Bezier Curves

An Introduction

Detlef Reimers
detlefreimers@gmx.de
http://detlefreimers.de

September 1, 2011

Chapter 1

Bezier Curve Basics

1.1 Linear Interpolation

This section will give you a basic introduction to bezier curves. We begin with the simplest curve form, a straight line. We can express any point $\mathbf{x}$ on this straight line by the two given points and the parameter t:

$$\mathbf{x} = \mathbf{x}(t) = (t-1)\mathbf{a} + t\mathbf{b}. \tag{1.1}$$

For $t = 0$ the straight line passes through $\mathbf{a}$, and for $t = 1$ it passes through $\mathbf{b}$. For $0 \le t \le 1$ the point $\mathbf{x}$ is between $\mathbf{a}$ and $\mathbf{b}$, while for all other values of t it is outside (see Figure 1).

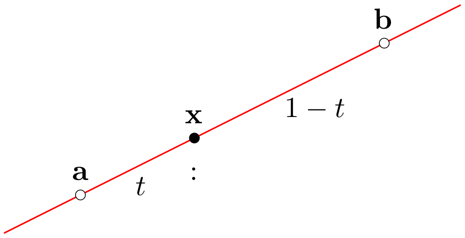

Figure 1: *Line interpolation on a straight line*

The point $\mathbf{x}$ divides the straight line segment between $\mathbf{a}$ and $\mathbf{b}$ in the ratio $t : 1-t$ and any point on the straight line can be calculated by changing this parameter. If t lies between $\mathbf{a}$ and $\mathbf{b}$, then $t \in [0,1]$. We can remove this restriction for t, if we define $t = (u-a)/(b-a)$ with $u \in [a,b]$, we get:

$$\mathbf{x} = \mathbf{x}(t) = \frac{b-u}{b-a}\mathbf{a} + \frac{u-a}{b-a}\mathbf{b}. \tag{1.2}$$

For $u = a$ the point is $\mathbf{x} = \mathbf{a}$ and for $u = b$ it is $\mathbf{x} = \mathbf{b}$ accordingly. Notice, that the value of the ratio $t : 1-t$ did not change after this reparametrisation.

$$ratio(\mathbf{a}, \mathbf{x}, \mathbf{b}) := \frac{t}{1-t} = \frac{u-a}{b-u}. \tag{1.3}$$

One of the most fundamental properties of the linear interpolation is its *affine invariance*. This means, if you apply any affine map like translation, scaling, rotation, shearing or parallel projection to the points and then calculate $\mathbf{x}$, the result will the same as if you first calculated the point and then applied the affine map to the three points.

It should also be mentioned, that from a numerical point of view the linear interpolation is a stable mathematical operation. This means that small changes in the supplied values never lead to sudden great changes in the result. The well known Horner's schema for instance, which can be used to evaluate polynomials, is a bad example in this area.

1.2 Quadratic Bezier Curves

We now want to use the results from the linear case to develop the most elementary nonlinear curve form, the *parabola.* The main idea is very simple: we use repeated linear interpolations to compute a point on a parabola (a quadratic curve). Let $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ be three given points and let $t \in \Re$. Now we build:

$$\begin{aligned}\mathbf{p}_0^1(t) &= (1-t)\mathbf{p}_0 + t\mathbf{p}_1 \\ \mathbf{p}_1^1(t) &= (1-t)\mathbf{p}_1 + t\mathbf{p}_2 \\ \mathbf{p}_0^2(t) &= (1-t)\mathbf{p}_0^1(t) + t\mathbf{p}_1^1(t).\end{aligned}$$

Inserting the first two equations into the third one, we get:

$$\mathbf{p}_0^2(t) = (1-t)^2\mathbf{p}_0 + 2(1-t)t\mathbf{p}_1 + t^2\mathbf{p}_2. \tag{1.4}$$

This formula represents a quadratic expression in t. We use the superscript here to denote the curve degree. So, we can say that $\mathbf{p}_0^2$ traces out a parabolic curve, if t varies from $-\infty$ to $+\infty$. The first three expressions clearly show that we only used repeated linear interpolation to compute a curve point and we can state, that this curve construction is *affine invariant.* Look at Figure 2 for the geometric construction of the curve point $\mathbf{p}_0^2$ for $t = 1/3$.

