Вариационные методы формирования математических моделей

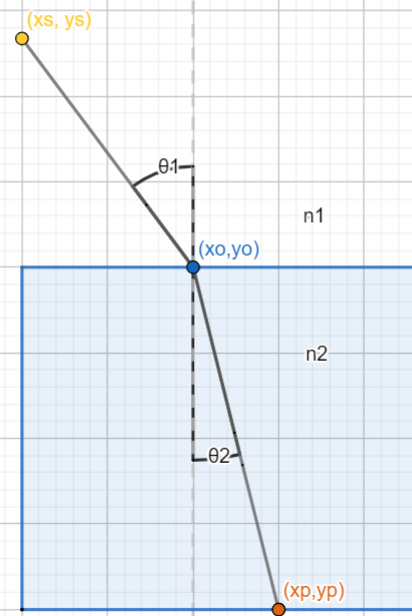
Объект исследования задачи

Луч света, проходящий через среды с различным коэффициентом преломления.

Задача

Закон преломления

1. Определять точность попадания света в приемник при различных начальных параметрах
2. Вычислять значение вариационной переменной, при котором свет попадает в приемник
3. Содержательная постановка задачи



n1 – показатель преломления среды, из которой свет падает на границу раздела

n2 – показатель преломления среды, в которую свет попадает, пройдя границу раздела

θ1 – угол падения света – угол между падающим на поверхность лучом и нормалью к поверхности

θ2 – угол преломления света – угол между прошедшим через поверхность лучом и нормалью к поверхности

(xs, ys) – координаты источника;

(xp, yp) – координаты приемника;

(xо, yо) – координаты точки преломления;

1. Концептуальная постановка задачи

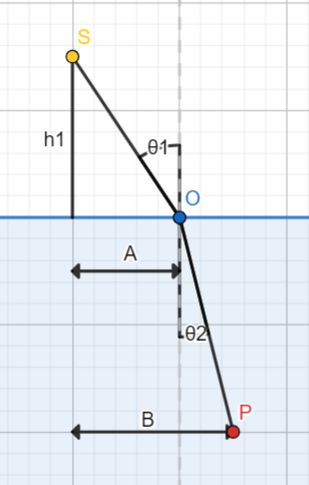
Будем рассматривать систему отражения/преломления луча со следующими условиями:

- Источник и приемник являются точками;

- Распространение солнечного луча идет строго в одной плоскости до источника;

- Применяется принцип Ферма.

1. Теоретические основы. Математическая постановка задачи



Оптический путь луча до точки О

L = n1\*SO+n2\*OP

SO =

OP =

Тогда

L = n1\*+n2\*

Экстремум по точке O

L' = n1\*A/ – n2\*(B-A)/ = 0

sin θ1 = A/

sin θ2 = (B-A)/

n1\*sin θ1 – n2 \* sin θ2 = 0

n1\*sin θ1 = n2 \* sin θ2

1. Реализация

Программная реализация задачи

double xs = 0, ys = 3,

xp = 9, yp = -6,

xo, yo = 0,

n1 = 1, n2 = 2,

A, B,

SO, OP,

L1, Lmin = Double.MaxValue, xomin = 0,

sinθ1 = -1, sinθ2 = -1;

for (xo = 0; xo < xp; xo += 0.1)

{

A = xo - xs;

SO = Math.Sqrt(Math.Pow(A, 2) + Math.Pow(ys, 2));

B = xp - xs;

OP = Math.Sqrt(Math.Pow(B - A, 2) + Math.Pow(yp, 2));

L1 = n1 \* SO + n2 \* OP;

Console.WriteLine(Math.Asin(A / SO)\*180/Math.PI + ";" + L1);

if (L1 < Lmin)

{

xomin = xo;

Lmin = L1;

sinθ1 = A / SO;

sinθ2 = (B - A) / OP;

}

}

Console.WriteLine(xs + ";" + ys);

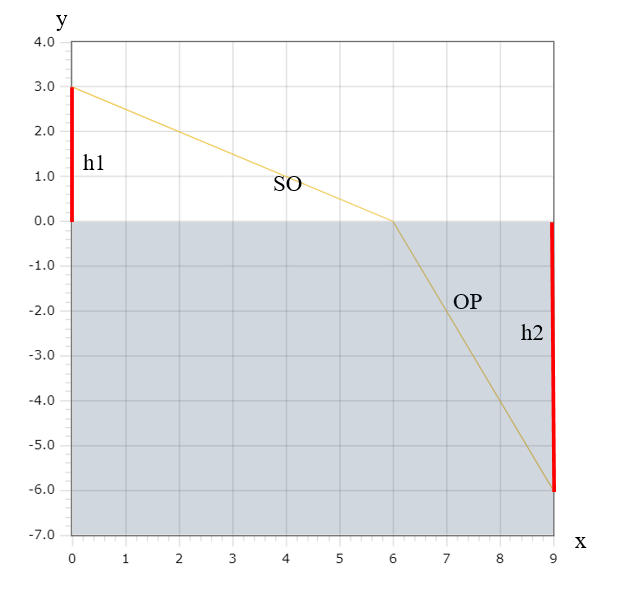
Console.WriteLine(xomin + ";" + yo);

Console.WriteLine(xp + ";" + yp);

Console.WriteLine(n1 \* sinθ1);

Console.WriteLine(n2 \* sinθ2);

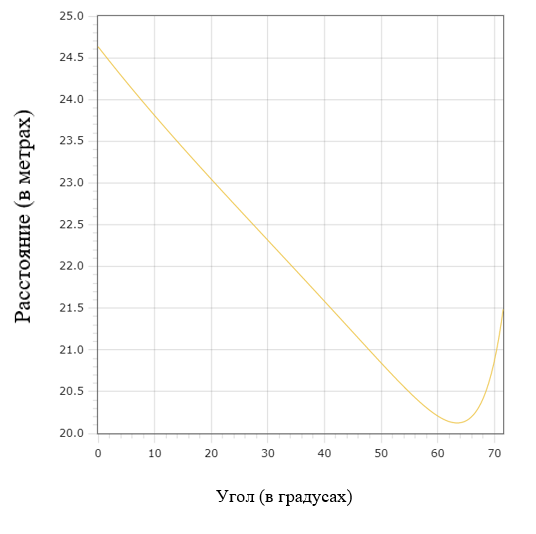
График преломления луча



n1 \* sin θ1 = 0,894

n2 \* sin θ2 = 0,894

График зависимости угла от расстояния



1. Проверка адекватности модели

В рамках решения данной задачи модель адекватна.

1. Практическое использование построенной модели

Модель применима при необходимости нахождения точки преломления, при которых будут выполняться законы преломления.