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.: | Коренев Д.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 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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вар. | R | n2 | f1(y) | Zf(y) | W\*1014 рад/с | y0 | α  град. |
| 19 | 1.8 | 1 | 1.4 + 0.3\*Cos[0.8\*y^4] | 12 + 3\*Sin[17.951958020513104\*y] | 3.4 | 0.5 | 42 |

Исходные данные

Основные теоретические положения

При переходе между средами с разным коэффициентом преломления n, луч преломляется. Новый угол можно определить по формуле:

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

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

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

**Выполнение работы:**

Выполнение работы можно разделить на несколько частей:

1. Создание координатной плоскости с границами волновода и входящим лучем, представлено на рисунке 2.

Изображение выглядит как снимок экрана, линия, текст, диаграмма

Автоматически созданное описание

Рисунок 2 границы волновода и входящий луч

1. Итеративным способом построить путь луча в волноводе, ход луча представлен на рисунке 3.

Изображение выглядит как линия, диаграмма, График, текст

Автоматически созданное описание

Рисунок 3 путь луча в волноводе

1. Во время итерации подсчитывать длину пути луча, и вывести ее в консоль, представлено на рисунке 4.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 4 длина траектории луча

**Вывод:**

В ходе выполнения работы был визуализирована траектория луча в прямолинейном дисперсионном оптоволоконном канале, высчитана его длина.

**ПРИЛОЖЕНИЕ А**

**ПРОГРАММА MAIN.PY**

# Коренев Данил 1303 вариант 19

from math import cos

from math import sin

from math import asin

from math import radians

from math import degrees

from math import tan

from math import pi

import matplotlib.pyplot as plt

import numpy as np

# график

plt.figure(figsize=(8, 3))

plt.xlim(-2, 17)

plt.ylim(-2, 2)

# границы оптоволокна

plt.plot([0, 9.6545], [-1.8, -1.8], color='blue')

plt.plot([0, 14.34], [1.8, 1.8], color='blue')

plt.plot([0, 0], [-1.8, 1.8], color='blue')

plt.yticks([-1.8, 0, 1.8], ['-1.8', '0', '1.8'])

# оси

plt.axhline(0, color='black', linewidth=0.5)

plt.axvline(0, color='black', linewidth=0.5)

plt.annotate('', xy=(16, 0), xytext=(15, 0), arrowprops=dict(arrowstyle='->', color='black'))

plt.scatter(15.8, 0, s=0.1)

plt.annotate('', xy=(0, 3), xytext=(0, 2), arrowprops=dict(arrowstyle='->', color='black'))

plt.axis('equal')

# луч

plt.plot([-2, 0], [0.5 + tan(radians(42)) \* (-2), 0.5], color='green', linewidth=2)

# config

R = 1.8

n2 = 1

f1 = lambda y: 1.4 + 0.3 \* cos(0.8 \* y \*\* 4)

Zf1 = lambda y: 12 + 3 \* sin(17.951958020513104 \* y)

n1 = lambda y, w: f1(y) \* (1 - ((0.35 \* 10 \*\* 14) / w) \*\* 2)

w = 3.4 \* 10 \*\* 14

y0 = 0.5

alpha = 42

hop = 0.0001

length = 0

direction = 1 if alpha > 0 else -1

# правая граница

Yarray = np.linspace(-1.8, 1.8, 100000)

Zarray = [Zf1(v) for v in Yarray]

plt.plot(Zarray, Yarray, color="blue")

def angleRefeaction(a, nextN, curN):

if (sin(a) \* curN) / nextN > 1:

global direction

direction \*= -1

return a

return asin((sin(a) \* curN) / nextN)

def calc():

global length

global direction

n1Input = n1(y0, w)

curAng = angleRefeaction(radians(alpha), n1Input, n2)

print("Угол β на входе в n1 = %.6f градусов" % degrees(curAng))

curAng = pi / 2 - curAng

Yvalye = []

Zvalue = []

curY = y0

curZ = 0

curN = n1Input

while curZ < Zf1(curY):

Yvalye.append(curY)

Zvalue.append(curZ)

nextY = curY + cos(curAng) \* hop \* direction

nextZ = curZ + abs(sin(curAng)) \* hop

if R >= abs(nextY):

nextN = n1(nextY, w)

else:

nextN = n2

nextAng = angleRefeaction(curAng, nextN, curN)

length += hop

curY = nextY

curZ = nextZ

curAng = nextAng

curN = nextN

plt.plot(Zvalue, Yvalye, color="red")

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

calc()

print(length)

plt.show()