Homework #2

Shading, Projections, Texture Mapping, Ray Tracing, and Bezier Curves

Assigned: Thursday, May 5th

Due: Thursday, May 19th
at the beginning of class

Directions: Please provide short written answers to the following questions on your own paper. Feel free to discuss the problems with classmates, but *please* ***follow the Gilligan's Island rule*, answer the questions on your own, and show your work.***

Late policy: The homework is due at the beginning of class. Late assignments are marked down at a rate of 25% per day (not per lecture), meaning that if you fail to turn in an assignment on time it is worth 75% for the first 24 hours after the deadline, 50% for the next 24 hours, etc.

Please write your name on your assignment!

* **The Gilligan's Island Rule:** This rule says that you are free to meet with fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. After the meeting, engage in a half hour of mind-numbing activity (like watching an episode of Gilligan's Island), before starting to work on the assignment. This will assure that you are able to reconstruct what you learned from the meeting, by yourself, using your own brain.

Problem 1. Blinn-Phong shading (16 Points)

The Blinn-Phong shading model for a scene illuminated by global ambient light and a single directional light can be summarized by the following equation:

$$I_{phong} = k_e + k_a I_a + k_d B\, I_L\, (\mathbf{N}\cdot\mathbf{L}) + k_s B\, I_L\, (\mathbf{N}\cdot\mathbf{H})_+^{n_s}$$

Imagine a scene with one white sphere illuminated by white global ambient light and a single white directional light. For sub-problems a) – f), describe – qualitatively, in words – the effect of each step on the shading of the object. At each incremental step, assume that all the preceding steps have been applied first. Assume that the directional light is oriented so that the viewer can see the shading over the surface, including diffuse and specular where appropriate.

a) (2 points) The directional light is off. How does the shading vary over the surface of the object?

b) (2 points) Now turn the directional light on. The specular reflection coefficient k_s of the material is zero, and the diffuse reflection coefficient k_d is non-zero. How does the shading vary over the surface of the object?

c) (2 points) Now translate the sphere straight toward the viewer. What happens to the shading over the object?

d) (2 points) Now increase the specular exponent n_s. What happens?

e) (2 points) Now increase the specular reflection coefficient k_s of the material to be greater than zero. What happens?

f) (2 points) Now decrease the specular exponent n_s. What happens?

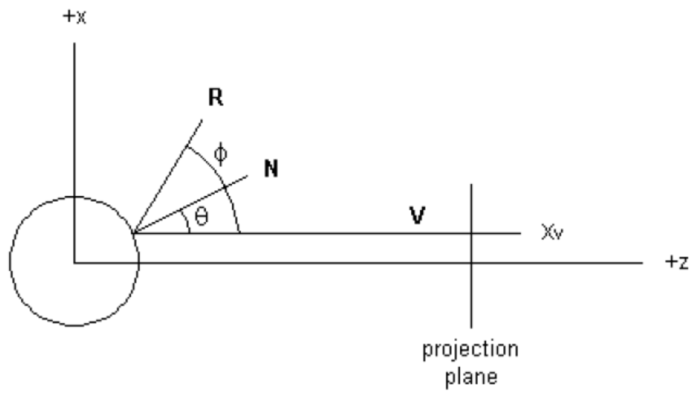
g) (2 points) Suppose we assume that the viewing direction **V** is constant regardless of which pixel it passes through. What does this imply about the viewer?

h) (2 points) Assuming that **L** and **V** are constant everywhere, then with a little pre-computation, it is possible to shade faster (i.e., using fewer operations) using the Blinn-Phong model above, than it is to shade using the Phong model, which bases the specular component on $(\mathbf{V}\cdot\mathbf{R})_+^{n_s}$. Why would Blinn-Phong be faster than Phong in this situation? Explain.

Problem 2. Environment mapping (20 points)

One method of environment mapping (reflection mapping) involves using a "gazing ball" to capture an image of the surroundings. The idea is to place a chrome sphere in a real environment, take a photograph of the sphere, and use the resulting image as an environment map. Each pixel that "sees" a point on the chrome sphere maps to a ray with direction determined by the reflection through the chrome sphere; the pixel records the color of a point in the surroundings along that reflected direction. You can turn this around and construct a lookup table that maps each reflection direction to a color. This table is the environment map, sometimes called a reflection map.

