19

Blossoming: A Connect-the-Dots Approach to Splines

by Lyle Ramshaw

June 21, 1987

digital

Systems Research Center
130 Lytton Avenue
Palo Alto, California 94301

Blossoming: A Connect-the-Dots Approach to Splines

Lyle Ramshaw

June 21, 1987

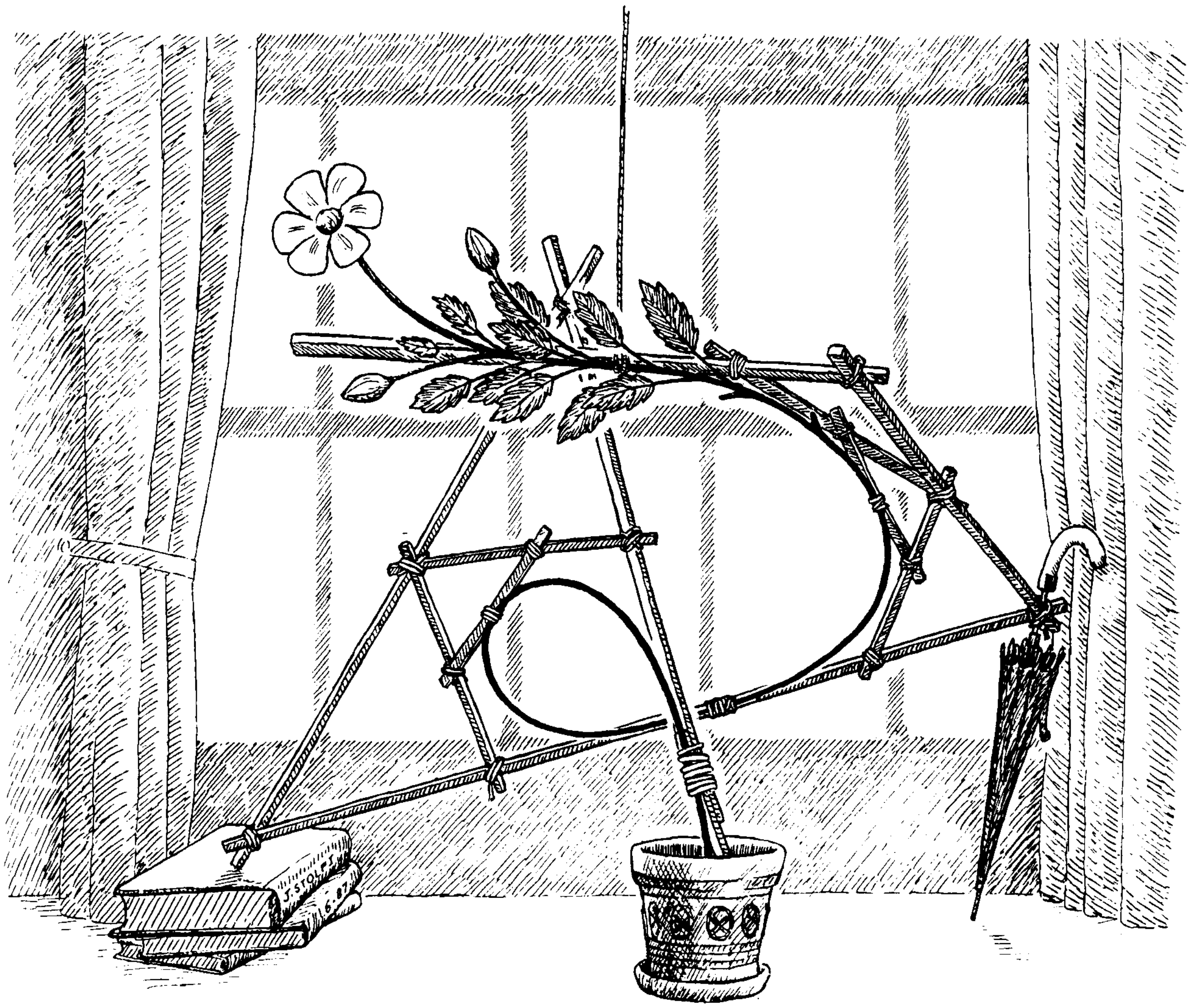


Figure 9.7, on page 56, is reprinted with permission from page 28 of *Catastrophe Theory and its Applications* by Tim Poston and Ian Stewart, published by Pitman of London in 1978.

Author's Abstract

The standard explanations of the theory underlying Bézier and B-spline curves and surfaces aren't as simple as they should be. They have beautiful figures, and those figures clearly illustrate the geometry of the associated algorithms. But the labels on the points in the figures are a disaster! In particular, there is no easy way to tell, from the labels, what geometric relationships hold among the labeled points. This paper proposes a new labeling scheme, based in large part on the work of Paul de Faget de Casteljau.

The key idea behind this new labeling scheme is a classical mathematical principle: a polynomial in a single variable of degree n is essentially equivalent to a symmetric polynomial in n variables that is linear in each variable separately. For example, a quadratic form on a vector space is essentially equivalent to a symmetric bilinear form on that vector space. We christen this principle the *Blossoming Principle*, and we refer to the bilinear form as the *blossom* of the quadratic form. The advantage of blossoming is that bilinear maps are simpler than quadratic maps. When even bilinear maps aren't simple enough, we can use the tensor-product construction to convert the bilinear blossom into a linear blossom, defined on a space of tensors.

The Bézier curves and surfaces that are used in computer-aided geometric design are polynomial functions from a parameter space to an object space. By applying the Blossoming Principle to them, we can transform each such function into its blossom, which is a symmetric, multiaffine function. The blossom of a polynomial curve or surface provides lucid labels both for its Bézier points and for all of the intermediate points that arise during the de Casteljau Algorithm.

Furthermore, as de Casteljau discovered, the blossoming technology extends to handle spline curves with parametric continuity quite neatly. The blossom of such a spline curve provides lucid labels both for its de Boor points and for all of the intermediate points that arise during the de Boor Algorithm. Spline curves with geometric continuity and spline surfaces with triangular patches present unsolved labeling challenges, however.

Capsule Review

A *Bézier curve* is a parametric polynomial curve. For example, the parametric equations $x = 4t^3 + 3t^2 - 6t + 5$ and $y = t^3 - 4t^2 + 2t - 1$ define a cubic Bézier curve in the x-y plane. A *spline curve* is a curve consisting of a sequence of segments of Bézier curves, where consecutive segments are constrained to meet at *joints* with some degree of continuity. Spline curves and the analogous spline surfaces are commonly used in computer aided design to model curved shapes ranging from typographic fonts to automobile bodies.

This paper presents a new perspective on Bézier and spline curves, based on a one-to-one correspondence between univariate polynomials of degree k and symmetric k-variate polynomials of degree 1 in each variable. (For example, the cubic polynomial $F(t) = 4t^3 + 3t^2 - 6t + 5$, which defines the x coordinate as a function of the parameter t in the curve mentioned earlier, is given by $F(t) = f(t, t, t)$, where $f(u, v, w) = 4uvw + uv + uw + vw - 2u - 2v - 2w + 5$.) Using this perspective, the author derives the standard constructions for Béziers and splines in a manner that gives perspicuous labels to the points arising at all the steps of these constructions, and discusses the conditions that govern the degree of continuity achieved at the joints of splines. He also explores a variety of related topics, including characterization of lower-degree Bézier curves as degenerate higher-degree curves, approximation of higher-degree Bézier curves by lower-degree curves, and extensions of the theory from curves to surfaces, from parametric continuity to geometric continuity, and from parametric polynomial functions to parametric rational functions. A number of tantalizing open questions remain concerning these extensions.

The text is intended to be self-contained, requiring no previous study of the theory of splines, and it succeeds in this goal. The key ideas are carefully introduced, examined from multiple points of view, and illustrated with numerous figures and examples. The text does, however, demand of the reader a certain ability to deal comfortably with mathematical abstactions—the quality that my college math professors referred to as "mathematical maturity." Less mathematically inclined readers may prefer to study the author's shorter paper [1] on splines, which presents a subset of the results given here and is less demanding mathematically. The reader who puts forth the necessary effort in studying the present text will be not only rewarded with considerable insight into the theory of Béziers and splines, but also treated to a variety of interesting mathematical sidelights, such as an unusual proof of the theorem that the medians of a triangle trisect each other and a discussion of Zeeman's "umbilic bracelet" (characterizing the set of cubics with three real roots).

Jim Saxe

[1] Lyle Ramshaw. "Béziers and B-splines as multiaffine maps." To appear in *Theoretical Foundations of Computer Graphics and CAD*, the proceedings of a NATO International Advanced Study Institute in Lucca, Italy during July, 1987. Springer-Verlag (1987).

Contents

Part A: Introduction

This is a mathematical paper that presents, in seven parts, a novel approach to the study of spline curves and surfaces. One important benefit of the new approach is a distinctive scheme for labeling the points in diagrams of splines. The primary intent of this introductory Part A is to give the reader a taste of the glories of that labeling scheme.

1. Some intriguing labels

One of the basic problems in computer-aided geometric design is the mathematical modeling of smooth shapes. A popular approach to this problem involves the use of piecewise parametric polynomial functions, which are called ***splines***. Each of the three adjectives "piecewise," "parametric," and "polynomial" reflects a choice of strategy.

The choice of a ***piecewise*** approach means that different regions of the smooth curve or surface being modeled are specified by different systems of equations. This is convenient because it allows for local flexibility: one region of the shape can be adjusted without affecting distant regions. In order to guarantee the smoothness of a piecewise model, appropriate constraints must be placed on adjacent pairs of regions so as to achieve a smooth joint between one region and the next.

Parametric methods are modeling methods that describe a shape in terms of auxiliary parameters. For example, the formula $F(t) := \langle \cos(t), \sin(t) \rangle$ is a parametric description of the unit circle in the plane, with t as the parameter.* The other popular class of modeling methods is ***implicit methods***. The unit circle is described implicitly by the equation $G(x, y) = 1$, where $G(x, y) := x^2 + y^2$. The choice between parametric and implicit methods arises because the primordial objects of mathematics are functions rather than shapes. Suppose that we want to model a p-dimensional shape S sitting in a q-dimensional surrounding space Q. In parametric methods, we invent an auxiliary parameter space P of dimension p and we model the shape S as the range $S := F(P)$ of a function $F: P \to Q$. In implicit methods, on the other hand, we invent an auxiliary space R of dimension $r := q - p$ and we model the shape S as the inverse image $S := G^{-1}(\mathbf{z})$, under some function $G: Q \to R$, of some chosen point $\mathbf{z}$ in R. Both schemes involve specifying the shape S as more than just a set. In parametric methods, each point $\mathbf{x}$ on the shape S has an associated parameter value $F^{-1}(\mathbf{x})$, while, in implicit methods, each point $\mathbf{y}$ not on the shape S has an associated value $G(\mathbf{y})$. Parametric and implicit methods are both useful and important, but this paper is concerned exclusively with parametric methods.

A function $F: P \to Q$ is called a ***polynomial function of degree n*** if each coordinate of the result point $F(\mathbf{u})$ in Q can be expressed as a polynomial of degree n in the coordinates of the argument point $\mathbf{u}$ in P. (Throughout this paper, we shall interpret the phrase "of degree n" to mean "of degree n or less," a concept

* We shall use the relation symbol ":=" in equations that are being used to define the symbol on the left-hand side. By contrast, equations written with the standard equality symbol "=" assert a relationship between quantities defined elsewhere.

that is often described by the phrase "of order $n+1$.") Since polynomials of low degree are easy to evaluate and to differentiate, the class of polyomial functions of degree n is one obvious choice to use when building models of smooth shapes, and it is the one on which we shall focus in this paper. Rational functions of degree n are another reasonable choice, which we shall discuss briefly in Part G. Various classes of transcendental functions could also be used; for example, the function $F(t) := \langle \cos(t), \sin(t) \rangle$ mentioned above is a transcendental parametric model of the unit circle. By the way, the unit circle also has quadratic rational parametric models, such as $F(t) := \langle (1-t^2)/(1+t^2), 2t/(1+t^2) \rangle$. But, as we shall see shortly, the unit circle has no polynomial parametric model of any finite degree.

To reiterate, our overall plan is to model smooth shapes as the ranges of piecewise polynomial functions. For the rest of Section 1, we shall restrict ourselves, for simplicity, to the case of smooth curves, and we shall choose the real numbers $P := \mathbf{R}$ to be the parameter space. That is, we shall build spline curves by assembling segments $F([s,t])$ of n-ic polynomial curves $F: \mathbf{R} \to Q$. We begin by considering a single curve segment in isolation.

What if n, the degree bound, is one? If $F: \mathbf{R} \to Q$ is a polynomial curve of degree one, a segment $F([s,t])$ is a line segment. The obvious way to specify such a segment $F([s,t])$ is by giving its endpoints $F(s)$ and $F(t)$ as points in the object space Q. Every choice of two endpoints determines a corresponding function F uniquely, and in a geometrically intuitive and numerically stable manner. (If it happens that $F(s) = F(t)$, the associated function F will be of degree zero, that is, constant, as well as being of degree one.) If we know the endpoints $F(s)$ and $F(t)$, we can compute any other point $F(u)$ for u in $[s,t]$ by performing a linear interpolation.

The parametric polynomial curves $F: \mathbf{R} \to Q$ of degree two are more interesting. First, since the acceleration vector $F''(u)$ is constant, we observe that the curve F must lie in a plane in Q. Everyone knows that the plane curves that can be described implicitly by a polynomial of degree two are precisely the conic sections. Indeed, any conic section lying in any plane in the object space Q can be described implicitly by using $q-2$ linear polynomials to determine the plane and then one quadratic polynomial to determine the conic. But we are interested in parametric modeling methods, and it turns out that only parabolas can be parameterized with quadratic polynomials. If each coordinate of the point $F(u)$ is given by a polynomial in u, then the curve F must go off to infinity in precisely one direction (or its opposite) as $u \to \pm\infty$. Thus, ellipses are out because they don't go to infinity at all, while hyperbolas are out because they go to infinity in more than one direction.

What is a good way to specify an arc $F([s,t])$ of a parabola $F: \mathbf{R} \to Q$? Fig. 1.1 shows one technique, in which the segment $F([0,1])$ is determined by specifying three points in Q, called the ***Bézier points*** of the parabolic arc $F([0,1])$: the starting point $F(0)$, the point where the starting and ending tangents intersect, and the ending point $F(1)$. In Fig. 1.1, these three Bézier points are labeled $f(0,0)$, $f(0,1)$, and $f(1,1)$. If we want to compute the point $F(t)$ for t in $[0,1]$ from the three Bézier points, we can do so by performing three linear interpolations as shown. First, we interpolate between $f(0,0)$ and $f(0,1)$ to find $f(0,t)$. Second, we interpolate

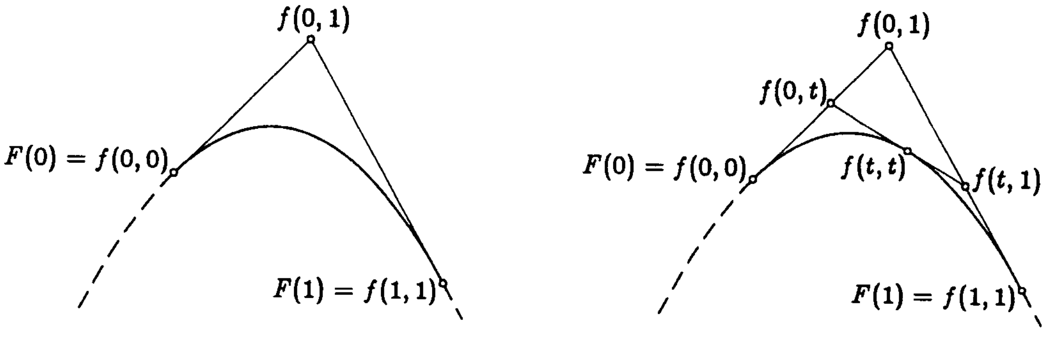


Fig. 1.1. The de Casteljau Algorithm for a quadratic Bézier curve

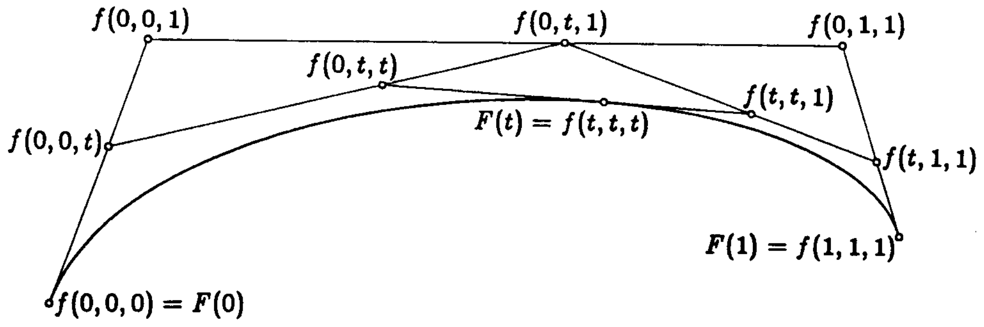


Fig. 1.2. The de Casteljau Algorithm for a cubic Bézier curve

between $f(0,1)$ and $f(1,1)$ to find $f(t,1)$. Finally, we interpolate between $f(0,t)$ and $f(t,1)$ to find $F(t) = f(t,t)$. This algorithm for computing $F(t)$ from the Bézier points of the segment $F([0,1])$ is called the ***de Casteljau Algorithm.*** Bézier points and the de Casteljau Algorithm are standard material; Böhm, Farin, and Kahmann have written a fine survey of this area [5]. The function $f: \mathbf{R}^2 \to Q$ appearing in the labels in Fig 1.1 is not standard, however; it is our first example of a ***blossom.*** In the quadratic case, we can think of the blossom value $f(u,v)$ as the point of intersection of the tangent lines to the parabola at $F(u)$ and $F(v)$.

Fig. 1.2 shows a segment $F([0,1])$ of a polynomial cubic curve. The four points labeled $f(0,0,0)$, $f(0,0,1)$, $f(0,1,1)$, and $f(1,1,1)$ are the four Bézier points of this cubic segment. (The cubic F shown in Fig. 1.2 is degenerate in the sense that it lies in a plane; a nondegenerate polynomial cubic is a twisted space curve, and the four Bézier points of a segment of such a cubic would not be coplanar.) The remaining points and lines are the result of using the de Casteljau Algorithm to compute the point $F(t) = f(t,t,t)$ by doing six linear interpolations, starting with the Bézier points. The function $f(u,v,w)$ appearing in the labels is the blossom of F. This blossom $f: \mathbf{R}^3 \to Q$ is a symmetric function of three real arguments; that is, the value $f(u,v,w)$ doesn't depend on the order of u, v, and w. Furthermore, the blossom f is related to the curve F by the identity $F(u) = f(u,u,u)$. Note that the incidence structure of the points and lines in Fig. 1.2 is reflected in the labels: two

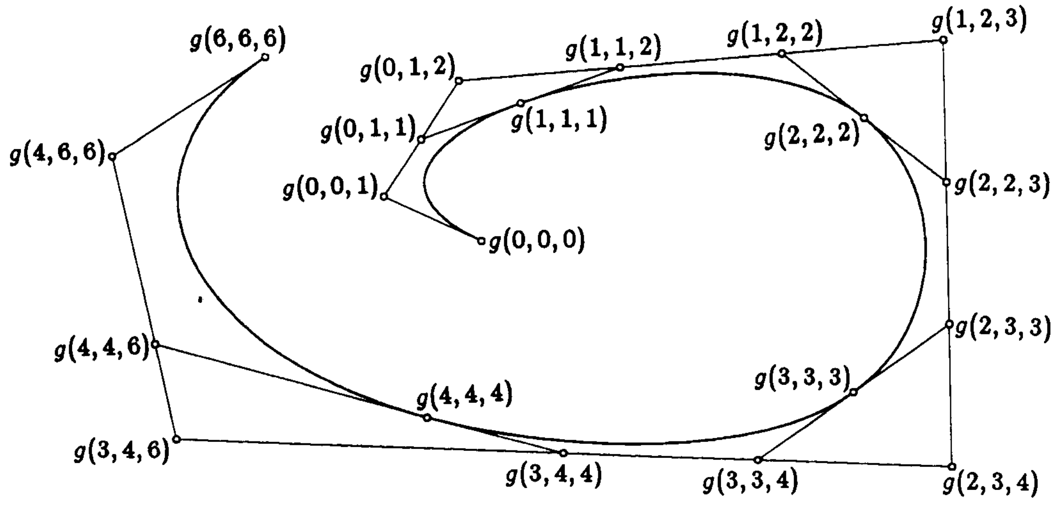


Fig. 1.3. The Böhm-Sablonnière Rules for a cubic spline curve

points lie on a common line in Fig. 1.2 if and only if two out of the three arguments in their labels are the same. The distance ratios are also implied by the labels: for example, the point $f(t,t,1)$ lies t of the way from $f(0,t,1)$ to $f(1,t,1)$. (Symmetry implies that $f(1,t,1) = f(t,1,1)$.) More generally, whenever we vary one of the arguments of f while holding the other two fixed, the value of f varies along a straight line at a constant rate—in fancier language, the function f is *triaffine*.

In fact, the labels in Fig. 1.2 have so many nice properties that they capture all of the structural information in the diagram. Suppose that we were told that f was a symmetric, triaffine function and that the four outer vertices in Fig. 1.2 were the points $f(0,0,0)$, $f(0,0,1)$, $f(0,1,1)$, and $f(1,1,1)$. And suppose that we wished to compute $f(t,t,t)$. We would be able to reconstruct Fig. 1.2 from this information, even if we had never heard of Bézier points or the de Casteljau Algorithm. We would linearly interpolate between the four outer points, taken in pairs, to compute the three points that have one t in their labels. A second stage of interpolations, taking these three points in pairs, would give us the two points $f(0,t,t)$ and $f(t,t,1)$. A third stage would use a single interpolation would give us the desired point $f(t,t,t) = F(t)$.

Labels this good are worthy of study. Our first major goal in this paper is to construct the blossom f of an n-ic polynomial curve or surface and to explore the connection between this blossom and the Bézier-de Casteljau theory of polynomial curves and surfaces. The other major goal is to investigate the extent to which the same ideas can be extended to parametric curves or surfaces with more than one polynomial piece, that is, to splines.

Fig. 1.3 shows a spline curve G defined on the interval $[0,6]$ and composed of five pieces. The curve $G(u)$ follows one polynomial cubic for u in $[0,1]$, then switches over to a different polynomial cubic for u in $[1,2]$, and so on for $[2,3]$, $[3,4]$, and $[4,6]$. (The fifth parameter interval is twice as long as the rest; in general, parameter intervals can have arbitrary lengths.) The joints between each adjacent pair of cubics

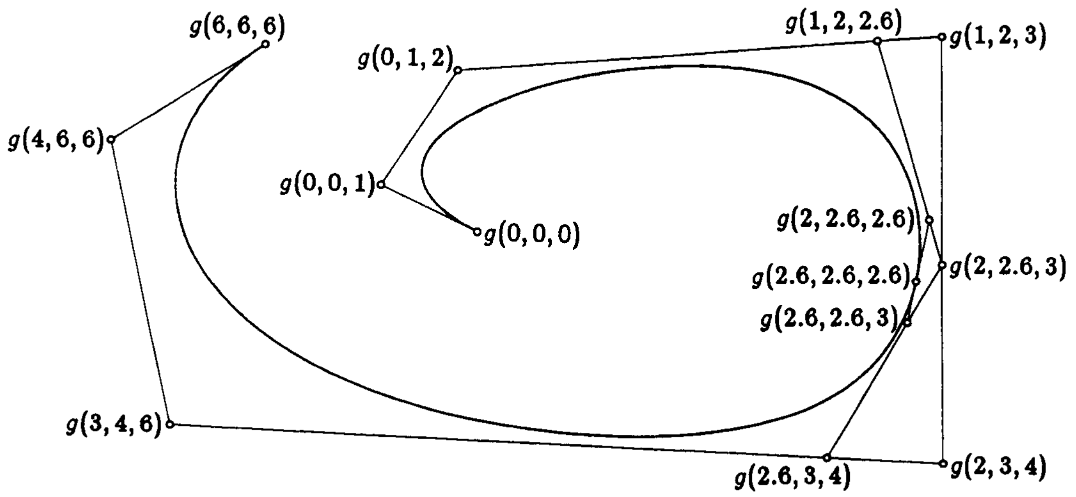


Fig. 1.4. The de Boor Algorithm for a cubic spline curve

have C^2 continuity; that is, no jump is allowed in position, velocity, or acceleration, although a jump in jerk is permitted. (*Jerk* is a name from the automobile industry for the third derivative of position with respect to time.) But the behavior at the endpoints $u = 0$ and $u = 6$ is entirely unconstrained. This state of affairs is described in the trade by saying that the *knot sequence* of G is $(0,0,0,0,1,2,3,4,6,6,6,6)$. As before, the labeling function $g(u,v,w)$ is a symmetric function of three arguments, related to the spline G by the identity $G(u) = g(u,u,u)$. Furthermore, the function g behaves triaffinely. For example, as u varies from 2 to 6, the point $g(u,3,4)$ moves at a constant rate along the line segment joining $g(2,3,4)$ to $g(6,3,4)$. The outermost points in this diagram have labels of the form $g(r,s,t)$ where (r,s,t) is a triple of consecutive knots. They are precisely the *de Boor points* by which the spline G is controlled in the standard theory [5]. Furthermore, for each pair (s,t) of consecutive knots, the Bézier points of the cubic polynomial segment $G([s,t])$ are the points $g(s,s,s)$, $g(s,s,t)$, $g(s,t,t)$, and $g(t,t,t)$. Thus, the multiaffine labels help to clarify the relationship between the de Boor points of a spline curve and the Bézier points of its segments, a relationship first explored by Böhm [3] and Sablonnière [37]. (The initial and final knots 0 and 6 are repeated four times, instead of only three, because even the position of $G(u)$ is allowed to jump discontinuously as u moves through 0 or 6. This four-fold repetition means that there are two different de Boor points labeled $g(0,0,0)$; the first of them gives $\lim_{u\uparrow 0} G(u)$, while the second gives $\lim_{u\downarrow 0} G(u)$. Only the second is shown in Fig. 1.3.)

A word of warning: since $G(u)$ is a cubic spline curve rather than a single polynomial cubic, the function g in Fig. 1.3 can't be as simple as the f in Fig. 1.2. The extra complexity shows up in the fact that $g(u,v,w)$ is well-defined only when its three arguments are fairly close together—in particular, when they don't skip over any knots. For example, the value $g(u,3,4)$ is well-defined only for u between 2 and 6.

Fig. 1.4 shows the same spline curve G as in Fig. 1.3, but with the auxiliary points and lines that we would draw to compute $G(2.6) = g(2.6, 2.6, 2.6)$. Starting from the de Boor points, we construct new points that have more and more copies of 2.6 among their g-arguments by performing linear interpolations between previously constructed points. This algorithm for evaluating a spline curve is called the *de Boor Algorithm* [5].* With these labels on the points, the de Boor Algorithm seems like a fairly obvious extension of the de Casteljau Algorithm. Indeed, one of the best reasons to use labels based on blossoms is the clarification that they bring to parametrically continuous spline curves.

Where did blossoming come from, and who deserves the credit? We shall close this introduction by considering those historical questions briefly.

As applied to a single polynomial function, blossoming is a standard mathematical tool in every respect except for the name "blossoming," which I coined for this paper. For example, mathematicians have known for some time that there is a unique symmetric, bilinear form associated with any quadratic form on a vector space [28, 32]. That theorem is precisely the homogeneous, degree two case of blossoming. Furthermore, there is nothing tricky about transferring that result to the nonhomogeneous, degree n case, which is the case that is needed to transform a polynomial curve such as the F of Figs. 1.1 or 1.2 into its multiaffine blossom f.

Paul de Faget de Casteljau was the first to apply blossoming to the polynomial curves and surfaces used in computer-aided geometric design. He also gets the credit for discovering that the blossoming technology extends very neatly to parametrically continuous spline curves, as described in his 1984 notes entitled "Formes à Pôles," that is, shapes specified by poles [14]. A *pole*, in de Casteljau's terminology, is a point such as $f(0,0,1)$ or $g(2,3,4)$, that is, a value of a blossom. Someone familiar with de Casteljau's work wouldn't have found anything new in this introduction, except for the terminology and notation that I developed while independently stumbling upon the same ideas. But many of the techniques and results in the subsequent sections, such as the use of the symmetric variant of the tensor-product construction, appear for the first time in these pages (to the best of my knowledge).

Carl de Boor and Klaus Höllig have been approaching blossoming in their recent work on B-splines. In the traditional theory, B-splines are defined by means of a formula involving divided differences. From this formula, one proves a recurrence relation. And from the recurrence relation, one derives the de Boor Algorithm. In their paper "B-Splines Without Divided Differences," de Boor and Höllig showed that one can take the recurrence relation as basic [13]. In this paper, we shall explore the remaining possibility, taking the de Boor Algorithm as basic and deriving the recurrence relation and the divided-difference formula from it. One consequence of this change of perspective is that primal and dual are reversed. In the work of de Boor and Höllig, B-splines are primal objects, and their duals are called the *dual functionals*, written $\lambda_{i,n+1}$. In this paper, the same objects that de Boor and Höllig refer to as dual functionals appear as a primal objects, in particular,

* Comparing our Fig. 1.4 with the corresponding Fig. 18 in the survey paper [5], it is amazing how well their figure is able to communicate the de Boor Algorithm under the severe handicap that its points are just labeled "d_i^j".

as the n-contravariant tensors $\lambda_{i,n+1} = \bar{t}_{i+1} \cdots \bar{t}_{i+n}$. B-splines are the duals of these n-contravariant tensors, that is, they are n-covariant tensors, as described in Remark 18.13.

In a fuzzier sense, the large body of work on algorithms for knot insertion was also moving in the general direction of blossoming. A spline curve, like the $G(u)$ in Fig. 1.4, is a function; the de Boor Algorithm provides a way to evaluate that function, that is, to evaluate blossom values of the form $g(u, u, u)$, whose arguments are all equal. Knot-insertion algorithms took a step closer to blossoming by widening the scope of discourse to include blossom values $g(u, v, w)$ whose arguments are not equal. In particular, one way to compute the blossom value $g(u, v, w)$ is to use a knot-insertion algorithm to insert u, v, and w as knots (assuming that u, v, and w are close enough together to make $g(u, v, w)$ well-defined). On the other hand, knot-insertion algorithms also took a step away from blossoming by working with sequences of knots and sequences of control points, hence foolishly and needlessly giving up on the concept of the spline as a function. Blossoming is what results when we take the spline $G(u) = g(u, u, u)$ and allow ourselves to evaluate it, as a function, on n-tuples of parameter values (u, v, w) whose components are not all equal.

Part B: The Three Principles

The theory behind the intriguing labels of Section 1 depends upon three classical mathematical principles, which we shall review and customize for our purposes in this part. The Blossoming Principle is the first of these, fairly easy to understand and yet very powerful. The Tensoring Principle comes next; it is technically more sophisticated, and its payoff is not as dramatic. But, once we have invested the necessary effort, it will reward us with valuable algebraic insights into polynomial curves and surfaces. Third is the Homogenizing Principle, which is both simple and familiar; it comes into play when differentiation is involved. While investigating these three principles, we shall also study the relationships between them and the standard techniques that are used for controlling polynomial curves and surfaces in computer-aided geometric design.

2. The Blossoming Principle

The Blossoming Principle is the core idea behind these new labelings. It allows us to replace a function of high degree in one variable by an equivalent function of degree one but with lots of variables.

Consider the cubic polynomial $F(t) = 7t^3 + 6t^2 - 3t + 5$. Suppose that we want to construct a trivariate polynomial $f(u, v, w)$ of the form

$$f(u, v, w) = r_1uvw + r_2uv + r_3uw + r_4vw + r_5u + r_6v + r_7w + r_8$$

that satisfies the identity $f(t, t, t) = F(t)$; in other words, we want the main diagonal of $f(u, v, w)$ to agree with F. To achieve this correspondence, we must have $r_1 = 7$, $r_2 + r_3 + r_4 = 6$, $r_5 + r_6 + r_7 = -3$, and $r_8 = 5$. We can determine f uniquely if we also demand that $f(u, v, w)$ be a symmetric function of its three arguments. This symmetry condition forces us to make $r_2 = r_3 = r_4$ and $r_5 = r_6 = r_7$, resulting in the unique choice

$$f(u, v, w) := 7uvw + 2uv + 2uw + 2vw - u - v - w + 5.$$

This argument can be generalized quite a bit; but first, a question of nomenclature. What shall we call a polynomial, such as the f above, that has degree one in each of its variables separately? The name "multilinear" pops to mind, but there is the problem that the word "linear" often implies "homogeneous" as well as "of degree one." (A polynomial is ***homogeneous of degree n*** when all of its terms have total degree precisely n; in particular, f would have to have $r_2 = r_3 = \cdots = r_8 = 0$ in order to be multilinear in the homogeneous sense.) To avoid confusion, we shall use the word "affine" instead of "linear" when we mean "of degree one, but not necessarily homogeneous." In particular, we shall say that f is ***multiaffine***.

In fact, there are two parallel worlds in which mathematics of this flavor can be done: the linear world and the affine world. The linear world consists of linear spaces (also called vector spaces) and homogeneous linear maps between them. The natural notion of a polynomial map in this world demands that the polynomials involved also be homogeneous. The less familiar affine world consists of affine spaces and affine maps between them. An affine space is like a linear space except that no point has been distinguished as the origin. An affine map is of degree one, but it is not necessarily homogeneous; indeed, since affine spaces don't have distinguished origins, it doesn't really make sense to ask whether an affine map is homogeneous or not. Similarly, in the affine world, the natural notion of polynomial maps does not demand that they be homogeneous. In this paper, we shall focus on the affine world more than most mathematicians typically do, because it is the home of the parametric polynomial curves and surfaces that are used in computer-aided geometric design. If $F: P \to Q$ is such a curve or surface, note that neither the parameter space P nor the object space Q has a distinguished origin, and note that the polynomials involved in defining F are generally not homogeneous.

With this nomenclature, we can describe the phenomenon exemplified by F and f quite concisely: if $F(t)$ is any cubic polynomial, there exists a unique symmetric triaffine polynomial $f(u, v, w)$ that agrees with F on the diagonal, that is, that satisfies the identity $f(t, t, t) = F(t)$. Going backwards from f to F is even easier: if $f(u, v, w)$ is triaffine, the formula $F(t) := f(t, t, t)$ always defines a cubic polynomial F, whether or not f is symmetric, and without any need to manipulate coefficients. Thus, univariate cubic polynomials and symmetric triaffine polynomials are equivalent concepts; they are just different forms of the same underlying mathematical object. That doesn't mean, however, that they are equally easy to work with in specific situations. We shall find that, in many cases, the symmetric triaffine version reveals the underlying structure more clearly than does the cubic version of the same abstract entity. Based on the intuition that the structure bound up in F is unrolled and revealed in f, we shall refer to the process of deriving f from F as *blossoming*, and we shall refer to f as the *multiaffine blossom* of F. (You may or may not like the name "blossoming," but I certainly feel that some name should be chosen. Mathematicians seem mysteriously content to refer to f as "the symmetric triaffine function that corresponds to F," where the sense of the correspondence must be inferred from the context. I prefer the phrase "the blossom of F.")

Any function defined by polynomials can be blossomed. For computer graphics applications, we shall extend our blossoming expertise in two ways. First, we shall generalize from degree three to degree n. Second, we shall deal with functions from one multidimensional affine space to another, rather than with functions from the reals to the reals. We begin by stating some definitions.

Let P be a parameter space and Q be an object space, both affine spaces over the real numbers $\mathbf{R}$. Addition and scalar multiplication aren't defined as separate operations in an affine space. Instead, there is a single operation of "taking an affine combination": given two points $\mathbf{u}$ and $\mathbf{v}$ in an affine space and any real number r, there is a well-defined point $(1-r)\mathbf{u}+r\,\mathbf{v}$, which we can think of as "r of the way from $\mathbf{u}$ to $\mathbf{v}$." A function $F: P \to Q$ is called *affine* if it preserves affine combinations, that is, if the identity $F\bigl((1-r)\mathbf{u}+r\mathbf{v}\bigr) = (1-r)F(\mathbf{u})+rF(\mathbf{v})$ holds. We shall call a function $f: P^n \to Q$ of n arguments *multiaffine* or *n-affine* if it is an affine function of each argument when the others are held fixed.

The concept of affineness makes perfect sense even for infinite-dimensional spaces P and Q, but we won't need that generality in this paper. In the finite-dimensional case, affine maps are precisely those that can be represented by polynomials of degree one. (Following our convention, the phrase "of degree one" here means "of degree one or less.") For example, if we choose Cartesian coordinate systems for the finite-dimensional affine spaces P and Q, a map $F: P \to Q$ is affine if and only if each coordinate of the result point $F(\mathbf{u})$ can be written as a polynomial of degree one in the coordinates of the argument point $\mathbf{u}$. For a multiaffine function f, the total degree of the polynomial that represents a coordinate of the point $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ might be as high as n; but each term of that polynomial will include at most one of the coordinates of any particular argument $\mathbf{u}_i$. For example, if $f: P^2 \to Q$ is a biaffine function whose arguments lie in the plane P of points $\mathbf{u} = \langle u, v \rangle$, then the polynomials that give the coordinates of $f(\mathbf{u}_1, \mathbf{u}_2) = f(\langle u_1, v_1 \rangle, \langle u_2, v_2 \rangle)$ could include terms of the form au_1u_2 or bu_1v_2. But a term of the form cu_1v_1 would imply non-affine dependence of $f(\mathbf{u}_1, \mathbf{u}_2)$ on $\mathbf{u}_1$.

If P and Q are finite-dimensional affine spaces, a function $F: P \to Q$ is called ***polynomial of degree n*** if, when we choose Cartesian coordinate systems for P and Q, each coordinate of the point $F(\mathbf{u})$ can be written as a polynomial of degree n in the coordinates of the argument point $\mathbf{u}$. This condition is independent of the choice of the coordinate systems in P and Q. The special names ***polynomial curve*** and ***polynomial surface*** are used for polynomial functions $F: P \to Q$ whose domains P have dimensions one and two, respectively.

Proposition 2.1: The Blossoming Principle (affine variant). *If P and Q are finite-dimensional affine spaces, then polynomial functions $F: P \to Q$ of degree n are equivalent to symmetric, n-affine maps $f: P^n \to Q$. In particular, given a function of either type, a unique function of the other type exists that satisfies the identity $F(\mathbf{u}) = f(\mathbf{u}, \mathbf{u}, \ldots, \mathbf{u})$. We shall call f the* multiaffine blossom *of F and F the* diagonal *of f.*

Proof. Since there is no interaction between the different coordinates of the points in the object space Q, it suffices to consider each coordinate separately, or equivalently, to consider the case in which Q is the real numbers $\mathbf{R}$.

Let p be the dimensionality of the parameter space P. Choose a Cartesian coordinate system for P, and let $\mathbf{u} = \langle u^1, \ldots, u^p \rangle$ denote the coordinates of the point $\mathbf{u}$ in P in this coordinate system. We are using superscripts to enumerate the p dimensions of P in order to leave subscripts free to enumerate the n arguments of the blossom f.

By our definitions, the map $F: P \to \mathbf{R}$ is a polynomial function of degree n if and only if the quantity $F(\mathbf{u})$ is given by a polynomial of total degree no greater than n in the variables u^1 through u^p. A term of this polynomial has the form

$$T_F := r(u^1)^{k_1}(u^2)^{k_2} \cdots (u^p)^{k_p}$$

for some real coefficient r and for some nonnegative integer exponents k_j satisfying $\sum_{j=1}^{p} k_j \leq n$. Similarly, the function $f: P^n \to \mathbf{R}$ is n-affine if and only if we can express the value $f(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ as a polynomial in the np variables v_i^j whose general term has the form

$$T_f := r \prod_{i=1}^{n} \prod_{j=1}^{p} (v_i^j)^{e_{ij}}$$

where each e_{ij} is either 0 or 1 and we have $\sum_j e_{ij} \leq 1$ for each fixed i.

Suppose that we are given F and we want to construct f satisfying $f(\mathbf{u},\ldots,\mathbf{u}) = F(\mathbf{u})$. Note that the term T_f will contribute to T_F when we substitute $\mathbf{u}$ for each of the $\mathbf{v}_i$ if and only if $\sum_{i=1}^{n} e_{ij} = k_j$ for each j. If we let l denote the difference $l := n - \sum_j k_j$, the number of terms T_f that contribute to the particular term T_F is given by the multinomial coefficient $\binom{n}{k_1 \ldots k_p\, l}$. If we want f to be symmetric, our only choice is to split up each term T_F into $\binom{n}{k_1 \ldots k_p\, l}$ symmetric terms T_f in f with equal coefficients.

Going the other way, suppose that we are given f, whose terms T_f satisfy $\sum_j e_{ij} \leq 1$ by assumption. When we substitute $\mathbf{u}$ for each of the $\mathbf{v}_i$, the total degree of the result will be $\sum_i \sum_j e_{ij} \leq n$. Hence, the identity $F(\mathbf{u}) := f(\mathbf{u},\ldots,\mathbf{u})$ will define a unique polynomial function of degree n or less. □

Corollary 2.2. *If P and Q are finite-dimensional affine spaces, a symmetric n-affine map $f\colon P^n \to Q$ is completely determined by its values on the main diagonal of P^n, that is, by the values of the form $f(\mathbf{u},\mathbf{u},\ldots,\mathbf{u})$.* □

The Blossoming Principle has a linear-world variant also: starting with a polynomial function F that is homogeneous gives a blossom f that is multilinear instead of just multiaffine. The only difference in the proof is that we always have $l = 0$ in the homogeneous case. The particular case $n = 2$ and $Q = \mathbf{R}$ of the linear variant is quite a famous theorem: it states that quadratic forms on a vector space are equivalent to symmetric bilinear forms [28, 32].

Our proof of the Blossoming Principle worked with explicit coordinate systems throughout. There are proofs that depend less on explicit coordinates, but they tend to obscure the essential simplicity of the Blossoming Principle. For example, consider a quadratic polynomial function $F(\mathbf{u})$ in the affine world. The associated biaffine blossom $f(\mathbf{u}_1,\mathbf{u}_2)$ can be defined without any reference to coordinates by means of the formula

$$f(\mathbf{u}_1,\mathbf{u}_2) = \frac{4F\left(\frac{\mathbf{u}_1+\mathbf{u}_2}{2}\right) - F(\mathbf{u}_1) - F(\mathbf{u}_2)}{2},$$

which is called the ***polarization identity*** [19, 29, 33]. For general degree n, the analog of this formula, which we shall prove in Section 4, has an inclusion-exclusion structure:

$$f(\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n) = \frac{1}{n!} \sum_{\substack{S\subseteq\{1,2,\ldots,n\}\\ k=|S|}} (-)^{n-k} k^n F\left(\frac{1}{k}\sum_{i\in S} \mathbf{u}_i\right). \qquad (2.3)$$

Two comments about this formula before we proceed. First, the corresponding formula in the linear world can be made a little simpler, because the factor of k^n outside F cancels against the $1/k$ in the argument when F is known to be homogeneous of degree n. Second, it may seem a bit dicey to allow $S = \emptyset$ in the summation, since the argument of F in the $S = \emptyset$ term is the indeterminate form

0/0. When $n > 0$, the indeterminacy in the argument of F doesn't matter, because that value of F is multiplied by $0^n = 0$. In the special case $n = 0$, the indeterminacy doesn't matter for a different reason: F is constant. Indeed, if we tried to avoid the indeterminacy by explicitly demanding $S \neq \emptyset$ in the summation, we would cause the formula to fail for the case $n = 0$.

While Eq. (2.3) is coordinate-free, it is rather complicated. Furthermore, a proof of the Blossoming Principle based on this formula still has to chose a coordinate system for P in order to prove that the right-hand side does indeed define a multiaffine map. In fact, there is no hope of our avoiding coordinates entirely, since our definition of a polynomial function involved coordinates. More abstract texts manage to avoid coordinates by turning the Blossoming Principle from a theorem into a definition: they define a "polynomial function" to be the diagonal of a multiaffine function. The argument above is then used to show that, in the finite-dimensional case, these "polynomial functions" are precisely the functions that are given by polynomials when expanded in coordinates.

Restriction 2.4. From now on, all affine and linear spaces mentioned in this paper are implicitly assumed to be finite-dimensional. The exploration of the infinite-dimensional situation is left as a challenge for the reader (and for the author).

The contrast between our proof of Prop. 2.1, which was based on choosing coordinates, and the coordinate-free Eq. (2.3) is a phenomenon that will come up again in Part D. One way to compute a blossom is to choose coordinate systems for all of the spaces involved, and then manipulate coefficients. This technique involves making lots of choices and defining lots of symbols. But, if the right coordinate systems are chosen, the actual arithmetic involved tends to be quite simple: just multiplying or dividing each coefficient by a factorial or two. We shall refer to this technique as *coordinate-based.* The alternative *coordinate-free* technique searches for a single formula that expresses an arbitrary value of the desired blossom as an affine combination of values of something known. The formulas that result from the coordinate-free technique are conceptually simpler, in the sense that they give the whole solution in one equation. But the combinatorics and arithmetic involved in them tend to be much more sophisticated. In the instance we have just seen, the coordinate-free approach may also seem more expensive, since the sum in Eq. (2.3) involves 2^n terms. On the other hand, note that the number of terms is independent of $p = \dim(P)$; thus, the same coordinate-free formula works for curves, for surfaces, or for $p \geq 3$.

3. The blossoms of Béziers

Parametric polynomial curves and surfaces are basic objects in computer-aided geometric design, and the affine variant of the Blossoming Principle applies to them directly. In this section, we shall relate blossoms to the Bézier-de Casteljau theory of curves and surfaces.

Before starting on curves, which are the easiest case, we shall pause to institute a new notational convention. A polynomial curve is a polynomial function $F: L \to Q$ whose domain L is a one-dimensional affine space, that is, an affine line. So far, we have been ignoring the distinction between a point of the domain space L and a real number. For example, in Section 1, we wrote $F(1)$ where we meant, more precisely, "the result of applying F to the point in L whose coordinate, in some chosen coordinate system for L, is the real scalar 1." While it might seem pedantic to object to this practice, it turns out that a little pedantry now will pay off when we consider tensors in Section 4. In general, if P is an affine space for which we have chosen some Cartesian coordinate system, we shall use angle brackets to denote a point $\mathbf{u}$ in P given in terms of coordinates $\mathbf{u} = \langle u^1, \ldots, u^p \rangle$. For the particular case of the affine line L, let us adopt the abbreviation $\bar{t} := \langle t \rangle$ for the point in L whose coordinate is the scalar t. Under this convention, the expression $2 + 3$ denotes the scalar 5; the expression $(\bar{2}+\bar{3})/2$ denotes the point in L midway between the points $\bar{2}$ and $\bar{3}$, which can also be written $\overline{2.5}$; and the expression $\bar{2}+\bar{3}$ is not well-formed, since it denotes a non-affine combination of points. To make this convention retroactive, we would have to go back and add lots of bars to Section 1. In particular, each argument to F, f, or g in the labels of Figs. 1.1 through 1.4 should have a bar over it.

Let $F: L \to Q$ be a polynomial curve of degree n, and let $f: L^n \to Q$ be its blossom; that is, f is the unique symmetric, n-affine map that satisfies $F(\bar{u}) = f(\bar{u}, \bar{u}, \ldots, \bar{u})$, where $\bar{u}$ denotes an arbitrary point in L. Suppose that we choose a reference segment $[\bar{s}, \bar{t}] \subset L$, and consider the value $f(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n)$. We can write the first argument point $\bar{u}_1$ of f as an affine combination of the points $\bar{s}$ and $\bar{t}$ as follows:

$$\bar{u}_1 = \left(\frac{t - u_1}{t - s}\right) \bar{s} + \left(\frac{u_1 - s}{t - s}\right) \bar{t}.$$

(This formula simplifies somewhat in the particular case of the reference segment $[\bar{0}, \bar{1}]$; we get $\bar{u}_1 = (1 - u_1)\bar{0} + u_1 \bar{1}$. But it goes against the grain of the affine approach to single out any particular reference segment.) Since f is affine in its first argument, it preserves affine combinations; therefore,

$$f(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n) = \left(\frac{t - u_1}{t - s}\right) f(\bar{s}, \bar{u}_2, \ldots, \bar{u}_n) + \left(\frac{u_1 - s}{t - s}\right) f(\bar{t}, \bar{u}_2, \ldots, \bar{u}_n).$$

The next step is conceptually straightforward, although notationally somewhat clumsy: we want to simultaneously expand all n of the argument points $\bar{u}_1$ through $\bar{u}_n$ as affine combinations of $\bar{s}$ and $\bar{t}$. The resulting sum has 2^n terms:

$$f(\bar{u}_1, \ldots, \bar{u}_n) = \sum_{\substack{I \cap J = \emptyset \\ I \cup J = \{1,2,\ldots,n\}}} \prod_{i \in I} \left(\frac{t - u_i}{t - s}\right) \prod_{j \in J} \left(\frac{u_j - s}{t - s}\right) f(\underbrace{\bar{s}, \ldots, \bar{s}}_{|I|}, \underbrace{\bar{t}, \ldots, \bar{t}}_{|J|}). \tag{3.1}$$

Note that we have taken advantage of the symmetry of f in order to put all of the $\bar{s}$'s before all of the $\bar{t}$'s in the arguments to f. The interesting thing about Eq. (3.1) is that it expresses every value of f in terms of just the $n+1$ special values whose arguments consist entirely of $\bar{s}$'s and $\bar{t}$'s. Thus, a symmetric n-affine function $f: L^n \to Q$ is completely determined once we know the particular values $f(\bar{s}, \ldots, \bar{s})$,

$f(\bar{s},\ldots,\bar{s},\bar{t}),\ \ldots,\ f(\bar{t},\ldots,\bar{t})$ as points in Q. Furthermore, there are no constraints on these special values: if $\mathbf{x}_0$ through $\mathbf{x}_n$ are any $n+1$ points in Q, the formula

$$f(\bar{u}_1,\ldots,\bar{u}_n) := \sum_{\substack{I\cap J=\emptyset \\ I\cup J=\{1,2,\ldots,n\}}} \prod_{i\in I}\left(\frac{t-u_i}{t-s}\right)\prod_{j\in J}\left(\frac{u_j-s}{t-s}\right)\mathbf{x}_{|J|}$$

defines a symmetric n-affine function $f\colon L^n \to Q$ that satisfies

$$f(\underbrace{\bar{s},\ldots,\bar{s}}_{n-k},\underbrace{\bar{t},\ldots,\bar{t}}_{k}) = \mathbf{x}_k.$$

This proves the following.

Proposition 3.2. *Let $[\bar{s},\bar{t}]$ be a reference segment in the affine line L. The formula*

$$f(\underbrace{\bar{s},\ldots,\bar{s}}_{n-k},\underbrace{\bar{t},\ldots,\bar{t}}_{k}) = \mathbf{x}_k$$

for k in $[0,n]$ provides a one-to-one correspondence between symmetric, multiaffine functions $f\colon L^n \to Q$ and $(n+1)$-tuples $(\mathbf{x}_0,\ldots,\mathbf{x}_n)$ of points in Q. □

Proposition 3.3: Bézier curve segments. *The formula in Prop. 3.2 also provides a one-to-one correspondence between polynomial curves $F\colon L\to Q$ of degree n and $(n+1)$-tuples $(\mathbf{x}_0,\ldots,\mathbf{x}_n)$ of points in Q, where the f in Prop. 3.2 denotes the blossom of F. Moreover, the points $\mathbf{x}_k$ are precisely the Bézier points by which the curve segment $F([\bar{s},\bar{t}])$ is controlled in the standard theory.*

Proof. The first claim follows at once from Prop. 3.2 and the Blossoming Principle. As for the connection with the Bézier theory, the Bézier n-ic curve $F(u)$ is most often defined by using the Bernstein basis polynomials [5] to blend together the Bézier points $\mathbf{x}_0$ through $\mathbf{x}_n$:

$$F(u) = \sum_{0\le k\le n}\binom{n}{k}\left(\frac{t-u}{t-s}\right)^{n-k}\left(\frac{u-s}{t-s}\right)^{k}\mathbf{x}_k.$$

In our theory, the curve $F(\bar{u})$ is the diagonal $f(\bar{u},\ldots,\bar{u})$ of its blossom. If we substitute u for each of the u_i in Eq. (3.1), we find that

$$F(\bar{u}) = f(\bar{u},\ldots,\bar{u}) = \sum_{0\le k\le n}\binom{n}{k}\left(\frac{t-u}{t-s}\right)^{n-k}\left(\frac{u-s}{t-s}\right)^{k} f(\underbrace{\bar{s},\ldots,\bar{s}}_{n-k},\underbrace{\bar{t},\ldots,\bar{t}}_{k}).$$

Comparing these two formulas for $F(\bar{u})$ completes the proof. □

It is interesting to note that the convex-hull property of the Bézier scheme for curves carries over from a curve F to its blossom f. The standard Bézier theory tells us that the point $F(\bar{u})$ will be within the convex hull of the Bézier points of the segment $F([\bar{s},\bar{t}])$ as long as $\bar{u}$ is in $[\bar{s},\bar{t}]$. More generally, the blossom value $f(\bar{u}_1,\ldots,\bar{u}_n)$ will also be in that convex hull as long as all of the $\bar{u}_i$ are in $[\bar{s},\bar{t}]$.

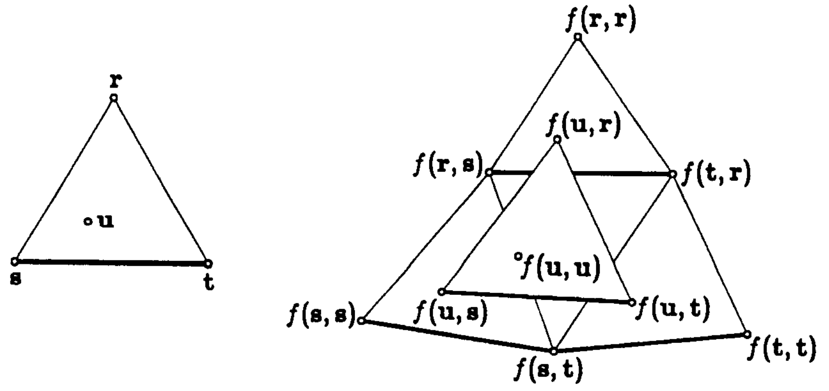


Fig. 3.4 The de Casteljau Algorithm for a quadratic Bézier surface

We can begin to develop a geometric intuition for the behavior of blossoms from Prop. 3.3. Consider a quadratic polynomial curve in the plane, that is, a parabola F. For any s and t, we know from Prop. 3.3 that the three Bézier points of the parabolic segment $F([\bar{s},\bar{t}])$ are the blossom values $f(\bar{s},\bar{s})$, $f(\bar{s},\bar{t})$, and $f(\bar{t},\bar{t})$. Hence, the point $f(\bar{s},\bar{t})$ must be the intersection of the tangents to the parabola $F(\bar{u})$ at $u = s$ and $u = t$, as shown (without the bars) in Fig. 1.1. As a consequence, the range of the blossom f is precisely the parabola itself together with its exterior. Given any point $\mathbf{x}$ exterior to the parabola, we can compute the unique $\bar{s}$ and $\bar{t}$ with the property that $\mathbf{x} = f(\bar{s},\bar{t})$ by finding the two tangents to the parabola that pass through $\mathbf{x}$. This type of geometric intuition is a valuable adjunct to the algebraic techniques that we used in proving the Blossoming Principle. As a sample of the algebraic approach, note that the blossom f of the standard parabola $F(\bar{u}) := \langle u, u^2\rangle$ is given by $f(\bar{u},\bar{v}) = \langle (u+v)/2, uv\rangle$.

Surfaces aren't much harder than curves. Recall that a polynomial surface is simply a polynomial function $F\colon P \to Q$ whose domain P is two-dimensional. The blossom of a surface F is a symmetric, multiaffine function $f\colon P^n \to Q$. Fig. 3.4 shows an application of the de Casteljau Algorithm computing a point on a quadratic surface from the six Bézier points of that surface. The surface F itself, which is not shown, is the locus of all points of the form $f(\mathbf{u},\mathbf{u})$ as $\mathbf{u}$ varies over the parameter plane P. The image under F of the domain triangle $\Delta\mathbf{rst}$ is a triangular surface patch with vertices $f(\mathbf{r},\mathbf{r})$, $f(\mathbf{s},\mathbf{s})$, and $f(\mathbf{t},\mathbf{t})$ and with parabolic edge curves. (Degeneracies are even harder to avoid for surfaces than for curves; since a quadratic surface F has six Bézier points, we would need five dimensions to draw one that is nondegenerate.)

Proposition 3.5. *Let $\Delta\mathbf{rst}$ be a reference triangle in the two-dimensional affine space P. A symmetric, n-affine function $f\colon P^n \to Q$ is completely determined by its values on argument bags (that is, multisets) that consist entirely of $\mathbf{r}$'s, $\mathbf{s}$'s, and $\mathbf{t}$'s. In particular, the formula*

$$f(\underbrace{\mathbf{r},\ldots,\mathbf{r}}_{i},\underbrace{\mathbf{s},\ldots,\mathbf{s}}_{j},\underbrace{\mathbf{t},\ldots,\mathbf{t}}_{k}) = \mathbf{x}_{ijk} \tag{3.6}$$

provides a one-to-one correspondence between symmetric, n-affine functions f and triangular arrays of $\binom{n+2}{2}$ points $(\mathbf{x}_{ijk})$ in Q, where i, j, and k are nonnegative integers that sum to n.

Proof. In our discussion of curves, we used the two endpoints of the reference segment $[\bar{s},\bar{t}]$ as an affine frame for L. (A *frame* for an affine space is the affine analog of a basis for a linear space.) Analogously, we shall use the three vertices $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{t}$ of the reference triangle to form an affine frame for the plane P. Every point $\mathbf{u}$ in P can be written uniquely as an affine combination of the points in the frame, in the form

$$\mathbf{u} = \beta_{\mathbf{r}}(\mathbf{u})\mathbf{r} + \beta_{\mathbf{s}}(\mathbf{u})\mathbf{s} + \beta_{\mathbf{t}}(\mathbf{u})\mathbf{t}.$$

The real coefficients $\beta_{\mathbf{r}}(\mathbf{u})$, $\beta_{\mathbf{s}}(\mathbf{u})$, and $\beta_{\mathbf{t}}(\mathbf{u})$ in this expansion are called the *barycentric coordinates* of the point $\mathbf{u}$, and they sum to 1.

Let $f\colon P^n \to Q$ be a symmetric, n-affine function. If we expand the first argument $\mathbf{u}_1$ of f in terms of $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{t}$, we deduce that $f(\mathbf{u}_1,\ldots,\mathbf{u}_n)$ is equal to

$$\beta_{\mathbf{r}}(\mathbf{u}_1)f(\mathbf{r},\mathbf{u}_2,\ldots,\mathbf{u}_n) + \beta_{\mathbf{s}}(\mathbf{u}_1)f(\mathbf{s},\mathbf{u}_2,\ldots,\mathbf{u}_n) + \beta_{\mathbf{t}}(\mathbf{u}_1)f(\mathbf{t},\mathbf{u}_2,\ldots,\mathbf{u}_n).$$

If we similarly expand each of the other arguments of f, we arrive at a sum for $f(\mathbf{u}_1,\ldots,\mathbf{u}_n)$ that has 3^n terms:

$$\sum_{\substack{I,J,K \text{ any}\\ \text{partition of}\\ \{1,2,\ldots,n\}}} \prod_{i\in I}\beta_{\mathbf{r}}(\mathbf{u}_i)\prod_{j\in J}\beta_{\mathbf{s}}(\mathbf{u}_j)\prod_{k\in K}\beta_{\mathbf{t}}(\mathbf{u}_k)f(\underbrace{\mathbf{r},\ldots,\mathbf{r}}_{|I|},\underbrace{\mathbf{s},\ldots,\mathbf{s}}_{|J|},\underbrace{\mathbf{t},\ldots,\mathbf{t}}_{|K|}). \tag{3.7}$$

This formula expresses every value of f in terms of the $\binom{n+2}{2}$ special values in which all of the arguments of f are either $\mathbf{r}$'s, $\mathbf{s}$'s, or $\mathbf{t}$'s. Furthermore, there are no constraints on these $\binom{n+2}{2}$ special values, since the formula

$$f(\mathbf{u}_1,\ldots,\mathbf{u}_n) := \sum_{\substack{I,J,K \text{ any}\\ \text{partition of}\\ \{1,2,\ldots,n\}}} \prod_{i\in I}\beta_{\mathbf{r}}(\mathbf{u}_i)\prod_{j\in J}\beta_{\mathbf{s}}(\mathbf{u}_j)\prod_{k\in K}\beta_{\mathbf{t}}(\mathbf{u}_k)\mathbf{x}_{|I||J||K|}$$

defines a symmetric, n-affine function whose special values satisfy Eq. (3.6). □

Proposition 3.8: Bézier triangular surface patches. *If $F\colon P \to Q$ is a polynomial surface of degree n and $\triangle\mathbf{rst}$ is a reference triangle in the parameter plane P, then the Bézier points by which the triangular surface patch $F(\triangle\mathbf{rst})$ is controlled in the standard theory are precisely the blossom values*

$$f(\underbrace{\mathbf{r},\ldots,\mathbf{r}}_{i},\underbrace{\mathbf{s},\ldots,\mathbf{s}}_{j},\underbrace{\mathbf{t},\ldots,\mathbf{t}}_{k}).$$

Proof. When all of the arguments of the blossom $f(\mathbf{u}_1,\ldots,\mathbf{u}_n)$ are equal, the 3^n terms in Sum (3.7) collapse down as follows:

$$f(\mathbf{u},\ldots,\mathbf{u}) = \sum_{i+j+k=n}\binom{n}{i\;j\;k}\beta_{\mathbf{r}}(\mathbf{u})^i\beta_{\mathbf{s}}(\mathbf{u})^j\beta_{\mathbf{t}}(\mathbf{u})^k f(\underbrace{\mathbf{r},\ldots,\mathbf{r}}_{i},\underbrace{\mathbf{s},\ldots,\mathbf{s}}_{j},\underbrace{\mathbf{t},\ldots,\mathbf{t}}_{k}).$$

Comparing this formula with the standard expansion of a Bézier surface in terms of its Bézier points [5] completes the proof. □

The case where the domain space P has arbitrary dimension p is completely analogous, although it is such a notational challenge that we won't bother to go through it in detail. One begins by choosing a reference p-simplex $[\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_p]$, whose vertices provide an affine frame for P. A polynomial function $F: P \to Q$ then has $\binom{n+p}{p}$ Bézier points, each of which is the result of choosing, with replacement, n points from among the $p+1$ simplex vertices and evaluating the blossom f on those n points. Every value of the blossom can be expressed as an affine combination of these Bézier points by means of a sum that involves $(p+1)^n$ terms.

4. The Tensoring Principle

Blossoming is quite a useful trick; this was suggested by the figures in Section 1 and will be borne out in more detail in what follows. But there are some problems with blossoming as we know it so far. To begin with, there is a notational hassle: we had to resort to horizontal braces in order to write down the kth Bézier point

$$f(\underbrace{\bar{s}, \ldots, \bar{s}}_{n-k}, \underbrace{\bar{t}, \ldots, \bar{t}}_{k})$$

of an n-ic curve $F: L \to Q$ as a value of the blossom f. It would be more convenient if the arguments to a blossom were multiplied, to form a product, rather than concatenated, to form a sequence. Then we could write the kth Bézier point something like "$f(\bar{s}^{n-k}\bar{t}^k)$", using exponential notation. It turns out that there is another important principle of mathematics, called the ***tensor-product construction***, that can be used to convert an old-style, multiaffine blossom f into a new-style, affine blossom $f^{\otimes}$, whose argument is a product of n points. More precisely, an affine blossom takes an ***n-tensor*** as its argument, where a point is a ***1-tensor***. Our goal in this section is to apply the tensor-product construction to convert our blossoms from multiaffine maps to affine maps, thus reaping both notational and conceptual benefits.

Warning 4.1. Tensoring is not as hard as most people think, but it is harder than blossoming—it demands some tolerance for abstraction. If you get confused in the next couple of sections, there are two things that you might try. One is to jump to Section 7, and then to Sections 14 and following, treating expressions of the form $f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$ as the author's funny way of writing the blossom value $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$. Another alternative is to give up on this paper and switch to the ***Reader's Digest*** version [35], in which I describe blossoming as clearly as I could in twenty pages, with no tensors.

More formally, the goal of our tensoring work is as follows: Given an affine space P and a degree n, we want to construct an affine space $P^{\otimes n}$, called the ***n-th symmetric tensor power*** of P. The interesting property of the tensor power space $P^{\otimes n}$ is that symmetric, multiaffine maps $f: P^n \to Q$ (such as old-style blossoms) are equivalent to affine maps $f^{\otimes}: P^{\otimes n} \to Q$ (new-style blossoms). In particular, the correspondence between f and $f^{\otimes}$ is reflected in the identity $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$, where, on the right-hand side, we are using the tensor product operation to build up an n-tensor in $P^{\otimes n}$ out of points, which are 1-tensors in $P = P^{\otimes 1}$.

Different variants of tensoring show up in different places in mathematics. In each case, the effect of the tensor-product construction is to replace multilinearity with simple linearity, at the price of complicating the domain space. In linear algebra, for example, an asymmetric, linear variant of the tensor-product construction is used to replace a bilinear map whose domain is a Cartesian product $V \times W$ of linear spaces with a linear map whose domain is the tensor product $V \otimes W$.

The variant of tensoring that we need is symmetric and affine, instead of asymmetric and linear. The affine versus linear issue doesn't make much difference, but the symmetry condition in our variant simplifies things in important ways. First, it means that we will be computing the tensor powers $P^{\otimes n}$ of a single space P, rather than the tensor products of different spaces. Second, the product operation that builds n-tensors in $P^{\otimes n}$ out of points in $P = P^{\otimes 1}$ in the symmetric case is commutative. Therefore, we will use simple multiplication, rather than the more standard "$\otimes$", to denote that product, writing an n-tensor as $\mathbf{u}_1 \cdots \mathbf{u}_n$, rather than as $\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_n$. Third, in the symmetric case, all of the tensors on a space P, under addition, scalar multiplication, and tensor product, form an algebra that is isomorphic to a very familiar algebra: the algebra of polynomials in $p+1$ independent variables, where $p = \dim(P)$. See, for example, Prop. 8.1 in Chapter XVI of the second edition of Lang's ***Algebra*** [34]. We will take advantage of this isomorphism by using certain spaces of formal polynomials as the raw material out of which we will build our tensor power spaces.

Warning: "Tensor-product surfaces" in computer-aided geometric design have "tensor" in their name because of an application of the more common, asymmetric variant of the Tensoring Principle. In Section 12, we will finally get around to considering tensor-product surfaces and to studying the relationship between that variant of the Tensoring Principle and the symmetric variant that we are about to develop and apply.

It's time for some mathematics. We know that the tensor power space $P^{\otimes n}$ should include elements that correspond to the products of n points in P. One way to build such a space is to treat the points in P as formal symbols and to allow ourselves to multiply them formally. If we do precisely that, we end up with a technique that applies, not to symmetric, multiaffine maps, but rather to arbitrary symmetric maps.

Let S be any set, let Q be an affine space, and let $h\colon S^n \to Q$ be an arbitrary symmetric function. For each element s of S, we shall invent a brand-new formal variable, written X_s. An *X-polynomial on S* is a polynomial in these new *X-variables*. Let $X[S]$ denote the algebra consisting of all X-polynomials on S (with coefficients in the real numbers $\mathbf{R}$). There are as many X-variables as there are points in S, possibly infinitely many; but any particular X-polynomial Φ in $X[S]$ can only involve a finite number of the X-variables.

One simple type of X-polynomial is an X-monomial of degree n, that is, an expression of the form $X_{s_1} \cdots X_{s_n}$, where the s_i are elements of S. Since the function h is symmetric, we can, in some sense, evaluate h at such a monomial. To be more precise, we can define a new function h_X whose arguments are X-monomials by means of the formula $h_X(X_{s_1} \cdots X_{s_n}) := h(s_1, \ldots, s_n)$. It is convenient to extend the domain of this new function h_X to include affine combinations of X-monomials as well. For example, if $n = 2$, we would like equations such as

$$h_X\left(\frac{X_r^2}{2} + \frac{X_s X_t}{2}\right) = \frac{h_X(X_r^2)}{2} + \frac{h_X(X_s X_t)}{2} = \frac{h(r,r)}{2} + \frac{h(s,t)}{2}$$

to hold.

In order to be a legal argument to this extended function h_X, an X-polynomial Φ must be homogeneous of degree n. In addition, since the range Q of h is only an affine space, the coefficients of Φ must sum to 1. To help formalize this latter restriction, let us, just for the purposes of this paper, refer to the sum of the coefficients of an X-polynomial Φ as its *flavor*, written $\mathrm{Flav}(\Phi)$; the latter restriction is then $\mathrm{Flav}(\Phi) = 1$. Note that we could characterize the flavor functional $\mathrm{Flav}\colon X[S] \to \mathbf{R}$ more abstractly as the unique algebra isomorphism with the property that $\mathrm{Flav}(X_s) = 1$ for all points s in S. Let $X_c^k[S]$ denote the affine subspace of the algebra $X[S]$ consisting of all polynomials that are k-homogeneous and c-flavored. Then, the domain of our extended function h_X is the affine space $X_1^n[S]$.

