Cubic Bèzier Curves

Shavez Kaleem

Al Abrahamsen

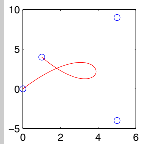
December 15, 2000

Abstract

This is a brief write up explaining and demonstrating cubic Bézier curves, how they can be represented in a matrix form, and showing how they can be used to generate fonts with the aid of matlab.

1. Introduction

The Bézier curve representation 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 drawing automotive components. The two engineers were Pierre Bézier who worked for Renault and Paul de Casteljau who worked for Citroën. The curve was named after Pierre Bézier, even though Casteljau first used the curve. Bézier was the first to publish and therefore the idea bears his name. Later the curve was developed in the 1970s for CAD/CAM operations. This



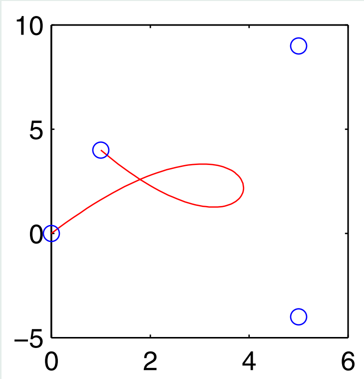


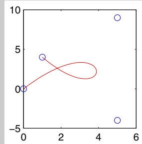
Figure 1: Example of a Cubic Bézier Curve

paper will discuss only cubic Bézier curves in two dimensions, although the application does expand to three dimensions.

2. Deriving the Curve

The first step to understanding Bézier curves is knowing how the curves are geometrically formed. The construction of a Bézier curve begins with picking three or more points, called control points. For the purposes of this paper we will be using four control points, P_0, P_1, P_2, and P_3 (see Figure 2), to create a Cubic Bézier curve.

The next step is to find the points on the line segments P_0P_1, P_1P_2, and P_2P_3. This is best done when thinking about the points as vectors. The first point is P_{1a} and it lies $t\%$



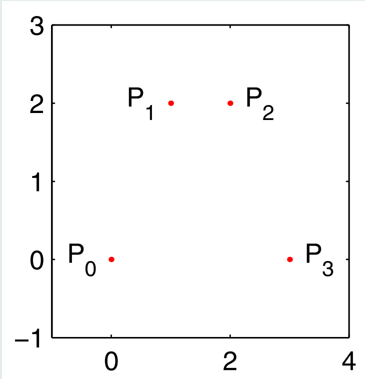
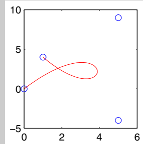


Figure 2: Four Control Points

of the way from point P_0 to P_1 (See Figure 3). This point is derived by:

$$
\begin{aligned}
P_{1a} &= P_0 + t(P_1 - P_0) \\
&= P_0 + tP_1 - tP_0 \\
&= (1-t)P_0 + tP_1
\end{aligned}
\tag{1}
$$

Repeating the process five more times we get the other points that form Figure 4. Only P_{1c} is on the actual curve. To find another point on the curve we repeat the process with a different t value, ranging from 0 to 1.



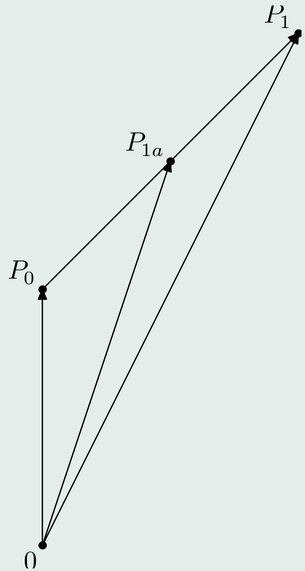
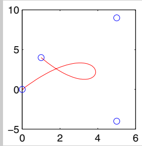


Figure 3: Formation of Individual Points



Home Page
Title Page
Page 4 of 13
Go Back
Full Screen
Close
Quit

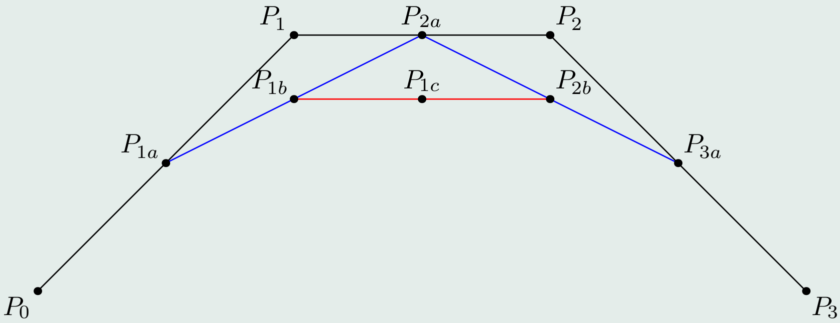
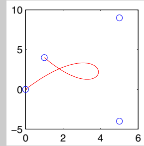


