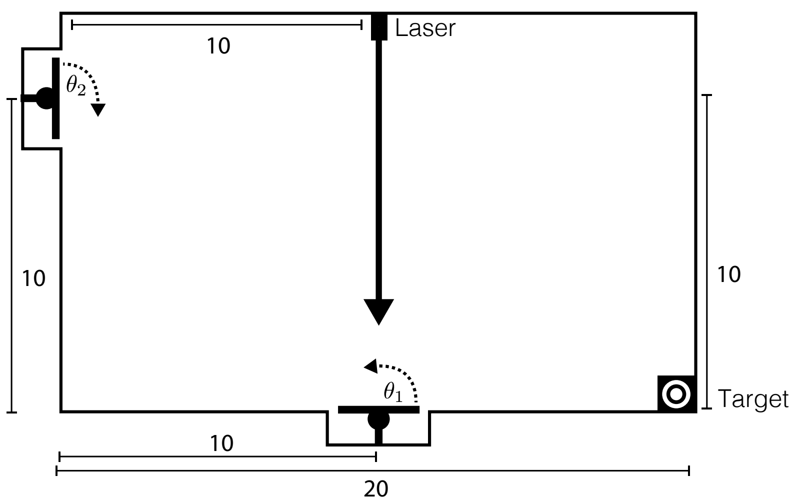
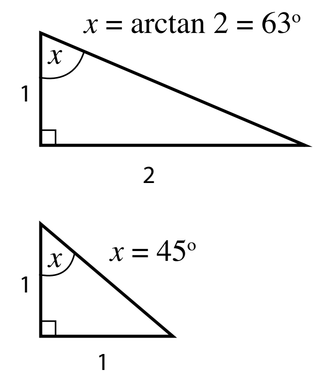
Stanford CS248: Interactive Computer Graphics
Exercise 5

Problem 1: Everyone Loves Lasers

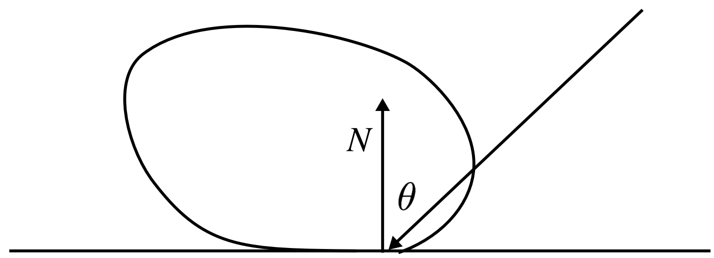
A mad scientist decides to design a fun physics experiment to amuse students in class. The goal of the experiment is to use two **perfectly reflective mirrors** to direct a laser beam, positioned downward from the top of the box, to hit a target in the bottom right corner of the box. The two mirrors can be rotated about their center point by the angles θ_1 and θ_2 as shown in the figure.





A. Please compute positive values of θ_1 and θ_2 to hit the box. (Some helpful triangles are given for you, which may or may not be useful.)

B. One challenge with perfect mirrors is that if you don't get them tilted just right, the laser will miss the target. One of the students, frustrated they couldn't hit the target, takes out a piece of sandpaper and scruffs up the two mirrors. The result is that the mirrors now have a BRDF that is almost fully diffuse, as given by the plot below. Note the surface reflects non-zero incoming light in all directions, but the fraction of light reflected in each of these directions is angle dependent. (More light is still reflected in the direction of perfect specular reflection.)

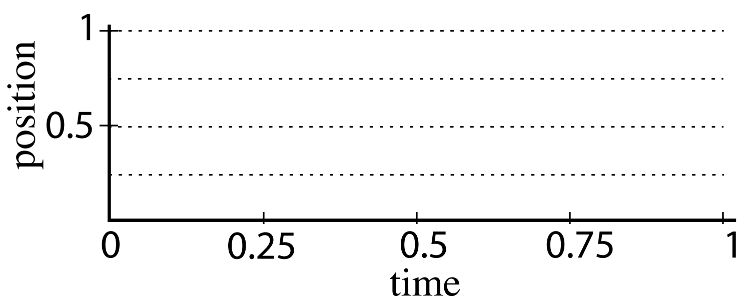


Assuming that (1) the mirrors are set so that $\theta_1 = \theta_2 = 0$ and that all the walls of the room reflect no light (they are perfectly black), does any laser light hit the target? If your answer is no, explain why not. If your answer is yes, please explain why, and also state whether target is brighter or darker compared to what it would look like in the case of well-aligned perfect mirrors from part A.

Problem 2: Splines

A. In class we talked about the animation principle of ease-in, ease out, where a motion starts slowly, accelerates, then decelerates to a stop. Consider specifying the spline control points for an animation of a ball, where the spline's value at time t determines the ball's Z coordinate in the scene. Assume that you are to supply control point values at times $t_0 = 0$, $t_1 = 0.25$, $t_2 = 0.75$, $t_3 = 1.0$ (let's call these values C_i's). The spline *interpolates* the control points at t_0 and t_3, and the difference between C_1 and C_0 as well as C_3 and C_2 is used to compute curve tangents at the spline's endpoints.

Please draw control points on the figure below that result in an ease-in, ease-out animation where the ball begins at Z=0 and ends at Z=0.75.



B. Now assume that the interior control points C_1 and C_2 can be placed at any t. In other words, there are interior control points at (t_1, C_1) and (t_2, C_2). Write down the four constraint equations for the spline $at^3 + bt^2 + ct + d$ in terms of the (t_i, C_i)'s. Keep in mind that the spline interpolates (t_0, C_0) and (t_3, C_3) and the spline tangent at t_0 is the slope of the line between (t_1, C_1) and (t_0, C_0) (similarly for the tangent at t_3).

C. Now turn your four constraints into a linear system. (Fill out the coefficients for the matrix $\mathbf{M}$, as well as the vector on the right-hand side of the equation below.)

$$\mathbf{M}\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad & \quad \\ \\ \\ \\ \end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \quad \\ \\ \\ \\ \end{bmatrix}$$

D. Imagine that instead of ease-in, ease-out, you instead want to make sure the position of the ball begins at Z=0.25, is located at Z=0.5 at t_1, at Z=0.75 at t_2 and ends at Z=0.5. What is the name of a cubic spline that provides a good representation for this problem? (Hermite? B-spline? Catmull-Rom? Bezier?) Why?

E. Let's return to the spline you derived in part C, which can be written in terms of the input control vector $\mathbf{c}$ (your right hand side in part C) and matrix $\mathbf{M}$, which to provided in part C as well. **(NOTE: a correct answer to part C is not necessary to do this problem.)**

$$Z(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \mathbf{M}^{-1}\mathbf{c}$$

Now consider a different spline representation, such as a Bezier spline, which can represent the same spline curve, but using different basis matrix and different input control points ($\mathbf{c}_{\text{bez}}$)

$$Z(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \mathbf{B}\mathbf{c}_{\text{bez}}$$

Imagine you only had an implementation of Bezier splines, but you wanted to implement the same curve that was specified in part B. How could you convert your spline control points $\mathbf{c}$ from part B to a vector of Bezier control points that result in the same curve? (Hint: you should think of this question as just asking you to make a change of basis!)

F. One of the benefits of a Bezier curve is that the control points forms *a convex hull* for the actual spline curve. Consider the challenge of determining if two spline curves intersect. How precisely might you use the convex hull property to accelerate intersection tests?