Let's examine this in two dimensions, using a "gazing circle" to capture the environment around a point. Below is a diagram of the setup. In order to keep the intersection and angle calculations simple, we will assume that each viewing ray **V** that is cast through the projection plane to the gazing circle is parallel to the z-axis. The circle is of radius 1, centered at the origin.



a) (5 points) If the x-coordinate of the view ray is x_V, what are the (x,z) coordinates of the point at which the ray intersects the circle? What is the unit normal vector at this point?

b) (3 points) What is the angle between the view ray **V** and the normal **N** as a function of x_V? Note that we will treat this as a "signed angle." In the figure above, the angle θ between **V** and **N** is positive. If the viewing ray hit the lower half of the circle (x_V is negative), then θ would be negative.

c) (5 points) Note that the (signed) angle φ between the view ray **V** and the reflection direction **R** is equal to 2θ, where θ is the angle between **V** and the normal **N**. Plot φ versus x_V. In what regions of the image do small changes in the x_V coordinate result in large changes in the reflection direction?

d) (4 points) We can now use the photograph of the chrome circle to build an environment map (for a 2D world); we store an array of colors drawn from the photograph, regularly sampled across reflection angles. When ray tracing a new chrome object, we compute the mirror reflection angle when a ray intersects the object, and then just look up the color from the environment map. (If the computed reflection angle lands between angles stored in the environment map, then you can use linear interpolation to get the desired color.) Would we expect to get exactly the same rendering as if we had placed the object into the original environment we photographed? Why or why not? In answering the question, you can neglect viewing rays that do not hit the object, assume that the new object is not itself a chrome circle, and assume that the original environment is some finite distance from the chrome circle that was originally photographed.

e) (3 points) Suppose you lightly sanded the chrome circle before photographing it, so that the surface was just a little rough.

- What would the photograph of the circle look like now, compared to how it looked before roughening its surface?
- If you used this image as an environment map around an object, what kind of material would the object seem to made of?
- If you did not want to actually roughen the object, what kind of image filter might you apply to the image of the original chrome circle to approximate this effect?

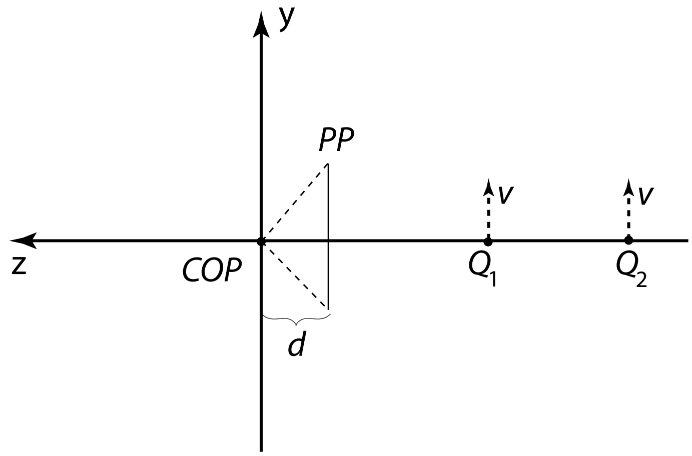
Problem 3: Projections (18 points)

The apparent motion of objects in a scene can be a strong cue for determining how far away they are. In this problem, we will consider the projected motion of points and line segments and their apparent velocities as a function of initial depths.

a) (6 points) Consider the projections of two points, Q_1 and Q_2, on the projection plane *PP*, shown below. Q_1 and Q_2 are described in the equations below. They are moving parallel to the projection plane, in the positive *y*-direction with speed *v*.

