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.: | Гирман А.В. |
| 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 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). Для лучшего понимания отчетности смотрите папку “Пример организации яндекс-папки студентов”.

Пример содержания файла IDZ1.txt:

4.53258

n2

Y

R

Zf(y)

y0

n1

Z

α

0

n2

-R

n2

Рисунок.1

Исходные данные

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вар | R | n2 | f1(y) | Zf(y) | ω\*10^14 рад/с. | y0 | α, град. |
| 22 | 1,2 | 1 | 1,5 + 0,3\*Cos[0.8\*y]^2 | 42 + 3\*Sin[17.951958020513104\*y] | 3 | -0.1 | -32 |

Основные теоретические положения

При переходе между средами с разным коэффициентом преломления n, луч преломляется (рис. 1.). Новый угол можно найти с помощью отношения:

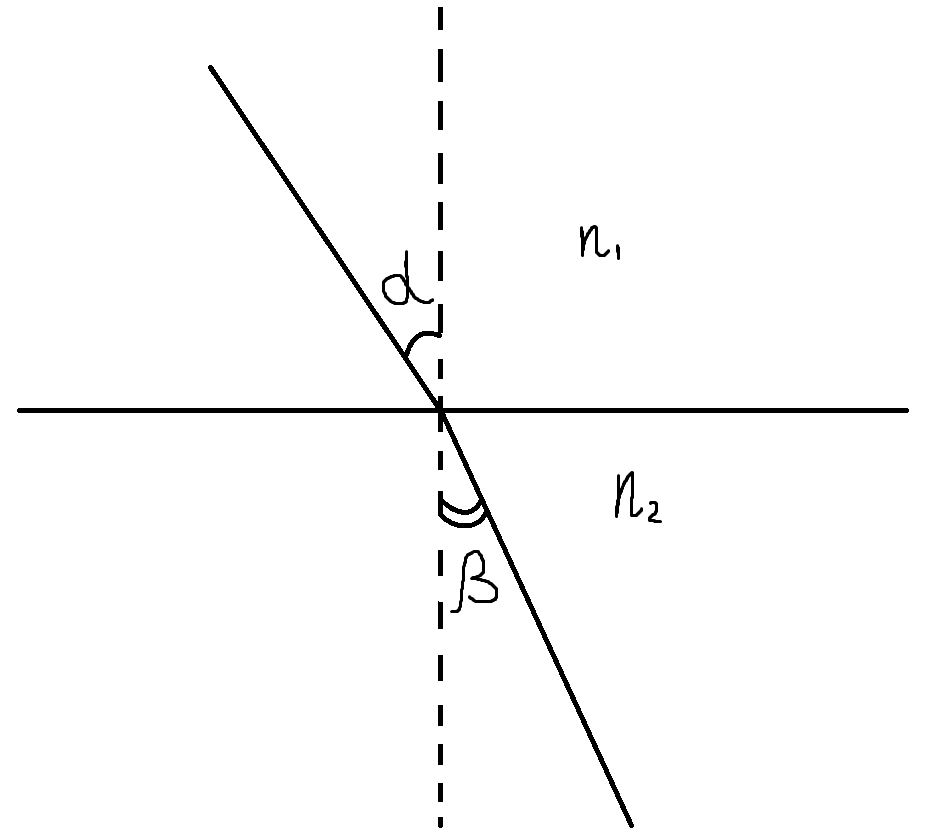


Рисунок 1. Преломления луча

Оптические волноводы служат для ограничения области в которой может распространяться свет.

Показатель преломления , где – скорость света в вакууме, а – скорость света в данной среде.

Луч стремиться к области с наибольшим показателем преломления. Волновод содержит область с увеличенным показателем преломления, удерживая луч.

**Приложение A**

**Программа IDZ1.py**

import numpy as np

import matplotlib.pyplot as p

alpha = -32 # угол в градусах

direct = -1 # направление

y0 = -0.1 # начальная координата по y

z0 = 0 # начальная координата по z

w = 3 \* 10 \*\* 14 # омега

n2 = 1 # коэффициент преломления

R = 1.2 # радиус

epsilon = 0.00001 # шаг

S = 0 # длина траектории

arr\_y = [y0] # ось y

arr\_z = [z0] # ось z

def f1(y):

return 1.5 + 0.3 \* np.cos(0.8 \* y) \*\* 3

def Zf(y):

return 42 + 3 \* np.sin(17.951958020513104 \* y)

def n1(y, w):

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

def algorithm():

global alpha

global direct

global S

# расчёт углов альфа и бета

alpha = np.deg2rad(alpha)

beta = np.arcsin((np.sin(alpha) \* n2) / n1(y0, w))

# новые значения y и z

arr\_y.append(y0 + np.sin(beta) \* epsilon)

arr\_z.append(0 + np.cos(beta) \* epsilon)

# новое значения n1 и n2

new\_n1 = n1(arr\_y[len(arr\_y) - 1], w)

new\_n2 = n1(y0, w)

# новый угол

new\_alpha = np.pi / 2 - beta

new\_beta = np.arcsin((np.sin(new\_alpha) \* new\_n2) / new\_n1)

# прибавляем в длину

S += epsilon

# while arr\_z[len(arr\_z) - 1] <= Zf(arr\_y[len(arr\_y) - 1]):

while abs(Zf(arr\_y[len(arr\_y) - 1]) - arr\_z[len(arr\_z) - 1]) >= 0.001:

if abs(1 - np.sin(new\_beta)) <= 0.000000001:

new\_beta = new\_alpha

direct \*= -1

arr\_y.append(arr\_y[len(arr\_y) - 1] + np.cos(new\_beta) \* epsilon \* direct)

arr\_z.append(arr\_z[len(arr\_z) - 1] + np.sin(new\_beta) \* epsilon)

new\_n1 = n1(arr\_y[len(arr\_y) - 1], w) if R >= abs(arr\_y[len(arr\_y) - 1]) else n2

new\_n2 = n1(arr\_y[len(arr\_y) - 2], w)

new\_alpha = new\_beta

new\_beta = np.arcsin((np.sin(new\_alpha) \* new\_n2) / new\_n1)

S += epsilon

def main():

algorithm()

print(f"Длина траектории светового луча S = {S}")

y = np.arange(-R, R, 0.001)

p.plot(arr\_z, arr\_y, "r", Zf(y), y, "b", [0, 45], [R, R], "b", [0, 45], [-R, -R], "b")

p.show()

if \_\_name\_\_ == '\_\_main\_\_':

main()