1

P. Bézier: How a Simple System Was Born

CAD/CAM mathematical problems have generated many solutions, each adapted to specific aspects of development. Most of the systems were invented by mathematicians, but UNISURF, at least at the start, was developed by mechanical engineers from the automotive industry. These engineers were familiar with parts described mainly by lines and circles; fillets and other blending auxiliary surfaces were scantly defined, their final shape being left to the skill and experience of patternmakers and die-setters.

Around 1960, designers of stamped parts such as car-body panels used French curves and sweeps. The final standard, however, was the "master model", whose shape, for many reasons, could not coincide with the curves traced on the drawing board. This inconsistency resulted in discussions, arguments, retouches, expenses and delay.

Obviously, no significant improvement could be expected as long as a method was not devised that provided an accurate, complete and indisputable definition of freeform shapes.

Computing and numerical control (NC) had made great progress at that time, and it was certain that only numbers, transmitted from drawing office to tool drawing office, manufacture, patternshop and inspection, could provide an answer; of course, drawings would remain necessary, but they

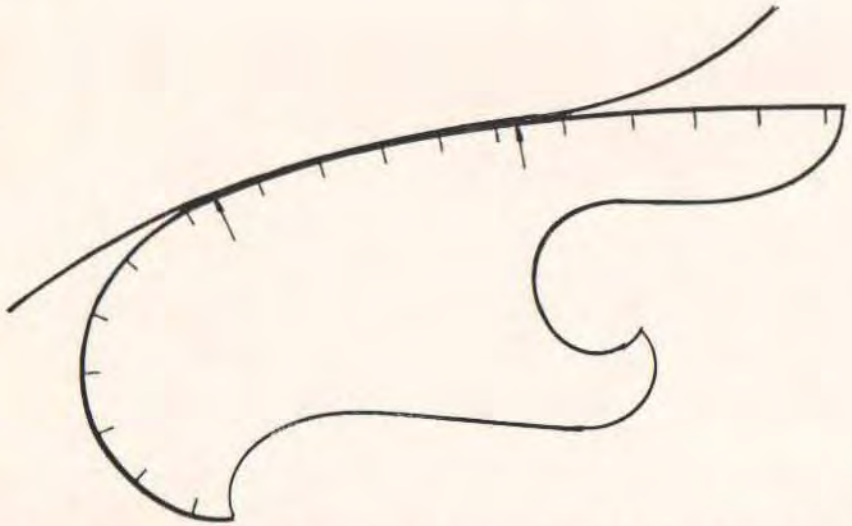

Figure 1.1. An arc of a handdrawn curve is approximated by a part of a template.

would only be explanatory, their accuracy having no importance. Numbers would be the only and final definition.

Certainly, no system could be devised without the help of mathematics—yet designers, in charge of operation, had good knowledge of geometry, especially descriptive geometry, but no basic training in algebra or analysis.

It should be noted that in France very little was known at that time about the work performed in the American aircraft industry. The papers by James Ferguson were not publicized before 1964, Citroen was secretive about the results obtained by Paul de Casteljau, and the famous technical report MAC-TR-41 (by S. A. Coons) did not appear until 1967. The works of W. Gordon and R. Riesenfield were printed in 1974.

The idea of UNISURF was initially oriented towards geometry rather than analysis, but kept in mind that every datum should be expressed exclusively by numbers.

For instance, an arc of a curve could be represented (Fig. 1.1) by the cartesian coordinates of its limit points, A and B, together with their curvilinear abscissae, related with a grid traced on the edge.

The shape of the middle line of a sweep is a cube, granted that its cross section is constant, its matter is homogeneous, and the effect of friction on the tracing cloth is not considered. It is difficult, however, to take into account the length between endpoints. Moreover, the curves employed for

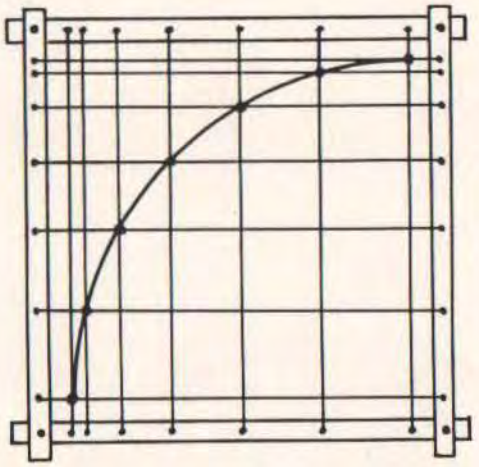

Figure 1.2. A circular arc is obtained by connecting the points in this rectangular grid.

softwares for NC machine tools, i.e., 2D milling machines, were lines and circles and, sometimes, parabolas. Hence, a spline shape would be divided and subdivided into small arcs of circles put end to end.

In order to transform an arc of a circle into a portion of an ellipse, one could imagine (Fig. 1.2) a square frame containing two sets of strings, the intersections of which would be located on an arc of a circle; the frame sides being hinged, the square is transformed into a diamond (Fig. 1.3) and the circle becomes an arc of an ellipse, which would be defined entirely as soon as the coordinates of points A, B, and C were known. If the hinged sides of the frame were replaced by pantographs (Fig. 1.4), the diamond would become a parallelogram, and the definition of the arc of an ellipse would still result from the coordinates of the three points A, B, and C (Fig. 1.5).

This idea was not realistic, but it was easily replaced by the computation of coordinates of successive points on the curve; harmonic functions were available through the use of analog computers, which gave excellent results.

However, employing only arcs of ellipses limited by conjugate diameters was far too restrictive, and a more flexible definition was required.

Another idea came from the practice of a speaker projecting with a handheld torch a small sign onto a screen displaying a figure printed on a slide. Replacing the sign with a curve and recording the exact location and orientation of the torch (Fig. 1.6) would define the image of the curve projected on the wall of the drawing office. One could even imagine having a variety of slides, each of which would bear a specific curve: circle, parabola, astroid, etc.

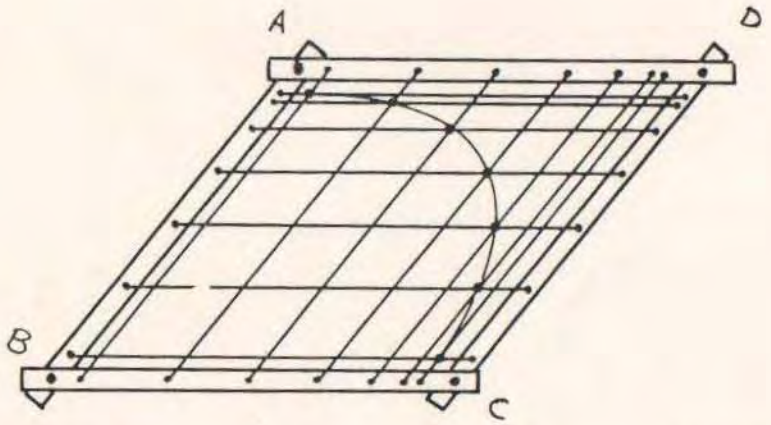

Figure 1.3. If the frame from the previous figure is sheared, an arc of an ellipse is obtained.

This was not a realistic idea, since the focal plane of the zoom would seldom be square to the axis; an optician's nightmare! but the principle could be translated, via projective geometry and matrix computation, into cartesian coordinates.

At that time, designers defined the shape of a car body by cross sections located one hundred millimeters apart, sometimes less. The advantage was that, from a drawing, one could derive templates for adjusting a clay model, a master or a stamping tool; the drawback was that a stylist does not define a shape by cross sections but with "character lines", which are seldom plane curves. Hence, a good system would be capable of manipulating and directly defining "space curves" or "free form curves". Of course, one could imagine working alternately (Fig. 1.7) on two projections of a space curve, but it is unlikely that a stylist would accept such a solution.

Theoretically, at least, a space curve could be expressed by a sweep having a circular section, constrained by springs or counterweights (Fig. 1.8), but this would prove quite impractical.

Would not the best solution be to revert to the basic idea of a frame? Instead of a curve inscribed in a square, it would be located in a cube (Fig. 1.9) that could become any parallelepiped (Fig. 1.10) by a linear transformation easy to compute. The first idea was to choose a basic curve that would be the intersection of two circular cylinders; the parallelepiped

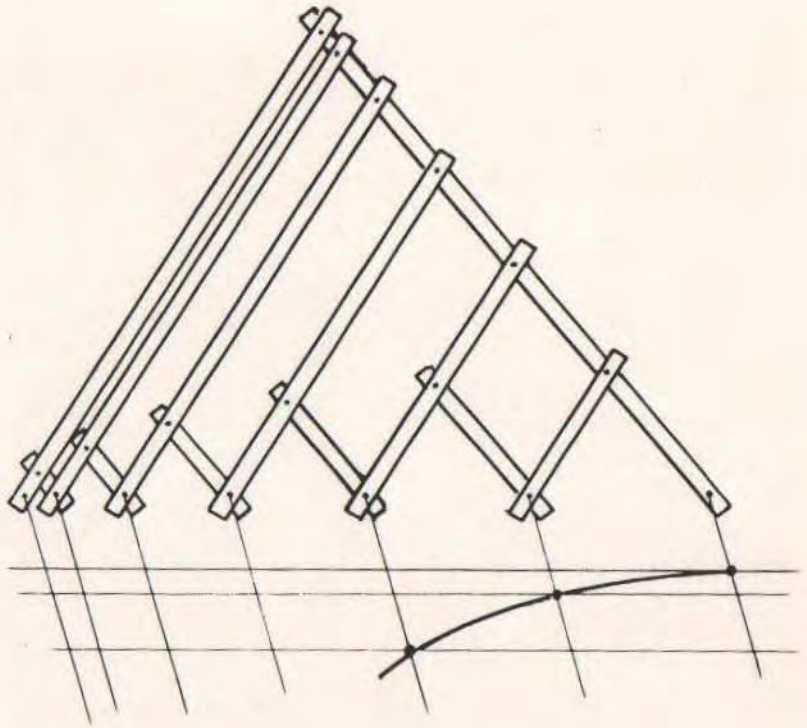

Figure 1.4. Pantograph construction of an arc of an ellipse.

would be defined (Fig. 1.10) by points O, X, Y, and Z. It is more practical, however, to put the basic vectors end to end to obtain a polygon OMNB (Fig. 1.10), which directly defines the end point B and its tangent NB; of course, points O, M, N, and B need not be coplanar and can define a space curve.

