

The vast majority of geographic information uses only linear geometries (ie line segments, polygons and polyhedra). When curved geometries are present, they are usually simple parametric shapes, such as spheres, cylinders and cones in CSG or circular arcs in certain 2D datasets (Figure 1.1). Moreover, most data models are designed with linear geometries in mind.

However, modelling curves and curved surfaces is still highly desirable in certain circumstances, as they make it possible to model many shapes a lot more compactly and without losing precision through discretisation. Most CAD and 3D modelling software thus support curves and curved surfaces, and BIM models routinely use them internally as well.

There are several methods that can be used to represent general curves in 2D/3D and curved surfaces in 3D. This lesson covers one of them that is relatively simple and works well in practice: Bézier curves and surfaces.

- 1.1 Background . . . . . 1
- 1.2 Bézier curves . . . . . 5
- 1.3 Bézier surfaces . . . . . 9
- 1.4 Exercises . . . . . 13
- 1.5 Notes and comments . . . . . 13

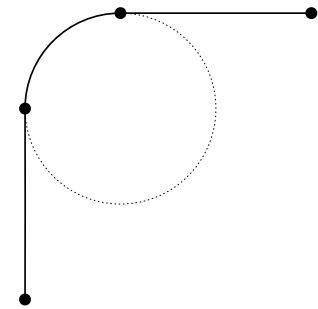


Figure 1.1: A composite curve made from two line segments and a circular arc.

## 1.1 Background

### 1.1.1 Types of points

In general, curve and surface modelling is done by specifying the locations of points, of which there are two kinds (Figure 1.2a):

**data points** are points that the curve/surface needs to pass through; and

**control points** are points that have some influence over the shape of the curve/surface, but through which the curve/surface does not necessarily pass. Intuitively, they ‘pull’ the curve in their direction.

data point

control point

Note that in some contexts, they might all be referred to as control points (eg graphics software), whereas in others (eg interpolation) the distinction is almost always made. If you have some experience with vector-based

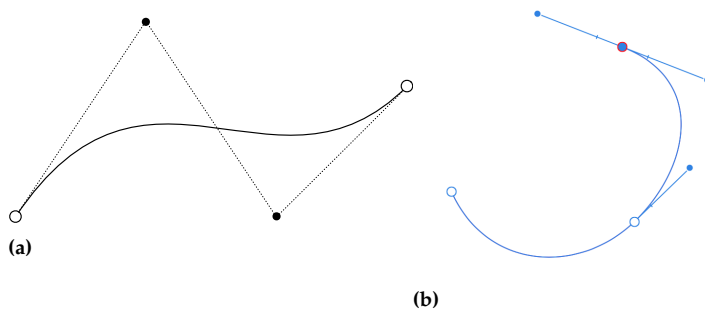


Figure 1.2: (a) The data points (white) and control points (black) used to draw a Bézier curve. (b) In Affinity Designer (shown here) and most other graphics editors, data points (large circles) are surrounded by handles with the control points at their ends (small blue circles)

graphic editors (eg Adobe Illustrator, Inkscape or Sketch), you have likely drawn curves using data points and control points (Figure 1.2b).

### 1.1.2 Types of curves and surfaces

There are three kinds of mathematical representations that are typically used to represent curves and surfaces. From most restrictive to least restrictive, these are:

explicit curve  
explicit surface

**Explicit curves/surfaces** are modelled using a function that defines the value of one coordinate, generally  $y$  in 2D and  $z$  in 3D, based on the other coordinate(s). For instance,  $f(x) = y = x^2$  can be used to define a parabola in 2D, and  $f(x, y) = z = x^2 + y^2$  to define a paraboloid in 3D. This makes it impossible to represent vertical lines in 2D and planes in 3D (without swapping the dependent and independent variables in the equation), and makes it difficult to have multiple values per dependent variable, with minor exceptions such as the use of plus or minus ( $\pm$ ), eg  $f(x) = y = \pm\sqrt{1 - x^2}$  for a circle or  $f(x, y) = z = \pm\sqrt{1 - x^2 - y^2}$  for a sphere.

implicit curve  
implicit surface

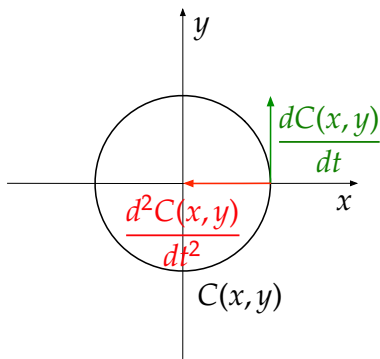
**Implicit curves/surfaces** are modelled using a single function with all coordinates as parameters. For instance,  $f(x, y) = x^2 + y^2 = 1$  defines a unit circle in 2D and  $f(x, y, z) = x^2 + y^2 + z^2 = 1$  defines a unit sphere in 3D. This works fine with vertical lines/planes and can represent multiple values per dependent variable, but completely functions have to be built to represent different curves.

parametric curve  
parametric surface

**Parametric curves/surfaces** are modelled using different functions per coordinate which have independent non-coordinate variables as parameters. For instance,  $f(t) = (\cos t, \sin t)$  with  $0 \leq t \leq 2\pi$  defines a unit circle in 2D, whereas  $f(t) = (\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi)$  with  $0 \leq \theta \leq 2\pi$  and  $0 \leq \phi \leq \pi$  defines a unit sphere in 3D. This is the most flexible approach because it allows us to define curves and surfaces in a general form that works independently of the coordinates used.