We can summarize the results of the last few paragraphs as the following easy proposition.

Proposition 4.2: The X-Polynomial Construction. *If S is any set and Q is any affine space, symmetric functions $h\colon S^n \to Q$ are equivalent to affine functions $h_X\colon X_1^n[S] \to Q$. In particular, given a function of either type, a unique function of the other type exists that satisfies the identity $h(s_1, \ldots, s_n) = h_X(X_{s_1} \cdots X_{s_n})$.* $\square$

The X-Polynomial Construction is a (rather trivial) tool for analyzing symmetric functions whose values lie in an affine space. Since an old-style, multiaffine blossom $f\colon P^n \to Q$ is such a function, we can apply the X-Polynomial Construction to f to obtain a corresponding affine map $f_X\colon X_1^n[P] \to Q$. Unfortunately, we can't always go backward: if we start with an arbitrary affine map $h_X\colon X_1^n[P] \to Q$, the symmetric function $h\colon P^n \to Q$ that corresponds to h_X under the X-Polynomial Construction probably won't be multiaffine, and hence won't be a blossom. For example, suppose that P is the affine line L, and consider an affine property of L, such as $\bar{1} = (\bar{0} + \bar{2})/2$. In order to make $h\colon L^n \to Q$ be multiaffine, the map $h_X\colon X_1^n[L] \to Q$ would have to map the X-polynomial $X_{\bar{1}}\Phi$ to the point in Q halfway between $h_X(X_{\bar{0}}\Phi)$ and $h_X(X_{\bar{2}}\Phi)$, where Φ denotes any X-polynomial in $X_1^{n-1}[L]$. Since h_X is affine, we are guaranteed that

$$\frac{h_X(X_{\bar{0}}\Phi) + h_X(X_{\bar{2}}\Phi)}{2} = h_X\left(\left(\frac{X_{\bar{0}} + X_{\bar{2}}}{2}\right)\Phi\right).$$

But there is no particular relationship, in the X-Polynomial Construction, between $(X_{\bar{0}} + X_{\bar{2}})/2$ and $X_{\bar{1}}$. Indeed, $X_{\bar{0}}$, $X_{\bar{1}}$, and $X_{\bar{2}}$ are just three different formal variables out of the uncountable collection X_P of formal variables. Intuitively, the X-Polynomial Construction doesn't handle blossoms properly because it ignores the affine structure of the parameter space P. More formally, the affine space $X_1^n[P]$ is too big to serve as the tensor power space $P^{\otimes n}$.

To get a better sense of where we stand, let us rephrase our blossom-based analysis of polynomial curves in Prop. 3.2 using the affine function f_X instead of the blossom f itself, even though we already know that the domain $X_1^n[P]$ of f_X is not the tensor power space for which we are searching. In particular, let $f\colon L^n \to Q$ be the blossom of an n-ic polynomial curve, and let $f_X\colon X_1^n[L] \to Q$ be the affine function that corresponds to f under the X-Polynomial Construction. In proving Prop. 3.2, we expressed an arbitrary blossom value $f(\bar{u}_1, \ldots, \bar{u}_n) = f_X(X_{a_1} \cdots X_{a_n})$ as an affine combination of the Bézier points $f_X(X_{\bar{s}}^{n-k} X_{\bar{t}}^k)$, where $[\bar{s}, \bar{t}]$ is a reference segment in L and k is in $[0, n]$. Converting to the use of f_X throughout, our proof of Prop. 3.2 looks like this:

$$\begin{aligned} f_X(X_{a_1} \cdots X_{a_n}) &= f_X\left(\prod_{1 \le i \le n} \left(\left(\frac{t-u_i}{t-s}\right) X_{\bar{s}} + \left(\frac{u_i - s}{t-s}\right) X_{\bar{t}} \right) \right) \\ &= f_X\left(\sum_{\substack{I \cup J = \{1,2,\ldots,n\} \\ I \cap J = \emptyset}} \prod_{i \in I} \left(\frac{t-u_i}{t-s}\right) \prod_{j \in J} \left(\frac{u_j - s}{t-s}\right) X_{\bar{s}}^{|I|} X_{\bar{t}}^{|J|} \right) \\ &= \sum_{\substack{I \cup J = \{1,2,\ldots,n\} \\ I \cap J = \emptyset}} \prod_{i \in I} \left(\frac{t-u_i}{t-s}\right) \prod_{j \in J} \left(\frac{u_j - s}{t-s}\right) f_X(X_{\bar{s}}^{|I|} X_{\bar{t}}^{|J|}). \end{aligned}$$

Of this three-step derivation, the second step is simple algebra and the third step is justified because f_X is affine. The first step is the subtle one. We are allowed to replace the factor X_{a_i} in the argument to f_X by the X-binomial

$$\left(\frac{t-u_i}{t-s}\right) X_{\bar{s}} + \left(\frac{u_i - s}{t-s}\right) X_{\bar{t}}$$

only because the multiaffineness of f implies that f_X has special properties. Once again, we have run across the problem that the X-Polynomial Construction ignores the affine structure of P. Our next task is to fix that problem.

There are two ways to impose on the X-variables the affine structure of their subscripts, and hence to shrink the space $X_1^n[P]$ down to something that can serve as the nth tensor power of P. The simplest approach is to choose, once and for all, an affine frame $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ for P and then to restrict ourselves to using only the $p+1$ distinguished X-variables $X_{\mathbf{r}_0}$ through $X_{\mathbf{r}_p}$. Since the points $\mathbf{r}_0$ through $\mathbf{r}_p$ form a frame, there are no affine relationships among them, so there is no need to impose any affine structure on the corresponding frame variables $X_{\mathbf{r}_0}$ through $X_{\mathbf{r}_p}$. In this approach, instead of writing $X_{\mathbf{u}}$ for $\mathbf{u}$ in P, we must first expand $\mathbf{u}$ as an affine combination of the frame points, say $\mathbf{u} = \beta_0(\mathbf{u})\mathbf{r}_0 + \cdots + \beta_p(\mathbf{u})\mathbf{r}_p$, and then write instead the corresponding affine combination of the frame variables. We can define a map $\rho\colon X_P \to X_1^1[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ that rewrites an arbitrary X-variable in terms of the frame variables by the formula

$$\rho(X_{\mathbf{u}}) := \beta_0(\mathbf{u}) X_{\mathbf{r}_0} + \cdots + \beta_p(\mathbf{u}) X_{\mathbf{r}_p}.$$

If Φ is any X-polynomial on P, we can apply ρ to each X-variable in Φ separately to rewrite Φ completely in terms of the frame variables. We shall refer to the resulting map $\rho: X[P] \to X[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ as the ***normal-form map associated with the affine frame*** $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ *for* P. Note that the normal form $\rho(\Phi)$ has the same degree and the same flavor as Φ; furthermore, if Φ is homogeneous, then $\rho(\Phi)$ will be homogeneous also. Using ρ, we can summarize this first approach to tensor powers as follows: Symmetric, multiaffine maps $f: P^n \to Q$ are equivalent to affine maps $f_X: X_1^n[\mathbf{r}_0, \ldots, \mathbf{r}_p] \to Q$. Given a map of either type, a unique map of the other type exists that satisfies $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f_X(\rho(X_{\mathbf{u}_1} \cdots X_{\mathbf{u}_n}))$.

One problem with this first approach is the need to apply the normal-form map ρ as part of the fundamental correspondence. But a more severe problem is the dependence of the entire approach on the choice of a particular affine frame for P. Note that the very definition of the tensor power space itself, $X_1^n[\mathbf{r}_0, \ldots, \mathbf{r}_p]$, is dependent on the frame. Such a frame dependence would be particularly clumsy when we started working with splines, since we will often want to use a different frame for each piece into which the parameter space P is partitioned. Therefore, we shall adopt a more high-powered, second approach to tensor powers, one in which we shrink the space $X_1^n[P]$ down by defining an equivalence relation on X-polynomials.

The key idea behind this second approach comes from considering the relation on X-polynomials determined by the equation $\rho(\Phi_1) = \rho(\Phi_2)$. The map ρ depends upon a choice of an affine frame for P, of course. But the truth or falsity of the relation $\rho(\Phi_1) = \rho(\Phi_2)$ is independent of the frame. We shall say that Φ_1 and Φ_2 are ***affinely equivalent***, written $\Phi_1 \sim \Phi_2$, if the equation $\rho(\Phi_1) = \rho(\Phi_2)$ holds in one frame, and hence in all frames. We will use square brackets to denote equivalence classes with respect to affine equivalence; that is, $[\Phi]$ denotes the set of all X-polynomials that are affinely equivalent to Φ. Note that, if $\Phi_1 \sim \Phi_2$ and $\Psi_1 \sim \Psi_2$, then $u\Phi_1 + v\Psi_1 \sim u\Phi_2 + v\Psi_2$, and also $\Phi_1\Psi_1 \sim \Phi_2\Psi_2$. Therefore, we can take linear combinations of equivalence classes or multiply equivalence classes and get well-defined results.

Let $\mathcal{X}[P]$ denote the set of all affine equivalence classes of X-polynomials in $X[P]$. Under addition, scalar multiplication, and polynomial multiplication, this set becomes an algebra, which is called the ***symmetric tensor algebra of P*** [34]. The elements of this algebra are called ***tensors on P***; that is, a tensor on P is an affine equivalence class $[\Phi]$ of X-polynomials on P. The multiplication operation on tensors $[\Phi] \cdot [\Psi] := [\Phi\Psi]$ is called the ***tensor product***.

If $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ is any affine frame for P, we can use the associated normal-form map ρ to choose a canonical representative out of each equivalence class, that is, to choose a normal form for any tensor. In particular, the normal form of the tensor $[\Phi]$ is the X-polynomial $\rho(\Phi)$. In fact, this correspondence is an algebra isomorphism between the tensor algebra $\mathcal{X}[P]$ and the polynomial algebra $X[\mathbf{r}_0, \ldots, \mathbf{r}_p]$, a fact that is worth remembering.

Proposition 4.3. *Let* $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ *be any affine frame for* P, *and let* ρ *be the associated normal-form map. Then, the correspondence* $[\Phi] \mapsto \rho(\Phi)$ *is an isomorphism between the symmetric tensor algebra* $\mathcal{X}[P]$ *of* P *and the algebra* $X[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ *of polynomials in the* $p+1$ *independent variables* $X_{\mathbf{r}_0}$ *through* $X_{\mathbf{r}_p}$. □

Corollary 4.4. *The symmetric tensor algebra $\mathcal{X}[P]$ is a unique factorization domain.* $\square$

A tensor $[\Phi]$ on P is called ***homogeneous of degree n***, or ***n-homogeneous***, or simply ***an n-tensor***, if the affine equivalence class $[\Phi]$ contains some X-polynomial that is n-homogeneous. The X-polynomial Φ itself may not be homogeneous. For example, over the line L, the tensor $[\Phi]$ where $\Phi = X_{\bar{3}}X_{\bar{4}} + X_{\bar{0}} + X_{\bar{2}} - 2X_{\bar{1}}$ is 2-homogeneous, since $X_{\bar{0}} + X_{\bar{2}} \sim 2X_{\bar{1}}$ implies $\Phi \sim X_{\bar{3}}X_{\bar{4}}$. Extending the concept of flavor from X-polynomials to tensors is easier: all of the X-polynomials in an affine equivalence class $[\Phi]$ must have the same flavor, and we shall refer to that common value as the ***flavor*** of the tensor $[\Phi]$. Let $\mathcal{X}_c^k[P]$ denote the affine subspace of the tensor algebra $\mathcal{X}[P]$ consisting of all c-flavored k-tensors on P. We shall use $\mathcal{X}_1^n[P]$ as our model of the nth symmetric tensor power $P^{\otimes n}$ of P. That is, we will define $P^{\otimes n} := \mathcal{X}_1^n[P]$ to be the affine space consisting of all 1-flavored n-tensors. If we choose any affine frame for P, the correspondence of Prop. 4.3 will give an isomorphism between $\mathcal{X}_1^n[P]$ and $X_1^n[\mathbf{r}_0, \ldots, \mathbf{r}_p]$. Thus, our second approach to tensor powers gets the same answer as the first approach did; but it does so in a frame-independent way.

Tensors on P that are homogeneous of degree one are particularly simple; in fact, a 1-flavored 1-tensor on P is essentially the same thing as a point in P. To see this, note that every X-polynomial Φ in $X_1^1[P]$ is simply an affine combination of X-variables. If we take the corresponding affine combination of the subscripts of those variables, we get a point $\mathbf{u}$ in P with the property that $\Phi \sim X_{\mathbf{u}}$. Thus, every equivalence class in $\mathcal{X}_1^1[P]$ has the form $[X_{\mathbf{u}}]$ for a unique point $\mathbf{u}$ in P, and the first tensor power $P^{\otimes 1} = \mathcal{X}_1^1[P]$ of P is actually isomorphic to P under the correspondence $[X_{\mathbf{u}}] \mapsto \mathbf{u}$. By an abuse of notation, we shall use this correspondence to identify $P^{\otimes 1}$ with P. This identification allows us to reinterpret the tensor-product operation as a multiplication on the points of P; that is, if $\mathbf{u}$ and $\mathbf{v}$ are points in P, and hence also 1-flavored 1-tensors on P, their product $\mathbf{uv}$ is the 1-flavored 2-tensor $[X_{\mathbf{u}}X_{\mathbf{v}}]$.

It is at this point that we earn the reward for our earlier pedantic insistence on distinguishing between points $\bar{u}$ in the affine line L and scalars u in the coefficient field $\mathbf{R}$. Note that the expression $\bar{2}\,\bar{3}$, being the product of two points in L, is a 1-flavored 2-tensor on L. If we weren't distinguishing between points and scalars, we would have to use a special symbol to denote the tensor product operation on points in L or else we would risk confusing the tensor product $\bar{2} \cdot \bar{3} = \bar{2}\,\bar{3}$ of the points $\bar{2}$ and $\bar{3}$ with the normal product $2 \cdot 3 = 6$ of the scalars 2 and 3.

For $n \geq 2$, n-tensors are a little more subtle. For example, every product of n points $\mathbf{u}_1 \cdots \mathbf{u}_n$ is a 1-flavored n-tensor; but not every 1-flavored n-tensor can be written as the product of n points. As we shall study in Section 9, the n-tensors that can be factored into a product of 1-tensors are called ***simple***, while the rest are called ***compound***—and compound n-tensors do exist for $n \geq 2$. While not every n-tensor is simple, the simple n-tensors do span the tensor power space $P^{\otimes n} = \mathcal{X}_1^n[P]$; this follows from the obvious fact that every X-polynomial in $X_1^n[P]$ is an affine combination of X-monomials of degree n.

Proposition 4.5: The Tensoring Principle, symmetric affine variant. *Let P and Q be finite-dimensional affine spaces, and let $P^{\otimes n} = \mathcal{X}_1^n[P]$ be the nth symmetric tensor power of P. Symmetric, n-affine maps $f\colon P^n \to Q$ are equivalent to affine maps $f^{\otimes}\colon P^{\otimes n} \to Q$. In particular, given a map of either type, a unique map of the other type exists that satisfies the identity $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$.*

Proof. Suppose that we start with a symmetric, multiaffine map $f\colon P^n \to Q$. The X-Polynomial Construction gives us an affine map $f_X\colon X_1^n[P] \to Q$ that satisfies $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f_X(X_{\mathbf{u}_1} \cdots X_{\mathbf{u}_n})$. Since f is multiaffine, the map f_X will have the additional property that, if two X-polynomials Φ_1 and Φ_2 in $X_1^n[P]$ are affinely equivalent, then $f_X(\Phi_1) = f_X(\Phi_2)$. Hence, we can define a map $f^{\otimes}\colon \mathcal{X}_1^n[P] \to Q$ by the formula $f^{\otimes}([\Phi]) := f_X(\Phi)$. It follows that $f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n) = f^{\otimes}([X_{\mathbf{u}_1} \cdots X_{\mathbf{u}_n}]) = f_X(X_{\mathbf{u}_1} \cdots X_{\mathbf{u}_n}) = f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$. The map $f^{\otimes}$ is affine because f_X is affine. Finally, the map $f^{\otimes}$ is unique because the correspondence with f determines the values of $f^{\otimes}$ on all simple 1-flavored n-tensors, and those simple tensors span the domain of $f^{\otimes}$.

The same procedure works in reverse. Suppose that we start with an affine map $f^{\otimes}\colon \mathcal{X}_1^n[P] \to Q$. We can define an affine map $f_X\colon X_1^n[P] \to Q$ by the formula $f_X(\Phi) := f^{\otimes}([\Phi])$. The X-Polynomial Construction then gives us a symmetric map $f\colon P^n \to Q$ that satisfies $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f_X(X_{\mathbf{u}_1} \cdots X_{\mathbf{u}_n}) = f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$. The map f will be multiaffine because f_X is constant on affine equivalence classes. Finally, the correspondence with $f^{\otimes}$ determines every value of f, and hence the map f is certainly unique. □

Corollary 4.6. *An affine map $f^{\otimes}\colon P^{\otimes n} \to Q$ is completely determined by its values on the n-tensors $\mathbf{u}^n$ that are perfect nth powers of points in P.*

Proof. This follows at once from the Tensoring Principle and Cor. 2.2. □

If $F\colon P \to Q$ is a polynomial map of degree n, we now have two different blossoms for F: a symmetric, multiaffine map $f\colon P^n \to Q$ that satisfies $F(\mathbf{u}) = f(\mathbf{u}, \ldots, \mathbf{u})$ and an affine map $f^{\otimes}\colon P^{\otimes n} \to Q$ that satisfies $F(\mathbf{u}) = f^{\otimes}(\mathbf{u}^n)$. We shall distinguish between these two blossoms by calling f the ***multiaffine blossom*** while calling $f^{\otimes}$ the ***affine blossom***. As a first step in understanding how these blossoms relate to each other, the following proposition considers their ranges.

Proposition 4.7. *Let $F\colon P \to Q$ be an n-ic polynomial function, let $f\colon P^n \to Q$ be the multiaffine blossom of F, and let $f^{\otimes}\colon P^{\otimes n} \to Q$ be the affine blossom of F. Also, let the set B in Q consist of the Bézier points of F with respect to some reference simplex in P. Finally, let* Span(S) *denote the affine span of a subset S of Q. Then, we have*

$$\operatorname{Span}(\operatorname{Range}(F)) = \operatorname{Span}(B) = \operatorname{Span}(\operatorname{Range}(f)) = \operatorname{Range}(f^{\otimes}).$$

Proof. Since we can express $F(\mathbf{u})$ as an affine combination of the Bézier points, we have $\operatorname{Range}(F) \subseteq \operatorname{Span}(B)$. Since each Bézier point is a value of the multiaffine blossom, we have $B \subseteq \operatorname{Range}(f)$. Finally, the fundamental correspondence $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$ shows that $\operatorname{Range}(f) \subseteq \operatorname{Range}(f^{\otimes})$. Putting these facts together, we have

$$\mathrm{Span}(\mathrm{Range}(F)) \subseteq \mathrm{Span}(B) \subseteq \mathrm{Span}(\mathrm{Range}(f)) \subseteq \mathrm{Span}(\mathrm{Range}(f^{\otimes})).$$

Since $f^{\otimes}$ is an affine map, its range is already a flat (that is, an affine subspace), so $\mathrm{Span}(\mathrm{Range}(f^{\otimes})) = \mathrm{Range}(f^{\otimes})$. We will be done if we can show that $\mathrm{Range}(f^{\otimes}) \subseteq \mathrm{Span}(\mathrm{Range}(F))$.

To show this, it is enough to show that $f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$ is in $\mathrm{Span}(\mathrm{Range}(F))$ whenever the $\mathbf{u}_i$ are points in P, since the simple n-tensors $\mathbf{u}_1 \cdots \mathbf{u}_n$ span the domain $P^{\otimes n}$ of $f^{\otimes}$. That is, it is enough to show that $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is in $\mathrm{Span}(\mathrm{Range}(F))$. This is demonstrated in an explicit way by Eq. (2.3),

$$f(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n) = \frac{1}{n!} \sum_{\substack{S \subseteq \{1,2,\ldots,n\} \\ k=|S|}} (-)^{n-k} k^n F\left(\frac{1}{k} \sum_{i \in S} \mathbf{u}_i\right), \tag{2.3}$$

which actually expresses the blossom value $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ as an affine combination of values of F. There is only one difficulty: we didn't prove Eq. (2.3)—we only stated it. We can prove Eq. (2.3) by proving the following equality between tensors:

$$\mathbf{u}_1 \cdots \mathbf{u}_n = \frac{1}{n!} \sum_{\substack{S \subseteq \{1,2,\ldots,n\} \\ k=|S|}} (-)^{n-k} k^n \left(\frac{1}{k} \sum_{i \in S} \mathbf{u}_i\right)^n. \tag{4.8}$$

Eq. (2.3) follows from Eq. (4.8) by applying $f^{\otimes}$ to both sides.

Proving Eq. (4.8) is a straightforward exercise. Working on the right-hand side, we first cancel the k^n in the numerator and denominator, and then apply the Multinomial Theorem to expand the nth power, getting

$$\frac{1}{n!} \sum_{S \subseteq \{1,\ldots,n\}} (-)^{n-|S|} \sum_{\substack{j_1+\cdots+j_n=n \\ j_i>0 \Rightarrow i \in S}} \binom{n}{j_1 \cdots j_n} \mathbf{u}_1^{j_1} \cdots \mathbf{u}_n^{j_n}.$$

Interchanging the order of summation, we have

$$\frac{1}{n!} \sum_{j_1+\cdots+j_n=n} \binom{n}{j_1 \cdots j_n} \mathbf{u}_1^{j_1} \cdots \mathbf{u}_n^{j_n} \sum_{\substack{S \subseteq \{1,\ldots,n\} \\ j_i>0 \Rightarrow i \in S}} (-)^{n-|S|}.$$

If there is any i such that $j_i = 0$, the inner sum will give zero, since we can pair off sets S that do and don't contain i. In order for all of the j_i to be positive, they must all be 1, which forces $S = \{1, \ldots, n\}$, so the right-hand side simplifies to $\mathbf{u}_1 \cdots \mathbf{u}_n$, and we are done. □

Turning from ranges to domains, the domain of the affine blossom $f^{\otimes}$ is the tensor power space $P^{\otimes n}$, which is a rather complicated thing. Whenever we want to know something about that space, however, we can choose an affine frame for P and then use the isomorphism of Prop. 4.3 to convert our question about $P^{\otimes n}$ into a question about X-polynomials. For example, suppose that we wanted to know the dimension of the tensor power space $P^{\otimes n}$.

Proposition 4.9. *If P is a p-dimensional affine space, the nth symmetric tensor power $P^{\otimes n}$ of P is an affine space of dimension $\binom{n+p}{p} - 1$.*

Proof. Choose an affine frame $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ for P. The normal-form correspondence of Prop. 4.3 then provides an isomorphism $[\Phi] \mapsto \rho(\Phi)$ between the nth tensor power $P^{\otimes n} = \mathcal{X}_1^n[P]$ of P and the space $\mathcal{X}_1^n[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ of 1-flavored $\mathcal{X}$-polynomials that are n-homogeneous in the frame variables $X_{\mathbf{r}_0}$ through $X_{\mathbf{r}_p}$. The latter space has an obvious affine frame consisting of the $\binom{n+p}{p}$ different X-monomials of the form $X_{\mathbf{r}_0}^{k_0} \cdots X_{\mathbf{r}_p}^{k_p}$ where $\sum_{i=0}^{p} k_i = n$. The dimension of the space $\mathcal{X}_1^n[\mathbf{r}_0, \ldots, \mathbf{r}_p]$, and hence also the dimension of $P^{\otimes n}$, is just one less than the number of points in this frame. □

Two closing remarks: First, the Tensoring Principle is frequently applied in mathematical areas where a duality is present. In such areas, tensors come in three varieties: the purely primal ones, which are called *contravariant*, the purely dual ones, which are called *covariant*, and the impure ones, which are called *mixed*. (See Dodson and Poston [18] for the sad story of how the primal concept got the name "contravariant.") In this paper, we are dealing exclusively with points and other primal objects—there is no duality in sight. In particular, all of our tensors are contravariant; normally, we won't bother to mention this fact. Second, Prop. 4.5 deals with the symmetric affine variant of the Tensoring Principle. Before we are done, we will find good uses for four different variants: symmetric and asymmetric, affine and linear. It is worth noting that there is a fifth variant, which we won't have any use for in this paper: the *antisymmetric* or *alternating* linear variant. This variant arises in the study of multilinear functions $f(e_1, \ldots, e_n)$ that change sign whenever any two arguments are interchanged.

5. The Homogenizing Principle

So far, we have been working in the affine world. That is appropriate, since we started out trying to understand the nonhomogeneous polynomial curves and surfaces that are used in computer-aided geometric design. But we won't be able to stay in the affine world forever: the derivative F' of a polynomial curve $F: L \to Q$ has values that are vectors on Q, rather than points in Q, and the set of all vectors on Q forms a linear space rather than an affine space. To prepare for such forays into the linear world, we shall devote this section to our third (and last) important mathematical principle, the Homogenizing Principle, which converts from the affine world to the linear world at the price of adding one more dimension to the spaces involved. With all three principles at our disposal, we will be able to consider any polynomial map F in six different guises, as shown in Fig. 5.1.

The Homogenizing Principle is quite common, both in mathematics and in computer-aided geometric design. For example, the usual way of representing affine transformations of 3-space in computer-aided geometric design is to use 4-by-4 matrices whose last column is the unit vector $\langle 0, 0, 0, 1 \rangle$. Converting an affine map between 3-spaces into a restricted type of linear map between 4-spaces is a typical application of the Homogenizing Principle.

The first step on the road to the Homogenizing Principle is to linearize the

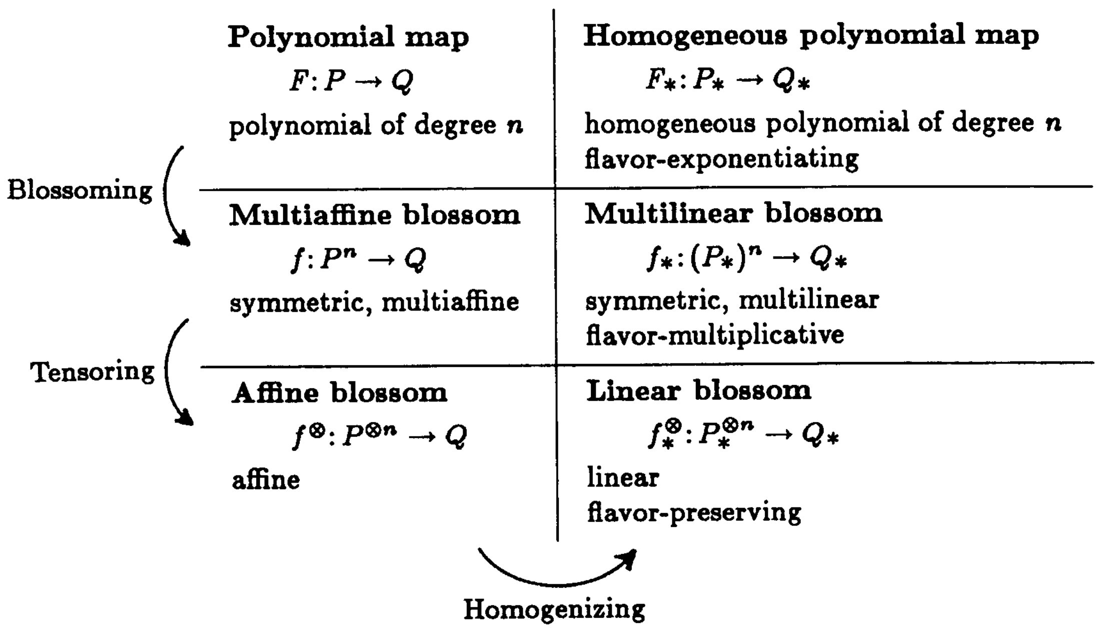


Fig. 5.1 The six guises of a polynomial map

affine spaces involved, that is, to imbed them as hyperplanes in linear spaces of one higher dimension. It turns out that we already did this as part of our work on the Tensoring Principle. Given an affine space P, we constructed the tensor algebra $\mathcal{X}[P]$. In this algebra, consider the set $\mathcal{X}_*^1[P]$ of all 1-tensors of all flavors. This set $\mathcal{X}_*^1[P]$ is a linear space, which we shall henceforth abbreviate as P_* and refer to as the *linearization of P*. (Comment for nit-pickers: The origin of the linear space P_* is the zero tensor $[0]$ on P, which we include in the set of 1-tensors $\mathcal{X}_*^1[P]$ by convention, even though the degree of the zero polynomial is not well-defined. More precisely, we make the convention that the zero polynomial is homogeneous of degree k for every k.) We shall also define $P_c := \mathcal{X}_c^1[P]$ to be the affine space consisting of all c-flavored 1-tensors on P, for any real number c. The spaces P_c are parallel hyperplanes in the linearization P_* of P. Since we are identifying points in P with 1-flavored 1-tensors on P, the space $P = P_1$ itself is a hyperplane in its linearization P_*.

The hyperplane parallel to P that goes through the origin of P_* is P_0, the set of all 0-flavored 1-tensors on P. Note that P_0 is actually a linear space, since sums or scalar multiples of 0-flavored tensors are 0-flavored. The elements of P_0 are called *vectors on P*, or, more precisely, *free vectors on P*. Note that the difference $\xi = \mathbf{v} - \mathbf{u}$ of two points in P is a vector on P, since the difference of two 1-flavored 1-tensors is a 0-flavored 1-tensor. The freedom in the term "free vector" is the freedom to slide the arrow from one place to another in P, as long as its length and direction are preserved; that is, if $\xi = \mathbf{v} - \mathbf{u}$, then we also have $\xi = (\mathbf{v} + \eta) - (\mathbf{u} + \eta)$ for any vector η on P.

The common types of coordinate systems on an affine space P extend easily to coordinate systems on its linearization P_*. The two cases of interest to us are

Cartesian coordinate systems and barycentric coordinate systems.

To form a Cartesian coordinate system for an affine space P, we choose an origin point $\mathbf{o}$ in P and a basis of vectors $\delta_1, \ldots, \delta_p$ for the linear space P_0. Then, each point $\mathbf{u}$ in P can be expressed uniquely in the form

$$\mathbf{u} = \langle u^1, \ldots, u^p \rangle = \mathbf{o} + \sum_{i=1}^{p} u^i \delta_i.$$

Note that the 1-tensors $\{\mathbf{o}, \delta_1, \ldots, \delta_p\}$ are linearly independent, when viewed as elements of the linearization P_*. Hence, we can get a coordinate system on P_* by using those $p + 1$ tensors as a basis. The effect of doing so is to extend the coordinate system from P to P_* by adding one new coordinate that measures flavor. In particular, an arbitrary 1-tensor e in P_* can be expressed uniquely in the form

$$e = \langle e^1, \ldots, e^p; e^0 \rangle = e^0 \mathbf{o} + \sum_{i=1}^{p} e^i \delta_i,$$

where $e^0 = \mathrm{Flav}(e)$. In order to follow the convention that is standard in computer-aided geometric design, we shall write the flavor coordinate e^0 last inside the angle brackets, even though it is numbered zero.

Barycentric coordinate systems are even easier to extend. A barycentric coordinate system for the affine space P begins with an affine frame $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ for P. Each point $\mathbf{u}$ in P can be written uniquely as an affine combination of the frame points, with the barycentric coordinates as the coefficients:

$$\mathbf{u} = \langle u^0, \ldots, u^p \rangle = \sum_{i=0}^{p} u^i \mathbf{r}_i, \qquad \text{where} \qquad \sum_{i=0}^{p} u^i = 1.$$

To extend such a coordinate system to the linearization P_* of P, we just drop the restriction about the coordinates summing to 1. That is, using the same frame points $\mathbf{r}_i$ as a linear basis for P_*, an arbitrary 1-tensor e in P_* can be expressed uniquely in the form

$$e = \langle e^0, \ldots, e^p \rangle = \sum_{i=0}^{p} e^i \mathbf{r}_i.$$

The flavor of the 1-tensor e is given by the sum $\mathrm{Flav}(e) = \sum_{i=0}^{p} e^i$ of its barycentric coordinates.

The Homogenizing Principle comes in many variants. The simplest variant is the one that converts an affine map into a linear map.

Proposition 5.2: The Homogenizing Principle, linear variant. *If P and Q are finite-dimensional affine spaces and P_* and Q_* are their linearizations, affine maps $h: P \to Q$ are equivalent to flavor-preserving linear maps $h_*: P_* \to Q_*$. That is, given a map of either type, a unique map of the other type exists that satisfies the identity $h(\mathbf{u}) = h_*(\mathbf{u})$ for all points $\mathbf{u}$ in P.*

Proof. One direction is trivial: If $h_*: P_* \rightarrow Q_*$ is linear, then the restriction of h_* to the hyperplane P in P_* will be an affine map $h: P \rightarrow Q_*$. If h_* is also flavor-preserving, it must map 1-flavored tensors to 1-flavored tensors, so we will have $h(P) \subseteq Q$.

In the reverse direction, suppose that $h: P \rightarrow Q$ is affine, and let $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ be an affine frame for P. The points $\mathbf{r}_i$ of the frame, interpreted as 1-tensors in P_*, form a linear basis for P_*. Hence, the $p+1$ conditions $h_*(\mathbf{r}_i) := h(\mathbf{r}_i)$ serve to uniquely define a linear map $h_*: P_* \rightarrow Q_*$. The restriction of h_* to P will agree with h on P because both maps are affine and they agree on an affine frame for P. Furthermore, the map h_* preserves the flavor of each basis element $\mathbf{r}_i$, since both $\mathbf{r}_i$ and $h_*(\mathbf{r}_i) = h(\mathbf{r}_i)$ are 1-flavored; since h_* is linear, it must preserve flavors in general. □

If we apply Prop. 5.2 to the affine blossom $f^{\circledast}: P^{\circledast n} \rightarrow Q$ of a polynomial map $F: P \rightarrow Q$, we can convert that affine blossom into the ***linear blossom of F***, which is a flavor-preserving linear map $f_*^{\circledast}: P_*^{\circledast n} \rightarrow Q_*$, as shown in the lower-right box in Fig. 5.1. The intuitive idea is that, while the affine blossom $f^{\circledast}$ demands a 1-flavored n-tensor on P as its argument and returns a 1-flavored 1-tensor on Q as its result, the linear blossom $f_*^{\circledast}$ will accept an n-tensor on P of any flavor as its argument, and will return as its result a 1-tensor on Q of that same flavor.

Strictly speaking, there are three different ways to interpret the expression $P_*^{\circledast n}$, which is used in Fig. 5.1 to denote the domain of the linear blossom $f_*^{\circledast}$. Fortunately, the three interpretations give canonically isomorphic results. The simplest choice is to interpret $P_*^{\circledast n}$ as $\mathcal{X}_*^n[P]$, the set of all n-tensors of any flavor on P. That was the interpretation that we assumed in the last paragraph. When we apply Prop. 5.2 to the affine blossom $f^{\circledast}$, however, the domain space of the resulting linear blossom $f_*^{\circledast}$ is actually $(P^{\circledast n})_*$, which, by our definitions, is the iterated tensor space $(P^{\circledast n})_* = (\mathcal{X}_1^n[P])_* = \mathcal{X}_*^1[\mathcal{X}_1^n[P]]$. The distinction arises here because we have chosen a particular linear space, constructed in a particular way, to be *the* linearization P_* of an affine space P. If we were being more abstract, we would define ***a linearization of P*** to be a triple (W, h, φ) where W is a linear space, $h: P \rightarrow W$ is a one-to-one affine map, and $\varphi: W \rightarrow \mathbf{R}$ is a linear flavor functional, satisfying $h(P) = \varphi^{-1}(1)$. It is straightforward to check that all linearizations W of a space P are canonically isomorphic. In particular, the two linearizations $P_*^{\circledast n}$ and $(P^{\circledast n})_*$ of $P^{\circledast n}$ are isomorphic.

The third choice is to interpret the expression $P_*^{\circledast n}$ to mean $(P_*)^{\circledast n}$, that is, the nth linear symmetric tensor power of the linearization of P. This is the interpretation that arises when using the linear variant of the Tensoring Principle to move down from the middle-right to the lower-right box in Fig. 5.1. We haven't studied how to compute tensor powers in the linear world, but, if we did, we would find that the space $(P_*)^{\circledast n}$ is also a linearization of the affine symmetric tensor power $P^{\circledast n}$. Thus, all three ways of parsing the domain of the linear blossom give the same result.

Exercise 5.3: The Tensoring Principle, symmetric linear variant. Develop the theory of symmetric tensor powers in the linear world by substituting linear notions for affine ones in the theory of Section 4, while removing any notion of flavor. Let W be a linear space. For each element $\mathbf{w}$ of W, let $Y_{\mathbf{w}}$ be a formal variable, and let $Y[W]$ be the resulting algebra of Y-polynomials. Define what it means for two Y-polynomials to be linearly equivalent, and let $\mathcal{Y}[W]$ be the resulting algebra of equivalence classes. Prove that symmetric, multilinear maps with domain W^n are equivalent to linear maps with domain $W^{\otimes n} := \mathcal{Y}^n[W]$.

There are other variants of the Homogenizing Principle that can move us from the affine world to the linear world at either the middle or the top level of Fig. 5.1. In each case, the homogenized map in the linear world has a property related to flavor whose effect is to guarantee that 1-flavored arguments are carried to 1-flavored results. The ***multilinear blossom*** $f_*: (P_*)^n \to Q_*$ of F is ***flavor-multiplicative***, meaning that $\operatorname{Flav}(f_*(e_1, \ldots, e_n)) = \prod_{i=1}^n \operatorname{Flav}(e_i)$. The homogeneous polynomial form $F_*: P_* \to Q_*$ of the polynomial map F is ***flavor-exponentiating***, meaning that $\operatorname{Flav}(F_*(e)) = \big(\operatorname{Flav}(e)\big)^n$.

Exercise 5.4. If P is an affine space, show that the two spaces $(P^n)_*$ and $(P_*)^n$ are quite different. Hence, the parentheses used above in describing the domain $(P_*)^n$ of the multilinear blossom f_* cannot be omitted.

We can get a more concrete perspective on our three principles by watching them in action on an explicit polynomial map, given in terms of coordinates. In particular, let us write down all six guises of the cubic polynomial curve $F: L \to Q$, sitting in a 3-space Q, that is given in Cartesian coordinates for L and Q by $F(\bar{u}) = F(\langle u \rangle) = \langle X(u), Y(u), Z(u) \rangle$ where

$$\begin{aligned} X(u) &:= 7u^3 + 6u^2 - 3u + 4 \\ Y(u) &:= 1u^3 + 3u^2 + 9u - 5 \\ Z(u) &:= 2u^3 - 3u^2 + 12u - 8. \end{aligned} \tag{5.5}$$

Moving down the left column of Fig. 5.1, the multiaffine blossom $f: L^3 \to Q$ of F will be the map $f(\bar{u}_1, \bar{u}_2, \bar{u}_3) = \langle x(u_1, u_2, u_3), y(u_1, u_2, u_3), z(u_1, u_2, u_3) \rangle$ given by

$$\begin{aligned} x(u_1, u_2, u_3) &:= 7u_1u_2u_3 + 2u_1u_2 + 2u_1u_3 + 2u_2u_3 - 1u_1 - 1u_2 - 1u_3 + 4 \\ y(u_1, u_2, u_3) &:= 1u_1u_2u_3 + 1u_1u_2 + 1u_1u_3 + 1u_2u_3 + 3u_1 + 3u_2 + 3u_3 - 5 \\ z(u_1, u_2, u_3) &:= 2u_1u_2u_3 - 1u_1u_2 - 1u_1u_3 - 1u_2u_3 + 4u_1 + 4u_2 + 4u_3 - 8. \end{aligned}$$

In general, as we saw in Section 2, blossoming a curve takes a term of the form cu^k in a polynomial of degree n and turns it into $\binom{n}{k}$ terms, each with coefficient $c/\binom{n}{k}$.

Continuing down the left column of Fig. 5.1, we next want to write a formula for the affine blossom $f^{\otimes}$. In order to do so, we must first choose a coordinate system for the third tensor power $L^{\otimes 3}$ of the affine line L. Let $\delta = \bar{1} - \bar{0}$ denote the unit vector on L. In this context, the natural coordinate system to use for $L^{\otimes 3}$ is the one that expresses a 1-flavored 3-tensor e in the form $e = \langle a, b, c \rangle = a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + \bar{0}^3$. We can convert any tensor e in $L^{\otimes 3}$ into this coordinate system by making use of the formula $\bar{u} = u\delta + \bar{0}$; for example, the product $\bar{u}_1\bar{u}_2\bar{u}_3$ of three points becomes

$$\bar{u}_1\bar{u}_2\bar{u}_3 = u_1u_2u_3\delta^3 + (u_1u_2 + u_1u_3 + u_2u_3)\delta^2\bar{0} + (u_1 + u_2 + u_3)\delta\bar{0}^2 + \bar{0}^3.$$

Comparing this expansion to our formula for the multiaffine blossom f, we deduce that the affine blossom $f^{\otimes}$ is given by $f^{\otimes}(e) = \langle x^{\otimes}(e), y^{\otimes}(e), z^{\otimes}(e)\rangle$, where

$$\begin{aligned} x^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + \bar{0}^3) &:= 7a + 2b - 1c + 4 \\ y^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + \bar{0}^3) &:= 1a + 1b + 3c - 5 \\ z^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + \bar{0}^3) &:= 2a - 1b + 4c - 8. \end{aligned}$$

The trick of rewriting a tensor on L in terms of the basis $\{\delta, \bar{0}\}$ for L_* is a generalization of Prop. 4.3 that is worth remembering. Recall that, in Prop. 4.3, we chose an affine frame $[\mathbf{r}_0, \ldots, \mathbf{r}_p]$ for P and then rewrote any tensor $[\Phi]$ on P as an X-polynomial $\rho(\Phi)$ in the corresponding frame variables $X_{\mathbf{r}_0}$ through $X_{\mathbf{r}_p}$. Instead of starting with an affine frame for P, we could start, more generally, with any linear basis $\{b_0, \ldots, b_p\}$ for P_*. In this more general setting, we don't get X-polynomials any more, because there are no X-variables associated with 1-tensors on P that aren't 1-flavored. But, if we treat the 1-tensors b_0 through b_p themselves as variables, it is still the case that every tensor e on P has a normal form $\beta(e)$ that is a polynomial in the variables b_0 through b_p. This normal form correspondence β is an algebra isomorphism between the tensor algebra $\mathcal{X}[P]$ and the algebra $\mathbf{R}[b_0, \ldots, b_p]$ of polynomials in the variables b_0 through b_p over the reals. The one thing to be careful about in this more general context is computing the flavor of a normal-form polynomial $\beta(e)$ in $\mathbf{R}[b_0, \ldots, b_p]$. So far, the flavor of any polynomial has been simply the sum of its coefficients, which corresponds to replacing each of its variables by 1 and then evaluating. To compute the flavor of a polynomial in $\mathbf{R}[b_0, \ldots, b_p]$, however, we replace each variable b_i by the flavor of the corresponding tensor b_i and then evaluate; this guarantees that $\mathrm{Flav}(\beta(e)) = \mathrm{Flav}(e)$.

Proposition 5.6. *Let $\{b_0, \ldots, b_p\}$ be any linear basis for the linearization P_* of an affine space P, and let β be the associated normal-form map, which rewrites each 1-tensor in some expression for a tensor e on P as a linear combination of the basis tensors b_0 through b_p. The normal-form map β is an algebra isomorphism between the symmetric tensor algebra $\mathcal{X}[P]$ of P and the algebra $\mathbf{R}[b_0, \ldots, b_p]$ of polynomials in the $p+1$ independent variables b_0 through b_p.* □

The bases $\{b_0, \ldots, b_p\}$ that are useful in practice tend to have elements b_i that are either points or vectors, that is, 1-tensors of flavor either 1 or 0. As we saw earlier, a barycentric coordinate system for P corresponds to a basis for P_* that contains only points, while a Cartesian coordinate system for P corresponds to a basis for P_* that contains one point and all the rest vectors. We shall refer to these types of bases for P_* in the future as ***barycentric bases*** and ***Cartesian bases***.

We return to the study of the explicit cubic curve $F: L \to Q$. In order to deal with the right-hand column in Fig. 5.1, we must extend our coordinate systems for L and Q to their linearizations L_* and Q_*. We shall do so by following our standard recipe for Cartesian bases: adding a flavor coordinate at the end. That is, we shall write a 1-tensor in L_* as $\langle u; r\rangle = u\delta + r\bar{0}$, where r is its flavor, and we shall write a 1-tensor in Q_* as $\langle x, y, z; w\rangle$, where w is the flavor. Note that we have $\bar{u} = \langle u; 1\rangle$ and $\delta = \langle 1; 0\rangle$. With these conventions, we can construct the homogenized form F_*

of F as follows: $F_*(\langle u;r\rangle) = \langle X_*(u;r), Y_*(u;r), Z_*(u;r); W_*(u;r)\rangle$ where

$$\begin{aligned} X_*(u;r) &:= 7u^3 + 6u^2r - \ \ 3ur^2 + 4r^3 \\ Y_*(u;r) &:= 1u^3 + 3u^2r + \ \ 9ur^2 - 5r^3 \\ Z_*(u;r) &:= 2u^3 - 3u^2r + 12ur^2 - 8r^3 \\ W_*(u;r) &:= 0u^3 + 0u^2r + \ \ 0ur^2 + 1r^3. \end{aligned} \tag{5.7}$$

We first set the output flavor $W_*(u;r)$ to be the constant 1. Then, we homogenize all four output coordinates by adding as many factors of the input flavor r to each term as necessary in order to bring that term up to total degree 3.

The multilinear blossom f_* is constructed analogously. We have $f_*(e_1,e_2,e_3) = \langle x_*(e_1,e_2,e_3), y_*(e_1,e_2,e_3), z_*(e_1,e_2,e_3); w_*(e_1,e_2,e_3)\rangle$ where $e_i = \langle u_i; r_i\rangle$ for i in $[1,3]$ and the four functions x_*, y_*, z_*, and w_* are given respectively by the rows of the table

$$\begin{array}{l} 7u_1u_2u_3 + 2u_1u_2r_3 + 2u_1r_2u_3 + 2r_1u_2u_3 - 1u_1r_2r_3 - 1r_1u_2r_3 - 1r_1r_2u_3 + 4r_1r_2r_3 \\ 1u_1u_2u_3 + 1u_1u_2r_3 + 1u_1r_2u_3 + 1r_1u_2u_3 + 3u_1r_2r_3 + 3r_1u_2r_3 + 3r_1r_2u_3 - 5r_1r_2r_3 \\ 2u_1u_2u_3 - 1u_1u_2r_3 - 1u_1r_2u_3 - 1r_1u_2u_3 + 4u_1r_2r_3 + 4r_1u_2r_3 + 4r_1r_2u_3 - 8r_1r_2r_3 \\ 0u_1u_2u_3 + 0u_1u_2r_3 + 0u_1r_2u_3 + 0r_1u_2u_3 + 0u_1r_2r_3 + 0r_1u_2r_3 + 0r_1r_2u_3 + 1r_1r_2r_3 \end{array}$$

Our last challenge is to write a formula for the linear blossom $f_*^{\otimes}$. To construct the requisite coordinate system for $L_*^{\otimes 3}$, we shall use the normal-form map β for tensors on L associated with the basis $\{\delta, \bar{0}\}$ of L_*: $e = \langle a,b,c;d\rangle$ where $\beta(e) = a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + d\bar{0}^3$. Using this coordinate system, we find that $f_*^{\otimes}(e) = \langle x_*^{\otimes}(e), y_*^{\otimes}(e), z_*^{\otimes}(e); w_*^{\otimes}(e)\rangle$, where

$$\begin{aligned} x_*^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + d\bar{0}^3) &:= 7a + 2b - 1c + 4d \\ y_*^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + d\bar{0}^3) &:= 1a + 1b + 3c - 5d \\ z_*^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + d\bar{0}^3) &:= 2a - 1b + 4c - 8d \\ w_*^{\otimes}(a\delta^3 + b\delta^2\bar{0} + c\delta\bar{0}^2 + d\bar{0}^3) &:= 0a + 0b + 0c + 1d. \end{aligned} \tag{5.8}$$

Note how the matrix of coefficients of F changes as we apply the various principles. Blossoming takes the column of coefficients of u^k and divides it up into $\binom{n}{k}$ equal parts. Tensoring eliminates the duplicate columns that blossoming introduced, by building in the symmetry condition. Homogenizing adds one more row, whose coefficients have a special unit form.

One advantage of the homogenized approach is that it allows us to interpret the columns of coefficients as geometric objects. In our original formula for F in Eq. (5.5), the constant column $\langle 4, -5, -8\rangle$ was obviously the point $F(\bar{0})$ in Q. But, even though the other columns also had three components, they weren't points in any obvious sense. Jumping to the linear blossom in Eq. (5.8), we find that its four 4-element columns of coefficients, from left to right, are just the following four 1-tensors on Q: $f_*^{\otimes}(\delta^3)$, $f_*^{\otimes}(\delta^2\bar{0})$, $f_*^{\otimes}(\delta\bar{0}^2)$, and $f_*^{\otimes}(\bar{0}^3)$. The first three are vectors on Q, since they are 0-flavored, while the fourth is a point in Q. If we rescale the argument tensor $\delta^k\bar{0}^{n-k}$ by $\binom{n}{k}$, we arrive at a geometric interpretation for the columns of coefficients of the homogeneous polynomial map F_* in Eq. (5.7). Comparing Eq. (5.7) to the original function F in Eq. (5.5), we conclude the following.

Proposition 5.9: The Power Basis for curves. *Let L be the affine line, and let δ be the unit vector on L. The n-tensors $\binom{n}{k}\bar{0}^{n-k}\delta^k$ for k in $[0,n]$ form a linear basis, called the* power basis, *for the space $L_*^{\otimes n}$. The coefficients of u^k in the formulas for the coordinates of the point $F(\bar{u})$ on a polynomial curve $F\colon L \to Q$ of degree n are precisely the non-flavor coordinates of the 1-tensor $f_*^{\otimes}\left(\binom{n}{k}\bar{0}^{n-k}\delta^k\right)$.* □

Thus, specifying a polynomial curve F by giving the coordinates of $F(\bar{u})$ as polynomials in u corresponds to specifying the linear blossom $f_*^{\otimes}$ by giving the images of the tensors in the power basis. The Bézier technique for specifying a polynomial curve isn't very different; it just uses a different basis, as the following restatement of Prop. 3.3 tells us.

Proposition 5.10: Bézier Bases for curves. *Let $[\bar{s},\bar{t}]$ be a reference interval in the affine line L. Then, the n-tensors $\bar{s}^{n-k}\bar{t}^k$ for k in $[0,n]$ form a linear basis, called* a Bézier basis, *for the space $L_*^{\otimes n}$. The images of the Bézier basis tensors under the linear blossom $f_*^{\otimes}$ are precisely the Bézier points of the curve segment $F([\bar{s},\bar{t}])$.* □

One important difference between the power basis and a Bézier basis is that all of the n-tensors in the Bézier basis are 1-flavored. Hence, a Bézier basis for $L_*^{\otimes n}$ is also an affine frame for $L^{\otimes n}$. Because of this property, we shall sometimes say ***Bézier frame*** instead of ***Bézier basis***. This also explains why the Bézier points $f^{\otimes}(\bar{s}^{n-k}\bar{t}^k)$ of a curve are values of the affine blossom $f^{\otimes}$, as well as being values of the linear blossom $f_*^{\otimes}$.

These insights all generalize from curves to surfaces quite straightforwardly. Let P be an affine 2-space, and let $\langle u,v\rangle$ be a Cartesian coordinate system on P. Let o denote the point $\langle 0,0\rangle$, let δ_u denote the unit vector on P in the u direction, and let δ_v denote the unit vector in the v direction; then, $\langle u,v\rangle = \mathrm{o} + u\delta_u + v\delta_v$. The ***power basis*** for the space $P_*^{\otimes n}$ is the set of tensors of the form $\binom{n}{i\ j\ k}\delta_u^i\delta_v^j\mathrm{o}^k$, where i, j, and k are nonnegative integers that sum to n. If $F\colon P \to Q$ is a polynomial surface of degree n, the coefficients of u^iv^j in the formulas for the coordinates of the point $F(\langle u,v\rangle)$ are precisely the non-flavor coordinates of the tensor $f_*^{\otimes}\left(\binom{n}{i\ j\ k}\delta_u^i\delta_v^j\mathrm{o}^k\right)$. Alternatively, as described in Prop. 3.8, the Bézier basis for $P_*^{\otimes n}$ consists of the tensors $\mathbf{r}^i\mathbf{s}^j\mathbf{t}^k$, where $\triangle\mathbf{rst}$ is a reference triangle for P.

6. The de Casteljau technique for specifying curves

By means of blossoming, tensoring, and homogenizing, we have found that specifying a polyonomial curve $F\colon L \to Q$ of degree n is the same thing as specifying its linear blossom, which is a flavor-preserving linear function $f_*^{\otimes}\colon L_*^{\otimes n} \to Q_*$. The obvious way to specify a linear function is to give its values on a basis for its domain. Using the power basis $\{\binom{n}{k}\delta^{n-k}\bar{0}^k\}$ for the domain $L_*^{\otimes n}$, we just learned, corresponds to the elementary approach of specifying the curve F by giving the coordinates of the point $F(\bar{u})$ as polynomials in u. Using a Bézier basis $\{\bar{s}^{n-k}\bar{t}^k\}$

for $L_*^{\otimes n}$ corresponds to specifying the curve F by giving the Bézier points of the segment $F([\bar{s},\bar{t}])$. But any other basis for $L_*^{\otimes n}$ would do just as well. In this section, we shall investigate a class of bases that might be called *de Casteljau bases*, because they will enable us to specify a curve by specifying a collection of blossom values (poles, in de Casteljau's terminology [14]) of a fairly general form. As in the case of a Bézier basis, the n-tensors in a de Casteljau basis are always 1-flavored. Thus, a de Casteljau basis can also be viewed as an affine frame for $L^{\otimes n}$, and, in that role, will be called a *de Casteljau frame.*

Let $(\bar{r}_1, \ldots, \bar{r}_{2n})$ be a reference sequence of $2n$ points in the parameter space L. For k in $[0,n]$, let e_k denote the 1-flavored n-tensor $e_k = \bar{r}_{k+1} \cdots \bar{r}_{k+n}$ on L; that is, each e_k is the tensor product of n consecutive points $\bar{r}_i$ from the sequence $(\bar{r}_1, \ldots, \bar{r}_{2n})$. The idea behind de Casteljau frames is to use the $n+1$ tensors e_k for k in $[0,n]$ as an affine frame for $L^{\otimes n}$. Note that the Bézier frame with reference interval $[\bar{s},\bar{t}]$ is a special case of a de Casteljau frame in which the first n points of the reference sequence, $\bar{r}_1$ through $\bar{r}_n$, are all equal to $\bar{s}$ and the last n points, $\bar{r}_{n+1}$ through $\bar{r}_{2n}$, are all equal to $\bar{t}$. Of course, not all reference sequences are legal; we have to put some constraints on the points $\bar{r}_i$ in order to guarantee that the tensors e_k will be affinely independent, and will hence form a frame.

Lemma 6.1: de Casteljau frames. *Let $(\bar{r}_1, \ldots, \bar{r}_{2n})$ be a reference sequence of $2n$ points in the affine line L. Then, the n-tensors $e_k := \bar{r}_{k+1} \cdots \bar{r}_{k+n}$ for k in $[0,n]$ form an affine frame for the tensor power space $L^{\otimes n}$ if and only if we have $r_j \neq r_{n+i}$ for $1 \leq i \leq j \leq n$.*

Proof. (Note that the stated condition does hold in the case of a Bézier frame, since $r_j = s$, $r_{n+i} = t$, and $s < t$.)

We shall tackle the easier half of the proof first. Suppose that $r_j = r_{n+i}$ for some i and j with $1 \leq i \leq j \leq n$. We must show that the n-tensors e_k for k in $[0,n]$ fail to be affinely independent. In fact, we shall show that there is an affine dependence among the $j-i+2$ tensors $\{e_{i-1}, \ldots, e_j\}$. If we let q denote the common value of r_j and r_{n+i}, we can write out the tensors e_{i-1} through e_j as the rows of a parallelogram with constant columns:

$$\begin{array}{ccccccccccc}
\bar{r}_i & \bar{r}_{i+1} & \cdots & \bar{q} & \bar{r}_{j+1} & \cdots & \bar{r}_{n+i-1} & & & & \\
 & \bar{r}_{i+1} & \cdots & \bar{q} & \bar{r}_{j+1} & \cdots & \bar{r}_{n+i-1} & \bar{q} & & & \\
 & & \ddots & \vdots & \vdots & & \vdots & \vdots & \ddots & & \\
 & & & \bar{q} & \bar{r}_{j+1} & \cdots & \bar{r}_{n+i-1} & \bar{q} & \cdots & \bar{r}_{n+j-1} & \\
 & & & & \bar{r}_{j+1} & \cdots & \bar{r}_{n+i-1} & \bar{q} & \cdots & \bar{r}_{n+j-1} & \bar{r}_{n+j}
\end{array}$$

Note that each row includes all of the points $\bar{r}_{j+1}$ through $\bar{r}_{n+i-1}$ as factors. In addition, each row also includes at least one $\bar{q}$. Therefore, every row has the form

$$\bar{u}_1 \cdots \bar{u}_{j-i}\, \bar{q}\, \bar{r}_{j+1} \cdots \bar{r}_{n+i-1}$$

for some points $\bar{u}_1$ through $\bar{u}_{j-i}$. But, as we vary $\bar{u}_1$ through $\bar{u}_{j-i}$ in L, the resulting tensors all lie in an affine flat of dimension $(j-i)$ in $L^{\otimes n}$. Since the rows above denote $j-i+2$ tensors that all lie in that flat, they must be affinely dependent.

In the other direction, suppose that $r_j \neq r_{n+i}$ for $1 \leq i \leq j \leq n$. We must show that the tensors e_k are affinely independent as elements of $L^{\otimes n}$, or, equivalently, that they are linearly independent as elements of the linearization $L_*^{\otimes n}$. To test this latter condition, we shall expand the tensors e_k into coordinates in terms of a known basis for $L_*^{\otimes n}$ and then evaluate a determinant. As our known basis, we will use the tensors $\bar{0}^{n-l}\delta^l$ for l in $[0, n]$, where δ denotes the unit vector on L—that is, the tensors of the power basis, but without the binomial coefficient scaling factor.

Expanding the tensor e_k in terms of this known basis, we have

$$\begin{aligned} e_k = \bar{r}_{k+1} \cdots \bar{r}_{k+n} &= \prod_{1 \leq i \leq n} (\bar{0} + r_{k+i}\delta) \\ &= \sum_{0 \leq l \leq n} \sigma_l(r_{k+1}, \ldots, r_{k+n}) \bar{0}^{n-l}\delta^l, \end{aligned}$$

where σ_l denotes the lth elementary symmetric function

$$\sigma_l(u_1, \ldots, u_n) := \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I| = l}} \prod_{i \in I} u_i.$$

Therefore, the tensors e_k will be independent if and only if $\det(M) \neq 0$ where $M = (m_{kl})$ is the $(n+1)$-by-$(n+1)$ matrix given by $m_{kl} = \sigma_l(r_{k+1}, \ldots, r_{k+n})$ for k and l in $[0, n]$.

To compute $\det(M)$, we shall use the same trick that is frequently used to compute the Vandermonde determinant. Since $\sigma_l(r_{k+1}, \ldots, r_{k+n})$ is a homogeneous polynomial of degree l in the variables r_i, the quantity $\det(M)$ must be given by a homogeneous polynomial of degree $\binom{n+1}{2}$ in the variables r_i. On the other hand, we already know $\binom{n+1}{2}$ distinct linear factors of $\det(M)$, since we know, from the first half of this proof, that $\det(M) = 0$ whenever $r_j = r_{n+i}$ for $1 \leq i \leq j \leq n$. Hence, we must have

$$\det(M) = w \prod_{1 \leq i \leq j \leq n} (r_{n+i} - r_j) \tag{6.2}$$

for some real constant w. To determine w, consider the coefficient of the term $T := \prod_{1 \leq i \leq n} r_{n+i}^{n-i+1}$. In the right-hand-side of Eq. (6.2), the coefficient of T is clearly 1, since we must choose the r_{n+i} term out of each binomial factor. In $\det(M)$, the only way to get the term T is to go down the main diagonal of M, taking the term $r_{n+1} \cdots r_{n+k}$ from each $m_{kk} = \sigma_k(r_{k+1}, \ldots, r_{k+n})$; hence, we again get a coefficient of 1, showing that $w = 1$. We conclude that the e_k can be dependent only if at least one of the equalities $r_j = r_{n+i}$ holds. □

Corollary 6.3: The de Casteljau technique for curves. *Let $r_1, \ldots, r_{2n}$ be real numbers satisfying the condition that $r_j \neq r_{n+i}$ for $1 \leq i \leq j \leq n$. Polynomial curves $F: L \to Q$ of degree n are in one-to-one correspondence with $(n+1)$-tuples $(\mathbf{x}_0, \ldots, \mathbf{x}_n)$ of points in Q under the correspondence $f(\bar{r}_{k+1}, \ldots, \bar{r}_{k+n}) = \mathbf{x}_k$ for k in $[0, n]$, where f denotes the multiaffine blossom of F.* □

In the de Casteljau technique, the curve F is controlled by specifying the points $f_*^{\otimes}(e_k) = f(\bar{r}_{k+1}, \ldots, \bar{r}_{k+n})$, which, following de Casteljau, we shall call *poles*. When we study spline curves in Section 15, we will find that specifying curve segments by giving their poles with respect to reference sequences that are substrings of the knot sequence will make it easy to assemble those segments into a spline curve. In the context of spline curves, however, these poles have another name; they are known as the *de Boor points* of the spline curve [5].

Challenge 6.4. Is the Tensoring Principle worth it? By using the Tensoring Principle in this section, we were able to reduce the study of de Casteljau frames to a straightforward problem in linear algebra: testing linear independence. As we shall find in Section 15, this means that we have already finished the hardest part of our development of the theory of spline curves. On the other hand, it took us quite a bit of work to construct the tensor power space $L^{\otimes n}$. Contrast the tensoring approach with more standard developments of the de Boor theory of spline curves.

Part C: Perspectives on Blossoming

What do the n different arguments to a blossom mean? Can we think of the blossom value $f(\bar{u}_1, \ldots, \bar{u}_n)$ as the result of evaluating the n-ic curve $F(\bar{u})$ at n different times? In the next three sections, we shall study this question from three perspectives: algorithmic, differential, and algebraic.

7. The algorithmic view of a blossom's n arguments

From an algorithmic perspective, the meaning of the original time parameter $\bar{u}$ of a curve $F(\bar{u})$ is a choice of interpolation ratio. Given the Bézier points of the curve segment $F([\bar{s}, \bar{t}])$ and given the ratio of the lengths of the line segments into which the point $\bar{u}$ divides the segment $[\bar{s}, \bar{t}]$, the de Casteljau Algorithm tells us how to construct the point $F(\bar{u}) = f(\bar{u}, \ldots, \bar{u})$ by performing n stages of linear interpolations with that ratio.

It makes perfect sense, however, to use different ratios for the different stages of the de Casteljau Algorithm. If we use the ratio corresponding to $\bar{u}_i$ during the ith stage, the point that we end up constructing is precisely the blossom value $f(\bar{u}_1, \ldots, \bar{u}_n)$. In more detail, the ith stage produces, as output, all of the blossom values whose argument bags consist of the i points $\bar{u}_1$ through $\bar{u}_i$, filled out with some collection of $(n-i)$ copies of either $\bar{s}$ or $\bar{t}$. Switching from the multiaffine blossom f to the affine blossom $F^{\otimes}$ in order to avoid horizontal braces, the ith stage computes the point $f^{\otimes}(\bar{u}_1 \cdots \bar{u}_i \bar{s}^{n-i-k} \bar{t}^k)$ for k in $[0, n-i]$ by using the recurrence

$$f^{\otimes}(\bar{u}_1 \cdots \bar{u}_i \bar{s}^{n-i-k} \bar{t}^k) = \tag{7.1}$$
$$\left(\frac{t-u_i}{t-s}\right) f^{\otimes}(\bar{u}_1 \cdots \bar{u}_{i-1} \bar{s}^{n-i-k+1} \bar{t}^k) + \left(\frac{u_i - s}{t-s}\right) f^{\otimes}(\bar{u}_1 \cdots \bar{u}_{i-1} \bar{s}^{n-i-k} \bar{t}^{k+1})$$

to interpolate between two adjacent outputs of the previous stage. For example, the left half of Fig. 7.2 shows the construction of the point $f(\bar{4}, \bar{3}, \bar{2}) = f(\bar{2}, \bar{3}, \bar{4})$ from the Bézier points of a cubic segment $F([\bar{0}, \bar{6}])$ by interpolating with the ratios 2/3, 1/2, and 1/3, in that order.

The symmetry of the blossom f implies that using the same bag of ratios in a different order will result in the same final point, even though the intermediate points and lines will be different. For example, the right half of Fig. 7.2 uses the ratios in the order 1/3, 2/3, and 1/2. Thus, the algorithmic way to interpret the n arguments of a curve's blossom is that each of them provides the ratio to use during one of the n stages of the de Casteljau Algorithm.

This insight works just as well for surfaces as it does for curves. If $F\colon P \to Q$ is a polynomial surface, any point $\mathbf{u}$ in P specifies a two-dimensional interpolation ratio. That is, the location of $\mathbf{u}$ with respect to the vertices of the reference triangle for P gives a way of choosing a corresponding point in the plane determined by any three points. Given the Bézier points of F, the de Casteljau Algorithm performs n stages of these two-dimensional interpolations in order to construct the point $F(\mathbf{u})$. Once again, it makes perfect sense to use different interpolation ratios during the different stages, and the final result does not depend on the order we choose. Ronald N. Goldman gets the credit for realizing that two-dimensional interpolation stages with different ratios commute with each other [25].

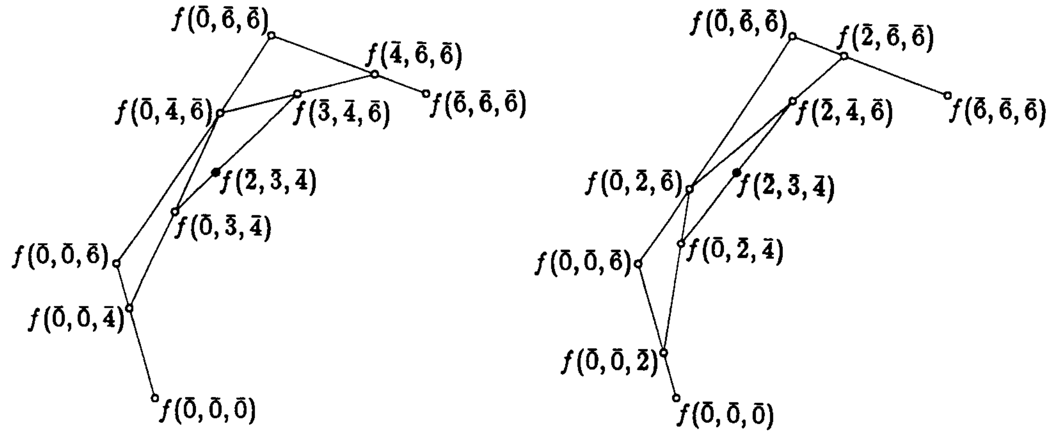

$f(\bar{0},\bar{6},\bar{6})$
$f(\bar{4},\bar{6},\bar{6})$
$f(\bar{0},\bar{4},\bar{6})$
$f(\bar{3},\bar{4},\bar{6})$
$f(\bar{6},\bar{6},\bar{6})$
$f(\bar{2},\bar{3},\bar{4})$
$f(\bar{0},\bar{3},\bar{4})$
$f(\bar{0},\bar{0},\bar{6})$
$f(\bar{0},\bar{0},\bar{4})$
$f(\bar{0},\bar{0},\bar{0})$
$f(\bar{0},\bar{6},\bar{6})$
$f(\bar{2},\bar{6},\bar{6})$
$f(\bar{2},\bar{4},\bar{6})$
$f(\bar{6},\bar{6},\bar{6})$
$f(\bar{0},\bar{2},\bar{6})$
$f(\bar{2},\bar{3},\bar{4})$
$f(\bar{0},\bar{2},\bar{4})$
$f(\bar{0},\bar{0},\bar{6})$
$f(\bar{0},\bar{0},\bar{2})$
$f(\bar{0},\bar{0},\bar{0})$

Fig. 7.2. Different ratios in different stages of the de Casteljau Algorithm

The same insight also applies to curves controlled with the de Casteljau technique of Section 6, in which poles are specified instead of Bézier points. Recall that, in that technique, an n-ic curve F is controlled by specifying the blossom values $f(\bar{r}_{k+1}, \ldots, \bar{r}_{k+n})$ for k in $[0, n]$, where $(\bar{r}_1, \ldots, \bar{r}_{2n})$ is a sequence of $2n$ points in L satisfying the nondegeneracy conditions $r_j \neq r_{n+i}$ for $1 \leq i \leq j \leq n$. The output of the ith stage of the de Casteljau Algorithm in this case is the collection of blossom values $f(\bar{u}_1, \ldots, \bar{u}_i, \bar{r}_{k+i+1}, \ldots, \bar{r}_{k+n})$ for k in $[0, n-i]$. The ith stage produces this output by using the formula

$$\begin{aligned} f(\bar{u}_1, \ldots, \bar{u}_i, \bar{r}_{k+i+1}, \ldots, \bar{r}_{k+n}) = & \\ & \left(\frac{r_{k+n+1} - u_i}{r_{k+n+1} - r_{k+i}}\right) f(\bar{u}_1, \ldots, \bar{u}_{i-1}, \bar{r}_{k+i}, \ldots, \bar{r}_{k+n}) + \\ & \left(\frac{u_i - r_{k+i}}{r_{k+n+1} - r_{k+i}}\right) f(\bar{u}_1, \ldots, \bar{u}_{i-1}, \bar{r}_{k+i+1}, \ldots, \bar{r}_{k+n+1}) \end{aligned} \tag{7.3}$$

to interpolate between two adjacent outputs of the previous stage. Just as in the Bézier case, each blossom argument $\bar{u}_i$ controls the interpolations that happen during one stage, and the arguments can be used in any order without altering the final result. We shall refer to the algorithm embodied in Eq. (7.3) as the *Generalized de Casteljau Algorithm*, to distinguish it from the special case in Eq. (7.1). When the Generalized de Casteljau Algorithm arises in the context of spline curves, however, we shall follow standard practice by referring to it as the *de Boor Algorithm* [5].