Polygons with three legs can define a large variety of curves (see Fig. 3.3 in section 3.3), but in order to increase it, we can make use of cubes and hypercubes of any order (Fig. 1.11) and the relevant polygons (Fig. 1.13).

At that point, it became necessary to do away with harmonic functions and revert to polynomials; this change was even more desirable since digital computers were gradually replacing analog computers. The polynomial functions were chosen according to the properties that were considered most important: tangency, curvature, etc.; later it was discovered that they could be considered as sums of Bernstein's functions.

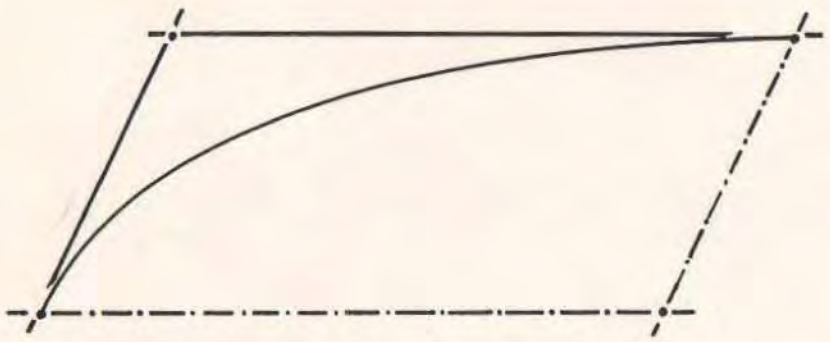

Figure 1.5. A "control polygon" for an arc of an ellipse.

When it was suggested that these curves replace sweeps and French curves, most stylists objected that they had invented their own templates and would not change their methods. It was therefore solemnly promised that their "secret" curves would be translated in secret listings and buried in the most secret part of the computer's memory, and that nobody but them would keep the key of the vaulted cellar. In fact, the standard curves were flexible enough and secret curves were soon forgotten; designers and draughtsmen easily understood the polygons and their relation with the shape of the corresponding curves.

In the traditional process of body engineering, a set of curves is carved in a 3D model and interpolation is left to the experience of highly skilled patternmakers; in order to obtain a satisfactory numerical definition, however, the surface had to be totally expressed with numbers.

At that time, circa 1960, very little if anything had been published about biparametric patches; the basic idea of UNISURF came from the study of a process often used in foundries to obtain a core: sand is compacted in a box (Fig. 1.12) and the shape of the upper surface of the core is obtained by scraping off the surplus with a timber plank cut as a template. Of course, a shape obtained by this method is relatively simple, since the shape of the plank is constant and that of the box edges is generally simple. To make the system more flexible, one could change the shape of the template at the same time as it is moved. This idea takes us back to a very old, and sometimes forgotten, definition of a surface: it is the locus of a curve which

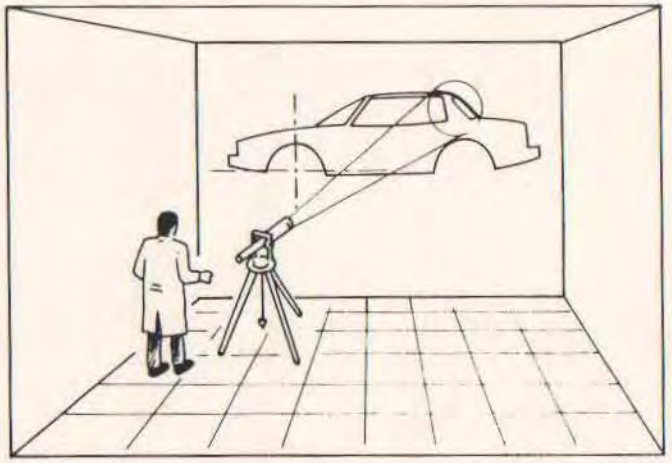

Figure 1.6. A projector producing a "template curve" on the drawing of an object.

is at the same time moved and distorted. About 1970, a Dutch laboratory sculptured blocks of styrofoam with a flexible strip of steel, heated by electricity, whose shape was controlled by the flexion torque imposed on its extremities.

This process could not produce a large variety of shapes, but the principle could be translated into a mathematical solution: the guiding edges of the box are similar to the curves AB and CD of Fig. 1.13, which can be considered as directrices of a surface, defined by their characteristic polygon. A curve such as EF being generatrix, defined by its own polygon, the ends of which run along lines AB and CD, and the intermediate vertices of the polygon being on curves GH and JK, the surface ABDC is known as soon as the four polygons are defined. Connecting the corresponding vertices of the polygons defines the "characteristic net" of the patch, which plays, regarding the surface, the same part as a polygon a curve. Hence, the cartesian coordinates of the points of the patch are computed according to the values of two parameters.

After the expression of this basic idea, many problems remained to be solved: choosing adequate functions, blending curves and patches and dealing with degenerate patches, to name only a few. Resolving these was a matter of relatively simple mathematics, the basic principle remaining untouched.

A system has thus been progressively created. We observe that the first solution—parallelogram, pantograph—is the result of an education oriented towards kinematics, the conception of mechanisms. Then appeared geom-

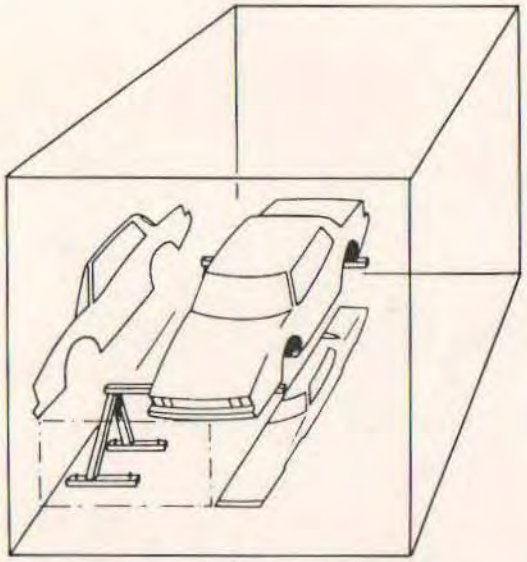

Figure 1.7. Two imaginary projections of a car.

etry and optics, which very likely came from some training in the army, when geometry, cosmography and topography played an important part. Reflexion was oriented towards analysis, parametric spaces and, finally, data processing; a theory, as convenient as it may look, must not impose too heavy a task on the computer and must be easily understood, at least in its principle, by the operators.

We note that the various steps of this conception have a point in common: each idea must be related to the principle of a material system, simple and primitive though it may look, on which a variable solution could be based.

An engineer is a man who defines what is to be done and how it could be done; he not only describes the goal, but he leads the way toward it.

Before we look any deeper into this subject, it should be observed that elementary geometry played a major part in its development, and it should not gradually disappear from the courses of a mechanical engineer; each idea and each hypothesis was expressed by a figure or a sketch representing a mechanism. It would have been extremely difficult to build a mental image of a somewhat elaborate system without the help of pencil and paper. Let us consider, for instance, Figs. 1.9 and 1.11; they are equivalent to equations (4.7) and (16.6) in subsequent chapters. Evidently, these formulas conveniently are best suited to express data given to a computer, but

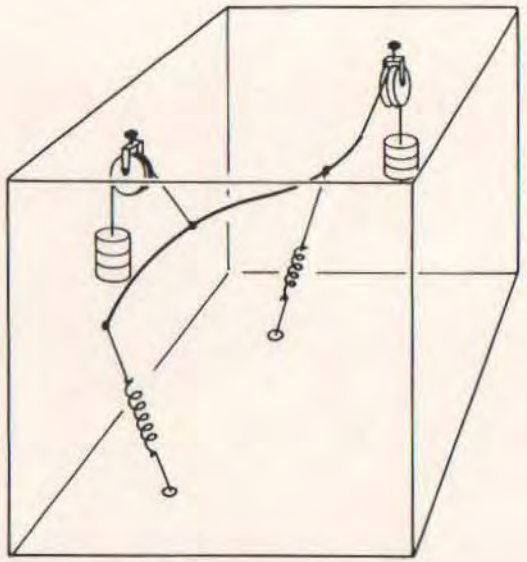

Figure 1.8. A curve held by springs.

most people would understand a simple figure better than the equivalent algebraic expression.

Napoleon said: "A short sketch is better than a long report."

Which are the parts played by experience, theory and imagination in the creation of a system? There is no definite answer to such a query. The importance of experience and of theoretical knowledge is not always clearly perceived; imagination seems a gift, a godsend or the result of a beneficial heredity; but is imagination not, in fact, the result of the maturation of knowledge gained during education and professional practice? Is it not born from facts apparently forgotten, stored in a distant part of the memory, and suddenly remembered when circumstances call them back? Is imagination not based partly on the ability to connect notions which, at first sight, look quite unrelated, such as mechanics, electronics, optics, foundry and data processing? Is it not the ability to catch barely seen analogies—as Alice in Wonderland did, to go "through the mirror"?

Will psychologists someday be able to detect in man such a gift that will be applicable to science and technology? Is it related to the sense of humor that can detect unexpected relationships between facts that look quite unconnected? Shall we learn how to develop it? Will it forever remain a gift, devoted by pure chance to some people while for others carefulness prevails?

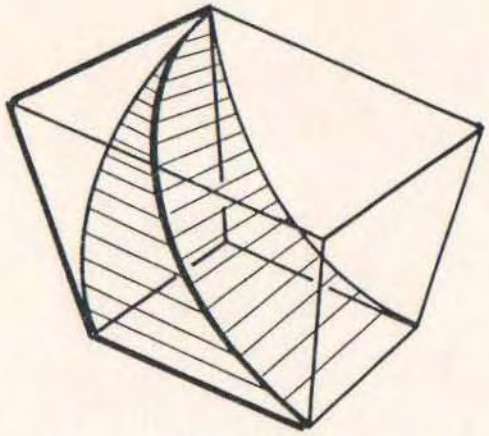

Figure 1.9. A curve defined inside a cube.

It is important that, sometimes, "sensible" men give free rein to imaginative people. "I succeeded," said Henry I. Ford, "because I let some fools try what wise people had advised me not to let them try."

