CUBIC BÉZIER CURVES

Kenneth I. Joy
Visualization and Graphics Research Group
Department of Computer Science
University of California, Davis

Overview

The Bézier curve representation is one that is utilized most frequently in computer graphics and geometric modeling. The curve is defined geometrically, which means that the parameters have geometric meaning – they are just points in three-dimensional space. It was developed by two competing European engineers in the late 1960s to attempt to draw automotive components.

In these notes, we develop the cubic Bézier curve. This curve can be developed through a divide-and-conquer approach similar to the quadratic curve However, in these notes, we will develop a parameterized version of the curve which proceeds almost identically to the development for the quadratic Bézier curve

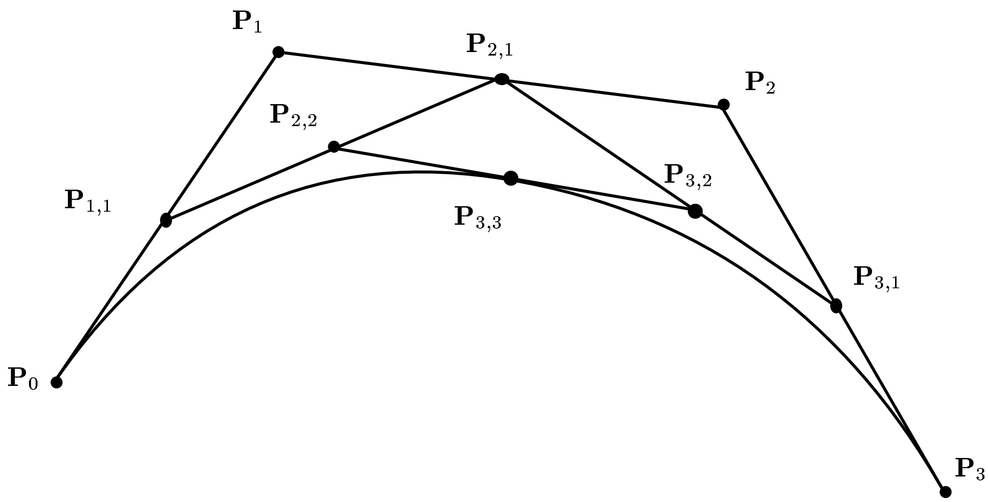
---

Defining The Cubic Bézier Curve

Given four control points, $\mathbf{P}_0,\ \mathbf{P}_1,\ \mathbf{P}_2,\ \mathbf{P}_3$, one can generate a curve $\mathbf{P}(t)$, as we did in the case of the quadratic Bézier curve, by

- let $\mathbf{P}_1^{(1)}(t) \ = \ t\mathbf{P}_1 + (1-t)\mathbf{P}_0$
- let $\mathbf{P}_2^{(1)}(t) \ = \ t\mathbf{P}_2 + (1-t)\mathbf{P}_1$
- let $\mathbf{P}_3^{(1)}(t) \ = \ t\mathbf{P}_3 + (1-t)\mathbf{P}_2$
- let $\mathbf{P}_2^{(2)}(t) \ = \ t\mathbf{P}_2^{(1)}(t) + (1-t)\mathbf{P}_1^{(1)}(t)$
- let $\mathbf{P}_3^{(2)}(t) \ = \ t\mathbf{P}_3^{(1)}(t) + (1-t)\mathbf{P}_2^{(1)}(t)$
- let $\mathbf{P}_3^{(3)}(t) \ = \ t\mathbf{P}_3^{(2)}(t) + (1-t)\mathbf{P}_2^{(2)}(t)$
- $\mathbf{P}_3^{(3)}(t)$ is defined to be $\mathbf{P}(t)$

This construction is shown in the figure below



notice that we did the same process as in the quadratic Bézier curve, but did one additional level. The procedure, as in the quadratic case, produces a point on the curve and subdivides the curve by producing 2 new sets of 4 control points.

Simplifying the above construction, we have

$$\begin{aligned}
\mathbf{P}(t) &= \mathbf{P}_3^{(3)}(t) \\
&= t\mathbf{P}_3^{(2)}(t) + (1-t)\mathbf{P}_2^{(2)}(t) \\
&= t\left[t\mathbf{P}_3^{(1)}(t) + (1-t)\mathbf{P}_2^{(1)}(t)\right] \\
&\quad + \ (1-t)\left[t\mathbf{P}_2^{(1)}(t) + (1-t)\mathbf{P}_1^{(1)}(t)\right] \\
&= t^2\mathbf{P}_3^{(1)}(t) + 2t(1-t)\mathbf{P}_2^{(1)}(t) + (1-t)^2\mathbf{P}_1^{(1)}(t) \\
&= t^2\left[t\mathbf{P}_3 + (1-t)\mathbf{P}_2\right] + 2t(1-t)\left[t\mathbf{P}_2 + (1-t)\mathbf{P}_1\right] \\
&\quad + \ (1-t)^2\left[t\mathbf{P}_1 + (1-t)\mathbf{P}_0\right] \\
&= t^3\mathbf{P}_3 + 3t^2(1-t)\mathbf{P}_2 + 3t(1-t)^2\mathbf{P}_1 + (1-t)^3\mathbf{P}_0
\end{aligned}$$

which is the analytic form of the curve.