Because of their flexibility, Bézier curves/surfaces and most other curve modelling methods (eg splines) are based on parametric curves/surfaces. For the rest of this lesson, we will therefore be working with parametric curves/surfaces only.

tangent vector



**Figure 1.3:** The parametric curve  $C(x, y) = (\cos t, \sin t)$  for the unit circle (black), its first derivative at  $t = 0$  (green), and the second derivative (red).

### 1.1.3 Tangent vectors and other derivatives

If we compute the first derivative of a parametric curve with respect to its parameter(s) at a given point, we get vector(s) with a direction that is tangent to the curve and a magnitude that tells us the rate of change of the parameter(s) at that point. For instance, in the curve  $C(x, y) = (\cos t, \sin t)$  with  $0 \leq t \leq 2\pi$  (Figure 1.3), which describes the unit circle, at  $t = 0$  we get the point  $(1, 0)$ , ie the rightmost point on the circle using the typical axis directions. The tangent vector is then  $dC(x, y)/dt = (-\sin t, \cos t)$ , which at  $t = 0$  is  $(0, 1)$ , ie a unit vector pointing upwards (which makes sense considering that it draws the circle in a counter-clockwise direction).

A common analogy to understand this concept is to consider a moving particle moving in time (hence you will often see  $t$  used in parametric functions). The parametric function tells us the position of the particle at any given time, and the tangent vector tells us the direction and speed of the particle at that time.

The second derivative of a parametric curve is a little harder to visualise, but it is also a vector that tells us the rate of change of the curvature at a point. In the particle analogy, it is the acceleration of the particle (and its acceleration direction). In the circle from the previous example, the second derivative is  $d^2C(x, y)/dt^2 = (-\cos t, -\sin t)$ , which at  $t = 0$  is  $(-1, 0)$ , ie a unit vector towards the centre of the circle.

### 1.1.4 Uniform vs. non-uniform

In order to fit a parametric curve through a series of points, we need to decide the curve equations and the values of the parameters that should be used. We will discuss the curve equations for Bézier curves and surfaces later in this chapter, so for the moment let us consider only two points  $p_0$  and  $p_1$ , which are connected by a straight line segment  $L$ . The parametric equation of the line segment would be given by:

$$L(t) = (1 - t)p_0 + tp_1, \text{ for } 0 \leq t \leq 1. \quad (1.1)$$

Here, note that  $t = 0$  corresponds to  $p_0$  and  $t = 1$  corresponds to  $p_1$ . When the equation is parametrised so that the parameter increases by a fixed amount for every point in a sequence of points, it is said to be a *uniform* parametric curve. When this is not the case, it is a *non-uniform* parametric curve.

uniform parametric curve  
non-uniform parametric curve

### 1.1.5 Polynomials, segments and patches

Polynomial functions can be used to directly model entire curves and surfaces. In 2D, a polynomial of degree one (ie a straight line) is a linear function of the form  $f(t) = at + b$  that can be defined so as to pass through two points, a polynomial of degree two (ie a parabola) is a quadratic function of the form  $f(t) = at^2 + bt + c$  that can be made to pass through three points, a polynomial of degree three is a cubic function of the form  $f(t) = at^3 + bt^2 + ct$  that can be made to pass through four points, and so on. We can therefore use a polynomial of degree  $n$  to model a curve passing through  $n + 1$  points.

However, high-degree polynomials wobble uncontrollably to pass through all the points, and a small change in the position of one of the points can cause large changes all over the curve. It is thus much better to split curves and surfaces into *segments* (for curves) and *patches* (for surfaces) passing through only a small number of points, and then to join these segments/patches using low-degree polynomial functions (generally quadratic or cubic).

curve segment  
surface patch

In a parametric curve  $C(t)$ , specific values of  $t$  can be used to split the curve into segments. Most commonly, the data points will be used for this

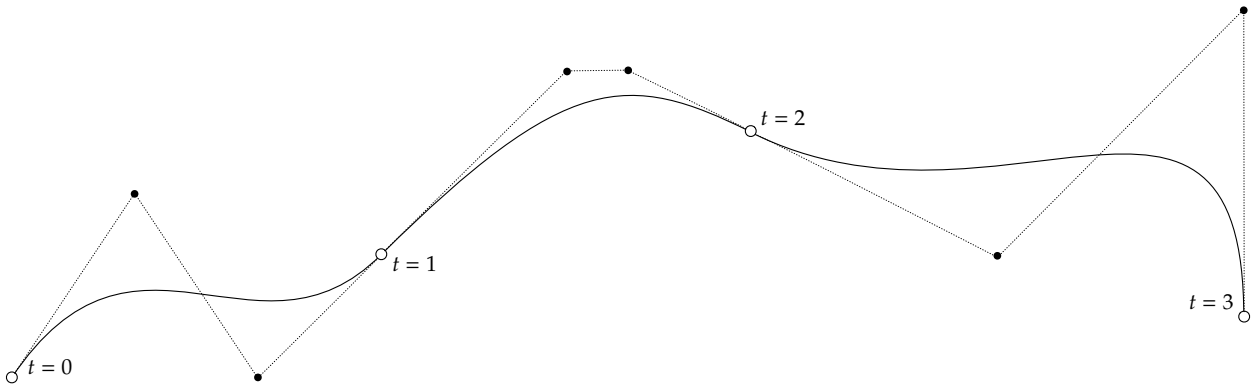


Figure 1.4: A composite Bézier curve made from three segments.