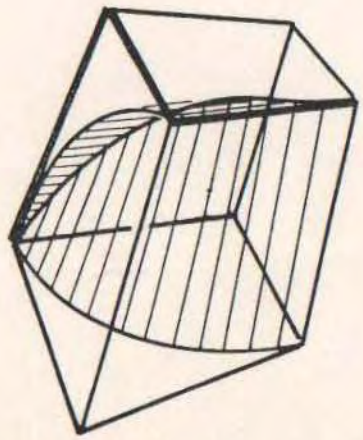

Figure 1.10. A curve defined inside a parallelepiped.

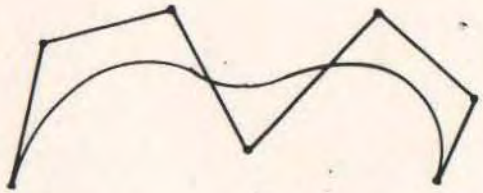

Figure 1.11. Higher order curves can be defined inside higher dimensional cubes.

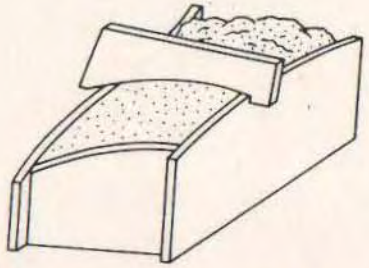

Figure 1.12. A surface is being obtained by scraping off excess material with wooden templates.

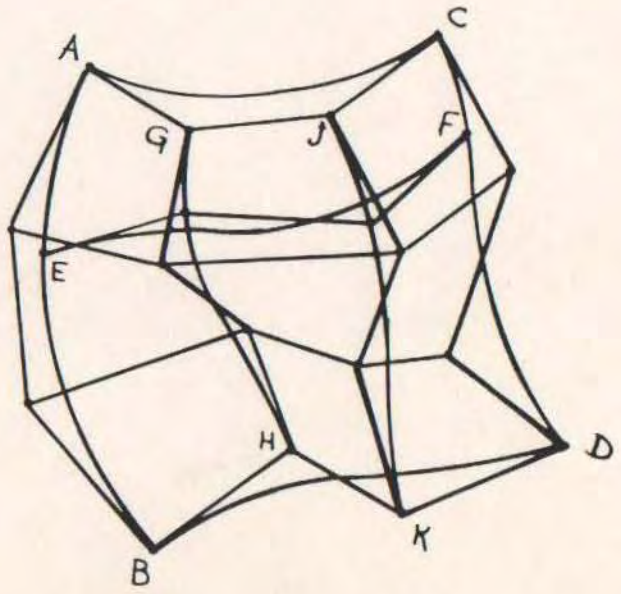

Figure 1.13. The characteristic net of a surface.

2

Introductory Material

2.1 Points and Vectors

When a designer or stylist works on an object, he or she does not think of that object in very mathematical terms. A point on the object would not be thought of as a triple of coordinates, but rather in functional terms: as a corner, the midpoint between two other points, and so on. The objective of this book, however, is to discuss objects that *are* defined in mathematical terms, the language that lends itself best to computer implementations. As a first step towards a mathematical description of an object, one therefore defines a *coordinate system* in which it will be described analytically.

The space in which we describe our object does not possess a preferred coordinate system – we have to define one ourselves. Many such systems could be picked (and some will certainly be more practical than others). But whichever one we choose, it should not affect any properties of the object itself. Our interest is in the object and not in its relationship to some arbitrary coordinate system; the methods that we will develop must therefore be independent of the choice of a coordinate system. We say that those methods must be *coordinate-free* .[1]

[1] More mathematically: the geometry of this book is affine geometry. The objects that we will consider "live" in affine spaces, not in linear spaces.

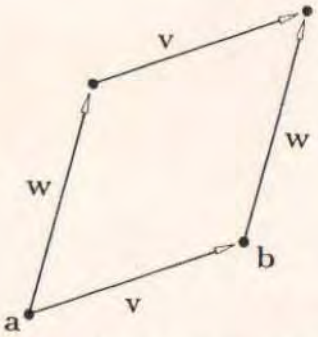

Figure 2.1. Points and vectors: vectors are are not affected by translations.

The concept of coordinate-free methods is stressed throughout in this book. It motivates the strict distinction between points and vectors as discussed next (for more details on this topic see R. Goldman [124]).

We shall denote *points*, elements of three-dimensional euclidean (or point) space $\mathbb{E}^3$, by lowercase boldface letters such as $\mathbf{a}$, $\mathbf{b}$ etc. (The term "euclidean space" is used here because it is a relatively familiar term to most people. More correctly, we should have used the term "affine space".) A point identifies a location, usually relative to other objects. Examples are the midpoint of a straight line segment or the center of gravity of a physical object.

The same notation (lowercase boldface) will be used for *vectors*, elements of three-dimensional linear (or vector) space $\mathbb{R}^3$. If we represent points or vectors by coordinates relative to some coordinate system, we shall adopt the convention of writing them as coordinate columns.

Although both points and vectors are described by triples of real numbers, we emphasize that there is a clear distinction between them: for any two points $\mathbf{a}$ and $\mathbf{b}$, there is a unique vector $\mathbf{v}$ that points from $\mathbf{a}$ to $\mathbf{b}$. It is computed by componentwise subtraction:

$$\mathbf{v} = \mathbf{b} - \mathbf{a}; \quad \mathbf{a}, \mathbf{b} \in \mathbb{E}^3, \quad \mathbf{v} \in \mathbb{R}^3.$$

On the other hand, given a vector $\mathbf{v}$, there are infinitely many pairs of points $\mathbf{a}$, $\mathbf{b}$ such that $\mathbf{v} = \mathbf{b} - \mathbf{a}$. For if $\mathbf{a}$, $\mathbf{b}$ is one such pair and if $\mathbf{w}$ is an arbitrary vector, then $\mathbf{a} + \mathbf{w}, \mathbf{b} + \mathbf{w}$ is another such pair since $\mathbf{v} = (\mathbf{b} + \mathbf{w}) - (\mathbf{a} + \mathbf{w})$ also. Figure 2.1 illustrates this fact.

Assigning the point $\mathbf{a} + \mathbf{w}$ to every point $\mathbf{a} \in \mathbb{E}^3$ is called a *translation*, and the above asserts that vectors are invariant under translations while points are not.

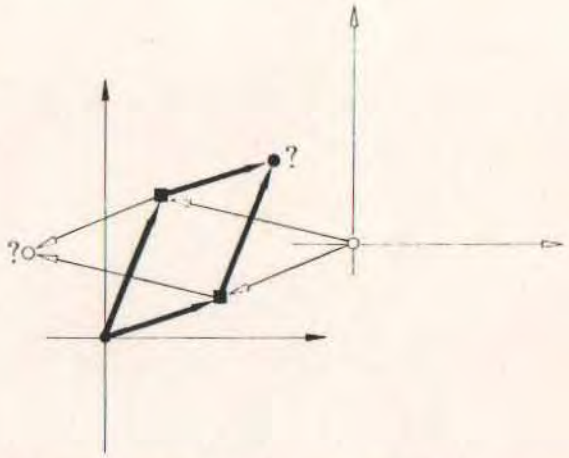

Figure 2.2. Addition of points: this is not a well-defined operation, since different coordinate systems would produce different "solutions".

Elements of point space $\mathbb{E}^3$ can only be *subtracted* – this operation yields a vector. They cannot be *added* – this operation is not defined for points. (It is defined for vectors.) Figure 2.2 gives an example.

However, addition-like operations are defined for points: they are *barycentric combinations.*[2] These are weighted sums of points where the weights sum to one:

$$\mathbf{b} = \sum_{j=0}^{n} \alpha_j \mathbf{b}_j; \quad \begin{matrix} \mathbf{b}_j \in \mathbb{E}^3 \\ \alpha_0 + \cdots + \alpha_n = 1 \end{matrix} . \tag{2.1}$$

At first glance, this looks like an undefined summation of points, but we can rewrite (2.1) as

$$\mathbf{b} = \mathbf{b}_0 + \sum_{j=1}^{n} \alpha_j (\mathbf{b}_j - \mathbf{b}_0),$$

which is clearly the sum of a point and a vector.

An example of a barycentric combination is the centroid $\mathbf{g}$ of a triangle with vertices $\mathbf{a}, \mathbf{b}, \mathbf{c}$, given by

$$\mathbf{g} = \frac{1}{3}\mathbf{a} + \frac{1}{3}\mathbf{b} + \frac{1}{3}\mathbf{c}.$$

The word "barycentric combination" is derived from "barycenter", meaning "center of gravity". The origin of this formulation is in physics: if the

[2] Also called *affine combinations.*

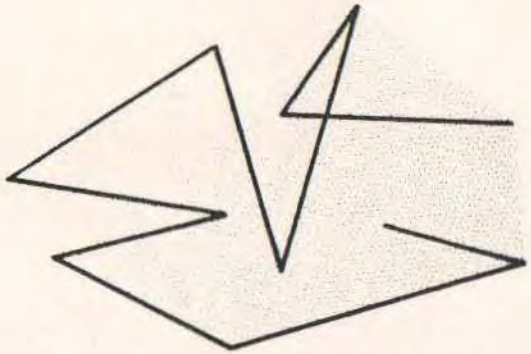

Figure 2.3. Convex hull: a point set (a polygon) and its convex hull.

$\mathbf{b}_j$ are centers of gravity of objects with masses m_j, then their center of gravity $\mathbf{b}$ is located at $\mathbf{b} = \sum m_j \mathbf{b}_j / \sum m_j$ and has the combined mass $\sum m_j$. (If some of the m_j are negative, the notion of electric charges may provide a better analogy; see Coxeter [67], p.214.) Since a common factor in the m_j is immaterial for the determination of the center of gravity, we may normalize them by setting $\sum m_j = 1$.

An important special case of barycentric combinations are the *convex combinations*. These are barycentric combinations where the coefficients α_j, in addition to summing to one, are also nonnegative. A convex combination of points is always "inside" those points, which is an observation that leads to the definition of the *convex hull* of a point set: this is the set that is formed by all convex combinations of a point set. Figure 2.3 gives an example. See also Problems.

2.2 Affine Maps

Most of the transformations that are used to position or scale an object in a computer graphics or CAD environment are *affine* maps. (More complicated, so-called projective maps are discussed in Section 14.) The name 'affine map' is due to L. Euler; affine maps were first studied systematically by A. Moebius, see collected works [181].

