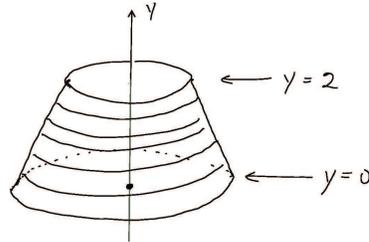


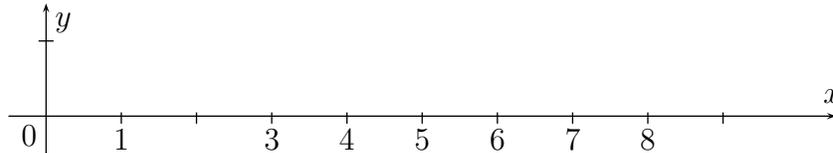
Math 155B — Computer Graphics — Spring 2019
Homework #9 — Due Friday, May 3, 12:00pm (Noon!)
 Hand in via Gradescope — Use separate pages for each problem.

PLEASE NOTE: An additional problem (#5) has been added.

1. A truncated cone is centered on the y -axis. Its base is the radius two disk $\{\langle x, 0, z \rangle : x^2 + z^2 \leq 4\}$ lying in the xz plane. Its top is the radius one disk $\{\langle x, 2, z \rangle : x^2 + z^2 \leq 1\}$ lying in the plane $y = 2$. Express the **front** of the truncated cone as a Bezier patch. You may use any degree Bezier patch you wish.



2. Consider the knot vector $[0, 0, 0, 0, 1, 3, 3, 4, 5, 5, 5, 6, 7, 8, 8, 8, 8]$ with 17 knots u_0, \dots, u_{16} . E.g., $u_4 = 1$ and $u_7 = 4$ and $u_{11} = 6$. Let this define a **degree three** B-spline curve $\mathbf{q}(u)$ with $n + 1$ control points $\mathbf{p}_0, \dots, \mathbf{p}_n$. Answer questions a.-i. below. You will need to make a copy of this graph:



- a. How many control points are needed for this curve? What is the value of n ?
- b. What is the largest value i such that $N_{i,4}(u)$ is defined?
- b. What is the domain of the B-spline curve $\mathbf{q}(u)$?
- c. What is the support of $N_{0,4}$?
- d. Make a copy of the graph shown above. Draw and label the graph of $N_{0,4}$ on the graph.
- e. What is the support of $N_{4,4}$?
- f. What is the support of $N_{5,4}$?
- g. Draw and label the graph of $N_{5,4}$. You can do this on the same graph as used in part d.
- h. For what value ℓ (if any) is it guaranteed that $\mathbf{q}(u)$ is C^ℓ -continuous at $u = 3$?
- i. For what pairs of integers i and j must it be that $\mathbf{q}(i) = \mathbf{p}_j$? (Explicitly list all of them.)

3. Consider the knot vector $[0, 0, 0, 1, 1, 2, 2, 2]$ for a degree two B-spline curve.
 - a. How many control points does this curve use?
 - b. How many blending functions $N_{i,3}(u)$ does this curve use?
 - c. For each of these blending functions $N_{i,3}(u)$, state what its support is, and give its formula.

4. Answer the questions below by specifying the appropriate option i.-v.. The questions assume the shader program has *only* a vertex shader and fragment shader, and also that the shaders do not access any buffers.
 - a. When Gouraud interpolation is used, the Phong lighting calculation is done: (Answer i., ii., iii., iv., or v.)
 - i. Before the vertex shader is called.
 - ii. By the vertex shader.
 - iii. After the vertex shader is called but before the fragment shader is called.
 - iv. By the fragment shader.
 - v. After the fragment shader is called.
 - b. When Phong interpolation is used, the Phong lighting calculation is done:
 - i. Before the vertex shader is called.
 - ii. By the vertex shader.
 - iii. After the vertex shader is called but before the fragment shader is called.
 - iv. By the fragment shader.
 - v. After the fragment shader is called.
 - c. When Gouraud interpolation is used, the Gouraud interpolation calculation is done:
 - i. Before the vertex shader is called.
 - ii. By the vertex shader.
 - iii. After the vertex shader is called but before the fragment shader is called.
 - iv. By the fragment shader.
 - v. After the fragment shader is called.
 - d. Suppose `glDrawArrays` is used to render vertices from data in a VBO using `GL_TRIANGLES`. If the same vertex v appears as a vertex in multiple triangles, then a vertex shader is invoked for v : (See item g. below for the definition of “same”.)
 - i. At most once, possibly not at all.

- ii. Exactly once.
 - iii. One or more times, but possibly not for every triangle containing v .
 - iv. Exactly once per triangle.
- e. Now suppose `glDrawArrays` is used to render vertices from data in a VBO using `GL_TRIANGLE_STRIP`. If the same vertex v appears as a vertex in multiple triangles, then a vertex shader is invoked for v :
- i. At most once, possibly not at all.
 - ii. Exactly once.
 - iii. One or more times, but possibly not for every triangle containing v .
 - iv. Exactly once per triangle.
- f. Suppose `glDrawElements` is used to render vertices from data in a VBO and EBO using `GL_TRIANGLE_STRIP`. If the same vertex v appears as a vertex in multiple triangles, then a vertex shader is invoked for v :
- i. At most once, possibly not at all.
 - ii. Exactly once.
 - iii. One or more times, but possibly not for every triangle containing v .
 - iv. Exactly once per triangle containing v .
- g. For parts d.,e,f., two vertices are considered to be the “same vertex” if they have exactly the same values for their vertex attributes (even if they have different vertex ID’s). Would your answers for parts e. and f. change if two vertices are considered to be the same only if they have the same vertex ID?

5. Prove that every **multiaffine** function $h(x_1, \dots, x_k)$ can be expressed in the form

$$h(x_1, \dots, x_k) = \sum_{J \subseteq \{1, \dots, k\}} \alpha_J x_J,$$

where the α_J ’s are scalars (possibly equal to 0), and for J a subset of $\{1, \dots, k\}$, the term x_J is the product $x_J = \prod_{j \in J} x_j$. (The formula for h is equation (IX.17) on page 323 (PDF page 343) of the current draft A.10.a of the text.) *Hint: Use induction on k . For the induction step, use the multiaffine property to prove that $h(x_1, \dots, x_k)$ is equal to $x_k \cdot h_0(x_1, \dots, x_{k-1}) + h_1(x_1, \dots, x_{k-1})$ where h_0 and h_1 are multiaffine functions.*