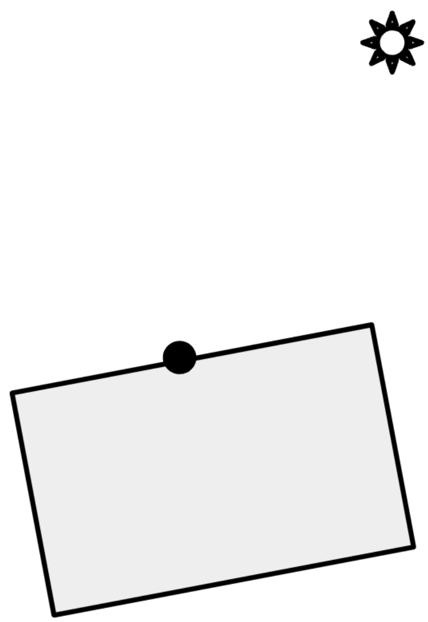
Rasterization, Splines, and Curves 4

CS 184: Foundations of Computer Graphics

1 Graphics Pipeline — Lightning Round!

1. Name and describe the three terms in the Blinn-Phong Reflection Model.

2. A light source shines on a tilted surface. Draw the light direction vector, **l**, and the normal vector, **n**, at the given point.



3. What is the light per unit area on this surface proportional to, according to Lambert's cosine law?

4. Complete the following implementation of the Z-Buffer Algorithm in C++.

```
const int WIDTH = 800; // Width of framebuffer
const int HEIGHT = 600; // Height of framebuffer

struct Color {
    float r, g, b;
};

struct Sample {
    int x, y;
    float z;
    Color color;
};

struct Triangle {
    std::vector<Sample> samples;
};

void zBufferAlgorithm(const ______________________________________ triangles,
                            ______________________________________ framebuffer,
                            ______________________________________ zbuffer) {
    for (const Triangle& T : triangles) {
        for (const Sample& sample : T.samples) {
            int x = sample.x;
            int y = sample.y;
            float z = sample.z;

            if (x >= 0 && x < WIDTH && y >= 0 && y < HEIGHT) {
                if (______________________________________) {
                    framebuffer[x][y] = sample.color;
                    ______________________________________;
                }
            }
        }
    }
}
```

5. Prior to running this algorithm, what should the Z-buffer values be initialized to?

2 A Polynomial Interpolation

Our goal is to fit a polynomial to given points and derivatives of a curve. We can solve this problem by formulating it as a system of linear equations in the coefficients of the polynomial.

1. List all degree 2 polynomials satisfying: $f(0) = 1$, $f(1) = 2$, $f(2) = 5$.

2. How many degree 3 polynomials satisfy the given constraints?

3. Suppose we have a list of constraints:

$$f(0) = p_0,\ f'(0) = d_0,\ f(1) = p_1,\ f'(1) = d_1,\ \ldots,\ f(k) = p_k,\ f'(k) = d_k.$$

   Since we have $2(k+1)$ constraints (one function and one derivative condition per point), the unique interpolating polynomial must have degree $2k+1$.

   For a function f, what are the tradeoffs when either

   - solving for a single degree $2k+1$ polynomial, versus
   - taking the point and derivative constraints at i and $i-1$ for $i = 1, \ldots, k$ and using them to fit k cubic Hermite splines?

4. Consider a cubic polynomial $f(t) = at^3 + bt^2 + ct + d$ that satisfies the following conditions:

$$f(0) = f_0,$$
$$f(1) = f_1,$$
$$f''(0) = f_0'',$$
$$f''(1) = f_1''.$$

Write the matrix that, when inverted and applied to the vector $(f_0, f_1, f_0'', f_1'')^T$, allows you to recover the coefficients $a, b, c,$ and d of the polynomial.

5. Given the numerical inverse of the matrix:

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1/6 & 1/6 \\ 0 & 0 & 1/2 & 0 \\ -1 & 1 & -1/3 & -1/6 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_0'' \\ f_1'' \end{pmatrix}$$

