Homework #2

Shading, Texture Mapping, Ray Tracing, and Bezier Curves

Assigned: Sunday, May 26th

Due: Thursday, June 13th
at the beginning of class

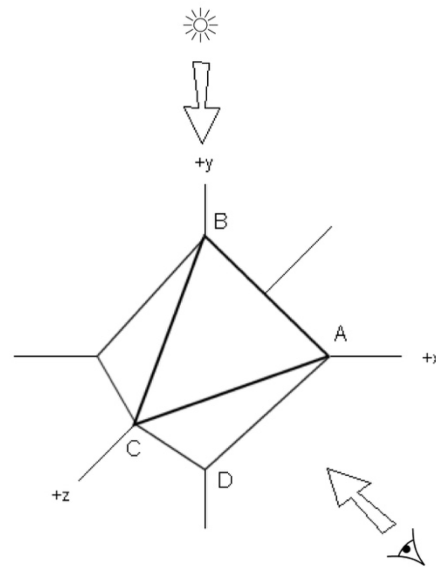
Directions: Please provide short written answers to the following questions on your own paper. Feel free to discuss the problems with classmates, but *please* ***follow the Gilligan's Island rule*, answer the questions on your own, and show your work.***

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 (21 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:



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

where

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

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

(a) (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?

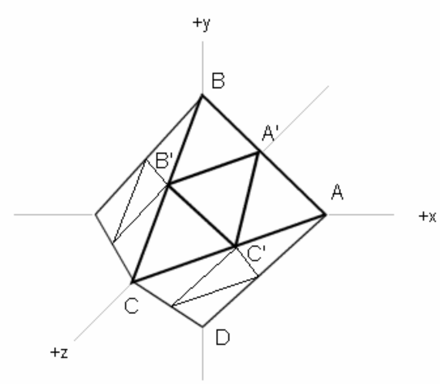
(b) (3 points) Assume that this object is really just a crude approximation of a sphere (e.g., perhaps you are using the octahedron to represent the sphere because your graphics card is slow). If you want to shade the octahedron so that it approximates the shading of a sphere, what would you specify as the unit normal at each vertex of triangle ABC?

(c) (5 points) Given the normals in (b), compute the rendered colors of vertices A, B, and C. Show your work.

(d) (2 points) Again given the normals in (b), describe the appearance of triangle ABC as seen by the viewer using Gouraud interpolation.

(e) (3 points) Now switch from Gouraud-interpolated shading to Phong-interpolated shading. How will the appearance of triangle ABC change (given the normals in (b)?

(f) (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 triangle (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?

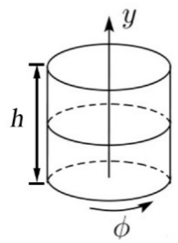


(g) (3 points) If you continued this subdivision process – repeatedly performing 4-to-1 subdivision, repositioning the inserted vertices, and computing their ideal normals – would the Gouraud-interpolated and Phong-interpolated renderings of the refined shape converge toward the same answer, i.e., the appearance of a ray traced sphere? Explain.

Problem 2. Texture mapping (15 points)

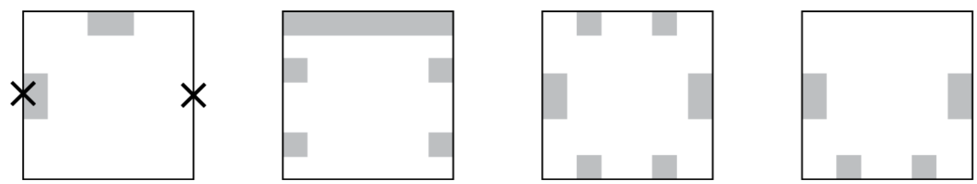
When a texture map is applied to a surface, points that are distinct in the rectangular texture map may be mapped to the same place on the object. For example, when a texture map is applied to a cylinder, the left and right edges of the texture map are mapped to the same place. We call a mapping "valid" if it does not map two points of different colors to the same point on the object. For each of the 16 cases below, indicate whether the mapping is valid (write "Yes" or "No" above the texture map). If it is not valid, mark with an **X** two points on the texture map that map to the same point on the object, but have different colors. Use the texture mapping formulas specified below for each primitive, where the u, v parameters range from 0 to 1. The (u, v) origin of each texture map is in the lower left corner of the texture. The first of the 16 cases below is done for you. ***Please write on this page and include it with your homework solution. You do not need to justify your answers.***