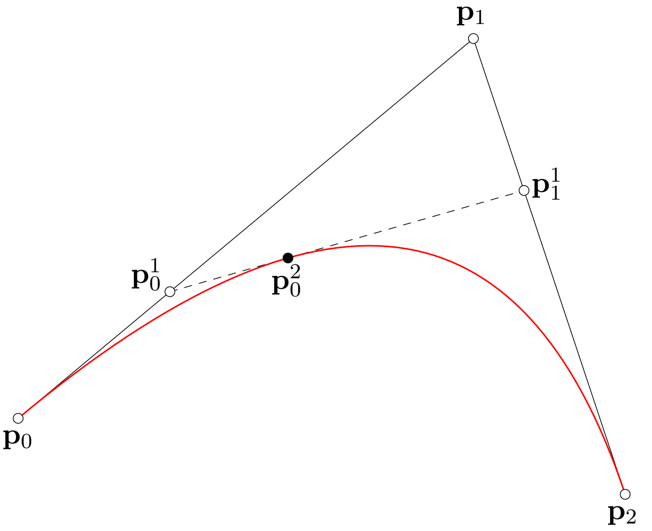

Figure 2: *Repeated Line interpolation on a parabola*

For $t \in [0,1]$ the curve lies in the triangle formed by $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$. This is called the *convex hull property.* As special points we have $\mathbf{p}^2(0) = \mathbf{p}_0$ and $\mathbf{p}^2(1) = \mathbf{p}_2$. If the point $\mathbf{p}_1$ only lies on the curve, it must be a straight line. The construction also shows the following property:

$$ratio(\mathbf{p}_0, \mathbf{p}_0^1, \mathbf{p}_1) = ratio(\mathbf{p}_1, \mathbf{p}_1^1, \mathbf{p}_2) = ratio(\mathbf{p}_0^1, \mathbf{p}_0^2, \mathbf{p}_1^1) = \frac{t}{1-t}. \tag{1.5}$$

All ratios are equal, this proves the affine invariance of the curve construction. If you look at the generated curve, you see that for $t = 0$ the curve is tangent to the line $\mathbf{p}_0\mathbf{p}_1$ and for $t = 1$ it is tangent to the line $\mathbf{p}_1\mathbf{p}_2$. We come back to this fact later.

The geometric construction is based on the principle of repeated linear interpolation and the curve point is obtained by the last interpolation. This principle is the underlying concept for the construction of all *bezier curves* of any degree n. If we want to construct an n degree curve, we need $n+1$ *control points.* The number of linear interpolations, needed to compute a point on a curve of degree n, is:

$$N = \frac{n(n+1)}{2} \tag{1.6}$$

1.3 Cubic Bezier Curves

Parabolas cannot form real space curves, because the three control points always build a plane. This leads us to the next class of curves, the cubic curves.
Here we have four control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, , \mathbf{p}_3$ and let $t \in \Re$. We compute a curve point with the following construction:

$$\begin{aligned}
\mathbf{p}_0^1(t) &= (1-t)\mathbf{p}_0 + t\mathbf{p}_1 \\
\mathbf{p}_1^1(t) &= (1-t)\mathbf{p}_1 + t\mathbf{p}_2 \\
\mathbf{p}_2^1(t) &= (1-t)\mathbf{p}_2 + t\mathbf{p}_3 \\
\mathbf{p}_0^2(t) &= (1-t)\mathbf{p}_0^1(t) + t\mathbf{p}_1^1(t) \\
\mathbf{p}_1^2(t) &= (1-t)\mathbf{p}_1^1(t) + t\mathbf{p}_2^1(t) \\
\mathbf{p}_0^3(t) &= (1-t)\mathbf{p}_0^2(t) + t\mathbf{p}_1^2(t).
\end{aligned}$$

