Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №2 по дисциплине

"Физические основы информационных технологий"

Название: Искривление луча в оптическом канале

|  |  |
| --- | --- |
| Фамилия И.О.: | Калмак Д.А. |
| группа: | 0303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 6.11.22

.

Санкт-Петербург 2022

Условие задания

Луч белого света попадает в оптоволокно, показатель преломления которого зависит от поперечной координаты “*y*” и от частоты “ω”:

.

Вследствие зависимости скорости света от частоты возникает явление дисперсии, выражающееся в разложении белого света в спектр, Рис.1. Найти угловое расширение β светового луча для видимого диапазона частот на выходе из оптоволокна. Расширение луча можно вычислить по граничным частотам видимого диапазона (красный цвет , фиолетовый цвет ). Функцию распределения показателя преломления по поперечной координате *Y*, начальный угол ввода луча α в волновод, длину канала *L*, диаметр канала *D* можно взять в таблице 1. Ввод луча осуществляется из центральной части канала с координатой *y*=0. Все координаты и геометрические размеры данысантиметрах.

Луч падает из среды с показателем преломления *n*0 под углом α. Показатель преломления *n*0 не зависит от частоты и координаты и определяется по формуле

Необходимо построить графики траектории красного и фиолетового лучей, а также записать ответ βв градусах в текстовый файл LR2\result.txt. Помимо текстового файла result.txt в папке LR2 должен находиться Word-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 result.txt:

4.53258

n2

*Y*

*n0*

β

n1

2*D*

α

Z

0

*L*

n2

Рисунок 1 - Пример

Вариант №9.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *L* | *D* | *n2* | *f1*(*y*) | *n0* | α, градусы |
| 18 | 0.8 | 1 | 1.4 + 0.12\*Cos[3\*y] | *f1*(0) | 40 |

**Выполнение работы.**

Написаны функции f\_1(y) для вычисления функции распределения показателя преломления и n\_1\_function(y, w) для вычисления показателя преломления, который зависит от поперечной координаты “*y*” и от частоты “ω”.

Функция calculate\_sin\_gamma() вычисляет синус угла преломления с помощью закона Снеллиуса:

Где – синус угла падения, – синус угла преломления, – показатели преломления сред. В случае, когда синус угла преломления больше единицы, то есть в случае полного отражения, направление по оси y меняется, а синус угла становится равным синусу угла падения. Функция calculate\_x() суммирует значение x с предыдущего шага с дельтой x, которая получается из теоремы Пифагора для шага длины (шаг взят ) и дельтой y.

Функция processing() обрабатывает координаты хода луча. Входной луч рассматривается перед циклом. Рассчитывается угол преломления по закону Снеллиуса, в списки значений x, y вносится 0. Рассчитывается дельта для y с помощью синуса угла преломления и шага длины. Новое значение y вносится в список значений y. Рассчитывается дельта x по теореме Пифагора для шага длины и дельты y. Значение вносится в список значений x. За угол падения берется разность между 90 градусами и углом преломления. Запускается цикл. Каждый шаг длины вычисляется синус угла преломления, вычисляется шаг x с помощью синуса угла преломления и шага длины, вычисляется шаг y по теореме Пифагора для шага длины и дельты x, переопределяется синус угла падения. Цикл вычислений продолжается до тех пор, пока x не станет больше L. Функция возвращает списки значений x, y и значение угла в градусах.

Было найдено угловое расширение β светового луча для видимого диапазона частот (использовались граничные частоты: красного и фиолетового цветов) в градусах 0.9658748330172102 и записано в файл result.txt.

Построены графики траектории красного и фиолетового лучей с использованием библиотеки matplotlib.pyplot (см. рис. 2). Рисунок представлен в Приложении Б.

Разработанный код представлен в Приложении А.

