К явлению радуги

Работу выполнили: А. В. Корнеенков Д. Ю. Бычков
Научный руководитель: А.С. Байгашов

Аннотация
В работе проведено исследование движения луча света в капле воды. Получены результаты, показывающие как происходит процесс преломления света. Был смоделирован ход луча и угол отклонения света, в результате чего получена демонстрация разделения пучка фотонов на несколько лучей. Построена зависимость максимального угла отклонения света шарообразной каплей жидкости от показателя преломления жидкости для заданной длины волны.

Введение
Явление радуги привлекает внимание людей с незапамятных времён. С развитием естественных наук оно перешло из категории красивых, но необъяснимых природных феноменов в разряд иллюстраций физических законов, а данном случае – преломления и разложения солнечного света ввиду его волновой природы. Пожалуй, радуга является самым наглядным примером справедливости корпускулярно-волнового дуализма световых лучей. Но можно ли смоделировать процесс появления радуги средствами численного решения уравнений?
В рамках настоящей работы мы постараемся дать ответ на этот вопрос, рассмотрев движение луча света в капле воды, где он претерпевает двукратные отражения и преломления. В ходе этого процесса даёт о себе знать волновая природа света – вместо движения луча по законам геометрической оптики, он расщепляется по спектру. Для моделирования этого явления используется язык программирования Python и его открытые библиотеки, позволяющие рассмотреть спектральное разложение как преобразование одного луча к ограниченному набору разноцветных лучей. Целью работы является отыскание численного решения задачи о движении луча света в капле воды.

Постановка задачи
На шарообразную каплю воды падает параллельный световой пучок. Необходимо определить зависимость угла отклонения световых лучей от прицельного параметра ρ и написать программу, которая бы строила ход луча в капле в результате преломления и рассчитывала бы угол отклонения света. Кроме того, представляется важным изучить зависимость максимального угла отклонения света шарообразной каплей жидкости от показателя преломления жидкости для данной длины волны.

Для начала введём прицельный параметр ρ и углы преломления, которые вычисляются по формулам:

$$\alpha = arctg\left(\frac{\rho}{\sqrt{R^2-\rho^2}}\right), \qquad \sin\beta = \frac{\sin\alpha}{n}, \qquad \cos\beta = \sqrt{1-\sin\beta^2},$$

$$\beta = arctg\left(\frac{\sin\beta}{\cos\beta}\right) = arctg\left(\frac{\sin\beta}{\sqrt{1-\sin\beta^2}}\right), \qquad \varphi = 4\beta - 2\alpha.$$

Далее необходимо создать программу, отрисовывающую траекторию луча света в соответствии с введёнными выше углами преломления и отражения внутри капли воды.

Начальные условия

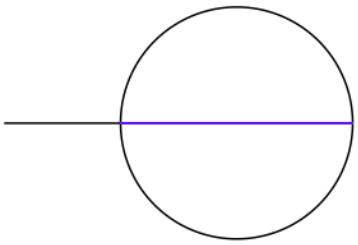
Для решения поставленной задачи необходимо определить следующие начальные условия:

- Показатель преломления. Рассмотрим несколько вариантов с небольшим шагом (например, 0.01) в интервале от 1.30 до 1.41
- Прицельный параметр. Аналогично рассмотрим несколько вариантов: 0; 0.1; 0.5; 0.9; 1.

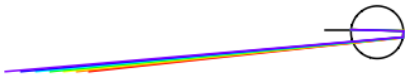
Результаты моделирования

В результате численного моделирования нескольких случаев, были получены следующие картины распространения света в капле воды:

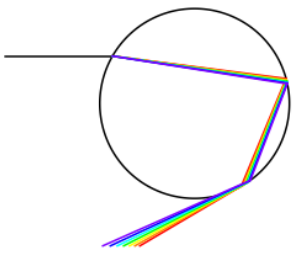
p = 0



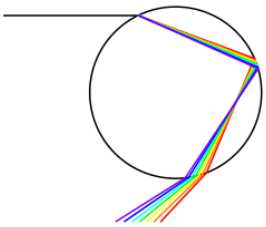
p = 0.1



p = 0.5



p = 0.9



p = 1

Приведённые картины распространения лучей света показывают, как меняется траектория его движения капле воды. Можно заметить, что с ростом прицельного параметра меняется картина преломления, всё сильнее и сильнее разлагая исходный луч на спектральные компоненты.