Inserting the first three equations into the next two, we obtain:

$$\begin{aligned}
\mathbf{p}_0^2(t) &= (1-t)^2\mathbf{p}_0 + 2(1-t)t\mathbf{p}_1 + t^2\mathbf{p}_2 \\
\mathbf{p}_1^2(t) &= (1-t)^2\mathbf{p}_1 + 2(1-t)t\mathbf{p}_2 + t^2\mathbf{p}_3.
\end{aligned}$$

Again, we insert these two equations into the last one, and get this:

$$\mathbf{p}_0^3(t) = (1-t)^3\mathbf{p}_0 + 2(1-t)^2t\mathbf{p}_1 + (1-t)t^2\mathbf{p}_2 + (1-t)^2t\mathbf{p}_1 + 2(1-t)t^2\mathbf{p}_2 + t^3\mathbf{p}_3.$$

After some simplifications, we obtain this result:

$$\mathbf{p}_0^3(t) = (1-t)^3\mathbf{p}_0 + 3(1-t)^2t\mathbf{p}_1 + 3(1-t)t^2\mathbf{p}_2 + t^3\mathbf{p}_3. \tag{1.7}$$

Now, $\mathbf{p}_0^3$ is our point on the curve at parameter value t and we see that the construction is principally the same as in the quadratic case. In Figure 3 you see the geometric construction for $t = 1/2$:

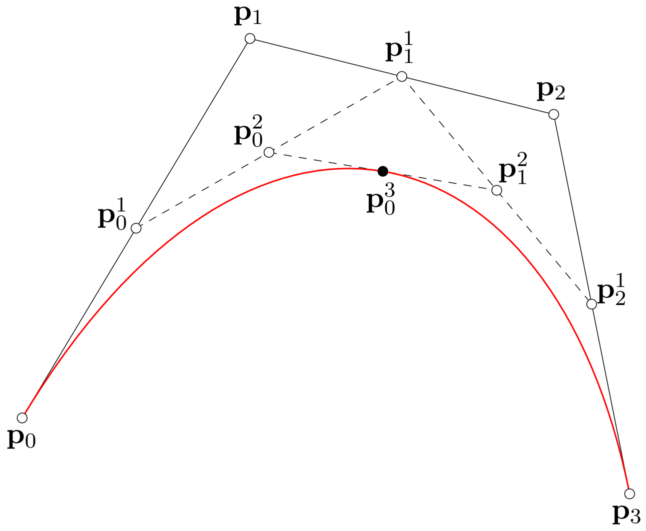

Figure 3: *Repeated Line interpolation on a cubic*

From equation (7) we see, that this is a cubic expression in t, so the obtained curve is a cubic curve. This is the first curve form that can build space curves, because four control points can live in space and not only in a plane.
This curve is also affine invariant and we need 6 linear interpolations, to compute a point $\mathbf{p}_0^3$ on the curve. If we look at the curve form, we see that for $t = 0$ the curve is tangent to the

line $\mathbf{p}_0\mathbf{p}_1$ and for $t = 1$ it is tangent to the line $\mathbf{p}_1\mathbf{p}_2$. We already mentioned this fact for the parabola. Cubic curve are always inside the convex hull of the four control points.

Parametric cubic curves have much more form variations than parabolas, they can have inflection points, nodes (points of self intersection) or even a cusp (point, in which the curve has two tangents). Therefore these cubic curves are used as the major curve forms in Postscript, PDF or in vector drawing and CAD programs.

1.4 Rational Quadratic Bezier Curves

As last curve form I want to introduce rational quadratic bezier curves. They are very important, because they can exactly produce conic curves like parabolas, hyperbolas, ellipses and circles. These curves are a generalisation of their so called integral counterparts, because they include them but they have even more form variations than the integral bezier curves.