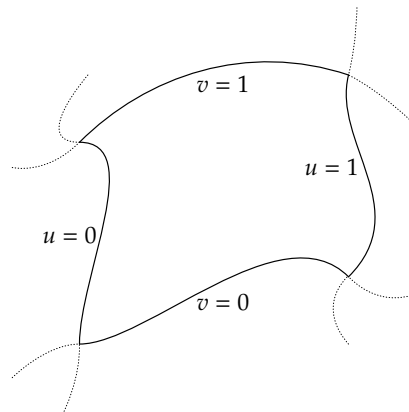


Figure 1.5: A Bézier rectangular patch.

purpose, which for uniform parametric curves will be at  $t = 0, 1, 2, \dots$  (Figure 1.4).

In a parametric surface  $S(u, v)$ , specific values of  $u$  are curves on the surface, as are values of  $v$ . Similarly, a set of two curves with fixed  $u$  and two curves with fixed  $v$  will bound a patch, eg  $S(u, 0)$ ,  $S(u, 1)$ ,  $S(0, v)$  and  $S(1, v)$  (Figure 1.5).

### 1.1.6 Continuity

In order to describe how segments/patches should be joined, we rely on the concept of continuity, of which there are two types: geometric and parametric. Geometric continuity can be defined as follows:

geometric continuity

positional continuity

$G^0$  continuity

tangential continuity

$G^1$  continuity

curvature continuity

$G^2$  continuity

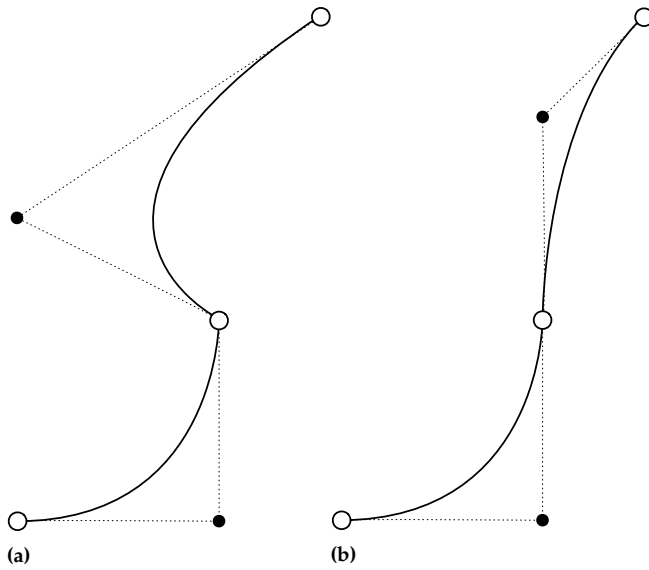
$G^n$  continuity

**Positional** ( $G^0$ ) continuity means that the boundary of a segment or patch matches that of its neighbours, ie there are no holes at common boundaries (Figure 1.6a);

**Tangential** ( $G^1$ ) means that the angles of segments or patches match those of its neighbours at their common boundaries, ie no sharp edges at common boundaries (Figure 1.6b);

**Curvature** ( $G^2$ ) means that the curvature of a segment or patch match that of its neighbours at their common boundaries, ie no 'soft' edges at common boundaries.

Generalising from here, we can say that a curve has  $G^n$  continuity at a boundary point when the  $n$ -th derivatives have the same direction at



**Figure 1.6:** Continuity: (a)  $G_0$  and (b)  $G_1$ .  $G_2$  continuity is difficult to achieve with Bézier curves.

that point. If they have the same magnitude as well, it is also said to have  $C^n$  continuity. Therefore,  $C^n$  continuity implies  $G^n$  continuity.

$C^n$  continuity

## 1.2 Bézier curves

Bézier curves are parametric curves that are based on a polynomial function with one parameter. They are named after Pierre Bézier, who developed them to model the stylised shapes of cars while working at Renault in the 1960s. Interestingly enough, they were also independently developed by Paul de Casteljau at Citroën, likely before Bézier, but it appears that he was not allowed to publish them. However, De Casteljau’s algorithm, which is used to evaluate Bézier curves, is named after him.

Bézier curve

### 1.2.1 A single Bézier segment

Given a sequence of points, the Bézier curve starts from the first point and ends at the last point, whereas the intermediate points are treated as control points that ‘pull’ the curve in their direction, but always remaining inside the convex hull of the points. This is known as a *Bézier segment*.

Bézier segment

The tangent vector at the first point points to the second point, whereas the tangent vector at the last point points from the next-to-last point to it. Similar constructions can be made for the higher derivatives, with the  $n$ -th derivative being determined only by  $n + 1$  points.

linear Bézier curve

If there are no intermediate points, the result is a linear Bézier curve, which is equivalent to a straight line between the two endpoints. The most common forms of Bézier curves are however quadratic Bézier curves (Figure 1.7), which have one intermediate point, eg  $p_{\text{quadratic}} = (p_0, p_1, p_2)$ , and cubic Bézier curves (Figure 1.8), which have two intermediate points, eg  $p_{\text{cubic}} = (p_0, p_1, p_2, p_3)$ . Note the tangent vectors in both figures, as well as how the curves always fit within the convex hull of the points.