The fundamental operation for points is the barycentric combination. We will thus base the definition of an affine map on the notion of barycentric combinations. *A map Φ that maps $\mathbb{E}^3$ into itself is called an affine map if it leaves barycentric combinations invariant.* So if

$$\mathbf{x} = \sum \alpha_j \mathbf{a}_j; \quad \mathbf{x}, \mathbf{a}_j \in \mathbb{E}^3$$

and Φ is an affine map, then also

$$\Phi\mathbf{x} = \sum \alpha_j \Phi\mathbf{a}_j; \quad \Phi\mathbf{x}, \Phi\mathbf{a}_j \in \mathbb{E}^3. \tag{2.2}$$

This definition looks fairly abstract, yet has a simple interpretation. The expression $\mathbf{x} = \sum \alpha_j \mathbf{a}_j$ specifies how we have to weight the points $\mathbf{a}_j$ so that their weighted average is $\mathbf{x}$. This relation is still valid if we apply an affine map to all points $\mathbf{a}_j$ and to $\mathbf{x}$. As an example, the midpoint of a straight line segment will be mapped to the midpoint of the affine image of that straight line segment. Also, the centroid of a number of points will be mapped to the centroid of the image points.

Let us now be more specific. In a given coordinate system, a point $\mathbf{x}$ is represented by a coordinate triple, which we also denote by $\mathbf{x}$. An affine map now takes on the familiar form

$$\Phi\mathbf{x} = A\mathbf{x} + \mathbf{v}, \tag{2.3}$$

where A is a 3x3 matrix and $\mathbf{v}$ is a vector from $\mathbb{R}^3$.

A simple computation verifies that (2.3) does in fact describe an affine map, i.e. that barycentric combinations are preserved by maps of that form. For the following, recall that $\sum \alpha_j = 1$:

$$\begin{aligned} \Phi\left(\sum \alpha_j \mathbf{a}_j\right) &= A\left(\sum \alpha_j \mathbf{a}_j\right) + \mathbf{v} \\ &= \sum \alpha_j A\mathbf{a}_j + \sum \alpha_j \mathbf{v} \\ &= \sum \alpha_j \left(A\mathbf{a}_j + \mathbf{v}\right) \\ &= \sum \alpha_j \Phi\mathbf{a}_j, \end{aligned}$$

which concludes our proof.

Some examples of affine maps:

The identity. It is given by $\mathbf{v} = \mathbf{0}$, the zero vector, and by $A = I$, the identity matrix.

A translation. It is given by $A = I$, and a *translation vector* $\mathbf{v}$.

A scaling. It is given by $\mathbf{v} = \mathbf{0}$ and by a diagonal matrix A. The diagonal entries define by how much each component of the preimage $\mathbf{x}$ is to be scaled.

A rotation. If we rotate around the z-axis, then $\mathbf{v} = \mathbf{0}$ and

$$A = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A shear. An example is given by $\mathbf{v} = \mathbf{0}$ and

$$A = \begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}.$$

This family of shears maps the (x, y)−plane onto itself.

An important special case of affine maps are the *euclidean maps*, also called *rigid body motions*. They are characterized by orthonormal matrices A that are defined by the property $A^{\mathrm{T}}A = I$. Euclidean maps leave lengths and angles unchanged; the most important examples are rotations and translations.

Affine maps can be combined, and a complicated map may be decomposed into a sequence of simpler maps. It can be shown that every affine map can be composed of translations, rotations, shears, and scalings.

The *rank* of A has an important geometric interpretation: if $\mathrm{rank}(A) = 3$, then the affine map Φ maps three-dimensional objects to three-dimensional objects. If the rank is less than three, Φ is a parallel projection onto a plane (rank = 2) or even onto a straight line (rank = 1).

An affine map of $\mathbb{E}^2$ to $\mathbb{E}^2$ is uniquely determined by a (nondegenerate) triangle and its image. Thus any two triangles determine an affine map of the plane onto itself. In $\mathbb{E}^3$, an affine map is uniquely defined by a (nondegenerate) tetrahedron and its image.

More important facts about affine maps are discussed in the following section.

2.3 Linear Interpolation

Let $\mathbf{a}, \mathbf{b}$ be two distinct points in $\mathbb{E}^3$. The set of all points $\mathbf{x} \in \mathbb{E}^3$ of the form

$$\mathbf{x} = \mathbf{x}(t) = (1-t)\mathbf{a} + t\mathbf{b}; \quad t \in \mathbb{R} \tag{2.4}$$

is called the *straight line* through $\mathbf{a}$ and $\mathbf{b}$. Any three (or more) points on a straight line are said to be *collinear*.

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 Fig. 2.4.

Equation (2.4) represents a barycentric combination of two points in $\mathbb{E}^3$. The same barycentric combination holds for the three points $0, t, 1$ in $\mathbb{E}^1$: $t = (1-t)\cdot 0 + t \cdot 1$. So t is related to 0 and 1 by the same barycentric combination that relates $\mathbf{x}$ to $\mathbf{a}$ and $\mathbf{b}$. But then, by the definition of affine

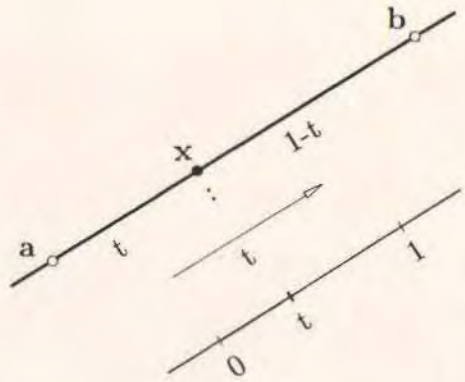

Figure 2.4. Linear interpolation: two points $\mathbf{a}, \mathbf{b}$ define a straight line through them. The point $\mathbf{x}$ on it divides the straight line segment between $\mathbf{a}$ and $\mathbf{b}$ in the ratio $t : 1-t$.

maps, the three points $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ in three-space are an affine map of the three points 0, t, 1 in one-space! *Thus linear interpolation is an affine map of the real line onto a straight line in* $\mathbb{E}^3$.[3]

It is now almost a tautology when we state: *linear interpolation is affinely invariant*. Written out as a formula: if Φ is an affine map of $\mathbb{E}^3$ onto itself, and (2.4) holds, then also

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

Closely related to linear interpolation is the concept of *barycentric coordinates*, due to Moebius [181]. Let $\mathbf{a}, \mathbf{x}, \mathbf{b}$ be three collinear points in $\mathbb{E}^3$:

$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b}; \quad \alpha + \beta = 1. \tag{2.5}$$

Then α and β are called *barycentric coordinates* with respect to $\mathbf{a}$ and $\mathbf{b}$. Note that by our previous definitions, $\mathbf{x}$ is a barycentric combination of $\mathbf{a}$ and $\mathbf{b}$.

The connection between barycentric coordinates and linear interpolation is obvious: we have $\alpha = 1 - t$ and $\beta = t$. This shows, by the way, that barycentric coordinates do not always have to be positive: for $t \notin [0, 1]$, either α or β is negative. For any three collinear points $\mathbf{a}, \mathbf{b}, \mathbf{c}$, the bary-

[3] Strictly speaking, we should therefore use the term "affine interpolation" instead of "linear interpolation." We use "linear interpolation" because of its widespread use.

centric coordinates of $\mathbf{b}$ with respect to $\mathbf{a}$ and $\mathbf{c}$ are given by

$$\alpha = \frac{\text{vol}_1(\mathbf{b},\mathbf{c})}{\text{vol}_1(\mathbf{a},\mathbf{c})},$$
$$\beta = \frac{\text{vol}_1(\mathbf{a},\mathbf{b})}{\text{vol}_1(\mathbf{a},\mathbf{c})}$$

where vol_1 denotes the one-dimensional volume, which is the signed distance between two points. Barycentric coordinates are not only defined on a straight line, but also on the plane. Section 18.1 has more details.

Another important concept in this context is that of *ratios*. The ratio of three collinear points $\mathbf{a}, \mathbf{b}, \mathbf{c}$ is defined by

$$\text{ratio}(\mathbf{a},\mathbf{b},\mathbf{c}) = \frac{\text{vol}_1(\mathbf{a},\mathbf{b})}{\text{vol}_1(\mathbf{b},\mathbf{c})}. \tag{2.6}$$

If α and β are barycentric coordinates of $\mathbf{b}$ with respect to $\mathbf{a}$ and $\mathbf{c}$, it follows that

$$\text{ratio}(\mathbf{a},\mathbf{b},\mathbf{c}) = \frac{\beta}{\alpha}. \tag{2.7}$$

The barycentric coordinates of a point do not change under affine maps, and neither does their quotient. Thus the ratio of three collinear points is not affected by affine transformations. So if (2.7) holds, then also

$$\text{ratio}(\Phi\mathbf{a},\Phi\mathbf{b},\Phi\mathbf{c}) = \frac{\beta}{\alpha}, \tag{2.8}$$

where Φ is an affine map.

The last equation states that *affine maps are ratio preserving*. This property may be used to define affine maps: every map that takes straight lines to straight lines and that is ratio preserving is an affine map.

The concept of ratio preservation may be used to derive another useful property of linear interpolation. We have defined the straight line segment $[\mathbf{a},\mathbf{b}]$ to be the affine image of the *unit interval* $[0,1]$, but we can also view that straight line segment as the affine image of any interval $[a,b]$. The interval $[a,b]$ may itself be obtained by an affine map from the interval $[0,1]$ or vice versa. With $t \in [0,1]$ and $u \in [a,b]$, that map is given by $t = (u-a)/(b-a)$. The interpolated point on the straight line is now given by both

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

and

$$\mathbf{x}(u) = \frac{b-u}{b-a}\mathbf{a} + \frac{u-a}{b-a}\mathbf{b}. \tag{2.9}$$

Since a, u, b and $0, t, 1$ are in the same ratio as the triple $\mathbf{a}, \mathbf{x}, \mathbf{b}$, we have shown that *linear interpolation is invariant under affine domain transformations*. By affine domain transformation, we simply mean an affine map of the real line onto itself. The parameter t is sometimes called a *local parameter* of the interval $[a,b]$.

A concluding remark: we have demonstrated the interplay between the two concepts of linear interpolation and ratios. In this book, we will often describe methods by saying that points have to be collinear and must be in a given ratio. This is the geometric (descriptive) equivalent of the algebraic (algorithmic) statement that one of the three points may be obtained by linear interpolation from the other two.

