

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.: Куклина Ю.Н.

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, с показателем преломления n_1 . Оптоволокно окружено средой с показателем преломления n_2 . Функцию распределения показателя преломления $n_1(y, \omega)$ можно представить как:

$$n_1(y, \omega) = f_1(y) \left(1 - \left(\frac{(0.35 \cdot 10^{14})}{\omega} \right)^2 \right),$$

где y – поперечная координата, ω – циклическая частота светового луча.

Функцию $f_1(y)$, функцию $Zf(y)$, описывающую координату z выходного торца волновода, начальный угол ввода луча α в волновод, координату ввода луча в волновод $y=y_0$, 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

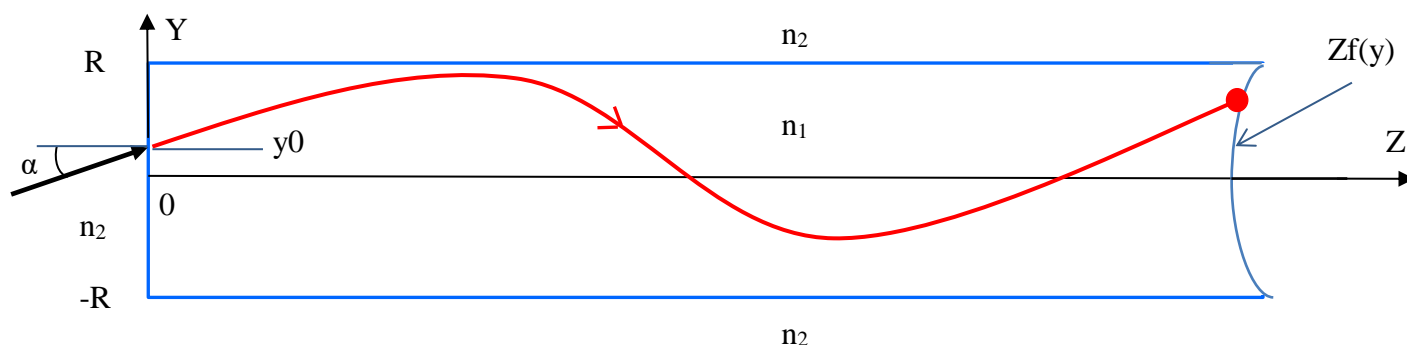


Рисунок.1

Вариант 7

R	n2	f1(y)	Zf(y)	$\omega \cdot 10^{14}$, рад/с	y0	α , град
0.8	1	$1.3 - 0.15 \cdot \cos[4 \cdot y]$	$12 + 3 \cdot \sin[17.951958020513104 \cdot y]$	3.6	0.3	20

Теоретические положения

Оптические волноводы - это пространственно-неоднородные структуры для направления света. Оптический волновод служит для ограничения области пространства, в которой может распространяться свет.

Луч, проходя между средами с разными показателями преломления, преломляется, меняя свою траекторию, угол преломления можно найти по формуле:

$$\frac{\sin(\alpha)}{\sin(\beta)} = \frac{n_2}{n_1}$$

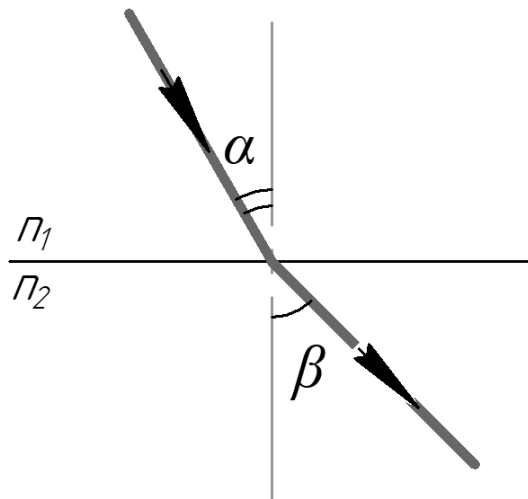


Рис1. Преломление света

Показатель преломления - это физическая величина, которая характеризует, как свет распространяется в конкретной среде в сравнении с его распространением в вакууме.

$$n = c / v$$

где:

c - скорость света в вакууме

v - скорость света в среде

Полное внутреннее отражение - это оптический эффект, который происходит при переходе света из более плотной среды в менее плотную среду под определенным углом падения (угол преломления становится больше 90 градусов). В этом случае свет полностью отражается от границы раздела двух сред и не проникает в менее плотную среду.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import numpy as np
import matplotlib.pyplot as plt

alpha = 20
direction = 1
y_0 = 0.3
z_0 = 0
w = 3.6 * 10 ** 14
n2 = 1
R = 0.8
h = 0.00001
traj_len = 0
arr_y = [y_0]
arr_z = [z_0]

def f1(y):
    return 1.3 - 0.15 * np.cos(4 * y)

def Z_f(y):
    return 12 + 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 direction
    global traj_len

    alpha = np.deg2rad(alpha)
    beta = np.arcsin((np.sin(alpha) * n2) / n1(y_0, w))

    arr_y.append(y_0 + np.sin(beta) * h)
    arr_z.append(0 + np.cos(beta) * h)

    new_n1 = n1(y_0, w)
    new_n2 = n1(arr_y[len(arr_y) - 1], w)

    # новый угол
    new_alpha = np.pi / 2 - beta

    angle = (np.sin(new_alpha) * new_n1) / new_n2
```

```

angle = np.clip(angle, -1, 1)
new_beta = np.arcsin(angle)

traj_len += h

while arr_z[-1] <= Z_f(arr_y[-1]):
    if 1 - abs(np.sin(new_beta)) <= 0.000001:
        new_beta = new_alpha
        direction *= -1

    arr_y.append(arr_y[-1] + np.cos(new_beta) * h *
direction)
    arr_z.append(arr_z[-1] + np.sin(new_beta) * h)

    if abs(arr_y[-1]) >= R:
        new_n2 = n2
    else:
        new_n2 = n1(arr_y[-1], w)

    new_n1 = n1(arr_y[-2], w)

    new_alpha = new_beta

    arg = (np.sin(new_alpha) * new_n1) / new_n2
    arg = np.clip(arg, -1, 1)
    new_beta = np.arcsin(arg)

    traj_len += h

```

```

def main():
    algorithm()
    print(f"Длина траектории луча = {traj_len}")
    y = np.arange(0, R, 0.0001)
    plt.plot(arr_z, arr_y, "red")
    plt.plot(Z_f(y), y, "blue")
    plt.show()

```

```

if __name__ == '__main__':
    main()

```