Note that the different interpolations that make up the ith stage of the Generalized de Casteljau Algorithm involve different ratios. In particular, the kth interpolation of the ith stage is controlled by the location of $\bar{u}_i$ with respect to the interval $[\bar{r}_{k+i}, \bar{r}_{k+n+1}]$.

Note also that the nondegeneracy conditions on the $\bar{r}_i$ are exactly what is needed to guarantee that none of the $\binom{n+1}{2}$ denominators that arise when applying Eq. (7.3) is zero. This observation provides an alternative proof of Lemma 6.1 that is computationally easier than our determinant-based proof in Section 6. If

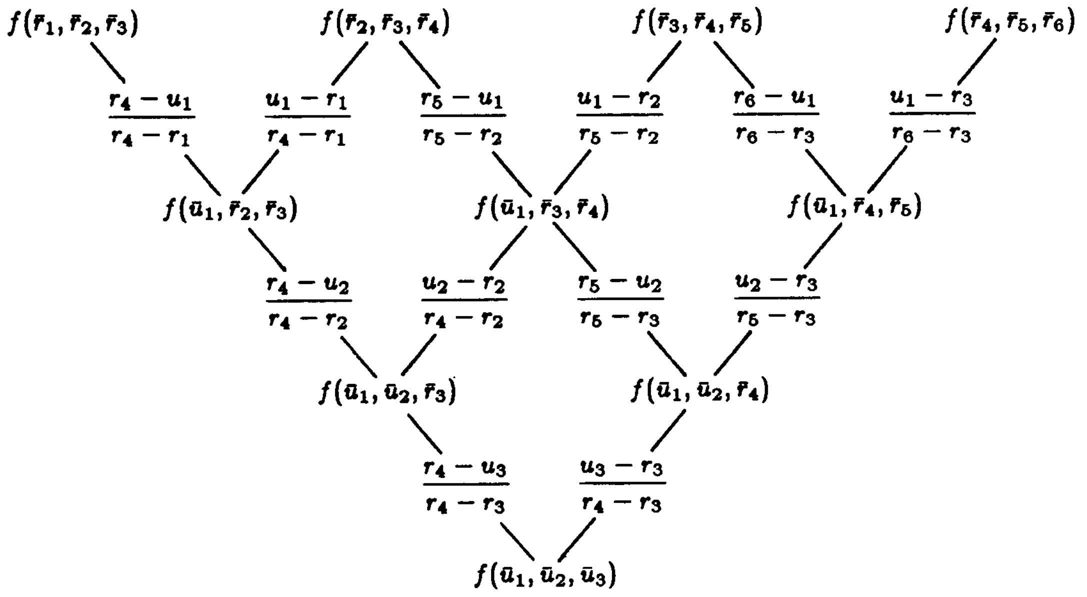
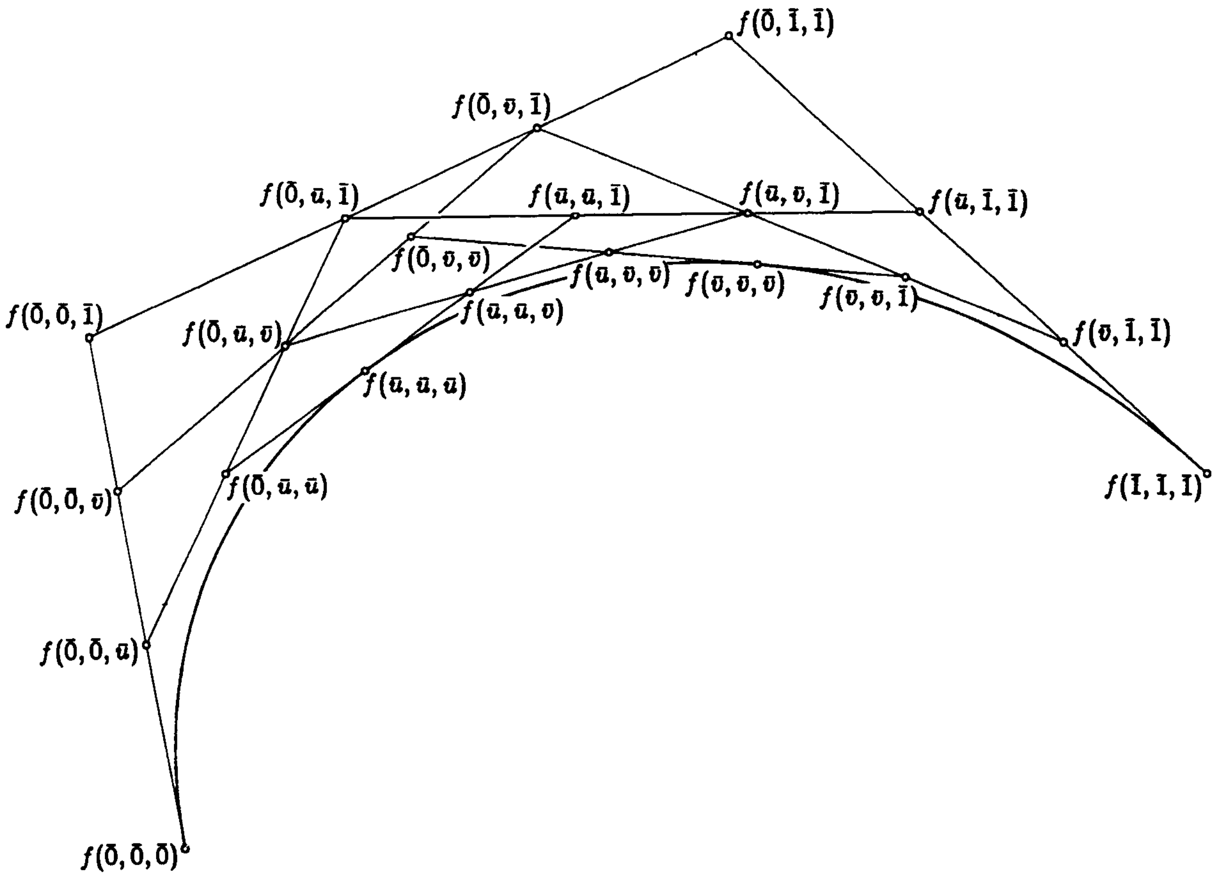

$f(\bar{r}_1,\bar{r}_2,\bar{r}_3)$
$f(\bar{r}_2,\bar{r}_3,\bar{r}_4)$
$f(\bar{r}_3,\bar{r}_4,\bar{r}_5)$
$f(\bar{r}_4,\bar{r}_5,\bar{r}_6)$
$\frac{r_4-u_1}{r_4-r_1}$
$\frac{u_1-r_1}{r_4-r_1}$
$\frac{r_5-u_1}{r_5-r_2}$
$\frac{u_1-r_2}{r_5-r_2}$
$\frac{r_6-u_1}{r_6-r_3}$
$\frac{u_1-r_3}{r_6-r_3}$
$f(\bar{u}_1,\bar{r}_2,\bar{r}_3)$
$f(\bar{u}_1,\bar{r}_3,\bar{r}_4)$
$f(\bar{u}_1,\bar{r}_4,\bar{r}_5)$
$\frac{r_4-u_2}{r_4-r_2}$
$\frac{u_2-r_2}{r_4-r_2}$
$\frac{r_5-u_2}{r_5-r_3}$
$\frac{u_2-r_3}{r_5-r_3}$
$f(\bar{u}_1,\bar{u}_2,\bar{r}_3)$
$f(\bar{u}_1,\bar{u}_2,\bar{r}_4)$
$\frac{r_4-u_3}{r_4-r_3}$
$\frac{u_3-r_3}{r_4-r_3}$
$f(\bar{u}_1,\bar{u}_2,\bar{u}_3)$

Fig. 7.4. The Generalized de Casteljau Algorithm when $n = 3$

$r_j \neq r_{n+i}$ for $1 \le i \le j \le n$, Eq. (7.3) with the f's omitted shows that every simple n-tensor $\bar{u}_1 \cdots \bar{u}_n$ can be expressed as an affine combination of the tensors $e_k = \bar{r}_{k+1} \cdots \bar{r}_{k+n}$ for k in $[0, n]$. Since the simple n-tensors span all of $L^{\otimes n}$, that is enough to conclude that the $\{e_k\}$ form an affine frame for $L^{\otimes n}$. The determinant-based proof in Section 6 is conceptually simpler, however, in the sense that it attacks a problem in linear algebra with the standard techniques of linear algebra, without recourse to an auxiliary algorithm.

Fig. 7.4 shows the computation scheme of the Generalized de Casteljau Algorithm in the particular case $n = 3$ in the form of a triangular array of blossom values. The four blossom values across the top are the poles of a cubic polynomial curve F with respect to a reference sequence $(\bar{r}_1, \ldots, \bar{r}_6)$. The points in lower rows are computed as affine combinations of their two ancestors with the indicated weights, as Eq. (7.3) specifies. It is not at all obvious, from looking at this computation scheme, that it computes a function $f(\bar{u}_1, \bar{u}_2, \bar{u}_3)$ that is symmetric.

Just as the de Casteljau Algorithm can help us to understand blossoms, we can also use blossoms to help understand the de Casteljau Algorithm. For example, let F be a cubic curve, and suppose that we have used the de Casteljau Algorithm to construct the point $F(\bar{v}) = f(\bar{v}, \bar{v}, \bar{v})$ for some $\bar{v}$ in $[\bar{0}, \bar{1}]$ from the Bézier points of the segment $F([\bar{0}, \bar{1}])$. In the process of doing so, we will also have constructed the points $f(\bar{0}, \bar{0}, \bar{v})$ and $f(\bar{0}, \bar{v}, \bar{v})$. One of the important properties of the Bézier-de Casteljau theory is that the four points $f(\bar{0}, \bar{0}, \bar{0})$, $f(\bar{0}, \bar{0}, \bar{v})$, $f(\bar{0}, \bar{v}, \bar{v})$, and $f(\bar{v}, \bar{v}, \bar{v})$ are, in fact, the Bézier points of the initial segment $F([\bar{0}, \bar{v}])$ of $F([\bar{0}, \bar{1}])$. Viewed algorithmically, this corresponds to the fact that two quite different construction processes always give the same result: For any $\bar{u}$ in $[\bar{0}, \bar{v}]$, one way to compute $F(\bar{u})$ starts with the Bézier points of the entire curve segment $F([\bar{0}, \bar{1}])$ and uses the de



$f(\bar{0},\bar{1},\bar{1})$
$f(\bar{0},\bar{v},\bar{1})$
$f(\bar{0},\bar{u},\bar{1})$
$f(\bar{u},\bar{u},\bar{1})$
$f(\bar{u},\bar{v},\bar{1})$
$f(\bar{u},\bar{1},\bar{1})$
$f(\bar{0},\bar{v},\bar{v})$
$f(\bar{u},\bar{v},\bar{v})$
$f(\bar{v},\bar{v},\bar{v})$
$f(\bar{v},\bar{v},\bar{1})$
$f(\bar{0},\bar{0},\bar{1})$
$f(\bar{0},\bar{u},\bar{v})$
$f(\bar{u},\bar{u},\bar{v})$
$f(\bar{v},\bar{1},\bar{1})$
$f(\bar{u},\bar{u},\bar{u})$
$f(\bar{0},\bar{0},\bar{v})$
$f(\bar{0},\bar{u},\bar{u})$
$f(\bar{1},\bar{1},\bar{1})$
$f(\bar{0},\bar{0},\bar{u})$
$f(\bar{0},\bar{0},\bar{0})$

Fig. 7.5. Repeated subdivision of a cubic curve

Casteljau Algorithm with ratio $0 : u : 1$. Alternatively, we could start with the four purported Bézier points of the initial subsegment $F([\bar{0},\bar{v}])$ and use the de Casteljau Algorithm with ratio $0 : u : v$. Thinking algorithmically, it isn't very clear why these two computations should produce the same answer. But, once we understand about blossoms, we can draw Fig. 7.5, which clarifies the situation.

Exercise 7.6. There are two intersections of line segments in Fig. 7.5 that are not labeled as blossom values. Convince yourself that those two intersections arise only because the four Bézier points of the segment $F([0,1])$ are coplanar, causing the resulting cubic F to be degenerate. In fact, those two intersections are drawn in Fig. 7.5 so as to suggest that one of the lines is above the other in each case. If the first three Bézier points of F are assumed to lie in the plane of the paper, where should the fourth Bézier point $f(\bar{1},\bar{1},\bar{1})$ be placed with respect to the plane of the paper to result in the crossings as drawn?

Exercise 7.7. Consider quadratic polynomial curves, that is, parabolas. Note that the reference sequence $(\bar{1},\bar{0},\bar{3},\bar{2})$ of length 4 satisfies the nondegeneracy conditions of the de Casteljau technique with $n = 2$. Hence, we can specify a parabola F

uniquely by giving arbitrary values to the three poles $f(\bar{1},\bar{0})$, $f(\bar{0},\bar{3})$, and $f(\bar{3},\bar{2})$. Sketch an example. Construct the blossom value $f(\bar{1},\bar{2})$ from the three given poles in two different ways. What does this have to do with the well-known fact that the centroid of a triangle is two-thirds of the way from any vertex to the midpoint of the opposite side?

8. The differential view of a blossom's n arguments

Recall that, for a quadratic curve $F(\bar{u})$, the blossom value $f(\bar{s},\bar{t})$ is simply the intersection of the tangent lines to the parabola $F(\bar{u})$ at $u = s$ and at $u = t$. For a second perspective on the meaning of the n arguments of a blossom, we would like to generalize this observation. In order to do so, we must study how differentiating interacts with blossoming. This is one area where the Homogenizing Principle is quite helpful.

As a warmup, let $F: L \to Q$ be a cubic polynomial curve. One possible correct formula for the derivative $F'(\bar{u})$ in terms of the blossom f is

$$F'(\bar{u}) = 3\big(f(\overline{u+1},\bar{u},\bar{u}) - f(\bar{u},\bar{u},\bar{u})\big). \tag{8.1}$$

Since f is affine in its first argument, there are lots of other ways to write this, including $F'(\bar{u}) = f(\overline{u+3},\bar{u},\bar{u}) - f(\bar{u},\bar{u},\bar{u})$ and $F'(\bar{u}) = 3\big(f(\overline{v+1},\bar{u},\bar{u}) - f(\bar{v},\bar{u},\bar{u})\big)$ for any v in L. Note that, when we subtract one point from another point in an affine space such as Q, the result of the subtraction is a vector, not a point. Recall from Section 5 that, associated with any affine space Q, there is a linear space Q_0 of vectors on Q; the spaces Q_0 and $Q = Q_1$ are parallel hyperplanes in the linearization Q_* of Q. The derivative of a curve F is a vector-valued map $F': L \to Q_0$.

The easiest way to prove Eq. (8.1) is to invoke more of the technology of Section 5. We learned in that section that a polynomial curve $F: L \to Q$ of degree n is equivalent to a flavor-preserving linear map $f_*^{\otimes}: L_*^{\otimes n} \to Q_*$ under the correspondence $F(\bar{u}) = f_*^{\otimes}(\bar{u}^n)$. In particular, we can consider the curve F to be the composition of two functions: the nth-power map $\varphi: L \to L^{\otimes n}$, which takes a point $\bar{u}$ to the 1-flavored n-tensor $\varphi(\bar{u}) := \bar{u}^n$ on L, followed by the linear map $f_*^{\otimes}$. Since linear maps commute with differentiation, we conclude that $F'(\bar{u}) = f_*^{\otimes}(\varphi'(\bar{u}))$. The derivative of φ is easy to compute using the Power Rule:

$$\varphi'(\bar{u}) = \frac{d}{du}\bar{u}^n = n\bar{u}^{n-1}\frac{d\bar{u}}{du} = n\bar{u}^{n-1}\delta,$$

where δ denotes the unit vector on L. Therefore, we have $F'(\bar{u}) = f_*^{\otimes}(n\bar{u}^{n-1}\delta) = nf_*^{\otimes}(\bar{u}^{n-1}\delta)$. To translate this answer back from the linear blossom $f_*^{\otimes}$ to the multilinear blossom f_*, we have to resort to a horizontal brace:

$$F'(\bar{u}) = nf_*(\delta,\underbrace{\bar{u},\ldots,\bar{u}}_{n-1}).$$

In order to translate back to the affine world, we must choose some way of writing the vector δ as a difference of points. If we choose the substitution $\delta := \overline{u+1} - \bar{u}$, we get $F'(\bar{u}) = n\big(f_*^{\otimes}(\overline{u+1}\,\bar{u}^{n-1}) - f_*^{\otimes}(\bar{u}^n)\big)$, or equivalently,

$$F'(\bar{u}) = n\big(f(\overline{u+1},\underbrace{\bar{u},\ldots,\bar{u}}_{n-1}) - f(\underbrace{\bar{u},\ldots,\bar{u}}_{n})\big),$$

which is the generalization of Eq. (8.1) to arbitrary degrees n.

The good news is that differentiating is quite easy in the linear world: just substitute the unit vector δ for one of the copies of the argument point $\bar{u}$. The bad news is the annoying factor of n. That factor of n arises because, when we follow the curve $F(\bar{u})$, we vary all n of the arguments of the multiaffine blossom $f(\bar{u},\ldots,\bar{u})$ in parallel. Suppose that we held all but one argument of f fixed at $\bar{u}$ while we varied that single argument. The resulting function $G(\bar{v}) := f(\bar{v},\bar{u},\ldots,\bar{u})$ is affine, and its (constant) first derivative is precisely $f_*^\otimes(\bar{u}^{n-1}\delta)$, without the factor of n. The effect, to first order, of varying all n arguments in parallel is to cause the resulting point to move n times faster.

By repeating the argument above, we can compute derivatives of any order. The *falling factorial power* notation is helpful in this context: let $n^{\underline{k}}$ denote the product $n(n-1)\cdots(n-k+1)$.

Proposition 8.2. *Let $F\colon L \to Q$ be an n-ic polynomial curve, and let $f_*^\otimes\colon L_*^{\otimes n} \to Q_*$ be the linear blossom of F. The kth derivative of F is given by the formula*

$$F^{(k)}(\bar{u}) = n^{\underline{k}} f_*^\otimes(\bar{u}^{n-k}\delta^k). \quad \square$$

Note that this formula gives the correct answer when $k = 0$, since $n^{\underline{0}} = 1$. For $k > n$, the factor of $n^{\underline{k}}$ takes the right-hand side to 0, which is also correct. If we want to replace the linear blossom $f_*^\otimes$ on the right-hand side with the multiaffine blossom f, we can just replace each factor of δ by an equivalent difference of points, such as $\overline{u+1} - \bar{u}$. This leads to the formula

$$F^{(k)}(\bar{u}) = n^{\underline{k}} \sum_{0\le j\le k} \binom{k}{j}(-)^{k-j} f(\underbrace{\overline{u+1},\ldots,\overline{u+1}}_{j},\underbrace{\bar{u},\ldots,\bar{u}}_{n-j}),$$

which expresses the kth derivative of F as a kth difference of values of the multiaffine blossom f. If we are interested in the endpoint derivatives $F^{(k)}(\bar{s})$ of the curve segment $F([\bar{s},\bar{t}])$, we could use the substitution $\delta := (\bar{t}-\bar{s})/(t-s)$ instead, which leads to the well-known formula

$$F^{(k)}(\bar{s}) = \frac{n^{\underline{k}}}{(t-s)^k} \sum_{0\le j\le k} \binom{k}{j}(-)^{k-j} f(\underbrace{\bar{s},\ldots,\bar{s}}_{n-j},\underbrace{\bar{t},\ldots,\bar{t}}_{j}) \tag{8.3}$$

for an endpoint derivative as a kth difference of the Bézier points.

Prop. 8.2 gives us another way to prove Prop. 5.9, which related the coefficients of the polynomials that define a curve $F(\bar{u})$ to the effect of the linear blossom $f_*^\otimes$ on the tensors in the power basis for $L_*^{\otimes n}$. From the theory of Taylor series, we know that the coefficient of u^k in the polynomial expansion for $F(\bar{u})$ is the vector $F^{(k)}(\bar{0})/k!$. Applying Prop. 8.2, we can rewrite that quantity as $(n^{\underline{k}}/k!)f_*^\otimes(\bar{0}^{n-k}\delta^k)$, which is precisely the image of the kth element $\binom{n}{k}\bar{0}^{n-k}\delta^k$ of the power basis under the linear blossom $f_*^\otimes$.

In order to state the analogous formulas for surfaces, we have to agree on a notation for the derivatives of a function $F\colon P \to Q$ whose domain is 2-dimensional.

If $\mathbf{u}$ is a point in P, let $dF_{\mathbf{u}}: P_0 \to Q_0$, given by

$$dF_{\mathbf{u}}(\xi) := \lim_{h \to 0} \frac{F(\mathbf{u} + h\xi) - F(\mathbf{u})}{h},$$

denote the linear map that approximates the local behavior of F to first order near $\mathbf{u}$. This map is variously called either the ***derivative*** of F at $\mathbf{u}$ or the ***differential*** of F at $\mathbf{u}$. The quantity $dF_{\mathbf{u}}(\xi)$ is also frequently written $D_\xi F(\mathbf{u})$ and, in those contexts, it is called ***the directional derivative*** of F in the ξ direction at the point $\mathbf{u}$. Which notation one uses doesn't make much difference; we will arbitrarily choose the former for now. The kth derivative of F at the point $\mathbf{u}$ is a symmetric k-linear map $d^kF_{\mathbf{u}}: (P_0)^k \to Q_0$, and a value of this map is written $d^kF_{\mathbf{u}}(\xi_1, \ldots, \xi_k)$ for vectors ξ_i on P. In the directional derivative notation, this same quantity would be written $D_{\xi_1} \cdots D_{\xi_k} F(\mathbf{u})$. To specialize these general notions to the particular case of curves, the convention is to use the unique unit vector δ on L for every direction ξ_i that arises. Thus, for a curve $F: L \to Q$, we have $F'(\bar{u}) = dF_{\bar{u}}(\delta)$ and $F^{(k)}(\bar{u}) = d^kF_{\bar{u}}(\delta, \ldots, \delta)$.

Proposition 8.4: Derivatives in terms of blossoms. *Let $F: P \to Q$ be a polynomial function of degree n, and let $f_*^\otimes: P_*^{\otimes n} \to Q_*$ be the linear blossom of F. The kth derivative of F is given by*

$$d^kF_{\mathbf{u}}(\xi_1, \ldots, \xi_k) = n^{\underline{k}} f_*^\otimes(\mathbf{u}^{n-k}\xi_1 \cdots \xi_k). \quad \square$$

One conclusion that we can draw from these formulas is that the more that we want to know about the behavior of a function F near a point, the more arguments to its multiaffine blossom we must be willing to vary away from that point. Computing a kth derivative of F at the point $\mathbf{u}$ involves varying k of the blossom arguments away from $\mathbf{u}$. If we fix all of the blossom arguments at $\mathbf{u}$, our knowledge of F is restricted to zeroth order, that is, to the value $F(\mathbf{u})$ itself. If we can vary all n of the blossom arguments, we can compute F to nth order, that is, we know F completely.

At this point, one must be careful to avoid falling into a tempting trap. Let's think about an n-ic polynomial curve $F: L \to Q$ for simplicity. Let $G: L \to Q$ be the unique polynomial curve of degree k that agrees with F to kth order at the point $\bar{r}$; the curves F and G are said to ***osculate*** to kth order at $\bar{r}$. We can construct G by truncating the Taylor series of F:

$$G(\bar{u}) := F(\bar{r}) + F'(\bar{r})(u - r) + \cdots + \frac{F^{(k)}(\bar{r})}{k!}(u - r)^k.$$

It would be wonderfully simple if the k-affine blossom g of G could be obtained simply by fixing $n - k$ of the arguments of the n-affine blossom f of F at the point $\bar{r}$—that is, if the formula

$$g(\bar{u}_1, \ldots, \bar{u}_k) = f(\bar{u}_1, \ldots, \bar{u}_k, \underbrace{\bar{r}, \ldots, \bar{r}}_{n-k})$$

held. Unfortunately, that formula does ***not*** hold, except in the trivial cases $k = 0$ and $k = n$. One way to see that it couldn't hold is to consider the case $k = 1$ and

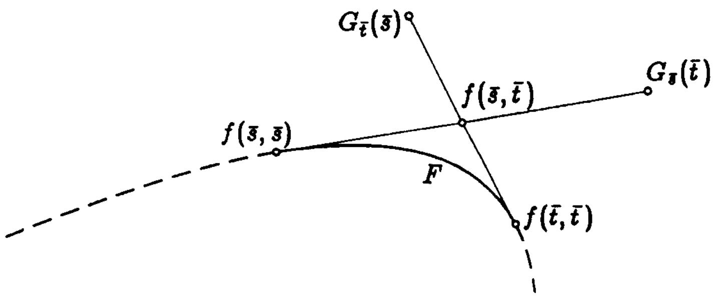


Fig. 8.5. Two affine maps that osculate a parabola

$n = 2$, when we are approximating a parabola $F(\bar u)$ by an affine function $G_{\bar r}(\bar u)$ for $\bar u$ near $\bar r$. If the formula above were true, the point $f(\bar s, \bar t)$ would be both $G_{\bar t}(\bar s)$ and also $G_{\bar s}(\bar t)$, that is, both the image of $\bar s$ under the best affine approximation to F at $\bar t$ and also the image of $\bar t$ under the best affine approximation to F at $\bar s$. In reality, however, as shown in Fig. 8.5, the two points $G_{\bar t}(\bar s)$ and $G_{\bar s}(\bar t)$ do not coincide; instead, we have $G_{\bar t}(\bar s) = 2f(\bar s,\bar t) - f(\bar t,\bar t)$, while $G_{\bar s}(\bar t) = 2f(\bar s,\bar t) - f(\bar s,\bar s)$. The source of the problem is the annoying factor of $n^{\underline{k}}$ in Prop. 8.2.

To determine the relationship that really does hold between the blossoms of F and G, we shall translate from multiaffine blossoms to linear blossoms. In that language, the tempting falsehood in the last paragraph states that $g_*^{\otimes}(e) = f_*^{\otimes}(e\,\bar r^{\,n-k})$ for any tensor e in $L_*^{\otimes k}$. The truth is that there is a bijective flavor-preserving linear map $m\colon L_*^{\otimes k} \to L_*^{\otimes k}$ with the property that $g_*^{\otimes}(e) = f_*^{\otimes}(m(e)\bar r^{\,n-k})$; but the map m is not the identity. To compute the map m, let us use the tensors $\delta^i \bar r^{\,k-i}$ for i in $[0,k]$ as a basis for $L_*^{\otimes k}$. From Prop. 8.2, we know that $G^{(i)}(\bar r) = k^{\underline{i}} g_*^{\otimes}(\delta^i \bar r^{\,k-i})$. Similarly for F, we know that $F^{(i)}(\bar r) = n^{\underline{i}} f_*^{\otimes}(\delta^i \bar r^{\,n-i})$. Since G osculates F to kth order at $\bar r$, these two quantities are equal. Hence, we conclude that

$$g_*^{\otimes}(\delta^i \bar r^{\,k-i}) = \frac{n^{\underline{i}}}{k^{\underline{i}}} f_*^{\otimes}(\delta^i \bar r^{\,n-i}) = f_*^{\otimes}\left(\left(\frac{n^{\underline{i}}}{k^{\underline{i}}}\delta^i \bar r^{\,k-i}\right)\bar r^{\,n-k}\right).$$

Let $m\colon L_*^{\otimes k} \to L_*^{\otimes k}$ be the linear map whose action on the basis tensors is given by $m(\delta^i \bar r^{\,k-i}) := (n^{\underline{i}}/k^{\underline{i}})\delta^i \bar r^{\,k-i}$; by the linearity of the linear blossoms $f_*^{\otimes}$ and $g_*^{\otimes}$, we may conclude that $g_*^{\otimes}(e) = f_*^{\otimes}(m(e)\bar r^{\,n-k})$ for all e in $L_*^{\otimes k}$.

All of this theory works perfectly well for surfaces as well as for curves. The surface G of degree k that osculates a surface $F\colon P \to Q$ of degree n to kth order near a point $\mathbf{r}$ is abstractly defined by truncating the Taylor series:

$$G(\mathbf{u}) := F(\mathbf{r}) + dF_{\mathbf{r}}(\mathbf{u}-\mathbf{r}) + \cdots + \frac{d^k F_{\mathbf{r}}(\mathbf{u}-\mathbf{r},\ldots,\mathbf{u}-\mathbf{r})}{k!}.$$

By the same reasoning as above, using Prop. 8.4 this time, we can conclude that there is a flavor-preserving linear bijection $m\colon P_*^{\otimes k} \to P_*^{\otimes k}$ with the property that $g_*^{\otimes}(e) = f_*^{\otimes}(m(e)\mathbf{r}^{n-k})$ for all k-tensors e on P. If the two vectors ξ and η are a basis of P_0, the effect of m on a basis tensor $\xi^i \eta^j \mathbf{r}^{k-i-j}$ of $P_*^{\otimes k}$ is given by $m(\xi^i \eta^j \mathbf{r}^{k-i-j}) := (n^{\underline{i+j}}/k^{\underline{i+j}})\xi^i \eta^j \mathbf{r}^{k-i-j}$.

Proposition 8.6. *If $F\colon P \to Q$ is an n-ic polynomial function, the unique polynomial function $G\colon P \to Q$ of degree k that osculates F to kth order at the point $\mathbf{r}$ in P is given by the formula $g_*^{\circledast}(e) := f_*^{\circledast}(m(e)\mathbf{r}^{n-k})$, where $m\colon P_*^{\circledast k} \to P_*^{\circledast k}$ is the invertible flavor-preserving linear map determined as follows. Let $\{\delta_1, \ldots, \delta_p\}$ be a basis of P_0. Then, the set of k-tensors of the form $\delta_1^{i_1}\cdots\delta_p^{i_p}\mathbf{r}^{k-i}$ form a basis for $P_*^{\circledast k}$, where $i := i_1 + \cdots + i_p$. The matrix of the map m with respect to this basis is diagonal, and the entries on the diagonal are determined by*

$$m(\delta_1^{i_1}\cdots\delta_p^{i_p}\mathbf{r}^{k-i}) := \frac{n^{\underline{i}}}{k^{\underline{i}}}\delta_1^{i_1}\cdots\delta_p^{i_p}\mathbf{r}^{k-i}. \quad \square$$

Note that the primary source of complexity in Prop. 8.6 was the annoying factors that arise when differentiating; indeed, the sole effect of the map m is to undo the damage caused by those factors. It is often more convenient to wash out the effects of the annoying factors by thinking about osculating flats (that is, affine subspaces) instead of osculating curves or surfaces. Define $\operatorname{Osc}_k F(\mathbf{r})$, the flat that osculates F to kth order at $\mathbf{r}$, to be the affine span of the range of G, where G is the curve or surface of degree k that osculates F to kth order at $\mathbf{r}$. For a nondegenerate curve F, the flats $\operatorname{Osc}_k F(\bar{r})$ will have dimension k. In particular, the flat $\operatorname{Osc}_1 F(\bar{r})$ is the tangent line to F at $\bar{r}$, while the flat $\operatorname{Osc}_2 F(\bar{r})$ is the osculating plane of F at $\bar{r}$, the plane containing the point $F(\bar{r})$ and spanned by the vectors $F'(\bar{r})$ and $F''(\bar{r})$. For a nondegenerate surface F, the flat $\operatorname{Osc}_1 F(\mathbf{r})$ is a tangent plane, while the flat $\operatorname{Osc}_2 F(\mathbf{r})$ is an affine 5-space containing the point $F(\mathbf{r})$ and spanned by the five vectors $dF_{\mathbf{r}}(\xi)$, $dF_{\mathbf{r}}(\eta)$, $d^2F_{\mathbf{r}}(\xi,\xi)$, $d^2F_{\mathbf{r}}(\xi,\eta)$, and $d^2F_{\mathbf{r}}(\eta,\eta)$, where $\{\xi,\eta\}$ is a basis for P_0.

The relation between osculating flats and blossoms can be computed as follows. If G is the k-ic function that osculates F to kth order at $\mathbf{r}$, the definition of osculating flats says that $\operatorname{Osc}_k F(\mathbf{r}) = \operatorname{Span}(\operatorname{Range}(G))$. Prop. 4.7 tells us that $\operatorname{Span}(\operatorname{Range}(G)) = \operatorname{Range}(g^{\circledast})$; thus, the flat $\operatorname{Osc}_k F(\mathbf{r})$ is precisely the locus of the points $g_*^{\circledast}(e)$ as e ranges over all 1-flavored k-tensors on P. From Prop. 8.6, we learn that $g_*^{\circledast}(e) = f_*^{\circledast}(m(e)\mathbf{r}^{n-k})$. Since m is a flavor-preserving bijection, we conclude that $\operatorname{Osc}_k F(\mathbf{r})$ is also precisely the locus of $f_*^{\circledast}(e\,\mathbf{r}^{n-k})$ as e varies over $P^{\circledast k}$. Notice how dealing with osculating flats, rather than curves or surfaces, washes out the effects of the annoying factors.

Translating back into multiaffine blossoms, we conclude that the range of the k-affine map

$$(\mathbf{u}_1, \ldots, \mathbf{u}_k) \mapsto f(\mathbf{u}_1, \ldots, \mathbf{u}_k, \underbrace{\mathbf{r}, \ldots, \mathbf{r}}_{n-k})$$

lies inside of $\operatorname{Osc}_k F(\mathbf{r})$, and also spans it. The "lying inside" part is particularly important.

Proposition 8.7. *If a point $\mathbf{r}$ in P occurs m times among the arguments of a value $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ of the n-affine blossom f, then that blossom value lies in the flat $\operatorname{Osc}_{n-m} F(\mathbf{r})$.* $\square$

This concept of osculating flats leads to a very natural differential perspective on the meaning of the n arguments of a multiaffine blossom. For example, let

$F: L \to Q$ be a nondegenerate polynomial cubic curve, and hence a space curve; and let $\bar{r}$, $\bar{s}$, and $\bar{t}$ be distinct points in L. The blossom value $f(\bar{r}, \bar{s}, \bar{t})$ is the unique point that lies in the intersection of the three osculating planes $\mathrm{Osc}_2\, F(\bar{r})$, $\mathrm{Osc}_2\, F(\bar{s})$, and $\mathrm{Osc}_2\, F(\bar{t})$. The value $f(\bar{s}, \bar{s}, \bar{t})$ is the intersection of the tangent line $\mathrm{Osc}_1\, F(\bar{s})$ with the osculating plane $\mathrm{Osc}_2\, F(\bar{t})$. More generally, for any nondegenerate polynomial curve or surface F, we can determine the blossom value $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ uniquely by intersecting the osculating flats corresponding to all of the distinct points $\mathbf{r}$ that occur in the argument bag $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$.

This differential intuition is so simple that one might be tempted to use it to define blossoms, instead of manipulating the coefficients of polynomials as we did in Section 2. Unfortunately, that idea runs into trouble if the polynomial curve or surface is degenerate, that is, if its Bézier points are not affinely independent. For example, if F is a polynomial cubic curve that happens to lie in a plane (that is, has zero torsion), all of its osculating planes $\mathrm{Osc}_2\, F(\bar{u})$ are coincident, and hence we can't use their intersections to define the blossom values $f(\bar{r}, \bar{s}, \bar{t})$.

Note that the osculating flats of surfaces interact with each other somewhat more subtly than do the osculating flats of curves. For example, let $F(\mathbf{u})$ be a nondegenerate quadratic polynomial surface. Since F has six Bézier points, the affine span of the range of F is a 5-flat. The blossom value $f(\mathbf{u}, \mathbf{v})$ is the intersection of the two tangent planes $\mathrm{Osc}_1\, F(\mathbf{u})$ and $\mathrm{Osc}_1\, F(\mathbf{v})$. The subtle point is that we would expect two planes in a 5-flat to be skew; since these two planes intersect in the point $f(\mathbf{u}, \mathbf{v})$, they aren't in general position. As the degree of the surface goes up, these coincidences get worse. A nondegenerate cubic surface F lies in a 9-flat. For each point $\mathbf{u}$, the flat $\mathrm{Osc}_2\, F(\mathbf{u})$ has dimension 5. Three 5-flats in general position in a 9-space don't come close to intersecting; but any three 5-flats of the form $\mathrm{Osc}_2\, F(\mathbf{u})$, $\mathrm{Osc}_2\, F(\mathbf{v})$, and $\mathrm{Osc}_2\, F(\mathbf{w})$ do intersect, at the point $f(\mathbf{u}, \mathbf{v}, \mathbf{w})$.

Working with osculating flats instead of osculating functions is one way to wash out the effects of the annoying factors. But there is another important context where the annoying factors aren't a problem. In particular, if we compare the behaviors of two different functions of the same degree, then the annoying factors cancel out, because they are the same on both sides.

Proposition 8.8. *If $F: P \to Q$ and $G: P \to Q$ are two polynomial functions of the same degree n, then F and G agree to kth order at $\mathbf{r}$ in P if and only if the identity*

$$f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_k \mathbf{r}^{n-k}) = g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_k \mathbf{r}^{n-k})$$

holds, that is, if and only if the blossoms f and g agree on all argument bags that include at least $n - k$ copies of $\mathbf{r}$.

Proof. Suppose first that the blossoms do agree as specified. We can rephrase this as the claim that $f_*^{\otimes}(e\mathbf{r}^{n-k}) = g_*^{\otimes}(e\mathbf{r}^{n-k})$ for all simple, 1-flavored k-tensors e on P. Since those tensors span all of $P_*^{\otimes k}$, we deduce that the same equality holds for every k-tensor e on P. Note that, in fact, it would have been enough to have assumed that $f^{\otimes}(\mathbf{u}^k \mathbf{r}^{n-k}) = g^{\otimes}(\mathbf{u}^k \mathbf{r}^{n-k})$ for all points $\mathbf{u}$ in P, since the k-tensors $\mathbf{u}^k$ that are perfect kth powers also span $P_*^{\otimes k}$, by Cor. 4.6.

Let $d^i F_{\mathbf{r}}(\xi_1, \ldots, \xi_i)$ be a value of the ith derivative of F at $\mathbf{r}$, for i in $[0, k]$. By Prop. 8.4, we have $d^i F_{\mathbf{r}}(\xi_1, \ldots, \xi_i) = n^{\underline{i}} f_*^{\otimes}(\mathbf{r}^{n-i}\xi_1 \cdots \xi_i)$, and similarly for G. We can deduce that these derivatives agree by letting the k-tensor e above assume the value $e := \xi_1 \cdots \xi_i \mathbf{r}^{k-i}$.

Conversely, suppose that F and G agree to kth order at $\mathbf{r}$, and let $\{\delta_1, \ldots, \delta_p\}$ be a basis for P_0. The k-tensors of the form $c := \delta_{j_1} \cdots \delta_{j_i} \mathbf{r}^{k-i}$ for i in $[0, k]$ form a basis for $P_*^{\otimes k}$. Furthermore, by Prop. 8.4 again, we have $f_*^{\otimes}(c\mathbf{r}^{n-k}) = (1/n^{\underline{i}}) d^i F_{\mathbf{r}}(\delta_{j_1}, \ldots, \delta_{j_i})$, and similarly for G. If the ith derivatives of F and G agree, then it follows that $f_*^{\otimes}(c\mathbf{r}^{n-k}) = g_*^{\otimes}(c\mathbf{r}^{n-k})$ for all c in the basis, hence for all c in $P_*^{\otimes k}$, and hence, in particular, for all c of the form $c = \mathbf{u}_1 \cdots \mathbf{u}_k$. □

Prop. 8.8 will be the starting point in Section 14 for our investigation of spline curves via the blossoming technology.

9. The algebraic view of a blossom's n arguments

Our third perspective on the meaning of a blossom's n arguments is a trivial observation with important consequences: the n arguments of a blossom value are the n factors of a simple n-tensor. This observation exploits the Tensoring Principle to give a clearer understanding of the geometric relationships between a polynomial function and its multiaffine blossom. Note that the range of the polynomial function F itself is the curve or surface that we started with, while the range of F's multiaffine blossom f is the set of points for which we have good labels. So far, f has provided us with labels for all of the important points, such as Bézier points and poles. But there are points that f doesn't label.

The Tensoring Principle tells us that a polynomial map $F: P \to Q$ is equivalent to an affine map $f^{\otimes}: P^{\otimes n} \to Q$, and an affine map is a pretty simple thing. Instead of trying to understand the geometry of $F(\mathbf{u}) = f^{\otimes}(\mathbf{u}^n)$ and its multiaffine blossom $f(\mathbf{u}_1, \ldots, \mathbf{u}_n) = f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n)$ in the object space Q, it is a better idea to study the geometry of the corresponding argument n-tensors $\mathbf{u}^n$ and $\mathbf{u}_1 \cdots \mathbf{u}_n$ back in the tensor power space $P^{\otimes n}$. The geometry in $P^{\otimes n}$ is independent of F. Furthermore, since the map $f^{\otimes}$ is affine, it can affect that geometry only in two fairly simple ways. First, the map $f^{\otimes}$ may fail to be surjective (onto). Prop 4.7 tells us that the range of $f^{\otimes}$ is precisely the flat in Q that forms the affine span of the range of F, or equivalently, the affine span of the Bézier points of F with respect to some reference simplex in P. This range flat $f^{\otimes}(P^{\otimes n})$ might be smaller than Q. But however big it is, it contains all of the points of interest for the study of F. By studying the geometry of $P^{\otimes n}$, we can avoid being distracted by the irrelevant points in Q that don't lie in $f^{\otimes}(P^{\otimes n})$. Second, the map $f^{\otimes}$ may fail to be injective (one-to-one). This happens precisely when the Bézier points of F are not affinely independent, causing the map $f^{\otimes}$ to collapse its domain $P^{\otimes n}$ down to a range flat $f^{\otimes}(P^{\otimes n})$ of smaller dimension. This type of degeneracy makes it particularly attractive to study the geometry back in $P^{\otimes n}$ rather than in Q, where things have collapsed on top of each other.

To study the geometry of $P^{\otimes n}$, we shall invoke Prop. 5.6. Let $\{b_0, \ldots, b_p\}$ be a linear basis for the linearization P_* of P, and let β be the associated normal-form map. Prop. 5.6 tells us that β is an algebra isomorphism between $\mathcal{X}[P]$ and

$\mathbf{R}[b_0, \dots, b_p]$. Thus, questions about the structure of the space $P^{\otimes n} = \mathcal{X}_1^n[P]$ can be translated, using β, into questions about 1-flavored n-homogeneous polynomials in $\mathbf{R}[b_0, \dots, b_p]$. The question that interests us at the moment is whether or not a tensor e in $P^{\otimes n}$ can be written as a product of n points $e = \mathbf{u}_1 \cdots \mathbf{u}_n$, since those are the tensors that correspond under $f^{\otimes}$ to points that have multiaffine labels.

If e is a tensor in $P^{\otimes n}$, the normal-form polynomial $\beta(e)$ is 1-flavored and n-homogeneous. In order for a factorization $e = \mathbf{u}_1 \dots \mathbf{u}_n$ to exist, the polynomial $\beta(e)$ must factor as the product $\beta(e) = \beta(\mathbf{u}_1) \cdots \beta(\mathbf{u}_n)$ of n 1-flavored 1-homogeneous polynomials $\beta(\mathbf{u}_i)$. In fact, it is enough that $\beta(e)$ split in any way into n linear factors. It is a standard algebraic result that any factor of a homogeneous polynomial must be homogeneous, so the homogeneity of the factors is guaranteed. As for their flavor, since the product is 1-flavored, we can always push scalars around so that each factor is also 1-flavored. Thus, an n-tensor e on P can be written as a product of n points if and only if its normal-form polynomial $\beta(e)$ splits into linear factors.

This notion of "splits into linear factors" is important enough that it has a name. An n-tensor e on P (of any flavor) is called ***simple*** if it can be factored into a product of 1-tensors; otherwise, it is called ***compound.*** (The term "decomposable" is sometimes used instead of "simple.") We started out building the tensor power space $P^{\otimes n}$ by multiplying points together; the tensors that result directly from this multiplication are the simple ones. The compound ones are the ones that have to be thrown in as well in order to make $P^{\otimes n}$ into an affine space.

For example, consider the 2-flavored 2-tensor e on the line L given by $e := \bar{0}^2 + \bar{2}\,\overline{12}$, and use the Cartesian basis $\{\bar{0}, \delta\}$ for L_*. Converting to normal form, we have

$$\begin{aligned}\beta(e) &= \bar{0}^2 + (\bar{0} + 2\delta)(\bar{0} + 12\delta)\\ &= 2\bar{0}^2 + 14\bar{0}\delta + 24\delta^2\\ &= 2(\bar{0} + 3\delta)(\bar{0} + 4\delta).\end{aligned}$$

Since $\beta(e)$ splits, the tensor e is simple; in fact, $e = 2\,\bar{3}\,\bar{4}$. For an example in the other direction, note that the tensor $c = \bar{0}^2 + \delta^2$ is already in normal form. Since $\beta(c) = c$ does not split (over the real numbers), the tensor c is compound.

In the rest of this section, we shall try to develop some intuition for the geometry of the set of simple tensors sitting in $P^{\otimes n}$ for small values of n and $p := \dim(P)$. We begin by reviewing the case of quadratic curves, where $n = 2$ and $p = 1$.

The second tensor power $L^{\otimes 2}$ of the affine line L consists of homogeneous quadratic polynomials in two variables with one flavor constraint. If we use the Cartesian basis $\{\bar{0}, \delta\}$ for L_*, a tensor e in $L^{\otimes 2}$ has the normal form $\beta(e) = \bar{0}^2 + b\bar{0}\delta + c\delta^2$ for some constants b and c, where the flavor constraint is embodied in the fact that the coefficient of $\bar{0}^2$ is 1. We can use the pair $\langle b, c\rangle$ as a Cartesian coordinate system on the plane $L^{\otimes 2}$. The factoring properties of the tensor e depend upon the sign of the discriminant $\mathrm{Disc}_\beta(e) = b^2 - 4c$. (The subscript of β warns us that the value of the discriminant, although not the sign of that value, depends upon the chosen basis for L_*.) If $\mathrm{Disc}_\beta(e) = b^2 - 4c = 0$, then e is a perfect square $e = \bar{u}^2 = (\bar{0} + u\delta)^2 = \bar{0}^2 + 2u\bar{0}\delta + u^2\delta^2$. Let D denote the set of such tensors

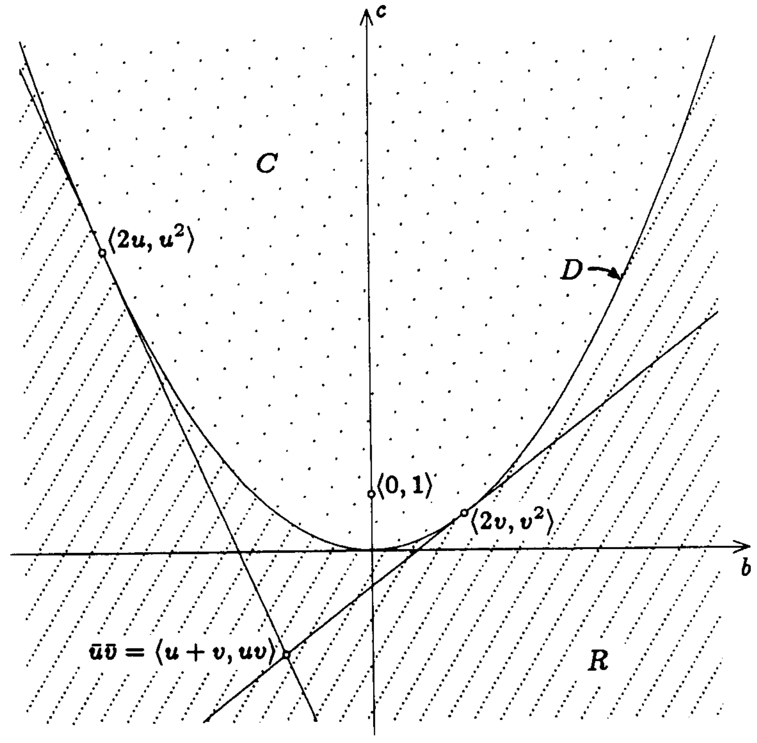


Fig. 9.1. An affine picture of the space $L^{\otimes 2}$ of 2-tensors on L

("D" for double root). Note that D forms a parabola $D(\bar{u}) := \langle 2u, u^2\rangle$ in the plane $L^{\otimes 2}$, as shown in Fig. 9.1. If $\mathrm{Disc}_\beta(e) > 0$, then e has two distinct real roots: $e = \bar{u}\bar{v} = \bar{0}^2 + (u+v)\bar{0}\delta + uv\delta^2$ for $u \neq v$. Let R denote the set of such tensors $\langle u+v, uv\rangle$ ("R" for real roots); R is the exterior of D. Finally, if $\mathrm{Disc}_\beta(e) < 0$, then e doesn't factor over the reals. Call the set of such tensors C (for compound tensors); C is the interior of D.

If we enlarged our coefficient field from the real numbers to the complexes, then the tensors in C would also factor, with a pair of complex conjugate roots. For example, the 2-tensor $\langle 0,1\rangle = \bar{0}^2 + \delta^2$ in C, which is compound over the reals, splits into $(\bar{0}+i\delta)(\bar{0}-i\delta)$ over the complexes. Indeed, every n-tensor on L for every n would then be simple, since the complex numbers are algebraically closed. But the real numbers are the natural coefficients to use in computer graphics, so we shall stick with them. Moving to the complexes would only eliminate the problem of compound tensors for curves, anyway. For surfaces, as we will see later on, most tensors are compound even over the complex numbers.

One difficulty with our current picture of the space $L^{\otimes 2}$ is that some of the interesting things are happening out at infinity, where they are hard to see. With a little work, we can get a finite picture of all of the simple 2-tensors on L, that is, of $R \cup D$. Indeed, some readers may have already realized that, viewed projectively, the set $R \cup D$ is a Möbius strip. We shall derive this fact in slow and careful algebraic detail in the next few paragraphs. The advantage of our plodding derivation is that a closely analogous method will then suffice to compute a finite picture of the simple 3-tensors on L—a picture that is less well-known. The first step on our plodding road is to invoke projective geometry to explain what it means for things to be happening "at infinity."

Consider the linearization $L_*^{\otimes 2}$ of $L^{\otimes 2}$. This linearization is a 3-space consisting of the tensors $\langle b, c; a\rangle = a\bar{0}^2 + b\bar{0}\delta + c\delta^2$, where we are writing the flavor-coordinate a last out of habit. The basic idea of projective geometry is to treat the lines through the origin of $L_*^{\otimes 2}$ as the "points" of a new space: a *projective plane*. Most of the lines through the origin of $L_*^{\otimes 2}$ contain precisely one point of each flavor; in particular, the same line that contains the point $\langle b, c; a\rangle$ also contains $\langle b/a, c/a; 1\rangle$ whenever $a \neq 0$. Such lines model "finite points" in the projective plane. The other lines consist entirely of points that are 0-flavored. Such lines model "points at infinity." We can study things that happen everywhere in $L^{\otimes 2}$, including "at infinity," by studying a neighborhood of the origin in the homogeneous linear space $L_*^{\otimes 2}$.

Carrying out this plan, we find that the factoring properties of a tensor $e = \langle b, c; a\rangle$ in the 3-space $L_*^{\otimes 2}$ depend upon the homogenized form of the discriminant $\mathrm{Disc}_\beta(e) := b^2 - 4ac$. The set D of tensors with a double root is described by the equation $\mathrm{Disc}_\beta(e) = 0$, and forms a double cone with elliptical cross section, centered around the line $a = c$, $b = 0$. The set R of tensors with two distinct real roots has $\mathrm{Disc}_\beta(e) > 0$, and forms the exterior of this cone, while the set C of compound tensors has $\mathrm{Disc}_\beta(e) < 0$ and forms the interior of the cone.

The second step on our plodding road involves using some type of projection to reduce this 3-dimensional picture down to a 2-dimensional one. Projection, in this context, merely means restricting our attention to some 2-dimensional surface in the 3-space $L_*^{\otimes 2}$ that cuts all (or most) of the lines through the origin in one point (or a small number of points). The most obvious surface to project down onto is a plane. If we pick any plane A in $L_*^{\otimes 2}$ not containing the origin, most lines through the origin will intersect A in a unique point. The lines that don't intersect A form a plane through the origin parallel to A. In projective language, those lines are viewed as "points at infinity," which, together, make up the "line at infinity." The problem with such projections is that every plane through the origin of $L_*^{\otimes 2}$ intersects the exterior R of the cone, and hence includes some lines of simple tensors. Since we want to get a single, finite picture of all of the simple tensors, we can't afford to miss such a line. For example, if we picked the plane $A := L^{\otimes 2}$, this projection would take us right back to Fig. 9.1, where we started, in which the entire "line at infinity" consists of simple tensors.

Instead of projecting onto a plane, we shall project onto an infinite cylinder, as in the Mercator projection used in map-making. The advantage of this choice is that only one line through the origin escapes intersecting our chosen surface: the axis of the cylinder. We can choose things so that this line consists of compound tensors, which we don't mind ignoring. One disadvantage of using a cylinder is that each line through the origin that does intersect the cylinder does so twice, in an antipodal pair of points. In order to achieve our final picture, we will have to identify the two points in such an antipodal pair.

To make the geometry of the cylindrical projection simpler, it is helpful to

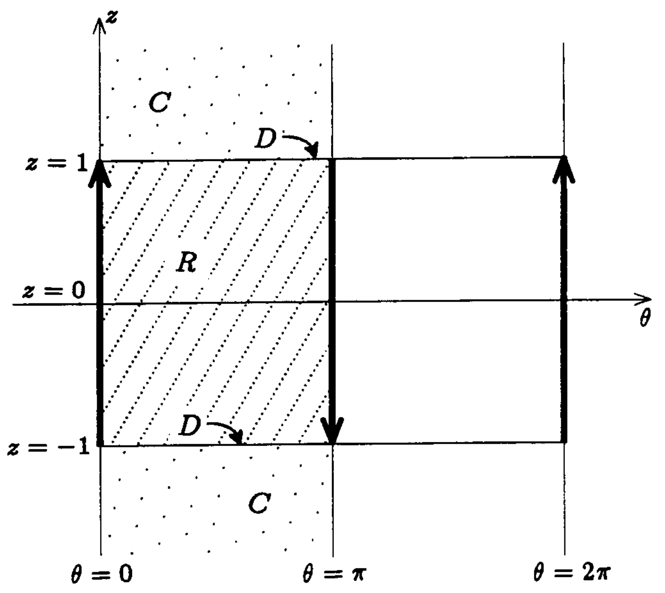


Fig. 9.2. A projective picture of the space $L^{\otimes 2}$ of 2-tensors on L

convert to a new Cartesian coordinate system for $L_*^{\otimes 2}$. In particular, let $x := b$, $y := c - a$, and $z := c + a$. In the $\langle x, y, z \rangle$ coordinate system, the discriminant has the simpler form $\mathrm{Disc}_\beta(e) = x^2 + y^2 - z^2$. The cone D determined by the equation $\mathrm{Disc}_\beta(e) = 0$ now has circular cross sections, an apex angle of 90 degrees, and is centered around the z-axis. If we imagine a copy of the earth centered at the origin of $L_*^{\otimes 2}$ and with its poles on the z-axis, the cone D intersects that earth along the parallels of 45 degrees north and south latitude.

We now project $L_*^{\otimes 2}$ onto the cylinder of radius 1 about the z axis; that is, we restrict our attention to tensors $\langle x, y, z \rangle$ that satisfy $x^2 + y^2 = 1$. Fix an angle θ, let $u := \cos\theta$ and $v := \sin\theta$, and consider the set of tensors $\ell_\theta = \{\langle \cos\theta, \sin\theta, z \rangle \mid z \in \mathbf{R}\}$, which is a generating line of the cylinder. The line ℓ_θ corresponds to a particular meridian of longitude in the Mercator analogy. For tensors e in ℓ_θ, we have $\mathrm{Disc}(e) = 1 - z^2$. Hence, the line ℓ_θ intersects D in the points $\langle \cos\theta, \sin\theta, \pm 1 \rangle$ and intersects R in the points $\langle \cos\theta, \sin\theta, z \rangle$ for z in $(-1, 1)$. Fig. 9.2 shows the cylinder unrolled onto the $\langle \theta, z \rangle$ plane. Remember that each line through the origin of $L_*^{\otimes 2}$ intersects this cylinder in two antipodal points: one with θ in $[0, \pi)$ and the other with θ in $[\pi, 2\pi)$. Thus, the set of simple 2-tensors on L is, up to a diffeomorphism, the shaded region in Fig. 9.2 with its left and right edges identified as indicated by the arrows. That is, the simple 2-tensors $R \cup D$ on L form a Möbius strip with the perfect squares D as its single edge.

To understand cubic curves, we must study the geometry of $L^{\otimes 3}$. Continuing with the Cartesian basis $\{\bar{0}, \delta\}$ for L_*, a generic tensor e in $L^{\otimes 3}$ has the normal-form polynomial $\beta(e) = \bar{0}^3 + b\bar{0}^2\delta + c\bar{0}\delta^2 + d\delta^3$, and the triple $e = \langle b, c, d \rangle$ forms a Cartesian coordinate system on $L^{\otimes 3}$. From a factorization point of view, the tensors e in $L^{\otimes 3}$ can be partitioned into four classes: T, the tensors with one triple real root; D, the tensors with one double real root and one single real root; R, the tensors with three distinct real roots; and C, the compound tensors. (Viewed over the complexes, the tensors in C have one real root and one pair of conjugate complex roots.) From the classical theory of cubic polynomials, we recall that the discriminant of the polynomial $\beta(e)$ is given by

$$\mathrm{Disc}_\beta(e) := b^2c^2 - 4c^3 - 4b^3d - 27d^2 + 18bcd.$$

The set R contains the tensors whose discriminant is strictly positive; the set C contains the tensors whose discriminant is strictly negative; and all of the tensors in $D \cup T$ have zero discriminant. Note that $D \cup T$ constitutes the boundary between R and C, since the discriminant varies smoothly. The set T, which forms part of this boundary, consists of the perfect cubes $T(\bar{u}) = \bar{u}^3 = \langle 3u, 3u^2, u^3\rangle$, and is hence a twisted cubic curve lying in the 3-space $L^{\otimes 3}$.

Fig. 9.3 is the analog, in the cubic case, of the parabolic picture in Fig. 9.1. Of course, the space $L^{\otimes 3}$ has three dimensions instead of two. Instead of attempting to draw a perspective view, Fig. 9.3 shows four different cross sections, corresponding to the planes $b = -3$, $b = 0$, $b = 3$, and $b = 6$. If we put the c axis vertically, as done in Fig. 9.3, each cross section of the form $b = 3\lambda$ looks rather like an ocean wave, with R as the water, C as the air, D as the water's surface, and T as the cusp of the wave. To plot any cross section of $L^{\otimes 3}$, it is enough to compute the intersection of the cutting plane with the boundary $D \cup T$, or equivalently, to compute all tensors of the form $\bar{u}^2\bar{v}$ that lie in the cutting plane. In Cartesian coordinates, such a tensor is given by: $e = \bar{u}^2\bar{v} = (\bar{0} + u\delta)^2(\bar{0} + v\delta) = \langle 2u + v, u^2 + 2uv, u^2v\rangle$. Cutting by a plane of the form $b = 3\lambda$ for some real constant λ is particularly simple, since the relation $b = 2u + v = 3\lambda$ gives us a linear relation between u and v, which allows us to express both c and d as cubic polynomials in either u or v. That is, the curve in which the boundary surface $D \cup T$ meets the plane $b = 3\lambda$ is a polynomial cubic. In fact, it is always an affine image of a semi-cubical parabola, with the cusp located on T at the tensor $\bar{\lambda}^3$.

Fig. 9.3 also illustrates another interesting point. A tensor of the form $e = \bar{s}^2\bar{t}$ in $D \cup T$ must lie on the tangent line $\mathrm{Osc}_1\, T(\bar{s})$ to the cubic curve $T(\bar{u})$ at $\bar{u} = \bar{s}$, as we can see by applying Prop. 8.7. Therefore, the boundary surface $D \cup T$ is actually a ruled surface, with the tangent lines to the curve $T(\bar{u})$ as its generators. The four circled points in Fig. 9.3 indicate the intersections of the tangent line $\mathrm{Osc}_1\, T(\bar{1})$ with the four drawn cross sections.

This first picture of the geometry of the simple tensors surrounding a cubic curve has the same problem that our first picture had in the quadratic case: lots of the interesting behavior is out at infinity. By following in the footsteps of E. C. Zeeman, we can obtain a finite picture of all of the simple 3-tensors on L, that is, an analog to the Möbius strip of Fig. 9.2 for cubics. Zeeman invented this finite picture for use in catastrophe theory, and he christened it the ***umbilic bracelet*** [39].

The first step is to projectivize. Instead of considering the affine 3-space $L^{\otimes 3}$, we consider lines through the origin of the linear 4-space $L_*^{\otimes 3}$. The factorization properties of a 3-tensor $e = \langle b, c, d; a\rangle = a\bar{0}^3 + b\bar{0}^2\delta + c\bar{0}\delta^2 + d\delta^3$ are determined by

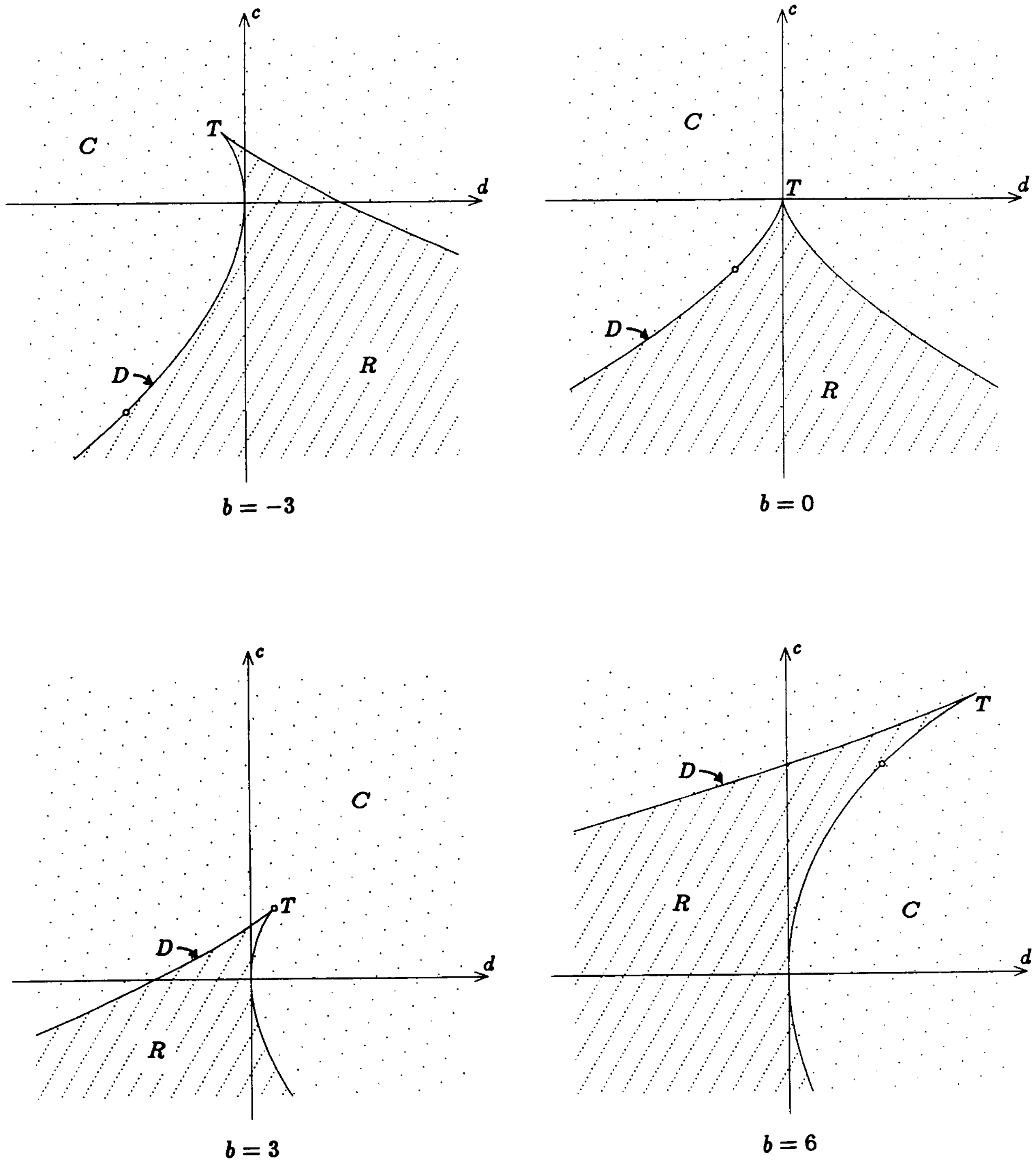


Fig. 9.3. Four affine cross sections through the space $L^{\otimes 3}$

the homogenized form of the cubic discriminant:

$$\mathrm{Disc}_\beta(e) := b^2c^2 - 4ac^3 - 4b^3d - 27a^2d^2 + 18abcd.$$

The next step to take isn't so clear, however. We shall simply follow Zeeman, treating his ideas like rabbits pulled from hats. Zeeman's own presentation of the umbilic bracelet in [39] is better motivated, at the price of invoking the machinery of group actions.

Zeeman first changes to a new $\langle u, v, x, y\rangle$ Cartesian coordinate system for $L_*^{\otimes 3}$ by the following rules: $u := (a-c)/4$, $v := (d-b)/4$, $x := (3a+c)/4$, and $y := (-3d-b)/4$. A little algebra reveals that, in terms of these new coordinates, the discriminant $\mathrm{Disc}_\beta(e)$ is given by

$$4\big(27(u^2+v^2)^2 - 18(u^2+v^2)(x^2+y^2) - (x^2+y^2)^2 + 8ux^3 + 24vx^2y - 24uxy^2 - 8vy^3\big).$$

Note that, if $u = v = 0$, then $\mathrm{Disc}_\beta(e) = (-4)(x^2+y^2)^2$. Thus, the lines through the origin of $L_*^{\otimes 3}$ that lie in the plane $u = v = 0$ all consist of tensors with negative discriminant, which are hence compound. Since we are searching for a finite picture of the simple tensors, we can safely ignore the plane $u = v = 0$. Every tensor $\langle u, v, x, y\rangle$ that does not satisfy $u = v = 0$ has two scalar multiplies that satisfy $u^2 + v^2 = 1$. We shall project $L_*^{\otimes 3}$ onto the set of tensors satisfying $u^2 + v^2 = 1$, which is a solid torus: the Cartesian product of the unit circle in the $\langle u, v\rangle$ plane with the entire $\langle x, y\rangle$ plane.