2.4 Piecewise Linear Interpolation

Let $\mathbf{b}_0, \ldots, \mathbf{b}_n \in \mathbb{E}^3$ form a *polygon* $\mathbf{B}$. A polygon consists of a sequence of straight line segments, each interpolating to a pair of points $\mathbf{b}_i, \mathbf{b}_{i+1}$. It is therefore also called the *piecewise linear interpolant* $\mathcal{PL}$ to the points $\mathbf{b}_i$. If the points $\mathbf{b}_i$ lie on a curve $\mathbf{c}$, then $\mathbf{B}$ is said to be a piecewise linear interpolant to $\mathbf{c}$, and we write

$$\mathbf{B} = \mathcal{PL}\mathbf{c}. \tag{2.10}$$

One of the important properties of piecewise linear interpolation is *affine invariance*: if the curve $\mathbf{c}$ is mapped onto a curve $\Phi\mathbf{c}$ by an affine map Φ, then the piecewise linear interpolant to $\Phi\mathbf{c}$ is the affine map of the original piecewise linear interpolant:

$$\mathcal{PL}\ \Phi\mathbf{c} = \Phi\ \mathcal{PL}\ \mathbf{c}. \tag{2.11}$$

Another property is the *variation diminishing property*: consider a continuous curve $\mathbf{c}$ and a piecewise linear interpolant $\mathcal{PL}\mathbf{c}$, and also an arbitrary plane. Let cross$\mathbf{c}$ be the number of crossings that the curve $\mathbf{c}$ has with this plane, and let cross$\mathcal{PL}\mathbf{c}$ be the number of crossings that the piecewise linear interpolant has with this plane. (Special cases may arise; see Problems.) Then we always have

$$\text{cross}\mathcal{PL}\ \mathbf{c} \leq \text{cross}\ \mathbf{c}. \tag{2.12}$$

This property follows from a simple observation: consider two points $\mathbf{b}_i$, $\mathbf{b}_{i+1}$. The straight line segment through them can cross a given plane at most at one point, while the curve segment from $\mathbf{c}$ that connects them may cross the same plane in arbitrarily many points. The variation diminishing property is illustrated in Fig. 2.5.

2.5 Function Spaces

This section contains material that will later simplify our work by allowing very concise notation. Although we shall try to develop our material with an emphasis on geometric concepts, it will sometimes simplify our work considerably if we can resort to some elementary topics from functional analysis. Good references are the books by Davis [68] and de Boor [74].

Let $C[a,b]$ be the set of all real-valued continuous functions defined over the interval $[a,b]$ of the real axis. We can define addition and multiplication by a constant for elements $f,g \in C[a,b]$ by setting $(\alpha f + \beta g)(t) = \alpha f(t) + \beta g(t)$ for all $t \in [a,b]$. With these definitions, it is easy to show that $C[a,b]$ forms a *linear space* over the reals. The same is true for the sets $C^k[a,b]$, the sets of all real-valued functions defined over $[a,b]$ that are k-times continuously differentiable. Furthermore, for every k, C^{k+1} is a *subspace* of C^k.

We say that n functions $f_1,\ldots,f_n \in C[a,b]$ are *linearly independent* if $\sum c_i f_i = 0$ for all $t \in [a,b]$ implies $c_1 = \ldots = c_n = 0$.

We mention some subspaces of $C[a,b]$ that will be of interest later. The spaces $\mathcal{P}^n$ of all *polynomials* of degree n:

$$p^n(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n; \quad t \in [a,b].$$

For fixed n, the dimension of $\mathcal{P}^n$ is $n+1$: each $p^n \in \mathcal{P}^n$ is determined uniquely by the $n+1$ coefficients $a_0,\ldots,a_n$. These can be interpreted as a vector in $(n+1)$-dimensional linear space $\mathbb{R}^{n+1}$, which has dimension $n+1$. We can also name a *basis* for $\mathcal{P}^n$: the *monomials* $1,t,t^2,\ldots,t^n$ are $n+1$ linearly independent functions and thus form a basis.

Another interesting class of subspaces of $C[a,b]$ is given the by piecewise linear functions: let $a = t_0 < t_1 < \cdots < t_n = b$ be a *partition* of the interval

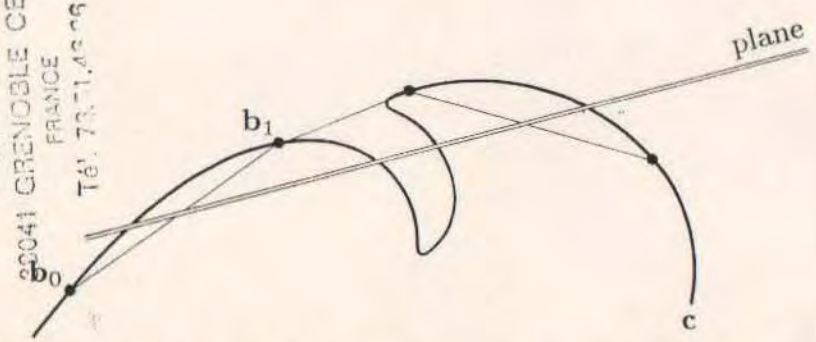

Figure 2.5. The variation diminishing property: a piecewise linear interpolant to a curve has no more intersections with any plane than the curve itself.

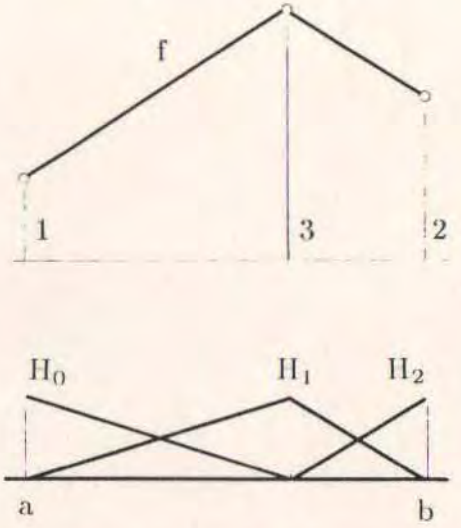

Figure 2.6. Hat functions: the piecewise linear function f can be written as $f = H_0 + 3H_1 + 2H_2$.

$[a,b]$. A continuous function that is linear on each subinterval $[t_i,t_{i+1}]$ is called a piecewise linear function. Over a fixed partition of $[a,b]$, the piecewise linear functions form a linear function space. A basis for this space is given by the *hat functions*: a hat function $H_i(t)$ is a piecewise linear function with $H_i(t_i) = 1$ and $H_i(t_j) = 0$ if $i \neq j$. A piecewise linear function l with $l(t_j) = f_j$ can always be written as

$$l(t) = \sum_{j=0}^{n} f_j H_j(t).$$

Figure 2.6 gives an example.

We will also consider *linear operators* that assign a function $\mathcal{A}f$ to a given function f. An operator $\mathcal{A} : C[a,b] \to C[a,b]$ is called *linear* if it leaves linear combinations invariant:

$$\mathcal{A}(\alpha f + \beta g) = \alpha \mathcal{A}f + \beta \mathcal{A}g; \quad \alpha,\beta \in \mathbb{R}.$$

An example is given by the derivative operator that assigns the derivative f' to a given function f: $\mathcal{A}f = f'$.

2.6 Problems

1. We have seen that affine maps leave the ratio of three collinear points constant, i.e., they are ratio-preserving. Show that the converse is also true: every ratio-preserving map is affine.
2. We defined the convex hull of a point set to be the set of all convex combinations formed by the elements of that set. Another definition is the following: the convex hull of a point set is the intersection of all convex sets that contain the given set. Show that both definitions are equivalent.
3. In the definition of the variation diminishing property we counted the crossings of a polygon with a plane. Discuss the case when the plane contains a whole polygon leg.
4. Show that the $n+1$ functions $f_i(t) = t^i;\ i = 0, \ldots, n$ are linearly independent.

3

The de Casteljau Algorithm

The algorithm described in this chapter is probably the most fundamental one in the field of curve and surface design, yet it is surprisingly simple. Its main attraction is the beautiful interplay between geometry and algebra: a very intuitive geometric construction leads to a powerful theory.

Historically, it is with this algorithm that the work of de Casteljau started in 1959. The only written evidence is in [78] and [77]; both technical reports that are not easily accessible. De Casteljau's work went unnoticed until W. Boehm obtained copies of the reports in 1975. From then on, de Casteljau's name gained more popularity.

3.1 Parabolas

We give a simple construction for the generation of a parabola; the straightforward generalization will then lead to Bézier curves. Let $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ be any three points in $\mathbb{E}^3$, and let $t \in \mathbb{R}$. Construct

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

$$\mathbf{b}_0^2(t) = (1-t)\mathbf{b}_0^1(t) + t\mathbf{b}_1^1(t).$$

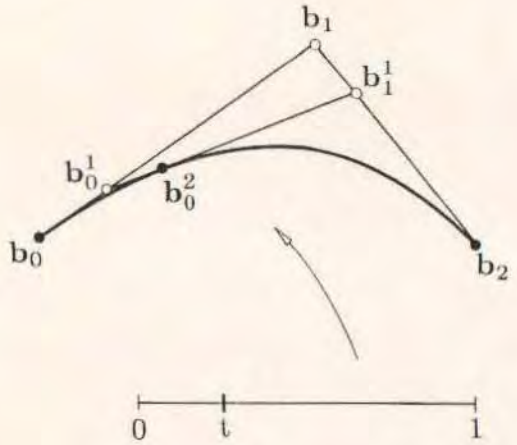

Figure 3.1. Parabolas: construction by repeated linear interpolation.

Inserting the first two equations into the third one, we obtain

$$\mathbf{b}_0^2(t) = (1-t)^2\mathbf{b}_0 + 2t(1-t)\mathbf{b}_1 + t^2\mathbf{b}_2. \tag{3.1}$$

This is a quadratic expression in t (the superscript denoting the degree), and so $\mathbf{b}_0^2(t)$ traces out a *parabola* as t varies from $-\infty$ to $+\infty$. We denote this parabola by $\mathbf{b}^2$.

The above construction consists of *repeated linear interpolation* ; its geometry is illustrated in Fig. 3.1. For t between 0 and 1, $\mathbf{b}^2(t)$ is inside the triangle formed by $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$, in particular $\mathbf{b}^2(0) = \mathbf{b}_0, \mathbf{b}^2(1) = \mathbf{b}_2$.