quadratic Bézier curve

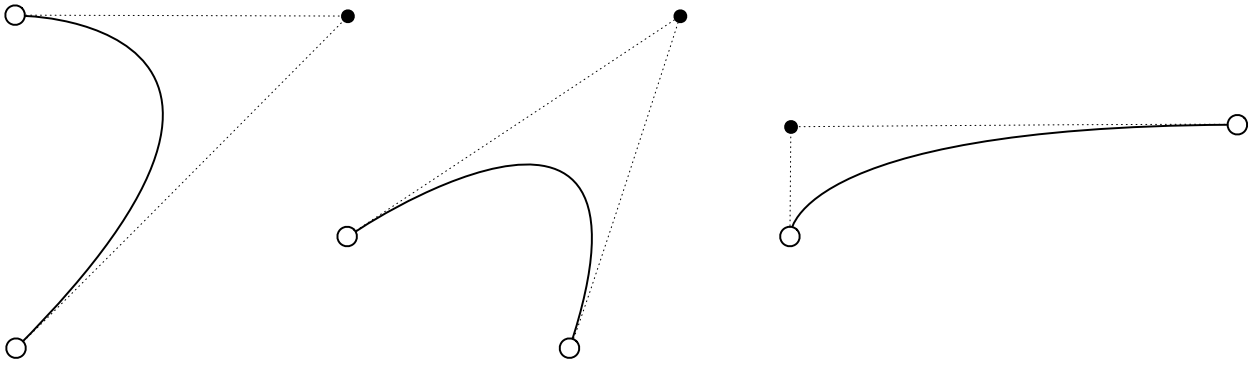


Figure 1.7: Three quadratic Bézier curves

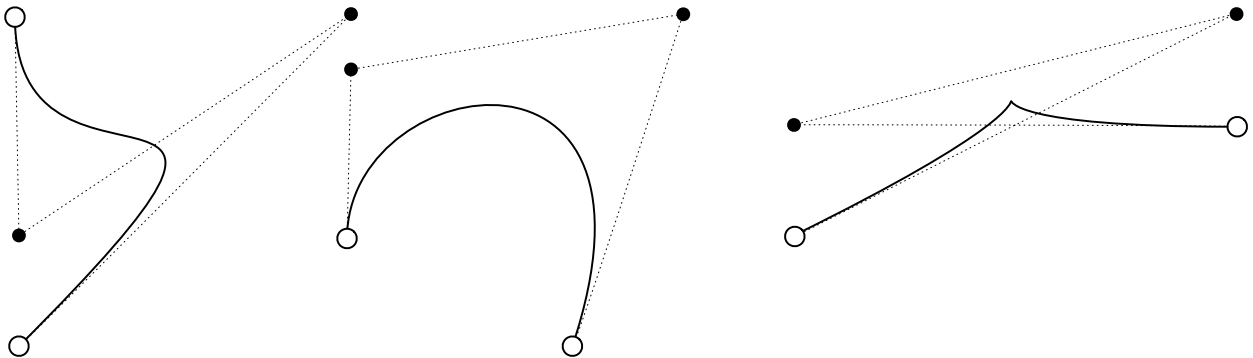


Figure 1.8: Three cubic Bézier curves

A Bézier curve  $C$  can be formulated as a sort of weighted average of its points  $(p_0, \dots, p_n)$ :

$$C(t) = \sum_{i=0}^n B_i^n(t)p_i, \text{ for } 0 \leq t \leq 1 \tag{1.2}$$

where  $B_i^n$  is the weight associated with the point  $p_i$ .

Intuitively, you can imagine that we want this weight to reach a maximum when we are close to the point and decrease as we move farther from it. For example, in a quadratic Bézier curve,  $B_0^n$  should start from its maximum value at  $t = 0$  and decrease as  $t$  increases,  $B_1^n$  should start low, increase to reach a maximum at  $t = 0.5$  and decrease afterwards, and  $B_2^n$  should start from its lowest point and increase to reach its maximum at  $t = 1$ .

Bernstein polynomials

The exact functions used to determine the weights in Bézier curves are called *Bernstein polynomials* (Figure 1.9), named after Sergei Natanovich Bernstein who discovered them in the 1910s. These are given by:

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \text{ where } \binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{1.3}$$

binomial coefficient  
Pascal's triangle

where  $n$  is the degree of the Bézier curve, ie 1 for linear, 2 for quadratic, 3 for cubic, etc.  $\binom{n}{i}$  is the *binomial coefficient*, which is equivalent to the  $i$ -th column of the  $n$ -th row in Pascal's triangle (Figure 1.10).

Linear Bézier curves ( $n = 1$ ) are thus given by:

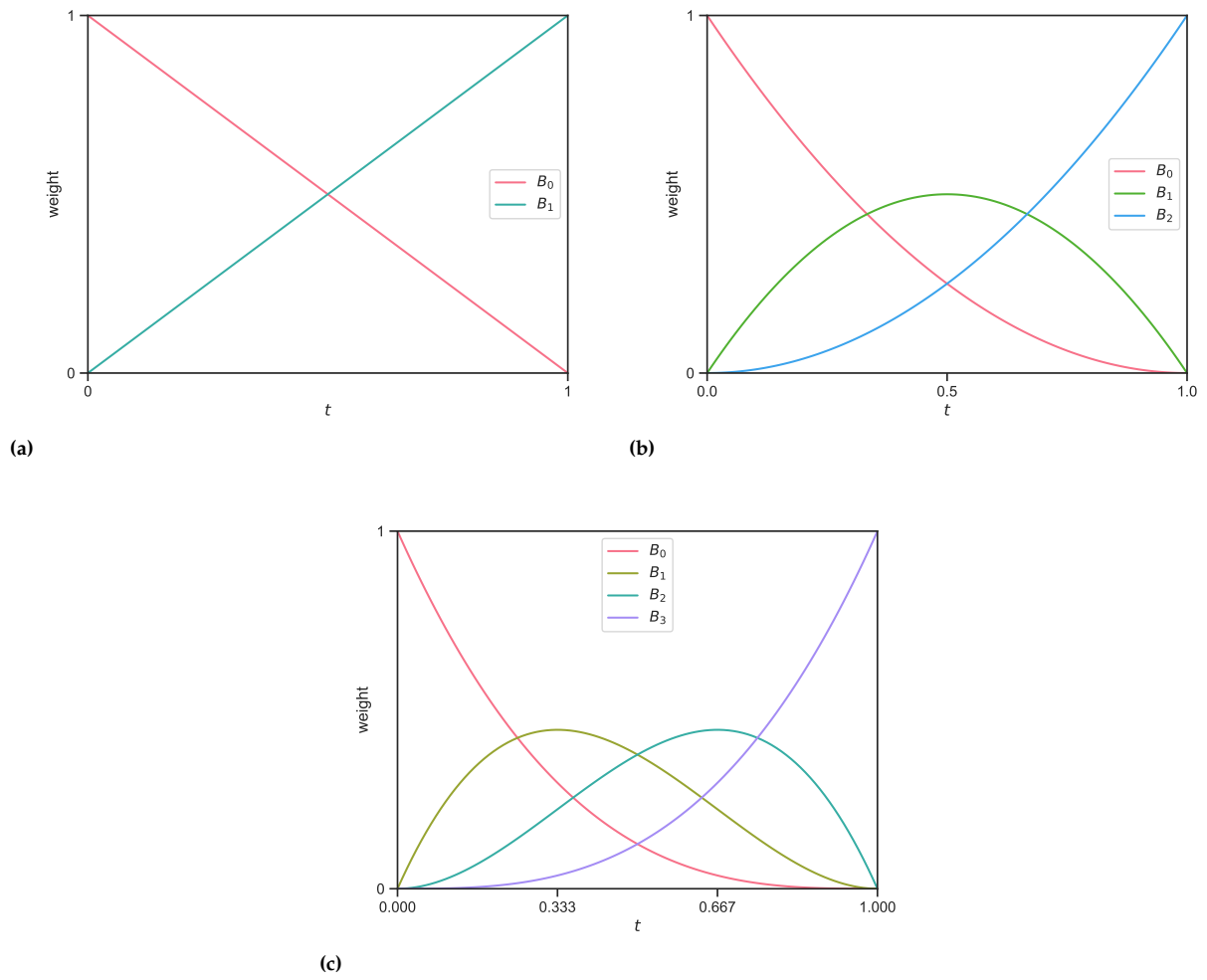


Figure 1.9: Weights obtained from the Bernstein polynomials for (a) linear, (b) quadratic and (c) cubic Bézier curves.

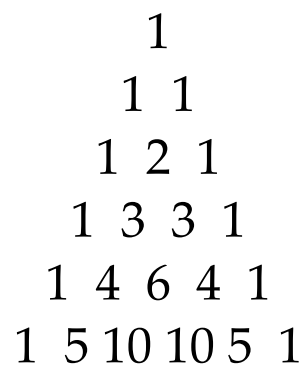


Figure 1.10: Pascal's triangle, where the numbers are obtained by adding the numbers in the row above (starting from a single 1).

$$\begin{aligned}
C(t) &= \sum_{i=0}^1 B_i^1(t)p_i \\
&= (1-t)p_0 + tp_1, \text{ for } 0 \leq t \leq 1.
\end{aligned} \tag{1.4}$$

which is equivalent to the parametric equation of the line we discussed in the background. Quadratic Bézier curves ( $n = 2$ ) are given by:

$$\begin{aligned}
C(t) &= \sum_{i=0}^2 B_i^2(t)p_i \\
&= (1-t)^2p_0 + 2t(1-t)p_1 + t^2p_2, \text{ for } 0 \leq t \leq 1.
\end{aligned} \tag{1.5}$$

And cubic Bézier curves ( $n = 3$ ) are given by:

$$\begin{aligned}
C(t) &= \sum_{i=0}^3 B_i^3(t)p_i \\
&= (1-t)^3p_0 + 3t(1-t)^2p_1 + 3t^2(1-t)p_2 + t^3p_3, \text{ for } 0 \leq t \leq 1.
\end{aligned} \tag{1.6}$$

## 1.2.2 Composite Bézier curves

Now, let us discuss how to join multiple Bézier segments together smoothly into a *composite Bézier curve* or a *polybezier* (accent usually omitted). When they are joined in a loop, ie joining the last to the first, it is sometimes called a *beziergon* or *bezigon*. Many common vector file formats use these, including several font formats, PDF files, and SVG images. Figure 1.2 also shows how these commonly look in software, where the intermediate control points are shown as ‘handles’ around the data points.

As we mentioned before, high-degree polynomials are undesirable because they tend to wobble and are hard to control. It is therefore usually better to create composite Bézier curves by connecting Bézier segments made from low-degree Bézier curves using only a few control points, most often cubics.