Fix an angle θ, let $u := \cos\theta$ and $v := \sin\theta$, and consider the factorization structure of the 3-tensors in the plane $A_\theta = \{\langle\cos\theta, \sin\theta, x, y\rangle \mid x, y \in \mathbf{R}\}$. For tensors e in A_θ, we have

$$\mathrm{Disc}_\beta(e) = 4\big(27 - 18(x^2+y^2) - (x^2+y^2)^2 + (8x^3 - 24xy^2)\cos\theta + (24x^2y - 8y^3)\sin\theta\big).$$

The equation $\mathrm{Disc}_\beta(e) = 0$ is a quartic equation in x and y whose solution is some algebraic plane curve of degree 4. In fact, that curve turns out to be a hypocycloid with three cusps. In words, mark a point on a circle of radius 1; roll that circle around the inside of a circle of radius 3 centered at the origin of A_θ; and then rotate the hypocycloid traced by the marked point so that one of its cusps makes an angle of $\theta/3$ with the x-axis. The result is the curve $\mathrm{Disc}_\beta(e) = 0$. To verify this claim without drowning in a sea of trigonometric identities, it is helpful to introduce a complex variable $z := x + iy$ and to view the plane A_θ as a copy of the complex numbers. The hypocycloid in question is then given parametrically by the formula

$$z := e^{i\theta/3}\big(2e^{i\varphi} + e^{-2i\varphi}\big) \tag{9.4}$$

for φ in $[0, 2\pi)$. The equations $u + iv = e^{i\theta}$ and $x + iy = z$ allow us to express the discriminant of e in A_θ as follows: $\mathrm{Disc}_\beta(e) := 4\Re(27 - 18z\bar{z} - (z\bar{z})^2 + 8z^3e^{-i\theta})$, where $\Re$ denotes the real part and $\bar{z}$ denotes the complex conjugate of z. If we plug in the z from Eq. (9.4), it is straightforward algebra to check that $\mathrm{Disc}_\beta(e) = 4\Re(8(e^{3i\varphi} - e^{-3i\varphi}) - 4(e^{6i\varphi} - e^{-6i\varphi})) = 0$. This verifies that all tensors on the hypocycloid have zero discriminant. To see that no other tensors in the plane A_θ

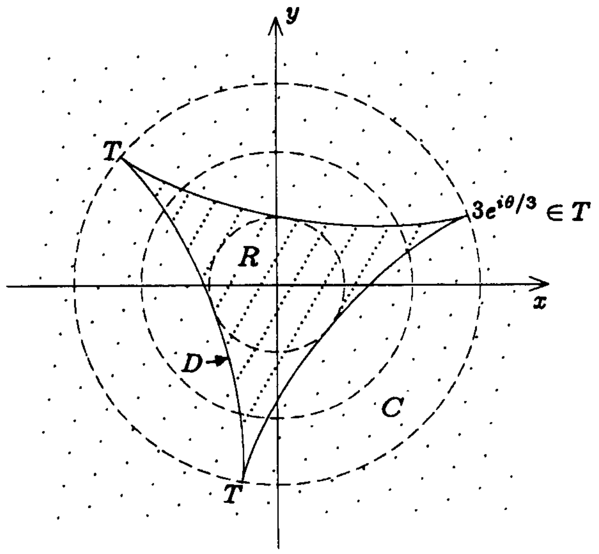


Fig. 9.5. The plane A_θ

could also have zero discriminant, consider rotating a line in the plane so that it remains tangent to the hypocycloid. Counting multiplicities, this rotating line always intersects the hypocycloid a total of four times. In general, the point of tangency is a double intersection and the two crossings of the other two sides are each single intersections. When the point of tangency coincides with a cusp, that intersection has multiplicity 3 and the midpoint of the opposite side provides the fourth. Since the discriminant is a polynomial of degree 4, the curve that it implicitly defines can't intersect any line more than 4 times; therefore, the hypocycloid must account for all of the solutions of $\mathrm{Disc}_\beta(e) = 0$ in the plane A_θ.

We can now deduce the factorization properties of all of the tensors in the plane A_θ, as shown in Fig. 9.5. Since the origin $\langle \cos\theta, \sin\theta, 0, 0\rangle$ has discriminant $4 \cdot 27 = 108$, which is positive, all the tensors inside the hypocycloid belong to R, while the ones outside belong to C. The tensors on the hypocycloid itself belong to $D \cup T$. Comparing with Fig. 9.3, it is geometrically clear that the three cusps are in T while the rest of the hypocycloid is in D. To verify this algebraically, we first verify by trigonometry that the tensor $4\big(\cos(\tau/3)\bar{0} - \sin(\tau/3)\bar{\delta}\big)^3$, when expressed in $\langle u, v, x, y\rangle$ coordinates, is given by $\langle \cos\tau, \sin\tau, 3\cos(\tau/3), 3\sin(\tau/3)\rangle$. Since the three cusps in A_θ correspond to the three values $\tau = \theta$, $\tau = \theta + 2\pi$, and $\tau = \theta + 4\pi$, we conclude that the three cusps are perfect cubes, and hence lie in T. No other points of the hypocycloid can lie in T because the cubic curve T can intersect the plane A_θ in at most three points.

To achieve the finite picture of $R \cup D \cup T$ that we have been searching for, it only remains to assemble the planes A_θ for different θ. Fig. 9.6 shows a picture of the relevant portion of $\langle \theta, x, y\rangle$ space. Just as in the quadratic case, we restrict θ to the interval $[0, \pi)$ in order to get only one point from each antipodal pair. As θ varies over this interval, the hypocycloid rotates one sixth of a turn. The plane A_0

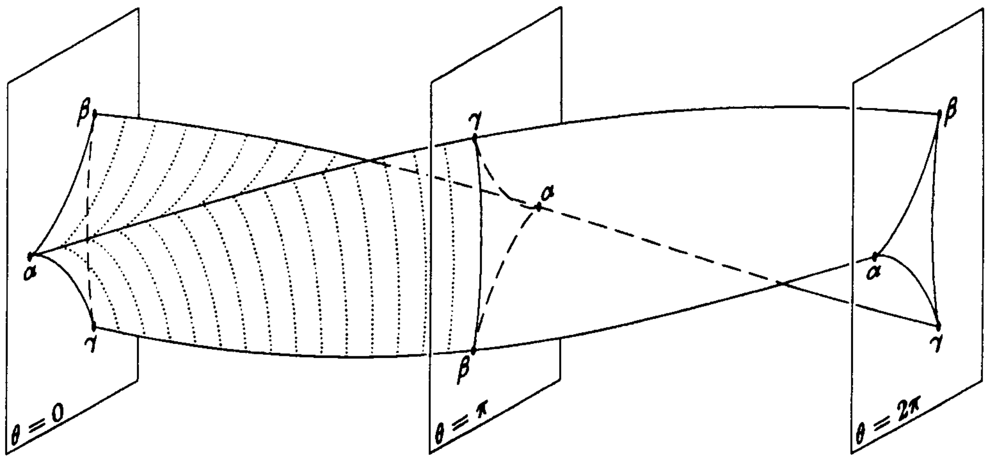


Fig. 9.6. A projective picture of the space $L^{\otimes 3}$

is identified with A_π as indicated by the letters. Note that this identification does not destroy orientability in the cubic case, although it did in the quadratic case; the reason is that negating all four axes of $L_*^{\otimes 3}$ is orientation preserving, while negating all three axes of $L_*^{\otimes 2}$ is not. If we didn't care at all about compound tensors, we could bend the screw of simple tensors in Fig. 9.6 around, giving it an extra half-twist so as to bring the two identified faces A_0 and A_π together. If we choose the correct sign for the extra half-twist, the resulting solid is the ***umbilic bracelet*** shown in Fig. 9.7.

Remark 9.8. In catastrophe theory, it is important not to identify two cubics that differ in sign, since this would confuse maxima and minima. Hence, instead of dealing with lines through the origin of $L_*^{\otimes 3}$, Zeeman deals with rays leaving the origin. He allows θ in Fig. 9.6 to vary over $[0, 2\pi)$ instead of just $[0, \pi)$ and he then identifies A_0 with $A_{2\pi}$ without any extra twisting. The resulting bracelet looks the same either way, however. Recent work by Jorge Stolfi suggests that computational geometry may be another area where it is better to keep the antipodal rays as separate objects, rather than to identify them, because this enables the construction of an oriented version of projective geometry [38].

Surfaces work out differently from curves in two respects. First, there are so many dimensions involved for even a quadratic surface that we won't waste any effort trying to visualize the simple tensors geometrically; instead, we will stick to algebra. The second difference also involves dimensionality. When studying curves, we found that the simple 1-flavored n-tensors on L formed an n-dimensional subset of the n-dimensional space $L^{\otimes n}$. Thus, not all tensors were simple, but there were as many dimensions of simple tensors as there were of tensors in general. For surfaces, that is no longer true. All of the 1-flavored n-tensors on a plane P form the space $P^{\otimes n}$, which has dimension $\binom{n+2}{2} - 1 = n(n+3)/2$, by Prop. 4.9. But each simple 1-flavored n-tensor can be written as the product $e = \mathbf{u}_1 \cdots \mathbf{u}_n$ of n points in P; hence, there can't be more than $2n$ dimensions' worth of them. This discrepancy in dimension corresponds to the fact that homogeneous polynomials in three variables

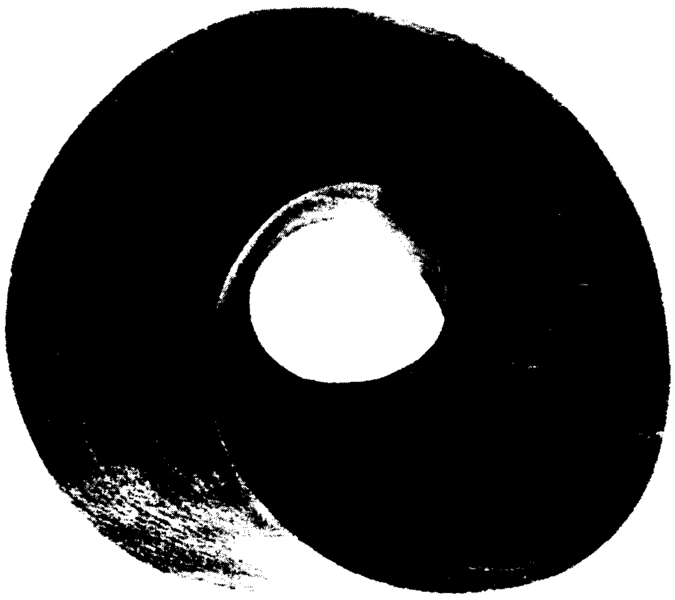


Fig. 9.7. The umbilic bracelet, as carved in maple by Tim Poston

generally don't split into linear factors, even over the complex numbers.

For example, consider quadratic surfaces. The study of quadratic surfaces is the study of $P^{\otimes 2}$, where P is an affine plane. Prop. 4.9 tells us that $P^{\otimes 2}$ has dimension five. If we pick a basis $\{b_1, b_2, b_3\}$ for P_*, the normal form $\beta(e)$ of a tensor e in $P_*^{\otimes 2}$ will be a homogeneous quadratic polynomial in the three variables b_i. It is technically convenient to pick a Cartesian basis, say $\{o, \xi, \eta\}$, where o is a point in P and ξ and η are vectors on P. With respect to this basis, we have

$$\beta(e) = uo^2 + v\xi^2 + w\eta^2 + x\xi\eta + yo\eta + zo\xi,$$

where $u = \mathrm{Flav}(e)$. Thus, the 6-tuple $\langle v, w, x, y, z; u\rangle$ forms a Cartesian coordinate system on $P_*^{\otimes 2}$, and $P^{\otimes 2}$ is the subspace $u = 1$. A tensor e in $P_*^{\otimes 2}$ is simple if it factors into, say, $e = (a_1 o + b_1\xi + c_1\eta)(a_2 o + b_2\xi + c_2\eta)$. Though there are six coefficients in this factored form, there are only five degrees of freedom in the product, since multiplying the first factor by λ and the second by $1/\lambda$ doesn't affect the result. Thus, the six coefficients of a general 2-tensor e must satisfy some identity in order for there to be any hope that e splits, even over the complex numbers. That identity turns out to be

$$ux^2 + vy^2 + wz^2 - 4uvw - xyz = 0, \tag{9.9}$$

as the reader can easily verify.

Challenge 9.10. A general 3-tensor e on P has the normal form

$$qo^3 + ro^2\xi + so^2\eta + to\xi^2 + uo\xi\eta + vo\eta^2 + w\xi^2 + x\xi^2\eta + y\xi\eta^2 + z\eta^3,$$

with 10 coefficients, of which the first is the flavor $q = \text{Flav}(e)$. The simple 3-tensors

$$(a_1o + b_1\xi + c_1\eta)(a_2o + b_2\xi + c_2\eta)(a_3o + b_3\xi + c_3\eta)$$

form a 7-dimensional subset of the space $P_*^{\otimes 3}$; there are 9 coefficients, but 2 of the degrees of freedom just move scalars around among the three factors. This implies that there are 3 dimensions' worth of constraints that must hold on the coefficients q through z in order for the tensor e to have any hope of splitting into linear factors, over any coefficient field. Find a basis for the ideal of polynomials in $\mathbf{R}[q, \ldots, z]$ that are zero on all simple 3-tensors. (Resultants or Gröbner bases might help.)

Part D: Adjusting the Degree

In most of our work so far, we have taken the degree n of our curves and surfaces to be a manifest constant; the exception was our study of osculating curves and surfaces in Section 8. There are various situations where it is useful to adjust the degree, however. Raising the degree means viewing a function F of degree n as a degenerate example of a function of some degree $m > n$. Lowering the degree means approximating a function of degree n by a function of degree m for some $m < n$. After we introduce bipolynomial surfaces, we will also study degree splitting and degree joining.

10. Degree raising

A parabola F is a polynomial curve of degree two, so it has a biaffine blossom $f(\bar{u}, \bar{v})$. But, if we like, we can also consider $F(\bar{u})$ to be a degenerate example of a cubic polynomial curve $G(\bar{u}) = F(\bar{u})$. The cubic G has a triaffine blossom $g(\bar{u}, \bar{v}, \bar{w})$. We know that $F(\bar{u}) = f(\bar{u}, \bar{u}) = g(\bar{u}, \bar{u}, \bar{u})$, of course; but it is interesting to ask what other relationships hold between the two blossoms f and g.

It turns out to be easy to express g in terms of f: we have

$$g(\bar{u}, \bar{v}, \bar{w}) = \frac{f(\bar{u}, \bar{v}) + f(\bar{u}, \bar{w}) + f(\bar{v}, \bar{w})}{3}.$$

To prove this claim, it is enough to note that the right-hand side is a symmetric, triaffine function whose diagonal agrees with $F = G$. For an arbitrary degree n and an arbitrary parameter space P, we have

$$g(\mathbf{u}_1, \ldots, \mathbf{u}_{n+1}) = \frac{1}{n+1} \sum_{i=1}^{n+1} f(\mathbf{u}_1, \ldots, \hat{\mathbf{u}}_i, \ldots, \mathbf{u}_{n+1}), \tag{10.1}$$

where the hat over the argument $\mathbf{u}_i$ indicates that that argument is omitted.

Both of the functions f and g that appear in Eq. (10.1) are multiaffine blossoms of F. Similarly, F also has two different homogeneous forms F_* and G_*, two different affine blossoms $f^{\otimes}$ and $g^{\otimes}$, two different multilinear blossoms f_* and g_*, and two different linear blossoms $f_*^{\otimes}$ and $g_*^{\otimes}$. In each case, one results from viewing F as having degree n, while the other results from viewing F as a degenerate function of degree $n + 1$. To distinguish between the various blossoms of the single function F, when confusion might arise, we shall prefix the word "blossom" with the relevant degree. Thus, we shall call f the ***multiaffine n-blossom*** of F, while g is the ***multiaffine (n+1)-blossom***, and so forth. Note that, of the six guises of a polynomial map shown in Fig. 5.1, the polynomial map itself, in the upper-left corner, is the only one where there is any chance of confusion about the intended degree.

To translate Eq. (10.1) into the world of tensors, we need to introduce an operator D on tensors that might be called ***total tensor differentiation***. To begin with, let $D: X[P] \to X[P]$ be the linear operator that takes an X-polynomial Φ on P into the polynomial

$$D(\Phi) := \sum_{\mathbf{u} \in P} \frac{\partial \Phi}{\partial X_{\mathbf{u}}}.$$

For example, we have

$$D(\bar{0}^2\bar{1}) = \frac{\partial(\bar{0}^2\bar{1})}{\partial\bar{0}} + \frac{\partial(\bar{0}^2\bar{1})}{\partial\bar{1}} = 2\bar{0}\bar{1} + \bar{0}^2.$$

If Φ and Ψ in $X[P]$ are affinely equivalent, then $D(\Phi)$ and $D(\Psi)$ will also be equivalent; hence, the operator D is also well-defined as a linear map $D\colon \mathcal{X}[P] \to \mathcal{X}[P]$ on tensors. In addition to being linear, the map D has the property $D(ab) = aD(b) + D(a)b$; such maps are called *derivations*. In fact, D is the unique derivation of the tensor algebra that maps each 1-tensor into its flavor: for e in $\mathcal{X}^1_*[P]$, we have $D(e) = \operatorname{Flav}(e)$. More generally, if e is a k-tensor, then $\operatorname{Flav}(D(e)) = k\operatorname{Flav}(e)$.

Proposition 10.2. *Let $F\colon P \to Q$ be an n-ic polynomial function, let $f^{\otimes}_*$ be the linear n-blossom of F, and let $g^{\otimes}_*$ be the linear $(n+1)$-blossom of F, that is, the linear blossom of the map $G\colon P \to Q$ that results from viewing F as a degenerate instance of a function of degree $n+1$. The functions $f^{\otimes}_*$ and $g^{\otimes}_*$ are related by the identity $g^{\otimes}_*(e) = f^{\otimes}_*\big(D(e)/(n+1)\big)$, where the operator D denotes total tensor differentiation.*

Proof. The map $e \mapsto D(e)/(n+1)$ is a flavor-preserving linear map from $P^{\otimes(n+1)}_*$ to $P^{\otimes n}_*$ that takes $\mathbf{u}^{n+1} \mapsto \mathbf{u}^n$. Thus, the composition $e \mapsto f^{\otimes}_*\big(D(e)/(n+1)\big)$ is a flavor-preserving linear map from $P^{\otimes(n+1)}_*$ to Q_* that takes $\mathbf{u}^{n+1} \mapsto F(\mathbf{u})$. By Cor. 4.6, the only such map is $g^{\otimes}_*$, so we must have $g^{\otimes}_*(e) = f^{\otimes}_*\big(D(e)/(n+1)\big)$ for all e. □

Prop. 10.2 allows us to answer many questions in a straightforward way. For example, if F is a parabola, what is a formula in terms of the 2-blossom f of F for the point $g(\bar{0},\bar{1},\bar{2})$, where g is the 3-blossom of F? We compute as follows: $g(\bar{0},\bar{1},\bar{2}) = g^{\otimes}(\bar{0}\bar{1}\bar{2}) = f^{\otimes}\big(D(\bar{0}\bar{1}\bar{2})/3\big) = f^{\otimes}\big((\bar{0}\bar{1}+\bar{0}\bar{2}+\bar{1}\bar{2})/3\big) = \big(f(\bar{0},\bar{1}) + f(\bar{0},\bar{2}) + f(\bar{1},\bar{2})\big)/3$. In some contexts, it might be better to have a formula for $g(\bar{0}\bar{1}\bar{2})$ as an affine combination of the Bézier points of F with respect to some reference interval, perhaps $[\bar{0},\bar{1}]$. If so, we can apply the normal-form operator β associated with the corresponding barycentric basis for L_*, the basis $\{\bar{0},\bar{1}\}$ in this case. Since $\beta(\bar{2}) = 2\bar{1} - \bar{0}$, we find that

$$g(\bar{0},\bar{1},\bar{2}) = \frac{2f(\bar{1},\bar{1}) + 2f(\bar{0},\bar{1}) - f(\bar{0},\bar{0})}{3}.$$

In other contexts, we might want a formula that labels the point $g(\bar{0},\bar{1},\bar{2})$ as a single value of the 2-blossom f. That is a trickier question, since such a formula will exist only if the 2-tensor $D(\bar{0}\bar{1}\bar{2})$ is simple. We shall return to that question later on.

Prop. 10.2 also gives us an easy proof of the standard degree-raising formula for Bézier curves. Suppose that we would like to compute, from the Bézier points of an n-ic curve segment $F([\bar{s},\bar{t}])$, the Bézier points of that same curve segment viewed as a degenerate curve of degree $n+1$. If f is the n-blossom of F and g is

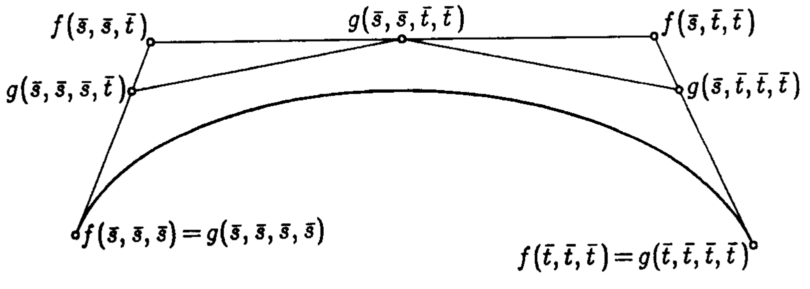

$f(\bar{s},\bar{s},\bar{t})$
$g(\bar{s},\bar{s},\bar{t},\bar{t})$
$f(\bar{s},\bar{t},\bar{t})$
$g(\bar{s},\bar{s},\bar{s},\bar{t})$
$g(\bar{s},\bar{t},\bar{t},\bar{t})$
$f(\bar{s},\bar{s},\bar{s})=g(\bar{s},\bar{s},\bar{s},\bar{s})$
$f(\bar{t},\bar{t},\bar{t})=g(\bar{t},\bar{t},\bar{t},\bar{t})$

Fig. 10.3. Raising the degree of a cubic curve

the $(n+1)$-blossom, we have, for $0 \le i \le n+1$,

$$\begin{aligned} g^{\otimes}(\bar{s}^i\bar{t}^{n+1-i}) &= f^{\otimes}(D(\bar{s}^i\bar{t}^{n+1-i})/(n+1)) \\ &= f^{\otimes}\left(\frac{i\bar{s}^{i-1}\bar{t}^{n+1-i} + (n+1-i)\bar{s}^i\bar{t}^{n-i}}{n+1}\right) \\ &= \frac{i}{n+1} f^{\otimes}(\bar{s}^{i-1}\bar{t}^{n-(i-1)}) + \left(1 - \frac{i}{n+1}\right) f^{\otimes}(\bar{s}^i\bar{t}^{n-i}). \end{aligned}$$

Fig. 10.3 shows an example in which a cubic curve is viewed as a degenerate quartic.

We can also use Prop. 10.2 in the reverse direction, to figure out a formula in terms of the $(n+1)$-blossom g for a point that we already know in terms of the n-blossom f. For example, in the case $n = 2$, suppose that we would like a formula in terms of the 3-blossom g for the point $f(\bar{0},\bar{1})$. We should expect to make some arbitrary choices in the process of producing such a formula, because of the degeneracy of g. From Prop. 10.2, we are searching for a 3-tensor e on L with the property that $D(e)/3 = \bar{0}\bar{1}$. We can find a particular solution to this inhomogeneous differential equation by a little experimentation: the 3-tensor $e := (3\bar{0}^2\bar{1} - \bar{0}^3)/2 = \bar{0}^2\overline{1.5}$ happens to work. To get an arbitrary solution of the differential equation, we can add to e any solution c of the homogeneous equation $D(c) = 0$. To determine the solutions of the homogeneous equation, recall that the curve G is a degenerate cubic, with the degeneracy that $G^{(3)}(\vec{u}) \equiv 0$. By Prop. 8.2, this is equivalent to saying that $g^{\otimes}_{*}(\delta^3) = 0$. Since $g^{\otimes}_{*}(e) = f^{\otimes}_{*}(D(e)/3)$ and F is unconstrained, it had better be the case that $D(\delta^3) = 0$; and, indeed, that is easy to verify. In fact, the kernel of the linear map D is precisely the set of scalar multiples of δ^3. Thus, we have $f(\bar{0},\bar{1}) = g^{\otimes}(\bar{0}^2\overline{1.5} + w\delta^3)$ for any scalar w. The value $w = 0$ gives $f(\bar{0},\bar{1}) = g(\bar{0},\bar{0},\overline{1.5})$, while the value $w = -1/2$ gives the symmetric formula $f(\bar{0},\bar{1}) = g(\overline{-0.5},\bar{1},\bar{1})$.

This argument involved a particular case of an important general pattern: a derivative constraint on a polynomial curve or surface $F\colon P \to Q$ corresponds to a linear subspace of the nth tensor power space $P^{\otimes n}_{*}$ on which the linear blossom of F is required to be zero. As a second example of this general pattern, note that a cubic polynomial surface has ten Bézier points while a quadratic surface has only six. What are the four constraints on the Bézier points of a cubic surface that, if they are satisfied, imply that the surface is actually quadratic? Let $G\colon P \to Q$ denote the surface, and let $\Delta\mathbf{rst}$ be a reference triangle in P. The surface G will be quadratic as well as cubic if its third derivative vanishes identically, that is, if $d^3G(\xi_1, \xi_2, \xi_3) \equiv 0$. To interpret this identity as constraints on the Bézier points of G, let $\alpha := \mathbf{r} - \mathbf{t}$, $\beta := \mathbf{s} - \mathbf{r}$, and $\gamma := \mathbf{t} - \mathbf{s}$ be the three sides of the reference triangle viewed as vectors on P. Any two of the three vectors α, β, and γ form a basis for the space P_0 of vectors on P. If we pick, for example, α and β, we can conclude that the constraints on the Bézier points of G are given by $g_*^{\circledast}(\alpha^3) = g_*^{\circledast}(\alpha^2\beta) = g_*^{\circledast}(\alpha\beta^2) = g_*^{\circledast}(\beta^3) = 0$. The constraint $g_*^{\circledast}(\alpha^3)$ corresponds to the demand that the $G([\mathbf{t}, \mathbf{r}])$ edge curve of the triangular surface patch $G(\Delta\mathbf{rst})$ be a quadratic curve, and similarly for the constraint $g_*^{\circledast}(\beta^3) = 0$ and the $G([\mathbf{r}, \mathbf{s}])$ edge curve. The bad thing about these four constraints as a whole is a lack of symmetry; while the four tensors α^3, $\alpha^2\beta$, $\alpha\beta^2$ and β^3 do form a basis of the constraint space, they aren't a symmetric basis. A little linear algebra reveals that one symmetric basis consists of the three simple tensors α^3, β^3, and γ^3, and the compound tensor

$$e = 2\mathbf{r}^3 + 2\mathbf{s}^3 + 2\mathbf{t}^3 - 3\mathbf{r}^2\mathbf{s} - 3\mathbf{r}^2\mathbf{t} - 3\mathbf{s}^2\mathbf{t} - 3\mathbf{s}^2\mathbf{r} - 3\mathbf{t}^2\mathbf{r} - 3\mathbf{t}^2\mathbf{s} + 12\mathbf{rst}.$$

The structure of e is easier to understand if we arrange its coefficients in a triangular array:

$$\begin{array}{ccccccc} & & & +2\mathbf{r}^3 & & & \\ & & -3\mathbf{r}^2\mathbf{s} & & -3\mathbf{r}^2\mathbf{t} & & \\ & -3\mathbf{r}\mathbf{s}^2 & & +12\mathbf{rst} & & -3\mathbf{r}\mathbf{t}^2 & \\ +2\mathbf{s}^3 & & -3\mathbf{s}^2\mathbf{t} & & -3\mathbf{s}\mathbf{t}^2 & & +2\mathbf{t}^3 \end{array}$$

The fact that $g_*^{\circledast}(e)$ must be zero in order for G to be a quadratic gives us an affine relationship on the ten Bézier points of G: if we let $V := (g(\mathbf{r},\mathbf{r},\mathbf{r}) + g(\mathbf{s},\mathbf{s},\mathbf{s}) + g(\mathbf{t},\mathbf{t},\mathbf{t}))/3$ denote the centroid of the three vertex Bézier points, $C := g(\mathbf{r},\mathbf{s},\mathbf{t})$ denote the central Bézier point, and O denote the centroid of the remaining six Bézier points, the relation $g_*^{\circledast}(e)/6 = 0$ becomes $2C - 3O + V = 0$, or, equivalently, O is located two-thirds of the way from V towards C. That constraint, together with the demands that the three edge curves of the patch $G(\Delta\mathbf{rst})$ be quadratic curves, is enough to guarantee that G will be a quadratic surface.

Let us return now to the question that we postponed earlier. Let F be a parabola with the 2-blossom f and the 3-blossom g. Is the point $g(\bar{0}, \bar{1}, \bar{2})$ in the range of f, or not? As we learned in Section 9, this depends upon whether the tensor $D(\bar{0}\bar{1}\bar{2})$ is simple or compound. Using the Cartesian basis $\{\bar{0}, \delta\}$ for L_*, we have

$$D(\bar{0}\bar{1}\bar{2})/3 = \frac{\bar{0}\bar{1} + \bar{0}\bar{2} + \bar{1}\bar{2}}{3} = \frac{3\bar{0}^2 + 6\bar{0}\delta + 2\delta^2}{3}.$$

Using the Quadratic Formula, we find that this 2-tensor does factor:

$$D(\bar{0}\bar{1}\bar{2}) = \left(\bar{0} + \left(1 + \frac{1}{\sqrt{3}}\right)\delta\right)\left(\bar{0} + \left(1 - \frac{1}{\sqrt{3}}\right)\delta\right) \approx \overline{0.423}\,\overline{1.577}.$$

This time, we were lucky: we were able to factor the 2-tensor $D(e)$ over the reals. The following proposition reveals that, in fact, there was no luck involved.

Proposition 10.4. *Let $D\colon L_*^{\otimes(n+1)} \to L_*^{\otimes n}$ denote the total differentiation operator for tensors on an affine line L, where the coefficient field is the real numbers* $\mathbf{R}$. *For all n, if a is a simple $(n+1)$-tensor, then $D(a)$ will be a simple n-tensor. But the converse fails in a strong way: for $n \geq 4$, there exist simple n-tensors b with the property that every tensor a satisfying $D(a) = b$ is compound.*

Proof. (The result in Prop. 10.4 is quite sensitive to the coefficient field; as we observed in Section 9, all n-tensors on the line L are simple if we adopt the complex numbers as our coefficient field.)

Let a be a simple $(n+1)$-tensor on L, and let's ignore the trivial case $a = 0$; we want to show that $D(a)$ is a simple n-tensor. If a is not 0-flavored, then a can be written in the form $a = w\bar{u}_1 \cdots \bar{u}_{n+1}$, as a scalar multiple of a product of points $\bar{u}_i$ in P. If a is 0-flavored, however, then some of its factors are vectors rather than points. Let l be the largest integer such that δ^l divides a. Then, we have

$$a = w\bar{u}_1 \cdots \bar{u}_k \delta^{n+1-k} = w\delta^{n+1-k} \prod_{1 \leq i \leq k} (\bar{0} + u_i\delta)$$

for some real constants u_i and nonzero scalar w, where $k := n+1-l$. Since $D(\delta) = 0$, we have, in terms of these constants,

$$D(a) = w\delta^{n+1-k} \sum_{1 \leq i \leq k} \prod_{\substack{1 \leq j \leq n \\ j \neq i}} (\bar{0} + u_j\delta).$$

We can make things look more familiar by performing the change of variables $\bar{0} := Y$ and $\delta := -1$. This change of variables carries the tensor a into the monic univariate polynomial $A(Y)$ of degree k in the variable Y given by

$$A(Y) := (-)^{n+1-k} w \prod_{1 \leq i \leq k} (Y - u_i).$$

Note that the numbers u_i are precisely the roots of $A(Y)$. Furthermore, the same change of variables carries $D(a)$ into $A'(Y)$. Thus, we are left with the following problem: if a polynomial $A(Y)$ of degree k has all real roots, does its derivative $A'(Y)$ also have all real roots, or might some of the roots of $A'(Y)$ be complex? By Rolle's Theorem, the function $A'(Y)$ must have at least one real root in the interval between any two distinct roots of $A(Y)$. If the k roots of $A(Y)$ are all distinct, this is enough to conclude that $A'(Y)$ has a full set of $k - 1$ real roots. A polynomial $A(Y)$ with multiple roots is the limit of a sequence of polynomials with clusters of closely-spaced distinct roots. Therefore, its derivative $A'(Y)$ is the limit of a sequence of polynomials with all real roots, and hence $A'(Y)$ itself must have all real roots. We conclude that $D(a)$ is simple whenever a is simple.

We can use the same change of variables to investigate the converse question. Let b be a nonzero simple n-tensor on L; in fact, we shall choose b to be 1-flavored, so that $b = \bar{u}_1 \cdots \bar{u}_n = \prod_i (\bar{0} - u_i\delta)$. After the change of variables $\bar{0} := Y$ and $\delta := -1$, we are left with the monic univariate polynomial $B(Y)$ whose roots are

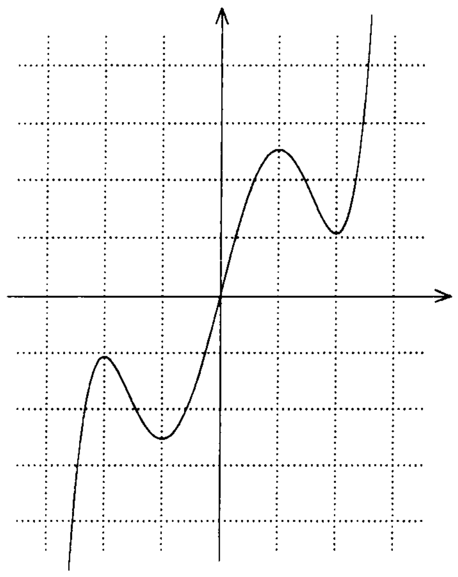


Fig. 10.5. The graph of the function $y = \int_0^x (Y+2)(Y+1)(Y-1)(Y-2)\,dY$

the u_i. An $(n+1)$-tensor a will satisfy $D(a) = b$ if and only if the polynomial $A(Y)$ that results from the change of variables satisfies $A'(Y) = B(Y)$, that is, if and only if $A(Y)$ is an integral of $B(Y)$. If $\int B(Y)$ represents one integral of $B(Y)$, the other integrals all differ by some constant: $A(Y) = \int B(Y) + C$ for some constant C of integration. Thus, we are left with the following problem: do there exist polynomials $B(Y)$ with all real roots but with the property that none of their integrals have all real roots? Such polynomials do indeed exist for $n \geq 4$; one quartic example is $B(Y) = (Y+2)(Y+1)(Y-1)(Y-2)$, whose integral $\int B(Y)$ is shown in Fig. 10.5. No matter how we shift the x-axis up or down, we can't make it intersect $\int B(Y)$ more than three times. Translating this example back into the blossoming world, we deduce that the value $f(\overline{-2}, \overline{-1}, \bar{1}, \bar{2})$ of the multiaffine 4-blossom f of a nondegenerate quartic F can't be labeled as a value $g(\bar{u}_1, \ldots, \bar{u}_5)$ of the multiaffine 5-blossom g of F. □

We can arrive at a more interesting quartic example by letting the pairs of roots $\{1, 2\}$ and $\{-1, -2\}$ coalesce to form two double roots, located, say, at s and at t. The resulting polynomial, $B(Y) = (Y-s)^2(Y-t)^2$ also has the property that no vertical shift of its integral $\int B(Y)$ has more than three real roots. Hence, the point $f(\bar{s}, \bar{s}, \bar{t}, \bar{t})$, which is the middle of the five Bézier points of the quartic segment $F([\bar{s}, \bar{t}])$, is one of the points that has no label of the form $g(\bar{u}_1, \ldots, \bar{u}_5)$.

Raising the degree of a polynomial curve gives us one more dimension's worth of labels, but Prop. 10.4 shows that the set of points for which we have at least one label never grows as a result of degree raising and, in fact, generally shrinks. Surfaces work out rather differently: there is no inclusion relationship in either direction, in general, between the points that can be labeled before the degree is raised and those that can be labeled after. Furthermore, this result continues to hold even if we adopt the complex numbers as the coefficient field.

Our first example will show that, by raising the degree of a polynomial surface, it may be possible to label points that couldn't be labeled formerly. Let $F\colon P \to Q$ be a nondegenerate quadratic surface (lying in some 5-flat), let f be its 2-blossom, let g be its 3-blossom, and consider the central Bézier point $g(\mathbf{r},\mathbf{s},\mathbf{t})$, where $\triangle\mathbf{rst}$ is a reference triangle for P. By Prop. 10.2, we have $g(\mathbf{r},\mathbf{s},\mathbf{t}) = f^{\circledast}((\mathbf{rs}+\mathbf{rt}+\mathbf{st})/3)$. In order for this point to lie in the range of f, we would have to be able to factor the polynomial

$$\frac{\mathbf{rs}+\mathbf{rt}+\mathbf{st}}{3}$$

into two linear factors. This is not possible, as we can determine simply by plugging the coefficients of this polynomial into the discriminant-like polynomial in Eq. (9.9).

Our second example will show that raising the degree of a polynomial surface may also result in having no labels for a point that formerly did have a label. Let $F\colon P \to Q$ be a nondegenerate cubic polynomial surface, let f be its 3-blossom, let g be its 4-blossom, and consider the point $f(\mathbf{r},\mathbf{s},\mathbf{t})$. We shall prove that the point $f(\mathbf{r},\mathbf{s},\mathbf{t})$ is not in the range of g. To prove this, we must consider all 4-tensors e on P that satisfy $D(e) = 4\mathbf{rst}$ and show that none of them is simple. The proof is not too difficult, but it takes awhile.

Before we begin the proof in earnest, let us count the degrees of freedom to check the plausibility of the result. From Prop. 4.9, we know that the tensor power space $P^{\circledast 4}$ has dimension $\binom{6}{2}-1 = 14$. In this 14-dimensional affine space, the simple tensors e form a subset of dimension 8, since there are two degrees of freedom in each of the four points that are factors of $e = \mathbf{u}_1\mathbf{u}_2\mathbf{u}_3\mathbf{u}_4$. The set of 4-tensors e that satisfy $D(e) = 4\mathbf{rst}$ is an affine subspace of dimension 5, since a 4-tensor a satisfies $D(a) = 0$ if and only if a is a linear combination of the five tensors ξ^4, $\xi^3\eta, \ldots, \eta^4$, where ξ and η are a basis for P_0. In general, we wouldn't expect a 5-flat to intersect a subset of dimension 8 if they are both lying in a space of dimension 14. Thus, it is plausible that these two sets of tensors are actually disjoint, as we claimed above.

The straightforward way to prove disjointness would begin by characterizing the set of simple 4-tensors on P algebraically, giving 6 dimensions' worth of constraints on their coordinates. Unfortunately, I don't know how to do that in practice even for simple 3-tensors on P, much less simple 4-tensors; see Challenge 9.10. Since the straightforward attack is not available, we shall resort to an ad hoc method that Andrew Odlyzko suggested to me.

Let e be a 4-tensor that satisfies $D(e) = 4\mathbf{rst}$; we want to show that e cannot be simple. As a first step, it is convenient to change from the barycentric $\{\mathbf{r},\mathbf{s},\mathbf{t}\}$ basis for P_* to a Cartesian basis, since that will simplify the behavior of the operator D. One obvious Cartesian basis is $\{\mathbf{r},\mathbf{t}-\mathbf{r},\mathbf{s}-\mathbf{r}\}$; but we shall make a somewhat more symmetric choice, as shown in Fig. 10.6. Let $\mathbf{q} := (\mathbf{r}+\mathbf{s}+\mathbf{t})/3$ be the centroid of $\triangle\mathbf{rst}$, let $\alpha := \mathbf{q}-\mathbf{s}$, and let $\beta := \mathbf{q}-\mathbf{t}$; we shall use the basis $\{\mathbf{q},\alpha,\beta\}$. Note that $\mathbf{r} = \mathbf{q}+\alpha+\beta$. In terms of this basis, the condition $D(e) = 4\mathbf{rst}$ becomes $D(e) = 4(\mathbf{q}+\alpha+\beta)(\mathbf{q}-\alpha)(\mathbf{q}-\beta) = 4\bigl(\mathbf{q}^3 - \mathbf{q}(\alpha^2+\alpha\beta+\beta^2) + \alpha\beta(\alpha+\beta)\bigr)$. Since $D(\mathbf{q}) = 1$ but $D(\alpha) = D(\beta) = 0$, we find upon integrating that we must have

$$e = \mathbf{q}^4 - 2\mathbf{q}^2(\alpha^2+\alpha\beta+\beta^2) + 4\mathbf{q}\alpha\beta(\alpha+\beta) + Z(\alpha,\beta),$$

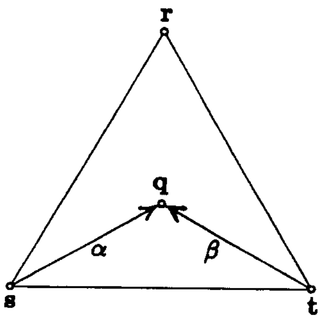


Fig. 10.6. Two different bases for P_*

where $Z(\alpha,\beta) = \sum_{i=0}^{4} z_i\alpha^{4-i}\beta^i$ is the constant of integration. We are left with the task of showing that no such 4-tensor e is simple.

If e were simple, its four factors would be points in P, which could be expressed in the form $(\mathrm{q} + a_i\alpha + b_i\beta)$ for scalars a_i and b_i, with i in $[1,4]$. Note that, if we substitute $\beta = 0$ in the polynomial $e(\mathrm{q},\alpha,\beta)$, we are left with the polynomial $e(\mathrm{q},\alpha,0) = \mathrm{q}^4 - 2\mathrm{q}^2\alpha^2 + z_0\alpha^4$. Since this polynomial only involves q to even powers, we deduce that the four numbers $\{a_1,a_2,a_3,a_4\}$ must have the form $\{+a,-a,+c,-c\}$ for some a and c. Similarly, by substituting $\alpha = 0$, we deduce that the four numbers $\{b_1,b_2,b_3,b_4\}$ must have the form $\{+b,-b,+d,-d\}$ for some b and d.

At first blush, it might seem that the numbering of the a_i and the numbering of the b_i could be related in $4! = 24$ different ways. A little thought shows that there are only two really different possibilities. In Case I, the sign-reversed pairs of a's line up with the sign-reversed pairs of b's, leading to the factorization

$$e = (\mathrm{q} + a\alpha + b\beta)(\mathrm{q} - a\alpha - b\beta)(\mathrm{q} + c\alpha + d\beta)(\mathrm{q} - c\alpha - d\beta).$$

In Case II, the sign-reversed pairs do not line up, leading to the form

$$e = (\mathrm{q} + a\alpha + b\beta)(\mathrm{q} - a\alpha - d\beta)(\mathrm{q} + c\alpha + d\beta)(\mathrm{q} - c\alpha - b\beta).$$

We can rule out Case I right away: any such polynomial can't involve any terms with q to the first power, but our e includes both $4\mathrm{q}\alpha^2\beta$ and $4\mathrm{q}\alpha\beta^2$. Thus, only Case II remains as an active possibility.

Of the 15 coefficients of $e(\mathrm{q},\alpha,\beta)$, we know the values of 10 of them, while the remaining 5 are the unknown constants of integration z_i. Each of the 10 known coefficients gives us an equation that relates the four quantities a, b, c, and d. Of these 10 equations, 5 turn out to be non-trivial:

$$\begin{aligned}
a^2 + c^2 &= 2 \text{ from the } \mathrm{q}^2\alpha^2 \text{ term;}\\
b^2 + d^2 &= 2 \text{ from the } \mathrm{q}^2\beta^2 \text{ term;}\\
(a+c)(b+d) &= 2 \text{ from the } \mathrm{q}^2\alpha\beta \text{ term;}\\
(a+c)(a-c)(b-d) &= 4 \text{ from the } \mathrm{q}\alpha^2\beta \text{ term;}\\
(a-c)(b+d)(b-d) &= 4 \text{ from the } \mathrm{q}\alpha\beta^2 \text{ term.}
\end{aligned}$$

The last two of these relations imply that $(a+c)=(b+d)$. In combination with the third relation, this implies that $(a+c)^2=2$. Since the first relation tells us that $a^2+c^2=2$, we deduce that $2ac=0$. Hence, the two numbers a and c are 0 and $\pm\sqrt{2}$ in some order. Similarly, the two numbers b and d must be 0 and $\pm\sqrt{2}$. But this implies that $(a+c)(a-c)(b-d)$ must be $\pm 2\sqrt{2}$, while the fourth relation states that it should be 4 instead. This contradiction completes the proof that no 4-tensor e satisfying $D(e)=4\mathbf{rst}$ can be simple.

11. Degree lowering

In the last section, we studied in some detail what happens when an n-ic is viewed as a degenerate case of an $(n+1)$-ic. In this section, we shall study a more general problem: how can we view an n-ic as an m-ic? If $m>n$, this involves raising the degree by $m-n$. If $m<n$, it involves lowering the degree by $n-m$. Of course, we can't lower the degree in general without changing the function. If F is an n-ic function, lowering the degree of F to m means approximating F in some sense by an m-ic function G. The sense of approximation that we will use is osculation at a fixed point. That is, we shall choose a point $\mathbf{r}$ in the domain space P, and we shall choose G to be the unique m-ic function that osculates F to mth order at $\mathbf{r}$. Note that, even in the degree raising case, where $m>n$, it is still true that the output m-ic function G osculates the input n-ic function F to mth order at $\mathbf{r}$. When $m>n$, however, we have $G=F$, independent of the location of $\mathbf{r}$.

The problems of degree raising and degree lowering can be tackled with either the coordinate-based or the coordinate-free approach. As it happens, we have already studied these problems from the coordinate-based approach, arriving at the answer described in Prop. 8.6. The following proposition restates that answer in a more symmetric fashion.

Proposition 11.1: Coordinate-based raising or lowering. *Let $F\colon P\to Q$ be an n-ic map, let $\mathbf{r}$ be a point in P, and let $\{\delta_1,\ldots,\delta_p\}$ be a basis for P_0. The m-ic map $G\colon P\to Q$ that osculates F to mth order at $\mathbf{r}$ is given by the formula*

$$m^{\underline{i}} g_*^{\circledast}(\delta_1^{i_1}\cdots\delta_p^{i_p}\mathbf{r}^{m-i}) = n^{\underline{i}} f_*^{\circledast}(\delta_1^{i_1}\cdots\delta_p^{i_p}\mathbf{r}^{n-i}),$$

where $i=i_1+\cdots+i_p$ and i lies in $[0,m]$.

Proof. Prop. 8.4 tells us that the left-hand and right-hand sides are corresponding values of the ith derivatives of F and G at $\mathbf{r}$. Note that, if $m>n$, the formula correctly specifies that the ith derivative of G should be identically zero whenever $i>n$. $\square$

In the coordinate-free approach, degree raising and degree lowering come out quite differently. The coordinate-free approach to degree raising is an easy generalization of Eq. (10.1), but the coordinate-free approach to degree lowering is more complicated.

Proposition 11.2: Coordinate-free raising. *Let $F\colon P\to Q$ be an n-ic map, and let $m>n$. The m-ic map $G\colon P\to Q$ that results from viewing F as a degenerate* *m-ic is given by*

$$g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m) = \frac{1}{\binom{m}{n}} \sum_{\substack{I \subseteq \{1,\ldots,m\} \\ |I|=n}} f^{\otimes}\Bigl(\prod_{i \in I} \mathbf{u}_i\Bigr). \tag{11.3}$$

Proof. Eq. (11.3) clearly defines a symmetric m-affine function g with the correct diagonal values. Alternatively, we could derive Eq. (11.3) by applying the total tensor differentiation operator D of Prop. 10.2 repeatedly, a total of m times. □

Proposition 11.4: Coordinate-free lowering. *Let $F: P \to Q$ be an n-ic map, and let $m < n$. The m-ic map $G: P \to Q$ that osculates F to mth order at the point $\mathbf{r}$ in P is given by*

$$g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m) = \binom{n}{m} \sum_{\substack{I \subseteq \{1,\ldots,m\} \\ k=m-|I|}} (-)^k \frac{l}{k+l} f^{\otimes}\Bigl(\mathbf{r}^{k+l} \prod_{i \in I} \mathbf{u}_i\Bigr), \tag{11.5}$$

where $l := n - m$ denotes the amount by which the degree is being lowered.

Proof. Suppose that we begin with an m-ic map G and that we produce a degenerate map F of degree n by raising the degree of G by l. For such an F, the m-ic map that osculates F to mth order at $\mathbf{r}$ is just the map G that we started with, and it doesn't depend on $\mathbf{r}$. Let us first check that Eq. (11.5) gives the correct answer in this special case. The blossom f of the degree-raised F will be given in terms of the blossom g of G by Eq. (11.3), but with the pairs (f, g) and (n, m) interchanged. To avoid confusion, let us write out the interchanged version:

$$f^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_n) = \frac{1}{\binom{n}{m}} \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=m}} g^{\otimes}\Bigl(\prod_{i \in I} \mathbf{u}_i\Bigr). \tag{11.6}$$

If we substitute Eq. (11.6) in for each occurrence of $f^{\otimes}$ in Eq. (11.5), we should get massive cancellation, leading to the tautology $g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m) = g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m)$.

To verify that this does happen, let $I' \subseteq \{1, \ldots, m\}$, let $k' := m - |I'|$, and consider the term

$$T := g^{\otimes}\Bigl(\mathbf{r}^{k'} \prod_{i \in I'} \mathbf{u}_i\Bigr).$$

When we substitute Eq. (11.6) into Eq. (11.5), what multiple of T do we end up with? The term corresponding to $I \subseteq \{1, \ldots, m\}$ and $k = m - |I|$ in Eq. (11.5) will contribute a multiple of T precisely when $I \supseteq I'$. In particular, such an I-term will contribute

$$(-)^k \frac{l}{k+l} \binom{k+l}{k'} T$$

since, when we apply Eq. (11.6) to the term $f^{\otimes}(\mathbf{r}^{k+l} \prod_{i \in I} \mathbf{u}_i)$, we have to choose which k' out of the $k + l$ factors of $\mathbf{r}$ to keep. The number of sets I that satisfy

$I' \subseteq I \subseteq \{1, \ldots, m\}$ and have size $|I| = m - k$ is $\binom{k'}{k}$. Therefore, the coefficient of T in the final result is given by the binomial coefficient sum

$$\sum_{0 \leq k \leq k'} (-)^k \frac{l}{k+l} \binom{k+l}{k'} \binom{k'}{k}.$$

If $k' = 0$, which implies $I' = \{1, \ldots, m\}$, this sum evaluates to 1 trivially; that case gives us the desired right-hand side $g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m)$. If $k' > 0$, the sum evaluates to 0 by straightforward binomial-coefficientology, which is left as an exercise. Thus, Eq. (11.5) does produce the correct answer when the F on the right-hand side was generated by degree raising.

To handle the general case, let F be any n-ic map, and let G be the m-ic map that osculates F to mth order at $\mathbf{r}$. If we raise the degree of G by l, we get an n-ic map H that also osculates F to mth order at $\mathbf{r}$. We have just verified that, if we apply Eq. (11.5) to H, rather than F, we do get back G. But note that every blossom application on the right-hand side of Eq. (11.5) involves $\mathbf{r}$ as an argument at least l times. Since H and F agree to mth order at $\mathbf{r}$, we can apply Prop. 8.8 to deduce that their blossoms h and f will agree on all argument bags that include at least l copies of $\mathbf{r}$. Hence, applying Eq. (11.5) to F will also return G, as claimed. $\square$

By the way, if we know that the n-ic map F is actually of degree m as well as being of degree n, we can simplify Eq. (11.5) somewhat. Note that, in this case, the point $\mathbf{r}$ is a free parameter: it doesn't matter where we decide to approximate F with an m-ic since F itself is an m-ic. We can reduce the number of terms in Eq. (11.5) by choosing an appropriate value of the free parameter $\mathbf{r}$. For example, if we choose $\mathbf{r} := \mathbf{u}_1$, we get, after a little algebra, a formula with only 2^{m-1} terms instead of 2^m terms:

$$g^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_m) = \binom{n}{m} \sum_{\substack{I \subseteq \{2, \ldots, m\} \\ k = m - |I|}} \frac{(-)^{k+1}\, l}{(k+l)(k+l-1)} f^{\otimes}\Big(\mathbf{u}_1^{k+l} \prod_{i \in I} \mathbf{u}_i\Big), \tag{11.7}$$

where $l := n - m$ is, once again, the amount of lowering. (A technical point: in the special case $m = 0$, this trick of setting $\mathbf{r} := \mathbf{u}_1$ doesn't work, because there is no such thing as $\mathbf{u}_1$.)

12. Bipolynomial surfaces

The graphics community recognizes two different types of surfaces defined by polynomials, and we have been focusing entirely on one of those two types so far. In this section, we shall redress that imbalance.

Let $F\colon \mathbf{R} \times \mathbf{R} \to Q$ be a polynomial function defined on the two-dimensional parameter space $\mathbf{R} \times \mathbf{R}$. By definition, this means that each coordinate of the point $F(u, v)$ is given by a polynomial in the two real variables u and v. There are two different types of degree bound that we could put on these coordinate polynomials. In a ***tensor product surface*** of degree $(m; n)$, the coordinates of $F(u, v)$ are required to have degree no higher than m in the variable u and degree no higher than n in the variable v separately. In a ***triangular patch surface*** of degree n, the coordinates of $F(u,v)$ are required to have total degree no higher than n in the variables u and v jointly.

The phrases "tensor product surface" and "triangular patch surface" are the current standard names in computer-aided geometric design for these two types of surfaces [5]. In this paper, however, we shall uncover good reasons why both of those names are poor choices. Fortunately, in the latter case, we have already become comfortable with a better name: a "triangular patch surface" of degree n is precisely what we have been calling simply a ***polynomial surface*** of degree n. That is, if we place a bound on the total degree of the coordinates of $F(u,v)$, the resulting condition is independent of the choice of affine frame for the domain space $P = \mathbf{R} \times \mathbf{R}$, and we can reinterpret F as a polynomial function $F(\langle u,v\rangle) = F(\mathbf{u})$ defined on a two-dimensional affine space P.

In "tensor product surfaces" of degree $(m;n)$, on the other hand, we place separate bounds on the degrees of the coordinates of $F(u,v)$ in u and in v. The resulting condition on F does depend on the product structure of the domain $\mathbf{R}\times\mathbf{R}$. For any fixed v, the map $u \mapsto F(u,v)$ is a polynomial function of degree m; for any fixed u, the map $v \mapsto F(u,v)$ is a polynomial function of degree n. Thus, it is quite natural to refer to F itself as a ***bipolynomial function*** of degree $(m;n)$, in particular, a ***bipolynomial surface***. We shall use that name as our replacement for "tensor product surface." Note that, when both degrees m and n happen to be 3, it is already standard practice to refer to the resulting surface as a ***bicubic***, so the name "bipolynomial" shouldn't sound too strange.

Note that every polynomial surface of degree n is also a bipolynomial surface of degree $(n;n)$. Similarly, every bipolynomial surface of degree $(m;n)$ is also a polynomial surface of degree $m+n$. Thus, the distinction between the two types of surfaces arises only when we enforce some bound on the degree.

Blossoming, tensoring, and homogenizing all apply to bipolynomial functions $F: U\times V \to Q$ of degree $(m;n)$ pretty much the same way that they did to polynomial functions. In the remainder of this section, we shall sketch that development briefly, concentrating on the differences between the polynomial and bipolynomial theories. For brevity, we shall discuss only the case where both of the domain spaces U and V are affine lines. The theory works perfectly well when U and V have dimension greater than one, but those cases don't come up often in computer-aided geometric design. If there were need, we could also generalize the results below to apply to tripolynomial functions $F: U \times V \times W \to Q$, or, more generally, to multipolynomial functions $F: \prod_i U_i \to Q$.

A bipolynomial surface $F: U\times V \to Q$ of degree $(m;n)$ can be viewed, in Curried fashion, as a higher-order function $F: U \to (V \to Q)$, that is, as an m-ic curve of n-ic curves. More precisely, let $\mathrm{Poly}^n(V,Q)$ denote the set of all polynomial curves $G: V \to Q$ of degree n, turned into an affine space by taking affine combinations pointwise. Note that, if V and Q are finite-dimensional, then $\mathrm{Poly}^n(V,Q)$ will be finite-dimensional also, so we aren't violating our prohibition against infinitely many dimensions. Saying that F is "an m-ic curve of n-ic curves" is really saying that we can view F as an m-ic function $F: U \to \mathrm{Poly}^n(V,Q)$. Vice versa, we could also view F as an n-ic function $F: V \to \mathrm{Poly}^m(U,Q)$. This Curried point of view is worth

	Bipolynomial map $F: U \times V \to Q$ bipoly. of degree $(m;n)$	**Homogeneous bipolynomial map** $F_*: U_* \times V_* \to Q_*$ homogeneous bipoly. of degree $(m;n)$ flavor-biexponentiating
Blossoming	**Multiaffine blossom** $f: U^m \times V^n \to Q$ $(m;n)$-symmetric $(m+n)$-affine	**Multilinear blossom** $f_*: (U_*)^m \times (V_*)^n \to Q_*$ $(m;n)$-symmetric $(m+n)$-linear flavor-multiplicative
Tensoring	**Affine blossom** $f^{\otimes}: U^{\otimes m} \otimes V^{\otimes n} \to Q$ affine	**Linear blossom** $f_*^{\otimes}: U_*^{\otimes m} \otimes V_*^{\otimes n} \to Q_*$ linear flavor-preserving

Homogenizing

Fig. 12.1. The six guises of a bipolynomial map

keeping in mind, since it can often be used to reduce a bipolynomial problem to a combination of two polynomial problems.

Fig. 12.1 provides an overview of how our three principles transform a bipolynomial surface, analogous to the overview in Fig. 5.1 of the transformations of a polynomial function. Given a function of any one of the six types, there exist unique functions of the other five types that correspond. We shall begin our whirlwind tour by moving down the left column. But before we do so, there is a notational question to address. In this paper, we have chosen to distinguish between a point $\bar{r}$ in the affine line L and a scalar r in the coefficient field $\mathbf{R}$. Now that there are two different affine lines U and V involved, we have to invent some new notation to distinguish points in U from points in V. We shall use slanted bars: if r is a scalar, let $\grave{r}$, read "r-in," denote the point with coordinate r in the line U, and let $\acute{r}$, read "r-out," denote the point with coordinate r in the line V. Thus, a typical point on the surface F will be written $F(\grave{u}, \acute{v})$.

The multiaffine blossom of a bipolynomial surface $F: U \times V \to Q$ of degree $(m;n)$ is a multiaffine function $f: U^m \times V^n \to Q$ that is symmetric in its first m arguments, symmetric in its last n arguments, and satisfies the correspondence identity $F(\grave{u}, \acute{v}) = f(\grave{u}, \ldots, \grave{u}; \acute{v}, \ldots, \acute{v})$. We shall abbreviate this type of symmetry condition by saying that f is $(m;n)$-*symmetric*. And we shall use a semicolon instead of a comma to help separate the first m arguments of f from its last n arguments. A variant of the Blossoming Principle tells us that every bipolynomial surface has a unique multiaffine blossom; this variant can be proved either from scratch or by viewing F in Curried fashion and applying Prop. 2.1 twice.

Don't get confused between the multiaffine blossoms of the two common types of surfaces. The multiaffine blossom of a bipolynomial surface $F: U \times V \to Q$ of degree $(m; n)$ takes, as arguments, m points in U and n points in V. If we think in terms of coordinates, that is an ordered pair of sequences of real numbers. The multiaffine blossom of a polynomial surface $F: P \to Q$ of degree n takes, as arguments, n points in the plane P. If we think in terms of coordinates, that is a sequence of ordered pairs of real numbers.

The Bézier theory of a bipolynomial surface starts by choosing reference intervals $[\grave{p}, \grave{q}]$ for U and $[\acute{r}, \acute{s}]$ for V. Each of the first m arguments of f can then be expressed as an affine combination of $\grave{p}$ and $\grave{q}$, while each of the last n arguments is an affine combination of $\acute{r}$ and $\acute{s}$. If we express all $m + n$ arguments of f in this way and then invoke the multiaffineness of f, we get a formula that expresses the blossom value $f(\grave{u}_1, \ldots, \grave{u}_m; \acute{v}_1, \ldots, \acute{v}_n)$ as a sum of 2^{m+n} terms, each of which is some coefficient times a Bézier point of the rectangular surface patch $F([\grave{p}, \grave{q}] \times [\acute{r}, \acute{s}])$, where the Bézier points are the $(m + 1)(n + 1)$ blossom values

$$f(\underbrace{\grave{p}, \ldots, \grave{p}}_{m-i}, \underbrace{\grave{q}, \ldots, \grave{q}}_{i}; \underbrace{\acute{r}, \ldots, \acute{r}}_{n-j}, \underbrace{\acute{s}, \ldots, \acute{s}}_{j})$$

for i in $[0, m]$ and j in $[0, n]$.

The de Casteljau Algorithm for a bipolynomial surface F computes the blossom value $f(\grave{u}_1, \ldots, \grave{u}_m; \acute{v}_1, \ldots, \acute{v}_n)$ by performing $m + n$ stages of linear interpolations, starting with the Bézier points. Each blossom argument provides the interpolation ratio for one of those stages, and the arguments can be used in any order. Fig. 12.2 shows an example of a bipolynomial surface patch of degree $(2; 1)$ defined on the rectangle $[\grave{0}, \grave{3}] \times [\acute{0}, \acute{3}]$, along with all of the construction lines that might be used when locating the point $F(\grave{1}, \acute{2}) = f(\grave{1}, \grave{1}; \acute{2})$.

We can also generalize from using Bézier frames to using de Casteljau frames. Let $(\grave{p}_1, \ldots, \grave{p}_{2m})$ be a reference sequence of $2m$ points in U that satisfies the condition $p_{m+i} \neq p_j$ for $1 \leq i \leq j \leq m$ and, similarly, let $(\acute{r}_1, \ldots, \acute{r}_{2n})$ be a sequence of $2n$ points in V that satisfies the condition $r_{n+i} \neq r_j$ for $1 \leq i \leq j \leq n$. Then, the bipolynomial surface F can be uniquely specified by arbitrarily specifying its *poles*, which are the blossom values $f(\grave{p}_{i+1}, \ldots, \grave{p}_{i+m}; \acute{r}_{j+1}, \ldots, \acute{r}_{j+n})$ for i in $[0, m]$ and j in $[0, n]$.

Moving down the left-hand column of Fig. 12.1, we next come to the affine blossom $f^{\otimes}$. We shall begin the construction of the affine blossom $f^{\otimes}$ by viewing the multiaffine blossom f in a Curried fashion. Letting the U-arguments vary first, and then the V-arguments, we can turn f into a symmetric, m-affine function $g: U^m \to \mathrm{SMA}(V^n, Q)$, where $\mathrm{SMA}(V^n, Q)$ is the affine space consisting of all symmetric, multiaffine functions from V^n to Q. Applying the Tensoring Principle from Prop. 4.5, we can convert g into an affine function $g^{\otimes}: U^{\otimes m} \to \mathrm{SMA}(V^n, Q)$. Switching things around and letting the V-arguments vary first, we can view $g^{\otimes}$ as a symmetric, n-affine function $h: V^n \to \mathrm{Affine}(U^{\otimes m}, Q)$. Applying Prop. 4.5 again, we get an affine function $h^{\otimes}: V^{\otimes n} \to \mathrm{Affine}(U^{\otimes m}, Q)$. Switching back to non-Curried form, we have converted f into a biaffine function $f^{\otimes;\otimes}: U^{\otimes m} \times V^{\otimes n} \to Q$, which satisfies the identity $f(\grave{u}_1, \ldots, \grave{u}_m; \acute{v}_1, \ldots, \acute{v}_n) = f^{\otimes;\otimes}(\grave{u}_1 \cdots \grave{u}_m, \acute{v}_1 \cdots \acute{v}_n)$.

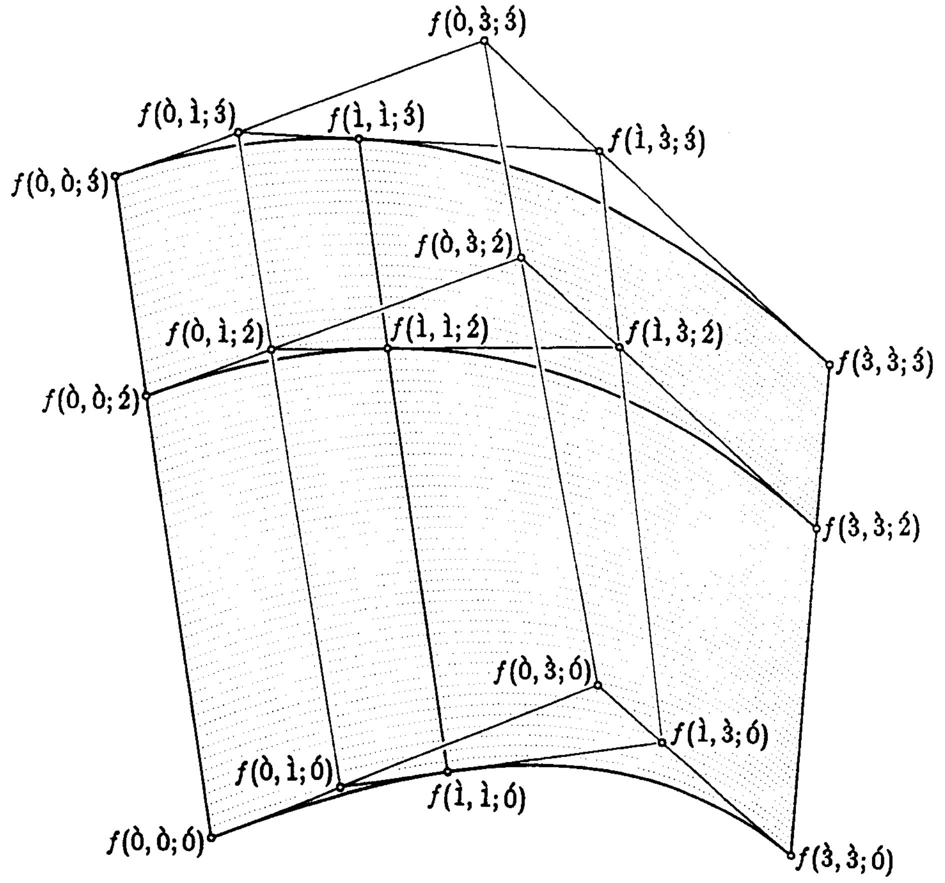

$f(\grave{0},\grave{3};\acute{3})$
$f(\grave{0},\grave{1};\acute{3})$
$f(\grave{1},\grave{1};\acute{3})$
$f(\grave{1},\grave{3};\acute{3})$
$f(\grave{0},\grave{0};\acute{3})$
$f(\grave{0},\grave{3};\acute{2})$
$f(\grave{0},\grave{1};\acute{2})$
$f(\grave{1},\grave{1};\acute{2})$
$f(\grave{1},\grave{3};\acute{2})$
$f(\grave{3},\grave{3};\acute{3})$
$f(\grave{0},\grave{0};\acute{2})$
$f(\grave{3},\grave{3};\acute{2})$
$f(\grave{0},\grave{3};\acute{6})$
$f(\grave{1},\grave{3};\acute{6})$
$f(\grave{0},\grave{1};\acute{6})$
$f(\grave{1},\grave{1};\acute{6})$
$f(\grave{0},\grave{0};\acute{6})$
$f(\grave{3},\grave{3};\acute{6})$

Fig. 12.2. The de Casteljau Algorithm for a bipolynomial surface of degree $(2;1)$

We need a new variant of the Tensoring Principle in order to finish the job, to turn the biaffine function $f^{\otimes;\otimes}$ into an affine function $f^{\otimes}$. The problem is that the biaffine function $f^{\otimes;\otimes}$ is not symmetric; indeed, its two arguments have different types, so it doesn't even make sense to ask whether it is symmetric. Hence, we need an asymmetric variant of tensoring.

Exercise 12.3: The Tensoring Principle, asymmetric affine variant. If S and T are affine spaces, construct a space $S \otimes T$ with the property that biaffine maps g with domain $S \times T$ are equivalent to affine maps $g^{\otimes}$ with domain $S \otimes T$ under the correspondence $g(\mathbf{s}, \mathbf{t}) = g^{\otimes}(\mathbf{s} \otimes \mathbf{t})$, where the mapping $(\mathbf{s}, \mathbf{t}) \mapsto \mathbf{s} \otimes \mathbf{t}$ is a fixed biaffine mapping from $S \times T$ to $S \otimes T$. Hint: Consider XY-polynomials in the formal variables $X_{\mathbf{s}}$ for $\mathbf{s}$ in S and $Y_{\mathbf{t}}$ for $\mathbf{t}$ in T. Choosing affine frames for S and T, define the normal form of an XY-polynomial, and define when two XY-polynomials are affinely equivalent. One model for $S \otimes T$ is then the space of affine equivalence classes of 1-flavored XY-polynomials that are $(1;1)$-homogeneous, that is, homogeneous of degree one in the X-variables and also homogeneous of degree one in the Y-variables. In this model, we have $\mathbf{s} \otimes \mathbf{t} := [X_{\mathbf{s}} Y_{\mathbf{t}}]$.

Some comments on Ex. 12.3. First, the reason for using two different formal symbols X and Y is to allow for the case $S = T$, that is, to enable the construction of $S \otimes S$. Note that we must distinguish between $\mathbf{s}_1 \otimes \mathbf{s}_2 = [X_{\mathbf{s}_1} Y_{\mathbf{s}_2}]$ and $\mathbf{s}_2 \otimes \mathbf{s}_1 = [X_{\mathbf{s}_2} Y_{\mathbf{s}_1}]$ in the case $S = T$, since there is no guarantee that $g(\mathbf{s}_1, \mathbf{s}_2) = g(\mathbf{s}_2, \mathbf{s}_1)$. This lack of commutativity also explains why we don't use simple multiplication "$\mathbf{s}_1\mathbf{s}_2$" to denote the tensor product of elements in the asymmetric variant. Warning: with our conventions, the spaces $S \otimes S$ and $S^{\otimes 2}$ are not the same; symmetry is implied in the latter case, but not in the former.

