

## Homework #2

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

**Assigned:** Sunday, May 7<sup>th</sup>

**Due:** Thursday, May 18<sup>th</sup>  
*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.

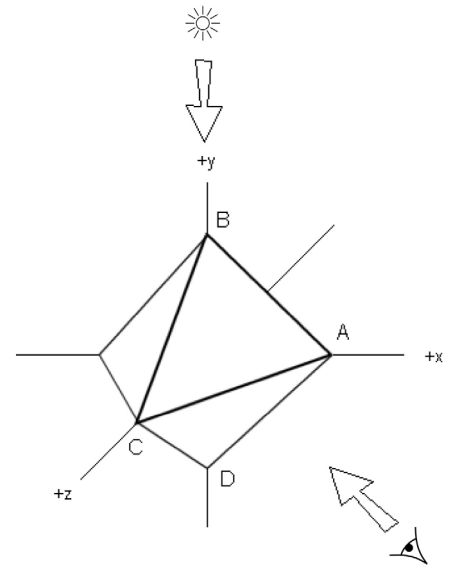
## Problem 1. Interpolated shading (19 points)

The faceted polyhedron shown in the figure at right is an octahedron and consists of two pyramids connected at the base comprised of a total of 8 equilateral triangular faces with vertices at  $(1,0,0)$ ,  $(0,1,0)$ ,  $(0,0,1)$ ,  $(-1,0,0)$ ,  $(0,-1,0)$ , and  $(0,0,-1)$ . The viewer is at infinity (i.e., views the scene under parallel projection) looking in the  $(-1,0,-1)$  direction, and the scene is lit by directional light shining down from above parallel to the  $y$ -axis with intensity  $I_L = (1,1,1)$ . The octahedron's materials have both diffuse and specular components, but no ambient or emissive components. The Blinn-Phong shading equation thus reduces to:

$$I = I_L B \left[ k_d (\mathbf{N} \cdot \mathbf{L}) + k_s (\mathbf{N} \cdot \mathbf{H})_+^{n_s} \right]$$

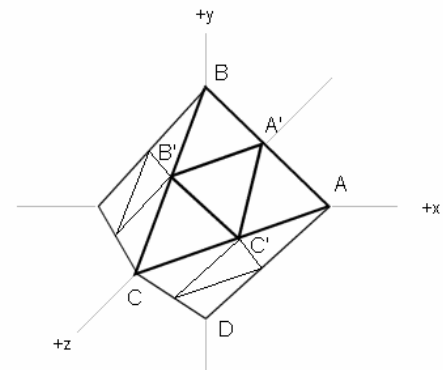
where

$$B = \begin{cases} 1 & \text{if } \mathbf{N} \cdot \mathbf{L} > 0 \\ 0 & \text{if } \mathbf{N} \cdot \mathbf{L} \leq 0 \end{cases}$$



For this problem,  $k_d = k_s = (0.5, 0.5, 0.5)$  and  $n_s=40$ .

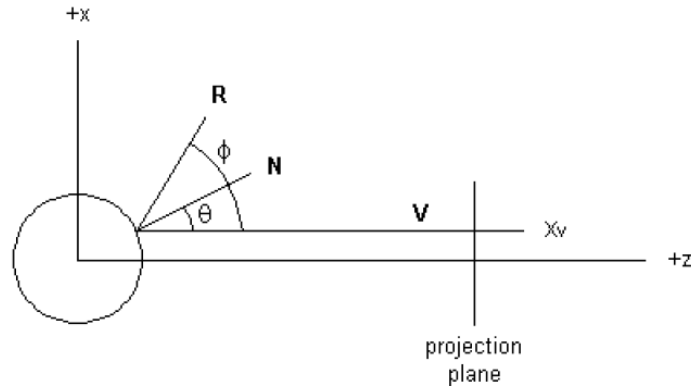
- (2 points) In order to draw the faces as flat-shaded triangles, we must shade them using only their face normals. In OpenGL, this could be accomplished by specifying the vertex normals as equal to the face normals. (The same vertex would get specified multiple times, once per triangle with the same coordinates but different normal each time.) What is the unit normal for triangle ABC?
- (3 points) Assume that this object is really just a crude approximation of a sphere (e.g., perhaps you are using the octahedron to represent the sphere because your graphics card is slow). If you want to shade the octahedron so that it approximates the shading of a sphere, what would you specify as the unit normal at each vertex of triangle ABC?
- (5 points) Given the normals in (b), compute the rendered colors of vertices A, B, and C. Show your work.
- (2 points) Again given the normals in (b), describe the appearance of triangle ABC as seen by the viewer using Gouraud interpolation.
- (2 points) Now switch from Gouraud-interpolated shading to Phong-interpolated shading. How will the appearance of triangle ABC change (given the normals in (b))?
- (3 points) Remember that this object is being used to simulate a sphere. One simple improvement to the geometry of the model is to subdivide each triangular face into four new equilateral triangles (sometimes called 4-to-1 triangular subdivision), and then move the newly inserted vertices to better approximate the sphere's shape. If you subdivided triangle ABC this way, as shown in the figure to the right, what would be the best choices for the new coordinates and unit normals of the three added vertices A', B', and C' in order to more closely approximate a unit sphere?
- (2 points) If you continued this subdivision process – repeatedly performing 4-to-1 subdivision, repositioning the inserted vertices, and computing their ideal normals – would the Gouraud-interpolated and Phong-interpolated renderings of the refined shape converge toward the same answer, i.e., the appearance of a ray traced sphere? 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.