Connecting multiple Bézier segments with  $G_0$  or  $C_0$  continuity simply means that their common endpoint should be the same. That is, if we have a composite Bézier curve formed by two adjacent Bézier segments, where the first is defined by the points  $(p_0, \dots, p_n)$  and the second is defined by the points  $(q_0, \dots, q_n)$ , we need to enforce that  $p_n = q_0$ .

As we previously discussed, the tangent vector of the endpoint of a Bézier curve is related only to the endpoint and its neighbour. Therefore,  $G_1$  continuity can be achieved by making sure that the common endpoint and its two neighbours, ie  $p_{n-1}$  and  $q_1$ , are collinear (Figure 1.11). For  $C_1$  continuity, they should also be evenly spaced.  $G_2$  and  $C_2$  continuity is hard to achieve, so we will not discuss it here.

composite Bézier curve  
polybezier  
beziergon  
bezigon



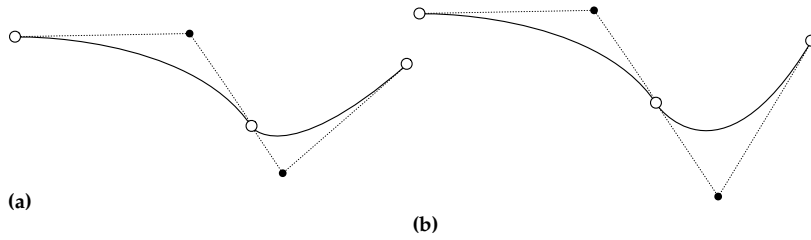


Figure 1.11: Two composite Bézier curves with: (a)  $G_1$  continuity and (b)  $C_1$  continuity (bottom).

### 1.3 Bézier surfaces

#### 1.3.1 Rectangular Bézier surfaces

Moving on to 3D, the most common implementation of Bézier surfaces uses rectangular patches, which are made of grids of points. The most common are biquadratic ( $3 \times 3$ ; Figure 1.12) and bicubic ( $4 \times 4$ ; Figure 1.13) surfaces, which are defined based on square matrices of points, such as:

biquadratic Bézier surface  
bicubic Bézier surface

$$p_{\text{biquadratic}} = \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} \\ p_{1,0} & p_{1,1} & p_{1,2} \\ p_{2,0} & p_{2,1} & p_{2,2} \end{pmatrix}, \text{ and} \tag{1.7}$$

$$p_{\text{bicubic}} = \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,0} & p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}, \tag{1.8}$$

where only the four corner points are data points and all the others are control points. Note that the four sides of a Bézier surface are Bézier curves using the points on the top/bottom/left/right of the matrix.

A rectangular Bézier surface is described as:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) p_{i,j}, \text{ for } 0 \leq u \leq 1, 0 \leq v \leq 1. \tag{1.9}$$

where  $p_{i,j}$  is a point in an  $m \times n$  matrix that defines the data points and control points used for the patch.

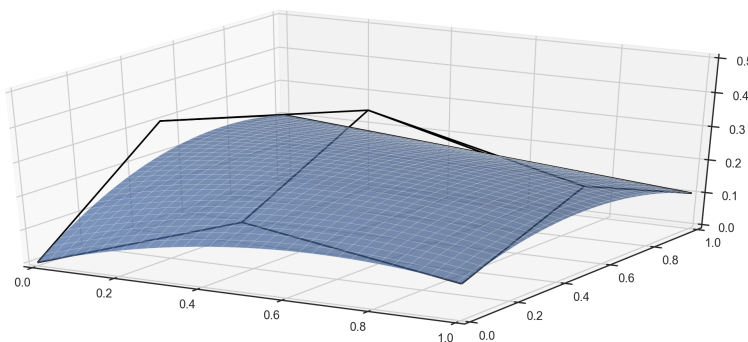
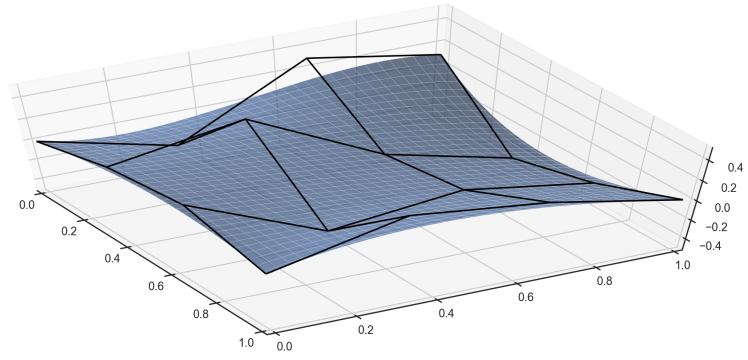


Figure 1.12: A Bézier biquadratic surface and the points that define it. Note how the four corners are the only data points and how the surface is tangent to them.



**Figure 1.13:** A Bézier bicubic surface and the points that define it. Note how the four corners are the only data points and how the surface is tangent to them.

composite Bézier surface  
Bézier patch

As with composite Bézier curves, *composite Bézier surfaces* can be created by joining together multiple rectangular Bézier patches. These follow the same logic as the composite Bézier curves.