Generally spoken, rational curves lie in another space as integral curves. If you draw a circle on a piece of paper and rotate the paper in front of your eyes, you'll see an ellipse. It's a so called projection of the circle. If you repeat this experiment with a parabola, you might see a hyperbola. If we look at this subject backwards, we can state that every rational quadratic bezier curve in 3D-space can be seen as a projection of an integral quadratic bezier curve in the plane. This is fundamental for the understanding of rational bezier curves.

We can deal with rational curves just the way we did with integral curves, but we have to put them first in a so called homogeneous space. This has one more coordinate, the weight of a point. In the rational case, every bezier point has an x-value, an y-value and a weight w. In normal space, each point of a bezier curve has the weight $w = 1$. This weight can be interpreted as the z-coordinate of a point.

Here is the mathematical description of a rational quadratic bezier curve:

$$\mathbf{p}(t) \quad = \quad \frac{(1-t)^2 w_0 \mathbf{p}_0 + 2(1-t)t w_1 \mathbf{p}_1 + t^2 w_2 \mathbf{p}_2}{(1-t)^2 w_0 + 2(1-t)t w_1 + t^2 w_2}. \tag{1.8}$$

As you can see, every point is associated with it's own weight. This is the general form of a rational quadratic bezier curve. With some more involved math it can be translated into a simpler form, which is called the *standard form* and it looks like this:

$$\mathbf{p}(t) \quad = \quad \frac{(1-t)^2 \mathbf{p}_0 + 2(1-t)t w \mathbf{p}_1 + t^2 \mathbf{p}_2}{(1-t)^2 + 2(1-t)t w + t^2}. \tag{1.9}$$

The outer weights w_0 and w_2 are gone (their value is 1) and the central inner weight is transformed to w. The term *rational* simply reflects the fact, that every rational bezier curve is build from a rational expression, which is much more complicated then the expression for integral curves.

But now comes the magic of homogeneous coordinates. If we do the following transformation at the very beginning of our curve calculation:

$$x \rightarrow x \cdot w \quad y \rightarrow y \cdot w \quad z \rightarrow w, \tag{1.10}$$

we can compute the curve points without any fractional math in the same way as we did in the integral case. But before we draw the point, we have to transform it back to the so called affine space (here: our normal 2D-space) with the following caculations:

$$x \rightarrow x/w \quad y \rightarrow y/w \quad z \rightarrow 1, \tag{1.11}$$

I'm not going to explain the mathematical background, but I want to mention that this is the reason, why homogeneous coordinates are used in so many advanced graphic algorithms. If we go back to our paper example, we can say that changing from affine space to homogeneous space is like the projective view of the original shape and changing it back gives us the original. In the style file you can look up the `Rcurve` macro to see how this is implemented in TEX. *Figure 4* shows several conic curves, which all share the same bezier points, only the inner weight w changes.

