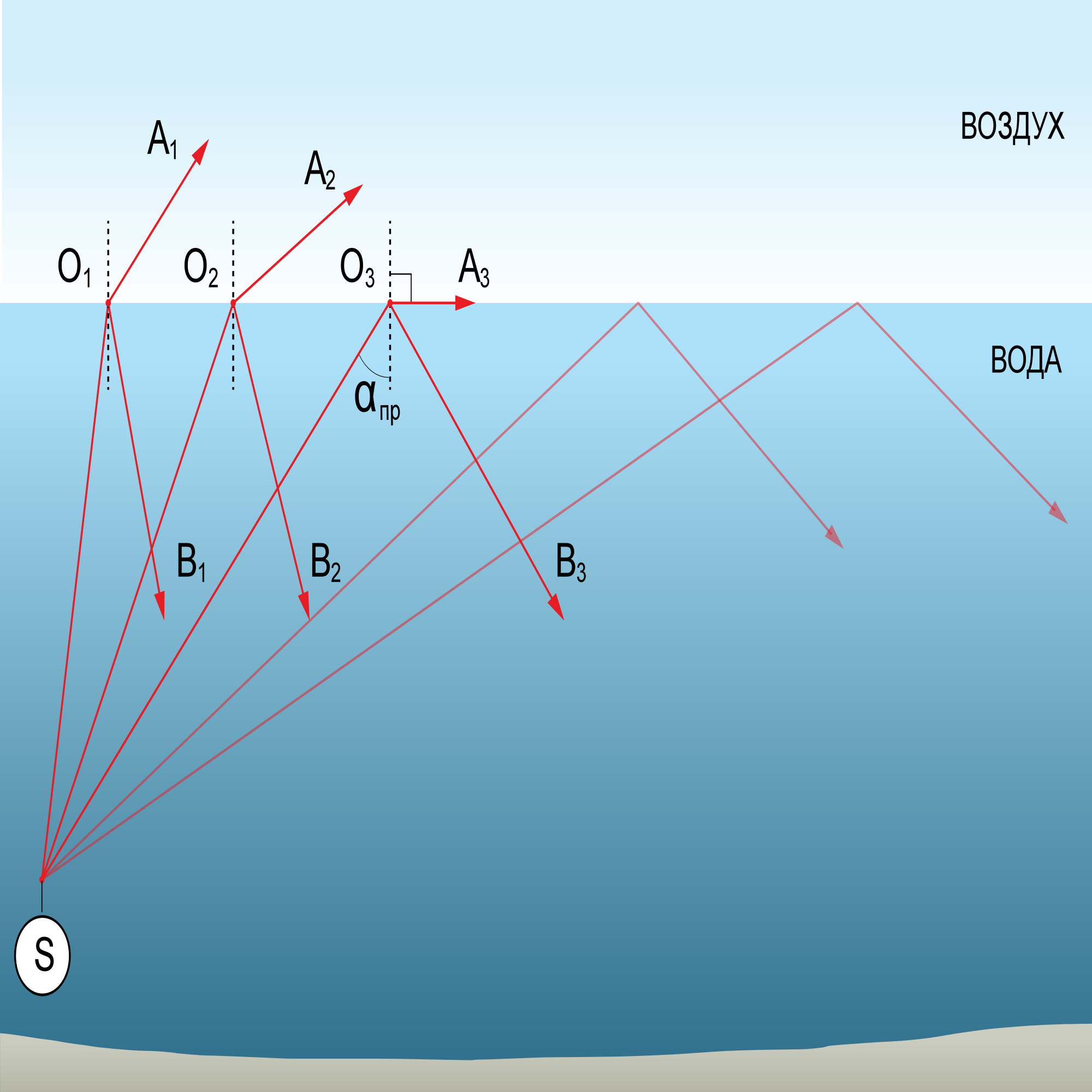
**Основные теоретические положения**

**Закон преломления**

1) Падающий и преломлённый лучи и перпендикуляр, проведённый 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 преломления:



**Полное внутреннее отражение** — внутреннее отражение, при условии, что угол падения превосходит некоторый критический угол. Учитывая, что угол преломления не может превышать 90°, получаем, что при угле падения, синус которого больше отношения меньшего показателя преломления к большему показателю, электромагнитная волна должна полностью отражаться в первую среду.



