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.: | Иевлев Е.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:

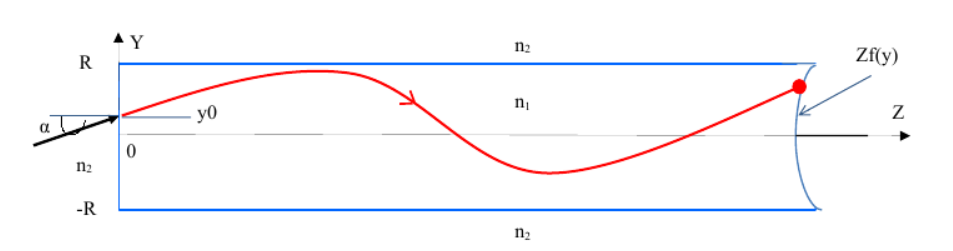
4.53258

Рисунок 1.

 Исходные данные

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вар | R | n2 | f1(y) | Zf(y) | w\*10^14 рад/с. | y0 | a, град. |
| 18 | 1,8 | 1 | 1,4 + 0,3\*Cos[0,5\*y^4] | 12 + 3\*Sin[17.951958020513104\*y] | 3,3 | -0.4 | -42 |

Основные теоретические положения

*Закон преломления света*

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



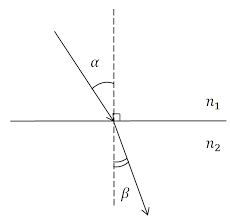


Рисунок 2. Преломления луча

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

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

*Закон* *полного внутреннего отражения*

При переходе световых лучей из оптически более плотной среды в оптически менее плотную наблюдается явление — полное внутреннее отражение.

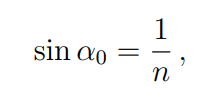
По мере увеличения угла падения прослеживается та же закономерность: всё большая доля энергии падающего луча достаётся отражённому лучу, и всё меньшая — преломлённому лучу.

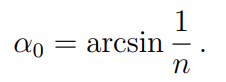
При дальнейшем увеличении угла падения преломлённый луч и подавно будет отсутствовать.

Среда не выпускает наружу лучи с углами падения, равными или превышающими некоторое значение α0 — все такие лучи целиком отражаются назад в среду. Угол α0 называется предельным углом полного отражения.

Величину α0 легко найти из закона преломления. Имеем:



Но sin 90◦ = 1, поэтому

откуда

*Принцип работы оптоволокна*

Важнейшим техническим применением полного внутреннего отражения является волоконная оптика. Световые лучи, запущенные внутрь оптоволоконного кабеля (световода) почти параллельно его оси, падают на поверхность под большими углами и целиком, без потери энергии отражаются назад внутрь кабеля. Многократно отражаясь, лучи идут всё дальше и дальше, перенося энергию на значительное расстояние.

 Ход работы

На языке Python и при помощи библиотек numpy и matplotlib просчитаем траекторию хода луча. Для этого сначала введем функции и значения переменных из дано. Затем с помощью метода итераций будем вычислять координаты следующей точки и добавлять эти значения в массив, на основе которого будет выполнено построение графика.

 Кроме того, отдельно просчитаем значения функции Zf(y) на промежутке от -R до R, для того чтобы построить поверхность торца волновода. Построим графики по отдельности и совместим их на общей координатной плоскости.

Вывод

С помощью Python и библиотеки numpy была рассчитана траектория хода луча в прямолинейном дисперсионном оптоволоконном канале с показателем преломления, задаваемым следующей формулой:

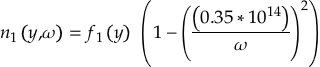


График данной траектории был построен при помощи библиотеки matplotlib, а также была рассчитана длина траектории.

**Приложение A**

**Программа IDZ1.py**

import numpy as np

import matplotlib.pyplot as plt

R = 1.8

n2 = 1

def f1(y):

return 1.4 + 0.3 \* np.cos(0.5 \* (y \*\* 4))

def Zf(y):

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

omega = 3.3 \* 10 \*\* 14

y0 = -0.4

alpha = -42

def getN1(y):

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

def getStartSin():

n1\_0 = getN1(y0)

sinBeta = np.sin(np.pi / 2 - np.arcsin(np.sin(abs(alpha)) \* n2 / n1\_0))

return sinBeta, np.sign(alpha)

hop = 0.001

currentY = y0

currentZ = 0

currentN = getN1(currentY)

currentSin, direction = getStartSin()

totalLength = 0

def makeStep():

global currentY

global currentZ

global currentN

global currentSin

global totalLength

global direction

newY = currentY + np.sqrt(1 - currentSin \*\* 2) \* hop \* direction

newZ = currentZ + np.abs(currentSin) \* hop

newN = getN1(newY) if R >= abs(newY) else n2

newSin = (currentSin \* currentN) / newN

totalLength += hop

currentY = newY

currentZ = newZ

if (abs(newSin) >= 1):

direction \*= -1

return

currentN = newN

currentSin = newSin

pointsByY = []

pointsByZ = []

def start():

while (currentZ < Zf(currentY)):

pointsByY.append(currentY)

pointsByZ.append(currentZ)

makeStep()

print(totalLength)

t1 = np.arange(-R, R, 0.001)

plt.plot(pointsByZ, pointsByY, Zf(t1), t1, [0, 15], [R, R], 'g', [0, 15], [-R, -R], 'g')

plt.show()

start()