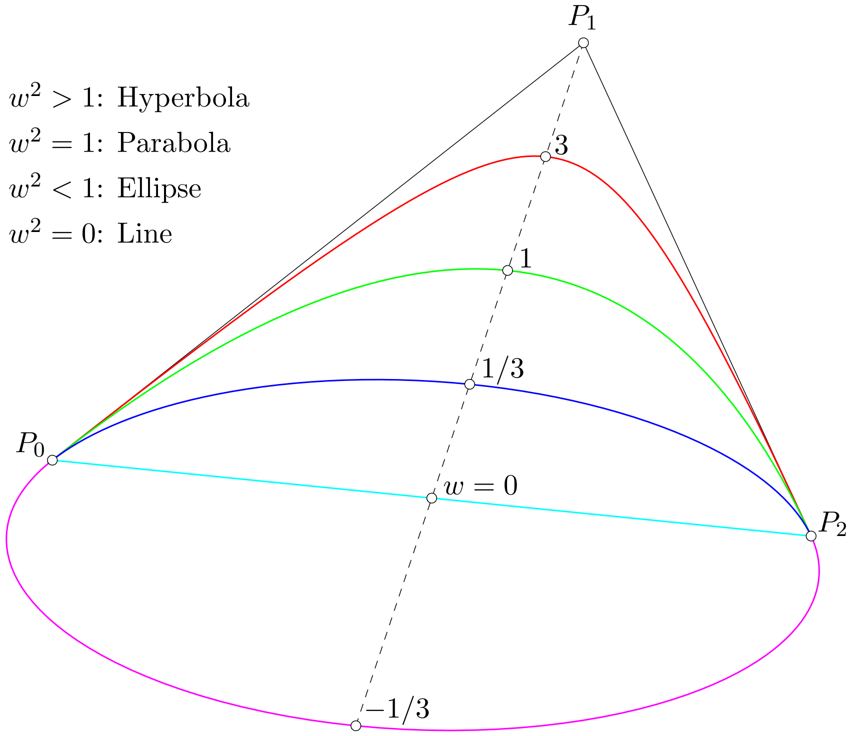

***Figure 4**: Various conic arcs defined by $w = (3, 1, 1/3, 0, -1/3)$.*

To the left of the picture is the curve form classification corresponding to different values of w. You may ask now: "*Where is the circle?*". Well, that's no real problem. Without any proof here follows the answer. Let's first name the angle, formed by the bezier polygon. We call it a. As we saw before, the circle is simply a special case of an ellipse. The ellipse is actually a circle, if the following condition holds:

$$w = \cos(a). \tag{1.12}$$

Here is an example of a circle, build with two rational quadratic bezier curves:

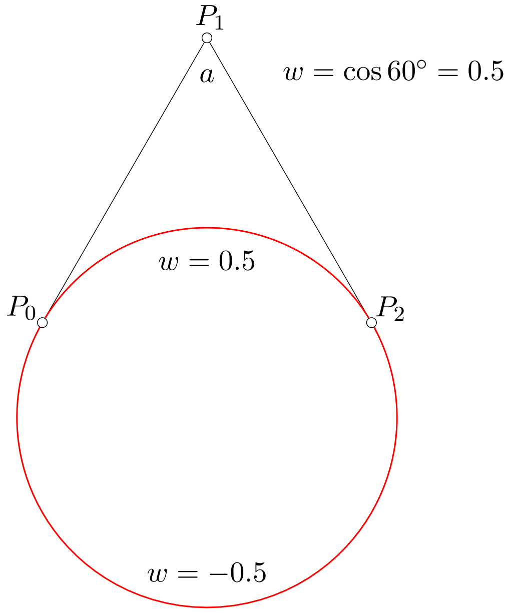

***Figure 5**: A full circle with two rational quadratic bezier curves.*

Both curves share the same bezier points, but the lower one (the complementary curve) has negative weight $-w$. Everytime you want to draw the complementary rational curve, you only have to negate the weight value.

Now it's up to you. I hope, this short introduction into the world of bezier curves has risen your appetite. There are many good books around, which deal with this subject. Look at the end of this paper for some suggestions.

1.5 Additional Readings

Depending on the levels of insight you want to achieve, there are lots of good books and free literature on the market and on the internet. I want to give you some suggestions for literature, that will help you to learn more about graphics programming.

Bibliography

[1] MORTENSON M.E.: *Mathematics for Computer Graphics Applications*, Industrial Press, Inc., 2 ed. 1999.

[2] FOLEY, VAN DAM.: *Computer Graphics – Principles and Practice*, Addison-Wesley, 2 ed. 1996.

[3] PAETH, ALAN W.: *Graphics GEMS I-V*, AP Professional, 1995.

[4] ABRASH, MICHAEL.: *Graphics Programming Black Book*, Coriolis Group Books, 1997.

[5] FARIN G.: *Curves and Surfaces For Computer Aided Geometric Design – A Practical Guide*, Academic Press, 2 ed. 1999.

[6] FARIN G.: *NURBS – from Projective Geometry to Practical Use*, A K Peters Ltd., Natick, MA, 2 ed. 1990.

[7] PIEGL L. & TILLER W.: *The NURBS Book*, Springer, 2 ed. 1997.