Second, the hint in Ex. 12.3 is phrased so as to emphasize the parallels with the construction of the symmetric tensor power space $P^{\otimes n}$. For the asymmetric tensor product $S \otimes T$ of two spaces, the only formal polynomials that are really used are $(1;1)$-homogeneous, so it wouldn't be hard to do without polynomials entirely. If one does choose to use formal polynomials, however, there are interesting connections with yet another variant of the tensor-product construction: the tensor product of algebras. In particular, the algebra of XY-polynomials is actually the tensor product $X[S] \otimes Y[T]$ of the algebras $X[S]$ and $Y[T]$. If we form equivalence classes by rewriting both X-variables and Y-variables in a normal form, the result is the tensor product $\mathcal{X}[S] \otimes \mathcal{Y}[T]$ of the algebras $\mathcal{X}[S]$ and $\mathcal{Y}[T]$, whose elements are ***tensors on*** $(S;T)$. The tensor product $S \otimes T$ consists of the 1-flavored $(1;1)$-tensors on $(S;T)$.

Our motivation for Ex. 12.3 was that we had transformed the $(m;n)$-ic surface $F\colon U \times V \to Q$ in Fig. 12.1 into an asymmetric biaffine function $f^{\otimes;\otimes}\colon U^{\otimes m} \times V^{\otimes n} \to Q$. Applying Ex. 12.3 finishes the job, converting $f^{\otimes;\otimes}$ into the ***affine blossom*** of F, which is the affine function $f^{\otimes}\colon U^{\otimes m} \otimes V^{\otimes n} \to Q$ that satisfies the identity $f(\grave{u}_1, \ldots, \grave{u}_m; \grave{v}_1, \ldots, \grave{v}_n) = f^{\otimes}(\grave{u}_1 \cdots \grave{u}_m \otimes \grave{v}_1 \cdots \grave{v}_n)$. Actually, if we apply Ex. 12.3 in the most straightforward way, setting $S := U^{\otimes m}$ and $T := V^{\otimes n}$, we end up with an iterated tensor space as the domain. An equivalent but simpler method is to set $S := U$ and $T := V$ and then to consider $U^{\otimes m} \otimes V^{\otimes n}$ as consisting of the 1-flavored $(m;n)$-tensors on $(U;V)$. Note that the algebra $\mathcal{X}[U] \otimes \mathcal{Y}[V]$ of all tensors on $(U;V)$ is a commutative algebra: if $a_1 \otimes b_1$ is a $(k_1;l_1)$-tensor on $(U;V)$ and $a_2 \otimes b_2$ is a $(k_2;l_2)$-tensor on $(U;V)$, then their product is $(a_1 \otimes b_1)(a_2 \otimes b_2) = a_1 a_2 \otimes b_1 b_2$, which is a $(k_1 + k_2; l_1 + l_2)$-tensor on $(U;V)$. The only noncommutative multiplication in sight is the one that takes a tensor on U and a tensor and V and turns that pair into a tensor on $(U;V)$. That noncommutative operation on elements is the one that we are writing with "$\otimes$".

The asymmetric tensor product operation on spaces that corresponds to the big "$\otimes$" in the domain $U^{\otimes m} \otimes V^{\otimes n}$ of the affine blossom is the instance of tensoring that gave bipolynomial surfaces the name "tensor product surfaces." Note that, from our point of view, there is nothing about bipolynomial surfaces that connects them more intimately than polynomial surfaces to the concept of the tensor product. The only difference is that, in the bipolynomial case, one of the tensor products involved in building the domain $U^{\otimes m} \otimes V^{\otimes n}$ of the affine blossom is asymmetric, while, in the polynomial case, all of the tensor products involved in building $P^{\otimes n}$ are symmetric. This observation suggests that the name "tensor-product surfaces" is a poor choice.

The right-hand column in Fig. 12.1 demands only a few remarks. The homogenized bipolynomial map F_* is called *flavor-biexponentiating*, meaning that $\operatorname{Flav}(F_*(a,b)) = (\operatorname{Flav}(a))^m(\operatorname{Flav}(b))^n$. The domain of the linear blossom $f_*^{\otimes}$ is most simply thought of as the set of all $(m;n)$-tensors on $(U;V)$ of any flavor. That precise set doesn't have an abstract name by our conventions; but there are several abstractly-named sets that are canonically isomorphic to that set. One is $(U^{\otimes m} \otimes V^{\otimes n})_*$, the linearization of the domain $U^{\otimes m} \otimes V^{\otimes n}$ of the affine blossom. Another is $U_*^{\otimes m} \otimes V_*^{\otimes n}$, in which the big "$\otimes$" denotes the asymmetric tensor product of linear spaces in the linear world, the same variant of the tensor-product construction that arises in textbooks on linear algebra.

Exercise 12.4. Let S and T be affine spaces of dimensions s and t and let W and Z be linear spaces of dimensions w and z. Verify the following four formulas (one of which is just Prop. 4.9): $\dim(S \otimes T) = st + s + t$; $\dim(S^{\otimes n}) = \binom{s+n}{n} - 1$; $\dim(W \otimes Z) = wz$; and $\dim(W^{\otimes n}) = \binom{w+n-1}{n}$.

Just as in the polynomial case, one advantage of the homogenized blossoms is that we can compute derivatives by replacing some of the points among the blossom arguments with vectors. If we let μ denote the unit vector on U and ν denote the unit vector on V, we have

$$\frac{\partial^k}{\partial u^k}\frac{\partial^l}{\partial v^l}F(\grave{u},\acute{v}) = m^{\underline{k}}n^{\underline{l}}f_*^{\otimes}(\grave{u}^{m-k}\mu^k \otimes \acute{v}^{n-l}\nu^l). \tag{12.5}$$

The falling-factorial coefficients are once again annoying, but we can wash out their effects if we consider osculating flats: the osculating flat $\operatorname{Osc}_{k;l} F(\grave{u},\acute{v})$ is precisely the locus of the points $f^{\otimes}(e(\grave{u}^{m-k} \otimes \acute{v}^{n-l}))$ as e varies over all 1-flavored $(k;l)$-tensors on $(U;V)$, that is, as e varies over $U^{\otimes k} \otimes V^{\otimes l}$. This implies that, if $\grave{u}$ appears p times among the arguments to f and $\acute{v}$ appears q times, the resulting blossom value must lie in the flat $\operatorname{Osc}_{m-p;n-q} F(\grave{u},\acute{v})$. For a nondegenerate bipolynomial surface F, this differential perspective allows us to compute the blossom value $f(\grave{u}_1,\ldots,\grave{u}_m;\acute{v}_1,\ldots,\acute{v}_n)$ by intersecting the osculating flats corresponding to all distinct pairs $(\grave{u}_i,\acute{v}_j)$.

The last topic that we will tackle for bipolynomial surfaces is the geometry of a surface and its multiaffine blossom. As in the case of polynomial curves and surfaces, the best way to study this question is to back up through the affine blossom and to study instead the geometry of its domain space $U^{\otimes m} \otimes V^{\otimes n}$. Tensors in this space of the form $\grave{u}^m \otimes \acute{v}^n$ correspond to points on the surface F; let us call such tensors *perfect $(m;n)$th powers*. Simple tensors, that is, tensors of the form $\grave{u}_1 \cdots \grave{u}_m \otimes \acute{v}_1 \cdots \acute{v}_n$, correspond to points for which we have multiaffine labels.

Things get non-trivial as soon as the total degree $m+n$ exceeds one; consider the case $m = n = 1$. A bipolynomial surface F of degree $(1;1)$ is actually itself biaffine. Hence, F is equal to its multiaffine $(1;1)$-blossom f: $F(\grave{u},\acute{v}) = f(\grave{u};\acute{v})$. For simplicity, let use choose $[\grave{0},\grave{1}]$ and $[\acute{0},\acute{1}]$ as our reference segments in U and V. The surface patch $F([\grave{0},\grave{1}] \times [\acute{0},\acute{1}])$ has the four Bézier points $f(\grave{0};\acute{0})$, $f(\grave{0};\acute{1})$, $f(\grave{1};\acute{0})$, and $f(\grave{1};\acute{1})$. If those four points are affinely independent, the resulting surface F will be a hyperbolic paraboloid sitting in a 3-space, with the two families of lines $f(\grave{u};\,\cdot\,)$ and $f(\,\cdot\,;\acute{v})$ as its generators.

This is precisely the geometry that is revealed by a study of the domain $U^{\otimes 1} \otimes V^{\otimes 1} = U \otimes V$ of the affine blossom $f^{\otimes}$. Choosing the Cartesian bases $\{\grave{0}, \mu\}$ for U_* and $\{\acute{0}, \nu\}$ for V_*, a general $(1;1)$-tensor e on $(U;V)$ has the normal form

$$e = a\grave{0} \otimes \acute{0} + b\grave{0} \otimes \nu + c\mu \otimes \acute{0} + d\mu \otimes \nu,$$

where $a = \mathrm{Flav}(e)$. The set $U \otimes V$ consists of the 1-flavored $(1;1)$-tensors, and is hence 3-dimensional, with $\langle b, c, d\rangle$ as a Cartesian coordinate system. The simple tensors in $U \otimes V$ are the products $\grave{u} \otimes \acute{v}$ of one point in each space and, in the case $m = n = 1$, every simple tensor is also a perfect $(1;1)$th power. The normal form $\beta(e)$ of the simple tensor $e = \grave{u} \otimes \acute{v}$ is given by $\beta(\grave{u} \otimes \acute{v}) = (\grave{0} + u\mu) \otimes (\acute{0} + v\nu) = \langle u, v, uv\rangle$. A tensor $e = \langle b, c, d\rangle$ in $U \otimes V$ will factor in this form only if $d = bc$; it is this equation that generates the curved shape of the hyperbolic paraboloid. One example of a compound tensor in $U \otimes V$ is given by $(\grave{0} \otimes \acute{0} + \grave{1} \otimes \acute{1})/2$. Since the two Bézier points $f(\grave{0}; \acute{0})$ and $f(\grave{1}; \acute{1})$ both lie on the paraboloid, but the line connecting them is not one of the generating lines, the midpoint of that line does not lie on the paraboloid.

13. Degree splitting and degree joining

The theory of bipolynomial surfaces gives rise to two new problems like degree raising, where we can study the relationship between two different blossoms of the same function. First, we can view a polynomial surface of degree n as a degenerate case of a bipolynomial surface of degree $(n;n)$. We shall call this adjustment of the degree *degree splitting*. Second, we can view a bipolynomial surface of degree $(k;l)$ as a degenerate case of a polynomial surface of degree $k + l$. We shall call this adjustment *degree joining*. In this section, we shall derive formulas for degree splitting and degree joining, using both the coordinate-based and the coordinate-free approaches.

We shall begin by reconsidering degree raising and degree lowering, but for the particular case of surfaces. Let $P = U \times V$ be a parameter plane with a product structure, and let $F: P \to Q$ be a polynomial or bipolynomial surface. For any nonnegative integers i and j, we can imagine computing the $(i;j)$th derivative

$$\frac{\partial^i}{\partial u^i}\frac{\partial^j}{\partial v^j}F(\langle u, v\rangle)$$

of F as a function of the position $\langle u, v\rangle$. If F is a polynomial surface of degree n, this derivative will be identically zero whenever $i + j > n$. If F is a bipolynomial surface of degree $(k;l)$, this derivative will be identically zero when either $i > k$ or $j > l$. If we associate the $(i;j)$th derivative of F with the lattice point $\langle i, j\rangle$ in some plane A, we can think of each type of surface as a geometric figure in A. The possibly-nonzero derivatives of an n-ic polynomial surface form an isoceles right triangle T_n in A, while the possibly-nonzero derivatives of a $(k;l)$-ic bipolynomial surface form a rectangle $R_{k;l}$. The processes of degree raising and degree lowering move around the borders of these A-figures.

To begin with, consider an n-ic polynomial surface F, whose A-figure is the triangle T_n. Raising the degree of F corresponds to moving the hypotenuse of this triangle outward to form T_m for $m > n$, choosing the output m-ic surface G to be degenerate in the sense that all of its derivatives in the strip $T_m \setminus T_n$ are zero. Lowering the degree of F corresponds to moving the hypotenuse inward to form T_m for $m < n$. In the process, we are forced to replace F by a simpler surface G, which osculates F to mth order at a chosen point $\mathbf{r}$, but whose derivatives in the strip $T_n \setminus T_m$ are zero rather than matching those of F. The formulas in Section 11 tell us how to raise or lower the degree of a polynomial surface, using either the coordinate-based or the coordinate-free approach.

Bipolynomial surfaces F of degree $(k; l)$ have two different degrees, each of which can be raised or lowered. Raising one of the degrees moves one edge of the rectangle $R_{k;l}$ away from the origin, while lowering that degree pulls that edge back towards the origin. The formulas in Section 11 can also be used to raise or lower either degree of a bipolynomial surface: while we work on one group of variables, the other group just comes along for the ride.

Thus, the formulas in Section 11 suffice for performing any adjustment to the degree of a surface that preserves the shape of the A-figure; they can take any triangle to any triangle, or any rectangle to any rectangle. It also makes sense, however, to convert back and forth between triangles and rectangles, that is, to osculate a polynomial surface with a bipolynomial surface or vice versa. The same rules apply: Points in the output A-figure that weren't in the input A-figure correspond to new derivatives that we choose to set equal to zero. If any of these exist, they are a source of degeneracy in the output surface. Points in the input A-figure that aren't in the output A-figure reflect behavior of the input that is of too high an order to be captured in the output. If any of these exist, the output will only osculate, rather than equal, the input.

Degree splitting is the transformation $T_n \mapsto R_{k;l}$, which osculates a polynomial surface with a bipolynomial one. Degree joining is the reverse transformation $R_{k;l} \mapsto T_n$. The following proposition describes degree splitting and joining from the coordinate-based approach.

Proposition 13.1: Coordinate-based splitting or joining. *Let $P = U \times V$ be an affine plane with a product structure; let $\mathbf{r} = \langle r_u, r_v \rangle$ be a point in P; let μ and ν denote the unit vectors on U and V; let $F: P \to Q$ be a polynomial surface of degree n; and let $G: U \times V \to Q$ be a bipolynomial surface of degree $(k; l)$. Consider the formula*

$$n^{\underline{i+j}} f_*^{\otimes}(\mu^i \nu^j \mathbf{r}^{n-i-j}) = k^{\underline{i}} l^{\underline{j}} g_*^{\otimes}(\mu^i \grave{r}_u^{k-i} \otimes \nu^j \acute{r}_v^{l-j}).$$

If this formula holds for $i \le k$ and $j \le l$, then G is the $(k; l)$-ic surface that osculates F to order $(k; l)$ at $\mathbf{r}$. If the formula holds for $i + j \le n$, then F is the n-ic surface that osculates G to order n at $\mathbf{r}$.

Proof. By Prop. 8.4 and Eq. 12.5 respectively, the left-hand and right-hand sides of that formula express the $(i; j)$th derivatives of F and G at $\mathbf{r}$. □

The coordinate-free approach is more complicated, and we shall therefore adopt a multi-step strategy. Note that a general degree joining of the form $R_{k;l} \mapsto T_n$ can be achieved in two steps: $R_{k;l} \mapsto T_{k+l} \mapsto T_n$. In the reverse direction, we can achieve a general splitting $T_n \mapsto R_{k;l}$ in three steps: $T_n \mapsto R_{n;n} \mapsto R_{k;n} \mapsto R_{k;l}$. Since we already know how to raise and lower degrees, it suffices for us to study the particular adjustments $R_{k;l} \mapsto T_{k+l}$ and $T_n \mapsto R_{n;n}$. Of course, if we build the formula for a general degree adjustment by using a multi-step strategy like one of these, the formula that results will probably be more complicated than necessary. But such strategies do provide, at least in principle, a way of building up any adjustment of the degree of a surface.

The particular adjustments $R_{k;l} \mapsto T_{k+l}$ and $T_n \mapsto R_{n;n}$ both have simple coordinate-free formulas. In the first case, we sum over $\binom{k+l}{k}$ partitions; in the second case, we sum over $n!$ permutations.

Proposition 13.2: Coordinate-free joining. *Let $F\colon U \times V \to Q$ be a bipolynomial surface of degree $(k;l)$, let f be the multiaffine $(k;l)$-blossom of F, and let g be the multiaffine $(k+l)$-blossom of F, that is, the multiaffine blossom of the map G that results from viewing F as a degenerate polynomial surface of degree $k+l$. The $(k+l)$-blossom g is given, in terms of the $(k;l)$-blossom f, by the formula:*

$$g(\langle u_1, v_1\rangle, \ldots, \langle u_{k+l}, v_{k+l}\rangle) = \frac{1}{\binom{k+l}{k}} \sum_{\substack{I \cap J = \emptyset \\ I \cup J = \{1,\ldots,k+l\} \\ |I|=k,\ |J|=l}} f^{\otimes}\Bigl(\prod_{i\in I} \grave{u}_i \otimes \prod_{j \in J} \acute{v}_j\Bigr). \quad \square$$

Proposition 13.3: Coordinate-free splitting. *Let $F\colon P \to Q$ be a polynomial surface of degree n, and suppose that the parameter plane P has a product structure $P = U \times V$. Let S_n denote the symmetric group on n letters, the set of all permutations of the set $\{1, \ldots, n\}$. The $(n;n)$-blossom g of F is given, in terms of the n-blossom f, by the formula:*

$$g(\grave{u}_1, \ldots, \grave{u}_n; \acute{v}_1, \ldots, \acute{v}_n) = \frac{1}{n!} \sum_{\pi \in S_n} f(\langle u_1, v_{\pi(1)}\rangle, \ldots, \langle u_n, v_{\pi(n)}\rangle). \quad \square$$

Part E: Spline Curves

It is time for a major change in focus. We have seen so far that blossoms are a useful tool for working with polynomial curves and surfaces. But a single polynomial curve or surface isn't a very complicated thing no matter how you think about it. The real power of the blossoming technology becomes apparent only when it is used to analyze splines. In particular, as de Casteljau discovered [14], blossoming extends to parametrically continuous spline curves with surprising ease and elegance.

14. On blossoms and joints

One comment before we begin: mathematicians generally think of a spline as a real-valued function, while we will be thinking of splines as taking values in an arbitrary affine object space Q. At some level, there isn't much difference between these two approaches. In particular, a function $F(\mathbf{u})$ is a spline in our sense if and only if each coordinate of $F(\mathbf{u})$ is a spline in the mathematician's sense. For our purposes, there are two advantages to dealing with a general object space Q rather than restricting ourselves to the real numbers $\mathbf{R}$. First, we are then dealing more directly with the objects of interest in computer-aided geometric design. Second, it is easier to draw informative figures when the object space Q has at least two dimensions. An n-ic polynomial curve is degenerate whenever the affine span of its range is a flat of dimension less than n. An n-ic polynomial curve $F\colon L \to \mathbf{R}$, with values in a one-dimensional object space, is so degenerate that it is impossible to draw reasonable pictures of it. Instead, one must resort to alternative strategies such as drawing the *graph* of F, which is the polynomial curve $G\colon L \to \mathbf{R}^2$ given by $G(\bar{u}) := \langle u, F(\bar{u})\rangle$.

Let L be the affine line, and let Q be an arbitrary affine object space. We shall suppose that the line L has been partitioned into intervals by a certain increasing sequence of points $\{\bar{t}_i\}$, which can be finite, infinite, or bi-infinite. For each i, we shall also suppose that we are given a certain polynomial map $F_i\colon L \to Q$ of degree no greater than n. We can assemble an n-ic *spline curve* F out of the given n-ic curves F_i by specifying that $F(\bar{u}) := F_i(\bar{u})$ whenever $t_i < u < t_{i+1}$. If the curves F_{i-1} and F_i happen to agree at the point $\bar{t}_i$, it is natural to specify that the spline F should also share that common value; but if F_{i-1} and F_i do not agree at $\bar{t}_i$, we shall consider $F(\bar{t}_i)$ to be indeterminate. Note that, from a formal point of view, a spline curve is the same thing as a piecewise polynomial curve. The difference is that, when dealing with spline curves, we usually enforce some sort of continuity conditions at the joints between adjacent segments.

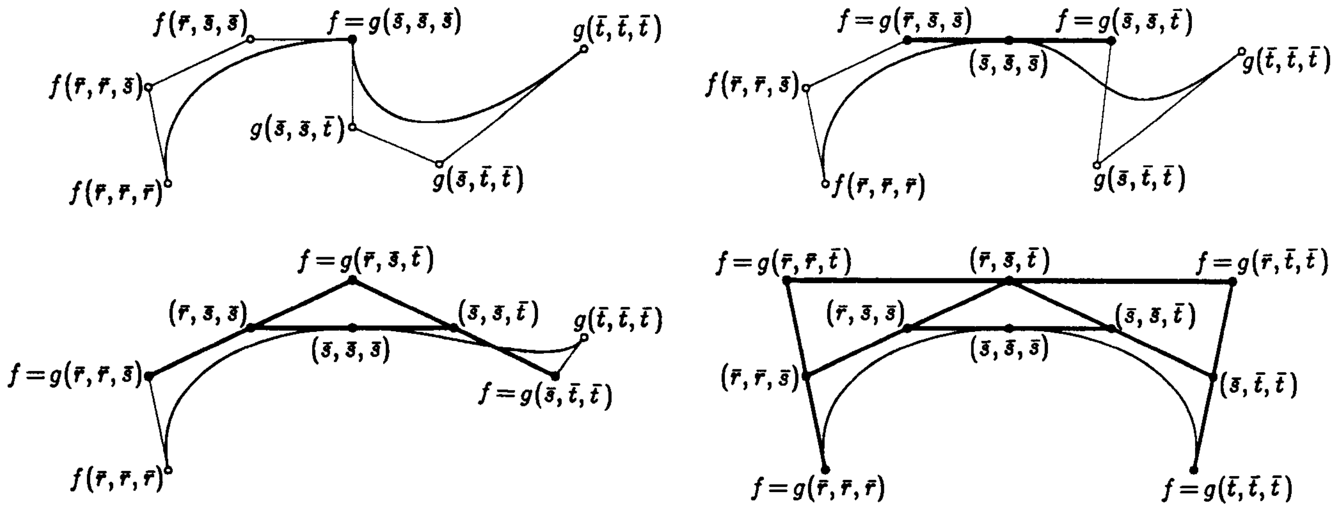
If $F\colon(\bar{r},\bar{s}) \to Q$ and $G\colon(\bar{s},\bar{t}) \to Q$ are polynomial curves defined on adjacent intervals in P, there are two different types of continuity constraints that we could impose on the joint at $\bar{s}$ between F and G. This joint is called *parametrically continuous of order* k, or C^k continuous, if the zeroth through kth derivatives of F and G agree at $\bar{s}$, that is, if the spline curve formed by combining F and G is C^k continuous over the entire interval $(\bar{r},\bar{t})$. Parametric continuity is the simplest notion of continuity, and it is the natural notion to use in situations where the parameterizations of the curves are important as well as their shapes.

In many cases in computer-aided geometric design, however, only the shapes are important. In such cases, we can use a more liberal notion of continuity. The joint at $\bar{s}$ between F and G is called ***geometrically continuous of order k*** or G^k continuous (or ***visually continuous of order k*** or VC^k continuous) if F and G can be reparameterized in a neighborhood of $\bar{s}$ in such a way that the reparameterized curves join with C^k continuity [15]. (One technical detail is that only ***regular*** reparameterizations should be allowed, that is, reparameterizations that are locally invertible.) Allowing reparameterization leads to a more purely geometric notion of continuity. For example, G^1 continuity for a curve means continuity of slope, while C^1 means continuity of velocity; G^2 means continuity of curvature, while C^2 means continuity of acceleration. (Lest the reader be seduced by the apparent pattern, it is important to note that there are more than two choices of continuity conditions for order three and above. Parametric, C^3 continuity is continuity of jerk, that is, continuity of the third vector derivative of position with respect to time. The reparameterization allowed in geometric continuity makes G^3 continuity a weaker condition than C^3. But continuity of torsion is an even weaker condition than G^3, as Böhm points out in a recent paper [4].) Geometric continuity is an interesting area. Unfortunately, as we shall discuss in Section 20, I haven't been able to figure out a good way to handle geometric continuity with the blossoming technology. For now, we shall restrict ourselves to spline curves with parametric continuity constraints.

Parametric continuity and blossoming were made for each other. In particular, Prop. 8.8 tells us that the two n-ic curves $F\colon (\bar{r},\bar{s}) \to Q$ and $G\colon (\bar{s},\bar{t}) \to Q$ join with C^k continuity at $\bar{s}$ if and only if the blossoms f and g agree on all argument bags that include at least $n-k$ copies of $\bar{s}$, that is, if and only if $f^{\otimes}(\bar{u}_1 \cdots \bar{u}_k \bar{s}^{n-k}) = g^{\otimes}(\bar{u}_1 \cdots \bar{u}_k \bar{s}^{n-k})$. Thus, each new order of parametric continuity means that one more blossom argument can be varied without destroying the agreement between the blossoms of the joining curves. If F and G meet with only C^0 continuity, we can't vary any of the arguments away from $\bar{s}$ without destroying agreement; all we know is that $F(\bar{s}) = f^{\otimes}(\bar{s}^n) = g^{\otimes}(\bar{s}^n) = G(\bar{s})$. If F and G meet with C^n continuity, so that they are actually identical, we can vary all n arguments to the blossoms without destroying agreement.

Fig. 14.1 shows examples of joining cubic segments with various orders of continuity. Note that, when the joint between F and G has C^k continuity, we have $f^{\otimes}(\bar{r}^i \bar{t}^j \bar{s}^{n-i-j}) = g^{\otimes}(\bar{r}^i \bar{t}^j \bar{s}^{n-i-j})$ whenever $i + j \le k$. The points given as values of both f and g by these equalities form a de Casteljau Diagram with k shells, which is indicated in Fig. 14.1 in bold. We can interpret this diagram in three different ways: either it extrapolates forward from the last $k+1$ Bézier points of $F([\bar{r},\bar{s}])$ to tell us where the first $k+1$ Bézier points of $G([\bar{s},\bar{t}])$ must be; or, vice versa, it extrapolates backwards from G to F; or, more symmetrically, it constrains both the last $k+1$ Bézier points of $F([\bar{r},\bar{s}])$ and the first $k+1$ Bézier points of $G([\bar{s},\bar{t}])$ so as to guarantee C^k continuity at the joint.

Spline aficianados have a different way of measuring the smoothness of the joint between two curves [8]. Rather than counting the number of derivatives of the joining curves that are guaranteed to agree, they count the number of derivatives, from the nth on down, that are allowed to disagree. In particular, suppose that



$f(\bar{r},\bar{s},\bar{s})$
$f=g(\bar{s},\bar{s},\bar{s})$
$g(\bar{t},\bar{t},\bar{t})$
$f(\bar{r},\bar{r},\bar{s})$
$g(\bar{s},\bar{s},\bar{t})$
$f(\bar{r},\bar{r},\bar{r})$
$g(\bar{s},\bar{t},\bar{t})$
$f=g(\bar{r},\bar{s},\bar{s})$
$f=g(\bar{s},\bar{s},\bar{t})$
$(\bar{s},\bar{s},\bar{s})$
$g(\bar{t},\bar{t},\bar{t})$
$f(\bar{r},\bar{r},\bar{s})$
$f(\bar{r},\bar{r},\bar{r})$
$g(\bar{s},\bar{t},\bar{t})$
$f=g(\bar{r},\bar{s},\bar{t})$
$(\bar{r},\bar{s},\bar{s})$
$(\bar{s},\bar{s},\bar{t})$
$g(\bar{t},\bar{t},\bar{t})$
$f=g(\bar{r},\bar{r},\bar{s})$
$(\bar{s},\bar{s},\bar{s})$
$f=g(\bar{s},\bar{t},\bar{t})$
$f(\bar{r},\bar{r},\bar{r})$
$f=g(\bar{r},\bar{r},\bar{t})$
$(\bar{r},\bar{s},\bar{t})$
$f=g(\bar{r},\bar{t},\bar{t})$
$(\bar{r},\bar{s},\bar{s})$
$(\bar{s},\bar{s},\bar{t})$
$(\bar{s},\bar{s},\bar{s})$
$(\bar{r},\bar{r},\bar{s})$
$(\bar{s},\bar{t},\bar{t})$
$f=g(\bar{r},\bar{r},\bar{r})$
$f=g(\bar{t},\bar{t},\bar{t})$

Fig. 14.1. Two cubic curves joining with C^0, C^1, C^2, and C^3 continuity

the n-ic curves $F\colon(\bar{r},\bar{s}) \to Q$ and $G\colon(\bar{s},\bar{t}) \to Q$ have a C^k joint at the point $\bar{s}$. The parameter value $\bar{s}$ at which the joint occurs is called a ***knot***, and the number $m := n-k$ of derivatives that are allowed to be discontinuous as we pass through the knot is called the ***multiplicity*** of the knot. This convention works out well because a knot of multiplicity m behaves very much like the limit of a cluster of m closely spaced single knots that have coalesced. Rephrasing Prop. 8.8, we observe that the multiplicity of a knot is precisely the number of blossom arguments that must be kept fixed at that knot in order to guarantee that the blossoms of the two joining curves will agree.

These conventions about knot multiplicities allow us to specify all of the boundary conditions of a spline curve by means of one sequence of points $\{\bar{t}_i\}$, called the ***knot sequence***, which consists of all of the knots repeated according to their multiplicities and sorted into non-decreasing order. If the common value $\bar{t}_{i+1} = \bar{t}_{i+2} = \cdots = \bar{t}_{i+m}$ is a knot of multiplicity m, then the curve $F_i\colon(\bar{t}_i,\bar{t}_{i+1}) \to Q$ is required to join the curve $F_{i+m}\colon(\bar{t}_{i+m},\bar{t}_{i+m+1}) \to Q$ at that knot with C^{n-m} continuity. In this way, a knot sequence $\{\bar{t}_i\}$ specifies both the locations and the smoothness levels of the joints allowed in a spline curve.

The knot sequence is just what we need to understand the extent to which the blossoms of two segments of a spline curve are guaranteed to agree. In particular, we can show that the argument bags on which lots of blossoms must agree are the bags that contain substrings of the knot sequence.

Let $\{\bar{t}_i\}$ be the knot sequence of an n-ic spline curve $F(\bar{u})$. For each i such that $\bar{t}_i < \bar{t}_{i+1}$, the behavior of $F(\bar{u})$ for $\bar{u}$ in $(\bar{t}_i,\bar{t}_{i+1})$ determines an n-ic polynomial curve $F_i(\bar{u})$, and that curve has an n-blossom $f_i(\bar{u}_1,\ldots,\bar{u}_n)$. For simplicity at first, let us suppose that there are no multiple knots. Consider the values of the various blossoms when applied to an argument bag of the form $B = \{\bar{t}_{i+1},\ldots,\bar{t}_{i+l},\bar{u}_{l+1},\ldots,\bar{u}_n\}$. Since the bag B includes the simple knot $\bar{t}_{i+1}$, the blossoms f_i and f_{i+1} must agree on B. Since B also includes $\bar{t}_{i+2}$, the blossoms f_{i+1} and f_{i+2} must agree on B. We deduce that, in fact, all of the blossoms from f_i through f_{i+l} must agree on B. Furthermore, this conclusion still holds even if we allow some or all of the l simple knots $\bar{t}_{i+1}$ through $\bar{t}_{i+l}$ to coalesce into knots of higher multiplicity. For example, if $\bar{t}_{i+1} = \bar{t}_{i+2}$ is a double knot, then there is no segment F_{i+1} and no associated blossom f_{i+1}. But the adjacent blossoms f_i and f_{i+2} must still agree on the argument bag B because it includes two copies of the double knot $\bar{t}_{i+1} = \bar{t}_{i+2}$. This sort of reasoning demonstrates the following.

Proposition 14.2. *Let the nonempty interval $(\bar{t}_i, \bar{t}_{i+1})$ be the domain of one segment of a spline curve F with knot sequence $\{\bar{t}_i\}$, and let the nonempty interval $(\bar{t}_j, \bar{t}_{j+1})$ be the domain of a later segment. The blossoms f_i and f_j of these two segments must agree on any argument bag that includes all of the intervening knots as often as their multiplicity, that is, on any bag that includes $\{\bar{t}_{i+1}, \ldots, \bar{t}_j\}$ as a sub-bag. All of the blossoms of intermediate segments, if any, will also agree with f_i and f_j on such argument bags.* □

To take full advantage of this proposition, let us extend our notation by allowing ourselves to write sets of indices as well as single indices as subscripts on f. If we know that all of the blossoms f_i for i in the nonempty set S must return the same value on the particular argument bag $\{\bar{u}_1, \ldots, \bar{u}_n\}$, we shall denote that common value by the term $f_S(\bar{u}_1, \ldots, \bar{u}_n)$. In particular, a common value of f_i and f_j as guaranteed by Prop. 14.2 would be denoted $f_{\{i,j\}}(\bar{t}_{i+1}, \ldots, \bar{t}_j, \bar{u}_{j-i+1}, \ldots, \bar{u}_n)$. The rules for manipulating with these new f_S terms are pretty straightforward. For example, if we know the points $f_S(\bar{x}, \bar{u}_2, \ldots, \bar{u}_n)$ and $f_T(\bar{y}, \bar{u}_2, \ldots, \bar{u}_n)$ and if the sets S and T are not disjoint, we can linearly interpolate (or extrapolate, as necessary) to compute the point $f_{S \cap T}(\bar{z}, \bar{u}_2, \ldots, \bar{u}_n)$ for any $\bar{z}$.

One minor problem with the f_S notation is that our indexing scheme for the segments of a spline is rather complex. The curve segment F_i is the one that the spline curve follows from the knot $\bar{t}_i$ until the knot $\bar{t}_{i+1}$, if these two knots are distinct. If $\bar{t}_i = \bar{t}_{i+1}$, there is no segment F_i, and hence we really shouldn't include i in the set S of an f_S term. To avoid having to worry about which indices are valid, we shall extend our notation once again by allowing ourselves to write f_I, where I is an interval in the domain space L. In particular, we define the term $f_I(\bar{u}_1, \ldots, \bar{u}_n)$ to denote $f_S(\bar{u}_1, \ldots, \bar{u}_n)$ where $S := \{i \mid (\bar{t}_i, \bar{t}_{i+1}) \cap I \neq \emptyset\}$. In order to write down a term of the form $f_I(\bar{u}_1, \ldots, \bar{u}_n)$, we must be able to show that the various f_i for i in this set S will agree on the argument bag $\{\bar{u}_1, \ldots, \bar{u}_n\}$, of course. In a term of the form $f_I(\bar{u}_1, \ldots, \bar{u}_n)$, we shall refer to I as the *validity interval.*

The result of Prop. 14.2 looks particularly simple when written in terms of validity intervals. It tells us that the term $f_{(\bar{t}_i, \bar{t}_{j+1})}(\bar{t}_{i+1}, \ldots, \bar{t}_j, \bar{u}_{j-i+1}, \ldots, \bar{u}_n)$ is well-defined whenever its validity interval is nonempty, that is, whenever $\bar{t}_i < \bar{t}_{j+1}$. This result sets the stage for the multiaffine view of the de Boor theory of spline curves, which we tackle next.

15. The blossoms of spline curves

What are the cases in which Prop. 14.2 is pushed as far as it can go? That is, what are the argument bags on which the widest range of segment blossoms must agree? The extreme case occurs when $j = i + n$, in the terms

$$f_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}).$$

Note that there are no free arguments $\bar{u}_j$ left in this term. What we have is a single argument bag on which $n + 1$ adjacent blossoms are known to agree. Let's call the agreed-upon value a *pole* of the spline F, since, as we will see shortly, it is a pole in the sense of the de Casteljau technique of Section 6 for the $n + 1$ segments of the spline F that do agree. (If there are multiple knots involved, some of the $n + 1$ blossoms that agree on a pole may have degenerated into nonexistence.) An amazingly simple thing happens at this point in the theory of spline curves—be warned that this doesn't work for spline surfaces: the poles form a basis for the space of all spline curves with $\{\bar{t}_i\}$ as their knot sequence. That is, not only can we compute the poles from the spline curve; we can also go backward and compute a unique corresponding spline curve from arbitrarily specified values for the poles.

Proposition 15.1. *Let n be a nonnegative integer and let $\{\bar{t}_i\}$ be a knot sequence in the affine line L that doesn't include any knots of multiplicity greater than $n+1$. Then, spline curves $F\colon L \to Q$ of degree n with the knot sequence $\{\bar{t}_i\}$ are in one-to-one correspondence with sequences of points $\{\mathbf{x}_i\}$ in Q by means of the formula*

$$\mathbf{x}_i = f_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}).$$

Proof. Given a spline curve $F\colon L \to Q$ with the knot sequence $\{\bar{t}_i\}$, Prop. 14.2 shows that the poles $\mathbf{x}_i$ are well-defined, as long as each validity interval $(\bar{t}_i, \bar{t}_{i+n+1})$ is nonempty. This validity interval must be nonempty because otherwise the value $\bar{t}_i$ would be a knot of multiplicity at least $n + 2$.

The reverse direction is harder. Let $\{\mathbf{x}_i\}$ be an arbitrary sequence of points in Q, and let $(\bar{t}_k, \bar{t}_{k+1})$ be a nonempty interval in the knot sequence. Our first task is to construct the curve segment F_k that the spline will follow during the interval $(\bar{t}_k, \bar{t}_{k+1})$. To do so, consider the $2n$ knots centered around that interval, that is, the knots in the sequence $(\bar{t}_{k-n+1}, \ldots, \bar{t}_{k+n})$. We want to build a de Casteljau frame based on this reference sequence. To do so, we must verify the nondegeneracy conditions $\bar{r}_j \neq \bar{r}_{n+i}$ for $1 \leq i \leq j \leq n$, as specified in Lemma 6.1, where $\bar{r}_i = \bar{t}_{k-n+i}$. In this case, that is easy to do: the first n points in the reference sequence are all at most $\bar{t}_k$, while the last n points are all at least $\bar{t}_{k+1}$, and we have $\bar{t}_k < \bar{t}_{k+1}$. Therefore, by the de Casteljau technique of Cor. 6.3, we deduce that we can specify a unique n-ic curve $F_k(\bar{u})$ by arbitrarily specifying its poles, that is, the blossom values $f_k(\bar{t}_{k-n+1}, \ldots, \bar{t}_k)$, $f_k(\bar{t}_{k-n+2}, \ldots, \bar{t}_{k+1})$, $\ldots$, $f_k(\bar{t}_{k+1}, \ldots, \bar{t}_{k+n})$. We choose to specify $F_k(\bar{u})$ by the conditions

$$f_k(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}) := \mathbf{x}_i \qquad \text{for } i \text{ in } [k-n, k]. \tag{15.2}$$

Suppose that we have used this technique to determine $F_k(\bar{u})$ for each k with $\bar{t}_k < \bar{t}_{k+1}$. As a consequence, for any fixed i, it will be the case that $\mathbf{x}_i = f_k(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n})$ for all k in $[i, i+n]$ that satisfy $\bar{t}_k < \bar{t}_{k+1}$. That same collection of indices k can also be described as those for which $(\bar{t}_k, \bar{t}_{k+1}) \cap (\bar{t}_i, \bar{t}_{i+n+1})$ is nonempty. Hence, by the definition of the f_I notation, we will have

$$\mathbf{x}_i = f_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}).$$

There is one thing left to verify: we must show that the resulting polynomial curves F_k really do fit together to form a spline with the appropriate continuity. Suppose that $\bar{t}_{i+1} = \cdots = \bar{t}_{i+m}$ is a knot of multiplicity exactly m, so that $\bar{t}_i < \bar{t}_{i+1}$ and $\bar{t}_{i+m} < \bar{t}_{i+m+1}$. We want to show that the curve F_i joins the curve F_{i+m} at that knot with C^{n-m} continuity. If $m = n+1$, which is the largest multiplicity that the hypotheses allow, we interpret C^{-1} continuity to mean that there is no relation between F_i and F_{i+m}, so there is nothing to prove. Therefore, let us assume that $m \leq n$, and let l denote the level of continuity $l := n - m$. By Prop. 8.8, showing C^l continuity at the joint is equivalent to showing that the blossoms f_i and f_{i+m} agree on all argument bags that include at least m copies of the knot, which, in symbols, is the identity

$$f_i(\bar{t}_{i+1}, \ldots, \bar{t}_{i+m}, \bar{u}_1, \ldots, \bar{u}_l) = f_{i+m}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+m}, \bar{u}_1, \ldots, \bar{u}_l).$$

We know something about the blossoms f_i and f_{i+m} already, from Eq. (15.2). In particular, we know that $f_i(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}) = \mathbf{x}_j$ for j in $[i-n, i]$, and we know that $f_{i+m}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}) = \mathbf{x}_j$ for j in $[i-l, i+m]$. Putting these facts together, we deduce that $f_i(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}) = f_{i+m}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n})$ for j in $[i-l, i]$. More pictorially, we know that f_i and f_{i+m} agree on the rows of the parallelogram

$$\begin{array}{ccccccccccc}
\bar{t}_{i-l+1} & \bar{t}_{i-l+2} & \cdots & \bar{t}_i & \bar{t}_{i+1} & \cdots & \bar{t}_{i+m} & & & & \\
 & \bar{t}_{i-l+2} & \cdots & \bar{t}_i & \bar{t}_{i+1} & \cdots & \bar{t}_{i+m} & \bar{t}_{i+m+1} & & & \\
 & & \ddots & \vdots & \vdots & & \vdots & \vdots & \ddots & & \\
 & & & \bar{t}_i & \bar{t}_{i+1} & \cdots & \bar{t}_{i+m} & \bar{t}_{i+m+1} & \cdots & \bar{t}_{i+m+l-1} & \\
 & & & & \bar{t}_{i+1} & \cdots & \bar{t}_{i+m} & \bar{t}_{i+m+1} & \cdots & \bar{t}_{i+m+l-1} & \bar{t}_{i+m+l}
\end{array}$$

We can finish off the proof by cutting the middle m columns out of this parallelogram, collapsing the two remaining triangles to form a smaller parallelogram, and then applying Cor. 6.3 once again, as follows.

Let g denote the symmetric, multiaffine function of l arguments that results from fixing m of the arguments of f_i at the value of the knot, that is,

$$g(\bar{u}_1, \ldots, \bar{u}_l) := f_i(\bar{t}_{i+1}, \ldots, \bar{t}_{i+m}, \bar{u}_1, \ldots, \bar{u}_l),$$

and let h be the corresponding restriction of f_{i+m}. The fact that f_i and f_{i+m} agree on the rows of the parallelogram above implies that g and h have the same poles with respect to the reference sequence $\langle \bar{t}_{i-l+1}, \ldots, \bar{t}_i, \bar{t}_{i+m+1}, \ldots, \bar{t}_{i+m+l} \rangle$, which consists of the l knots that precede $\bar{t}_{i+1}$ concatenated with the l knots that follow $\bar{t}_{i+m}$. Furthermore, this reference sequence satisfies the nondegeneracy condition of

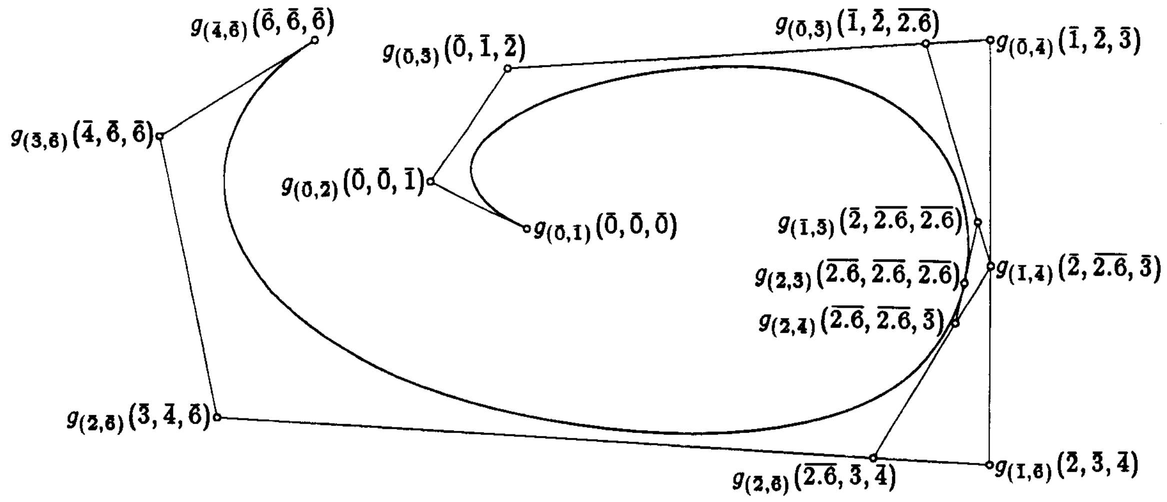

$g_{(\bar4,\bar6)}(\bar6,\bar6,\bar6)$
$g_{(\bar0,\bar3)}(\bar0,\bar1,\bar2)$
$g_{(\bar0,\bar3)}(\bar1,\bar2,\overline{2.6})$
$g_{(\bar0,\bar4)}(\bar1,\bar2,\bar3)$
$g_{(\bar3,\bar6)}(\bar4,\bar6,\bar6)$
$g_{(\bar0,\bar2)}(\bar0,\bar0,\bar1)$
$g_{(\bar0,\bar1)}(\bar0,\bar0,\bar0)$
$g_{(\bar1,\bar3)}(\bar2,\overline{2.6},\overline{2.6})$
$g_{(\bar2,\bar3)}(\overline{2.6},\overline{2.6},\overline{2.6})$
$g_{(\bar1,\bar4)}(\bar2,\overline{2.6},\bar3)$
$g_{(\bar2,\bar4)}(\overline{2.6},\overline{2.6},\bar3)$
$g_{(\bar2,\bar6)}(\bar3,\bar4,\bar6)$
$g_{(\bar2,\bar6)}(\overline{2.6},\bar3,\bar4)$
$g_{(\bar1,\bar6)}(\bar2,\bar3,\bar4)$

Fig. 15.4. A cubic spline curve G with knot sequence $(\bar0,\bar0,\bar0,\bar0,\bar1,\bar2,\bar3,\bar4,\bar6,\bar6,\bar6,\bar6)$

Lemma 6.1 because each left-half element is at most $\bar{t}_i$, each right-half element is at least $\bar{t}_{i+m+1}$, and we have $\bar{t}_i < \bar{t}_{i+1} = \bar{t}_{i+m} < \bar{t}_{i+m+1}$. Thus, we may conclude by Cor. 6.3 that g and h are identical. This implies that the blossoms f_i and f_{i+m} agree on all argument bags that include m copies of the knot $\bar{t}_{i+1} = \bar{t}_{i+m}$. By Prop. 8.8, we conclude that F_i and F_{i+m} do join with C^l continuity at that knot. □

Our next task is to demonstrate that the spline-controlling technique based on poles in Prop. 15.1 is just the standard de Boor technique in a new guise. One warning: many authors use the variable n to denote the *order* of a spline curve, while we have been using (and will continue to use) n to denote the bound on the degree instead. For example, a cubic spline has degree 3 (or less), but has order 4. For our purposes, the degree is a more important concept than the order, since the degree counts the number of arguments of the blossom.

Proposition 15.3. *The poles* $\mathbf{x}_i = f_{(\bar{t}_i,\bar{t}_{i+n+1})}(\bar{t}_{i+1},\ldots,\bar{t}_{i+n})$ *of an n-ic spline curve F are precisely the de Boor points by which that spline curve is controlled in the standard theory. Furthermore, the validity interval* $(\bar{t}_i,\bar{t}_{i+n+1})$ *of the pole* $\mathbf{x}_i$ *is precisely the region of the parameter space L over which* $\mathbf{x}_i$ *influences the value of the spline curve; that is, the pole* $\mathbf{x}_i$ *influences the spline segments* $F_k(\bar{u})$ *precisely for k in* $[i, i+n]$.

Proof. We shall prove this by writing down the multiaffine labels for all of the points that occur in the de Boor Algorithm. If you are reading for concepts rather than details, you might want to skip this proof. But you should take a moment to study the labels in Fig. 15.4, which shows an example of the de Boor Algorithm computing the point $G(\overline{2.6})$ on a cubic spline curve G, whose knot sequence is $(\bar0,\bar0,\bar0,\bar0,\bar1,\bar2,\bar3,\bar4,\bar6,\bar6,\bar6,\bar6)$.

The de Boor Algorithm [11] defines a spline $F(\bar{u})$ of degree n on the knot sequence $\{\bar{t}_i\}$ given a sequence of de Boor points $\{\mathbf{y}_i\}$ by setting $F(\bar{u}) := \alpha_k^{[n+1]}(\bar{u})$

where k is chosen so that $\bar{t}_k < \bar{u} < \bar{t}_{k+1}$ and the functions $\alpha_i^{[j+1]}(\bar{u})$ are defined by

$$\alpha_i^{[j+1]}(\bar{u}) := \begin{cases} \mathbf{y}_i, & \text{if } j = 0 \\ \dfrac{(t_{i+n-j+1} - u)\alpha_{i-1}^{[j]}(\bar{u}) + (u - t_i)\alpha_i^{[j]}(\bar{u})}{t_{i+n-j+1} - t_i}, & \text{if } j > 0. \end{cases}$$

The correspondence with the multiaffine point of view is given by the identity

$$\alpha_i^{[j+1]}(\bar{u}) = f^{\otimes}_{(\bar{t}_i, \bar{t}_{i+n-j+1})}(\bar{t}_{i+1} \cdots \bar{t}_{i+n-j}\bar{u}^j).$$

For $j = 0$, de Boor points $\mathbf{y}_i = \alpha_i^{[1]}(\bar{u})$ themselves are the same as the poles $\mathbf{x}_i = f^{\otimes}_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1} \cdots \bar{t}_{i+n})$.

For $j > 0$, the formula that computes $\alpha_i^{[j+1]}(\bar{u})$ by interpolating between $\alpha_i^{[j]}(\bar{u})$ and $\alpha_{i-1}^{[j]}(\bar{u})$ corresponds to computing

$$f^{\otimes}_{(\bar{t}_i, \bar{t}_{i+n-j+1})}(\bar{t}_{i+1} \cdots \bar{t}_{i+n-j}\bar{u}^j)$$

by interpolating between

$$f^{\otimes}_{(\bar{t}_i, \bar{t}_{i+n-j+2})}(\bar{t}_{i+1} \cdots \bar{t}_{i+n-j}\bar{t}_{i+n-j+1}\bar{u}^{j-1})$$

and

$$f^{\otimes}_{(\bar{t}_{i-1}, \bar{t}_{i+n-j+1})}(\bar{t}_i\bar{t}_{i+1} \cdots \bar{t}_{i+n-j}\bar{u}^{j-1}).$$

The latter two points share all but one argument; the varying argument is $\bar{t}_{i+n-j+1}$ in the former case and $\bar{t}_i$ in the latter case. We can turn this varying argument into a jth copy of $\bar{u}$ by interpolating with the ratio $\bar{t}_i : \bar{u} : \bar{t}_{i+n-j+1}$, just as the $j > 0$ case of the $\alpha_i^{[j+1]}(\bar{u})$ recurrence tells us to do. The result of this interpolation receives, as validity interval, the intersection of the validity intervals of the two input points, just as it should.

For $j = n$, we get $\alpha_i^{[n+1]}(\bar{u}) = f_{(\bar{t}_i, \bar{t}_{i+1})}(\bar{u}, \dots, \bar{u}) = F_i(\bar{u})$ for all i. If we then choose k so that $\bar{u}$ lies in $(\bar{t}_k, \bar{t}_{k+1})$, we have $\alpha_k^{[n+1]}(\bar{u}) = F_k(\bar{u}) = F(\bar{u})$.

To verify the claim about the validity interval of the pole $\mathbf{x}_i$, we can prove by induction on j that the point $\mathbf{x}_i = f_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1}, \dots, \bar{t}_{i+n})$ affects the values $\alpha_k^{[j+1]}(\bar{u})$ precisely for k in $[i, i + j]$. In particular, $\mathbf{x}_i$ affects $F_k(\bar{u}) = \alpha_k^{[n+1]}(\bar{u})$ precisely for k in $[i, i + n]$. This means that the spline $F(\bar{u})$ is affected precisely for $\bar{u}$ in the validity interval $(\bar{t}_i, \bar{t}_{i+n+1})$. □

Project 15.5. Pick a computer scientist at your table who doesn't know anything about splines. Explain the de Boor technique for specifying polynomial spline curves to that lucky person, drawing all your figures on one paper napkin. Don't limit your presentation to cubics, and don't assume uniform knot spacing.

16. Overloading the notation for a spline blossom

When working with the blossom f of a spline curve F, it is irritating to have to keep writing the validity intervals as subscripts on f all the time. Indeed, the labels in Figs. 1.3 and 1.4 were spline blossom values without validity intervals. The way to avoid writing the subscripts is to invent a scheme for deducing the proper subscript from the points in the bag of arguments. Note that the notation for the spline curve F itself involves just such a deduction scheme. To interpret the term $F(\bar{u})$, we first deduce the proper i by determining which domain interval $(\bar{t}_i, \bar{t}_{i+1})$ contains the argument $\bar{u}$; then we evaluate $F_i(\bar{u})$. We shall refer to the adoption of a scheme for deducing the proper subscript as *overloading*. The non-overloaded notation $F_i(\bar{u})$ is more powerful, since it allows us to evaluate any segment of the spline at any point, even points where $\bar{u} < \bar{t}_i$ or $\bar{u} > \bar{t}_{i+1}$. But the overloaded notation $F(\bar{u})$ is more convenient. Achieving the same convenience by overloading the notation for the spline blossom f is rather delicate, because there are n arguments instead of just one.

Starting at the beginning, note that we can't possibly make overloading work when $n = 0$, since the blossom f doesn't have any arguments to examine. Indeed, in the case $n = 0$, the validity intervals convey just the right information. A de Boor point of a zeroth degree spline curve (a "constant spline") $F(\bar{u})$ has a label of the form $f_{(\bar{t}_i, \bar{t}_{i+1})}()$, and we have $F(\bar{u}) = f_{(\bar{t}_i, \bar{t}_{i+1})}()$ whenever $\bar{u}$ is in $(\bar{t}_i, \bar{t}_{i+1})$.

When $n = 1$, each affine segment $F_i(\bar{u})$ of the spline is equal to its blossom $f_i(\bar{u})$, so we can overload the spline blossom in the same way that we overload the spline itself. The only problems arise at knots of multiplicity at least 2.

When $n \geq 2$, there are two reasonable conventions that one could adopt, which we shall refer to as *tame* and *wild* overloading. In each case, the heart of the convention is a rule for deciding, given a bag of arguments $\{\bar{u}_j\}$, which parameter intervals $(\bar{t}_i, \bar{t}_{i+1})$ should be *candidates* for use when assigning a value to the blossom expression $f(\bar{u}_1, \ldots, \bar{u}_n)$. Given a rule for candidacy, a blossom expression $f(\bar{u}_1, \ldots, \bar{u}_n)$ is well-defined if and only if there is at least one candidate interval and all of the values $f_{(\bar{t}_i, \bar{t}_{i+1})}(\bar{u}_1, \ldots, \bar{u}_n)$ for candidate intervals $(\bar{t}_i, \bar{t}_{i+1})$ agree.

In *tame overloading*, the interval $(\bar{t}_i, \bar{t}_{i+1})$ is a candidate for use when defining $f(\bar{u}_1, \ldots, \bar{u}_n)$ whenever the closed intervals $[\bar{t}_i, \bar{t}_{i+1}]$ and $[\min\{\bar{u}_j\}, \max\{\bar{u}_j\}]$ intersect. For example, if the knot sequence $\{\bar{t}_i\}$ is the integers $t_i := i$, there are three candidate intervals for the term $f(\overline{0.5}, \bar{2})$ under tame overloading: $(\bar{0}, \bar{1})$, $(\bar{1}, \bar{2})$, and $(\bar{2}, \bar{3})$. Appealing to Prop. 14.2, we can deduce that the expression $f(\bar{u}_1, \ldots, \bar{u}_n)$ is well-defined under tame overloading if and only if every knot in the closed interval $[\min\{\bar{u}_j\}, \max\{\bar{u}_j\}]$ is included in the bag $\{\bar{u}_j\}$ at least as often as it is a knot. In the example above, the term $f(\overline{0.5}, \bar{2})$ is not well-defined, because the simple knot $\bar{1}$ doesn't appear as an argument; but the term $f(\overline{0.5}, \bar{1}, \bar{2})$ would be well-defined.

In *wild overloading*, there are two cases. If the blossom arguments $\bar{u}_j$ are not all equal, then the interval $(\bar{t}_i, \bar{t}_{i+1})$ is a candidate for defining the term $f(\bar{u}_1, \ldots, \bar{u}_n)$ only when the open intervals $(\bar{t}_i, \bar{t}_{i+1})$ and $(\min\{\bar{u}_j\}, \max\{\bar{u}_j\})$ intersect. If all of the $\bar{u}_j$ are equal, the interval $(\bar{t}_i, \bar{t}_{i+1})$ is a candidate whenever the common value of the $\bar{u}_j$'s lies in $[\bar{t}_i, \bar{t}_{i+1}]$. With the integers as the knots once again, only the two intervals $(\bar{0}, \bar{1})$ and $(\bar{1}, \bar{2})$ are candidates for use when assigning a value to the term

$f(\overline{0.5}, \bar{2})$ under wild overloading. Both of these intervals would also be candidates in the case of $f(\bar{1}, \bar{1})$, but the interval $(\bar{1}, \bar{2})$ would be the only candidate for $f(\bar{1}, \bar{2})$. Appealing once again to Prop. 14.2, we deduce that the expression $f(\bar{u}_1, \ldots, \bar{u}_n)$ is well-defined under wild overloading if and only if either $\min\{\bar{u}_j\} < \max\{\bar{u}_j\}$ and every knot in the interval $(\min\{\bar{u}_j\}, \max\{\bar{u}_j\})$ is included among the $\bar{u}_j$ at least as often as it is a knot; or $\min\{\bar{u}_j\} = \max\{\bar{u}_j\}$ and the multiplicity of that common value as a knot does not exceed n. For example, if we adopt the knot sequence $\{\bar{t}_i\}$ where $t_i := \lfloor i/2 \rfloor$, in which each integer is a double knot, then $f(\overline{0.5}, \bar{1}, \bar{1}, \bar{2})$ is well-defined under wild overloading, even though it would not be well-defined under tame overloading.

The power of wild overloading is quite helpful, as we can see by comparing the labels in Fig. 1.4 to those in Fig. 15.4 (remembering that the labels in Fig. 1.4 are missing the bars over their arguments). Note that the label $g(\bar{0}, \bar{1}, \bar{1})$ in Fig. 1.4 is not well-defined under tame overloading, since $\bar{0}$ appears only once as an argument, while $\bar{0}$ appears four times as a knot. Indeed, the presence of one $\bar{0}$ among the arguments of g tells us that $g(\bar{0}, \bar{1}, \bar{1})$ should lie in the osculating plane $\mathrm{Osc}_2\, G(\bar{0})$. But $G(\bar{u})$ is not even C^0 at $u = 0$; hence, it has two different osculating planes there, one associated with small positive u and the other with small negative u. The wild overloading convention makes $g(\bar{0}, \bar{1}, \bar{1})$ well-defined by choosing the positive plane, on the grounds that the other argument values are all nonnegative and at least one of them is positive. More generally, the term $f^{\otimes}(\bar{t}_i^{\,n-j} \bar{t}_{i+1}^{\,j})$ is a well-defined reference to the jth Bézier point of the n-ic segment $F([\bar{t}_i, \bar{t}_{i+1}])$ in almost every case, under wild overloading. The lone exception is the term $f^{\otimes}(\bar{t}_i^{\,n})$ when $\bar{t}_i$ is a knot of multiplicity exceeding n. The label $g(\bar{0}, \bar{0}, \bar{0})$ in Fig. 1.4 is an example of this exception; it isn't well-defined even under wild overloading, just as $G(\bar{0})$ isn't well-defined.

But the power of wild overloading can lead to confusion in some cases. Consider a quadratic spline curve $F(\bar{u})$ with knot sequence $(\ldots, \bar{0}, \bar{1}, \bar{2}, \bar{2}, \bar{3}, \bar{4}, \ldots)$. The term $f(\bar{1}, \bar{2})$ is well-defined under wild overloading, with the value $f_{(\bar{0},\bar{2})}(\bar{1}, \bar{2})$. Similarly, the term $f(\bar{2}, \bar{3})$ is well-defined, with the value $f_{(\bar{2},\bar{4})}(\bar{2}, \bar{3})$. Furthermore, the terms $f(\bar{1}, \bar{2})$ and $f(\bar{2}, \bar{3})$ have all but one argument in common. We might be tempted to compute $f(\bar{x}, \bar{2})$ for $1 < x < 3$ by interpolating between $f(\bar{1}, \bar{2})$ and $(\bar{2}, \bar{3})$. But the validity intervals of $f(\bar{1}, \bar{2})$ and $f(\bar{2}, \bar{3})$ are disjoint; hence we have no right to do so. Such apparent failures of multiaffineness can't happen under tame overloading. In particular, suppose that $f(\bar{x}, \bar{u}_2, \ldots, \bar{u}_n)$ and $f(\bar{y}, \bar{u}_2, \ldots, \bar{u}_n)$ are both well-defined under tame overloading. Then, for any z between x and y, the term $f(\bar{z}, \bar{u}_2, \ldots, \bar{u}_n)$ will also be well-defined under tame overloading, and the latter point can be found by interpolating between the two former points.

17. The trellis of a spline curve

In this section, we shall pause to reap the rewards of our labors by looking at some pictures of spline curves. Let the latter part of the Roman alphabet from $\bar{p}$ through $\bar{z}$ denote a portion of the knot sequence of a parametrically continuous spline curve F of degree n. The figures below will be drawn as if the knots were equally spaced, since that makes it easier to detect visual patterns; but we will refer

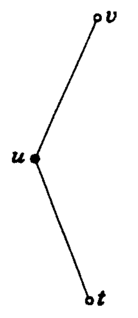


Fig. 17.1. An affine trellis

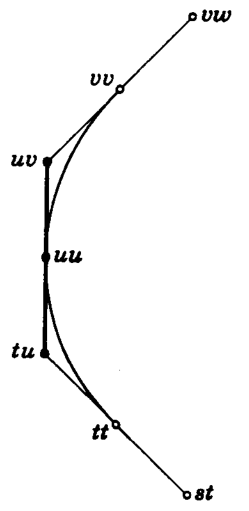


Fig. 17.2. A quadratic trellis

to the knots as $\bar{u}$, $\bar{v}$, and the like, rather than $\bar{2}$, $\bar{3}$, and the like, in order to emphasize that arbitrary knot spacings are possible.

For the first batch of figures, suppose that we are particularly interested in the behavior of the curve F over the two adjacent intervals $(\bar{t}, \bar{u})$ and $(\bar{u}, \bar{v})$. Figs. 17.1 through 17.6 depict the neighborhood of those two segments of F for degrees $n = 1$ through $n = 6$. In each case, the points are labeled with a string of letters, with the special-purpose convention that, for example, "$tuvv$" means $f_{(\bar{s},\bar{w})}(\bar{t}, \bar{u}, \bar{v}, \bar{v})$. That is, we leave off the symbol f, the parentheses, the commas, and the bars, and we deduce the validity intervals by the use of an overloading convention—the tame convention will suffice if we assume that there are no multiple knots.

In each case, the points whose labels consist entirely of t's and u's are the Bézier points of the segment $F_{(\bar{t},\bar{u})}([\bar{t}, \bar{u}])$, and those whose labels consist entirely of u's and v's are the Bézier points of $F_{(\bar{u},\bar{v})}([\bar{u}, \bar{v}])$. If u is a simple knot, as is the case in the figures, we have C^{n-1} continuity at $\bar{u}$. The bold lines in each diagram show the de Casteljau Diagram of order $n - 1$ that geometrically demonstrates this level of continuity. Note that the points of this de Casteljau Diagram are precisely those whose labels are drawn from the set $\{t, u, v\}$. Each of them contains at least one u, reflecting the fact that this diagram lies in the osculating flat $\mathrm{Osc}_{n-1} F(\bar{u})$.

The diagram in Fig. 17.3 for a cubic spline is well-known, although not with these labels. It is important to realize that this type of diagram is associated with every cubic spline curve, whether we used the de Boor theory to draw that spline or not. For example, consider the classical case of C^2 interpolating cubic spline curves. In this scheme, the designer chooses the locations of the joints $f(\bar{u}, \bar{u}, \bar{u})$, $f(\bar{v}, \bar{v}, \bar{v})$, and the like, while the knots $\bar{u}$, $\bar{v}$, and the like are chosen by some other rule—perhaps equally spaced or with spacing proportional to the chord length between the corresponding joints. Given this data, we must solve a system of linear equations over the entire spline curve in order to compute the geometry shown in Fig. 17.3.

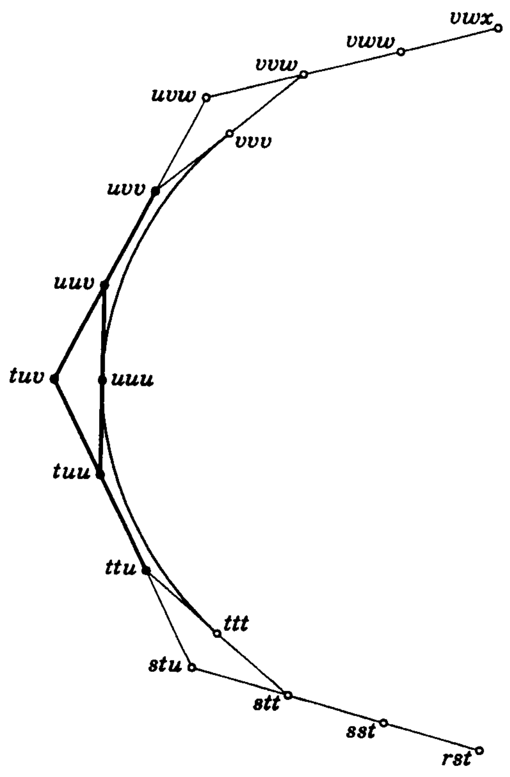


Fig. 17.3. A cubic trellis

(We must also choose some end conditions, in order to give us as many equations as unknowns in this linear system.) In a scheme based on cubic B-splines, the designer chooses the de Boor points $f(\bar{s}, \bar{t}, \bar{u})$, $f(\bar{t}, \bar{u}, \bar{v})$, and the like, and the rest of the geometry shown in Fig. 17.3 is computed locally from them. But the same geometry exists in both cases. The only difference between the two schemes is which points and parameters are chosen as the input from which everything else must be calculated.

Two interesting new phenomena arise in the quartic case, shown in Fig. 17.4. First, this is lowest degree for which the de Boor points of the curve are not part of the de Casteljau Diagrams that enforce continuity at the joints. As the degree goes up, the de Casteljau Diagrams move in more and more, getting further and further away from the de Boor points and the control polygon that connects them. The second interesting thing that happens when $n = 4$ is exemplified by the point *uuvv*: it is the first point we have seen that lies at an intersection that looks like an X rather than a Y. This means that there are two different ways to compute the point *uuvv* by linear interpolation from other points: we can interpolate between *tuuv* and *uuvw* or between *tuvv* and *uvvw*. It is not immediately obvious, geometrically, why these two interpolations always give the same answer; but the blossoming technology makes it clear that they do.

There wasn't enough space to label all of the points in the diagram for $n = 5$ in

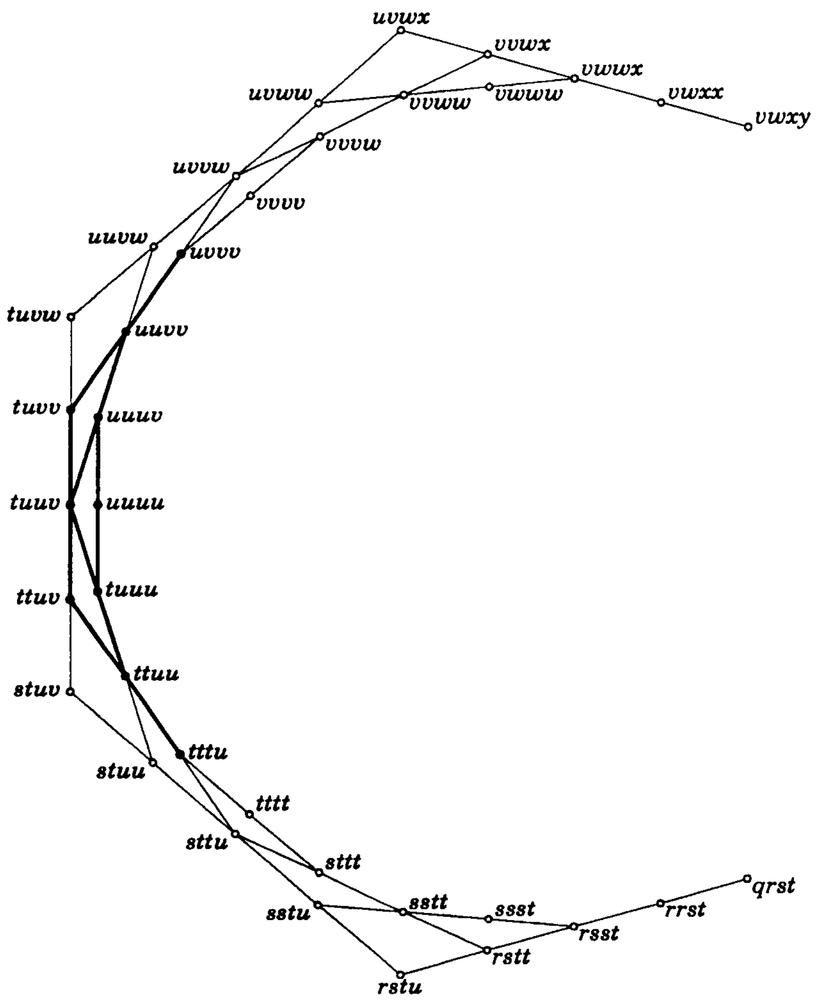


Fig. 17.4. A quartic trellis

Fig. 17.5. The new phenomenon that arises here is pairs of line segments that cross in the digram without a dot at the crossing: for example, a little above the middle, the pair of lines ***tuvv**** and ***tuuv****. Such crossings only arise because our figures are degenerate, by virtue of being drawn in a plane. If the de Boor points in Fig. 17.5 were in general position, such pairs of lines would be skew.

