Zajan Ondřej

trubka.for - documentation

---

1 Introduction

This program is based on theory of geometric optic. It complies four out of five rules of geometric optic, with two special conditions:

1. Light travels through specific geometry (tube specified by radius and lenght)
2. Relative index of refraction equals to one - Program does not comprise Snells law

Program let's user decide length, radius of the tube and number of iterations. In one iteration the program compute one parametric equation of the light beam. Equations are saved with parameter of collision into external text file.

2 Program trubka

Purpose: For computing trajectories of the light beam, travelling trough a cylindrical enviroment
Algorithm author: Zajan Ondřej
Code Author: Zajan Ondřej
Code Editors: Zajan Ondřej
Created: 13.3.2019
Last edit: 18.4.2019
Inputs: n(1) - number of trajectories between the collisions
r(1) - radius of the tube
l(1) - length of the tube
Outputs: 'kq.txt' - file for storing the trajectories
Required subroutines: start_pos,start_vec,next_colision

Program firstly establishes main output to peripheral number 1. Then a header of the output is created. Next the **RANDOM_SEED** subroutine initializes the pseudorandom number generator used by subroutine **random.number**. The starting conditions are randomly generated by subroutines **start_pos** and **start_vec**. On line 26 is cycle in which the iterations are done by subroutine **next_colision**. On the end of the program the peripheral number 1 is closed. All the inputs are specified in *data* block.

2.1 Subroutine start_pos

Purpose: Get random starting position
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 13.3.2019
Last edit: 20.3.2019
Inputs: r(1) - radius of the tube
Outputs: st(3) - coordinate of the starting position

This subroutine is based on subroutine **RANDOM_NUMBER**. The **RANDOM_NUMBER** subroutine returns a uniformly distributed pseudorandom number or numbers in the range $0 \leq x < 1$. The starting position is generated in such way so $st(1) \in (-r; r)$, $st(2) \in (-r; r)$ and $st(3) = 0$.

2.2 Subroutine start_vec

Purpose: Get random starting directional vector "a"
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 13.3.2019
Last edit: 5.4.2019
Inputs: r(1) - radius of the tube
l(1) - length of the tube
Outputs: a(3) - starting vector

This subroutine works likewise the subroutine **start_pos**, except that generated vector $a(1) \in (-r; r)$, $a(2) \in (-r; r)$ and $a(3) \in (-l; l)$.

2.3 Function is_eq

Purpose: Function for determining if the two variables are equal
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 14.3.2019
Last edit: 14.3.2019
Inputs: a(1) - first variable
b(1) - second variable
Outputs: is_eq - returns logical value, a==b => $.true.$ || a!=b => $.false.$

This function is based on **EPSILON** function. The **EPSILON** function returns a positive real value that is almost negligible compared to unity. It is the smallest value of x such that 1.+x is not equal to 1. If the distance between values is smaller then $\mathbf{EPSILON}(0.\mathrm{d}0) \cdot 10^3$, then the function returns $.true.$, otherwise it returns $.false.$.

2.4 Subroutine next_colision

Purpose: Subroutine for computing trajectories of the light beam, travelling trough a cylindrical environment
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 13.3.2019
Last edit: 18.4.2019
Inputs: k(3) - directional vector
st(3) - starting position
r(1) - tube radius
l(1) - tube length
Outputs: k(3) - reflecting directional vector
st(3) - starting position of the directional vector
s(1) - logical value describing direction of the light
Peripheral number (write) 1 - for writing the parametric equations of the trajectories
Required subroutines: tecna_normala,normalize,projection,rotate Required functions: is_eq,t_end,vec_angle

The first place where the light hits the wall is determined by function **t_end**. The equation with its parameter $t0$ is then written to output file. Subroutine which determines tangent g and normal g_m in the place of contact **tecna_normala** is called. These are then normalized with subroutine **normalize**. After that the angle between the normal and directional vector of the light f is determined by function **vec_angle**. To get a rotation axis, firstly the directional vector k is projected to a plane formed by z axis which is the longitudinal part of the tube and by tangent g, which was already determined. Next the cross product n of the vector g_n and projected vector is done by subroutine **axb** and the result is then multiplied by -1. The angle f is then multiplied by 2 and directional vector f

is rotated by angle f by axis n. The subroutine returns reflected beam.

2.5 Subroutine tecna_normala

Purpose: Get the tangent and the normal of the tube on the position of the collision
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 13.3.2019
Last edit: 18.4.2019
Inputs: st(3) - position of the needed tangent
r(1) - radius of the tube
Outputs: g(3) - directional vector of the tangent
g_n(3) - directonal vector of the normal
Required functions: is_eq

Here is assumed, that the shape of the environment is cylindrical. For circle we have this equation

$$y^2 + x^2 = r^2, \tag{1}$$

where r is radius of the object. Tangent of function f in point a is defined by this equation

$$y - f(a) = f'(a)(x - a). \tag{2}$$

By substituting (1) to (2) and assuming, that $a = st(1)$, which is x axis value of the place of collision, and $x = 1$ we get two results:

$$y = \pm \frac{st(1)}{\sqrt{r^2 - st(1)^2}}(1 - st(1)) \pm \sqrt{r^2 - 1}, \tag{3}$$

where $st(2) = \pm \sqrt{r^2 - 1}$, which is y axis value of the place of collision. Because is wanted to have the tangent putted to the middle of the plot, from the final tangent vector g is subtracted vector st. Because the tangent can also be perpendicular to the y-axis, it is necessary to check if the $st(1)^2$ value equals to r^2. Normal in the same place as tangent is found simply by replacing the x and y value of the tangent and then changing sign of x coordinate. Finally normal g_n is redirected to right place by changing its sign.