In order to get  $G_0$  and  $C_0$  continuity, the common points at the boundary of the two matrices should be the same. That is, if we have a composite Bézier surface formed by two adjacent Bézier rectangular patches, where the first is defined by the matrix  $p$  and the second by the matrix  $q$ , which are defined as:

$$p = \begin{pmatrix} p_{0,0} & \cdots & p_{0,n} \\ \vdots & \ddots & \vdots \\ p_{m,0} & \cdots & p_{m,n} \end{pmatrix}, q = \begin{pmatrix} q_{0,0} & \cdots & q_{0,n} \\ \vdots & \ddots & \vdots \\ q_{m,0} & \cdots & q_{m,n} \end{pmatrix}, \quad (1.10)$$

and they are joined at the curve defined by  $p_{i,n}$  and  $q_{i,0}$ , for  $0 \leq i \leq n$ , we simply need to enforce that  $p_{i,n} = q_{i,0}$ .

For  $G_1$  continuity, we need to ensure that the tangent vector at the common curve has the same direction, which is given by:

$$\left. \frac{\partial p(u, v)}{\partial v} \right|_{v=1} = a \left. \frac{\partial q(u, v)}{\partial v} \right|_{v=0}. \quad (1.11)$$

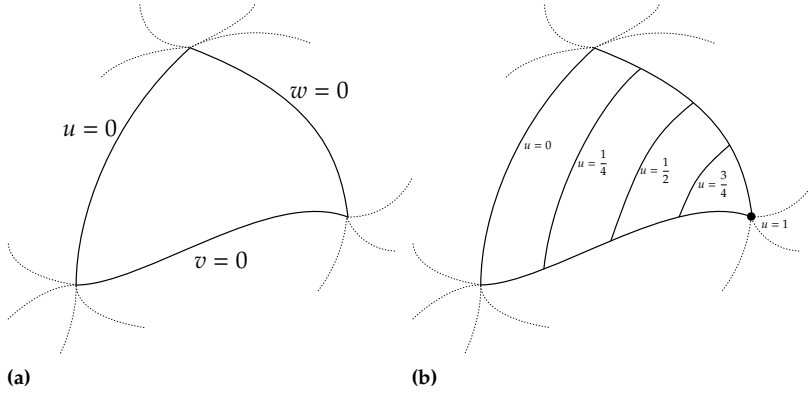
where  $a$  can have any positive value. For  $C_1$  continuity, the magnitude of the vector needs to be the same, which means that  $a = 1$  in the previous equation. Just as with composite Bézier curves, these conditions are achieved when each point along the common boundary curve and its neighbours on either side patch. That is, for all  $i$ ,  $p_{i,n-1}$ ,  $q_{i,0}$  and  $q_{i,1}$ , are collinear (for  $G_1$ ) and also evenly spaced (for  $C_1$ ).

### 1.3.2 Triangular Bézier surfaces

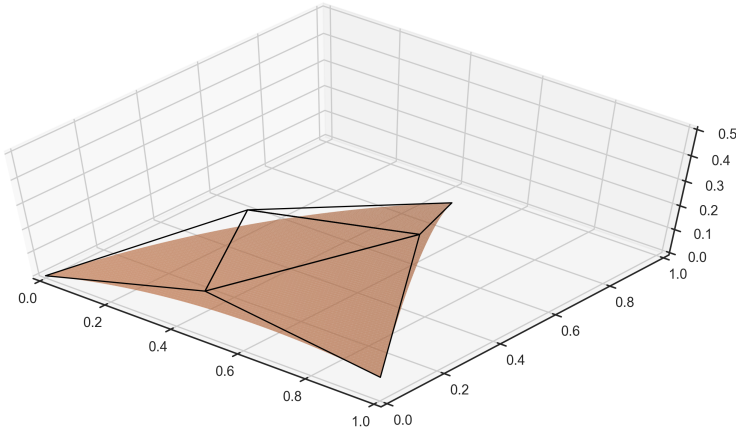
triangular Bézier surface  
Bézier triangle

Triangular Bézier surfaces, or simply Bézier triangles, are the other common type of Bézier surface. These are better parametrised in terms of three barycentric coordinates, which we here denote as  $u$ ,  $v$  and  $w$  (Figure 1.14). Note however that the three coordinates not linearly independent, as they always add up to one.

Triangular Bézier surfaces are defined based on triangular arrangements of points of the form:



**Figure 1.14:** The barycentric coordinates used to parametrise triangular Bézier surfaces



**Figure 1.15:** A Bézier quadratic triangle and the points that define it. Note how the three corners are the only data points and how the triangle is tangent to them.

$$p_{\text{linear}} = \begin{matrix} p_{0,1,0} \\ p_{0,0,1} & p_{1,0,0} \end{matrix}, \quad (1.12)$$

$$p_{\text{quadratic}} = \begin{matrix} p_{0,2,0} \\ p_{0,1,1} & p_{1,1,0} \\ p_{0,0,2} & p_{1,0,1} & p_{2,0,0} \end{matrix}, \quad (1.13)$$

$$p_{\text{cubic}} = \begin{matrix} p_{0,3,0} \\ p_{0,2,1} & p_{1,2,0} \\ p_{0,1,2} & p_{1,1,1} & p_{2,1,0} \\ p_{0,0,3} & p_{1,0,2} & p_{2,0,1} & p_{3,0,0} \end{matrix}, \quad (1.14)$$

where the surface is described by:

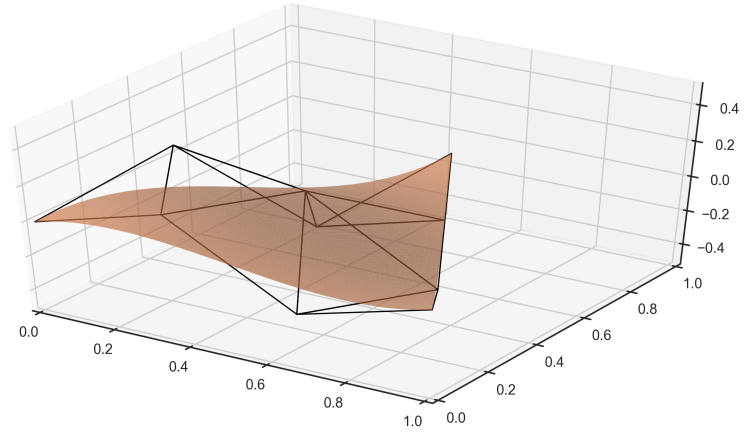
$$S(u, v, w) = \sum_{\substack{i+j+k=n \\ i,j,k \geq 0}} B_{i,j,k}^n(u, v, w) p_{i,j,k}, \quad (1.15)$$

and  $B_{i,j,k}^n$  are the *bivariate Bernstein polynomials*, which are given by:

$$B_{i,j,k}^n(u, v, w) = \frac{n!}{i!j!k!} u^i v^j w^k \quad (1.16)$$

bivariate Bernstein polynomials

For the first few values of  $n$ , these are:



**Figure 1.16:** A Bézier cubic triangle and the points that define it. Note how the three corners are the only data points and how the triangle is tangent to them.

$$B_{i,j,k}^1 = \begin{matrix} v \\ w & u \end{matrix}, \tag{1.17}$$

$$B_{i,j,k}^2 = \begin{matrix} v^2 \\ 2vw & 2uv \\ w^2 & 2uw & u^2 \end{matrix}, \tag{1.18}$$

$$B_{i,j,k}^3 = \begin{matrix} v^3 & & & \\ 3v^2w & 3uv^2 & & \\ 3vw^2 & 6uvw & 3u^2v & \\ w^3 & 3uw^2 & 3u^2w & u^3 \end{matrix} \tag{1.19}$$

As with rectangular Bézier surfaces, the points on any of the three edges of the triangular Bézier surface define a Bézier curve.

In order to create a composite Bézier surface from triangular Bézier patches, we need to have similar constraints as before. For  $G_0$  and  $C_0$  continuity, the common boundary points should be the same. That is, if we have a composite Bézier surface formed by two adjacent Bézier triangular patches, where the first is defined by the matrix  $p$  and the second by the matrix  $q$ , which are defined as:

$$p = \begin{matrix} & & p_{0,n,0} & & \\ & \ddots & & \ddots & \\ p_{n,0,0} & & & & \\ & \ddots & & \ddots & \\ & & p_{0,0,n} & & \end{matrix}, q = \begin{matrix} & & q_{0,n,0} & & \\ & \ddots & & \ddots & \\ q_{n,0,0} & & & & \\ & \ddots & & \ddots & \\ & & q_{0,0,n} & & \end{matrix}, \tag{1.20}$$

and they are joined at the curve defined by  $u = 0$  on both, ie  $p_{0,j,n-j}$  and  $q_{0,j,n-j}$ , for all  $0 \leq j \leq n$ , we need to enforce that  $p_{0,j,n-j} = q_{0,j,n-j}$ .

For  $G_1$  continuity, it is somewhat more complex than for rectangular Bézier surfaces. Basically, we need to ensure that along the common boundary curve, three specific vectors starting at every common point on the boundary surface except for the last one, ie  $p_{0,j,n-j}$ , for all  $0 \leq j < n$ , should be coplanar. These three vectors are the ones pointing to: (i) the next point along the common curve (ie  $p_{0,j,n-j}$  to  $p_{0,j+1,n-j-1}$ ), (ii) the next

point in the  $u$  direction in the patch on the left (ie  $p_{0,j,n-j}$  to  $p_{1,j,n-j-1}$ ), and (iii) the next point in the  $u$  direction in the patch on the right (ie  $q_{0,j,n-j}$  to  $q_{1,j,n-j-1}$ ).

## 1.4 Exercises

1. What are the explicit, implicit and parametric equations for a line? And for a plane? Hint: start from two and three points.
2. Open any graphics editing software that can model curves. What degree of continuity does it enforce?
3. Derive the weights given by Bernstein polynomials for quartic (degree four) and quintic (degree five) Bézier curves. At which values of  $t$  do they reach a maximum?
4. How would you store the data points and control points for:
  - a) a composite Bézier curve with cubic segments?
  - b) a rectangular Bézier patch in a half-edge data structure?
  - c) a triangular Bézier patch in a triangle-based data structure?
5. Cubic Bézier curves and surfaces are the most commonly used ones. Why do you think that is? Hint: think of inflection points

## 1.5 Notes and comments

Salomon (2006) is a nice book covering all aspects of modelling curves and curved surfaces. Some of the equations described in this lesson are adapted (and usually simplified) from this book.

Farin (2004) covers the history of how Bézier curves and other curve modelling approaches were created (with nice historical pictures).

There are nice animations showing a graphical interpretation of Bézier curves in their [Wikipedia article](#), although the rest of the article is not as good.

## Bibliography

Farin, G. (2004). A History of Curves and Surfaces in CAGD. *Handbook of Computer Aided Geometric Design*. Ed. by G. Farin, J. Hoschek, and M.-S. Kim. Elsevier.

Salomon, D. (2006). *Curves and Surfaces for Computer Graphics*. Springer New York.