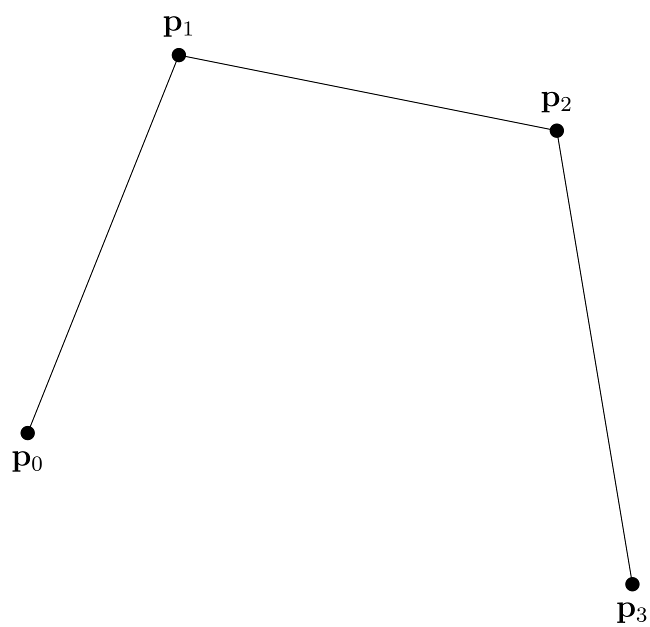
what are the **basis polynomials** that allow expressing $f(t)$ in terms of f_0, f_1, f_0'',, f_1''?

3 de Casteljau's algorithm

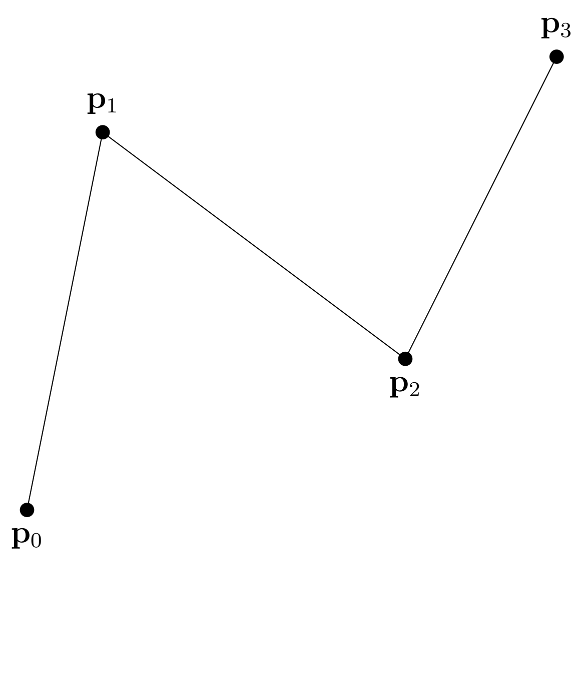
Given $k+1$ points $\mathbf{p}_0, \ldots, \mathbf{p_k}$, create a new set of k points $\mathbf{p'_0}, \ldots, \mathbf{p'_{k-1}}$ by computing $\mathbf{p'_i} = \text{lerp}(\mathbf{p_i}, \mathbf{p_{i+1}}, t)$, where $\text{lerp}(\mathbf{p_i}, \mathbf{p_{i+1}}, t) = (1-t)\mathbf{p_i} + t\mathbf{p_{i+1}}$.

Perform k times to yield a single point, $f(t)$.

1. Use de Casteljau's algorithm to construct $f(1/2)$ with the given control points.



2. Use de Casteljau's algorithm to construct $f(1/3)$ with the given control points.



3. Show that the point with parameter t on the Bézier curve with control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ is given by $s^3\mathbf{p}_0 + 3s^2t\mathbf{p}_1 + 3st^2\mathbf{p}_2 + t^3\mathbf{p}_3$, where $s = 1 - t$. (Hint: apply de Casteljau's algorithm algebraically to the control points. With this setup, linear interpolation between two points $\mathbf{q_0}$ and $\mathbf{q_1}$ looks like $s\mathbf{q_0} + t\mathbf{q_1}$.)

4. What is this matrix product? (Hint: *don't* expand it. Instead, think about what each matrix in the product does. How are they related to de Casteljau's algorithm?)

$$\begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} s & t & 0 \\ 0 & s & t \end{pmatrix} \begin{pmatrix} s & t & 0 & 0 \\ 0 & s & t & 0 \\ 0 & 0 & s & t \end{pmatrix}$$

5. For a Bézier curve defined by 3 control points, what is the degree of the polynomial that results from de Casteljau's algorithm? What about for k points?

[1] Here, we consider the control points $\mathbf{p}_j$ to be variables and the resulting function to be a polynomial in t. If you can fix $\mathbf{p}_j$'s, it is possible to obtain a polynomial with lower degree. For example, if every $\mathbf{p}_j = \mathbf{0}$, then the resulting polynomial is $\mathbf{0}$.