Inspecting the ratios of points in Fig. 3.1, we see that

$$\text{ratio}(\mathbf{b}_0, \mathbf{b}_0^1, \mathbf{b}_1) = \text{ratio}(\mathbf{b}_1, \mathbf{b}_1^1, \mathbf{b}_2) = \text{ratio}(\mathbf{b}_0^1, \mathbf{b}_0^2, \mathbf{b}_1^1) = t/(1-t).$$

Thus our construction of a parabola is *affinely invariant* because piecewise linear interpolation is affinely invariant; see section 2.4.

We also note that a parabola is a plane curve, due to the fact that $\mathbf{b}^2(t)$ is always a barycentric combination of three points, as is clear from inspecting (3.1). A parabola is a special case of *conic sections*; these will be discussed in Chapter 14.

Finally we state a theorem from analytic geometry, closely related to our parabola construction. Let $\mathbf{a}, \mathbf{b}, \mathbf{c}$ be three distinct points on a parabola. Let the tangent at $\mathbf{b}$ intersect the tangents at $\mathbf{a}$ and $\mathbf{c}$ in $\mathbf{e}$ and $\mathbf{f}$, respectively. Let the tangents at $\mathbf{a}$ and $\mathbf{c}$ intersect in $\mathbf{d}$. Then $\text{ratio}(\mathbf{a}, \mathbf{e}, \mathbf{d}) = \text{ratio}(\mathbf{e}, \mathbf{b}, \mathbf{f}) = \text{ratio}(\mathbf{d}, \mathbf{f}, \mathbf{c})$. This *three tangent theorem* describes a property of parabolas; the de Casteljau algorithm can be viewed as the constructive counterpart.

3.2 The de Casteljau Algorithm

Parabolas are plane curves. Many applications require true space curves, however.[1] For those purposes, the above construction for a parabola can be generalized to generate a polynomial curve of arbitrary degree n:

de Casteljau algorithm

Given: $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{E}^3$ and $t \in \mathbb{R}$,

set

$$\mathbf{b}_i^r(t) = (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t) \qquad \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases} \tag{3.2}$$

and $\mathbf{b}_i^0(t) = \mathbf{b}_i$. Then $\mathbf{b}_0^n(t)$ is the point with parameter value t on the *Bézier curve* $\mathbf{b}^n$.

The polygon $\mathbf{P}$ formed by $\mathbf{b}_0, \ldots, \mathbf{b}_n$ is called the *Bézier polygon* or *control polygon* of the curve $\mathbf{b}^n$. Similarly, the polygon vertices $\mathbf{b}_i$ are called *control points* or *Bézier points*. Fig. 3.2 illustrates the cubic case.

Sometimes we also write $\mathbf{b}^n(t) = \mathcal{B}[\mathbf{b}_0, \ldots, \mathbf{b}_n; t] = \mathcal{B}[\mathbf{P}; t]$ or, shorter, $\mathbf{b}^n = \mathcal{B}[\mathbf{b}_0, \ldots, \mathbf{b}_n] = \mathcal{B}\mathbf{P}$. This notation defines $\mathcal{B}$ to be the (linear) operator that associates the Bézier curve with its control polygon. We sometimes say that the curve $\mathcal{B}[\mathbf{b}_0, \ldots, \mathbf{b}_n]$ is the *Bernstein-Bézier approximation* to the control polygon, a terminology borrowed from approximation theory. (See also section 5.10.)

The intermediate coefficients $\mathbf{b}_i^r(t)$ are conveniently written into a triangular array of points, the *de Casteljau scheme* – we give the example of the cubic case:

$$\begin{array}{llll} \mathbf{b}_0 & & & \\ \mathbf{b}_1 & \mathbf{b}_0^1 & & \\ \mathbf{b}_2 & \mathbf{b}_1^1 & \mathbf{b}_0^2 & \\ \mathbf{b}_3 & \mathbf{b}_2^1 & \mathbf{b}_1^2 & \mathbf{b}_0^3. \end{array} \tag{3.3}$$

This triangular array of points seems to suggest the use of a two-dimensional array in writing code for the de Casteljau algorithm. That would be a waste of storage, however: it is sufficient to use the left column only and to overwrite it appropriately.

[1] Compare the comments by P. Bézier in Chapter 1!

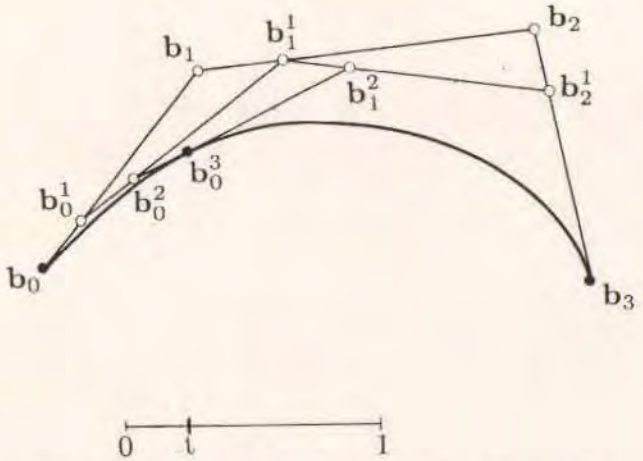

Figure 3.2. The de Casteljau algorithm: the point $\mathbf{b}_0^3(t)$ is obtained from repeated linear interpolation. The cubic case $n = 3$ is shown for $t = 1/4$.

3.3 Some Properties of Bézier Curves

The de Casteljau algorithm allows us to infer several important properties of Bézier curves. We will infer these properties from the geometry underlying the algorithm. In the next chapter, we will also show how they can be derived analytically.

Affine invariance. Affine maps were discussed in section 2.2. They are in the tool kit of every CAD system: objects must be repositioned, scaled, and so on. An important property of Bézier curves is that they are invariant under affine maps, which means that the following two procedures yield the same result: a) first, compute the point $\mathbf{b}^n(t)$ and then apply an affine map to it. b) first, apply an affine map to the control polygon and then evaluate the mapped polygon at parameter value t.

Affine invariance is, of course, a direct consequence of the de Casteljau algorithm: the algorithm is composed of a sequence of linear interpolations (or, equivalently, of a sequence of affine maps). These are themselves affinely invariant, and so is a finite sequence of them.

Let us discuss a practical aspect of affine invariance. Suppose we plot a cubic curve $\mathbf{b}^3$ by evaluating at 100 points and then plotting the resulting point array. Suppose now that we would like to plot the curve after a rotation has been applied to it. We can take the hundred

computed points, apply the rotation to each of them and plot. Or, we can apply the rotation to the four control points, then evaluate one hundred times and plot. The first method needs one hundred applications of the rotation, while the second needs only four!

Affine invariance may not seem to be a very exceptional property for a useful curve scheme; in fact, it is not straightforward to think of a curve scheme that does not have it (exercise!). It is perhaps worth noting that Bézier curves do *not* enjoy another, also very important, property: they are *not projectively invariant*. Projective maps are used in computer graphics when an object is to be rendered realistically. So if we try to make life easy and simplify a perspective map of a Bézier curve by mapping the control polygon and then computing the curve, we have actually cheated: that curve is not the perspective image of the original curve! More details on perspective maps can be found in Chapter 14.

Invariance under affine parameter transformations Very often, one thinks of a Bézier curve as being defined over the interval $[0, 1]$. This is done because it is convenient, not because it is necessary: the de Casteljau algorithm is "blind" to the actual interval that the curve is defined over because it uses ratios only. One may therefore think of the curve as being defined over any arbitrary interval $a \le u \le b$ of the real line – after the introduction of local coordinates $t = (u - a)/(b - a)$, the algorithm proceeds as usual. This property is inherited from linear interpolation, as described in section 2.3.

The transition from the interval $[0, 1]$ to the interval $[a, b]$ is an *affine map*. Therefore, we can say that Bézier curves are invariant under affine parameter transformations. Sometimes, one sees the term *linear parameter transformation* in this context, but this terminology is not quite correct: the transformation of the interval $[0, 1]$ to $[a, b]$ typically includes a translation, which is not a linear map.

Convex hull property For $t \in [0, 1]$, $\mathbf{b}^n(t)$ lies in the convex hull (see section 2.3) of the control polygon. This follows since every intermediate $\mathbf{b}_i^r$ is obtained as a convex barycentric combination of previous $\mathbf{b}_j^{r-1}$ – at no step of the de Casteljau algorithm do we produce points outside the convex hull of the $\mathbf{b}_i$.

A simple consequence is that a planar control polygon always generates a planar curve.

Endpoint interpolation The Bézier curve passes through $\mathbf{b}_0$ and $\mathbf{b}_n$: we have $\mathbf{b}^n(0) = \mathbf{b}_0, \mathbf{b}^n(1) = \mathbf{b}_n$. This is easily verified by writing down

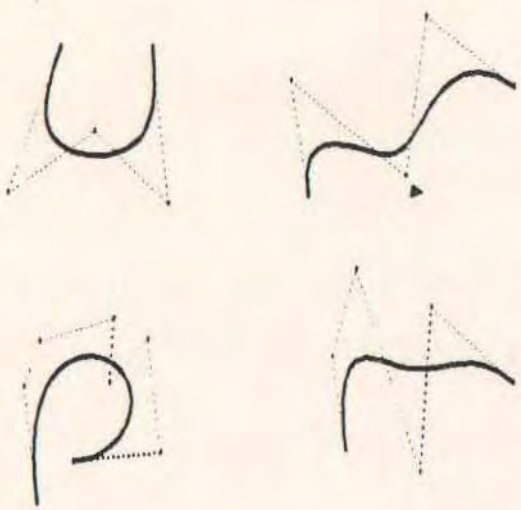

Figure 3.3. Bézier curves: some examples. The marked control vertex is multiply defined: $\mathbf{b}_2 = \mathbf{b}_3 = \mathbf{b}_4$.

the scheme (3.3) for the cases $t = 0$ and $t = 1$. In a design situation, the endpoints of a curve are certainly two very important points. It is therefore essential to have direct control over them, which is assured by endpoint interpolation.

Designing with Bézier curves Figure 3.3 shows several Bézier curves. From the inspection of these examples, one gets the impression that in some sense the Bézier curve "mimicks" the Bézier polygon – this statement will be made more precise later. This is the reason why Bézier curves provide such a handy tool for the *design* of curves: In order to reproduce the shape of a handdrawn curve, it is sufficient to specify a control polygon that somehow 'exaggerates' the shape of the curve. One lets the computer draw the Bézier curve defined by the polygon, and, if necessary, adjusts the location (possibly also the number) of the polygon vertices. Typically, an experienced person will reproduce a given curve after two to three iterations of this *interactive* procedure.

