

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»

Институт информатики, математики и электроники
Факультет информатики
Кафедра технической кибернетики

**Отчет по лабораторной работе № 1 по курсу
«Математическое моделирование оптических систем»**

Выполнила: Архипова Д.В.
Проверил: Дегтярёв С.А.
Группа: 6130-010402D

САМАРА, 2019

Расчет оптических характеристик объектива методом трассировки лучей

В ходе лабораторной работы №1 было разработано программное обеспечение на языке python версии 3.7.3 реализующее поиск фокусного расстояния линзы с заданными геометрическими параметрами, а также поиск геометрических параметров линзы с заданным фокусным расстоянием. Вычисление фокусного расстояния проводилось с использованием следующих методов: минимизации среднеквадратичного отклонения точки положения луча от центра, минимизации среднеквадратичного отклонения оптического пути и отклонения луча от центра, минимизации среднеквадратичного отклонения оптического пути луча от среднего значения. Вычисление геометрических параметров вычислялось методом перебора.

Здесь и далее направление перехода из одной среды в другую с показателями преломления n_1 и n_2 соответствует направлению нормали в точке пересечения луча и поверхности.

В работе приняты следующие условные обозначения: a, b – радиусы кривизн линзы.

Ниже представлены результаты работы демонстративного примера.

Пример 1

Параметры:

- $a = 0.8$
- $b = 3$
- $n_1 = 1$
- $n_2 = 1.3$

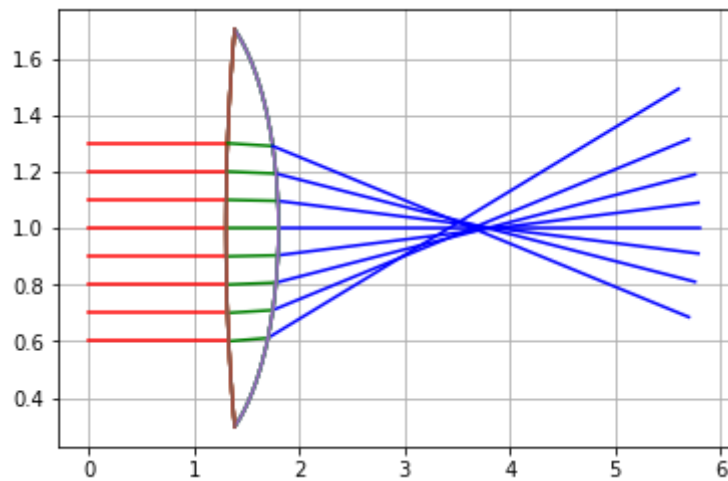


Рисунок 1 – Результат примера 1; красным цветом обозначены падающие лучи, зеленым – преломленные в линзе, синим – преломленные после линзы

Для данного примера были получены следующие значения фокусного расстояния:

- путем минимизации среднеквадратичного отклонения оптического пути луча от среднего значения – 3.589;
- путем минимизации среднеквадратичного отклонения точки положения луча от центра – 3.643;
- путем минимизации среднеквадратичного отклонения оптического пути и отклонения луча от центра – 3.577.

Пример 2

Параметры:

