

Light Table Simulation

Introduction:

The bending of light when it passes through a surface can be described by Snell's Law. Snell's Law will describe the path that the light beam will take and it is simple enough to calculate for multiple surfaces. The program described in this report automates the process allowing for larger numbers prisms to be used on the table. The title of this program is light and it need to have the file prisms located in the same file as the program.

Experiment Procedure:

To run this program the user must set the placement of at least one prism. The program has some several prisms programmed in already but more prisms can be created within the program by defining the vertices of the prism.

The Light Table Simulation program that is used in this project determines the behavior of a light beam that is shot through a series of prisms. One of the main equations that the program uses is the basic line equation (eqn. 1) which is used to govern

$$\text{Equation of a Line} \qquad y=mx+b \qquad (\text{eqn. 1})$$

the beam as well as the sides of the various prisms on the table. The program uses the equation of the beam and the equation of the object sides to find any intersection of that are between the two end points of the side of the various objects. The program then finds the closes intersection that is in the correct direction. The other equation that the program requires is Snell's Law (eqn. 2). In Snell's Law each n and each θ are the refractive indexes and the angles from the normal of the surface on ether side of the boundary

$$\text{Snell's Law} \qquad n_1 \sin\theta_1 = n_2 \sin\theta_2 \qquad (\text{eqn. 2})$$

respectively. Using this equation the program solves for the out going angle and then uses it and the previous incoming angle to determine the new direction of the light beam. However under certain conditions to solve for the out going angle, $\frac{n_1}{n_2} \sin\theta_1 > 1$ so we will assume that the light beam will reflect instead of refract. When the beam reflects it will reflect at the same angle from the normal that the beam came in at. Now that the program has found the point of intersection and then how beam will behave

the program repeats the above steps until closest intersect point lies on one of the table's edges.

Experimental Results:

When first running the program the user will be asked if they would like to create a new prism. If the user wants to create a new prism they just need to follow the direction and put in the position of the three vertices. After saving the prism it will be added to the option menu for later in the program.

After creating any new prism the user can then perform a run of the simulation. Following the instruction the user can create many different situations, but the below is run with simple values (Table 1).

Values for example run	
Left side of table	-8 units
Right side of table	8 units
Far side of table	8 units
Near side of table	-8 units
Refractive index of space on table	1
Number of prisms	1 prism
Prism type	prism3 0454590.f90
Prism scale	6
Invert prism	No
Theta offset	0 degrees
Position of first vertex	X=0, Y=1
Refractive index of prism3 1	1.55
Origin of light beam	X=0, Y=0
Angle for light beam to x-axis	45 degrees

(Table 1)

List of input value used in the example run of the program light.

Using the inputs in Table 1 we have a table that is 16 units by 16 units with the empty space having a refractive index of 1 which in this case means air. There is one 45x45x90 prism placed on the table with the first vertex sitting at the point (0,1). The prism is not inverted about the y-axis and is not rotated at all. After the prism options are set the program will ask if the user would like to proceed with the prism set. If the user types Yes, the program will prompt for the origin of the light beam and the angle at which the light beam is shot at. The program will then create a picture of the prisms and the path of the light beam (Figure 1). In addition to the picture in the shell window the program prints out a list of the intersection points and the angle that the new beam segment will leave that point at (Table 2).

```

=====
Intersection point | Angle after Intersection
=====
0.00 0.00 | 45.00
1.00 1.00 | 62.86
2.05 3.05 | 27.14
6.00 5.08 | 45.00
8.00 7.08 | *****
=====
They're are 5 intersections.
=====

```

Table 2
This is a list of all intersection and the angle that the beam will leave at.

In addition to Table 2 the program also prints out a list of the prisms and their refractive indexes (Table 3).

```

=====
Prism | Refractive Index
=====
Prism3_1 | 1.55
=====

```

Table 3
This table lists the prism as they are listed on figure 1. With each prism it list the refractive index of that prism.

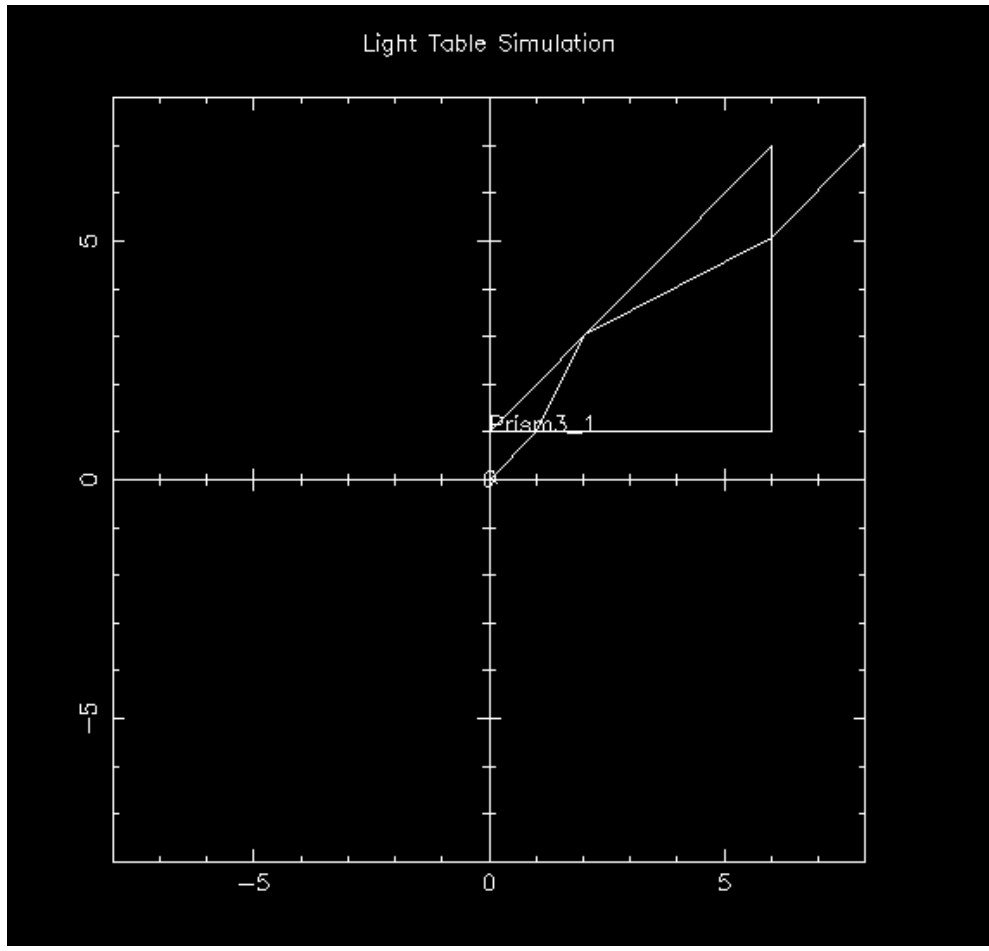


Figure 1
This the picture that program light will print out when the values from Table 1 are used.

Conclusion:

Using the program light we are able to find how a light beam will react when it paces through a prism. With the use of the automating process we are also able to speed up the process and determine how the beam reacts when multiple prisms are used.

References:

Frank L. Pedrotti, S.J., Leno S. Pedrotti, Introduction to Optics, 1993, Prentice-Hall, Inc. A Simon & Schuster Company Englewood Cliffs, New Jersey 07632.