Заключение

Проведённое исследование наглядно продемонстрировало возможности языка Python и его библиотек по численному моделированию движения светового потока в преломляющем его теле. Было показано, что изменение прицельного параметра меняет картину отражения и преломления. Так, при его нулевом значении процесса преломления не происходит вовсе, а с его ростом двойное отражение и преломление луча света в капле воды обеспечивает всё более и более заметное разложение светового пучка на спектральные компоненты.

Приложение

Листинг кода решения задач:

```
# Начальные параметры R0-радиус, менять нельзя
# Прицельный параметр ro должен от 0 и до 1
R0 = 1
ro = 1


# Функция расчитывающая углы
def func(R=R0, ro0=1, n0=1):
    alpha = np.arctan(ro0/np.sqrt(R**2 - ro0**2))
    k = np.sin(alpha)/n0
    beta = np.arctan(k/np.sqrt(1 - k**2))
    phi = 4*beta - 2*alpha
    return phi, beta, alpha

# Массив значений коэффициента преломления
n = np.arange(1.30, 1.41, 0.016)

# Построения падающего луча
xl = []
yl = []

xl0 = -2
yl0 = ro

xl.append(xl0)
yl.append(yl0)

xl1 = -np.sqrt(R0 - ro**2)
yl1 = ro

xl.append(xl1)
yl.append(yl1)
```

Построение отражённых лучей:

```
# Построения первых отражённый лучей
xlr = []
ylr = []

for i in n:
    h = np.tan(np.pi - func(R0, ro, n0=i)[2] + func(R0, ro, n0=i)[1])

    xlr0 = - np.sqrt(R0 - ro**2)
    ylr0 = ro
    ylr1 = (ylr0 - h*xlr0 - np.sqrt((-ylr0 + h*xlr0)**2 - (1 + h**2)*(ylr0**2 + h**2*xlr0**2
                                    - 2*ylr0*h*xlr0 - h**2*R0**2)))/(1 + h**2)
    xlr1 = np.sqrt(R0 - ylr1**2)

    xlr.append(xlr0)
    ylr.append(ylr0)
    xlr.append(xlr1)
    ylr.append(ylr1)

# Построения вторых отражённый лучей
XL = []
YL = []
k = 1

for i in n:
    h = np.tan(3*func(R0, ro, n0=i)[1] - func(R0, ro, n0=i)[2])

    xlr0 = xlr[k]
    ylr0 = ylr[k]
    ylr1 = (ylr0 - h*xlr0 - np.sqrt((-ylr0 + h*xlr0)**2 - (1 + h**2)*(ylr0**2 + h**2*xlr0**2
                                    - 2*ylr0*h*xlr0 - h**2*R0**2)))/(1 + h**2)
    xlr1 =np.sqrt(R0 - ylr1**2)

    XL.append(xlr0)
    YL.append(ylr0)
    XL.append(xlr1)
    YL.append(ylr1)
    k += 2

# Построения третьих отражённый лучей
XL1 = []
YL1 = []
k = 1

for i in n:
    h = np.tan(func(R0, ro, n0=i)[0])

    xlr0 = XL[k]
    ylr0 = YL[k]
    ylr1 = -1.5
    xlr1 = (ylr1 - ylr0)/h + xlr0

    XL1.append(xlr0)
    YL1.append(ylr0)
    XL1.append(xlr1)
    YL1.append(ylr1)
    k += 2

```