Figure 4: Formation of the first point P_{1c} on the curve

$$
\begin{aligned}
P_{1a}(t) &= (1-t)P_0 + tP_1 \\
P_{2a}(t) &= (1-t)P_1 + tP_2 \\
P_{3a}(t) &= (1-t)P_2 + tP_3 \\
P_{1b}(t) &= (1-t)P_{1a}(t) + tP_{2a}(t) \\
P_{2b}(t) &= (1-t)P_{2a}(t) + tP_{3a}(t)
\end{aligned}
$$

$$P_{1c}(t) = (1-t)P_{1b}(t) + tP_{2b}(t) \tag{2}$$

Using Equation 2 we can form a specific polynomial called the Bernstein Polynomial (Equation 3) with the variable t. The Bernstein Polynomial can be derived by setting $P(t) = P_{1c}(t)$.



$$
\begin{aligned}
P(t) = P_{1c}(t) &= (1-t)P_{1b}(t) + tP_{2b}(t) \\
&= (1-t)[(1-t)P_{1a}(t) + tP_{2a}(t)] + t[(1-t)P_{2a}(t) + tP_{3a}(t)] \\
&= (1-t)[(1-t)[(1-t)P_0 + tP_1 + t((1-t)P_1 + tP_2)]] \\
&\quad + t[(1-t)[(1-t)P_1 + tP_2 + t((1-t)P_2 + tP_3)]] \\
&= (1-t)[(1-t)^2 P_0 + t(1-t)P_1 + t(1-t)P_1 + t^2 P_2] \\
&\quad + t[(1-t)^2 P_1 + t(1-t)P_2 + t(1-t)P_2 + t^2 P_3 \\
&= (1-t)^3 P_0 + t(1-t)^2 P_1 + t(1-t)^2 P_1 + t^2(1-t)P_2 \\
&\quad + t(1-t)^2 P_1 + t^2(1-t)P_2 + t^2(1-t)P_2 + t^3 P_3
\end{aligned}
$$

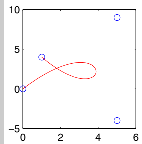
$$
P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \tag{3}
$$

Because we now have a polynomial that can give us the points on the curve we could consider ourselves lucky; however, since there are points P_0, P_1, P_2, P_3 in the polynomial the desired curve is a little hard to generate. To find other points on the curve without having to recalculate P_{1c} every time we put the Bernstein Polynomial in matrix form.

This is done by looking at the polynomial as a linear combination of of the four control points and their coefficients. We can then break the coefficient vector into a vector times a matrix.

$$
P(t) = \begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}
$$

To break the coefficients into a vector and a matrix, the coefficients have to be expanded.



$$P(t) = \begin{bmatrix} 1 - 3t + 3t^2 - t^2 & 3t - 6t^2 + 3t^3 & 3t^2 - 3t^3 & t^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Now the vector can be expanded to include a matrix which will isolate the t values and allow us to quickly calculate multiple points on our Bézier curve.

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{4}$$

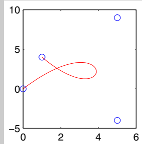
With the matrix representation of the Bernstein Polynomial, multiple values of t can be quickly entered and calculated using a computer to generate points on the Bézier curve.

3. Subdivisions and Generating New Control Points

Sometimes it is useful to adjust part of a curve and not the whole thing. The easiest way to do this is to subdivide the curve into parts and find new control points for each of the subdivisions. To do this, take the matrix form of the Bernstein Polynomial Equation (4), then decide which part of the curve needs to be changed. For this example, the curve will be divided into two equal parts. In order to do this, the Bernstein Polynomial needs to be reparamerterized, which is easily done by adjusting t.

$$P(t) = R(t/2) + R(1/2 + t/2) \tag{5}$$

Taking the first part of the reparameterization of $P(t)$, $R(t/2)$, which is the first half of $P(t)$, and writing in matrix form, the control points of the matrix can be determined. Reparameterizing $P(t)$ we get the matrix equation:



Introduction
Deriving the Curve
Subdivisions and . . .
Properties of Cubic . . .
An Application of . . .
Conclusion
Bibliography
Home Page
Title Page
Go Back
Full Screen
Close
Quit

$$\begin{bmatrix}1 & t/2 & (t/2)^2 & (t/2)^3\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}P_0\\ P_1\\ P_2\\ P_3\end{bmatrix}$$

Next we expand the vector T_t into a vector matrix form labeling the matrix $N_{(1/2t)}$. We get:

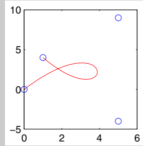
$$\begin{bmatrix}1 & t/2 & (t/2)^2 & (t/2)^3\end{bmatrix} = \begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1/2 & 0 & 0\\ 0 & 0 & 1/4 & 0\\ 0 & 0 & 0 & 1/8\end{bmatrix}$$

Putting this new matrix into our equation $P(t)$ we get $R(t)$ which is exactly half of $P(t)$ but this does not find the new control points. To find these points we have to multiply the P_p vector of the points by a relationship of N. In other words we need to put the matrix equation into a form resembling $R(t) = T_t * M * N_{(0,1/2)} * P_p$. We know $R(t) = T_t * N_{(t/2)} * M * P_p$, which leaves us with the matrix equation $N_{(t/2)} * M = M * N_{(0,1/2)}$.

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1/2 & 0 & 0\\ 0 & 0 & 1/4 & 0\\ 0 & 0 & 0 & 1/8\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}? & ? & ? & ?\\ ? & ? & ? & ?\\ ? & ? & ? & ?\\ ? & ? & ? & ?\end{bmatrix}$$

Multiplying both sides by M^{-1}, we get our $N_{(0,1/2)}$.

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1/3 & 0 & 0\\ 1 & 2/3 & 1/3 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1/2 & 0 & 0\\ 0 & 0 & 1/4 & 0\\ 0 & 0 & 0 & 1/8\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix} = \begin{bmatrix}? & ? & ? & ?\\ ? & ? & ? & ?\\ ? & ? & ? & ?\\ ? & ? & ? & ?\end{bmatrix}$$



Introduction
Deriving the Curve
Subdivisions and . . .
Properties of Cubic . . .
An Application of . . .
Conclusion
Bibliography
Home Page
Title Page
Page 8 of 13
Go Back
Full Screen
Close
Quit

Now we have calculated the matrix $N_{(0,1/2)}$.

$$N_{(0,1/2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \end{bmatrix}$$

Using the $N_{(0,1/2)}$ we found we can now multiply our control points vector P_p on the left by $N_{(0,1/2)}$ to generate our new points for $R(t)$. This same process can be done for $R(1/2 + t/2)$ to generate the $N_{(1/2,1)}$ matrix which can be used to find the new control points for the other half of the curve $P(t)$.

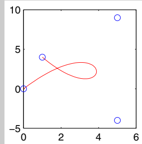
$$N_{(0,1/2)} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} P'_0 \\ P'_1 \\ P'_2 \\ P'_3 \end{bmatrix}$$

$$\begin{bmatrix} P'_0 \\ P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} = \begin{bmatrix} P_0 \\ 1/2P_1 + 1/2P_0 \\ 1/4P_2 + 1/2P_1 + 1/4P_0 \\ 1/8P_3 + 3/8P_2 + 3/8P_1 + 1/8P_0 \end{bmatrix}$$

Calculating $N_{(1/2+1/2t)}$ similarly to what we did for $N_{(t/2)}$

$$N_{1/2+t/2)} = \begin{bmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1/2 & 1/2 & 3/8 \\ 0 & 0 & 1/4 & 3/8 \\ 0 & 0 & 0 & 1/8 \end{bmatrix}$$

$$N_{(1/2,1)} = \begin{bmatrix} 1/8 & 3/8 & 3/8 & 1/8 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$N_{(1/2+t/2)}$ is similar to $N_{(1/2,1)}$ as $N_{(t/2)}$ is similar to $N_{(0,1/2)}$. Once we have one we can easily find the other using the matrix M. This works for any subdivision of the original matrix and allows us to find the new control points for the subdivision. Once the subdivisions are found we can move two of the control points, P_1' or P_2', to change just part of the curve. This tool is highly practical in drafting and allows for more complex changes.

4. Properties of Cubic Bézier Curves

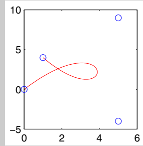
The cubic bézier curve has some basic properties to it. These can be verified from the given equations portrayed earlier.