2.6 Subroutine normalize

Purpose: Normalize the vector "x"
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 5.4.2019
Last edit: 5.4.2019
Inputs: x(3) - normalizing vector
Outputs: x(3) - normalized vector
Required functions: is_eq

Vector $\vec{x}$ is multiplicated by $\frac{1}{\| \vec{x} \|}$ so $\| \vec{x} \| = 1$.

2.7 Subroutine projection

Purpose: Projection of vector "st" on plane created by "g" and "(0,0,1)"
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 5.4.2019
Last edit: 5.4.2019
Inputs: g(3) - tangent of the tube
st(3) - projecting vector

Outputs: st(3) - projected vector

The projection of vector $\vec{b} = \begin{pmatrix} st(1) \\ st(2) \\ st(3) \end{pmatrix}$ to the plane defined by matrix $A = \begin{pmatrix} g(1) & 0 \\ g(2) & 0 \\ g(3) & 1 \end{pmatrix}$. Projected vector $\vec{proj}$ then equals to [1]

$$\vec{proj} = A \cdot \vec{x}, \tag{4}$$

where

$$\vec{x} = \left(A^T A\right)^{-1} A^T \vec{b}. \tag{5}$$

From (4) the vector is estimated as

$$\vec{proj} = \begin{pmatrix} \frac{st(1) \cdot g(1)^2}{g(1)^2 + g(2)^2} + \frac{st(2) \cdot g(1) \cdot g(2)}{g(1)^2 + g(2)^2} \\ \frac{st(1) \cdot g(1)\ g(2)}{g(1)^2 + g(2)^2} + \frac{st(2) \cdot g(2)^2}{g(1)^2 + g(2)^2} \\ st(3) \end{pmatrix}. \tag{6}$$

2.8 Subroutine rotate

Purpose: General ratation matrix - ratating vector "vektor" according to axis "n" by angle "f" (rad)
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 20.3.2019
Last edit: 23.3.2019
Inputs: n(3) - rotation axis
f(1) - rotation angle
vektor(3) - rotating vector
Outputs: vektor(3) - rotated vector

Subroutine **rotate** uses rotation matrix R defined by Rodrigues formula [3]

$$R(\vec{n},\theta) = cos(\theta)\delta_{ij} + (1 - cos(\theta))n_i n_j - sin(\theta)\varepsilon_{ijk}n_k, \tag{7}$$

where $\vec{n}$ is rotation axis and θ is angle of rotation.

2.9 Function t_end

Purpose: Get the parameter "t_end" of the parametric equation T=k.t+q, where the light hits the wall of the tube
Algorithm author: Zajan Ondøej
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 13.3.2019
Last edit: 18.4.2019
Inputs: r(1) - radius of the tube
k(3) - directional vector
q(3) - starting position
Outputs: t_end(1) - the parameter of the collision
Required functions: is_eq

For x and y coordinates of collision point greater than zero, applies this equationss

$$x = k(1)t + q(1) \tag{8}$$

$$y = k(2)t + q(2) \tag{9}$$

$$x^2 + y^2 = r^2 \tag{10}$$

From these equations the parameter t is evaluated

$$t_{1,2} = \frac{-k(1)q(1) - k(2)q(2) \pm \sqrt{-k(1)^2q(2)^2 + k(1)^2r^2 + 2k(1)k(2)q(1)q(2) - k(2)^2q(1)^2 + k(2)^2r^2}}{k(1)^2 + (2)^2}. \tag{11}$$

Coordinates for these parameters are then estimated from parametric equations. It is checked, if any of these coefficients corresponds to the initial conditions, when the beam does not starts on the wall of the tube. If the beam start on the wall, the right parameter for the collision is the one that is greater than zero. If the beam starts on the wall, one parameter responds to the starting position and the other one to the position of the collision.

2.10 Function vec_angle

Purpose: Gen the angle between the vector "vec_1" and "vec_2"
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej, Endrychová Alžběta
Created: 5.4.2019
Last edit: 18.4.2019
Inputs: vec_1(3) - first vector
vec_2(3) - second vector
Outputs: vec_angle - angle between the vectors

The angle between the vectors α is computed from scalar product

$$v\vec{e}c_1 \cdot \ v\vec{e}c_2 = \| \ v\vec{e}c_1 \| \cdot \ \| \vec{ve}c_2 \| \cdot cos\alpha. \tag{12}$$

2.11 Function vec_size

Purpose: Get the size of the vector "vec"
Code Author: Zajan Ondøej
Code Editors: Zajan Ondøej
Created: 5.4.2019
Last edit: 5.4.2019
Inputs: vec(3) - Given vector
Outputs: vec_size - size of the given vector

This function simply compute the size of the vector from this formula

$$\| \ v\vec{e}c \| = \sqrt{vec(1)^2 + vec(2)^2 + vec(3)^2}. \tag{13}$$

3 Bibliography

References

[1] L. Balková *Lineární algebra 1: 5.6 Projektor* [Online]. [cit. June 15, 2019].
`http://kmlinux.fjfi.cvut.cz/~balkolub/Vyuka/zima2012/LAPzima2012/main.pdf`

[2] J. Tolar *VLNĚNÍ, OPTIKA A ATOMOVÁ FYZIKA* [Online]. [cit. June 15, 2019].
`https://physics.fjfi.cvut.cz/files/predmety/02VOAF/VOAF2017.pdf`

[3] H. Haber *Three-Dimensional Rotation Matrices* [Online]. [cit. June 15, 2019].
`http://scipp.ucsc.edu/~haber/ph216/rotation_12.pdf`