Uncapped cylinder (top and bottom are open)

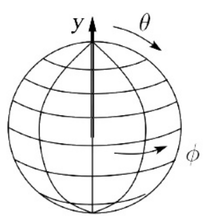


$$\begin{cases} u = \phi/2\pi \\ v = y/h \end{cases}$$

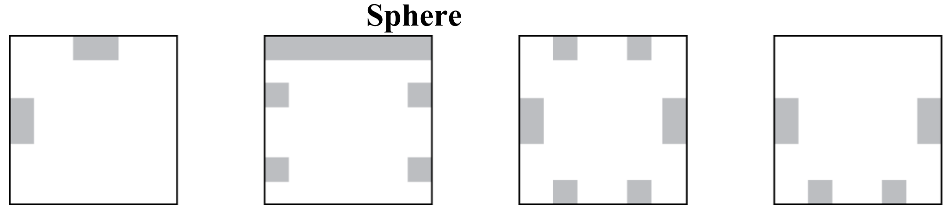
No



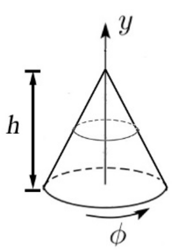
Sphere



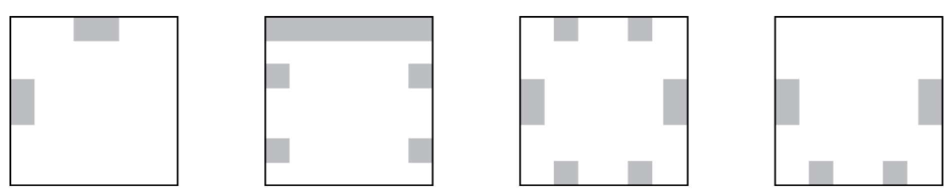
$$\begin{cases} u = \phi/2\pi \\ v = \theta/\pi \end{cases}$$



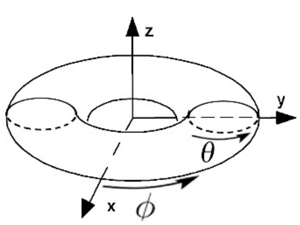
Uncapped cone (bottom is open)



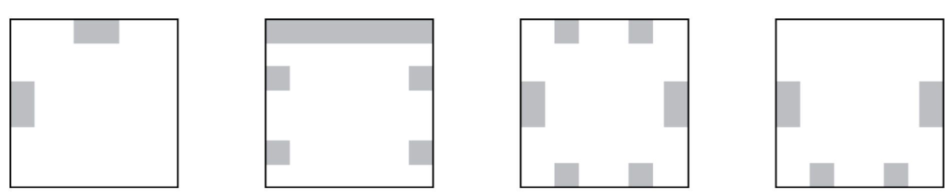
$$\begin{cases} u = \phi/2\pi \\ v = y/h \end{cases}$$