We will subsequently derive more properties of Bézier curves; in order to do so, we shall develop the so-called Bernstein representation in the next section.

3.4 Problems

1. Use the notation $\mathbf{b}^n(t) = B[\mathbf{b}_0, \ldots, \mathbf{b}_n; t]$ to formulate the de Casteljau algorithm. Comment on the computation count if you implement such a recursive algorithm in a language like C.

2. Suppose a planar Bézier curve has a control polygon that is symmetric with respect to the $y-$axis. Is the curve also symmetric with respect to the $y-$axis? Generalize to other symmetry properties.

3. Show that every nonplanar cubic in $I\!E^3$ can be obtained as an affine map of the *standard cubic* (see Boehm [43])

$$\mathbf{x}(t) = \begin{bmatrix} t \\ t^2 \\ t^3 \end{bmatrix}.$$

4

The Bernstein Form of a Bézier Curve

Bézier curves can be defined by a recursive algorithm, and this is how de Casteljau first developed them. It is also necessary, however, to have an *explicit* representation for them, i.e., to express a Bézier curve in terms of a nonrecursive formula rather than in terms of an algorithm. This will facilitate further theoretical development considerably.

4.1 Bernstein Polynomials

We will express Bézier curves in terms of *Bernstein polynomials*, defined explicitly by

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}. \tag{4.1}$$

There is a fair amount of literature on these polynomials. We cite just a few: Bernstein [33], Lorentz [174], Davis [68], Korovkin [163]. An extensive bibliography is in Gonska and Meier [127].

Before we explore the importance of Bernstein polynomials to Bézier curves, let us first examine them more closely. One of their important

properties is that they satisfy the following recursion:

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \tag{4.2}$$

with

$$B_0^0(t) \equiv 1 \tag{4.3}$$

and

$$B_j^n(t) \equiv 0 \quad \text{for } j \notin \{0,\dots,n\}. \tag{4.4}$$

The proof is simple:

$$\begin{aligned} B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n-1}{i} t^i (1-t)^{n-i} + \binom{n-1}{i-1} t^i (1-t)^{n-i} \\ &= (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t). \end{aligned}$$

Another important property is that Bernstein polynomials form a *partition of unity*:

$$\sum_{j=0}^{n} B_j^n(t) \equiv 1. \tag{4.5}$$

This fact is proved with the help of the binomial theorem:

$$1 = (t + (1-t))^n = \sum_{j=0}^{n} \binom{n}{j} t^j (1-t)^{n-j} = \sum_{j=0}^{n} B_j^n(t).$$

Figure 4.1 shows the family of the five quartic Bernstein polynomials. Note that the B_i^n are nonnegative over the interval $[0,1]$.

We are now ready to see why Bernstein polynomials are important for the development of Bézier curves. The intermediate de Casteljau points $\mathbf{b}_i^r$ can be expressed in terms of Bernstein polynomials of degree r:

$$\mathbf{b}_i^r(t) = \sum_{j=0}^{r} \mathbf{b}_{i+j} B_j^r(t) \qquad \begin{matrix} \in \{0,n\} \\ i \in \{0,n-r\}. \end{matrix} \tag{4.6}$$

This equation shows exactly how the intermediate point $\mathbf{b}_i^r$ depends on the given Bézier points $\mathbf{b}_i$. The main importance of (4.6), however, is for the case $r = n$. The corresponding de Casteljau point is the point on the curve and is given by

$$\mathbf{b}^n(t) = \sum_{j=0}^{n} \mathbf{b}_j B_j^n(t). \tag{4.7}$$

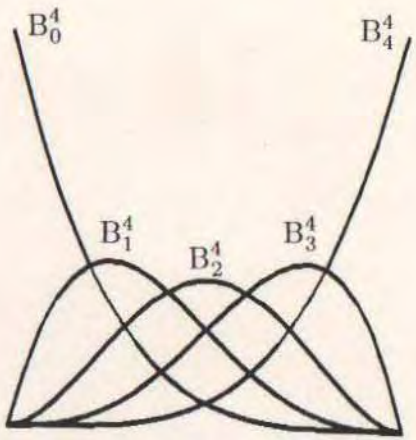

Figure 4.1. Bernstein polynomials: the quartic case.

We still have to prove (4.6). To that end, we use the recursive definition of the $\mathbf{b}_i^r$ (Equation 3.2) and the recursion for the Bernstein polynomials (4.2) and (4.4) in an inductive proof:

$$\begin{aligned} \mathbf{b}_i^r(t) &= (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t) \\ &= (1-t)\sum_{j=i}^{i+r-1} \mathbf{b}_j B_{j-i}^{r-1}(t) + t\sum_{j=i+1}^{i+r} \mathbf{b}_j B_{j-i-1}^{r-1}(t). \end{aligned}$$

Invoking (4.4), we can rewrite this as

$$\begin{aligned} \mathbf{b}_i^r(t) &= (1-t)\sum_{j=i}^{i+r} \mathbf{b}_j B_{j-i}^{r-1}(t) + t\sum_{j=i}^{i+r} \mathbf{b}_j B_{j-i-1}^{r-1}(t) \\ &= \sum_{j=i}^{i+r} \mathbf{b}_j [(1-t)B_{j-i}^{r-1}(t) + tB_{j-i-1}^{r-1}(t)], \end{aligned}$$

which completes the proof. Note that (4.2) also defines B_0^n and B_n^n, since $B_{-1}^{n-1} = B_n^{n-1} = 0$ by (4.4).

4.2 Properties of Bézier Curves

Many of the properties in this section have already appeared in the previous chapter. They were derived using geometric arguments. We shall now

rederive several of them, using algebraic arguments. If the same heading is used as in the last chapter, the reader should look there for a complete description of the property in question.

Affine invariance Barycentric combinations are invariant under affine maps, and so (4.5) gives the algebraic verification of this property.

Invariance under affine parameter transformations Algebraically, this property reads

$$\sum_{i=0}^{n} \mathbf{b}_i B_i^n(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(\frac{u-a}{b-a}). \tag{4.8}$$

Convex hull property This follows, since for $t \in [0,1]$, the Bernstein polynomials are nonnegative. They sum to one as shown in (4.5).

Endpoint interpolation This is a consequence of the identities

$$\begin{aligned} B_i^n(0) &= 1 \text{ iff } i = 0, \\ B_i^n(1) &= 1 \text{ iff } i = n \end{aligned} \tag{4.9}$$

and (4.5).

Symmetry Looking at the examples in Fig. 3.3, it is clear that it does not matter if the Bézier points are labeled $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n$ or $\mathbf{b}_n, \mathbf{b}_{n-1}, \ldots, \mathbf{b}_0$. The curves that correspond to the two different orderings look the same; they only differ in the direction in which they are traversed. Written out as a formula:

$$\sum_{j=0}^{n} \mathbf{b}_j B_j^n(t) = \sum_{j=0}^{n} \mathbf{b}_{n-j} B_j^n(1-t). \tag{4.10}$$

This follows from the identity

$$B_j^n(t) = B_{n-j}^n(1-t), \tag{4.11}$$

which follows from inspection of (4.1). We say that Bernstein polynomials are *symmetric* with respect to t and $1-t$.

Invariance under barycentric combinations The process of forming the Bézier curve from the Bézier polygon leaves barycentric combinations invariant: for $\alpha + \beta = 1$, we get

$$\sum_{j=0}^{n} (\alpha \mathbf{b}_j + \beta \mathbf{c}_j) B_i^n(t) = \alpha \sum_{j=0}^{n} \mathbf{b}_j B_j^n(t) + \beta \sum_{j=0}^{n} \mathbf{c}_j B_j^n(t). \tag{4.12}$$

In words: we can construct the weighted average of two Bézier curves either by taking the weighted average of corresponding points on the curves, or by taking the weighted average of corresponding control vertices and then computing the curve.

This linearity property is essential for many theoretical purposes, the most important one being the definition of tensor product surfaces in Chapter 16.

Linear precision A useful identity is the following:

$$\sum_{j=0}^{n} \frac{j}{n} B_j^n(t) = t, \tag{4.13}$$

which has the following application: suppose the polygon vertices $\mathbf{b}_j$ are uniformly distributed on a straight line joining two points $\mathbf{p}$ and $\mathbf{q}$:

$$\mathbf{b}_j = (1 - \frac{j}{n})\mathbf{p} + \frac{j}{n}\mathbf{q}; \quad j = 0, \ldots, n.$$

The curve that is generated by this polygon is the straight line between $\mathbf{p}$ and $\mathbf{q}$, i.e., the initial straight line is reproduced. This property is called *linear precision*.[1]

Pseudo-local control The Bernstein polynomial B_i^n has only one maximum and attains it at $t = i/n$. This has a design application: if we move only one of the control polygon vertices, say, $\mathbf{b}_i$, then the curve is mostly affected by this change in the region of the curve around the parameter value i/n. This makes the effect of the change reasonably predictable, although the change does affect the whole curve.

4.3 The Derivative of a Bézier Curve

The derivative of a Bernstein polynomial B_i^n is obtained as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} B_i^n(t) &= \frac{\mathrm{d}}{\mathrm{d}t} \binom{n}{i} t^i (1-t)^{n-i} \\ &= \frac{i\,n!}{i!(n-i)!} t^{i-1}(1-t)^{n-i} - \frac{(n-i)n!}{i!(n-i)!} t^i (1-t)^{n-i-1} \\ &= \frac{n(n-1)!}{(i-1)!(n-i)!} t^{i-1}(1-t)^{n-i} - \frac{n(n-1)!}{i!(n-i-1)!} t^i (1-t)^{n-i-1} \\ &= n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)). \end{aligned}$$

[1] If the points are not uniformly spaced, we will also recapture the straight line segment. However, it will not be linearly parametrized.