Fig. 17.6 shows the case $n = 6$. The pictures become harder to interpret around $n = 6$ because of the number of near-coincidences that start to arise. For example, just above the Bézier point ***uuuvvv*** are two points quite close together. Of that pair, the one on the de Casteljau Diagram is ***tuvvvv***, while the other one is ***uuuuvw***. The major new structural phenomenon that occurs for $n = 6$ takes place at the midpoint of the outermost shell of the bold de Casteljau diagram, at the point ***ttuuvv***. This point can be computed by interpolating between other points in three different ways, corresponding to the three line segments in the interior of which it lies: [***sttuuv***, ***ttuuvw***], [***sttuvv***, ***ttuvvw***], and [***stuuvv***, ***tuuvvw***].

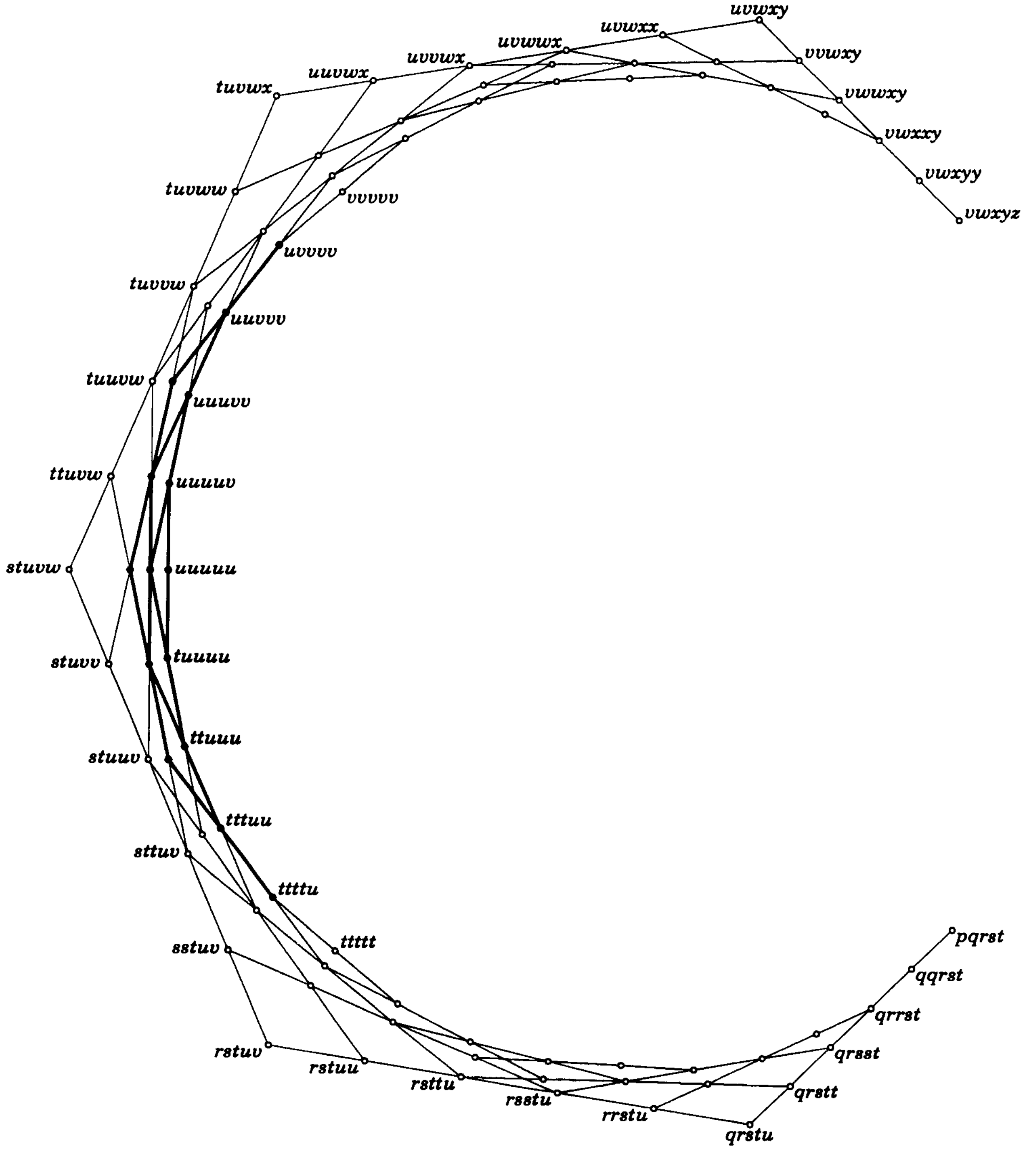


Fig. 17.5. A quintic trellis

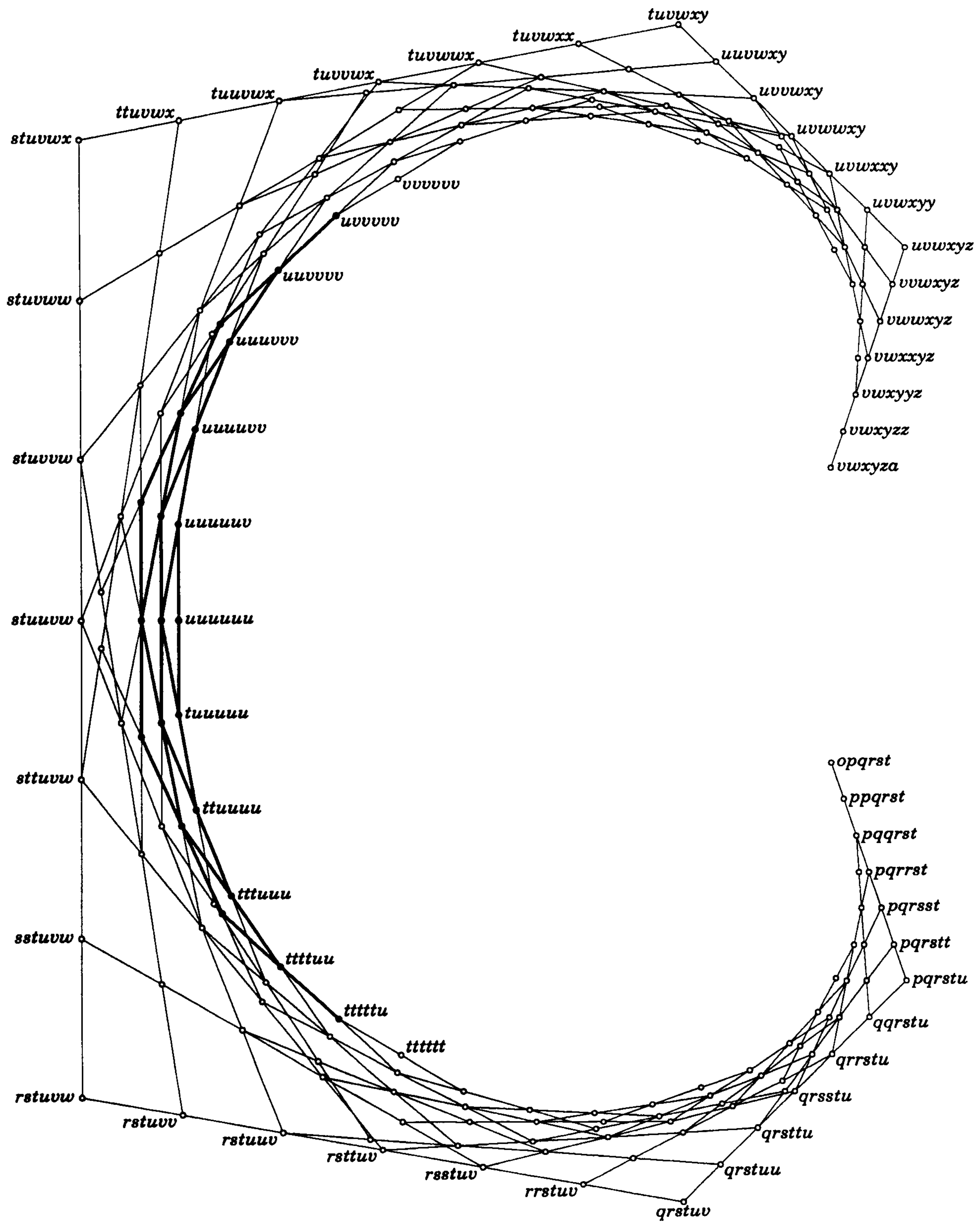


Fig. 17.6. A sextic trellis

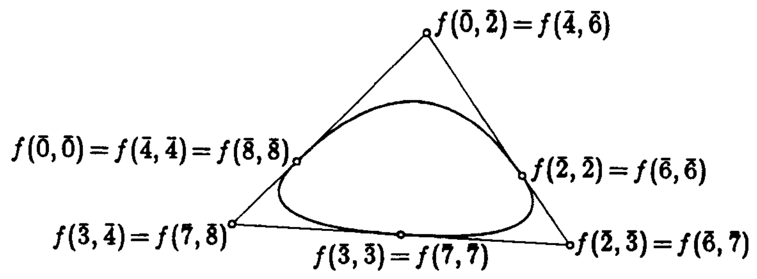

$f(\bar{0},\bar{2})=f(\bar{4},\bar{6})$
$f(\bar{0},\bar{0})=f(\bar{4},\bar{4})=f(\bar{8},\bar{8})$
$f(\bar{2},\bar{2})=f(\bar{6},\bar{6})$
$f(\bar{3},\bar{4})=f(\bar{7},\bar{8})$
$f(\bar{3},\bar{3})=f(\bar{7},\bar{7})$
$f(\bar{2},\bar{3})=f(\bar{6},\bar{7})$

Fig. 17.7. A quadratic spline curve that is additively periodic

As these figures have shown, every spline curve has a framework of line segments surrounding it and supporting it. Since this framework of sticks is covered with blossoms, it is overwhelmingly tempting to refer to it as the *trellis* of the spline curve. More technically, the vertices of the trellis are those blossom values $f(\bar{u}_1, \ldots, \bar{u}_n)$ that are well-defined under wild overloading and all of whose arguments $\bar{u}_i$ are knots. The line segments of the trellis consist of all blossom values that are well-defined under wild overloading and all but one of whose arguments are knots. The algebraic properties of the blossom provide a helpful guide to the geometric intricacies of the trellis.*

One difficulty in drawing the trellis of a spline curve is knowing when to stop. In Figs. 17.1 through 17.6, we drew all of the lines involved in determining the behavior of the spline over the two adjacent segments $F([\bar{t}, \bar{u}])$ and $F([\bar{u}, \bar{v}])$. We can avoid the problem of knowing when to stop by considering a periodic spline, where we can draw everything with a finite amount of work. There are two types of periodicity that a spline can possess: additive and multiplicative. In the additive case, we have $F(\overline{u+C}) = F(\bar{u})$ and $\bar{t}_{i+M} = \bar{t}_i + C$ for some positive real constant C and positive integer M. Fig. 17.7 shows an example of a additively periodic quadratic spline in which $C = 4$, $M = 3$, and the knot sequence has the form $(\ldots, \bar{0}, \bar{2}, \bar{3}, \bar{4}, \bar{6}, \bar{7}, \bar{8}, \ldots)$. Each de Casteljau Diagram around the cycle of a periodic spline has an associated ratio. In Fig. 17.7, those ratios are 1, 2, and 1/2. In order for additive periodicity to arise, the product of these ratios around the cycle must be precisely 1. There is no reason why this product should take any particular value, however. If the product is not 1, the result is a spline that is multiplicatively periodic: we have $F(\overline{Cu}) = F(\bar{u})$ and $\bar{t}_{i+M} = C\bar{t}_i$ for a positive real C and a positive integer M. Fig. 17.8 shows an example of a multiplicatively periodic quadratic spline in which $C = 8$, $M = 3$, and the knot sequence is given by $t_i = 2^i$.

Whatever flavor of periodicity a spline possesses, the periodicity allows us to draw the entire trellis of the spline in a finite diagram. Figs. 17.9 through 17.13 are examples constructed as follows: I copied a closed control polygon with 12 vertices from a paper by Robin Forrest [23]. I then constructed the additively periodic spline

* Without the blossom as a guide, one might resort to drawing less than the full trellis; for example, Fig. 5 in Sablonnière's paper [37] is a subset of the quintic trellis shown in our Fig. 17.5.

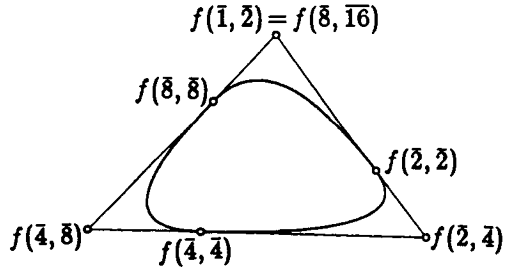


Fig. 17.8 A quadratic spline curve that is multiplicatively periodic

curves of degrees 2 through 6 with equally spaced knots determined by those 12 de Boor points. It isn't really relevant to this paper, but Fig. 17.14 shows what happens as the degree gets higher; it shows the splines of degree 2^i based on the same control polygon for i in $[0,8]$. All of the curves for degrees 512 and up are located within the small dot at the centroid of the 12 control vertices. Finite Fourier transforms are a good tool for studying the process of convergence to the centroid.

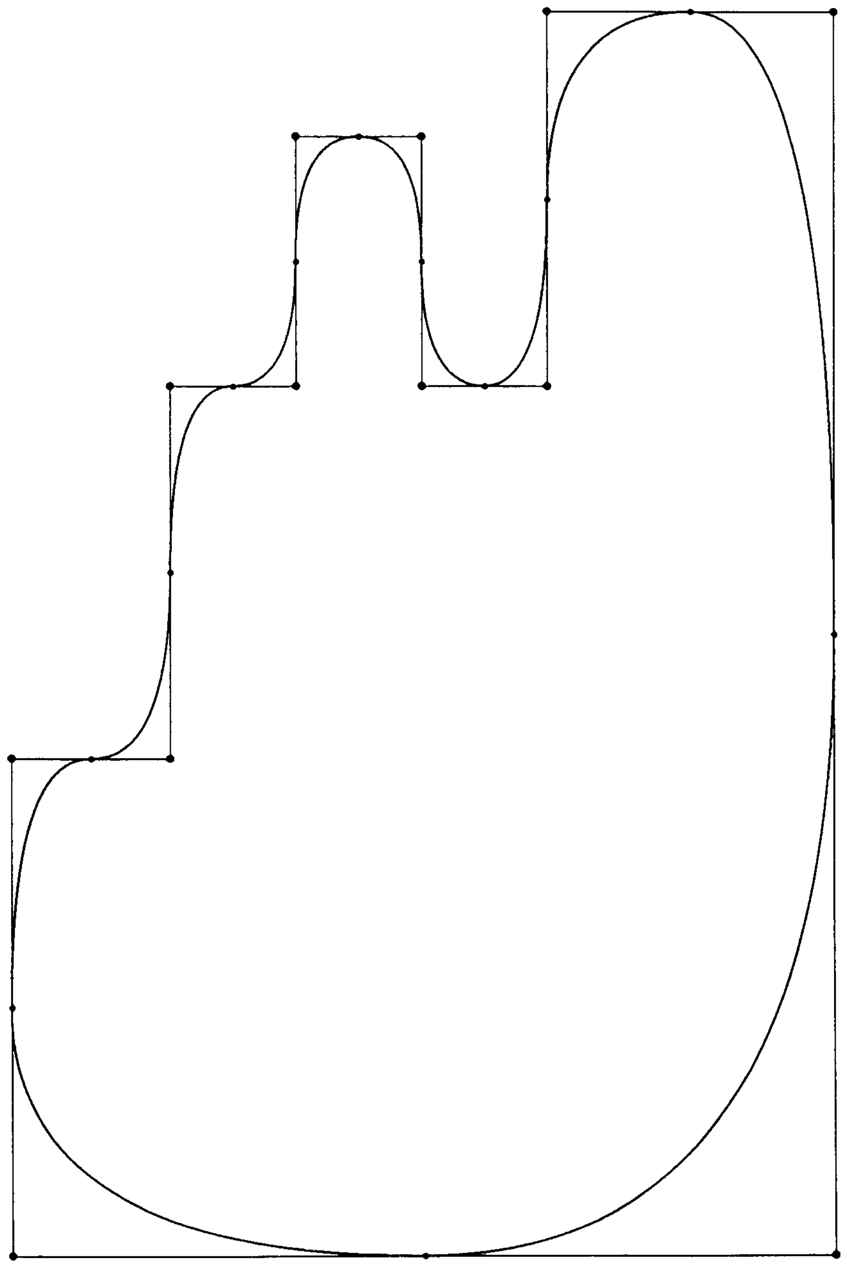


Fig. 17.9. The trellis of a periodic quadratic spline curve

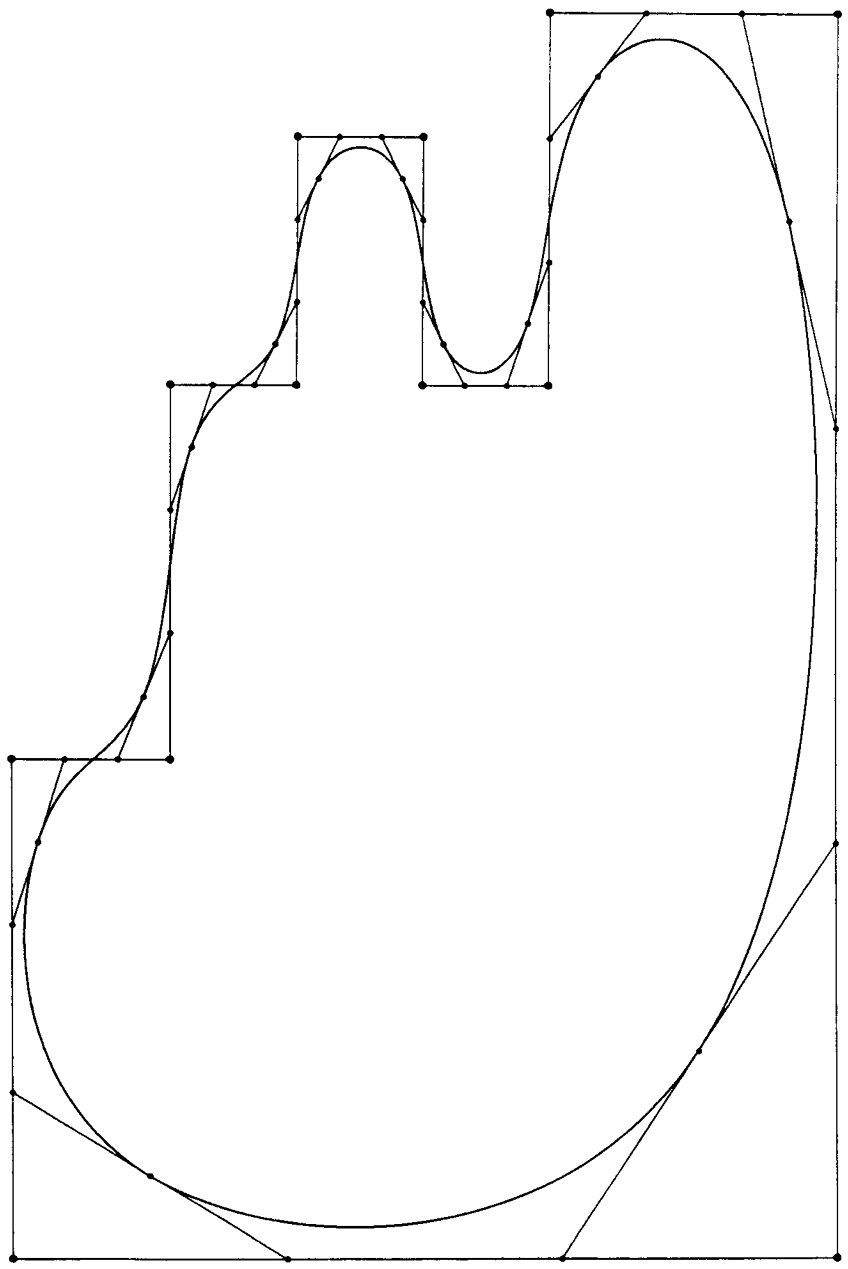


Fig. 17.10. The trellis of a periodic cubic spline curve

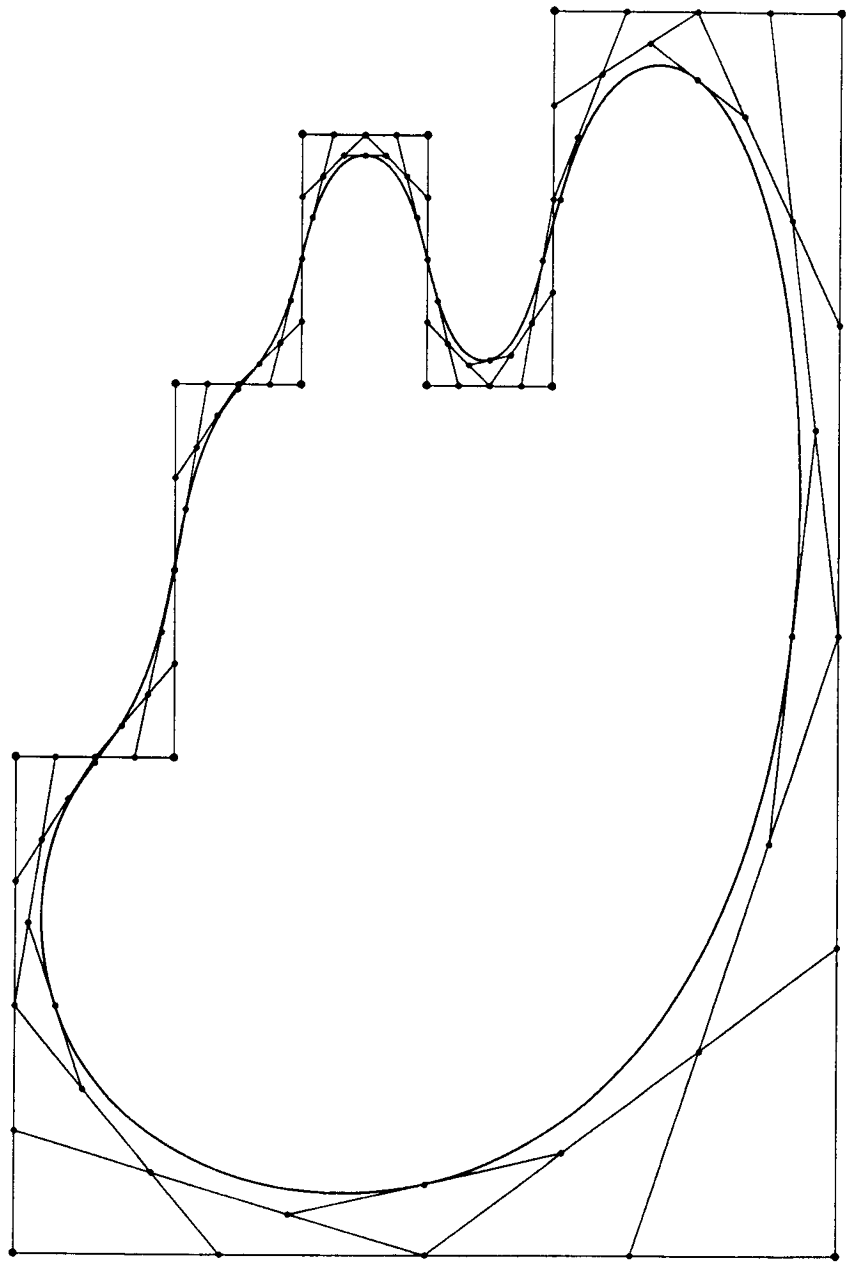


Fig. 17.11. The trellis of a periodic quartic spline curve

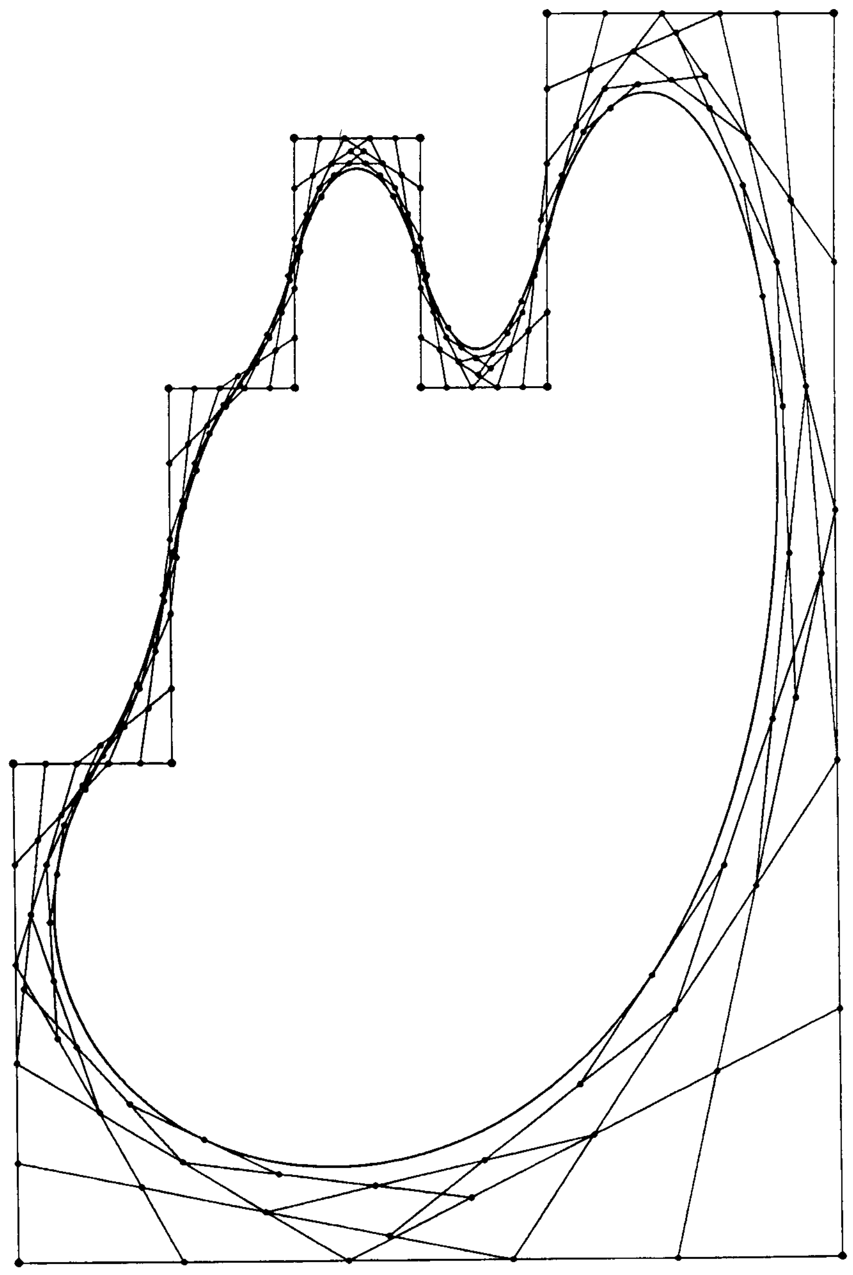


Fig. 17.12. The trellis of a periodic quintic spline curve

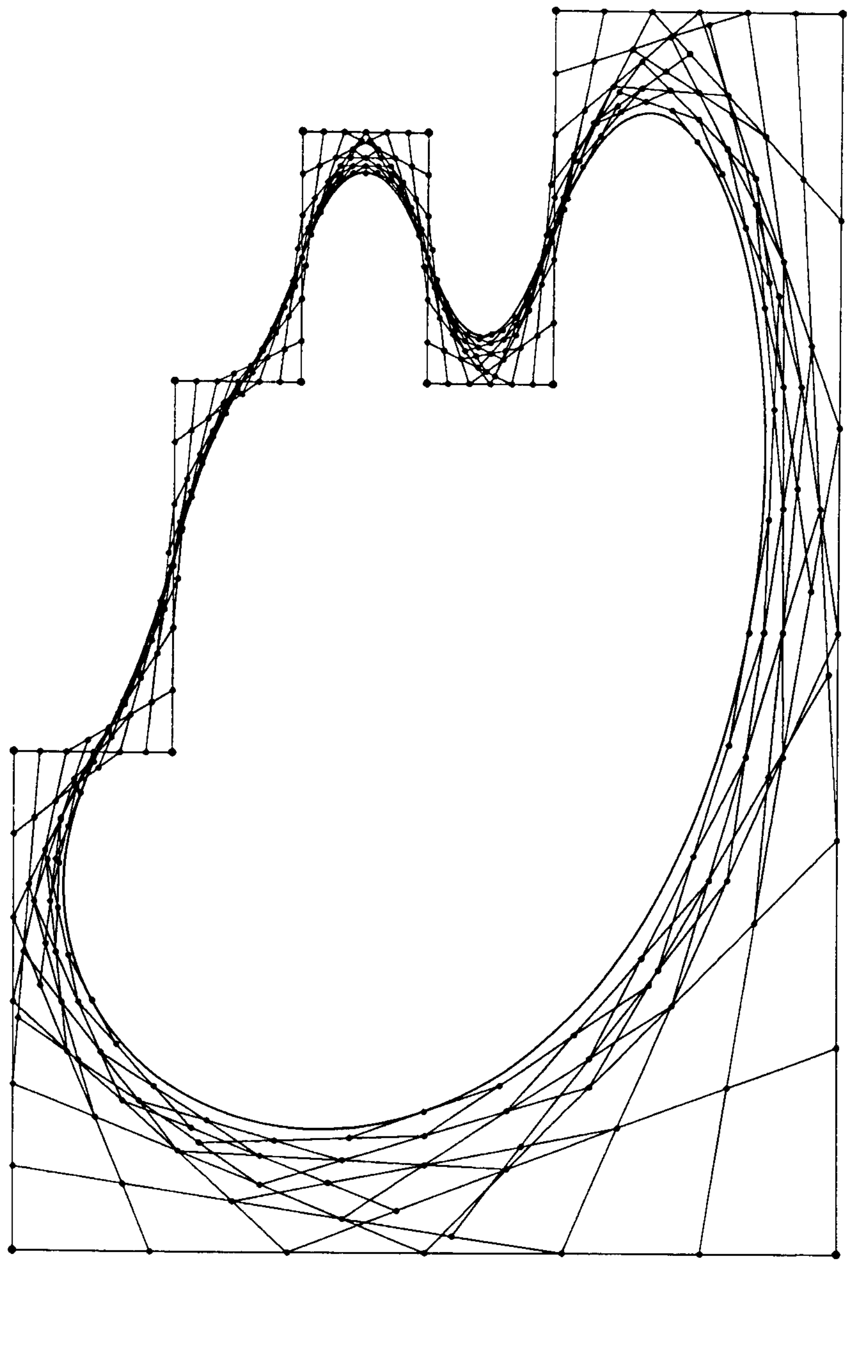


Fig. 17.13. The trellis of a periodic sextic spline curve

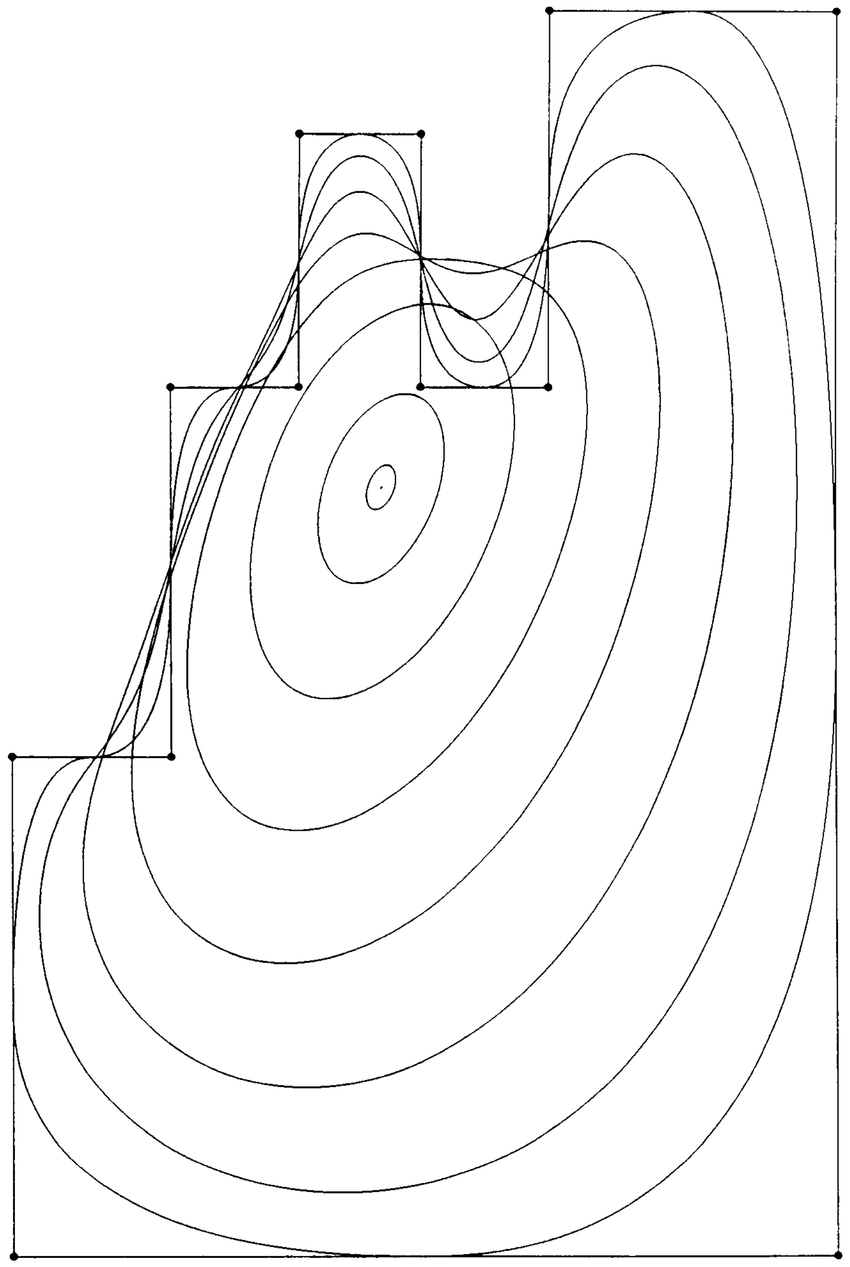


Fig. 17.14. The spline curves of degrees 2^i for i in $[0,9]$

18. The blossoms of B-splines

In the standard theory, a Bézier curve is the result of blending together its Bézier points using the Bernstein basis polynomials as the weights. Analogously, a spline curve is the result of blending together its de Boor points using *B-splines* as the weights. Our blossoming technology led us to the de Boor Algorithm, which is a way of computing a spline curve from its de Boor points. Hence, we already know about B-splines in a sense. But the connections between what we know so far and the standard explanations of B-splines are a bit subtle. In the standard theory [8], B-splines are defined either by a recurrence [10] or by a formula involving divided differences [9]. Spline curves are then defined as linear combinations of B-splines. Finally, it is shown that spline curves can be computed using the de Boor Algorithm [11]. In this section, we shall reverse that process. We shall define B-splines to be the weighting functions that result from the de Boor Algorithm. Then, we shall prove that those B-splines satisfy the recurrence and the divided-difference formula.

Let L be the affine line, let $\{\bar{t}_i\}$ be a knot sequence in L, let $F: L \to Q$ be an n-ic spline curve with $\{\bar{t}_i\}$ as its knot sequence, and let $\{\mathbf{x}_j\}$ where $\mathbf{x}_j = f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n})$ be the sequence of de Boor points of F. The de Boor Algorithm computes each point $F(\bar{u})$ on the spline F as an affine combination of the de Boor points. We define the real-valued spline functions $B_{j,n+1}(\bar{u})$ to be the coefficients in this affine combination:

$$F(\bar{u}) = \sum_j B_{j,n+1}(\bar{u})\, f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}). \tag{18.1}$$

These coefficient functions $B_{j,n+1}(\bar{u})$ are called the ***normalized univariate B-splines of order $n+1$ for the knot sequence*** $\{\bar{t}_i\}$; the notation $N_{j,n+1}(\bar{u})$ is also common. The "$+1$" in the second subscript arises because we are using n to denote the degree rather than the order.

We can compute B-splines by considering spline curves $F: L \to \mathbf{R}$, whose object space is the real numbers. The de Boor points $\mathbf{x}_j = f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n})$ of such a spline are themselves real numbers. From Eq. (18.1), we can see that the B-spline $B_{j,n+1}: L \to \mathbf{R}$ is simply the n-ic spline curve whose de Boor points $\mathbf{x}_i$ in $\mathbf{R}$ are all equal to 0 except for $\mathbf{x}_j$, which is 1. Using the Kronecker delta notation, we can write this more compactly as $\mathbf{x}_i = \delta_{ij}$. Therefore, we can compute the B-spline $B_{j,n+1}$ by applying the de Boor Algorithm, starting with the de Boor points $\mathbf{x}_i = \delta_{ij}$.

One source of complexity in that computation is overloading. Note that both the $F(\bar{u})$ and the $B_{j,n+1}(\bar{u})$ in Eq. (18.1) are splines, that is, piecewise polynomials. The pieces that make up $F(\bar{u})$ are the polynomial curves $F_k(\bar{u})$ for each k with $(\bar{t}_k, \bar{t}_{k+1})$ nonempty. In a similar way, let us define $B_{j,n+1,k}(\bar{u})$ to be the n-ic polynomial that the B-spline $B_{j,n+1}(\bar{u})$ follows for $\bar{u}$ in $(\bar{t}_k, \bar{t}_{k+1})$; we shall call $B_{j,n+1,k}$ a *B-polynomial.* The non-overloaded version of Eq. (18.1) is

$$F_k(\bar{u}) = \sum_j B_{j,n+1,k}(\bar{u})\, f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}), \tag{18.2}$$

which expresses the polynomial curve F_k as a weighted sum of de Boor points, with the B-polynomials as the weights. In particular, the B-polynomial value $B_{j,n+1,k}(\bar{u})$

gives the influence of the jth de Boor point on the value of the kth spline segment at the time $\bar{u}$. If we blossom both sides of Eq. (18.2), we can generalize this observation to bags of times $\{\bar{u}_1, \ldots, \bar{u}_n\}$ whose n components aren't equal. We have

$$f_k(\bar{u}_1, \ldots, \bar{u}_n) = \sum_j b_{j,n+1,k}(\bar{u}_1, \ldots, \bar{u}_n)\, f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}), \qquad (18.3)$$

where $b_{j,n+1,k}$ denotes the blossom of the B-polynomial $B_{j,n+1,k}$.

Although the sums in Eqs. (18.2) and (18.3) are formally infinite, only $n+1$ of the terms are actually nonzero. From Prop. 15.3, we know that the jth de Boor point influences the kth spline segment precisely for k in $[j, j+n]$. Turning this around, we deduce that the kth segment is influenced by the jth de Boor point precisely for j in $[k-n, k]$. For j outside of $[k-n, k]$, the B-polynomial $B_{j,n+1,k}(\bar{u})$ is identically zero. (Throughout this section, the index j will always name a de Boor point and the index k will always name a spline segment. Any value of j is legitimate, but the only legal values of k are those whose domain intervals $(\bar{t}_k, \bar{t}_{k+1})$ are nonempty.)

There is an interesting way to think about the evaluation of the segments $F_k(\bar{u})$ of a spline curve. For each k with $(\bar{t}_k, \bar{t}_{k+1})$ nonempty, we can compute the blossom value $f_k(\bar{u}_1, \ldots, \bar{u}_n)$ by using the Generalized de Casteljau Algorithm starting with the $n+1$ poles of f_k, which are precisely the de Boor points that influence f_k, that is, the points $f_k(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}) = \mathbf{x}_j$ for j in $[k-n, k]$. We found in Section 7 that the resulting computation scheme is triangular, shown for the case $n = 3$ as the downward-pointing triangle in Fig. 7.4. Something interesting happens if we consider fixing the blossom arguments $\{\bar{u}_1, \ldots, \bar{u}_n\}$, but varying k. Of course, this is a pretty weird thing to do. Normally, we are interested in the value of $f_k(\bar{u}_1, \ldots, \bar{u}_n)$ only when the points $\bar{u}_i$ are in the interval $(\bar{t}_k, \bar{t}_{k+1})$, or at least close to that interval. Indeed, if we hadn't removed the overloading first, there wouldn't be any k around to vary: its value would be deduced by some overloading convention from the arguments $\bar{u}_i$. But weird ideas sometimes pay off. What happens in this case is that the triangular computation schemes for different values of k overlap in large part, and can hence be assembled to form one long, diamond-celled lattice, reminiscent of the old-fashioned, unsafe type of baby's gate. Fig. 18.4 shows a chunk of this lattice in the case $n = 3$. The de Boor points $\mathbf{x}_j$ of the spline F form the top row of the lattice. Each subsequent row is computed from the row above by taking affine combinations controlled by one of the arguments $\bar{u}_i$, as specified in the de Boor Algorithm. The bottom row gives the values $\mathbf{y}_k := f_k(\bar{u}_1, \ldots, \bar{u}_n)$ that result from evaluating the blossom f_k of each segment F_k of the spline F on the fixed argument bag $\{\bar{u}_1, \ldots, \bar{u}_n\}$.

If there are multiple knots, this lattice will have triangular notches cut out of the bottom of it, whose heights correspond to knot multiplicities. There are two ways to see this. The naive way is to note that multiple knots would cause some of the edge-labels in the full lattice of Fig. 18.4 to specify division by zero; those edges and the points underneath them must be deleted. More formally, if $\bar{t}_{k+1} = \cdots = \bar{t}_{k+m}$ is a knot of multiplicity $m \geq 2$, we have a computational triangle for f_k and one

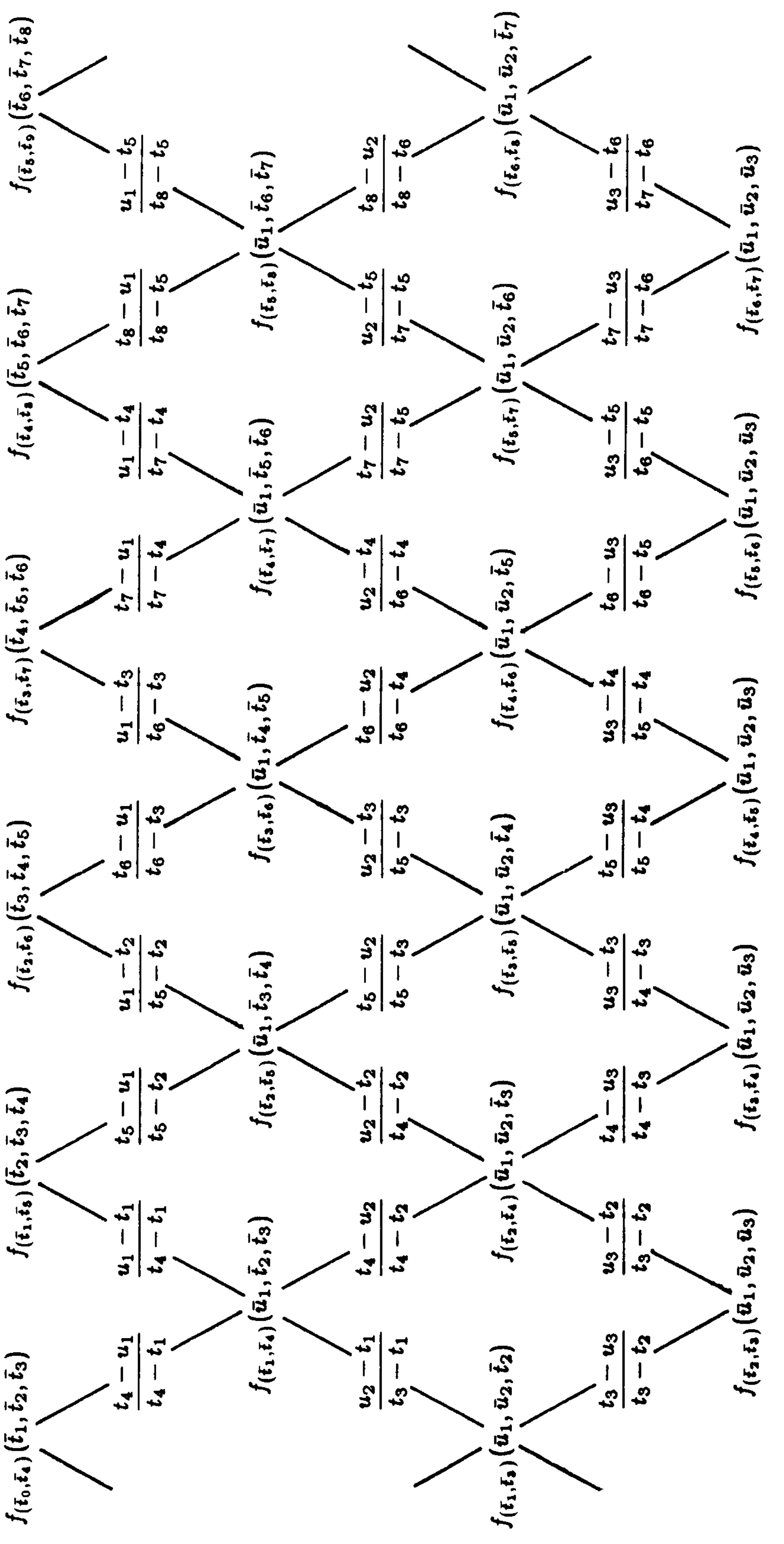

$f_{(\bar t_0,\bar t_4)}(\bar t_1,\bar t_2,\bar t_3)$
$f_{(\bar t_1,\bar t_5)}(\bar t_2,\bar t_3,\bar t_4)$
$f_{(\bar t_2,\bar t_6)}(\bar t_3,\bar t_4,\bar t_5)$
$f_{(\bar t_3,\bar t_7)}(\bar t_4,\bar t_5,\bar t_6)$
$f_{(\bar t_4,\bar t_8)}(\bar t_5,\bar t_6,\bar t_7)$
$f_{(\bar t_5,\bar t_9)}(\bar t_6,\bar t_7,\bar t_8)$
$\frac{t_4-u_1}{t_4-t_1}$
$\frac{u_1-t_1}{t_4-t_1}$
$\frac{t_5-u_1}{t_5-t_2}$
$\frac{u_1-t_2}{t_5-t_2}$
$\frac{t_6-u_1}{t_6-t_3}$
$\frac{u_1-t_3}{t_6-t_3}$
$\frac{t_7-u_1}{t_7-t_4}$
$\frac{u_1-t_4}{t_7-t_4}$
$\frac{t_8-u_1}{t_8-t_5}$
$\frac{u_1-t_5}{t_8-t_5}$
$f_{(\bar t_1,\bar t_4)}(\bar u_1,\bar t_2,\bar t_3)$
$f_{(\bar t_2,\bar t_5)}(\bar u_1,\bar t_3,\bar t_4)$
$f_{(\bar t_3,\bar t_6)}(\bar u_1,\bar t_4,\bar t_5)$
$f_{(\bar t_4,\bar t_7)}(\bar u_1,\bar t_5,\bar t_6)$
$f_{(\bar t_5,\bar t_8)}(\bar u_1,\bar t_6,\bar t_7)$
$\frac{u_2-t_1}{t_3-t_1}$
$\frac{t_4-u_2}{t_4-t_2}$
$\frac{u_2-t_2}{t_4-t_2}$
$\frac{t_5-u_2}{t_5-t_3}$
$\frac{u_2-t_3}{t_5-t_3}$
$\frac{t_6-u_2}{t_6-t_4}$
$\frac{u_2-t_4}{t_6-t_4}$
$\frac{t_7-u_2}{t_7-t_5}$
$\frac{u_2-t_5}{t_7-t_5}$
$\frac{t_8-u_2}{t_8-t_6}$
$f_{(\bar t_1,\bar t_3)}(\bar u_1,\bar u_2,\bar t_2)$
$f_{(\bar t_2,\bar t_4)}(\bar u_1,\bar u_2,\bar t_3)$
$f_{(\bar t_3,\bar t_5)}(\bar u_1,\bar u_2,\bar t_4)$
$f_{(\bar t_4,\bar t_6)}(\bar u_1,\bar u_2,\bar t_5)$
$f_{(\bar t_5,\bar t_7)}(\bar u_1,\bar u_2,\bar t_6)$
$f_{(\bar t_6,\bar t_8)}(\bar u_1,\bar u_2,\bar t_7)$
$\frac{t_3-u_3}{t_3-t_2}$
$\frac{u_3-t_2}{t_3-t_2}$
$\frac{t_4-u_3}{t_4-t_3}$
$\frac{u_3-t_3}{t_4-t_3}$
$\frac{t_5-u_3}{t_5-t_4}$
$\frac{u_3-t_4}{t_5-t_4}$
$\frac{t_6-u_3}{t_6-t_5}$
$\frac{u_3-t_5}{t_6-t_5}$
$\frac{t_7-u_3}{t_7-t_6}$
$\frac{u_3-t_6}{t_7-t_6}$
$f_{(\bar t_2,\bar t_3)}(\bar u_1,\bar u_2,\bar u_3)$
$f_{(\bar t_3,\bar t_4)}(\bar u_1,\bar u_2,\bar u_3)$
$f_{(\bar t_4,\bar t_5)}(\bar u_1,\bar u_2,\bar u_3)$
$f_{(\bar t_5,\bar t_6)}(\bar u_1,\bar u_2,\bar u_3)$
$f_{(\bar t_6,\bar t_7)}(\bar u_1,\bar u_2,\bar u_3)$

Fig. 18.4. The computation scheme of the de Boor Algorithm when $n = 3$

for f_{k+m}, but the intervening index values correspond to empty domain intervals, for which there are no corresponding functions or computational triangles. It is the absence of the triangles for f_{k+1} through f_{k+m-1} that generates a triangular hole of height m at the bottom of the lattice.

The nodes in the lattice in Fig. 18.4 are labeled to reflect a computation of the blossom values $f_k(\bar{u}_1, \dots, \bar{u}_n)$ for the arbitrary spline F. Fig. 18.5 shows the same lattice, but with the node labels omitted. We can use this lattice to compute B-polynomial values $b_{j,n+1,k}(\bar{u}_1, \dots, \bar{u}_n)$ in three different ways.

The first way uses the lattice top-down, the way that we developed it, exploiting the observation that B-splines are just real-valued spline curves whose de Boor points are all 0 except for a single 1. Choosing a fixed j, we set the top-row nodes $\mathbf{x}_i$ by the rule $\mathbf{x}_i := \delta_{ij}$ and then we compute successive rows, as in the de Boor Algorithm. The output of this computation, on the bottom row, will be the values $\mathbf{y}_k = b_{j,n+1,k}(\bar{u}_1, \dots, \bar{u}_n)$ for all k with $\bar{t}_k < \bar{t}_{k+1}$. Intuitively, we have computed the influence of our chosen jth de Boor point on all segments f_k of the spline at the fixed bag of times $\{\bar{u}_1, \dots, \bar{u}_n\}$.

If we label each node in the lattice as we do this top-down computation, what interpretation can we give to the labels? For each node, consider the monotonic paths in the lattice that reach that node, starting from the chosen top-row node $\mathbf{x}_j$. The label on each node will be precisely the sum, over all such paths, of the product of the edge-weights along that path. In particular, suppose that we also choose a fixed node $\mathbf{y}_k$ on the bottom row, that is, a fixed segment of the spline. There will be precisely $\binom{n}{k-j}$ monotonic paths in the lattice from the chosen top-row node $\mathbf{x}_j$ to the chosen bottom-row node $\mathbf{y}_k$. The blossom value $b_{j,n+1,k}(\bar{u}_1, \dots, \bar{u}_n)$ is just the sum, over all such paths, of the products of the edge-weights along that path. This constitutes a second way of using the lattice of Fig. 18.5 to compute values of the B-polynomials.

The symmetry of that second approach between top and bottom suggests that we could also use the lattice yet a third way: bottom-up. Choose a fixed k, with $\bar{t}_k < \bar{t}_{k+1}$. Set the bottom-row nodes $\mathbf{y}_i$ in the lattice by the rule $\mathbf{y}_i := \delta_{ik}$. Then, compute values for nodes in successively higher rows, taking linear combinations as specified by the edge weights. The resulting labels on the interior nodes of the lattice will be quite different, under this bottom-up approach, than they were under the top-down approach. Note, in particular, that the two weights involved in one linear combination on the way down the lattice always sum to 1, making that linear combination an affine combination. But the two weights involved in a linear combination on the way up the lattice generally don't sum to 1. What labels will this bottom-up process give to the nodes in the top row? Since the notion of weighted paths is symmetric, we deduce that the label on the jth top-row node will be $\mathbf{x}_j = b_{j,n+1,k}(\bar{u}_1, \dots, \bar{u}_n)$. Intuitively, the bottom-up approach chooses a spline segment and computes how that segment is influenced by all de Boor points, while the top-down approach chooses a de Boor point and computes how that de Boor point influences all segments.

The bottom-up way of evaluating the lattice in Fig. 18.5 corresponds to one of the two standard definitions for B-splines, the one based on a recurrence relation.

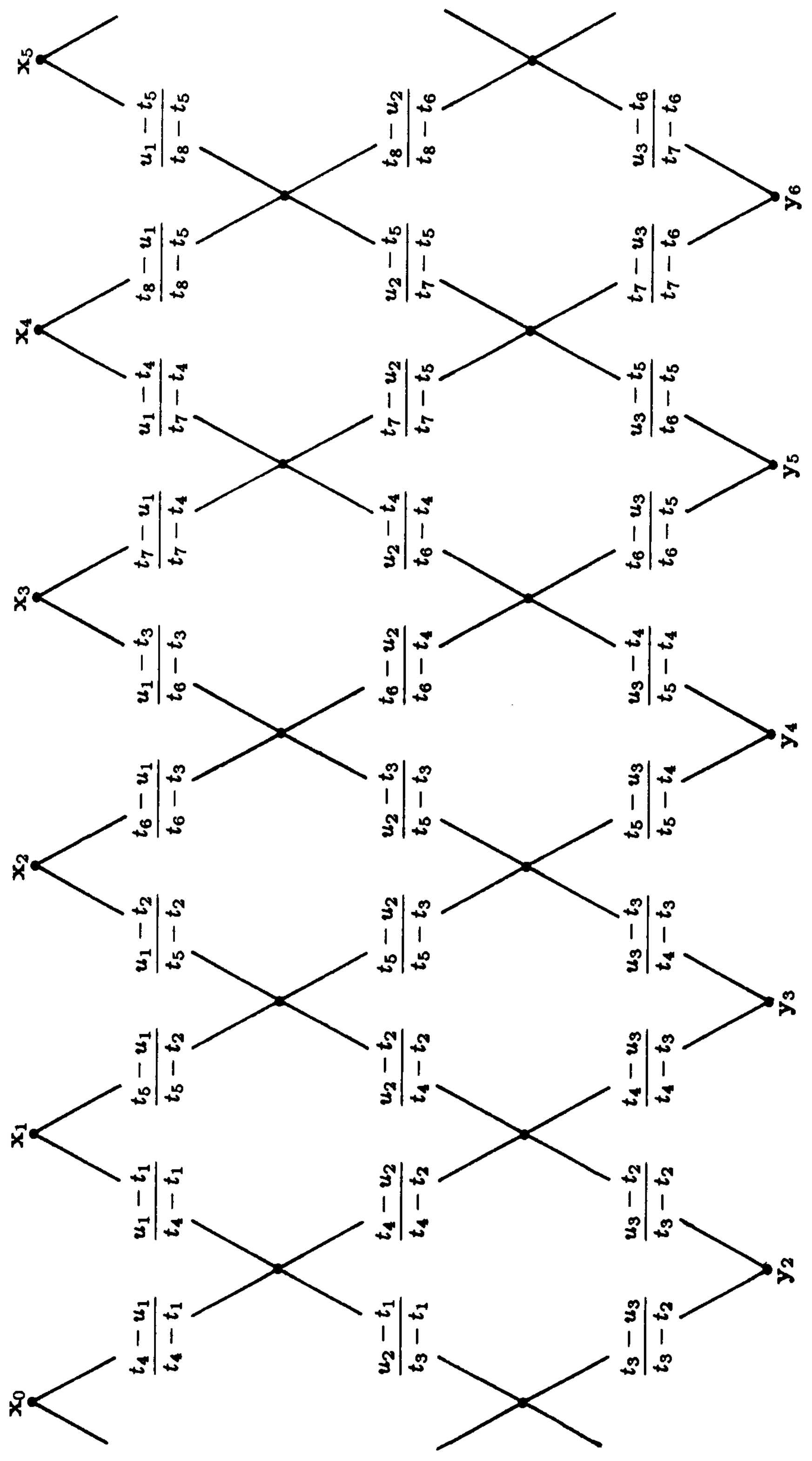

x_0
x_1
x_2
x_3
x_4
x_5
$\frac{t_4-u_1}{t_4-t_1}$
$\frac{u_1-t_1}{t_4-t_1}$
$\frac{t_5-u_1}{t_5-t_2}$
$\frac{u_1-t_2}{t_5-t_2}$
$\frac{t_6-u_1}{t_6-t_3}$
$\frac{u_1-t_3}{t_6-t_3}$
$\frac{t_7-u_1}{t_7-t_4}$
$\frac{u_1-t_4}{t_7-t_4}$
$\frac{t_8-u_1}{t_8-t_5}$
$\frac{u_1-t_5}{t_8-t_5}$
$\frac{u_2-t_1}{t_3-t_1}$
$\frac{t_4-u_2}{t_4-t_2}$
$\frac{u_2-t_2}{t_4-t_2}$
$\frac{t_5-u_2}{t_5-t_3}$
$\frac{u_2-t_3}{t_5-t_3}$
$\frac{t_6-u_2}{t_6-t_4}$
$\frac{u_2-t_4}{t_6-t_4}$
$\frac{t_7-u_2}{t_7-t_5}$
$\frac{u_2-t_5}{t_7-t_5}$
$\frac{t_8-u_2}{t_8-t_6}$
$\frac{t_3-u_3}{t_3-t_2}$
$\frac{u_3-t_2}{t_3-t_2}$
$\frac{t_4-u_3}{t_4-t_3}$
$\frac{u_3-t_3}{t_4-t_3}$
$\frac{t_5-u_3}{t_5-t_4}$
$\frac{u_3-t_4}{t_5-t_4}$
$\frac{t_6-u_3}{t_6-t_5}$
$\frac{u_3-t_5}{t_6-t_5}$
$\frac{t_7-u_3}{t_7-t_6}$
$\frac{u_3-t_6}{t_7-t_6}$
y_2
y_3
y_4
y_5
y_6

Fig. 18.5. The computation lattice underlying the de Boor Algorithm

The blossomed form of that recurrence relation is

$$\begin{aligned} b_{j,1,k}(\,) &:= \delta_{jk} \\ b_{j,n+1,k}(\bar{u}_1,\ldots,\bar{u}_n) &:= \frac{u_n - t_j}{t_{j+n} - t_j} b_{j,n,k}(\bar{u}_1 \ldots, \bar{u}_{n-1}) + \\ &\qquad \frac{t_{j+n+1} - u_n}{t_{j+n+1} - t_{j+1}} b_{j+1,n,k}(\bar{u}_1,\ldots,\bar{u}_{n-1}). \end{aligned}$$

If we set all of the $\bar{u}_i$ equal to a common value $\bar{u}$ and then use overloading to infer the proper value of k from the value of $\bar{u}$, we arrive at the standard version of the B-spline recurrence [10].

Proposition 18.6. *The normalized univariate B-splines $B_{j,n+1}$ for the knot sequence $\{\bar{t}_i\}$, which we defined in Eq. (18.1) as the weights with which the de Boor points are blended together to form an n-ic spline curve, are given by the recurrence*

$$\begin{aligned} B_{j,1}(\bar{u}) &:= \begin{cases} 1 & \textit{if } \bar{u} \in (\bar{t}_j, \bar{t}_{j+1}) \\ 0 & \textit{otherwise} \end{cases} \\ B_{j,n+1}(\bar{u}) &:= \frac{u - t_j}{t_{j+n} - t_j} B_{j,n}(\bar{u}) + \frac{t_{j+n+1} - u}{t_{j+n+1} - t_{j+1}} B_{j+1,n}(\bar{u}). \quad \square \end{aligned}$$

The bottom-up approach has an interesting feature: given a fixed knot sequence $\{\bar{t}_i\}$, we can start off the bottom-up approach without knowing what the degree n is. We first compute the linear B-splines based on the knot sequence $\{\bar{t}_i\}$, then the quadratic B-splines based on that knot sequence, and so on; we can stop whenever we like. By contrast, we have to know the degree at the outset in order to use the top-down approach.

Proposition 18.7. *The normalized univariate B-splines $B_{j,n+1}(\bar{u})$ for the knot sequence $\{\bar{t}_i\}$ can also be computed by the divided-difference formula*

$$B_{j,n+1}(\bar{u}) = (t_{j+n+1} - t_j)[t_j,\ldots,t_{j+n+1}](\cdot - u)^n_+. \tag{18.8}$$

Proof. First, we have to interpret the formula. The expression $(\cdot - u)^n_+$ denotes that function of x whose value is $(x-u)^n$ when $x \geq u$ and is zero otherwise, a so-called ***truncated power function.*** Using an inequality-based form of the Kronecker delta, we could also write that function as

$$x \mapsto (\delta_{x \geq u})(x-u)^n.$$

The expression $[t_j,\ldots,t_{j+n+1}]$ that precedes the truncated power function is an instance of the ***divided-difference operator.*** If $g\colon \mathbf{R} \to \mathbf{R}$ is some function, the divided difference $[t_j,\ldots,t_{j+n+1}]g(x)$ is the coefficient of x^{n+1} in the unique polynomial of degree $n+1$ that interpolates the function g at the $n+2$ sampling points $\{t_j,\ldots,t_{j+n+1}\}$. Divided differences have a beautiful theory, which Carl de Boor lucidly explains in his book on splines [9]. We will hardly scratch the surface of that theory in our verification of Eq. (18.8). To begin that verification, we shall remove the overloading from Eq. (18.8) and then blossom the result.

The overloading in Eq. (18.8) comes from the fact that u is used both in the power function $(x-u)^n$ and also to control the location at which that power function gets truncated. To remove the overloading, let $\bar{s}_k$ be some point in the interval $(\bar{t}_k, \bar{t}_{k+1})$ for each k with $\bar{t}_k < \bar{t}_{k+1}$. Then, we have

$$B_{j,n+1,k}(\bar{u}) = (t_{j+n+1} - t_j)[t_j, \ldots, t_{j+n+1}]\big(x \mapsto (\delta_{x \geq s_k})(x-u)^n\big). \tag{18.9}$$

It doesn't matter which point $\bar{s}_k$ in the interval $(\bar{t}_k, \bar{t}_{k+1})$ we choose as the threshold of truncation because the divided-difference operator only examines the values of its functional argument at the knots t_i. We can blossom both sides of Eq. (18.9) quite easily; switching the constant factor $(t_{j+n+1} - t_j)$ over to the left-hand side to save space, we have

$$\frac{b_{j,n+1,k}(\bar{u}_1, \ldots, \bar{u}_n)}{t_{j+n+1} - t_j} = [t_j, \ldots, t_{j+n+1}]\Big(x \mapsto (\delta_{x \geq s_k}) \prod_{l=1}^{n}(x-u_l)\Big). \tag{18.10}$$

Having generalized Eq. (18.8) to form Eq. (18.10), we now begin to specialize again. Instead of allowing an arbitrary argument bag $\{\bar{u}_1, \ldots, \bar{u}_n\}$, we claim that it is enough to consider the particular argument bags $E_i := \{\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}\}$ for i in $[k-n, k]$. The proof of this claim essentially repeats an argument that we used in the proof of Prop. 15.1. Note first that both sides of Eq. (18.10) are symmetric, multiaffine functions of the points $\bar{u}_i$; for this, we need to know only that the divided-difference operator is linear. Fix any k with the property that $\bar{t}_k < \bar{t}_{k+1}$, and consider the reference sequence $(\bar{t}_{k-n+1}, \ldots, \bar{t}_{k+n})$. From Lemma 6.1, we know that the $n+1$ simple n-tensors $e_i := \bar{t}_{i+1} \cdots \bar{t}_{i+n}$ for i in $[k-n, k]$ form an affine frame for the space $L^{\otimes n}$. We conclude that it is enough to verify Eq. (18.10) for the argument bags E_{k-n} through E_k. This leaves us with the chore of proving

$$\frac{b_{j,n+1,k}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n})}{t_{j+n+1} - t_j} = [t_j, \ldots, t_{j+n+1}]\Big(x \mapsto (\delta_{x \geq s_k}) \prod_{l=1}^{n}(x-t_{i+l})\Big) \tag{18.11}$$

for all i, j, and k with $\bar{t}_k < \bar{t}_{k+1}$ and i in $[k-n, k]$.

Let's work on the left-hand side of Eq. (18.11) first. If we substitute the argument bag $E_i = \{\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}\}$ into Eq. (18.3), we get

$$f_k(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}) = \sum_j b_{j,n+1,k}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n})\, f_{(\bar{t}_j, \bar{t}_{j+n+1})}(\bar{t}_{j+1}, \ldots, \bar{t}_{j+n}).$$

Since k is in $[i, i+n]$, the left-hand side of this equation is actually the ith de Boor point $f_{(\bar{t}_i, \bar{t}_{i+n+1})}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n})$. Since the de Boor points are independent variables, we conclude that the coefficients on the right-hand side must satisfy $b_{j,n+1,k}(\bar{t}_{i+1}, \ldots, \bar{t}_{i+n}) = \delta_{ij}$. Thus, the left-hand side of Eq. (18.11) is given by $\delta_{ij}/(t_{j+n+1} - t_j)$.

It remains to show that the right-hand side of Eq. (18.11) has the same value. Recall that, if $g\colon \mathbf{R} \to \mathbf{R}$ is some function, the divided difference $[t_j, \ldots, t_{j+n+1}]g(x)$

is the coefficient of x^{n+1} in the unique polynomial $Z(x)$ of degree $n+1$ that interpolates the function g at the $n+2$ sampling points $\{t_j, \ldots, t_{j+n+1}\}$. (Repeated sampling points imply osculatory interpolation; that is, if some real number y occurs more than once as a sampling point, say m times for $m > 1$, we demand that the polynomial $Z(x)$ osculate the function $g(x)$ to $(m-1)$st order at y. See Ex. 18.12.) In Eq. (18.11), the function $g(x)$ to be differenced is the truncated n-ic polynomial $g(x) := (\delta_{x \geq s_k})h(x)$ where $h(x) := \prod_{l=1}^{n}(x - t_{i+l})$.

All three indices i, j, and k can now be thought of as specifying the positions of significant events in the knot sequence:

$$\begin{aligned} (t_{i+1}, \ldots, t_{i+n}) &: \text{ roots of } h(x) \\ (t_j, \ldots, t_{j+n+1}) &: \text{ sampling points} \\ \ldots, t_k \mid t_{k+1}, \ldots &: \text{ truncation threshold} \end{aligned}$$

If we sample $g(x)$ at a root of $h(x)$, we will get zero, regardless of the truncation threshold. What happens if we sample $g(x)$ at a knot that isn't a root? Call the knots t_m for $m < i$ the *pre-roots* and the knots t_m for $m > i+n$ the *post-roots*. Since k is in $[i, i+n]$, the truncation definitely affects all of the pre-roots and definitely doesn't affect any of the post-roots. If $j < i$, all sampling points are either roots or pre-roots. Hence, all samplings return zero, the interpolating polynomial $Z(x)$ is identically zero, and the coefficient of x^{n+1} in $Z(x)$ is zero. If $j > i$, all sampling points are either roots or post-roots. Hence, all samplings return $h(x)$, the interpolating polynomial $Z(x)$ is equal to $h(x)$, and the coefficient of x^{n+1} is again zero, since $h(x)$ is an n-ic. The only chance for a nonzero value on the right-hand side of Eq. (18.11) is the case $j = i$. In this case, the interpolating polynomial $Z(x)$ must have all the roots of $h(x)$ as roots and also the last pre-root t_j as a root. Therefore, up to a scalar multiple w, we have $Z(x) = w(x - t_j)h(x)$. To determine the scalar w, we use the fact that Z and h must agree at the first post-root t_{j+n+1}. This implies that $w = 1/(t_{j+n+1} - t_j)$. Since w is also the coefficient of x^{n+1} in $Z(x)$, we conclude that the right-hand side of Eq. (18.11) has the value $\delta_{ij}/(t_{j+n+1} - t_j)$, which agrees with the left-hand side. □

Exercise 18.12. Verify that the arguments in the last few paragraphs of the proof of Prop. 18.7 remain valid even in the presence of multiple knots.