Summarizing the Development of the Curve

As in the quadratic case, we have developed two methods for generating points on the curve.

- **The Geometric Method** – Given the control points $\mathbf{P}_0,\ \mathbf{P}_1,\ \mathbf{P}_2,\ \mathbf{P}_3$, and a value $t \in [0,1]$, we can generate the point $\mathbf{P}(t)$ on the Bézier curve by

$$\mathbf{P}(t) \ = \ \mathbf{P}_3^{(3)}(t)$$

where

$$\mathbf{P}_i^{(j)}(t) \ = \ \begin{cases} (1-t)\mathbf{P}_{i-1}^{(j-1)}(t) \ + \ t\mathbf{P}_i^{(j-1)}(t) & \text{if } j \ > \ 0 \\ \mathbf{P}_i & \text{otherwise} \end{cases}$$

- **The Analytic Method** – Given the control points $\mathbf{P}_0,\ \mathbf{P}_1,\ \mathbf{P}_2,\ \mathbf{P}_3$, we define the Bézier curve to be

$$\mathbf{P}(t) \ = \ \sum_{i=0}^{3} \mathbf{P}_i B_{i,3}(t)$$

where

$$\begin{aligned} B_{0,3}(t) &= (1-t)^3 \\ B_{1,3}(t) &= 3t(1-t)^2 \\ B_{2,3}(t) &= 3t^2(1-t) \\ B_{3,3}(t) &= t^3 \end{aligned}$$

the Bernstein polynomials of degree three.

Properties of the Cubic Bézier Curve

The cubic Bézier curve has properties similar to that of the quadratic curve. These can be verified directly from the equations above.

- $\mathbf{P}_0$ and $\mathbf{P}_3$ are on the curve.

- The curve is continuous, infinitely differentiable, and the second derivatives are continuous (automatic for a polynomial curve).
- The tangent line to the curve at the point $\mathbf{P}_0$ is the line $\overline{\mathbf{P}_0\mathbf{P}_1}$. The tangent to the curve at the point $\mathbf{P}_3$ is the line $\overline{\mathbf{P}_2\mathbf{P}_3}$.
- The curve lies within the convex hull of its control points. This is because each successive $\mathbf{P}_i^{(j)}$ is a convex combination of the points $\mathbf{P}_i^{(j-1)}$ and $\mathbf{P}_{i-1}^{(j-1)}$.
- Both $\mathbf{P}_1$ and $\mathbf{P}_2$ are on the curve only if the curve is linear.

Summary

The procedure for developing the cubic Bézier curve is nearly identical to that for the quadratic curve – the primary difference is that we have four control points and must proceed one additional level in the recursion to get a point on the curve. This procedure is extendable so that Bézier curves can be developed for any number of control points.

A MATRIX FORMULATION OF THE CUBIC BÉZIER CURVE

Kenneth I. Joy
Visualization and Graphics Research Group
Department of Computer Science
University of California, Davis

Overview

A cubic Bézier curve has a useful representation in a matrix form. This is a non-standard representation but extremely valuable if we can multiply matrices quickly. The matrix which we develop, when examined closely, is uniquely defined by the cubic Bernstein polynomials. We can use this form to develop "subdivision matrices" that allow us to use matrix multiplication to generate different Bézier control polygons for the cubic curve.

Developing the Matrix Equation

A cubic Bézier Curve can be written in a matrix form by expanding the analytic definition of the curve into its Bernstein polynomial coefficients, and then writing these coefficients in a matrix form using the polynomial power basis. That is,

$$\begin{aligned}
\mathbf{P}(t) &= \sum_{i=0}^{3} \mathbf{P}_i B_i(t) \\
&= (1-t)^3\mathbf{P}_0 + 3t(1-t)^2\mathbf{P}_1 + 3t^2(1-t)\mathbf{P}_2 + t^3\mathbf{P}_3 \\
&= \begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}
\end{aligned}$$

and so a cubic Bézier curve is can be written in a matrix form of

$$\begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}$$

where

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

The matrix M defines the blending functions for the curve $\mathbf{P}(t)$ – i.e. the cubic Bernstein polynomials. In reality there are three equations here, one for each of the x, y and z components of $\mathbf{P}(t)$.

Utilizing equipment that is designed for fast 4×4 matrix calculations, this formulation can be used to quickly calculate points on the curve.