**ПРИЛОЖЕНИЕ А  
ИСХОДНЫЙ КОД ПРОГРАММЫ**

import math  
import matplotlib.pyplot as plt  
from matplotlib.patches import Rectangle  
  
  
def f\_1(y):  
 return 1.4 + 0.12 \* math.cos(3 \* y)  
  
  
def n\_1\_function(y, w):  
 return f\_1(y) \* (1 - (0.35 \* 1e14 / w) \*\* 2)  
  
  
def calculate\_sin\_gamma(sin\_alpha, n\_1, n\_2, direction):  
 sin\_gamma = n\_1 \* sin\_alpha / n\_2  
 if sin\_gamma > 1:  
 direction \*= -1  
 sin\_gamma = sin\_alpha  
 return sin\_gamma, direction  
  
  
def calculate\_x(x\_prev, dx):  
 return x\_prev + dx  
  
  
def processing(L, D, n\_0, alpha, w):  
 sin\_alpha = math.sin(alpha)  
 n\_1 = [n\_1\_function(0, w)]  
 direction = 1  
 sin\_gamma, direction = calculate\_sin\_gamma(sin\_alpha, n\_0, n\_1[0], direction)  
 y = [0]  
 step\_line = 0.0001  
 y.append(y[0] + sin\_gamma \* step\_line)  
 x = [0]  
 i = 1  
 x.append(step\_line \*\* 2 - y[1] \*\* 2)  
 sin\_alpha = math.sin(math.radians(90) - math.asin(sin\_gamma))  
 # print(math.radians(90), math.asin(sin\_gamma), sin\_gamma)  
 new\_y = 0  
 new\_x = 0  
 while new\_x <= L:  
 n\_1.append(n\_1\_function(y[i], w))  
 sin\_gamma, direction = calculate\_sin\_gamma(sin\_alpha, n\_1[i-1], n\_1[i], direction)  
 dx = sin\_gamma \* step\_line  
 new\_x = calculate\_x(x[i], dx)  
 x.append(new\_x)  
 dy = math.sqrt(step\_line \*\* 2 - dx \*\* 2)  
 new\_y = y[i] + direction \* dy  
 if abs(new\_y) > D:  
 direction \*= -1  
 new\_y = y[i] + direction \* dy  
 y.append(new\_y)  
 sin\_alpha = sin\_gamma  
 i += 1  
 alpha = math.degrees(math.atan((y[len(y) - 1] - y[len(y) - 2]) / (x[len(x) - 1] - x[len(x) - 2])))  
 return x, y, alpha  
  
  
L = 18  
D = 0.8  
n\_2 = 1  
n\_0 = f\_1(0)  
alpha = math.radians(40)  
w\_red = 3.3 \* 1e14  
w\_purple = 6.5 \* 1e14  
x1, y1, angle1 = processing(L, D, n\_0, alpha, w\_red)  
x2, y2, angle2 = processing(L, D, n\_0, alpha, w\_purple)  
beta = abs(angle1 - angle2)  
print(beta)  
file = open('./result.txt', 'w')  
file.write(str(beta))  
file.close()  
fig, ax = plt.subplots()  
ax.plot(x1, y1, color="red")  
ax.plot(x2, y2, color="purple")  
ax.add\_patch(Rectangle((0, -D), L, 2 \* D, fill=False))  
fig.savefig('graphics.jpeg')

**ПРИЛОЖЕНИЕ Б  
РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

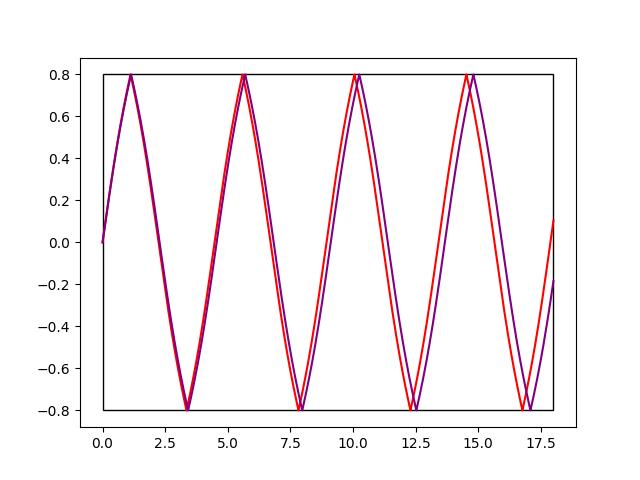


Рисунок 2 – Траектории красного и фиолетового лучей