$$Q_1(t) = \begin{bmatrix} 0 \\ vt \\ z_1 \\ 1 \end{bmatrix} \qquad Q_2(t) = \begin{bmatrix} 0 \\ vt \\ z_2 \\ 1 \end{bmatrix}$$

$$0 > z_1 > z_2$$



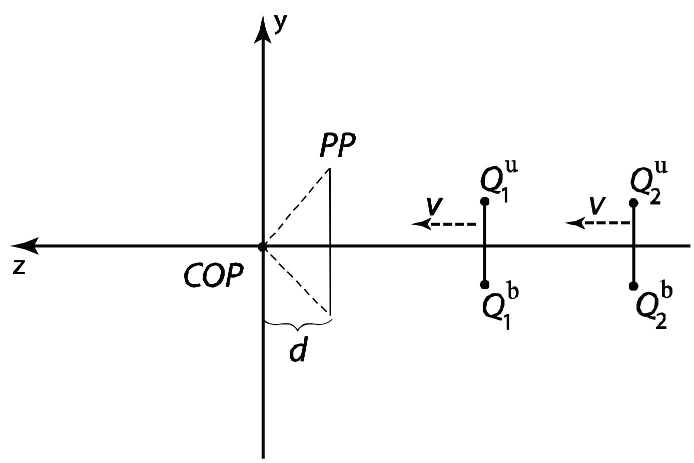
Compute the projections $q1$ and $q2$ of points Q_1 and Q_2, respectively. Then, compute the velocities, dq_1 / dt and dq_2 / dt, of each projected point in the image plane. Which appears to move faster? Show your work.

b) (8 points) Consider the projections of two vertical line segments, S_1 and S_2, on the projection plane *PP*, shown below. S_1 has endpoints, Q_1^u and Q_1^b. S_2 has endpoints, Q_2^u and Q_2^b. The line segments are moving perpendicular to the projection plane in the positive *z*-direction with speed *v*.

$$Q_1^u(t) = \begin{bmatrix} 0 \\ 1 \\ z_1 + vt \\ 1 \end{bmatrix} \qquad Q_2^u(t) = \begin{bmatrix} 0 \\ 1 \\ z_2 + vt \\ 1 \end{bmatrix}$$

$$Q_1^b(t) = \begin{bmatrix} 0 \\ -1 \\ z_1 + vt \\ 1 \end{bmatrix} \qquad Q_2^b(t) = \begin{bmatrix} 0 \\ -1 \\ z_2 + vt \\ 1 \end{bmatrix}$$

$$0 > z_1 > z_2$$



Compute the projected lengths, l_1 and l_2, of the line segments. Then, compute the rates of change, dl_1 / dt and dl_2 / dt, of these projected lengths. Are they growing or shrinking? Which projected line segment is changing length faster? Show your work.

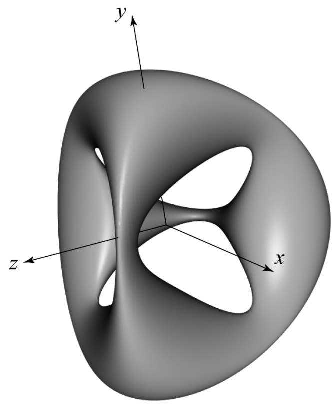
c) (4 points) Suppose now we replace the perspective camera in (a) and (b) with an orthographic camera. Which point, if any, in (a) would appear to move faster? Will the line segments in (b) appear to grow or shrink, and if so, which would change faster? Justify your answers in words or with equations.

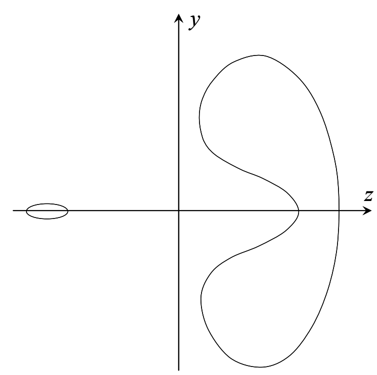
Problem 4. Ray intersection with implicit surfaces (23 points)

There are many ways to represent a surface. One way is to define a function of the form $f(x, y, z) = 0$. Such a function is called an *implicit surface* representation. For example, the equation $f(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$ defines a sphere of radius *r*. Suppose we wanted to ray trace a "quartic chair," described by the equation:

$$(x^2 + y^2 + z^2 - ak^2)^2 - b\left[(z-k)^2 - 2x^2\right]\left[(z+k)^2 - 2y^2\right] = 0$$

On the left is a picture of a quartic chair, and on the right is a slice through the *y-z* plane.





