Splines and Curves 3

CS184: Computer Graphics and Imaging

February 12, 2019

1 Polynomial interpolation

In polynomial interpolation, our goal is to fit a polynomial given some information about points and derivatives of the desired curve. As seen in lecture, we can solve this problem by formulating it as a system of linear equations in the coefficients of the polynomial, and then finding a solution to these equations.

1. List all the degree 2 polynomials satisfying: $f(0) = 1$, $f(1) = 2$, $f(2) = 5$? What about degree 3?

2. Suppose we have a list of constraints:

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

For a function f, what are the tradeoffs when either

- solving for a single $2k+1$ degree 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?

3. A cubic polynomial $f(t) = at^3 + bt^2 + ct + d$ is uniquely determined by specifying both its values and its second derivatives at $t = 0$ and $t = 1$ (as opposed to its values and its first derivatives, as in Hermite interpolation). Write out the system of linear equations given by these constraints on $f(0), f(1), f''(0), f''(1)$.

4. Write the matrix which you would invert and apply to the vector $(f(0), f(1), f''(0), f''(1))^T$ to recover $a, b, c,$ and d in the previous problem.

5. Now, invert the matrix and use its columns to identify four "basis polynomials" for this problem.

2 de Casteljau's algorithm

de Casteljau's algorithm allows us to create a smooth Bézier curve from a series of control points. Though we most commonly apply it to four points to get a cubic Bézier curve, it can be applied to any number of points.

In order to find $f(t)$ on a curve defined for $t \in [0, 1]$, de Casteljau gives us the following iterative step:

- 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

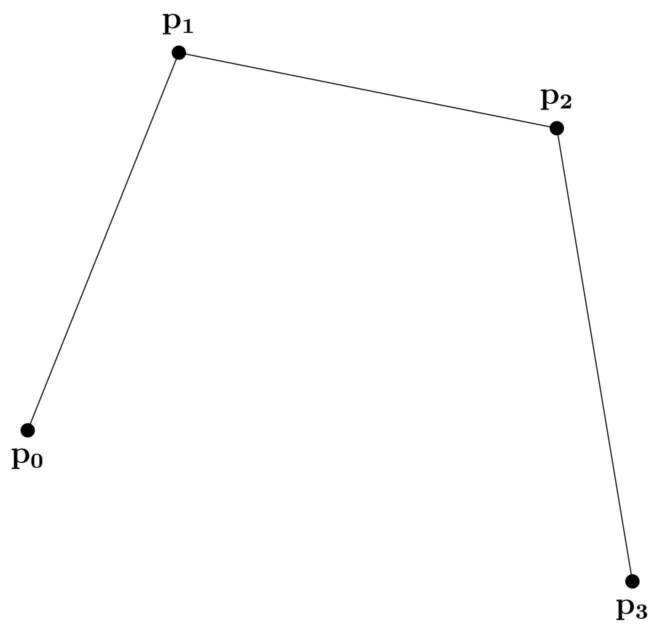
$$\mathbf{p'_i} = \mathrm{lerp}(\mathbf{p_i}, \mathbf{p_{i+1}}, t) \ \ ,$$

  where $\mathrm{lerp}(\mathbf{p_i}, \mathbf{p_{i+1}}, t) = (1-t)\mathbf{p_i} + t\mathbf{p_{i+1}}$.

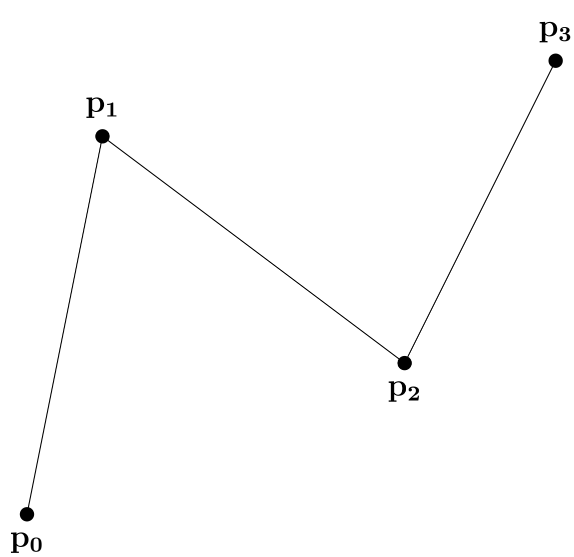
Iteratively applying this step until we are left with a single point yields $f(t)$ for the Bézier curve defined by the initial set of points.

1. For a Bézier curve defined by 3 control points, what is the degree of the polynomial you get from de Casteljau's algorithm? What about for n points?

2. Use de Casteljau's algorithm to find the point where $t = 1/2$ on the Bézier curve defined by these control points.



3. Use de Casteljau's algorithm to find the point where $t = 1/3$ on the Bézier curve defined by these control points.



4. 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 $s\mathbf{q_0} + t\mathbf{q_1}$.)

5. 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?)

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