**Вывод**

С помощью Python была рассчитана траектория хода луча в прямолинейном дисперсионном оптоволоконном канале с показателем

преломления, задаваемым формулой. Полученная траектория была построена на графике, а также была рассчитана её длина.

**ПРИЛОЖЕНИЕ А**

**ПРОГРАММА MAIN.PY**

import matplotlib.pyplot as plt

from math import \*

# Поменять

R = 1.4

n0 = 1

omega = 3.5

y0 = -0.8

alpha0 = radians(-20)

# Поменять

N = 1000 # Количество слоев разбиения

h = 2\*R/N # Шаг разбиения

coefficient = 1 - pow((0.35/omega), 2)

def f1(y):

return 1.3 - 0.15 \* cos(4\*y)

def Zf(y):

return 12 + 3 \* sin(17.951958020513104 \* y)

def n1(y):

return f1(y)\*coefficient

def find\_beta(alpha, n\_alpha, n\_beta):

return asin(sin(alpha)\*n\_alpha/n\_beta)

def is\_reflection(alpha, n\_alpha, n\_beta, layer\_alpha, direction):

if sin(alpha) > n\_beta/n\_alpha or (abs(layer\_alpha) == floor(N/2) and get\_sign(layer\_alpha) == direction):

return True

return False

def get\_sign(x):

if x != 0:

return int(x/abs(x))

return 1

def main():

plt.ylabel("y")

plt.xlabel("z")

plt.ylim(-R, R)

plt.xlim(-1, 35)

n\_arr = dict()

z\_arr = dict()

for i in range(N):

n\_arr[i - floor(N/2)] = n1((i - floor(N/2))\*h + h/2)

z\_arr[i - floor(N/2)] = Zf((i - floor(N / 2))\*h + h/2)

if h\*(i-floor(N/2)) + h/2 < y0 < h\*(i-floor(N/2) + 1) + h/2:

start\_index = i-floor(N/2) + 1

plt.plot([Zf((i - floor(N / 2)) \* h + h / 2), Zf((i - floor(N / 2)) \* h + h / 2)],

[(i - floor(N/2))\*h - h/2, (i - floor(N/2))\*h + h - h/2])

plt.plot([0, Zf((i - floor(N / 2)) \* h + h / 2)], [(i - floor(N / 2)) \* h + h / 2, (i - floor(N / 2)) \* h + h / 2])

n\_arr[ceil(N/2)] = 1

n\_arr[-ceil(N/2)] = 1

alpha\_layer = start\_index

alpha = pi/2 - find\_beta(alpha0, n0, n\_arr[alpha\_layer])

h0 = start\_index\*h - h/2 - y0

path\_len = h0/cos(alpha)

z\_len = h0\*tan(alpha)

z, y = [0], [y0]

direction = -1

n\_alpha = n\_arr[alpha\_layer]

n\_beta = n\_arr[alpha\_layer+direction]

while z\_len < z\_arr[alpha\_layer]:

y.append(alpha\_layer \* h + direction\*h/2)

z.append(z\_len)

if is\_reflection(alpha, n\_alpha, n\_beta, alpha\_layer, direction):

direction = -direction

else:

alpha = find\_beta(alpha, n\_alpha, n\_beta)

alpha\_layer += direction

n\_alpha = n\_arr[alpha\_layer]

n\_beta = n\_arr[alpha\_layer + direction]

path\_len += h / cos(alpha)

z\_len += h \* tan(alpha)

plt.plot(z, y, marker='o')

print(path\_len)

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

main()