For this problem, we will assume a = 0.95, b = 0.8, and k = 5.

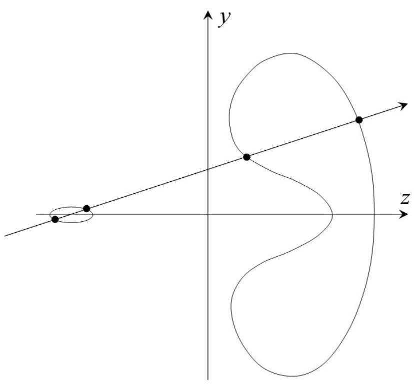
In the next problem steps, you will be asked to solve for and/or discuss ray intersections with this primitive. Performing the ray intersections will amount to solving for the roots of a polynomial, much as it did for sphere intersection. For your answers, you need to keep a few things in mind:

- You will find as many roots as the order (largest exponent) of the polynomial.
- You may find a mixture of real and complex roots. When we say complex here, we mean a number that has a non-zero imaginary component.
- All complex roots occur in complex conjugate pairs. If *A* + i*B* is a root, then so is *A* – i*B*.
- Sometimes a real root will appear more than once, i.e., has multiplicity > 1. Consider the case of sphere intersection, which we solve by computing the roots of a quadratic equation. A ray that intersects the sphere will usually have two distinct roots (each has multiplicity = 1) where the ray enters and leaves the sphere. If we were to take such a ray and translate it away from the center of the sphere, those roots get closer and closer together, until they merge into one root. They merge when the ray is tangent to the sphere. The result is one distinct real root with multiplicity = 2.

a) (8 points) Consider the ray $P + t\mathbf{d}$, where $P = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$. Solve for all values of *t* where the ray intersects the quartic chair (including negative values of *t*). Which value of *t* represents the intersection we care about for ray tracing? In the process of solving for t, you will be computing the roots of a polynomial. How many distinct real roots do you find? How many of them have multiplicity > 1? How many complex roots do you find?

Problem 4 (cont'd)

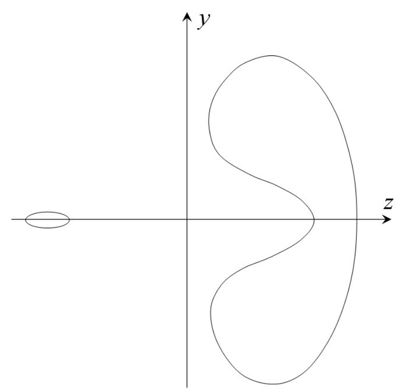
b) (15 points) What are all the possible combinations of roots, not counting the one in part (a)? For each combination, describe the 4 roots as in part (a), draw a ray in the *y-z* plane that gives rise to that combination, and place a dot at each intersection point. There are five diagrams below that have not been filled in. You may not need all five; on the other hand, if you can actually think of more distinct cases than spaces provided, then we might just give extra credit. The first one has already been filled in. (Note: not all conceivable combinations can be achieved on this particular implicit surface. For example, there is no ray that will give a root with multiplicity 4.) ***Please write on this page and include it with your homework solution. You do not need to justify your answers.***



# of distinct real roots: **4**

# of real roots w/ multiplicity > 1: **0**

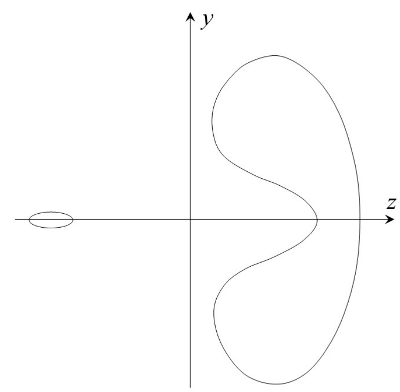
# of complex roots: **0**



# of distinct real roots:

# of real roots w/ multiplicity > 1:

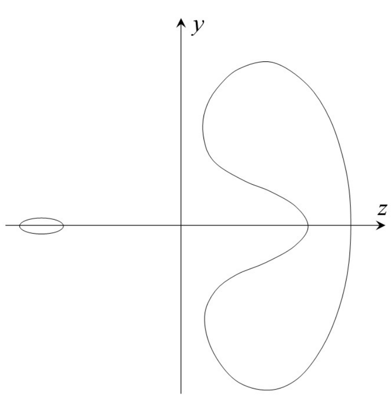
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