Remark 18.13. I made the bold assertion, back in Section 4, that vector-space duality would never arise in this paper. Well, hardly ever. The linear blossoms $b^{\otimes}_{j,n+1,k,*}$ of the B-polynomials are actually *n-covariant tensors* on L. In fact, for each fixed k, the n-covariant tensors $\{b^{\otimes}_{j,n+1,k,*}\}$ for j in $[k-n, k]$ form a basis for the dual space $(L_*^{\otimes n})^*$ (sorry about the double use of "$*$"), which is the dual of the basis for the linear space $L_*^{\otimes n}$ formed by the n-contravariant tensors $\{\bar{t}_{i+1} \cdots \bar{t}_{i+n}\}$ for i in $[k-n, k]$. That is precisely what the equation

$$b^{\otimes}_{j,n+1,k}(\bar{t}_{i+1} \cdots \bar{t}_{i+n}) = \delta_{ij}$$

says. Perhaps dualizing should be added as a fourth principle, to supplement blossoming, tensoring, and homogenizing.

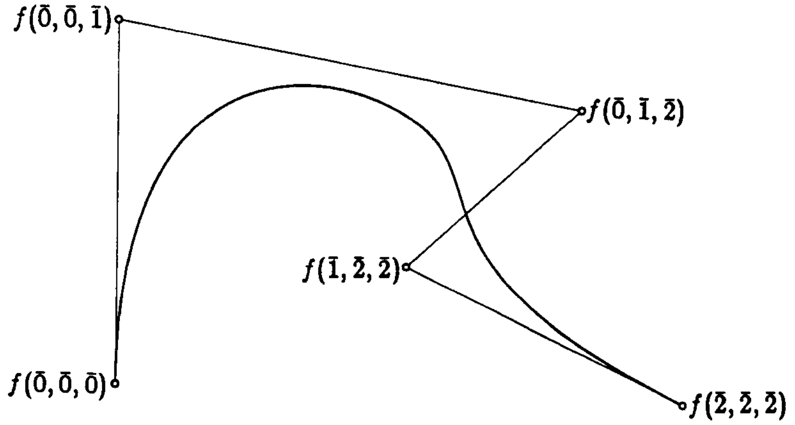

$f(\bar 0,\bar 0,\bar 1)$
$f(\bar 0,\bar 1,\bar 2)$
$f(\bar 1,\bar 2,\bar 2)$
$f(\bar 0,\bar 0,\bar 0)$
$f(\bar 2,\bar 2,\bar 2)$

Fig. 19.1. A cubic spline curve with knot sequence $(\bar 0,\bar 0,\bar 0,\bar 0,\bar 1,\bar 2,\bar 2,\bar 2,\bar 2)$

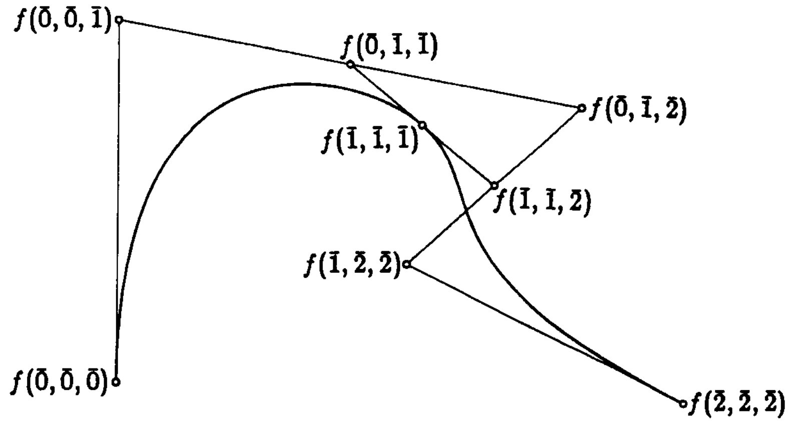

$f(\bar 0,\bar 0,\bar 1)$
$f(\bar 0,\bar 1,\bar 1)$
$f(\bar 0,\bar 1,\bar 2)$
$f(\bar 1,\bar 1,\bar 1)$
$f(\bar 1,\bar 1,\bar 2)$
$f(\bar 1,\bar 2,\bar 2)$
$f(\bar 0,\bar 0,\bar 0)$
$f(\bar 2,\bar 2,\bar 2)$

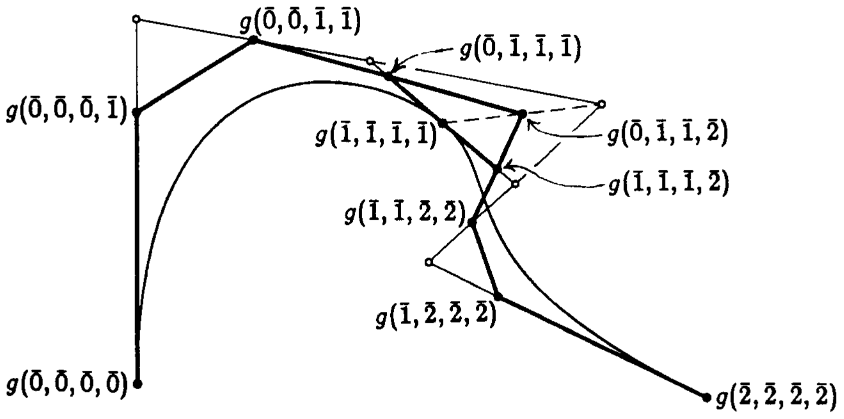
Fig. 19.2. The cubic trellis of the spline curve in Fig. 19.1

19. Raising the degree of a spline curve

Elaine Cohen, Tom Lyche, and Larry L. Schumaker recently published a paper giving algorithms for raising the degree of a spline curve [6]. As another demonstration of blossoming technology, let's go over one example from their paper. We won't be proving anything new in this section. Our goal instead is to get some feel for the rather different perspective on such problems that blossoming provides.

Fig. 19.1 shows a cubic spline curve F, analogous to the curve shown in Fig. 2 of [6], with the knot sequence $(\bar 0,\bar 0,\bar 0,\bar 0,\bar 1,\bar 2,\bar 2,\bar 2,\bar 2)$. The labels on the middle three of the five de Boor points are well-defined under wild overloading; we shall trust to context to disambiguate the first and last de Boor points $f(\bar 0,\bar 0,\bar 0)$ and $f(\bar 2,\bar 2,\bar 2)$. Fig. 19.2 shows the trellis of F, which includes the four Bézier points of each of its two cubic segments.

The problem that Cohen, Lyche, and Schumaker pose is to view this curve F as a degenerate case of a quartic spline. Now, the joint between $F_{(\bar 0,\bar 1)}$ and $F_{(\bar 1,\bar 2)}$ has C^2 continuity. If we think of F as a cubic spline, this fact makes $\bar 1$ a simple knot.



$g(\bar 0,\bar 0,\bar 1,\bar 1)$
$g(\bar 0,\bar 1,\bar 1,\bar 1)$
$g(\bar 0,\bar 0,\bar 0,\bar 1)$
$g(\bar 1,\bar 1,\bar 1,\bar 1)$
$g(\bar 0,\bar 1,\bar 1,\bar 2)$
$g(\bar 1,\bar 1,\bar 1,\bar 2)$
$g(\bar 1,\bar 1,\bar 2,\bar 2)$
$g(\bar 1,\bar 2,\bar 2,\bar 2)$
$g(\bar 0,\bar 0,\bar 0,\bar 0)$
$g(\bar 2,\bar 2,\bar 2,\bar 2)$

Fig. 19.4. The quartic trellis of the spline curve in Fig. 19.1

But, if we think of F as a quartic spline, we must make $\bar 1$ a double knot. In general, every knot has to increase in multiplicity by one when we raise the degree, since the level of continuity of the corresponding joint remains unchanged. If we let G denote F viewed as a quartic spline, the knot sequence of G is $(\bar 0,\bar 0,\bar 0,\bar 0,\bar 0,\bar 1,\bar 1,\bar 2,\bar 2,\bar 2,\bar 2,\bar 2)$.

In Eq. (10.1), we learned how the blossom g_k of any particular segment of G could be computed from the corresponding blossom f_k:

$$g_k(\bar u_1,\bar u_2,\bar u_3,\bar u_4)=\frac{f_k(\bar u_1,\bar u_2,\bar u_3)+f_k(\bar u_1,\bar u_2,\bar u_4)+f_k(\bar u_1,\bar u_3,\bar u_4)+f_k(\bar u_2,\bar u_3,\bar u_4)}{4}. \tag{19.3}$$

How does this equation interact with overloading? If $g(\bar u_1,\bar u_2,\bar u_3,\bar u_4)$ is well-defined under either the tame or the wild convention, the four f-terms in Eq. (19.3) will also be well-defined under the same convention, since the multiplicity of every knot goes down by one as we move from G to F. Therefore, Eq. (19.3) also holds for the overloaded functions f and g, with the subscripts of k deleted. From this, we can easily compute the points of the trellis of G as affine combinations of trellis points of F. Fig. 19.4 shows the trellis of G in bold, superimposed on the trellis of F.

Of the ten trellis points of G, the expressions that we get for nine of them are affine combinations of no more than two trellis points of F. In those nine cases, the G trellis point lies somewhere on the F trellis. For example, we have

$$g(\bar 0,\bar 0,\bar 1,\bar 1)=\frac{2f(\bar 0,\bar 0,\bar 1)+2f(\bar 0,\bar 1,\bar 1)}{4}=f(\bar 0,\overline{0.5},\bar 1).$$

The tenth case is more interesting:

$$g(\bar 0,\bar 1,\bar 1,\bar 2)=\frac{f(\bar 0,\bar 1,\bar 1)+2f(\bar 0,\bar 1,\bar 2)+f(\bar 1,\bar 1,\bar 2)}{4}.$$

It is easy enough to reduce these three f-terms to two in various ways. For example, Fig. 19.4 shows $g(\bar 0,\bar 1,\bar 1,\bar 2)$ as the midpoint of the line from $f(\bar 0,\bar 1,\bar 2)$ to $f(\bar 1,\bar 1,\bar 1)$. Alternatively, it could have shown $g(\bar 0,\bar 1,\bar 1,\bar 2)$ as the midpoint of the line from $f(\bar 0,\bar 1,\overline{1.5})$ to $f(\overline{0.5},\bar 1,\bar 2)$.

If we want a single f-term that names the tenth trellis point $g(\bar{0},\bar{1},\bar{1},\bar{2})$, we can apply Prop. 10.4. It tells us that such an f-term will exist, and that the three arguments of that f-term will be points in the intervals $[\bar{0},\bar{1}]$, $[\bar{1},\bar{1}]$, and $[\bar{1},\bar{2}]$. Computing those points corresponds to factoring the 3-tensor

$$e = \frac{\bar{0}\bar{1}^2 + 2\,\bar{0}\bar{1}\bar{2} + \bar{1}^2\bar{2}}{4}.$$

Using the basis $\{\bar{0},\delta\}$ for L_*, we have

$$\begin{aligned} e &= \bar{1}\left(\frac{\bar{0}(\bar{0}+\delta) + 2\bar{0}(\bar{0}+2\delta) + (\bar{0}+\delta)(\bar{0}+2\delta)}{4}\right) \\ &= \bar{1}\left(\frac{4\bar{0}^2 + 8\bar{0}\delta + 2\delta^2}{4}\right) \\ &= \bar{1}\left(\overline{1+\frac{1}{\sqrt{2}}}\right)\left(\overline{1-\frac{1}{\sqrt{2}}}\right) \approx \bar{1}\,\overline{1.707}\,\overline{0.293}. \end{aligned}$$

20. Geometric continuity for spline curves

As we mentioned back in Section 14, a joint between two polynomial curves is called geometrically continuous of order k, or G^k continuous, if the joining curves can be reparameterized so as to meet with C^k continuity [15]. Let $\bar{s}$ be the parameter value corresponding to the joint. A reparameterization is a smooth (that is, a C^∞) function h from a neighborhood of $\bar{s}$ in the affine line L to another neighborhood of $\bar{s}$ with the property that $h(\bar{s}) = \bar{s}$. In order to guarantee that the reparameterization does not change the order of continuity of the joint, we restrict ourselves to *regular* reparameterizations, which are those with $h'(\bar{s}) \neq 0$. (The derivative $h'(\bar{s})$ is a vector on L, of course, so the zero that we are comparing it with is the zero vector on L.) We also want the alternative notions of time that h relates to agree about what is the past and what is the future; that is, we want h to be *orientation-preserving*. Hence, we actually enforce the sharper constraint $h'(\bar{s}) > 0$. Every regular, orientation-preserving reparameterization has a functional inverse, which is also regular and orientation-preserving. Therefore, instead of reparameterizing both the incoming and outgoing curves, it suffices to reparameterize one of them. We shall follow the majority convention of reparameterizing the incoming curve. Be warned that some sources adopt the opposite convention and reparameterize the outgoing curve [16], with the result that their reparameterizations are the functional inverses of ours.

Let $F\colon (\bar{r},\bar{s}) \to Q$ and $G\colon (\bar{s},\bar{t}) \to Q$ be two n-ic polynomial curves, and suppose that we are interested in the joint between F and G at $\bar{s}$. To test G^k continuity, we compare the derivatives up through order k of the reparameterized incoming curve $F \circ h$ to those of the outgoing curve G. Suppose that the Taylor series of h has the form

$$h(\bar{u}) = \bar{s} + \beta_1(u-s)\delta + \beta_2\frac{(u-s)^2}{2!}\delta + \cdots + \beta_k\frac{(u-s)^k}{k!}\delta + \cdots,$$

where δ denotes the unit vector on L and β_i is the scalar by which δ must be multiplied to give the ith derivative $h^{(i)}(\bar{s})$ of h at $\bar{s}$. Note that only the first k terms in this series matter when testing G^k continuity. For $k = 0$, the condition of G^0 continuity is the trivial relation $F(\bar{s}) = G(\bar{s})$, the same as C^0 continuity. For $k = 1$, we can use the Chain Rule of calculus to find that $G'(\bar{s}) = (F \circ h)'(\bar{s}) = \beta_1 F'(\bar{s})$. For $k > 1$, we can use Faà di Bruno's Formula [30], which is the higher-order analog of the Chain Rule, to find that

$$G^{(k)}(\bar{s}) = \sum_{0 \le j \le k} \sum_{\substack{i_1+i_2+\cdots+i_k=j \\ i_1+2i_2+\cdots+ki_k=k \\ i_1,i_2,\ldots,i_k \ge 0}} \frac{k!}{i_1!(1!)^{i_1}\cdots i_k!(k!)^{i_k}} \beta_1^{i_1}\cdots\beta_k^{i_k} F^{(j)}(s).$$

De Rose calls these equations the ***univariate Beta Constraints*** [15]. Here are the first few of them:

$$G(\bar{s}) = F(\bar{s}) \qquad (BC_0)$$
$$G'(\bar{s}) = \beta_1 F'(\bar{s}) \qquad (BC_1)$$
$$G''(\bar{s}) = \beta_1^2 F''(\bar{s}) + \beta_2 F'(\bar{s}) \qquad (BC_2)$$
$$G^{(3)}(\bar{s}) = \beta_1^3 F^{(3)}(\bar{s}) + 3\beta_1\beta_2 F''(\bar{s}) + \beta_3 F'(\bar{s}). \qquad (BC_3)$$

The curves F and G meet with G^k continuity if and only if real constants β_1 through β_k exist with $\beta_1 > 0$ that satisfy Constraints (BC_0) through (BC_k). The numbers β_i are called ***shape parameters.***

The first shape parameter β_1 is rather special. Consider the reparameterization $h(\bar{u}) := \bar{s} + \beta_1(u - s)\delta$. Since h is an affine map, the reparameterized incoming curve $F \circ h$ will also be an n-ic polynomial curve. Indeed, the behavior of $F \circ h$ on the interval $[h^{-1}(\bar{r}), \bar{s}]$ will correspond precisely to the behavior of F itself on the interval $[\bar{r}, \bar{s}]$, the only difference being that the two domain intervals differ in length by a factor of β_1. But the length of the parametric interval on which each spline segment is defined is already an arbitrary value in our framework, set by the designer's choice of the knot sequence. If we also let the designer specify an arbitrary value for β_1, we would end up with two handles on the same degree of freedom, which would be both pointless and confusing. To avoid this, we shall restrict ourselves to the case $\beta_1 = 1$. The other reasonable choice would be to allow β_1 to assume different values at different joints but to restrict the placements of the knots in such a way that each segment of the resulting spline curve is defined over an interval of unit length. One problem with this alternative scheme is that it makes it harder to study the limiting process in which two simple knots coalesce to form a double knot. In the scheme that we are adopting, the knots can merge smoothly. Indeed, if two simple knots coalesce, each corresponding to a joint with G^{n-1} continuity, the result is a double knot, corresponding to a joint with G^{n-2} continuity, whose associated reparameterization is the functional composition of the reparameterizations of the coalescing knots. In the alternative scheme where β_1 is allowed to vary but all intervals on the parameter line must have unit length, the β_1 value of the joint entering the disappearing segment tends to zero while the β_1 value of the joint leaving that segment goes to infinity in such a way that the product of these two β_1 values remains constant.

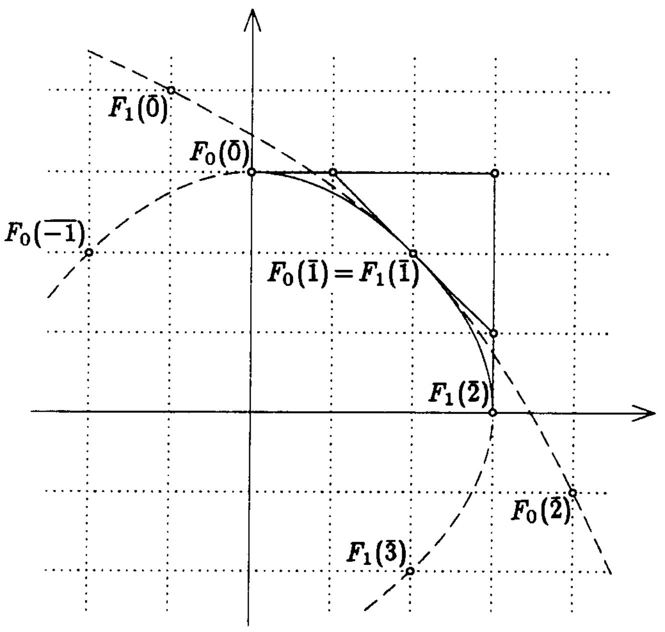


Fig. 20.1. Two parabolic segments that join with G^2 continuity

The shape parameters from β_2 on up really do give us the ability to form new joints. As an example, consider the two parabolas shown in Fig. 20.1, the curves $y = 3 - x^2/4$ and $x = 3 - y^2/4$. These two parabolas both go through the point $\langle 2,2\rangle$, and they both have slope -1 there. By symmetry, we deduce that they have the same curvature there as well. Suppose that we form a quadratic polynomial spline curve as follows. For u in $(\bar{0},\bar{1})$, we define $F_0(\bar{u}) = \langle 2u, 3-u^2\rangle$, thus following the parabola $y = 3-x^2/4$ from the point $\langle 0,3\rangle$ to the point $\langle 2,2\rangle$. For $\bar{u}$ in $(\bar{1},\bar{2})$, we define $F_1(\bar{u}) = \langle 4u - u^2 - 1, 4 - 2u\rangle$, thus following the parabola $x = 3 - y^2/4$ from the point $\langle 2,2\rangle$ to the point $\langle 3,0\rangle$. An easy computation shows that $F_0'(\bar{1}) = \langle 2,-2\rangle$ and $F_0''(\bar{1}) = \langle 0,-2\rangle$, while $F_1'(\bar{1}) = \langle 2,-2\rangle$ and $F_1''(\bar{1}) = \langle -2,0\rangle$. Hence, from the perspective of parametric continuity, the spline curve F is C^1 but not C^2 at $\bar{u} = \bar{1}$; that is to say, $\bar{1}$ is a simple knot.

Since F has curvature continuity at $\bar{1}$, as we argued above by symmetry, we would expect F to be G^2 continuous there. To verify that this is the case, we substitute the derivatives above into the Beta Constraints. Constraint (BC_0) holds and Constraint (BC_1) holds for $\beta_1 = 1$, which is the only legitimate value for β_1 according to our convention. We can make (BC_2) hold by choosing $\beta_2 = -1$:

$$F_1''(1) = \langle -2,0\rangle = \beta_1^2 F_0''(1) + \beta_2 F_0'(1) = \langle 0,-2\rangle - \langle 2,-2\rangle.$$

Thus, the spline F does have G^2 continuity at the point $u = 1$.

Note that the joint between $F_0([\bar{0},\bar{1}])$ and $F_1([\bar{1},\bar{2}])$ is surrounded by three lines that form a roman letter "A", much like a de Casteljau Diagram with 2 shells, except that the ratios of the lengths of the segments aren't in the de Casteljau pattern. This observation is valid in general. Starting from Eq. (8.3) and the Beta Constraints, it

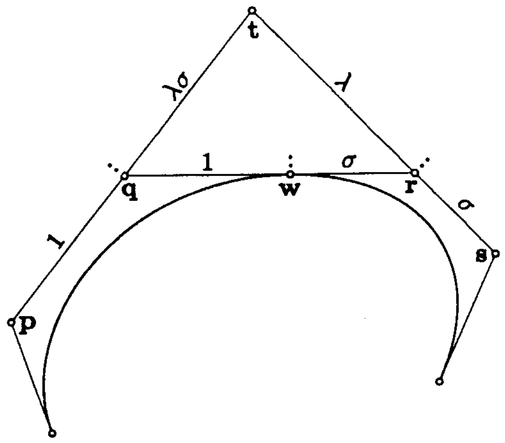


Fig. 20.2. The A-frame surrounding a G^2 joint between cubic segments

is straightforward algebra to check that the two Bézier n-ic segments $F\colon (\bar{r}, \bar{s}) \to Q$ and $G\colon (\bar{s}, \bar{t}) \to Q$ meet with G^2 continuity at $\bar{s}$ if and only if the last three Bézier points of $F([\bar{r}, \bar{s}])$ and the first three Bézier points of $G([\bar{s}, \bar{t}])$ are coplanar and they can be connected by lines to form an *A-frame*, as shown in Fig. 20.2, where the apex of the "A" is a point added in the construction. The distance ratios σ and λ are related to the knot values and the shape parameters by the formulas

$$\sigma = \beta_1 \frac{t-s}{s-r} \quad \text{and} \quad \frac{1}{\lambda} - 1 = \frac{\beta_2}{(n-1)\sigma(\sigma+1)} \left(\frac{t-s}{s-r}\right)^2 .$$

In our special case of $\beta_1 = 1$, these formulas simplify to

$$\sigma = \frac{t-s}{s-r} \quad \text{and} \quad \frac{1}{\lambda} - 1 = \frac{\beta_2}{n-1} \left(\frac{\sigma}{\sigma+1}\right) .$$

The joint between F and G has C^2 continuity when $\beta_2 = 0$, and hence $\lambda = 1$. In this case, the joint is surrounded by an honest de Casteljau Diagram of order 2, in which all three line segments are divided in the ratio $1 : \sigma$. As we move λ away from 1, we get joints that are G^2 continuous but not C^2 continuous. The example joint between parabolas in Fig. 20.1 has $\lambda = 2$.

The name "A-frame" is most appropriate in the case where λ is some positive constant. It is also possible to have G^2 joints in which $\lambda = \infty$ or $\lambda < 0$. In the case $\lambda = \infty$, the legs of the "A" are parallel and the apex of the "A" has moved off to the line at infinity; the name Π-*frame* seems appropriate for this case. When $\lambda < 0$, the legs of the "A" point in towards the bottom, and we might call the result a ∇-*frame*.

Exercise 20.3. Generalize the concept of an A-frame from $k = 2$ to arbitrary k. More precisely, prove the following: for any k, G^k continuity at the joint between two n-ic curves is enforced by a diagram with the same points, lines, and incidence structure as a de Casteljau Diagram with k shells, but with different distance ratios. The distance ratios that arise for $k \geq 3$ are, unfortunately, rather complicated functions of the shape parameters.

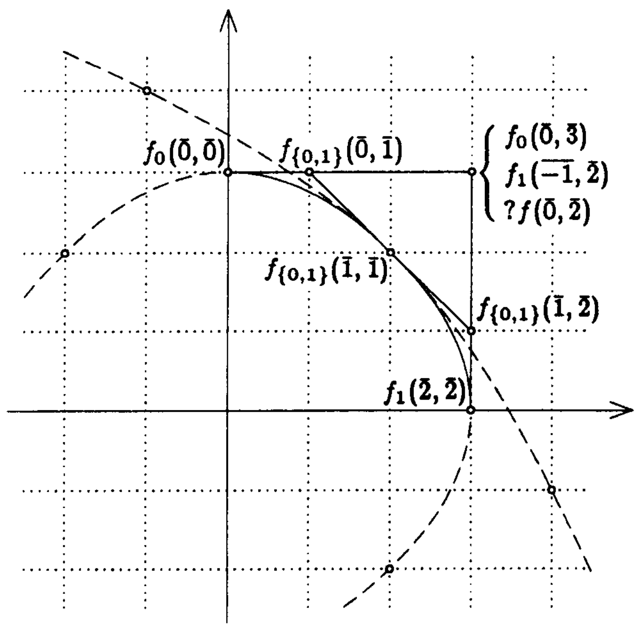

$f_0(\bar{0},\bar{0})$
$f_{\{0,1\}}(\bar{0},\bar{1})$
$f_0(\bar{0},\bar{3})$
$f_1(\overline{-1},\bar{2})$
$?f(\bar{0},\bar{2})$
$f_{\{0,1\}}(\bar{1},\bar{1})$
$f_{\{0,1\}}(\bar{1},\bar{2})$
$f_1(\bar{2},\bar{2})$

Fig. 20.5. Attempts at labeling the vertices of an A-frame

Challenge 20.4. Figure out a comprehensible general form for those distance ratios. Use this form to analyze the different ways that a G^k joint can be "turned inside out," as the shape parameters are adjusted, by having portions of the enforcing diagram pass through each other or through the hyperplane at infinity. While you are at it, figure out good ways to control spline curves with G^k joints for $k \geq 3$.

What happens if we study the A-frame in Fig. 20.1 from a blossoming point of view? If f_0 and f_1 denote the blossoms of F_0 and F_1, five of the six points in the A-frame are easy to label as blossom values, as is done in Fig. 20.5. Note that f_0 and f_1 will agree on all argument bags that include a $\bar{1}$, since the curves F_0 and F_1 join with C^1 continuity there. The big question is what to label the apex of the A-frame.

Since f_0 is multiaffine, one possible label for the apex is $f_0(\bar{0},\bar{3})$. Symmetrically, we could also label it $f_1(\overline{-1},\bar{2})$. The difficulty with these labels is that it isn't obvious where the new arguments $\bar{3}$ and $\overline{-1}$ have come from. Note that the proper values for these new arguments depend upon λ, and hence upon the shape parameter β_2.

From the point of view of understanding the structure of the spline curve F, it would be much nicer to label the apex of the A-frame $f(\bar{0},\bar{2})$, for some meaning of the function symbol f. Note that the curvature continuity of F at the joint implies that the tangent lines $\operatorname{Osc}_1 F(\bar{0})$ and $\operatorname{Osc}_1 F(\bar{2})$ to F at the points $\bar{u}=\bar{0}$ and $\bar{u}=\bar{2}$ are not skew, as they might be if the joint at $\bar{u}=\bar{1}$ was only C^1, but actually intersect. By the differential perspective on the meaning of a blossom's arguments, we would like to label that intersection point $f(\bar{0},\bar{2})$. If we use the same tangent-line rule to determine the value of $f(\bar{u},\bar{v})$ for all $\bar{u}$ and $\bar{v}$ in $[\bar{0},\bar{2}]$, the resulting function will be rather complicated. If we have both $u \leq 1$ and $v \leq 1$ or both $u \geq 1$ and $v \geq 1$, the value $f(\bar{u}, \bar{v})$ will behave in a biaffine manner as we vary u and v. But, if we have $u < 1 < v$, the value $f(\bar{u}, \bar{v})$ responds to changes in u and v in a complex manner, which depends upon the shape parameter β_2.

Challenge 20.6. Find a simple and enlightening way to blossom a spline curve that has geometric continuity. If your technique gives the apex of the A-frame in Fig. 20.5 the label $f(\bar{0}, \bar{2})$, it should include an insightful study of the properties of a non-multiaffine blossom, such as the resulting f.

Geometric continuity is more subtle than parametric continuity in many ways; here are three to think about.

First, geometric joints have associated shape parameters. It is tempting to try to use those new degrees of freedom to get more smoothness or more flexibility out of spline curves of some given degree. Unfortunately, such schemes are fraught with peril because of the nonlinear nature of the Beta Constraints. In order to exploit the shape parameters to achieve some abstract goal, one would generally have to determine their values by solving nonlinear equations—in the worst case, a global system of nonlinear equations over the whole spline. The current successful schemes for drawing spline curves with geometric continuity skirt this peril in different ways. The various flavors of Beta-splines [1, 2, 26] avoid the nonlinearity entirely by treating the shape parameters as dials that the designer can turn to explore a wider universe of smooth shapes. Once the shape parameters have been set by the designer, the problem of nonlinearity disappears. Gerald Farin suggested a wonderfully simple technique for drawing G^2 cubic splines, in which the designer specifies the two middle Bézier points of each cubic segment [21]. With the notations of Fig. 20.2, Farin's insight was that the location of the joint $\mathbf{w}$ could be computed from the locations of the four coplanar points $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{s}$ by the equation

$$\left(\frac{\mathbf{r}-\mathbf{w}}{\mathbf{w}-\mathbf{q}}\right)^2 = \sigma^2 = \left(\frac{\lambda\sigma}{1}\right)\left(\frac{\sigma}{\lambda}\right) = \left(\frac{\mathbf{t}-\mathbf{q}}{\mathbf{q}-\mathbf{p}}\right)\left(\frac{\mathbf{s}-\mathbf{r}}{\mathbf{r}-\mathbf{t}}\right).$$

The problem of nonlinearity shows up in Farin's scheme as the restriction that the points $\mathbf{p}$ and $\mathbf{s}$ must lie on the same side of the crossbar, the line joining $\mathbf{q}$ to $\mathbf{r}$; otherwise, the formula for σ demands taking the square root of a negative number.

A second interesting thing about geometric continuity is that it enables the existence of knots with multiplicity zero. A knot of multiplicity m in an n-ic spline is, in the geometric case, the parameter value of a joint that has G^{n-m} continuity, with associated shape parameters β_2 through β_{n-m}. It is perfectly possible to have $m = 0$, that is, to have a G^n joint in an n-ic spline. In fact, the quadratic spline in Fig. 20.5 is a case in point. It has $\bar{0}$ and $\bar{2}$ as triple knots, and it has $\bar{1}$ as a zero-fold knot with $\beta_2 = -1$.

It is even possible to have knots with negative multiplicity, since, in certain degenerate situations, two n-ic curves can meet with G^k continuity for some $k > n$ without being the same curve. Consideration of the Beta Constraints shows that this can happen only when the n-ic curves being joined are degenerate in the sense that they lie in a flat of dimension less than n, since the relations (BC_{n+1}) through

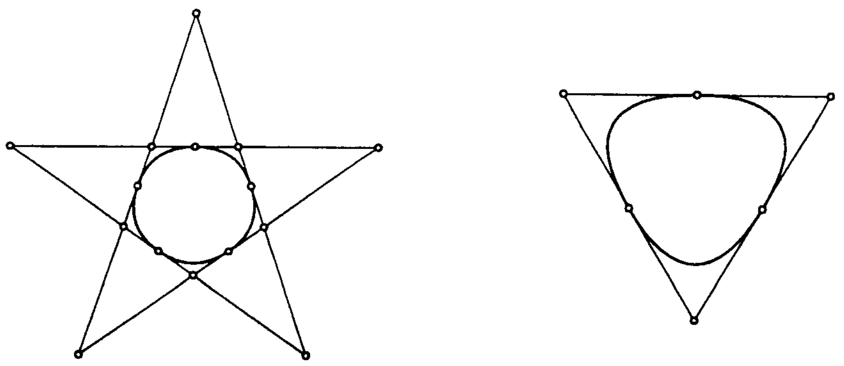


Fig. 20.7. Two quadratic spline curves with G^2 joints

(BC_k) will be linear dependencies among the derivatives. An extreme example is the two n-ic plane curves $F(\bar{u}) := \langle u, u^n \rangle$ and $G(\bar{u}) := \langle u + u^n, u^n \rangle$. These two curves meet with G^{2n-2} continuity at $\bar{u} = \bar{0}$, but not with G^{2n-1} continuity. To see this, one can examine the Beta Constraints to find that the values $\beta_2 = \cdots = \beta_{n-1} = 0$, $\beta_n = n!$, $\beta_{n+1} = \cdots = \beta_{2n-2} = 0$ are the unique choices that will make (BC_0) through (BC_{2n-2}) hold, but that no value of β_{2n-1} will make (BC_{2n-1}) hold. The corresponding reparameterization is $h(\bar{u}) := \overline{u + u^n}$. This means that the parameter value $\bar{0}$ where the curve segments $F([\overline{-1}, \bar{0}])$ and $G([\bar{0}, \bar{1}])$ meet is a knot of multiplicity $2 - n$. Note that a knot of negative multiplicity, by enforcing a degeneracy, reduces the number of control points needed, just as a knot of positive multiplicity increases their number. We would need $n + 1$ control points to specify a single polynomial n-ic segment $H([\overline{-1}, \bar{1}])$; but the two-segment spline formed out of $F([\overline{-1}, \bar{0}])$ and $G([\bar{0}, \bar{1}])$ needs only $(n + 1) + (2 - n) = 3$ control points.

For our third example of the mysteries of geometric continuity, consider the five arcs of parabolas shown in the pentagram on the left in Fig. 20.7. Each joint between one arc and the next is G^2 continuous, since it is surrounded by an A-frame with $\sigma = 1$ and $\lambda = 2\phi$ (corresponding to $\beta_2 = \phi - 3$), where $\phi := (1+\sqrt{5})/2$ denotes the golden ratio. Therefore, suppose that a designer asked us to draw a quadratic spline curve with the parameter interval $[\bar{0}, \bar{5}]$, with $\bar{0}$ and $\bar{5}$ as triple knots, and with each point in the set $\{\bar{1}, \bar{2}, \bar{3}, \bar{4}\}$ a zero-fold knot with $\beta_2 = \phi - 2$. Zero-fold knots don't affect the number of control points, so we would expect such a spline curve to need three control points, the same as a single parabolic segment. In fact, every such spline curve F will have, as trellis, an affine transform of the pentagram shown in Fig. 20.7; in particular, every such spline will satisfy $F(\bar{0}) = F(\bar{5})$. Thus, it does indeed take three control points to specify an instance of such a spline, but we couldn't use the starting point and the ending point as two of those three control points. The triangle on the right in Fig. 20.7 shows that the same phenomenon of enforced circularity can arise with only three segments, if we are willing to allow ∇-frames, that is, $\lambda < 0$. Each joint between two parabolic arcs in that triangle is G^2 with $\sigma = 1$, $\lambda = -2$, and $\beta_2 = -3$. The corresponding construction in a square would have four Π-frames, each with $\sigma = 1$, $\lambda = \infty$, and $\beta_2 = -2$.

Part F: Spline Surfaces

Spline surfaces are considerably more subtle than spline curves. Blossoming can give us a new perspective on the subtleties involved, but it doesn't magically make them go away. Because of those subtleties, we shall study spline surfaces with a rather different strategy. Instead of trying to be general, we shall focus on two of the simplest and most symmetric situations imaginable: patches that are squares and patches that are equilateral triangles. In Section 22, when we deal with square patches, we will end up succeeding in our quest for good labels. In Section 23, when we tackle triangular patches, we will be left with the tantalizing possibility that good labels of some as-yet-unknown structure might exist. Section 21 sets the stage for those two investigations.

21. Joints between polynomial surfaces

Before one can construct a spline function $F: P \to Q$, one must partition the parameter space P into regions. In the case of spline curves, the parameter space P is the affine line L; since we presumably want our regions to be connected, the only choice is to partition L into intervals. But the parameter space of a spline surface is a plane, and there is a tremendous variety of ways to partition a plane into connected regions. Since we will want the surface patches that are defined on adjacent regions to join with at least C^0 continuity, the boundaries between adjacent regions can't be too chaotic. In particular, they had better be segments of algebraic plane curves. But that still leaves plenty of freedom of choice.

In our first step away from generality and towards simplicity, we shall restrict ourselves to the case where the parameter plane P is partitioned into convex polygons. This means that the boundary between two adjacent regions in P is a line segment.

We want the surface patches defined on adjacent regions in P to join with some flavor of continuity. One could choose to enforce either a parametric or a geometric continuity constraint. In our second step towards simplicity, we shall restrict ourselves to the parametric case. That is, if A and B are two adjacent regions in the partition of P, we shall demand that the polynomial surfaces $F_A: A \to Q$ and $F_B: B \to Q$ agree to kth order at all points $\mathbf{u}$ along the boundary between A and B, for some choice of k, without any reparameterization.

Taken together, these two simplifying assumptions have an interesting consequence, when viewed from a blossoming point of view.

Proposition 21.1. *Let A and B be two adjacent regions in the parameter plane P that meet along the segment $[\mathbf{s}, \mathbf{t}]$ of the line L in P, where $\mathbf{s} \neq \mathbf{t}$. Two n-ic polynomial surfaces $F_A: A \to Q$ and $F_B: B \to Q$ join with C^k continuity along the segment $[\mathbf{s}, \mathbf{t}]$ if and only if their multiaffine blossoms f_A and f_B agree on all argument bags that include at least $(n-k)$ points anywhere along L.*

Proof. By Prop. 8.8, F_A and F_B will agree to kth order at some point $\mathbf{v}$ in $[\mathbf{s}, \mathbf{t}]$ if and only if their blossoms satisfy $f_A^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_k \mathbf{v}^{n-k}) = f_B^{\otimes}(\mathbf{u}_1 \cdots \mathbf{u}_k \mathbf{v}^{n-k})$. If this identity holds for all $\mathbf{v}$ in the segment $[\mathbf{s}, \mathbf{t}]$, it must also hold for $\mathbf{v}$ anywhere on the entire line L, since f_A and f_B are given by polynomials. Furthermore, suppose that we fix $\mathbf{u}_1$ through $\mathbf{u}_k$ for a moment. The resulting function

$$\mathbf{v} \mapsto f_{\{A,B\}}(\mathbf{u}_1, \ldots, \mathbf{u}_k, \underbrace{\mathbf{v}, \ldots, \mathbf{v}}_{n-k})$$

is a polynomial curve of degree $n - k$ defined on the line L. The blossom of this curve can be computed either by the rule

$$\{\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\} \mapsto f_A(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n)$$

or by the same rule with f_B replacing f_A. Since blossoms are unique, we conclude that $f_A(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n)$ must equal $f_B(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n)$, even if $\mathbf{v}_{k+1}$ through $\mathbf{v}_n$ are different points along L. □

In this situation, the line L that separates the regions A and B is called a *knotline*, and the number $m := n - k$ is its *multiplicity*. We can restate Prop. 21.1 in these terms as follows: two polynomial surfaces of degree n meet with C^{n-m} continuity along a knotline L if and only if their blossoms agree on all argument bags that include at least m points along L.

It is also worthwhile figuring out what the multiplicity of a knotline means in terms of the locations of the Bézier points of the joining surface patches. In the case of a C^k joint between two curve segments, we found that the last $k+1$ Bézier points of the entering segment $F([\bar{r}, \bar{s}])$ were joined to the first $k+1$ Bézier points of the leaving segment $G([\bar{s}, \bar{t}])$ by a de Casteljau Diagram with k shells. Joints between surface patches are analogous, but the analogy is rich enough that a few pictures are in order. In particular, it will turn out that C^k continuity between two n-ic surface patches is guaranteed by $n-k+1$ overlapping two-dimensional de Casteljau Diagrams, each with k shells. (A two-dimensional de Casteljau Diagram is one, like the one shown in Fig. 3.4, that is constructed using two-dimensional interpolations; it is each interpolation, not the diagram as a whole, that is two-dimensional.)

Before we can compute Bézier points for the joining patches, we must choose a reference triangle for each of them. This choice was pretty obvious in the case of curves: the domain of each curve segment was an interval, and we used that same interval as the Bézier reference interval. If our surface patches are triangular, it is similarly natural to choose the domain triangle of each patch as the Bézier reference triangle for that patch. But domain regions that are polygons with more than three vertices don't present any such obvious choice. Fortunately, it isn't hard to convert from one reference triangle to another. Suppose that we are given the Bézier points of an n-ic surface F with respect to the reference triangle $\Delta\mathbf{rst}$ in P; that is, we are given the triangle of blossom values $f^{\otimes}(\mathbf{r}^i\mathbf{s}^j\mathbf{t}^k)$ for $i + j + k = n$. If we apply the de Casteljau Algorithm for the point $\mathbf{u}$ in P, it produces for us the tetrahedron of blossom values $f^{\otimes}(\mathbf{r}^i\mathbf{s}^j\mathbf{t}^k\mathbf{u}^l)$ for $i + j + k + l = n$. The four faces of this tetrahedron are the Bézier points of F with respect to each of the four reference triangles $\Delta\mathbf{rst}$, $\Delta\mathbf{rsu}$, $\Delta\mathbf{rtu}$, and $\Delta\mathbf{stu}$. By applying the de Casteljau Algorithm three times, we could convert from the initial reference triangle $\Delta\mathbf{rst}$ to an arbitrary reference triangle $\Delta\mathbf{uvw}$.

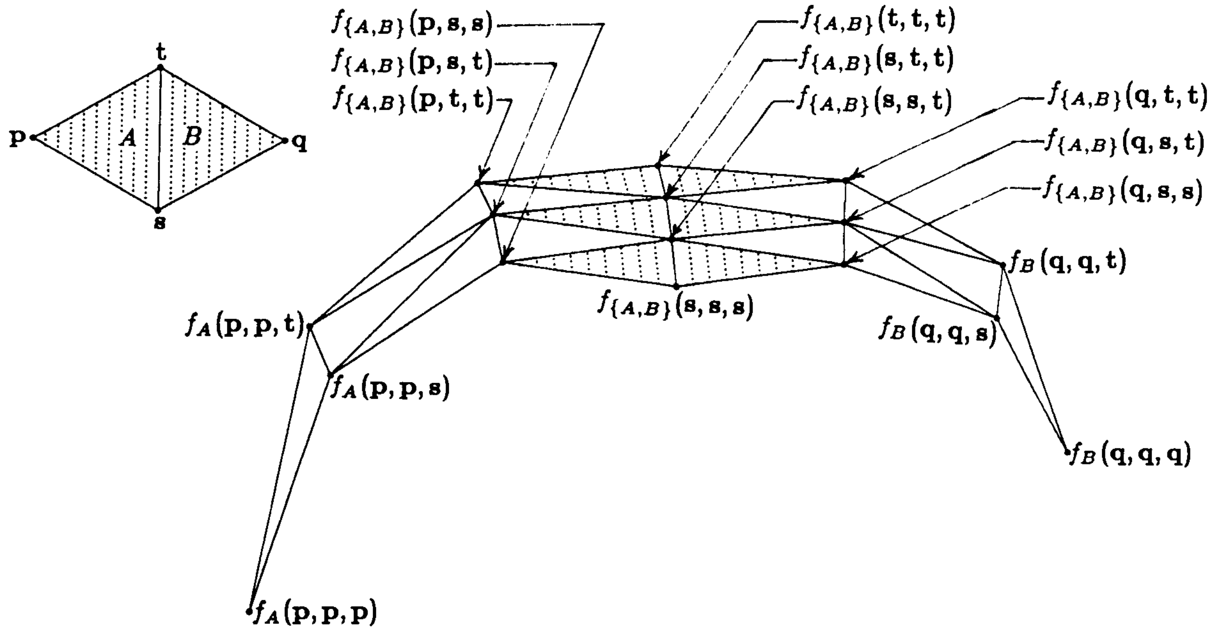


Fig. 21.2. The Bézier points of two cubic surfaces that join with C^1 continuity

For the study of a knotline L, it is convenient to choose the reference triangles of the two joining surface patches so that they share an edge that lies along L. Let A be some region to the left of L in P and let B be some region to the right of L. Suppose that we know the Bézier points of the surface patch F_A with respect to the reference triangle Δpst and those of F_B with respect to Δqst, where s and t are two points on L. If L is a knotline of multiplicity m, Prop. 21.1 tells us that $f_A^{\circledast}(p^i q^j s^k t^l) = f_B^{\circledast}(p^i q^j s^k t^l)$ for $k + l \geq m$. It is these points, on which the two blossom agree, that form the constraint diagrams surrounding the joint.

For C^0 continuity, when $m = n$, the only blossom agreement is given by $f_A^{\circledast}(s^k t^{n-k}) = f_B^{\circledast}(s^k t^{n-k})$; that is, the Bézier points that determine the edge curve $F_A(L)$ must agree with the Bézier points of $F_B(L)$.

For C^1 continuity, we can replace one of the copies of either s or t in the argument bag by either p or q without destroying agreement between the blossoms. Fig. 21.2 depicts the case $n = 3$. Note that the joint curve is surrounded by n affine images of the quadrilateral [p, s, q, t] in P formed by the two reference triangles. These quadrilaterals can be thought of in three ways: they either extrapolate rightward from the rightmost two rows of Bézier points of $F_A(\Delta$pst$)$ to compute the leftmost two rows of Bézier points of $F_B(\Delta$qst$)$; or, vice versa, they extrapolate from right to left; or, more symmetrically, they constrain the two rows of Bézier points of both F_A and F_B that are nearest the knotline.

Raising the continuity to C^2 constrains the third row of Bézier points as well. If $n = 3$, there are two points in each third row, constrained as shown in Fig. 21.3. Note that this diagram includes two points, $f_{\{A,B\}}(p, q, s)$ and $f_{\{A,B\}}(p, q, t)$, that are not Bézier points of either surface patch.

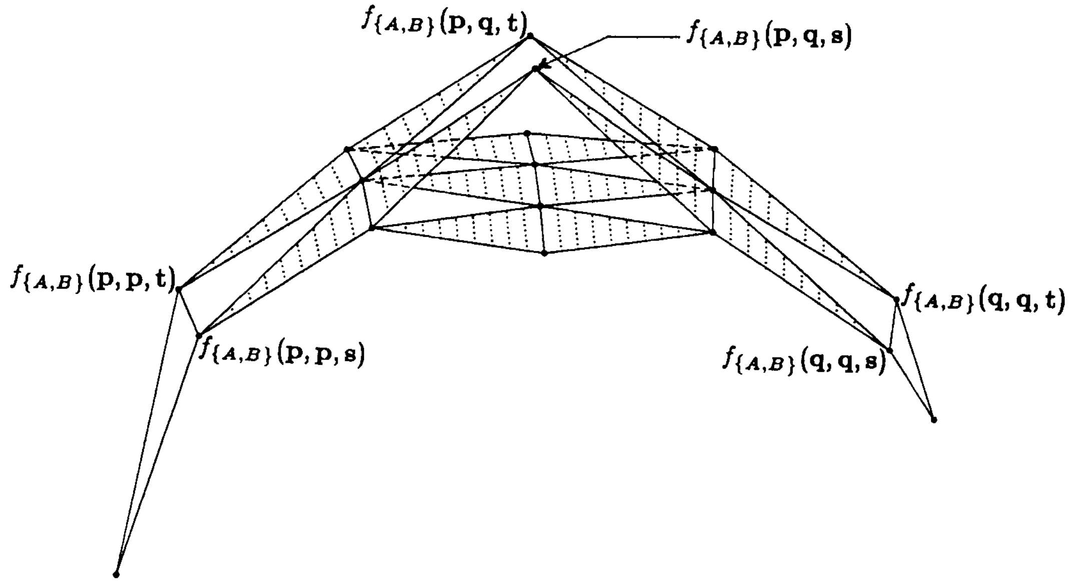

$f_{\{A,B\}}(\mathbf{p},\mathbf{q},\mathbf{t})$
$f_{\{A,B\}}(\mathbf{p},\mathbf{q},\mathbf{s})$
$f_{\{A,B\}}(\mathbf{p},\mathbf{p},\mathbf{t})$
$f_{\{A,B\}}(\mathbf{q},\mathbf{q},\mathbf{t})$
$f_{\{A,B\}}(\mathbf{p},\mathbf{p},\mathbf{s})$
$f_{\{A,B\}}(\mathbf{q},\mathbf{q},\mathbf{s})$

Fig. 21.3. The Bézier points of two cubic surfaces that join with C^2 continuity

For cubics, going all the way to C^3 continuity forces F_A and F_B to be identical; see Fig. 21.4. The extreme Bézier point $f_A(\mathbf{p},\mathbf{p},\mathbf{p})$ of $F_A(\Delta\mathbf{pst})$ is joined to the extreme Bézier point $f_B(\mathbf{q},\mathbf{q},\mathbf{q})$ of $F_B(\Delta\mathbf{qst})$ by a chain of three quadrilaterals, each an affine image of $[\mathbf{p},\mathbf{s},\mathbf{q},\mathbf{t}]$. The interior vertices along this chain are the blossom values $f_{\{A,B\}}(\mathbf{p},\mathbf{p},\mathbf{q})$ and $f_{\{A,B\}}(\mathbf{p},\mathbf{q},\mathbf{q})$. Fig. 21.4 is a two-dimensional de Casteljau Diagram with three shells. It looks rather different from the analogous diagram with two shells back in Fig. 3.4, but the difference is only skin deep. In Fig. 3.4, the newly added point $\mathbf{u}$ lies inside the original reference triangle $\Delta\mathbf{rst}$, so we are doing two-dimensional interpolation. In Fig. 21.4, say viewed from left to right, the new point $\mathbf{q}$ lies outside the original triangle $\Delta\mathbf{pst}$, so we are doing two-dimensional extrapolation. In the one-dimensional case, interpolation and extrapolation look pretty much the same, since they both involve three points along a line. In the two-dimensional case, the de Casteljau Algorithm is controlled by four points in a plane, and quite different geometries are possible.

Returning to the C^2 case in Fig. 21.3, we can interpret that figure as two overlapping instances of extrapolating de Casteljau Diagrams with two shells, one consisting of all points with at least one $\mathbf{s}$ in their labels, the other consisting of all points with at least one $\mathbf{t}$. In general, as claimed above, C^k continuity between n-ic surface patches is enforced by $n - k + 1$ overlapping two-dimensional de Casteljau Diagrams, each with k shells.

Now that we understand what a C^k joint between two surface patches looks like, the next step in building a spline surface is to choose a partition of the parameter plane P into convex polygonal regions and to select multiplicities for the various knotlines where those regions abut. Before starting that, we shall close this section with some general remarks about overloading the notation for the blossom of a spline

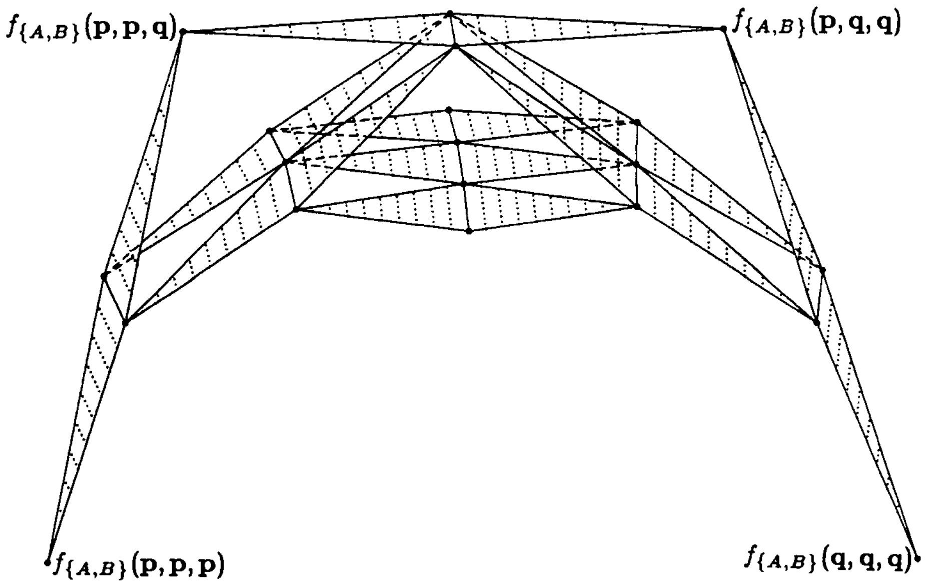

$f_{\{A,B\}}(\mathbf{p},\mathbf{p},\mathbf{q})$
$f_{\{A,B\}}(\mathbf{p},\mathbf{q},\mathbf{q})$
$f_{\{A,B\}}(\mathbf{p},\mathbf{p},\mathbf{p})$
$f_{\{A,B\}}(\mathbf{q},\mathbf{q},\mathbf{q})$

Fig. 21.4. The Bézier points of two cubic surfaces that join with C^3 continuity

surface.

So far, we have always indicated by means of an explicit subscript which region's blossom we meant to apply to a particular bag of argument points $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$. The expression $f_A(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ tells us to apply the blossom f_A of the patch F_A, while the expression $f_{\{A,B\}}(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ tells us to apply either f_A or f_B and asserts that the two results will be the same. Just as in the case of spline curves, an overloading rule is a convention for determining the correct subscript from the locations of the points $\mathbf{u}_i$. Let $\mathrm{Hull}(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ denote the closed convex hull of the points $\mathbf{u}_i$, and let $\overline{A}$ denote the closure of the region A. Generalizing the case of curves, the ***tame overloading convention*** considers a region A to be a ***candidate*** for defining the value $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ if and only if

$$\overline{A} \cap \mathrm{Hull}(\mathbf{u}_1, \ldots, \mathbf{u}_n) \neq \emptyset.$$

The term $f(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is then well-defined under the tame overloading convention if and only if all of the values $f_A(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ for all candidate regions A agree. The natural generalization of ***wild overloading*** to the case of surfaces considers a region A to be a candidate if and only if

$$\dim(\overline{A} \cap \mathrm{Hull}(\mathbf{u}_1, \ldots, \mathbf{u}_n)) = \dim(\mathrm{Hull}(\mathbf{u}_1, \ldots, \mathbf{u}_n)).$$

Deciding whether or not an overloaded term is well-defined can be a rather subtle question for surfaces. For example, consider the situation in Fig. 21.5. Suppose that we want to assemble a quadratic spline surface out of three patches, defined on the three triangular regions A, B, and C. We require F_A and F_B to meet with C^1 continuity along the line $\overline{\mathbf{qs}}$, and we require F_B and F_C to meet with C^1 continuity

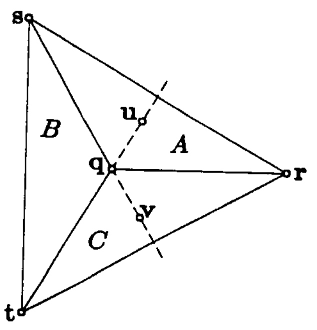


Fig. 21.5. A subtle geometry in the domain plane of a spline surface

along the line $\overline{\mathbf{qt}}$. But we don't place any continuity constraint on the joint between F_A and F_C along $\overline{\mathbf{qr}}$.

We should pause to verify that these constraints are actually honest. Note that, if we had demanded C^2 continuity between A and B and also C^2 continuity between B and C, then C^2 continuity between A and C would have been automatic; for quadratics, C^2 continuity means equality, and equality is transitive. Thus, one might worry that C^1 continuity across the (A,B) joint and also across the (B,C) joint might imply something about the (A,C) joint. Here is an example to show that the (A,C) joint can fail to be even C^0. Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be three distinct points in the object space Q. Define $f_A(\mathbf{r},\mathbf{r}) := \mathbf{x}$, $f_C(\mathbf{r},\mathbf{r}) := \mathbf{z}$, and let the other five Bézier points of $F_A(\Delta\mathbf{qrs})$ and $F_C(\Delta\mathbf{qtr})$, as well as all six Bézier points of $F_B(\Delta\mathbf{qst})$, be $\mathbf{y}$. The (A,B) joint in the resulting spline surface is C^1 continuous because C^1 continuity only depends on the two rows of Bézier points nearest the joint line and all those Bézier points have the common value $\mathbf{y}$. Similarly, the (B,C) joint is C^1 continuous. But the (A,C) joint is not even C^0 continuous, since $F_A(\mathbf{r}) = \mathbf{x} \neq \mathbf{z} = F_C(\mathbf{r})$.

Consider the overloaded term $f(\mathbf{u},\mathbf{v})$; is it well-defined? Since Hull$(\mathbf{u},\mathbf{v})$ is the line segment joining $\mathbf{u}$ to $\mathbf{v}$, the two candidate regions are A and C, under either tame or wild overloading. At first glance, it seems that there is no reason why the equality $f_A(\mathbf{u},\mathbf{v}) = f_C(\mathbf{u},\mathbf{v})$ should hold, since there is no continuity constraint at all on the joint between A and C. But note that the point $\mathbf{v}$ happens to lie on the line that separates A from B. Since that line is a knotline of multiplicity one and the argument bag $\{\mathbf{u},\mathbf{v}\}$ includes one point along that line, we deduce from Prop. 21.1 that $f_A(\mathbf{u},\mathbf{v}) = f_B(\mathbf{u},\mathbf{v})$. Symmetrically, since $\mathbf{u}$ lies on the simple knotline that separates B from C, we deduce that $f_B(\mathbf{u},\mathbf{v}) = f_C(\mathbf{u},\mathbf{v})$. Therefore, $f_A(\mathbf{u},\mathbf{v}) = f_C(\mathbf{u},\mathbf{v})$; the overloaded term $f(\mathbf{u},\mathbf{v})$ is well-defined, but for a rather subtle reason.

22. Spline surfaces with rectangular patches

The two most symmetric ways to partition the parameter plane P into convex polygons are to use congruent squares or congruent equilateral triangles. In this section, we shall consider the case of squares.

Technically, the parameter plane P is just an affine space, so the concept of a square isn't defined. What we shall really do is to divide P into a lattice of congruent parallelograms. Having done so, it doesn't hurt anything to choose two adjacent sides of one of those parallelograms as orthonormal basis vectors, thus giving P the structure of a Euclidean plane divided into squares. Doing so has the advantage of making the symmetries more apparent.

We hope to associate a polynomial surface of degree N with each square in such a way that the resulting patches fit together smoothly. To keep things as symmetric as possible, we choose to enforce the same level of continuity across each joint between two adjacent squares. What level of continuity should we demand? In the case of curves, we can ask for C^{n-1} continuity between two segments of degree n and still leave sufficient flexibility. But for surfaces, it turns out that we can't do nearly that well; we can only get C^k continuity for k some fraction of N.

To see why, let us first clarify our goals somewhat. The higher the level of continuity that we demand, the fewer spline surfaces there will be. In computer-aided geometric design, it is important that we stop while the splines still have *local flexibility*. That is, we want to be able to change a small region of a design without affecting the bulk of the surface. The spline surface techniques that are most important in practice are those that achieve, for a given degree, the highest possible continuity that still allows local flexibility.

Note that local flexibility is not as stringent a demand as local control. A method of drawing splines has ***local control*** if each control point of the method affects only a small region of the design. For example, consider C^2 cubic spline curves with uniform knot spacing, say knots $\{\bar{t}_i\}$ at the integers $t_i := i$. One method of controlling such curves involves specifying their de Boor points. This method has local control: moving the de Boor point $f_{(\overline{i-2},\overline{i+2})}(\overline{i-1},\bar{i},\overline{i+1})$ affects only the four adjacent segments with domains $(\overline{i-2},\overline{i-1})$ through $(\overline{i+1},\overline{i+2})$. Another method of controlling the same curves involves specifying the locations of the joints. This method does not have local control: moving one joint affects the entire spline curve, although the effect dies off exponentially. Fig. 22.1 shows one example of this. The straight line on top is the cubic spline that interpolates the joints $\langle i,0\rangle$, while the spline curve in the middle is what results if we lift the joint at $\langle 0,0\rangle$ up to $\langle 0,1\rangle$. (The de Boor points of the middle spline are at $\langle i,\sqrt{3}(\sqrt{3}-2)^{|i|}\rangle$.) Despite their lack of local control, interpolating cubic splines do have local flexibility. If we want to make a local modification, we just have to remember, when we perturb a joint by some vector ξ in Q_0, that we must also perturb its two neighbors by $\xi/4$. The bottom spline curve in Fig. 22.1 shows what happens if we also lift the joints at $\langle \pm 1,0\rangle$ up to $\langle \pm 1,\frac{1}{4}\rangle$. (The de Boor points of this spline are at $\langle i,3\delta_{i0}/2\rangle$.) In fact, local control and local flexibility are really properties of different things. Local flexibility is a property of the spline space itself, while local control is a property of a scheme for choosing one spline out of the space.

How high a level k of C^k continuity can we demand at the joints between square patches of degree N if we want to preserve local flexibility in the resulting surface? To ask the same question a different way, how high a degree N is needed to allow C^k splines with square N-ic patches to be locally flexible? We shall first compute a lower bound on N, and then show that that lower bound is actually the correct

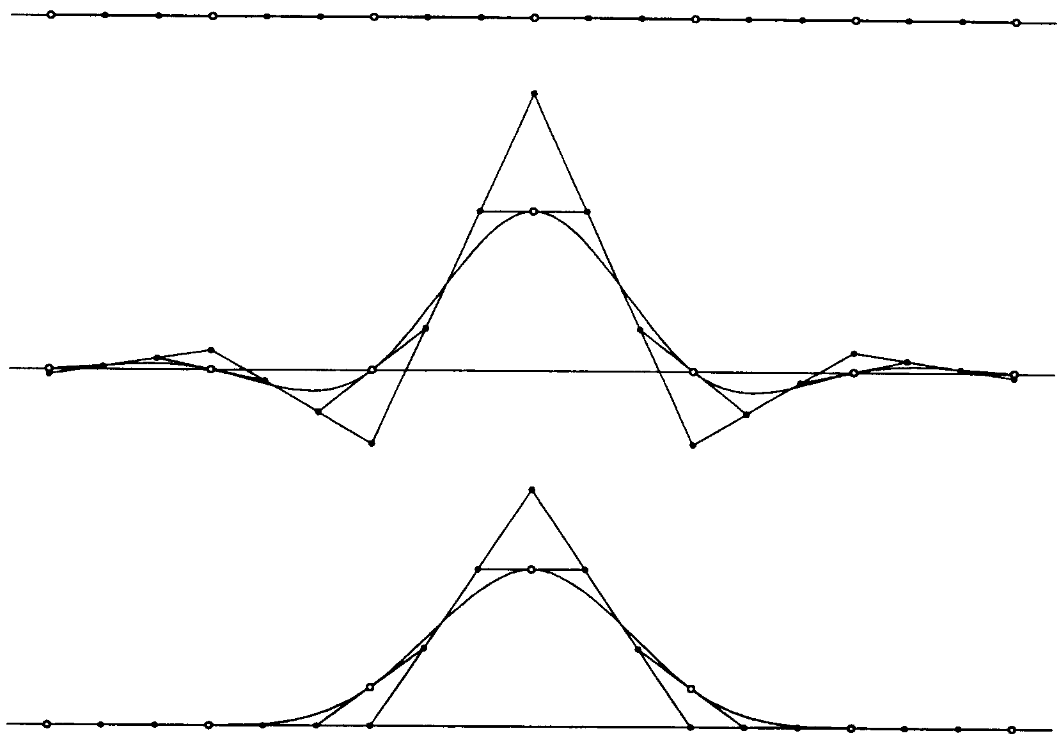


Fig. 22.1. Two different ways of perturbing a cubic spline curve

answer.

Proposition 22.2. *Spline surfaces built by assembling square patches of degree N with C^k continuity at the joints can't possibly have local flexibility if $N \le 2k+1$. In the borderline case where $N = 2k+2$, the resulting spline surfaces have at most one control point's worth of freedom per internal patch.*

Proof. Let $F: P \to Q$ be a spline surface with square patches of degree N that join with C^k continuity. We first want to show that $N \le 2k+1$ precludes local flexibility.

Imagine determining the square patches of F in raster-scan order, say from left to right and from bottom to top. In this scan, suppose that we have just arrived at the new patch F_A, whose domain is the square region A shown in Fig. 22.3. We can assume that the square patches F_B and F_D associated with the regions B and D have already been determined. The demand for C^k continuity between A and its neighbors B and D constrains our choice of F_A. We shall show that, if $N \le 2k+1$, then F_A is completely determined by F_B and F_D. Since every square patch in the interior of the spline's domain is just like F_A, this is enough to prove that C^k spline surfaces for $N \le 2k+1$ can't have local flexibility.

Suppose, then, that $N \le 2k+1$; we want to show that F_A is completely determined by F_B and F_D. Consider the Bézier points of F_A with respect to the reference triangle $\Delta\mathbf{xyz}$, which are the blossom values $f_A^{\circledast}(\mathbf{x}^i\mathbf{y}^{N-i-j}\mathbf{z}^j)$ for $i+j \le N$. Since F_A joins F_B with C^k continuity along the knotline connecting $\mathbf{x}$ and $\mathbf{y}$,

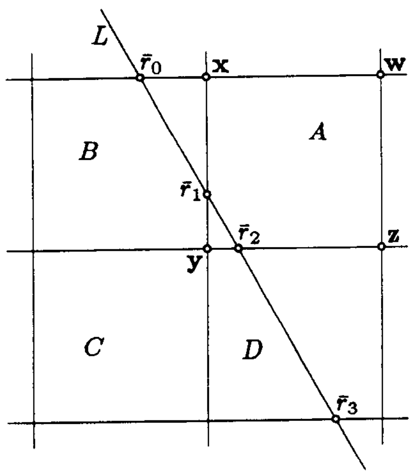


Fig. 22.3. Determining the next square patch of a spline surface

we deduce from Prop. 21.1 that $f_A^{\otimes}(\mathbf{x}^i\mathbf{y}^{N-i-j}\mathbf{z}^j) = f_B^{\otimes}(\mathbf{x}^i\mathbf{y}^{N-i-j}\mathbf{z}^j)$ for $j \le k$. Similarly, we deduce that $f_A^{\otimes}(\mathbf{x}^i\mathbf{y}^{N-i-j}\mathbf{z}^j) = f_D^{\otimes}(\mathbf{x}^i\mathbf{y}^{N-i-j}\mathbf{z}^j)$ for $i \le k$. But we have $i + j \le N \le 2k + 1$; hence, every Bézier point of F_A must satisfy either $i \le k$ or $j \le k$. Therefore, every Bézier point of F_A is determined.

Note, in fact, that some Bézier points of F_A are determined both by F_B and by F_D. If F_B and F_D disagree about any such Bézier point, there won't be any legal choice of F_A. For example, this will certainly be the case if F_B and F_D don't agree with each other to kth order at the point $\mathbf{y}$. But the possibility that F_A might be over-constrained doesn't affect the validity of our argument, because we are giving only an upper bound on the amount of freedom in F_A.

It remains to bound the amount of choice we have in determining F_A in the borderline case when $N = 2k+2$. In this case, there will be precisely one Bézier point of F_A that is not determined: $f_A^{\otimes}(\mathbf{x}^{k+1}\mathbf{z}^{k+1})$. Thus, a square-patch spline surface of degree $2k + 2$ with C^k continuity has at most one point's worth of freedom per interior patch. □

The phrase "one point's worth of freedom" is rather clumsy. In the future, we shall say simply "one degree of freedom," with the understanding that each degree of freedom means the free choice of a point in the object space Q.

There is a trickier way to prove Prop. 22.2, due to C. de Boor and R. DeVore [12], which we might call ***corner-sectioning***. We shall discuss corner-sectioning both because of its intrinsic interest and because we will want to use it in earnest when discussing triangular-patch splines in Section 23.

Draw a line L that cuts off a corner of A, and let $\bar r_0$ through $\bar r_3$ be the points where L crosses the boundaries of the squares B, A, and D, as shown in Fig. 22.3. If we restrict the spline surface F to the line L, we will get an N-ic spline curve $G: L \to Q$, a section of the surface F. Furthermore, if $m := N - k$ is the multiplicity of each knotline, the knot sequence of G is given by

$$(\ldots,\underbrace{\bar{r}_0,\ldots,\bar{r}_0}_{m},\underbrace{\bar{r}_1,\ldots,\bar{r}_1}_{m},\underbrace{\bar{r}_2,\ldots,\bar{r}_2}_{m},\underbrace{\bar{r}_3,\ldots,\bar{r}_3}_{m},\ldots).$$

A de Boor point of G has the form $g_{(\bar{t}_i,\bar{t}_{i+N+1})}(\bar{t}_{i+1},\ldots,\bar{t}_{i+N})$ where $\{\bar{t}_i\}$ is an indexing of this knot sequence. Imagine choosing a substring of length N from the knot sequence of G. If that substring includes the first occurrence of $\bar{r}_1$, then the validity interval of the corresponding de Boor point will include the interval $(\bar{r}_0,\bar{r}_1)$. Such a de Boor point can be computed from the behavior of G over the interval $(\bar{r}_0,\bar{r}_1)$, and is hence determined by F_B. Similarly, if the substring of length N includes the last occurrence of $\bar{r}_2$, the validity interval of the corresponding de Boor point will include $(\bar{r}_2,\bar{r}_3)$ and hence that de Boor point will be determined by F_D. If $N \le 2k+1$, and hence $N \ge 2m-1$, there isn't room to fit a substring of N consecutive knots of G in between the first $\bar{r}_1$ and the last $\bar{r}_2$, so every de Boor point of G that influences the segment $G([\bar{r}_2,\bar{r}_3])$ will be determined, either by F_B or by F_D. Hence, the segment $G([\bar{r}_2,\bar{r}_3])$ will be determined. Since this same argument applies to any line L that cuts off a corner of A, we deduce that, when $N \le 2k+1$, the values of F_A throughout the triangle $\Delta\mathbf{xyz}$ are determined by F_B and F_D. Since F_A is given by polynomials, this implies that all values of F_A are determined.