- $a = 1$
- $b = 3$
- $n_1 = 1$
- $n_2 = 1.3$

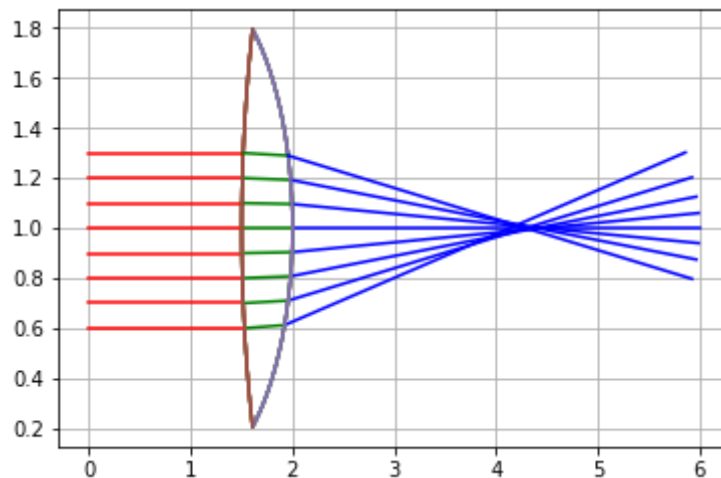


Рисунок 2 – Результат примера 2; красным цветом обозначены падающие лучи, зеленым – преломленные в линзе, синим – преломленные после линзы

Этому примеру соответствуют следующие значения фокусного расстояния:

- путем минимизации среднеквадратичного отклонения оптического пути луча от среднего значения – 4.261;
- путем минимизации среднеквадратичного отклонения точки положения луча от центра – 4.292;
- путем минимизации среднеквадратичного отклонения оптического пути и отклонения луча от центра – 3.253.

ПРИЛОЖЕНИЕ

Исходный код программы

Расчет ско оптического пути от среднего значения

```
def give_l3 (t1,t2,h,n_r):
    return (h-t1*n_r[0]-t2*n_r[1])/n_r[2]
def give_r3 (l3,r2,e):
    return sum (r2 , mult(e,l3))
def give_r0 (R3, count):
    summ = [0,0]
    for i in range (count):
        summ =sum(summ, R3[i])
    summ = mult(summ, 1/count)
    return summ
def f (r0,r3):
    return scal_mult(sub(r0,r3), sub(r0,r3))
def sco (h):
    summ = 0
    L3 = []
    R3 = []
    for i in range (count):
        L3.append(give_l3(T1[i],T2[i],h,n_r))
    for i in range (count):
        R3.append(give_r3(L3[i],R2[i],e[i]))
    r0 = give_r0(R3,count)
    for i in range (count):
        summ = summ + f(r0,R3[i])
    return math.sqrt(summ/count)
def focus (h):
    summ = 0
    L3 = []
    R3 = []
    for i in range (count):
        L3.append(give_l3(T1[i],T2[i],h,n_r))
    for i in range (count):
        R3.append(give_r3(L3[i],R2[i],e[i]))
    r0 = give_r0(R3,count)
    return r0
```

Расчет ско точки положения луча от центра

```
def fun (z):  
  
    t = (scal_mult(n,sub([z,1,0],cr)))/(scal_mult(n,e))  
  
    cross_point_2 = sum(cr, (mult(e,t)))  
  
    return math.sqrt(((cross_point_2[1]-y0)**2+(cross_point_2[2]-x0)**2)/n_r[1])
```

Расчет ско пути и отклонения луча от центра

```
def give_L (z):  
    L=[]  
    summ1 = 0  
    for i in range (count1):  
        t = (scal_mult(n,sub([z,1,0],cros_p[i])))/(scal_mult(n,e_ref[i]))  
        summ1 = summ1 + L1[i] + L2[i] + t*n_r[2]  
        L.append( L1[i] + L2[i] + t*n_r[2])  
    return summ1/count1, L  
  
def function (z):  
    l_s, L = give_L(z)  
    s=0  
    for i in range (count1):  
        t = (scal_mult(n,sub([z,1,0],cros_p[i])))/(scal_mult(n,e_ref[i]))  
        cross = sum(cros_p[i], mult(e_ref[i],t))  
        s = s + math.sqrt(((cross[1]-y0)**2+(l_s-L[i])**2+(cross[2]-x0)**2)/(2*count1))  
    return s
```

Расчет фокуса методом перебора

```
def ff (a,b):  
    return (1/(n_r[1]-n_r[0])*(a*b)/(a+b))  
  
def give_focus (f):  
    a = np.arange(0.1, 10,0.1)  
    b = np.arange(0.1, 10,0.1)  
    for i in a:  
        for j in b:  
            if ff(i,j) == f:  
                return i,j
```