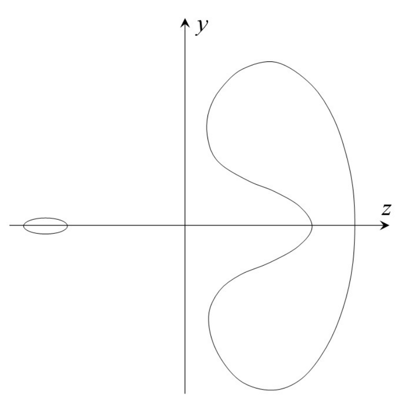
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

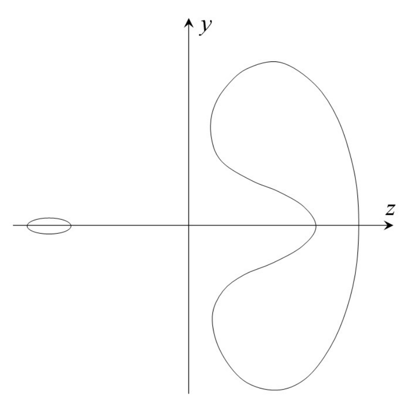
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

# of complex roots:

Problem 5. Bezier splines (23 points)

Consider a Bezier curve segment defined by three control points V_0, V_1, and V_2.

a) (3 points) What is the polynomial form of this curve, when written out in the form $Q(u) = A_n u^n + A_{n-1} u^{n-1} + \ldots + A_0$, where n is determined by the number of control points. The coefficients $A_0, \ldots, A_n$ should be substituted in the polynomial equation with expressions that depend on the control points V_0, V_1, and V_2. You may start with recursive subdivision or with the summation over Bernstein polynomials provided in lecture. Either way, show your work.

b) (2 points) What is the first derivative of $Q(u)$ evaluated at $u = 0$ and at $u = 1$ (i.e., what are $Q'(0)$ and $Q'(1)$)? Show your work.

c) (2 points) What is the second derivative of $Q(u)$ evaluated at $u = 0$ and at $u = 1$ (i.e., what are $Q''(0)$ and $Q''(1)$)? Show your work.

d) (5 points) To create a spline curve, we can stitch together consecutive Bezier curves. In this problem, we can add control points W_0, W_1, and W_2. What constraints must be placed on W_0, W_1, and/or W_2 so that, when combined with V_0, V_1, and V_2, the resulting spline curve is C^1 continuous at the joint between the Bezier segments? Write out equations for W_0, W_1, and/or W_2 in terms of V_0, V_1, and/or V_2. (It may be that not all of the W control points are constrained, in which case you would have fewer than three equations.) Show your work. Draw a copy of the control polygon below (shown at the bottom of the page) and place all constrained vertices exactly, and unconstrained vertices wherever you like, and then sketch the spline curve.

e) (5 points) Suppose we wanted to make the spline curve C^2 continuous at the joint between the Bezier segments. Now what constraints must be placed on W_0, W_1, and W_2? Write out equations for W_0, W_1, and/or W_2 in terms of V_0, V_1, and/or V_2. (It may be that not all of the W control points are constrained, in which case you would have fewer than three equations.) Show your work. Draw a copy of the control polygon below (shown at the bottom of the page) and place all constrained vertices exactly, and unconstrained vertices wherever you like, and then sketch the spline curve.

f) (3 points) Is it possible to achieve C^3 continuity with this spline? Explain.

g) (3 points) Suppose again that the control points are in two dimensions, but now $V_1 = V_2 = W_0 = W_1$. Think of this as sliding V_2 over on top of V_1 in the figure below, then placing W_0 and W_1 on top of those points, and then adding W_2 at some arbitrary position, somewhere to the right but not collinear with V_0 and V_1. Sketch the resulting curve. Will this curve be C^1? Justify your answer.