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.: | Самохин К.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). 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

Вариант 10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R | n2 | f1(y) | Zf(y) | Ω \* 10^14, рад/с | y0 | α, град |
| 0.8 | 1 | 1.4 + 0.12\*Cos[6\*y] | 28 + 3\*Sin[17.951958020513104\*y] | 3.2 | -0.4 | -30. |

Основные теоретические положения

При переходе между средами с разными коэффициентами преломления, луч света преломляется, и угол падения луча изменяется. Связь углов падения и коэффициентов преломления выражается следующей формулой:

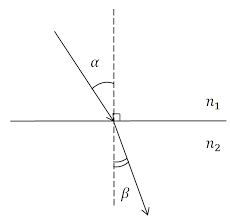


Рисунок. 2

Коэффициент преломления вычисляется как отношение скорости движения света в вакууме и скорости света в рассматриваемой среде:

**Оптические волноводы** - это пространственно-неоднородные структуры для направления света. Оптический волновод служит для ограничения области пространства, в которой может распространяться свет. Применяются в качестве передающей среды в системах оптической связи.

**Внутреннее отражение** — явление отражения электромагнитных или звуковых волн от границы раздела двух сред при условии, что волна падает из среды, где скорость её распространения меньше (в случае световых лучей это соответствует большему показателю преломления).

**Полное внутреннее отражение** — внутреннее отражение, при условии, что угол падения превосходит некоторый критический угол. При этом падающая волна отражается полностью.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Файл main.py:

import numpy as np

import matplotlib.pyplot as plt

alpha = np.deg2rad(-30)

vert\_direct = -1

y0 = -0.4

z0 = 0

omega = 3.2 \* 10 \*\* 14

n2 = 1

R = 0.8

eps = 0.00001

S = 0

arrY = [y0]

arrZ = [z0]

def f1(y):

return 1.4 + 0.12 \* np.cos(6 \* y)

def Zf(y):

return 28 + 3 \* np.sin(17.951958020513104 \* y)

def n1(y, w):

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

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

arrY.append(y0 + np.sin(beta) \* eps)

arrZ.append(0 + np.cos(beta) \* eps)

next\_n = n1(arrY[len(arrY) - 1], omega)

cur\_n = n1(y0, omega)

alpha = np.pi / 2 - beta

sinB = (np.sin(alpha) \* cur\_n) / next\_n

sinB = np.clip(sinB, -1, 1) # ограничение аргумента в [-1, 1]

beta = np.arcsin(sinB)

S += eps

while abs(Zf(arrY[len(arrY) - 1]) - arrZ[len(arrZ) - 1]) >= 0.001:

if abs(1 - np.sin(beta)) <= 0.000000001:

beta = alpha

vert\_direct \*= -1

arrY.append(arrY[len(arrY) - 1] + np.cos(beta) \* eps \* vert\_direct)

arrZ.append(arrZ[len(arrZ) - 1] + np.sin(beta) \* eps)

next\_n = n1(arrY[len(arrY) - 1], omega) if R >= abs(arrY[len(arrY) - 1]) else n2

cur\_n = n1(arrY[len(arrY) - 2], omega)

alpha = beta

sinB = (np.sin(alpha) \* cur\_n) / next\_n

sinB = np.clip(sinB, -1, 1)

beta = np.arcsin(sinB)

S += eps

print(S)

y = np.arange(-R, R, 0.00001)

plt.plot(arrZ, arrY, "r", Zf(y), y, "b", [0, 35], [R, R], "b", [0, 35], [-R, -R], "b")

plt.show()