Torus



$$\begin{cases} u = \phi/2\pi \\ v = \theta/2\pi \end{cases}$$

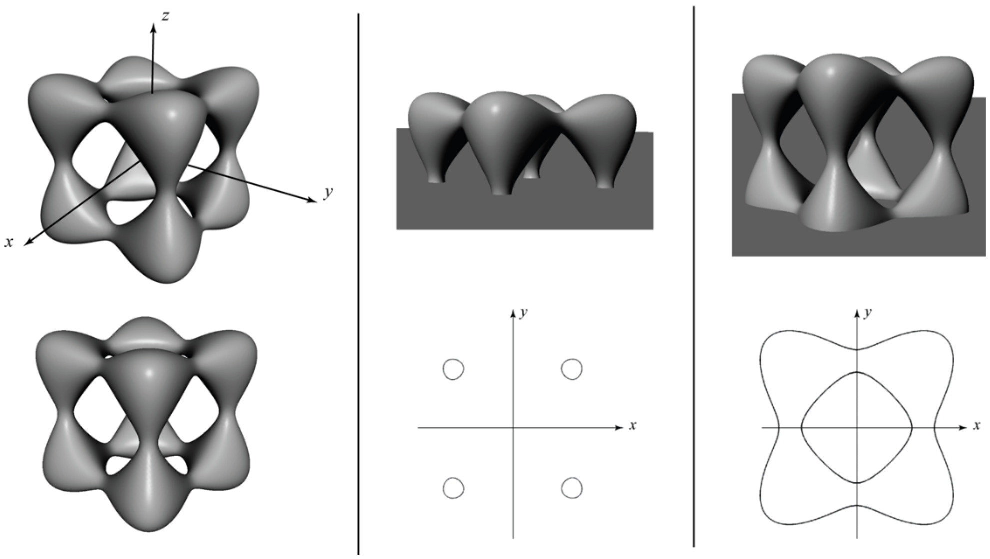


Problem 3. Ray intersection with implicit surfaces (23 points)

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 ≈ -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:

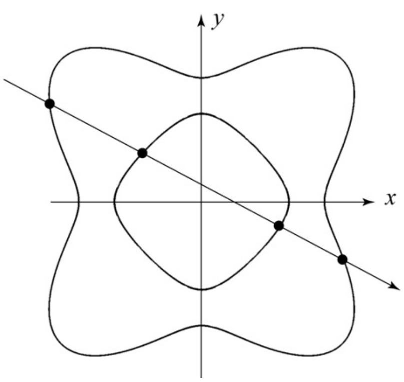
- You will find as many roots as the order (largest exponent) of the polynomial.
- You may find a mixture of real and complex roots. When we say complex here, we mean a number that has a non-zero imaginary component.
- All complex roots occur in complex conjugate pairs. If *A* + i*B* is a root, then 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 $\mathbf{d} = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}$. Typically, we normalize **d,** but for simplicity (and without loss of generality) you can work with the un-normalized **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 3 (cont'd)

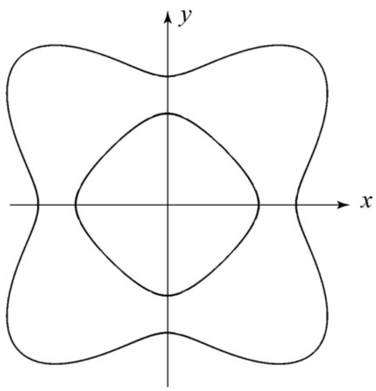
b) (15 points) What are all the possible combinations of roots when ray tracing this surface, not counting the one in part (a)? For each combination, describe the 4 roots as in part (a), draw a ray in the x-y plane that gives rise to that combination, and place a dot at each intersection point. Assume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

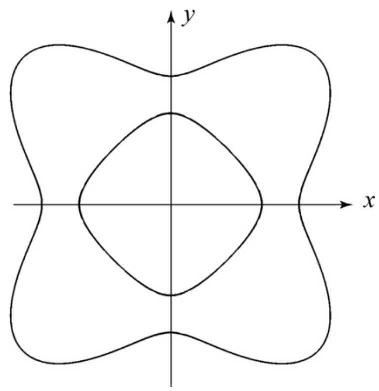
# of complex roots: **0**



# of distinct real roots:

# of real roots w/ multiplicity > 1:

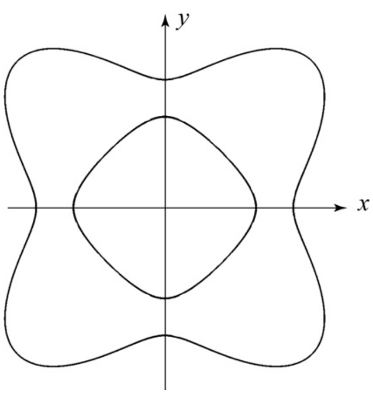
# of complex roots:



# of distinct real roots:

# of roots w/ multiplicity > 1:

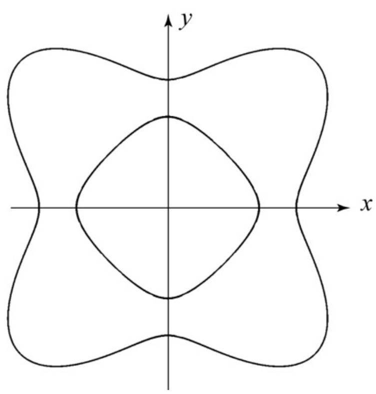
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