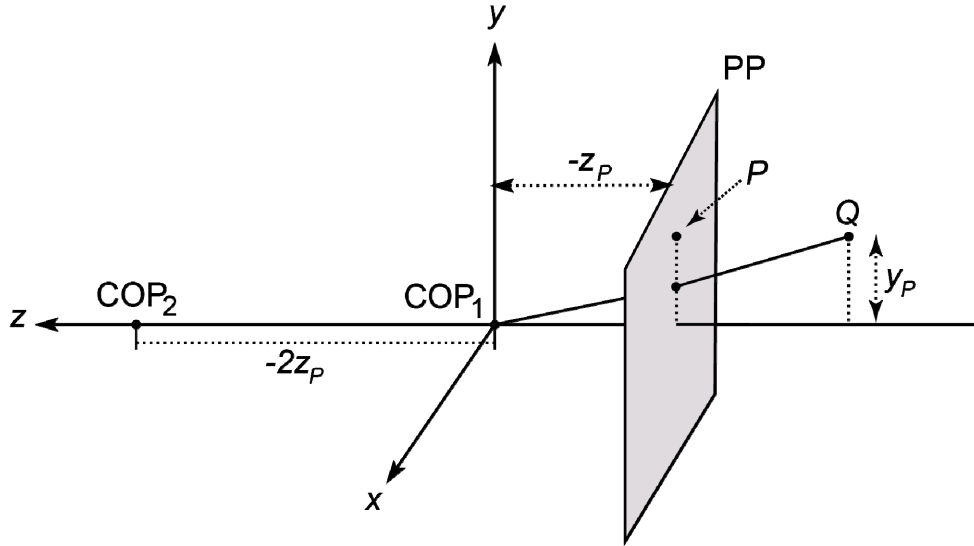
Let’s examine this in two dimensions, using a “gazing circle” to capture the environment around a point. Below is a diagram of the setup. In order to keep the intersection and angle calculations simple, we will assume that each viewing ray  $\mathbf{V}$  that is cast through the projection plane to the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



- (5 points) If the  $x$ -coordinate of the view ray is  $x_v$ , what are the  $(x, z)$  coordinates, as a function of  $x_v$ , of the point at which the ray intersects the circle? What is the unit normal vector at this point?
- (3 points) What is the angle between the view ray  $\mathbf{V}$  and the normal  $\mathbf{N}$  as a function of  $x_v$ ? Note that we will treat this as a “signed angle.” In the figure above, the angle  $\theta$  between  $\mathbf{V}$  and  $\mathbf{N}$  is positive. If the viewing ray hit the lower half of the circle ( $x_v$  is negative), then  $\theta$  would be negative.
- (5 points) Note that the (signed) angle  $\phi$  between the view ray  $\mathbf{V}$  and the reflection direction  $\mathbf{R}$  is equal to  $2\theta$ , where  $\theta$  is the angle between  $\mathbf{V}$  and the normal  $\mathbf{N}$ . Plot  $\phi$  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?
- (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.
- (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 be 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 (15 points)

Imagine there is a pinhole camera located at the origin,  $\text{COP}_1$ , that is looking in the  $-z$  direction. The projection plane (PP) is the plane  $z = z_P$  (note:  $z_P$  is a negative number), so that the distance from  $\text{COP}_1$  to PP is  $f = -z_P$ , as shown in the figure below. Let there be two points in the scene,  $P = [0 \ y_P \ z_P \ 1]^T$  and  $Q = [0 \ y_P \ 2z_P \ 1]^T$ .



