Overview

In these notes, we develop the quadratic Bézier curve. Previously, we developed this curve through a divide-and-conquer approach, whose basic operation consists of the generation of a single point on the curve, and the division of the curve into two similar components, each of which can be handled separately. This time, we will develop the curve directly, and get a useful parameterization for the curve.

As motivation for this new procedure, let's quickly reexamine the divide-conquer-procedure that we used before.

In this procedure, we started with three control points $P_0$, $P_1$, and $P_2$, and generated three new points, $P_1^{(1)}$, $P_2^{(1)}$, and $P_2^{(2)}$. $P_2^{(2)}$ was defined as a point “on” the curve, and we could use the generated control points to define two curve segments, each defined by three control points, on which we could generate additional points by the same procedure.
To start parameterizing our curve, so that we can write the curve as $P(t)$, we will define

$$P(0) = P_0,$$

$$P(1) = P_2, \text{ and}$$

$$P\left(\frac{1}{2}\right) = P_2^{(2)}.$$

[This looks kind of arbitrary, but it makes some sense because we always used $\frac{1}{2}$ to calculate our points.]

If we take the left subcurve defined by the control points $P_0, P_1^{(1)}, \text{ and } P_2^{(2)}$, it seems reasonable to call the generated point $P\left(\frac{1}{4}\right)$, and call the generated point $P\left(\frac{3}{4}\right)$ for the right subcurve. This relabeling of the points is shown in the following figure:

![Diagram with labeled points showing parameterization](image)

It is fairly easy to see that we can use our subdivision procedure to generate any point that can be labeled as $P\left(\frac{k}{2^n}\right)$, for any integers $n > 0$ and $k > 0$. So for example, we could calculate $P\left(\frac{5}{8}\right)$ by using our subdivision three times.

Now, the insight I want you to get from this “parameterization” of the curve is shown in the following picture.

![Diagram showing parameterization](image)
Here, \( P\left(\frac{3}{4}\right) \) has been calculated by two different methods. The one on the left has been calculated by the subdivision process (essentially calculating \( P\left(\frac{1}{2}\right) \), and then using the subdivision procedure again on the right-generated subcurve). The one on the right has been calculated from the original control points \( P_0, P_1 \) and \( P_2 \) but we modified the method to generate the points \( P_{1}^{(1)}, P_{2}^{(1)} \) and \( P_{2}^{(2)} \) by not taking midpoints, but by taking the points \( \frac{3}{4} \) of the way along the lines. **AND THE RESULTING POINTS ARE THE SAME.**

This suggests that we should be able to calculate our points directly from the original points (thus avoiding all the recursive subdivision steps), and, in fact, this is what we do in the next section.

---

**Development of the Quadratic Bézier Curve**

So, lets start again, but not just calculate midpoints.

Given three control points \( P_0, P_1 \) and \( P_2 \) we develop a method to generate a point on the curve based upon a parameter \( t \), which is a number between 0 and 1 (Note: the illustrations below utilize the value \( t = .75 \)). So here we go...

- First let \( P_{1}^{(1)} \) be the point on the segment \( P_0P_1 \) defined by
  \[
P_{1}^{(1)} = (1-t)P_0 + tP_1
  \]

  ![Diagram of P0, P1, and P1(1)]

  [Note that if \( t = \frac{1}{2} \), we just have our midpoint procedure.]

- then let \( P_{2}^{(1)} \) be the point on the segment \( P_1P_2 \) defined by
  \[
P_{2}^{(1)} = (1-t)P_1 + tP_2
  \]
• and finally let $P_2^{(2)}$ be the point on the segment $\overline{P_1^{(1)}P_2^{(1)}}$ defined by

$$P_2^{(2)} = (1 - t)P_1^{(1)} + tP_2^{(1)}$$

• Finally, we have $P(t) = P_2^{(2)}$.

In reality, we get a similar procedure to our divide-and-conquer method, but now can use values different from $t = \frac{1}{2}$ to generate points on the curve. Each time a new point is calculated, the control points are subdivided into two sets, each of which may be use to generate new subcurves. However, it is now much easier to generate points on the curve for any parameter value.

[Just to make sure that everything makes sense, one should observe that if $t = 0$ the procedure outputs the point $P_0$, and if $t = 1$ the procedure outputs $P_2$]