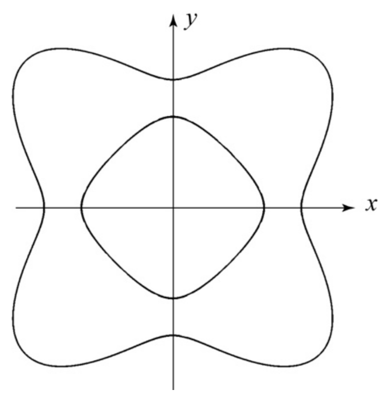
# of complex roots:



# of distinct real roots:

# of real roots w/ multiplicity > 1:

# of complex roots:



# of distinct real roots:

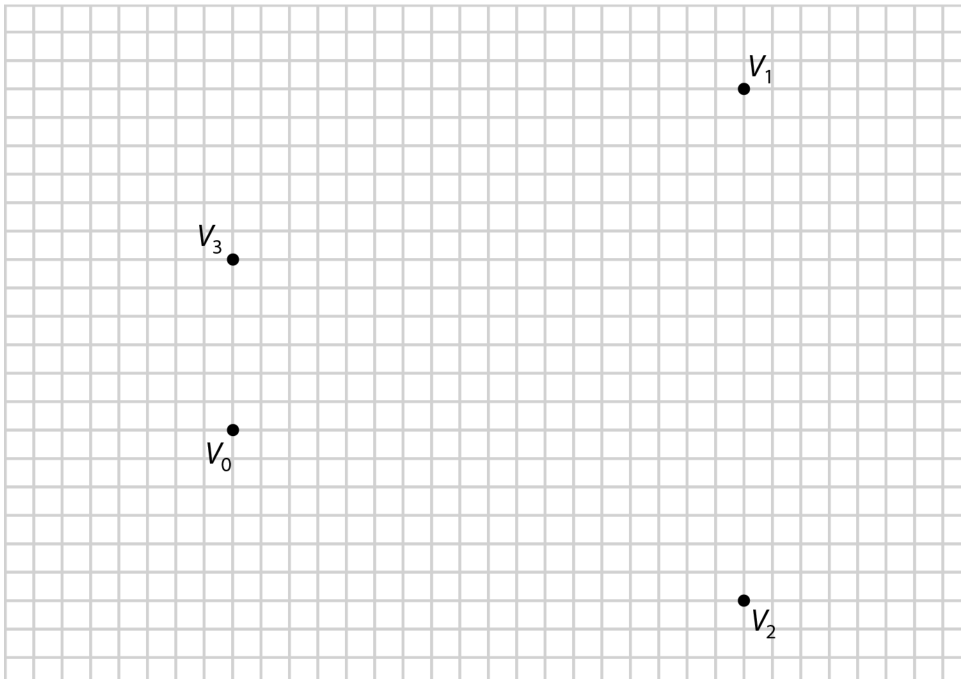
# of real roots w/ multiplicity > 1:

# of complex roots:

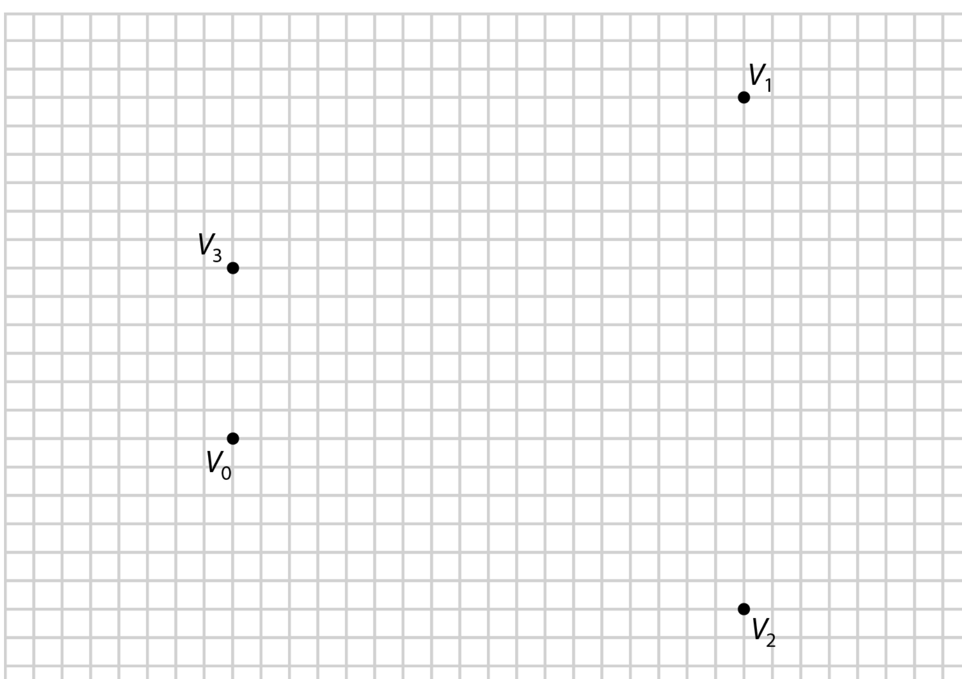
Problem 4. Parametric curves (21 points)

In this problem, we will explore the construction of parametric. ***Please write on the pages for this problem and include them with your homework solution. Also, carefully note the order of the points as given.***

a) (3 points) Given the following Bezier control points, construct all of the de Casteljau lines and points needed to evaluate the curve at u=1/4. Label this point Q(1/4).

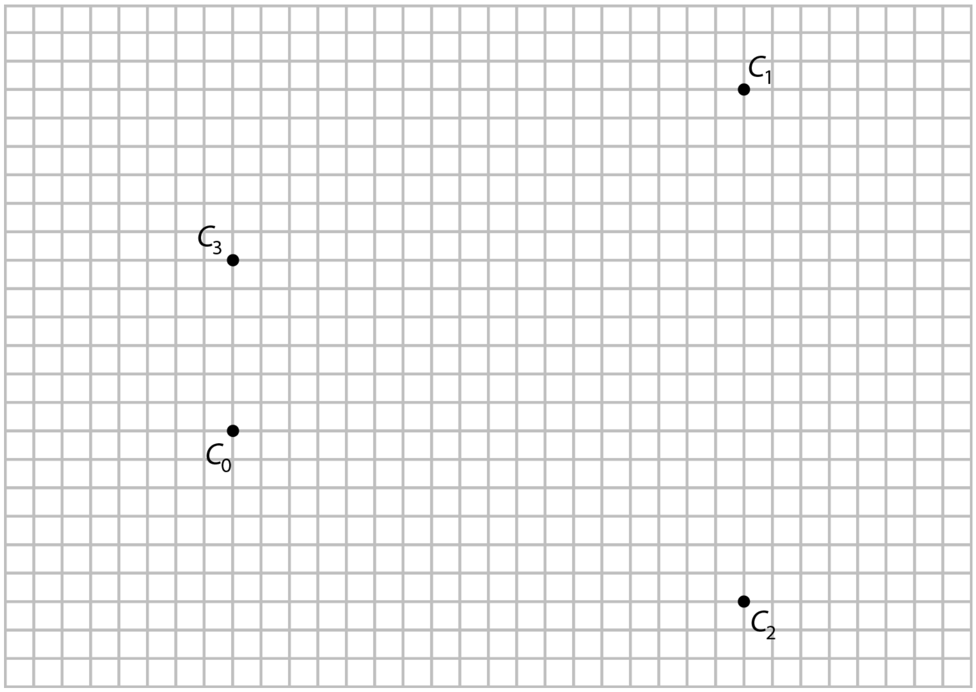


b) (6 points) For the same Bezier control points, shown again below, construct all of the de Casteljau lines and points needed to evaluate the curve at u=1/2. Label this point Q(1/2).Then add the point from part (a) and label it Q(1/4). Now sketch the path the Bezier curve will take. The curve does not need to be exact, but it should conform to some of the geometric properties of Bezier curves (convex hull condition, tangency at endpoints).