The projection matrix for this camera is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/z_P & 0 \end{bmatrix}$$

This projects  $P$  to the point  $[0 \ y_P \ 1]^T$  and  $Q$  to the point  $[0 \ \frac{1}{2}y_P \ 1]^T$ .

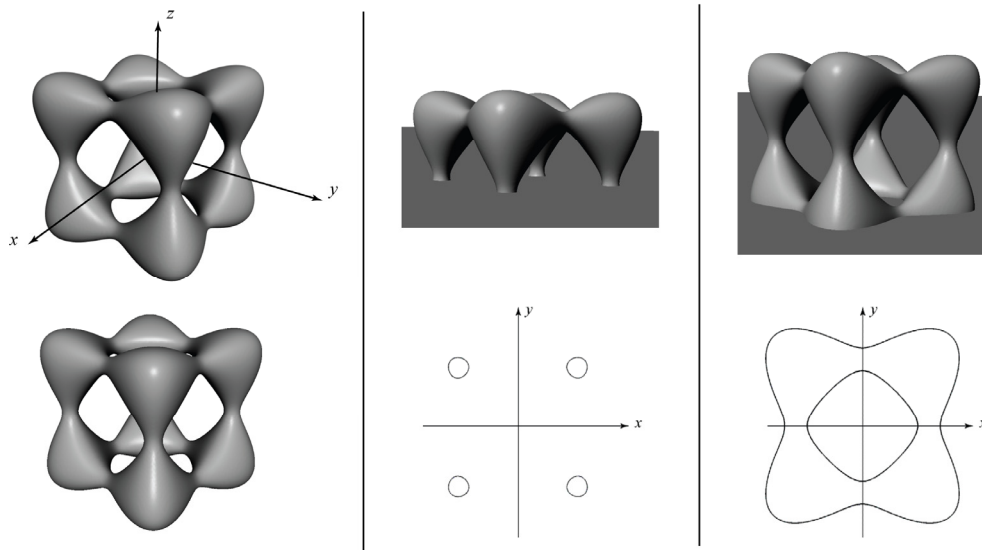
- (4 points) Now, assume that the camera has moved to  $\text{COP}_2$  (shown above) at  $[0 \ 0 \ -2z_P \ 1]^T$ . Assume the PP stays at  $z = z_P$ . Derive the new projection matrix that maps points onto the PP. Show your work. Hint: a translation matrix may help you solve this problem.
- (2 points) If your matrix in part a) is correct, point  $P$  should project to the same image point as before. Calculate the projection of point  $Q$ . How did the projection of point  $Q$  change when the camera moved from  $\text{COP}_1$  to  $\text{COP}_2$ ?
- (6 points) Suppose we want to keep the projection of  $Q$  constant at  $[0 \ \frac{1}{2}y_P \ 1]^T$ . Suppose the center of projection is at  $\text{COP} = [0 \ 0 \ z_{\text{COP}} \ 1]^T$ . To keep the projection of  $Q$  constant, we will need to vary the  $z$ -coordinate of PP; let the updated PP be  $z = \tilde{z}_P$ . Solve for  $\tilde{z}_P$  needed to keep  $Q$ 's projection constant as  $z_{\text{COP}}$  varies. Show your work. Note that the  $z$ -coordinate of  $Q$  is fixed at  $2z_P$ , but the  $z$ -coordinate of PP is now a variable,  $\tilde{z}_P$ .
- (3 points) Now consider moving the COP infinitely far back along the positive  $z$ -axis while keeping PP at its original location,  $z = z_P$ . Derive the new projection matrix for this case. Show your work. What is this sort of projection called?

**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 so-called “tangle cube,” described by the equation:

$$x^4 + y^4 + z^4 - 5x^2 - 5y^2 - 5z^2 + 12 = 0$$

In the figure below, the left column shows two renderings of the tangle cube, the middle column illustrates taking a slice through the  $x$ - $y$  plane (at  $z = 0$ ), and the right column shows a slice parallel to the  $x$ - $y$  plane taken toward the bottom of the tangle cube (plane at  $z \approx -1.5$ ):



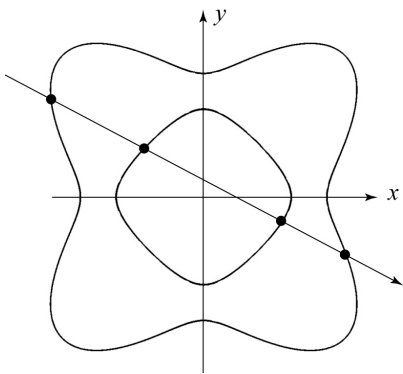
In the next problem steps, you will be asked to solve for and/or discuss ray intersections with this primitive. Performing the ray intersections will amount to solving for the roots of a polynomial, much as it did for sphere intersection. For your answers, you need to keep a few things in mind:

- 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  $A + iB$  is a root, then so is  $A - iB$ .
- 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  $P = (0 \ 0 \ 0)$  and  $\mathbf{d} = (1 \ 1 \ 0)$ . Typically, we normalize  $\mathbf{d}$ , but for simplicity (and without loss of generality) you can work with the un-normalized  $\mathbf{d}$  as given here.
- Solve for all values of  $t$  where the ray intersects the tangle cube (**including** any negative values of  $t$ ). Your solution should be algebraic and lead to the **exact**  $t$  values. Show your work.
  - In the process of solving for  $t$ , you should have computed the roots of a polynomial. How many distinct real roots did you find? How many of them have multiplicity  $> 1$ ? How many complex roots did you find?
  - Which value of  $t$  represents the intersection we care about for ray tracing?

**Problem 4 (cont'd)**

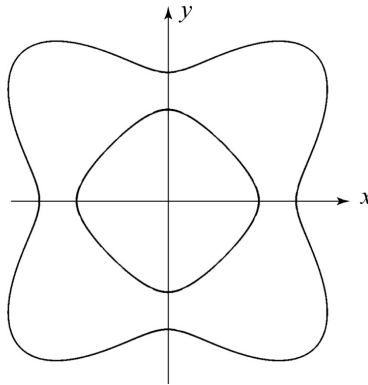
b) (15 points) What are all the possible combinations of roots when ray tracing this surface, not counting the one in part (a)? For each combination, describe the 4 roots as in part (a), draw a ray in the  $x$ - $y$  plane that gives rise to that combination, and place a dot at each intersection point. Assume the origin of the ray is outside of the bounding box of the object. 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**

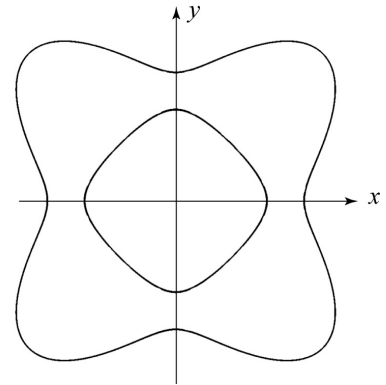
# of complex roots: **0**



# of distinct real roots:

# of real roots w/ multiplicity > 1:

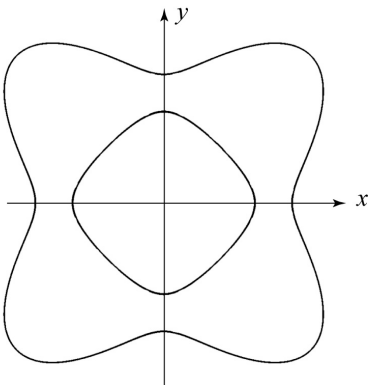
# of complex roots:



# of distinct real roots:

# of roots w/ multiplicity > 1:

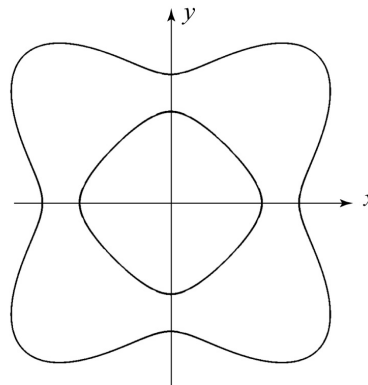
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

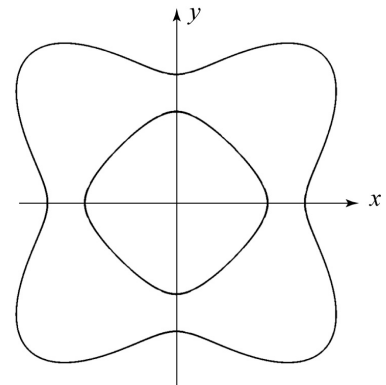
# of complex roots:



# 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  $Q(u) = A_n u^n + A_{n-1} u^{n-1} + \dots + A_0$ , where  $n$  is determined by the number of control points. The coefficients  $A_0, \dots, A_n$  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) The  $C^2$  curve from part (e) actually exhibits higher order continuity. What is the largest value of  $n$  for which the curve is  $C^n$  continuous? Show your work.
- 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.