1. P_0 and P_3 are on the curve.
2. The curve is continuous, infinitely differentiable, and the second derivatives are continuous.
3. The tangent line to the curve at the point P_0 is the line P_0P_1. The tangent to the curve at the point P_3 is the line P_2P_3.
4. Both P_1 and P_2 are on the curve only if the curve is linear.

The figure Figure 5 shows that the point P_0 is tangent to the line P_0P_1. This is emphasized with the equation $y = x^3 + x^2 + x$, which shares tangent lines at the the two control points P_0 and P_3. It can also be noted the the Bézier curve is completely encased by the box formed from connecting the four control points. Because the Bézier curve can be represented by a Bernstein Polynomial it can be differentiated.

5. An Application of Bézier Curve

Much of the applications of Bézier curves deals with the generation of computer fonts. With the use of matlab, we were able to generate Bézier curves that produced a letter



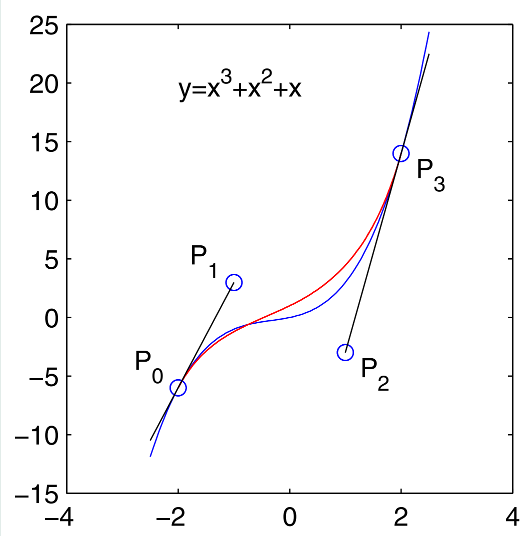
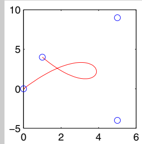
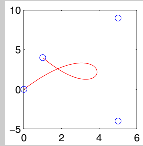


Figure 5: P_0P_1 and P_2P_3 are tangent to P_0 and P_3



Home Page
Title Page
Page 11 of 13
Go Back
Full Screen
Close
Quit



Home Page
Title Page
Page 12 of 13
Go Back
Full Screen
Close
Quit

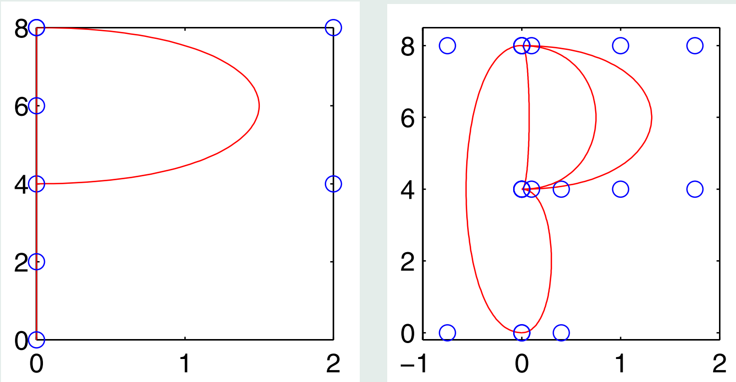


Figure 6: Changing a simple P using Bézier curves

which can be manipulated to a form a font. Below are some results of what we produced.

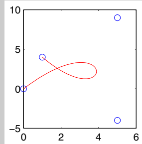
The P is elaborated and changed to a more characteristic and exciting P by expanding and adding Bźier curves. Most fonts are generated using Bézier curves but are generally much more complex.

6. Conclusion

We have seen how the Bézier curves are developed. The curve can be developed by a geometric approach. Using that principle, we developed a parameterization of the cubic Bézier curve, the Bernstein Polynomial. A cubic Bézier curve has a useful representation in a matrix form. This is a non-standard representation but extremely valuable is we can multiply matrices quickly. The matrix that we developed earlier, when examined closely, was uniquely defined by the cubic Bernstein Polynomials. We used this form to develop subdivision matrices that allowed us to use matrix multiplication to generate different Bézier control points for new cubic curves. Bézier curves have a great being importance in the computer graphics industry. The automobile industry has also incorporated Bézier curves for part designs for which it was originally designed.

References

[1] http://graphics.cs.ucdavis.edu/CAGDNotes/Bezier-Curves/Bezier-Curves.html



Home Page
Title Page
Page 13 of 13
Go Back
Full Screen
Close
Quit