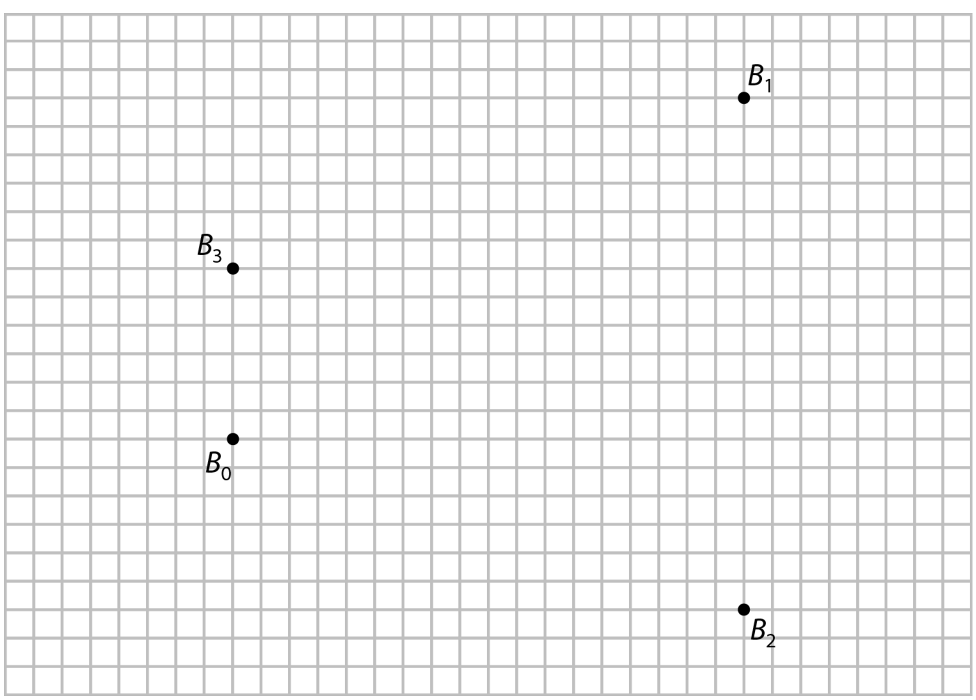


Problem 4 (cont'd)

c) (6 points) Given the following Catmull-Rom control points, construct all of the lines and points needed to generate the Bezier control points for the Catmull-Rom curve. **Use a slackness value of τ=1**. Assume that we insert "phantom" control points $C_{-1}=C_0$ and $C_4=C_3$, so that the spline is endpoint interpolating. You must mark each Bezier point (*including any that coincide with a Catmull-Rom control point*) with an X, but you do not need to label it (i.e., no need to give each Bezier point a name). Sketch the resulting spline curve, respecting the properties of Bezier curves noted above.



d) (6 points) Given the following de Boor points, construct all of the lines and points needed to generate the Bezier control points for the B-spline. Assume that we insert "phantom" control points $B_{-2}=B_{-1}=B_0$ and $B_5=B_4=B_3$, so that the spline is endpoint interpolating. You must mark each Bezier point (*including any that coincide with a de Boor point*) with an X, but you do not need to label it (i.e., no need to give each Bezier point a name). Sketch the resulting spline curve, respecting the properties of Bezier curves noted above.



Problem 5. Animation artifact. (20 Points)

For your final project artifact, you will create a short animation, bringing your model to life to tell a story, to entertain your audience.

Guidelines

- Aim for **30 - 60 seconds**…shorter is usually better. Don't make an animation that feels like "slow motion"!
- Try to use some of the animation principles given in lecture.
- See the project page for pointer to video creation.
- Audio is permitted, though optional.

Turn in

- One artifact per person
- Submit **a representative image**, in addition to final video. The images will help everyone keep track of submissions during voting.
- Due Thursday, June 13 at **12pm sharp**.

Grading

- The course staff will grade your artifact based on technical and artistic merit.
- We understand how hard it is to create these animations and will be gentle in our grading.

Voting

- We will have in-class, non-anonymous voting on Thursday, June 13.
- The staff will preview the submissions and may reduce the number of entrants that are part of the competition, to keep the in-class process lively and not too lengthy. All submissions will still be evaluated by the staff and will appear on the final artifact page, regardless.
- Winners will get more extra credit than usual…and will be awarded cool prizes in class.
- This will be fun!