Subdivision Using the Matrix Form

Suppose we wish to generate the control polygon for the portion of the curve $\mathbf{P}(t)$ where t ranges between 0 and $\frac{1}{2}$ – subdivide the curve at the point $t = \frac{1}{2}$. This can be done by defining a new curve $\mathbf{Q}(t)$ which is equal to $\mathbf{P}(\frac{t}{2})$. Clearly this new curve is a cubic polynomial, and traces out the desired portion of $\mathbf{P}$ as t ranges between 0 and 1. We can calculate the Bézier control polygon for $\mathbf{Q}$ by using the matrix form of the curve $\mathbf{P}$.

$$\begin{aligned}
\mathbf{Q}(t) &= \mathbf{P}(\frac{t}{2}) \\
&= \begin{bmatrix} 1 & (\frac{t}{2}) & (\frac{t}{2})^2 & (\frac{t}{2})^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M S_{[0,\frac{1}{2}]} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}
\end{aligned}$$

where the matrix $S_{[0,\frac{1}{2}]}$ is defined as

$$
\begin{aligned}
S_{[0,\frac{1}{2}]} &= M^{-1}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} M \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{6} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{12} & 0 \\ 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{bmatrix}
\end{aligned}
$$

So $\mathbf{Q}(t)$ is a Bézier curve, with a control polygon given by

$$
\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{bmatrix}\begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \frac{1}{2}\mathbf{P}_1 + \frac{1}{2}\mathbf{P}_0 \\ \frac{1}{4}\mathbf{P}_2 + \frac{1}{2}\mathbf{P}_1 + \frac{1}{4}\mathbf{P}_0 \\ \frac{1}{8}\mathbf{P}_3 + \frac{3}{8}\mathbf{P}_2 + \frac{3}{8}\mathbf{P}_1 + \frac{1}{8}\mathbf{P}_0 \end{bmatrix}
$$

In the same way, we can obtain the Bézier control polygon for the second half of the curve – the portion where t ranges between $\frac{1}{2}$ and 1. If we call this new curve $\mathbf{Q}(t)$, then

$$\begin{aligned}
\mathbf{Q}(t) &= \mathbf{P}(\frac{1}{2}+\frac{t}{2}) \\
&= \begin{bmatrix} 1 & (\frac{1}{2}+\frac{t}{2}) & (\frac{1}{2}+\frac{t}{2})^2 & (\frac{1}{2}+\frac{t}{2})^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{8} \\ 0 & 0 & \frac{1}{4} & \frac{3}{8} \\ 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M S_{[\frac{1}{2},1]} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}
\end{aligned}$$

where

$$S_{[\frac{1}{2},1]} = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

obtaining a matrix that can be applied to the original Bézier control points to produce Bézier control points for the second half of the curve.

Generating a Sequence of Bézier Control Polygons.

Using matrix calculations similar to those above, we can generate an iterative scheme to generate a sequence of points on the curve. To do this, we need one additional S matrix. If we consider the portion of the cubic curve $\mathbf{P}(t)$ where t ranges between 1 and 2, We generate the Bézier control points of $\mathbf{Q}(t)$ by reparameterization of the original curve – namely by replacing t by $t+1$ – to obtain

$$\begin{aligned}
\mathbf{Q}(t) &= \mathbf{P}(t+1) \\
&= \begin{bmatrix} 1 & (t+1) & (t+1)^2 & (t+1)^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M S_{[1,2]} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}
\end{aligned}$$

where, after some calculation, $S_{[1,2]}$ is given by

$$S_{[1,2]} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 2 \\ 0 & 1 & -4 & 4 \\ -1 & 6 & -12 & 8 \end{bmatrix}$$

Now, using a combination of $S_{[0,\frac{1}{2}]}$, $S_{[\frac{1}{2},1]}$ and $S_{[1,2]}$, we can produce Bézier control polygons along the curve similar to methods developed with divided differences. To see what I mean here, first notice that

$$S_{[1,2]} S_{[0,\frac{1}{2}]} = S_{[\frac{1}{2},1]}$$

This states that by applying $S_{[0,\frac{1}{2}]}$ to obtain a Bézier control polygon for the first half of the curve, we can then apply $S_{[1,2]}$ to this control polygon to obtain the Bézier control polygon for the second half of the curve.

Extending this, if we apply

$$S^i_{[1,2]} S^k_{[0,\frac{1}{2}]}$$

(that is, apply $S_{[0,\frac{1}{2}]}$ k times and then $S_{[1,2]}$ i times), we obtain the Bézier control polygon for the portion of the curve where t ranges between $\frac{i}{2^k}$ and $\frac{i+1}{2^k}$. By repeatedly applying $S_{[1,2]}$, we move our control polygons along the curve.

Summary

We have developed a matrix form for the cubic Bézier curve. Using reparameterization, we then developed matrices which enabled us to produce Bézier control polygons for sections of the curve, and to move from one Bézier control polygon to an adjacent for on the curve. These operations are extremely useful when utilizing hardware with geometry engines that multiply 4×4 matrices rapidly.