Thus

$$\frac{\mathrm{d}}{\mathrm{d}t}B_i^n(t) = n\left(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)\right). \tag{4.14}$$

We can now determine the derivative of a Bézier curve $\mathbf{b}^n$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n\sum_{j=0}^{n}\left(B_{j-1}^{n-1}(t) - B_j^{n-1}(t)\right)\mathbf{b}_j.$$

Because of (4.4), this can be simplified to

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n\sum_{j=1}^{n} B_{j-1}^{n-1}(t)\mathbf{b}_j - n\sum_{j=0}^{n-1} B_j^{n-1}(t)\mathbf{b}_j,$$

and now an index transformation of the first sum yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n\sum_{j=0}^{n-1} B_j^{n-1}(t)\mathbf{b}_{j+1} - n\sum_{j=0}^{n-1} B_j^{n-1}(t)\mathbf{b}_j,$$

and finally

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n\sum_{j=0}^{n-1}(\mathbf{b}_{j+1} - \mathbf{b}_j)B_j^{n-1}(t).$$

The last formula can be simplified somewhat by the introduction of the *forward difference operator* Δ:

$$\Delta\mathbf{b}_j = \mathbf{b}_{j+1} - \mathbf{b}_j. \tag{4.15}$$

We now have for the derivative of a Bézier curve:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n\sum_{j=0}^{n-1}\Delta\mathbf{b}_j B_j^{n-1}(t); \quad \Delta\mathbf{b}_j \in \mathbb{R}^3. \tag{4.16}$$

The derivative of a Bézier curve is thus another Bézier curve, obtained by differencing the original control polygon. However, this derivative Bézier curve does not "live" in $\mathbb{E}^3$ any more! Its coefficients are differences of points, i.e., *vectors*, which are elements of $\mathbb{R}^3$. In order to visualize the derivative curve and polygon in $\mathbb{E}^3$, we can construct a polygon in $\mathbb{E}^3$ that consists of the points $\mathbf{a} + \Delta\mathbf{b}_0, \ldots, \mathbf{a} + \Delta\mathbf{b}_{n-1}$. Here $\mathbf{a}$ is arbitrary; one reasonable choice is $\mathbf{a} = \mathbf{0}$. Figure 4.2 illustrates a Bézier curve and its derivative curve (with the choice $\mathbf{a} = \mathbf{0}$). This derivative curve is sometimes called a *hodograph*. For more information on hodographs, see Forrest [114].

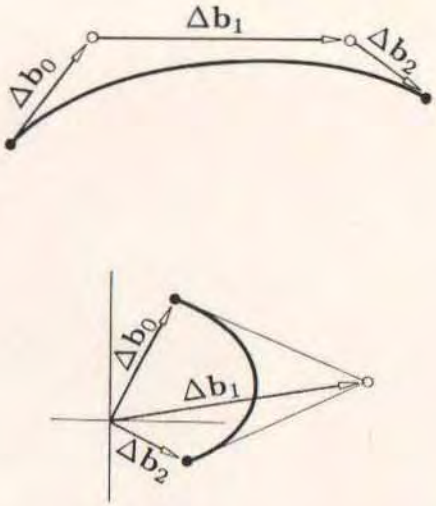

Figure 4.2. Derivatives: a Bézier curve and its first derivative curve (scaled down by a factor of three). Note that this derivative curve does not change if a translation is applied to the original curve.

4.4 Higher Order Derivatives

In order to compute higher derivatives, we first generalize the forward difference operator (4.15): the *iterated forward difference operator* Δ^r is defined by

$$\Delta^r\mathbf{b}_j = \Delta^{r-1}\mathbf{b}_{j+1} - \Delta^{r-1}\mathbf{b}_j. \tag{4.17}$$

We list a few examples:

$$\begin{aligned}
\Delta^0\mathbf{b}_i &= \mathbf{b}_i \\
\Delta^1\mathbf{b}_i &= \mathbf{b}_{i+1} - \mathbf{b}_i \\
\Delta^2\mathbf{b}_i &= \mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i \\
\Delta^3\mathbf{b}_i &= \mathbf{b}_{i+3} - 3\mathbf{b}_{i+2} + 3\mathbf{b}_{i+1} - \mathbf{b}_i.
\end{aligned}$$

The factors on the right hand sides are binomial coefficients, forming a Pascal-like triangle. This pattern holds in general:

$$\Delta^r\mathbf{b}_i = \sum_{j=0}^{r}\binom{r}{j}(-1)^{r-j}\mathbf{b}_{i+j}. \tag{4.18}$$

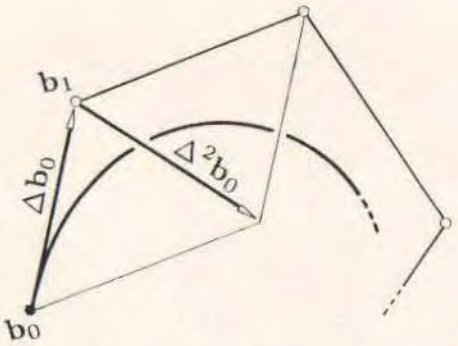

Figure 4.3. Endpoint derivatives: the first and second derivative vectors at $t = 0$ are multiples of the first and second difference vectors at $\mathbf{b}_0$.

We are now in a position to give the formula for the r-th derivative of a Bézier curve:

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbf{b}^n(t) = \frac{n!}{(n-r)!}\sum_{j=0}^{n-r}\Delta^r\mathbf{b}_j B_j^{n-r}(t). \tag{4.19}$$

The proof of (4.19) is by repeated application of (4.16).

Two important special cases of (4.19) are given by $t = 0$ and $t = 1$. Because of (4.9) we get

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbf{b}^n(0) = \frac{n!}{(n-r)!}\Delta^r\mathbf{b}_0, \tag{4.20}$$

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbf{b}^n(1) = \frac{n!}{(n-r)!}\Delta^r\mathbf{b}_{n-r}. \tag{4.21}$$

Thus the r-th derivative of a Bézier curve at an endpoint depends only on the $r+1$ Bézier points near (and including) that endpoint. For $r = 0$, we get the already established property of endpoint interpolation. The case $r = 1$ states that $\mathbf{b}_0$ and $\mathbf{b}_1$ define the tangent at $t = 0$, provided they are distinct.[2] Similarly, $\mathbf{b}_{n-1}$ and $\mathbf{b}_n$ determine the tangent at $t = 1$. The cases $r = 1$, $r = 2$ are illustrated in Fig. 4.3.

[2] In general, the tangent at $\mathbf{b}_0$ is determined by $\mathbf{b}_0$ and the first $\mathbf{b}_i$ that is distinct from $\mathbf{b}_0$. Thus the tangent may be defined even if the tangent vector is the zero vector.

4.5 Derivatives and the de Casteljau Algorithm

Derivatives of a Bézier curve can be expressed in terms of the intermediate points generated by the de Casteljau algorithm:

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbf{b}^n(t) = \frac{n!}{(n-r)!}\Delta^r\mathbf{b}_0^{n-r}(t). \tag{4.22}$$

This follows since summation and taking differences commute:

$$\sum_{j=0}^{n-1}\Delta\mathbf{b}_j = \sum_{j=1}^{n}\mathbf{b}_j - \sum_{j=0}^{n-1}\mathbf{b}_j. \tag{4.23}$$

Using this, we have

$$\begin{aligned}\frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbf{b}^n(t) &= \frac{n!}{(n-r)!}\sum_{j=0}^{n-r}\Delta^r\mathbf{b}_j B_j^{n-r}(t)\\ &= \frac{n!}{(n-r)!}\Delta^r\sum_{j=0}^{n-r}\mathbf{b}_j B_j^{n-r}(t)\\ &= \frac{n!}{(n-r)!}\Delta^r\mathbf{b}_0^{n-r}(t).\end{aligned}$$

As a practical implication, we see that derivatives of a Bézier curve may be computed as a "byproduct" of the de Casteljau algorithm. If we compute a point on a Bézier curve using a triangular arrangement as in (3.3), then for any $n - r$, the corresponding $\mathbf{b}_i^{n-r}$ form a column (with r entries) in that scheme. To obtain the r^{th} derivative at t, we simply take the r^{th} difference of these points and then multiply by the constant $n!/(n-r)!$. In some applications (curve/plane intersection, for example), one needs not only a point on the curve, but its first and/or second derivative at the same time. The de Casteljau algorithm offers a quick solution to this problem.

The case $r = 1$ is important enough to warrant special attention:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{b}^n(t) = n(\mathbf{b}_1^{n-1}(t) - \mathbf{b}_0^{n-1}(t)). \tag{4.24}$$

The intermediate points $\mathbf{b}_0^{n-1}$ and $\mathbf{b}_1^{n-1}$ thus determine the *tangent vector* at $\mathbf{b}^n(t)$, which is illustrated in Figs. 3.1 and 3.2.

4.6 The Matrix Form of a Bézier Curve

Some authors (Faux and Pratt [106], Mortenson [182], Chang [58]) prefer to write Bézier curves and other polynomial curves in matrix form. A curve of the form

$$\mathbf{x}(t) = \sum_{j=0}^{n} \mathbf{c}_i C_i(t)$$

can be interpreted as a dot product:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{c}_0 & \dots & \mathbf{c}_n \end{bmatrix} \begin{bmatrix} C_0(t) \\ \vdots \\ C_n(t) \end{bmatrix}.$$

One can take this one step further and write

$$\begin{bmatrix} C_0(t) \\ \vdots \\ C_n(t) \end{bmatrix} = \begin{bmatrix} m_{00} & \dots & m_{0n} \\ \vdots & & \vdots \\ m_{n0} & \dots & m_{nn} \end{bmatrix} \begin{bmatrix} t^0 \\ \vdots \\ t^n \end{bmatrix}. \tag{4.25}$$

The matrix $M = \{m_{ij}\}$ describes the basis transformation between the basis polynomials $C_i(t)$ and the *monomial basis* t^i.

If the C_i are Bernstein polynomials, $C_i = B_i^n$, the matrix M has elements

$$m_{ij} = (-1)^{j-i} \binom{n}{j} \binom{j}{i}. \tag{4.26}$$

We list the cubic case explicitly:

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

Why the matrix form? Mathematically, it is equivalent to other curve formulations. When it comes to computer implementations, however, the matrix form may be advantageous if matrix multiplication is hard-wired.

4.7 Problems

1. Show that the Bernstein polynomials B_i^n form a basis for the linear space of all polynomials of degree n.

2. Show that the Bernstein polynomial B_i^n attains its maximum at $t = i/n$. Find the maximum value. What happens for large n?

3. A cusp is a point on a curve where the first derivative vector vanishes. Can a nonplanar cubic Bézier curve have a cusp?