The same corner-sectioning idea can also be used to show that F_A has only one degree of freedom in the borderline case when $N = 2k+2$, and hence $N = 2m-2$. In this case, there is exactly one substring of N knots of the spline curve G that fits between the first $\bar{r}_1$ and the last $\bar{r}_2$. This means that the section G has precisely one de Boor point that is free to vary. Choose a point $\mathbf{q}$ in the interior of $\Delta\mathbf{xyz}$, and consider the various corner-sectioning lines L that pass through $\mathbf{q}$, which form a double wedge with vertex $\mathbf{q}$. The value $F_A(\mathbf{q})$ in Q is enough to determine the one free de Boor point of the section G associated with each line L. Thus, the value $F_A(\mathbf{q})$ determines all of the values of F_A in the double wedge, which is enough to determine F_A completely.

As part of our quest for simplicity, we shall focus in this paper on the borderline case in Prop. 22.2, in which the degree $N = 2k+2$ is just high enough to avoid precluding local flexibility. For notational convenience, let n be a nonnegative integer and set $N := 2n$ and $k := n-1$. The following proposition shows that locally flexible square-patch spline surfaces with those borderline parameters really do exist, thus demonstrating that the bound in Prop. 22.2 is tight.

Proposition 22.4. *There is a method for drawing square-patch spline surfaces of degree $N = 2n$ with C^{n-1} continuity that involves specifying one control point per interior patch. (If n is odd, it is more natural to think of the control points as associated with vertices rather than with patches.) In this method, each control point influences $(n+1)^2$ patches, which means that each patch is influenced by $(n+1)^2$ control points.*

Proof sketch. We shall describe this method by indicating how its blending functions can be defined, where a *blending function* is the real-valued spline function that gives the influence of a particular control point on the location of the eventual surface.

In the base case $n = 0$, we want to put together square constant patches with C^{-1} continuity, that is, with no continuity at all. It is easy to see how to do this with one control point per patch: each control point specifies the value of the entire associated patch. In particular, each blending function is identically equal to 1 on one square and is 0 everywhere else; that is, each blending function is a translate of the function

$$B_0(\mathbf{u}) = \begin{cases} 1 & \text{if } \mathbf{u} \in [0,1] \times [0,1] \\ 0 & \text{otherwise.} \end{cases}$$

Note that all of the translates of $B_0(\mathbf{u})$ together form a partition of unity (if we don't worry about what happens on the boundaries of the squares).

For larger values of n, we can produce blending functions by convolving $B_0(\mathbf{u})$ with itself some number of times. If G and H are real-valued functions defined on P, the ***convolution*** $G * H$ is given by

$$(G * H)(\mathbf{u}) := \iint_P F(\mathbf{v})G(\mathbf{u} - \mathbf{v})\, d\mathbf{v}.$$

Let $B_n(\mathbf{u}) := (B_{n-1} * B_0)(\mathbf{u})$ for $n \geq 1$. For example, $B_1(\mathbf{u})$ contains four nonzero patches, each of which is a hyperbolic paraboloid:

$$B_1(\mathbf{u}) = B_1(\langle u,v\rangle) = \begin{cases} uv & \text{if } \mathbf{u} \in [0,1] \times [0,1] \\ (2-u)v & \text{if } \mathbf{u} \in [1,2] \times [0,1] \\ u(2-v) & \text{if } \mathbf{u} \in [0,1] \times [1,2] \\ (2-u)(2-v) & \text{if } \mathbf{u} \in [1,2] \times [1,2] \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $B_n(\mathbf{u})$ will be nonzero precisely on the $(n+1)^2$ unit squares that form the square $[0,n] \times [0,n]$. It is also true, though we won't verify it here, that each convolution of $B_0(\mathbf{u})$ with itself raises the degree by 2 and the level of continuity by 1. Thus, the blending function $B_n(\mathbf{u})$ is a real-valued spline function with square patches of degree $2n$ and with C^{n-1} continuity. For even n, the maximum of $B_n(\mathbf{u})$ is located at the center of a patch; for odd n, the maximum occurs at a vertex where four patches join.

The convolution process preserves the property that the various translates of $B_n(\mathbf{u})$ form a partition of unity. Hence, we can use those translates as blending functions. We arrive at a method for controlling C^{n-1} spline surfaces of degree $2n$ with one control point per patch. In this method, each patch of the resulting surface is influenced by $(n+1)^2$ control points. □

A general polynomial surface of degree $2n$ has $\binom{2n+2}{2} = (n+1)(2n+1)$ Bézier points with respect to any reference triangle. The surface patches that arise in the spline surfaces of Prop. 22.4 are influenced by only $(n+1)^2$ control points. Hence, there must be $n(n+1)$ constraints of some sort that hold for those surfaces. It turns out that we can think of these constraints as constraints on their high-order derivatives, as shown in the following lemmas.

If $F\colon P \to Q$ is a polynomial surface and ξ is a vector on P, we can construct a new surface whose values are vectors on Q by associating with each point $\mathbf{u}$ in P

the derivative $dF_{\mathbf{u}}(\xi) = D_\xi F(\mathbf{u})$ of F in the ξ direction at $\mathbf{u}$. Since we are holding the direction ξ fixed while varying $\mathbf{u}$, the notation of directional derivatives is more natural here; we shall use the symbol $D_\xi F$ to denote this derived surface.

Lemma 22.5. *If $F: P \to Q$ and $G: P \to Q$ are two polynomial surfaces that meet with C^0 continuity along a line L in P, and if ξ is a vector on P parallel to L, then the derived surfaces $D_\xi F$ and $D_\xi G$ also meet with C^0 continuity along L.*

Proof. If $\mathbf{u}$ is a point on L, we can compute $D_\xi F(\mathbf{u})$ by evaluating F at $\mathbf{u}$ and at points near $\mathbf{u}$ along L. Since F and G take on the same values everywhere along L, we must get the same answers when we compute $D_\xi F(\mathbf{u})$ and $D_\xi G(\mathbf{u})$; hence, $D_\xi F$ and $D_\xi G$ also agree along L. □

Lemma 22.6. *In the situation of Lemma 22.5, if F and G meet with C^k continuity along L, then the derived surfaces $D_\xi F$ and $D_\xi G$ meet with C^k continuity along L also.*

Proof. Let η_1 through η_k be arbitrary vectors on P. If the surfaces F and G meet with C^k continuity along L, then the derived surfaces $D_{\eta_1} \cdots D_{\eta_k} F$ and $D_{\eta_1} \cdots D_{\eta_k} G$ must meet with C^0 continuity along L. If we differentiate one more time, in the ξ direction, Lemma 22.5 tells us that the resulting derived surfaces $D_\xi D_{\eta_1} \cdots D_{\eta_k} F$ and $D_\xi D_{\eta_1} \cdots D_{\eta_k} G$ will also meet with C^0 continuity along L. Since the various directional derivatives commute with each other, this implies that $D_{\eta_1} \cdots D_{\eta_k} D_\xi F$ and $D_{\eta_1} \cdots D_{\eta_k} D_\xi G$ meet with C^0 continuity. Since η_1 through η_k were arbitrary, we conclude that $D_\xi F$ and $D_\xi G$ must meet with C^k continuity along L. □

Lemma 22.7. *If $F: P \to Q$ and $G: P \to Q$ are two polynomial surfaces of degree $2n$ that meet with C^{n-1} continuity along a line L, and if ξ is a vector on P parallel to L, then the derived surfaces $(D_\xi)^{n+1}F$ and $(D_\xi)^{n+1}G$ must be identical, that is, they must agree everywhere.*

Proof. If we apply Lemma 22.6 repeatedly, a total of $n+1$ times, we can deduce that the derived surfaces $(D_\xi)^{n+1}F$ and $(D_\xi)^{n+1}G$ must meet with C^{n-1} continuity along L. But these derived surfaces are polynomial surfaces of degree at most $n-1$; if they agree at any point up through order $n-1$, they must agree everywhere, that is, they must be identical. □

Suppose, then, that we build a spline surface F with C^{n-1} continuity out of square patches F_A of degree $2n$. Lemma 22.7 will apply to any two adjacent surface patches. In particular, let ξ be a horizontal vector on P and let η be a vertical vector. All of the surface patches F_A for A in a single horizontal stripe must share the same value of $(D_\eta)^{n+1}F_A$, while all of the patches F_A for A in any vertical stripe must share the same value of $(D_\xi)^{n+1}F_A$. Thus, these high-order derivatives of our surface patches are not locally flexible. We have to make a global choice for each horizontal stripe of a single polynomial function of degree $n-1$ that will give $(D_\eta)^{n+1}F_A$ for all A in that stripe, and similarly for vertical stripes.

How shall we choose, for each stripe, what the derived surfaces shall be along that stripe? Since we can't vary this choice locally, the simplest course of action, which we shall adopt, is not to vary it at all. In particular, we shall restrict ourselves to square surface patches F_A for which $(D_\xi)^{n+1}F_A$ and $(D_\eta)^{n+1}F_A$ are both identically zero. In defense of this choice, we claim that the square patches F_A used by the spline-drawing method of Prop. 22.4 do satisfy the constraints $(D_\xi)^{n+1}F_A = (D_\eta)^{n+1}F_A = 0$. To see why, note that it is enough to show that the blending function $B_n(\mathrm{u})$ satisfies those constraints. And $B_n(\mathrm{u})$ must satisfy them because it has compact support: if we move out along any stripe far enough, we get to patches in $B_n(\mathrm{u})$ that are identically zero, and hence certainly have their derivatives of all orders equal to zero.

This explains why each internal patch F_A in one of the spline surfaces of Prop. 22.4 has only $(n+1)^2$ degrees of freedom in it, instead of the $(2n+1)(n+1)$ that we would naively expect. The derived surface $(D_\xi)^{n+1}F_A$, being a surface of degree $n-1$, has $\binom{n+1}{2}$ degrees of freedom in it, and the constraint $(D_\xi)^{n+1}F_A = 0$ eliminates that freedom. Considering the two directions of stripes, we have accounted for all $2\binom{n+1}{2} = n(n+1)$ missing degrees of freedom.

Our decision to adopt the constraints $(D_\xi)^{n+1}F_A = (D_\eta)^{n+1}F_A = 0$ has a very important consequence: it means that we are actually building spline surfaces out of bipolynomial patches of degree $(n;n)$. In particular, let F_A be a polynomial surface of degree $2n$ that satisfies the constraints $(D_\xi)^{n+1}F_A = (D_\eta)^{n+1}F_A = 0$. If we view the parameter plane P of F_A as the product $U \times V$ of two affine lines, one in the ξ direction and the other in the η direction, the resulting function $F_A\colon U \times V \to Q$ will be bipolynomial of degree $(n;n)$.

With this insight, we can generalize our results somewhat and also obviate a lot of the earlier work in this section by defining ***bipolynomial spline surfaces*** as follows. Let m and n be nonnegative integers. Let $\{\grave{s}_i\}$ and $\{\acute{t}_j\}$ be two knot sequences, located in the affine lines U and V respectively. We can form a bipolynomial spline surface $F\colon U \times V \to Q$ of degree $(m;n)$ by assigning arbitrary values to the de Boor points

$$\mathbf{x}_{ij} := f_{(\grave{s}_i,\grave{s}_{i+m+1})\times(\acute{t}_j,\acute{t}_{j+n+1})}(\grave{s}_{i+1},\ldots,\grave{s}_{i+m};\acute{t}_{j+1},\ldots,\acute{t}_{j+n}).$$

Note that these de Boor points have ***validity rectangles*** rather than validity intervals. The patches of the resulting spline surface have domain rectangles of the form $R_{kl} := (\grave{s}_k,\grave{s}_{k+1}) \times (\acute{t}_l,\acute{t}_{l+1})$, where k and l are chosen so that $\grave{s}_k < \grave{s}_{k+1}$ and $\acute{t}_l < \acute{t}_{l+1}$. The patch defined on the rectangle R_{kl} is determined by the $(m+1)(n+1)$ de Boor points $\{\mathbf{x}_{ij}\}$ for i in $[k-m,k]$ and j in $[l-n,l]$. Two patches that are adjacent in the U direction join with C^{m-r} continuity, where r is the multiplicity of the knot $\grave{s}_i$ that divides them. Similarly, two patches adjacent in the V direction join with C^{n-r} continuity, where r is the multiplicity of the dividing knot $\acute{t}_j$.

The availability of the concept of a bipolynomial surface makes square-patch spline surfaces much easier to understand than they otherwise would be. Let us recapitulate the developments in this section. If we assemble square patches of polynomial surfaces together with continuity constraints on the joints, we found that we perforce end up with global constraints relating the high-order derivatives of various patches in the directions parallel to the edges of the squares. Since we weren't interested in modifying the surface globally, we chose to satisfy these global constraints trivially, by demanding that the relevant derivatives be zero in every patch. The critical fact is that precisely those derivative constraints are built into

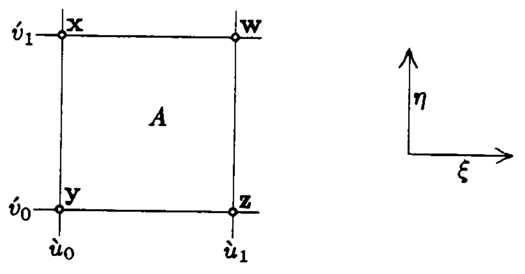


Fig. 22.8. The domain A of one square patch of a spline surface

the very concept of a bipolynomial surface. Thus, a patch F_A of a square-patch spline surface is not a general polynomial surface of degree $2n$; but it is a general bipolynomial surface of degree $(n; n)$.

To be explicit, consider the case $n = 1$, in which each patch F_A is a hyperbolic paraboloid. We can view a hyperbolic paraboloid as an instance of a polynomial surface of degree two, and blossom it in that way; let $g: P^2 \to Q$ be the resulting 2-blossom. The 2-blossom g of F_A gives us six Bézier points with respect to any reference triangle, but they have to satisfy certain constraints. For definiteness, let us choose the reference triangle $\Delta\mathbf{xyz}$, as shown in Fig. 22.8. The constraint $D_\xi D_\xi F_A = 0$ corresponds to the condition $g(\mathbf{y},\mathbf{y}) - 2g(\mathbf{y},\mathbf{z}) + g(\mathbf{z},\mathbf{z}) = 0$. Similarly, the constraint $D_\eta D_\eta F_A = 0$ correponds to the condition $g(\mathbf{y},\mathbf{y}) - 2g(\mathbf{x},\mathbf{y}) + g(\mathbf{x},\mathbf{x}) = 0$. The bipolynomial point of view neatly eliminates the redundancies and constraints. The $(1;1)$-blossom f of F_A gives F_A the four unconstrained Bézier points $f(\grave{u}_0; \acute{v}_0)$, $f(\grave{u}_0; \acute{v}_1)$, $f(\grave{u}_1; \acute{v}_0)$, and $f(\grave{u}_1; \acute{v}_1)$. Note, in fact, that the constrained 2-blossom g is precisely the result of applying degree-joining to the unconstrained $(1;1)$-blossom f. Working with g instead of f makes things more complicated because the complex degree-adjusting formulas of Part D become relevant.

23. Spline surfaces with triangular patches

In this section, we shall study what happens if we attempt to build a spline surface based on a partition of the parameter plane P into equilateral triangles. On the positive side, we will find that we can get a higher level of continuity for the same degree of surface. On the negative side, after we have figured out what the resulting derivative constraints are, we won't find an analog of the bipolynomial theory lying around, waiting to come to our rescue by having precisely those derivative constraints built in.

Let N be the degree of the triangular surface patches. To keep things symmetric, we shall enforce the same C^k continuity constraint on all the joints between pairs of adjacent patches. Just as in the case of square patches, our first result will warn us that setting the degree too low for a fixed continuity precludes local flexibility. By the way, note that a triangular grid has twice as many regions as vertices: each vertex is both the highest point in some upward-pointing triangle and also the lowest point in some downward-pointing triangle.

Proposition 23.1. *Spline surfaces built by assembling triangular patches of sur-*

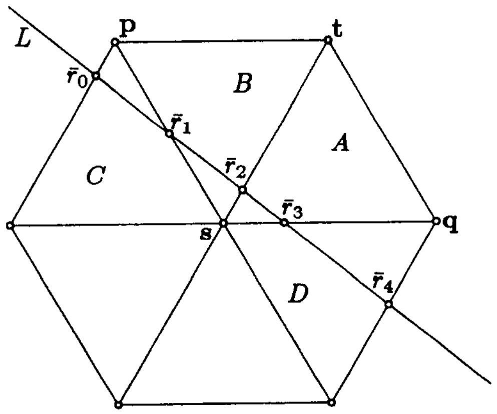


Fig. 23.2. Determining the next two triangular patches of a spline surface

faces of degree N with C^k continuity across the joints can't possibly have local flexibility if $2N \leq 3k+1$. In the borderline case where $2N = 3k+2$, the resulting spline surfaces have at most one degree of freedom per internal vertex, that is, at most one control point's worth of freedom for every two internal patches.

Proof. Let $F: P \to Q$ be a spline surface with triangular patches of degree N that join with C^k continuity. Imagine determining the patches of F in pairs, where each pair consists of an upward-pointing triangle and its left-hand downward-pointing neighbor, and where we take the pairs in left-to-right, bottom-to-top raster-scan order. In this scan, suppose that we have arrived at the pair of patches F_A and F_B, shown in Fig. 23.2. We may assume that the patches F_C and F_D have already been determined. If F_A and F_B are completely determined by F_C, F_D, and the continuity constraints across the three joints (D, A), (A, B), and (B, C), then there is no chance for local flexibility.

We can analyze whether or not F_A and F_B are determined in two different ways: either by reasoning about Bézier points or by using the corner-sectioning technique of de Boor and DeVore [12]. We shall use corner-sectioning, since it is easier.

Draw a line L that cuts off a corner of both A and B, and let $\bar{r}_0$ through $\bar{r}_4$ be the points where L crosses knotlines as shown in Fig. 23.2. Restricting the surface F to the line L gives an N-ic spline curve G, whose knot sequence has the form

$$(\ldots, \underbrace{\bar{r}_0, \ldots, \bar{r}_0}_{m}, \underbrace{\bar{r}_1, \ldots, \bar{r}_1}_{m}, \underbrace{\bar{r}_2, \ldots, \bar{r}_2}_{m}, \underbrace{\bar{r}_3, \ldots, \bar{r}_3}_{m}, \underbrace{\bar{r}_4, \ldots, \bar{r}_4}_{m}, \ldots),$$

where $m := N - k$ is the multiplicity of each knotline. The inequality $2N \leq 3k+1$ is equivalent to $N \geq 3m-1$. If this holds, a substring of N knots that is far enough to the right to avoid the leftmost copy of $\bar{r}_1$ will have to include the rightmost copy of $\bar{r}_3$. Hence, every de Boor point of G that influences either of the two segments $G([\bar{r}_1, \bar{r}_2])$ or $G([\bar{r}_2, \bar{r}_3])$ will be determined, either by F_C or by F_D. Wiggling the line L around a bit, we conclude that all of F_A and F_B will be determined.

In the borderline case where $2N = 3k+2$, we have $N = 3m-2$. Thus, the spline curve G will have precisely one free de Boor point. By rotating the line L about a fixed point $\mathbf{q}$ in $\triangle\mathbf{psq}$, as in the discussion following Prop. 22.2, we conclude that the patches F_A and F_B together have at most one degree of freedom in them. □

Exercise 23.3. Reprove Prop. 23.1 by explicit reasoning about the Bézier points of F_A and F_B. This is harder than one might expect because there is no single reference triangle that has two vertices on each of the three relevant knotlines. One way to deal with this complication is to consider the entire tetrahedra of blossom values $f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)$ and $f_B^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)$ for $i+j+l+m=N$, the faces of which are triangular arrays of Bézier points. Note that

$$f_A^{\otimes}(\mathbf{s}^{i+1}\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)+f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^{j+1}\mathbf{p}^l\mathbf{q}^m)=f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^{l+1}\mathbf{q}^m)+f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^{m+1})$$

whenever $i+j+l+m=N-1$, since $\mathbf{s}+\mathbf{t}=\mathbf{p}+\mathbf{q}$; and the same holds for f_B. The constraint of C^k continuity between F_A and F_D means that the values $f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)$ are determined for $j+l\le k$. Similarly, C^k continuity between F_B and F_C determines the values $f_B^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)$ for $j+m\le k$. Finally, C^k continuity between F_A and F_B implies the equality $f_A^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)=f_B^{\otimes}(\mathbf{s}^i\mathbf{t}^j\mathbf{p}^l\mathbf{q}^m)$ for $l+m\le k$. If $2N\le 3k+1$, show that these conditions are enough to determine both tetrahedra entirely. If $2N=3k+2$, show that there is precisely one degree of freedom in the two tetrahedra.

The bound in Prop. 23.1 suggests that using triangular patches will allow us to have smoother surfaces for the same degree, since k is roughly $2N/3$ instead of roughly $N/2$. Continuing our policy of pursuing simplicity, we shall focus on the borderline case where $N:=3n+1$ and $k:=2n$ for some nonnegative integer n. The following proposition states that triangular-patch spline surfaces with these borderline parameters do exist. The convolutions used in the proof are frequently credited to M. A. Sabin with a 1977 reference [36], but they were also known, in 1971, to P. O. Frederickson [24].

Proposition 23.4. *There is a method for drawing triangular-patch spline surfaces of degree $N=3n+1$ with C^{2n} continuity that involves specifying one control point per interior vertex. In this method, each patch is influenced by $3(n+1)^2$ control points, while each control point influences $3(n+1)^2$ upward-pointing patches and also $3(n+1)^2$ downward-pointing ones.*

Proof sketch. In the base case $n=0$, we want to assemble affine patches with C^0 continuity. It is easy to see how to do this with one control point per vertex: define each patch by performing a two-dimensional linear interpolation between the control points associated with its three corners. Let $B_0(\mathbf{u})$ be the corresponding blending function, which gives the influence of a specified control point. The graph of $B_0(\mathbf{u})$ is a pyramid of height 1 with a regular hexagon as its base. Note that, if we translate one copy of this pyramid to each vertex in P, the resulting functions form a partition of unity.

For larger values of n, we construct a blending function $B_n(\mathbf{u})$ by convolving $B_0(\mathbf{u})$ with itself n times. It is convenient to choose the distance scale in the plane P so that the volume of the hexagonal pyramid is one, that is, so that $\iint_P B_0(\mathbf{u})\,d\mathbf{u}=1$, which corresponds to making each equilateral triangle have area $1/2$ and hence side-length s, where $s^4=4/3$. If we do this, each blending function $B_n(\mathbf{u})$ defined by

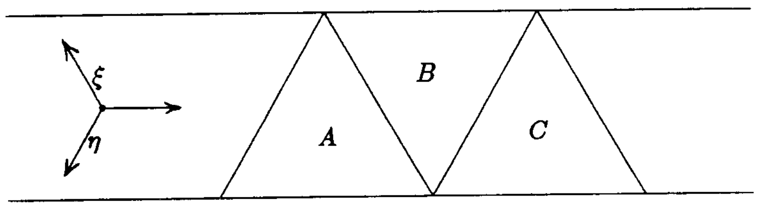


Fig. 23.5. A stripe in the domain plane of a triangular-patch spline surface

$B_n(\mathbf{u}) := (B_{n-1} * B_0)(\mathbf{u})$ will have the property that its translates form a partition of unity. It is easy to see that the function $B_n(\mathbf{u})$ will be nonzero precisely for $\mathbf{u}$ in a regular hexagon of side-length $(n+1)s$, which consists of $6(n+1)^2$ triangles, half pointing up and the other half down. The harder part, which we omit, is to show that each convolution with $B_0(\mathbf{u})$ raises the degree of the blending function by 3 and raises the level of continuity of its joints by 2. After n such convolutions, the function $B_n(\mathbf{u})$ has degree $3n+1$ and C^{2n} continuity. Spline surfaces built using the translates of $B_n(\mathbf{u})$ as the blending functions inherit these same properties. □

A general polynomial surface of degree $3n+1$ has $\binom{3n+3}{2}$ Bézier points, while the surface patches that arise in the splines of Prop. 23.4 are controlled by only $3(n+1)^2$ points. This means that the latter surface patches must satisfy $\binom{3n+3}{2} - 3(n+1)^2 = 3\binom{n+1}{2}$ constraints. Just as in the square-patch case, those constraints turn out to involve the high-order derivatives of the surface.

Let F_A and F_B be two adjacent surface patches of a triangular-patch spline surface, and let ξ and η be vectors parallel to the edges of the triangles as shown in Fig. 23.5. Consider the derived surfaces $(D_\eta)^{n+1}F_A$ and $(D_\eta)^{n+1}F_B$. Since F_A and F_B are of degree $3n+1$ and meet with C^{2n} continuity, these derived surfaces are of degree $2n$ and meet with C^{n-1} continuity. Therefore, we can apply Lemma 22.7 to deduce that the doubly derived surfaces $(D_\xi)^{n+1}(D_\eta)^{n+1}F_A$ and $(D_\xi)^{n+1}(D_\eta)^{n+1}F_B$ must be identical. Comparing F_B with F_C, where C is the next region along the stripe, as shown in Fig. 23.5, we can apply the same argument with the roles of ξ and η reversed to show that the doubly derived surfaces $(D_\xi)^{n+1}(D_\eta)^{n+1}F_B$ and $(D_\xi)^{n+1}(D_\eta)^{n+1}F_C$ are also identical. We conclude that all of the triangular patches F_A in any horizontal stripe of the spline surface must share a single common value of the $(n-1)$-ic function $(D_\xi)^{n+1}(D_\eta)^{n+1}F_A$. That is, these high-order derivatives are not locally flexible.

The same arguments apply to the stripes in all three directions. Let α, β, and γ be vectors on P that are parallel to the three axes of the triangular grid; it is convenient to arrange that $\alpha+\beta+\gamma=0$. For any triangular-patch surface F, the derived surface $(D_\alpha)^{n+1}(D_\beta)^{n+1}F_A$ is constant for triangles A in a stripe parallel to the γ axis; the derived surface $(D_\alpha)^{n+1}(D_\gamma)^{n+1}F_A$ is constant along β stripes; and $(D_\beta)^{n+1}(D_\gamma)^{n+1}F_A$ is constant along α stripes.

Just as in the square-patch case, we shall respond to the lack of local flexibility of high-order derivatives by setting those derivatives to zero, once and for all. From now on, when building triangular-patch splines, we restrict ourselves to patches F_A that satisfy $(D_\alpha)^{n+1}(D_\beta)^{n+1}F_A = (D_\alpha)^{n+1}(D_\gamma)^{n+1}F_A = (D_\beta)^{n+1}(D_\gamma)^{n+1}F_A = 0$. Note that the patches that are generated by the method of Prop. 23.4 must satisfy these constraints, because the blending function $B_n(\mathrm{u})$ used to build them includes patches that are identically zero, far enough out in every stripe.

Each function $(D_\xi)^{n+1}(D_\eta)^{n+1}F_A$, where ξ and η are two of $\{\alpha, \beta, \gamma\}$, has degree $n-1$, so setting it to zero gives up $\binom{n+1}{2}$ degrees of freedom. If we can show that the constraints arising from the three different choices of $\{\xi, \eta\}$ are independent, we will have accounted for all $3\binom{n+1}{2}$ degrees of freedom that are missing in a patch drawn with the method of Prop. 23.4. To verify independence, it is easiest to think about the tensor space $P_*^{\otimes(3n+1)}$. A constraint on the high-order derivatives of a polynomial surface $F: P \to Q$ of degree $3n+1$ corresponds to a linear subspace of $P_*^{\otimes(3n+1)}$ on which the linear blossom $f_*^{\otimes}$ is required to be zero. Our three constraints correspond to the three subspaces

$$\begin{aligned}&\{\alpha^{n+1}\beta^{n+1}e \mid e \in P_*^{\otimes(n-1)}\},\\ &\{\alpha^{n+1}\gamma^{n+1}e \mid e \in P_*^{\otimes(n-1)}\}, \text{ and}\\ &\{\beta^{n+1}\gamma^{n+1}e \mid e \in P_*^{\otimes(n-1)}\}.\end{aligned}$$

These subspaces are certainly pairwise disjoint, since a tensor would have to have degree at least $3n+3$ in order to have all three of α, β, and γ as factors $n+1$ times each. Furthermore, the subspace spanned by the first two together lies inside the subspace $\{\alpha^{n+1}e \mid e \in P_*^{\otimes 2n}\}$, which, by the same argument, is disjoint from the third. Thus, the three constraints are independent.

We now come to the point where the analogy between the square-patch and triangular-patch theories breaks down, at least with my current state of knowledge. The derivative constraints that arose in the square-patch case implied that a square-patch spline should be built, not from arbitrary surfaces of degree $2n$, but rather from bipolynomial surfaces of degree $(n; n)$. In the triangular-patch case, it is once again true that the surfaces involved are not arbitrary surfaces of degree $3n+1$. But it isn't so clear just what they are instead. In lieu of a better name, let us call them triangular-patch surfaces. That is, let P be an affine parameter plane, let α, β, and γ be three vectors on P that satisfy $\alpha+\beta+\gamma=0$, and let n be a nonnegative integer. We shall call a polynomial surface F of degree $3n+1$ a ***triangular-patch surface*** if F satisfies the ***triangular-patch derivative constraints***

$$(D_\alpha)^{n+1}(D_\beta)^{n+1}F = (D_\alpha)^{n+1}(D_\gamma)^{n+1}F = (D_\beta)^{n+1}(D_\gamma)^{n+1}F = 0.$$

Warning: Current standard nomenclature uses the term "triangular-patch surface" to refer to what we are calling simply a polynomial surface, without any constraints on the derivatives. Our unconventional definition has the advantage that triangular-patch surfaces in our sense are actually the surfaces whose patches appear in triangular-patch splines.

The stage is now set for the juiciest challenge of this challenge-laden paper.

Challenge 23.6. Find a natural way to blossom a triangular-patch surface $F: P \to Q$ of degree $3n+1$, that is, a way that builds in the appropriate derivative constraints.

Restating this from the algorithmic perspective, suppose that we are given $3(n+1)^2$ control points at arbitrary places in Q. Find a way to construct the value $F(\mathbf{u})$ by performing $3n+1$ iterated stages of either one-dimensional or two-dimensional linear interpolations whose ratios are determined somehow by the location of $\mathbf{u}$ in P, starting with the control points. The resulting function $F(\mathbf{u})$ must satisfy the triangular-patch derivative constraints, but must be otherwise arbitrary.

Remark 23.7: Integrating tripolynomial spline flows. A cube viewed along its main diagonal looks like a hexagon. This idea can be parlayed into a different way of constructing triangular-patch spline surfaces, first exploited by de Boor and Höllig in their *box splines* [7], which might be useful in solving Challenge 23.6. Let us use the term *flow* for a parametric function whose parameter space is 3-dimensional; that is, "flow" is the third element in the sequence that starts with "curve" and "surface." By exploiting the theory of tripolynomial functions, it is easy to build spline flows of degree $3n$ by assembling cubic cells cut out of tripolynomial flows of degree $(n;n;n)$ in such a way as to guarantee C^{n-1} continuity across the squares where two cubic cells abut. Let $A_n: U \times V \times W \to \mathbf{R}$ denote the blending function that gives the influence of a particular control point in this method. The base-case blending function A_0 is identically equal to 1 on one cube in the 3-space $U \times V \times W$ and is 0 everywhere else, while A_n for general n is the $(n+1)$st convolutional power of A_0. Suppose that we integrate A_0 along each line parallel to the main diagonal. As a function of the position of the line of integration, the graph of this integral will be a hexagonal pyramid, just like B_0, the base-case blending function in Prop. 23.4. Because convolution and integration commute, the same coincidence holds for all degrees n. That is, the blending function B_n that we used in Prop. 23.4 to build our triangular-patch spline surfaces, which is the convolutional power of a hexagonal pyramid, can be computed by taking the convolutional power A_n of the characteristic function of a cube and integrating it along lines parallel to the main diagonal. This point of view might be a fruitful way to tackle Challenge 23.6.

Until a solution to Challenge 23.6 is discovered, our only option is to study triangular-patch spline surfaces using the blossoms that we get by treating their patches as degenerate cases of polynomial surfaces of degree $3n+1$. We should expect the answers that we get via this poor man's approach to be rather confusing.

To get a feeling for how bad this confusion is likely to be, suppose that we were studying square-patch spline surfaces, but that we hadn't invented the bipolynomial theory. For definiteness, consider bicubic spline surfaces with square patches, using the integers $\{\grave{\imath}\}$ and $\{\acute{\jmath}\}$ as the knot sequences in both U and V. Let us focus on the region $A = [\grave{2}, \grave{3}] \times [\acute{2}, \acute{3}]$ and the vertex $\mathbf{r} = \langle 2, 2\rangle$ at its lower left corner, as shown in Fig. 23.8. The control point $C(\mathbf{r})$ associated with the vertex $\mathbf{r}$ is given by $C(\mathbf{r}) := f_{(\grave{0},\grave{4})\times(\acute{0},\acute{4})}(\grave{1}, \grave{2}, \grave{3}; \acute{1}, \acute{2}, \acute{3})$, where f denotes the blossom of F as a bicubic spline. Without the bipolynomial theory to draw upon, we would have had to consider each patch of F as a degenerate hexic surface. Let $g_A: P^6 \to Q$ denote the 6-blossom of the patch F_A, while $f_A: U^3 \times V^3 \to Q$ is its $(3;3)$-blossom, and let $\mathbf{s} := \langle 3, 2\rangle$ and $\mathbf{t} := \langle 2, 3\rangle$. We can express the control point $C(\mathbf{r})$ as an affine combination of the Bézier points $g_A^{\circledast}(\mathbf{r}^i \mathbf{s}^j \mathbf{t}^k)$ for $i+j+k=6$ of $F_A(\Delta\mathbf{rst})$; the

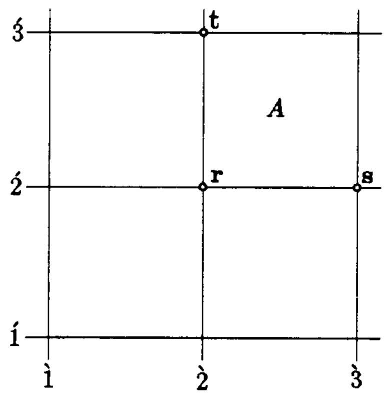


Fig. 23.8. The domain of one patch of a square-patch spline surface

formula turns out to be

$$C(\mathbf{r}) = \begin{cases} +5g_A^{\otimes}(\mathbf{r}^4\mathbf{t}^2) & -20g_A^{\otimes}(\mathbf{r}^3\mathbf{s}\mathbf{t}^2) & +10g_A^{\otimes}(\mathbf{r}^2\mathbf{s}^2\mathbf{t}^2) \\ -10g_A^{\otimes}(\mathbf{r}^5\mathbf{t}) & +40g_A^{\otimes}(\mathbf{r}^4\mathbf{s}\mathbf{t}) & -20g_A^{\otimes}(\mathbf{r}^3\mathbf{s}^2\mathbf{t}) \\ +g_A^{\otimes}(\mathbf{r}^6) & -10g_A^{\otimes}(\mathbf{r}^5\mathbf{s}) & +5g_A^{\otimes}(\mathbf{r}^4\mathbf{s}^2) \end{cases}$$

Deriving this formula is a straightforward application of the degree-adjusting formulas from Part D, starting from the fact that $C(\mathbf{r}) = f_A(\grave{1},\grave{2},\grave{3};\acute{1},\acute{2},\acute{3})$. Treating g_A as known, we can produce a formula for f_A in terms of g_A by first using Prop. 13.3 to split the degree of g_A from 6 to (6;6) and then using Eq. (11.7) twice to lower each of the resulting degrees from 6 to 3 separately. Note that the control point $C(\mathbf{r})$ had a simple label in terms of f_A, but its label in terms of g_A is both complicated and mysterious. The upcoming formulas for the control points of triangular-patch splines are also complicated and mysterious, presumably because they are stated in terms of the wrong blossom, a blossom that doesn't have the proper derivative constraints built in.

Fig. 23.9 shows a vertex $\mathbf{r}$ of the domain grid of a triangular-patch spline F and a region $A := \Delta\mathbf{rst}$ adjacent to that vertex. In the method of Prop. 23.4, the patch F_A is influenced by $3(n+1)^2$ control points. By computing the convolutions in Prop. 23.4, we can determine the linear relations that give the patch F_A in terms of its control points. By inverting those relations, we can compute the control points in terms of the patch. Let $C(\mathbf{r})$ denote the control point associated with the vertex $\mathbf{r}$. The case $n = 0$ is trivial; we have $C(\mathbf{r}) = F(\mathbf{r})$.

For $n = 1$, the spline F has quartic patches. Let g_A be the 4-blossom of the patch F_A; we are using the letter "g" rather than "f" to remind ourselves that the derivative constraints on F are not built into g. The control point $C(\mathbf{r})$ can be expressed as the following affine combination of the Bézier points of $F_A(\Delta\mathbf{rst})$:

$$C(\mathbf{r}) = g_A^{\otimes}\left(\begin{array}{ccccc} & & -\mathbf{r}^4 & & \\ & +2\mathbf{r}^3\mathbf{s} & & +2\mathbf{r}^3\mathbf{t} & \\ -2\mathbf{r}^2\mathbf{s}^2 & & +2\mathbf{r}^2\mathbf{s}\mathbf{t} & & -2\mathbf{r}^2\mathbf{t}^2 \end{array} \right).$$

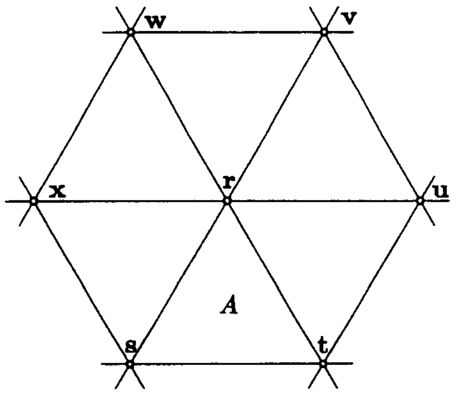


Fig. 23.9. The domain of one patch of a triangular-patch spline surface

Since F_A is a triangular-patch surface, we must have

$$(D_{\mathbf{r}-\mathbf{s}})^2(D_{\mathbf{r}-\mathbf{t}})^2F_A = 4!\,g^{\otimes}_{A,*}\left((\mathbf{r}-\mathbf{s})^2(\mathbf{r}-\mathbf{t})^2\right) = 0.$$

Therefore, we can get another equally valid formula by adding $(\mathbf{r}-\mathbf{s})^2(\mathbf{r}-\mathbf{t})^2$ to the argument of $g^{\otimes}_A$:

$$C(\mathbf{r}) = g^{\otimes}_A\begin{pmatrix} -\mathbf{r}^2\mathbf{s}^2 & & +6\mathbf{r}^2\mathbf{s}\mathbf{t} & & -\mathbf{r}^2\mathbf{t}^2 \\ & -2\mathbf{r}\mathbf{s}^2\mathbf{t} & & -2\mathbf{r}\mathbf{s}\mathbf{t}^2 & \\ & & +\mathbf{s}^2\mathbf{t}^2 & & \end{pmatrix}.$$

Exercise 23.10. Continuing our study of the $n = 1$ case in Fig. 23.9, show that the following is yet another valid formula for $C(\mathbf{r})$:

$$C(\mathbf{r}) = g(\mathbf{r},\mathbf{s},\mathbf{u},\mathbf{w}) + g(\mathbf{r},\mathbf{t},\mathbf{v},\mathbf{x}) - g(\mathbf{r},\mathbf{r},\mathbf{r},\mathbf{r}).$$

The function g here is the overloaded 4-blossom of the spline surface F. Note that the 4-blossoms of all six patches shown in Fig. 23.9 will agree on all three argument bags in this formula, since each bag includes two points on each of the three double knotlines. This formula can be turned into a geometric construction for the point $C(\mathbf{r})$ as follows. The C^2 continuity constraint between $A = \Delta\mathbf{rst}$ and $\Delta\mathbf{rtu}$ is enforced by three two-dimensional de Casteljau Diagrams with two shells. The central vertices of the outer shells of these three diagrams, going from $\mathbf{t}$ towards $\mathbf{r}$, are the points $g(\mathbf{t},\mathbf{t},\mathbf{s},\mathbf{u})$, $g(\mathbf{r},\mathbf{t},\mathbf{s},\mathbf{u})$, and $g(\mathbf{r},\mathbf{r},\mathbf{s},\mathbf{u})$. If we draw a line from the second of these to the third and extend it the same distance again, we end up at $g(\mathbf{r},\mathbf{w},\mathbf{s},\mathbf{u})$. The analogous constructions on three of the six edges radiating from $\mathbf{r}$ give the point $g(\mathbf{r},\mathbf{s},\mathbf{u},\mathbf{w})$, while the other three edges give $g(\mathbf{r},\mathbf{t},\mathbf{v},\mathbf{x})$. These two points are opposite vertices in a parallelogram in Q whose other two vertices are the surface point $F(\mathbf{r})$ and the control point $C(\mathbf{r})$.

The mysteries deepen when $n = 2$. If g_A is the 7-blossom of F_A, the control point $C(\mathbf{r})$ is given by

$$2C(\mathbf{r}) = g^{\otimes}_{A,*}\begin{pmatrix} & +57\mathbf{r}^4\mathbf{s}^3 & -99\mathbf{r}^4\mathbf{s}^2\mathbf{t} & -99\mathbf{r}^4\mathbf{s}\mathbf{t}^2 & +57\mathbf{r}^4\mathbf{t}^3 & \\ -20\mathbf{r}^3\mathbf{s}^4 & -215\mathbf{r}^3\mathbf{s}^3\mathbf{t} & +771\mathbf{r}^3\mathbf{s}^2\mathbf{t}^2 & -215\mathbf{r}^3\mathbf{s}\mathbf{t}^3 & -20\mathbf{r}^3\mathbf{t}^4 \\ & +132\mathbf{r}^2\mathbf{s}^4\mathbf{t} & -321\mathbf{r}^2\mathbf{s}^3\mathbf{t}^2 & -321\mathbf{r}^2\mathbf{s}^2\mathbf{t}^3 & +132\mathbf{r}^2\mathbf{s}\mathbf{t}^4 & \\ & -36\mathbf{r}\mathbf{s}^4\mathbf{t}^2 & +275\mathbf{r}\mathbf{s}^3\mathbf{t}^3 & -36\mathbf{r}\mathbf{s}^2\mathbf{t}^4 & & \\ & -20\mathbf{s}^4\mathbf{t}^3 & -20\mathbf{s}^3\mathbf{t}^4 & & & \end{pmatrix}.$$

The solver of Challenge 23.6 will have the satisfaction of explaining why these mysterious coefficients are correct.

Exercise 23.11: Osculating a polynomial surface by a triangular-patch surface. What should it mean to osculate an arbitrary polynomial surface F of degree $3n+1$ by a triangular-patch surface G, also of degree $3n+1$, at a point $\mathbf{r}$ in P? The triangular-patch derivative constraints tell us certain derivatives of G that have to be zero, and hence can't match the corresponding derivatives of F at $\mathbf{r}$. But which of the remaining derivatives should we choose to match? Let f and g be the $(3n+1)$-blossoms of F and G. Consider the tetrahedra of blossom values $\{f^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^l)\}$ and $\{g^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^l)\}$ for $i+j+k+l=3n+1$. The relation $\alpha+\beta+\gamma=0$ implies that

$$f^{\otimes}_{*}(\mathbf{r}^i\alpha^{j+1}\beta^k\gamma^l)+f^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^{k+1}\gamma^l)+f^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^{l+1})=0$$

whenever $i+j+k+l=3n$, and similarly for $g^{\otimes}_{*}$. Since we want G to be a triangular-patch surface, we must have $g^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^l)=0$ whenever any two of the three indices j, k, and l are both at least $n+1$. Show that G is uniquely determined by the requirement that $g^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^l)=f^{\otimes}_{*}(\mathbf{r}^i\alpha^j\beta^k\gamma^l)$ whenever $i+\min(j,k,l)\geq n$. The surface G determined in this way seems to be the natural choice to called the triangular-patch surface that osculates F at $\mathbf{r}$.

Part G: The Rational Case

At the outset, we made three decisions about how we would model shapes: our models were to be piecewise, parametric, and polynomial. There are some situations in computer-aided geometric design where the restriction to polynomial curves and surfaces is unpleasant. For example, circles are not polynomial curves. Of course, they can be approximated arbitrarily well by polynomial splines; but it would be simpler if such basic shapes could be handled exactly. A related difficulty is that the class of polynomial curves is not closed under projective maps. Extending our models to allow rational maps instead of just polynomial ones solves both of these problems. In this concluding part, we will investigate how the blossoming technology can be extended to the rational case.

24. On choosing scale factors

So far, we have studied blossoming in two worlds. In the affine world, we blossomed a polynomial map into a multiaffine map, and then tensored to get an affine map. In the linear world, we blossomed a homogeneous polynomial map into a multilinear map, and then tensored to get a linear map. There is yet a third world where blossoming can be applied: the projective world, which consists of projective spaces and projective maps between them. Rational maps turn out to be the natural notion of a "map of degree n" in the projective world. By analogy, we expect that blossoming a rational map should produce a multiprojective map, and then tensoring should give a projective map. Roughly speaking, that is what does happen. But some subtle issues are involved, associated with choosing scale factors. To clarify those issues, we shall start by investigating rational functions in the more familiar context of maps between affine spaces.

If P and Q are affine spaces, a map $R\colon P \to Q$ is called *rational* if every coordinate of the point $R(\mathbf{u})$ is a rational function of the coordinates of $\mathbf{u}$, that is, a quotient of polynomials in the variables u^i where $\mathbf{u} = \langle u^1, \ldots, u^p \rangle$. For example, the formula $S(\bar{u}) := \langle u, 1/u \rangle$ defines a rational parametric curve $S\colon L \to A$ in the plane A. In fact, the curve S is one parameterization of the standard rectangular hyperbola, which is given implicitly by the equation $xy = 1$. A rational map between affine spaces is undefined at an argument point $\mathbf{u}$ that causes the denominator of some coordinate of $R(\mathbf{u})$ to be zero. For example, $S(\bar{0})$ is undefined. We don't permit the denominator of any coordinate of R to be identically zero; this restriction guarantees that $R(\mathbf{u})$ will be well-defined except for $\mathbf{u}$ in some set of measure zero in P.

Computing the degree of a rational map is a bit tricky, as the example curve S demonstrates. A hyperbola like S is surely a curve of degree 2; but the definition $S(\bar{u}) := \langle u, 1/u \rangle$ doesn't involve any polynomial of degree greater than 1. The solution for this difficulty is to make the convention that all of the coordinates of the point $R(\mathbf{u})$ must be put over a common denominator before we compute the degree. This rule forces us to rewrite the hyperbola $S(\bar{u})$ in the equivalent form

$$S(\bar{u}) = \left\langle \frac{u^2}{u}, \frac{1}{u} \right\rangle,$$

which is obviously quadratic.

Common denominators are not unique, so we have a choice to make at this point. If we are trying to compute the degree of a rational map R, we should choose a least common denominator, that is, a common denominator of minimum degree. Even then, we will still have to choose a scalar factor: for example, we could equally well have written

$$S(\bar{u}) = \left\langle \frac{wu^2}{wu}, \frac{w}{wu} \right\rangle$$

for any nonzero scalar w. If we are dealing with rational maps of degree n, but the particular map R happens to be degenerate and have some smaller degree m, it makes perfect sense to adopt a common denominator whose degree exceeds the minimum possible by as much as $n - m$. For example, we could view the hyperbola S as a degenerate rational cubic curve by writing it

$$S(\bar{u}) = \left\langle \frac{u^2(au+b)}{u(au+b)}, \frac{au+b}{u(au+b)} \right\rangle,$$

where $(au + b)$ is an arbitrary linear factor.

Once we have made an explicit choice of a common denominator, there is no reason to keep separate copies of that denominator in each coordinate of R. Instead, it is more convenient to think of each numerator as a coordinate and the common denominator as one additional coordinate. Recall, from Section 5, that the affine object space Q sits as a hyperplane inside its linearization Q_*, which consists of all 1-tensors on Q of all flavors. Let $c: P \to \mathbf{R}$ denote the chosen common denominator for the coordinates of the rational function $R: P \to Q$, and consider the map $F: P \to Q_*$ defined by $F(\mathbf{u}) := c(\mathbf{u})R(\mathbf{u})$. Since $c(\mathbf{u})$ is a common denominator, the map F is a polynomial map. Furthermore, the rational map R is determined as the projection (with center at the origin) of the polynomial map F onto the 1-flavored hyperplane $Q = Q_1$, sitting in Q_*. For example, if we choose the common denominator u for the quadratic rational curve $S(\bar{u}) := \langle u, 1/u \rangle$ in the plane A, the corresponding quadratic polynomial curve $G: L \to A_*$ is the parabola given by $G(\bar{u}) := \langle u^2, 1; u \rangle$. Projecting the parabola G down into the plane A in A_* gives the hyperbola S. Alternatively, if we decide on $u(au+b)$ as the common denominator for S, we would get the cubic polynomial curve $H(\bar{u}) := \langle u^2(au+b), (au+b); u(au+b) \rangle$, which also projects down onto S.

Proposition 24.1. *Let P and Q be affine spaces. The equation $F(\mathbf{u}) = c(\mathbf{u})R(\mathbf{u})$ provides a one-to-one correspondence between polynomial maps $F: P \to Q_*$ whose flavor coordinates are not identically zero and pairs (R, c), where $R: P \to Q$ is a rational map and $c: P \to \mathbf{R}$ is a common denominator for the coordinates of R.*

Proof. Suppose that we start with a pair (R, c). We already noted that the formula $F(\mathbf{u}) := c(\mathbf{u})R(\mathbf{u})$ will define a polynomial map $F: P \to Q_*$, since $c(\mathbf{u})$ is a common denominator for all of the coordinates of $R(\mathbf{u})$. Furthermore, the flavor coordinate of this map F is given by $\mathrm{Flav}(F(\mathbf{u})) = c(\mathbf{u})$ and is hence nonzero, since the zero polynomial is not a legitimate common denominator.

Conversely, given a polynomial map F, let $c(\mathbf{u}) := \mathrm{Flav}(F(\mathbf{u}))$ be the flavor coordinate of F, which, by assumption, is not the zero polynomial. For all points $\mathbf{u}$ with $c(\mathbf{u}) \neq 0$, the formula $R(\mathbf{u}) := F(\mathbf{u})/c(\mathbf{u})$ defines the value $R(\mathbf{u})$ to be a 1-flavored 1-tensor on Q, and hence a point in Q. Since each coordinate of R is the quotient of a coordinate of F by $c(\mathbf{u})$, the function R is rational, with c as a common denominator for its coordinates. □

The stipulation, in Prop. 24.1, that the flavor coordinate of the polynomial map F must not be identically zero arises because we have been considering rational maps, so far, as maps between affine spaces. We can eliminate this restriction by converting to the projective world. That conversion will also reduce the frequency of cases where a value of a rational function is undefined, although it won't eliminate the problem of undefined values completely. We proceed as follows.

In Section 5, we saw that the Homogenizing Principle could be used to convert a polynomial map $M: P \to Q$ of degree n into a flavor-exponentiating homogeneous polynomial map $M_*: P_* \to Q_*$, also of degree n. More generally, suppose that we apply the Homogenizing Principle to the polynomial map $F: P \to Q_*$ obtained from a rational map by Prop. 24.1, whose values $F(\mathbf{u})$ are not necessarily 1-flavored. The only effect of dropping the flavor constraint on F is that we lose the corresponding flavor constraint on F_*. That is, the homogenized map $F_*: P_* \to Q_*$ will be an arbitrary homogeneous polynomial map of degree n, not necessarily flavor-exponentiating. In the hyperbola example, the homogenized form of $G(\bar{u}) = \langle u^2, 1; u\rangle$ is $G_*(\langle u; v\rangle) := \langle u^2, v^2; uv\rangle$. The homogenized form F_*, just like F itself, depends on the choice of common denominator. If we chose $u(au+b)$ as the common denominator for the hyperbola, we would get the cubic homogenized form $H_*(\langle u; v\rangle) := \langle u^2(au+bv), v^2(au+bv); uv(au+bv)\rangle$.

The domain P_* and codomain Q_* of the homogenized map F_* are both linear spaces. As we discussed in Section 9, the basic idea of projective geometry is to treat the lines through the origin of a linear space as the "points" of a new space, called a ***projective space***. If e is a nonzero tensor in a linearized space P_*, we shall use the expression $[e] := \{we \mid w \in \mathbf{R}\}$ to denote the line through the origin of P_* that contains e; the expression $[0]$ is not defined. For example, the expression $[\langle u; v\rangle]$ denotes a particular line through the origin of L_*, and hence a "point" in the ***projective line***. For notational convenience, we shall omit the angle brackets in the expression $[e]$ when the 1-tensor e is given in explicit coordinates, writing simply $[u; v]$, for example, to denote that point in the projective line. (From now on, we shall also elide the quotation marks around the word "point" when referring to points in projective spaces.) The numbers inside these square brackets are called ***homogeneous coordinates***. They are defined only up to a common scalar multiple; that is, it is the ratios between different homogeneous coordinates that are well-defined, not the values of the coordinates individually.

Since the map F_* is homogeneous, it carries a line $[e]$ through the origin of P_* into the line $[F_*(e)]$ through the origin of Q_*. Let $P_{[*]}$ denote the set of all lines through the origin of P_*, which constitutes a projective space of dimension $p = \dim(P)$, and similarly for $Q_{[*]}$. We can define a map $F_{[*]}: P_{[*]} \to Q_{[*]}$ by the equation $F_{[*]}([e]) := [F_*(e)]$ whenever the right-hand side is defined, that is, whenever $F_*(e) \neq 0$. The map $F_{[*]}$ is the projective analog of the rational map R; it is a rational map between projective spaces, according to our upcoming definition. Note that, since we ignore scalar multiples in the domain and codomain when we build $F_{[*]}$, the map $F_{[*]}$ is independent of the choice of common denominator, that is, of the scale factor that is built into F_*. The hyperbola in our continuing example, when viewed as a quadratic rational map from the projective line to the projective plane, is given by $G_{[*]}([u;v]) = [u^2, v^2; uv]$. Note that the map $H_{[*]}$ given by $H_{[*]}([u;v]) = [u^2(au+bv), v^2(au+bv); uv(au+bv)]$ is the same map as $G_{[*]}$, although G_* and H_* are differently scaled.

If $e \neq 0$ but $F_*(e) = 0$, then the line $[e]$ is collapsed by F_* into the origin of Q_*. This is a more complete form of undefinedness than the ones we have considered so far. It corresponds to a value of the argument $\mathbf{u}$ that makes all of the numerators and the common denominator of $R(\mathbf{u})$ simultaneously zero. Such cases will remain undefined even in the projective world. The advantage of the projective world is that only those cases have to be left undefined. By contrast, a rational map between affine spaces is undefined whenever the common denominator is zero, regardless of the values of the numerators.

To clarify the different types of undefinedness, consider the hyperbola example. In the affine world, no matter what common denominator we choose, the hyperbola $S(\bar{u}) := \langle u, 1/u \rangle$ will be undefined when $u = 0$. If we choose a common denominator of more than the minimum degree, we may add more instances of undefinedness. For example, using the denominator $u(au + b)$ would make $S(\overline{-b/a})$ undefined. In the projective world, it is also the case that the frequency of undefinedness depends upon the scale factor we choose. If we define $G_{[*]}([u;v]) := [u^2, v^2; uv]$ by using the homogeneous quadratic map $G_*(\langle u;v \rangle) := \langle u^2, v^2; uv \rangle$ that results from the common denominator u, then $G_{[*]}$ will be defined everywhere. The relation $G_{[*]}([0;1]) = [0,1;0]$ tells us that the value $S(\bar{0})$, viewed projectively, is the point at infinity in the y direction. The relation $G_{[*]}([1;0]) = [1,0;0]$ tells us that the projective form of the hyperbola also passes through the point at infinity in the x direction, a point we might refer to as $S(\overline{\infty})$. Suppose, on the other hand, that we use the common denominator $u(au+b)$. We then get the function $H_{[*]}([u;v]) := [u^2(au+bv), v^2(au+bv); uv(au+bv)]$, which agrees with $G_{[*]}$ almost everywhere. The value $H_{[*]}([-b;a])$, however, is undefined; by using a scale factor of higher degree than necessary, we have introduced an undefined case.

Even in the projective world, we can't get rid of undefinedness completely—at least, not for surfaces. Rational curves are a special case. The coordinates of a rational curve are homogeneous bivariate polynomials, which behave essentially like nonhomogeneous univariate polynomials. If a collection of such polynomials have a common root, then they, in fact, have a common factor. If we remove all common factors by choosing a polynomial form F_* of minimum degree, the resulting rational curve will be defined everywhere. But the coordinates of a rational surface are homogeneous trivariate polynomials, which behave like nonhomogeneous bivariate polynomials. The fact that a collection of such polynomials have a common root does not come close to implying that they have a common factor. For example, the polynomials u and v have the common root $u = v = 0$, but they have no common factor. As a result, the rational map of degree one from the projective plane to the projective line given by $[u, v; w] \mapsto [u; v]$ is unavoidably undefined at the origin $[0, 0; 1]$.

Let $P_{[*]}$ and $Q_{[*]}$ be projective spaces, and let $F_{[*]}: P_{[*]} \to Q_{[*]}$ be a function defined almost everywhere. We shall call a nonzero homogeneous polynomial map $F_*: P_* \to Q_*$ a ***scaled form*** of $F_{[*]}$ if the identity $F_{[*]}([e]) = [F_*(e)]$ holds almost everywhere. A map $F_{[*]}$ between projective spaces is called ***a rational map*** if some scaled form F_* of $F_{[*]}$ exists. The ***degree*** of the rational map $F_{[*]}$, written $\deg(F_{[*]})$, is the minimum n with the property that $F_{[*]}$ has a scaled form F_* of degree n. The analog of Prop. 24.1 in the projective world is the following easy proposition.

Proposition 24.2. *Let $P_{[*]}$ and $Q_{[*]}$ be projective spaces. Every nonzero homogeneous polynomial map $F_*: P_* \to Q_*$ is a scaled form of a unique rational map $F_{[*]}: P_{[*]} \to Q_{[*]}$, which can be determined by the identity $F_{[*]}([e]) = [F_*(e)]$. On the other hand, every rational map $F_{[*]}$ has an infinite number of scaled forms. If we pick one scaled form F_* of minimum degree as the reference, a nonzero homogeneous polynomial map $G_*: P_* \to Q_*$ will be a scaled form of $F_{[*]}$ if and only if $G_*(e) = W(e)F_*(e)$ for some homogeneous polynomial $W: P_* \to \mathbf{R}$. In particular, the various scaled forms of minimum degree are scalar multiples of each other, and they can be distinguished from scaled forms of more than the minimum degree by the property that their coordinate polynomials have no nontrivial common factor.* □

The easiest way to handle a rational map $F_{[*]}$ is to choose an explicit scaled form F_* right at the outset. We can apply the linear-world variants of blossoming and tensoring to this scaled form, with the results shown in the right-hand column of Fig. 24.3: blossoming converts the scaled form F_* into a multilinear map f_*, and tensoring then gives us a linear map $f_*^{\otimes}$. Restricting these three maps to arguments that are 1-flavored gives us the maps in the middle column of Fig. 24.3: a polynomial map F, its multiaffine blossom f_*, and its affine blossom $f_*^{\otimes}$. These six maps are the six different scaled guises of a rational map. Note that the middle and right-hand columns in Fig. 24.3 are almost identical to all of Fig. 5.1. The middle column of Fig. 24.3 differs from the left-hand column of Fig. 5.1 only in that the codomains are Q_* instead of Q in each case. The right-hand column of Fig. 24.3 differs from the right-hand column of Fig. 5.1 only in the lack of the constraints that the functions in each box must carry 1-flavored arguments to 1-flavored results. The differences between Fig. 24.3 and Fig. 5.1 form a consistent set, in the sense that the six scaled guises of a rational map are in one-to-one correspondence: given a map of any one of the six types, unique maps of the other five types exist that correspond.

The multiple dividing lines in the left half of Fig. 24.3 are warnings that some of those correspondences are not one-to-one. As described in Prop. 24.2, moving from a rational map $F_{[*]}$ in the upper-left box to some scaled guise in the middle or right-hand column involves choosing a polynomial scale factor of degree $n - \deg(F_{[*]})$. Even when $\deg(F_{[*]}) = n$, there is a scalar scale factor to choose; the second thin vertical line represents the choice of that scalar. The thick line separating the upper-left box from all the rest represents the choice of the scaling polynomial modulo its scalar part, which is a nontrivial choice only when $\deg(F_{[*]}) < n$.

Rational map $F_{[*]}: P_{[*]} \to Q_{[*]}$ rational of degree n	**Polynomial map** $F: P \to Q_*$ polynomial of degree n	**Homogeneous poly. map** $F_*: P_* \to Q_*$ homogeneous poly. of deg. n
Multiprojective blossom $f_{[*]}: (P_{[*]})^n \to Q_{[*]}$ symmetric, multiprojective	**Multiaffine blossom** $f: P^n \to Q_*$ symmetric, multiaffine	**Multilinear blossom** $f_*: (P_*)^n \to Q_*$ symmetric, multilinear
Projective blossom $f^{\otimes}_{[*]}: P^{\otimes n}_{[*]} \to Q_{[*]}$ projective	**Affine blossom** $f^{\otimes}: P^{\otimes n} \to Q_*$ affine	**Linear blossom** $f^{\otimes}_*: P^{\otimes n}_* \to Q_*$ linear

Scaling

Fig. 24.3. The nine guises of a rational map

The lower-left box in Fig. 24.3 is the easier of the remaining two to deal with. If $P_{[*]}$ and $Q_{[*]}$ are projective spaces, a map $H_{[*]}: P_{[*]} \to Q_{[*]}$ is called ***projective*** if it has a linear scaled form $H_*: P_* \to Q_*$. That is, projective maps are precisely rational maps of degree one. Ignoring scalar multiples in the domain and codomain of the linear blossom $f^{\otimes}_*: P^{\otimes n}_* \to Q_*$ gives us the ***projective blossom***, which is a projective map $f^{\otimes}_{[*]}: P^{\otimes n}_{[*]} \to Q_{[*]}$. The second thin line separating the lower-left box from the six scaled guises warns that choosing a particular linear scaled form of the projective blossom involves choosing a scalar scale factor.

To handle the remaining, middle-left box in Fig. 24.3, we must define the concept of a multiprojective map. For our current purposes, the appropriate definition is as follows: a map $f_{[*]}: (P_{[*]})^n \to Q_{[*]}$ is ***multiprojective*** if it has a multilinear scaled form, that is, if a multilinear map $f_*: (P_*)^n \to Q_*$ exists that satisfies the identity $f_{[*]}([e_1], \ldots, [e_n]) = [f_*(e_1, \ldots, e_n)]$ for almost all n-tuples $(e_1, \ldots, e_n)$. (Section 26 considers a weaker notion of multiprojectivity and studies how that notion relates to the notion defined here.) To convert a multilinear blossom into a ***multiprojective blossom***, we ignore scalar multiples in all of the domain spaces and in the codomain space. Choosing a particular multilinear scaled form for a multiprojective blossom involves choosing a scalar scale factor; hence, there is a second thin line separating the multiprojective blossom from the six scaled guises.

Applying blossoming and tensoring in the projective world corresponds to moving down the left-hand column of Fig. 24.3. Tensoring is a one-to-one correspondence in the projective world, because the same scalar scale factor that we must choose when going from the multiprojective box to the multilinear box is washed out again when we move from the linear box to the projective box. But blossoming is not a one-to-one correspondence in the projective world. Moving from rational to homogeneous-polynomial involves choosing a polynomial scale factor of degree

$n - \deg(F_{[*]})$. Only the scalar part of this polynomial factor gets washed out again when we return from multilinear to multiprojective; the thick line separating the rational box from the multiprojective one represents the remaining choice.

Warning 24.4. *The multiprojective n-blossom of a rational map $F_{[*]}$ between projective spaces is not unique when $\deg(F_{[*]}) < n$.*

As an example of Warning 24.4, take the identity function $G_{[*]}: L_{[*]} \to L_{[*]}$ on the projective line $L_{[*]}$, which is a rational map with $\deg(G_{[*]}) = 1$, and consider it as a degenerate case of a quadratic rational curve. Quadratic scaled forms G_* for $G_{[*]}$ have the form $G_*(\langle u; v\rangle) := \langle u(au + bv); v(au + bv)\rangle$ for scalars a and b, not both zero. The bilinear blossom g_* of G_* is given by

$$g_*(\langle u_1; v_1\rangle, \langle u_2; v_2\rangle) := \left\langle au_1u_2 + b\frac{u_1v_2 + u_2v_1}{2};\ a\frac{u_1v_2 + u_2v_1}{2} + bv_1v_2\right\rangle.$$

When we project this down to get a biprojective blossom $g_{[*]}$ for $G_{[*]}$, we find that

$$g_{[*]}([u_1; v_1], [u_2; v_2]) := \left[au_1u_2 + b\frac{u_1v_2 + u_2v_1}{2};\ a\frac{u_1v_2 + u_2v_1}{2} + bv_1v_2\right].$$

On the diagonal, we have $g_{[*]}([u; v], [u; v]) = [u(au + bv); v(au + bv)] = [u; v]$, independent of the scalars a and b. But the off-diagonal values of the blossom $g_{[*]}$ do depend on the choices of a and b. This phenomenon may seem less mysterious when considered in the affine world, where $v = v_1 = v_2 = 1$, and the corresponding formulas are

$$G_{[*]}(u) = u = \frac{u(au + b)}{au + b} = \frac{au^2 + bu}{au + b}$$

$$g_{[*]}(u_1, u_2) = \frac{au_1u_2 + b(u_1 + u_2)/2}{a(u_1 + u_2)/2 + b}.$$

Note that $g_{[*]}(\bar{0}, \bar{1}) = b/(a + 2b)$. Thus, the off-diagonal values of the biprojective blossom $g_{[*]}$ do depend on the ratio $a : b$, which is the part of the polynomial scale factor $au + bv$ that remains when scalar factors are ignored.

25. Controlling and joining rational curves

In brief, the moral of Section 24 was the following: the way to draw a rational curve or surface in an affine space Q is to draw a polynomial curve or surface in the linearization Q_* and then to project down onto the hyperplane $Q = Q_1$. In this section, we shall investigate a few of the consequences of this two-step approach.

To begin with, suppose that we want to draw a segment of a rational curve $F_{[*]}: L_{[*]} \to Q_{[*]}$ of degree n. One of the six guises of any scaled form F_* of $F_{[*]}$ is an n-ic polynomial map $F: L \to Q_*$. We shall focus on that guise, since a polynomial curve of degree n is such a familiar object.

The obvious way to control the scaled form F is to specify its Bézier points with respect to some reference interval $[\bar{s}, \bar{t}]$ for L. The name *Bézier tensor* is probably better in this context, since a Bézier point of F is a point in Q_*, that is, a 1-tensor on Q. When specifying a Bézier tensor e, it is common practice to adopt a rather unusual, non-Cartesian coordinate system for the space Q_*. Instead of specifying the coordinates of e directly, we specify the coordinates of the point $\mathbf{x}$ in Q to which e projects, which is called a *Bézier point of the rational curve* $F_{[*]}$, and we separately specify the flavor $w := \mathrm{Flav}(e)$ of the Bézier tensor e, which, in this context, is called its *weight*. For example, suppose that Q is a plane. We shall specify a Bézier tensor e in the 3-space Q_* by giving x, y, and w where $e = \langle wx, wy; w\rangle$; that is, the tensor e projects to the Bézier point $\mathbf{x} = \langle x, y\rangle$ in Q, and the weight of e is w. The advantage of this approach is that the designer generally prefers to think about the geometry of Q rather than the geometry of Q_*, and hence the projected coordinates x and y are more relevant than the unprojected coordinates wx and wy. One difficulty with using x, y, and w as coordinates on Q_* is that tensors in Q_0, that is, vectors on Q, don't have well-defined coordinates. For example, there are no values of x, y, and w that will make $\langle wx, wy; w\rangle = \langle 1, 3; 0\rangle$. This is not a crippling problem in practice, however, since the weights of Bézier tensors are usually restricted to be positive numbers.

Thus, we specify the rational curve segment $F_{[*]}([\bar{s}, \bar{t}])$ by giving its Bézier points and their associated weights, which determines the Bézier tensors of the scaled polynomial segment $F([\bar{s}, \bar{t}])$. From a blossoming point of view, the Bézier tensors e_i are the values $e_i := f^{\circledast}(\bar{s}^{n-i}\bar{t}^{i})$ in Q_* for i in $[0, n]$, and their projections in Q (more precisely, in $Q_{[*]}$) are the Bézier points $\mathbf{x}_i := f^{\circledast}_{[*]}(\bar{s}^{n-i}\bar{t}^{i})$. The weight w_i of the ith Bézier point is the scalar by which that point must be multiplied in order to give the ith Bézier tensor: $e_i = w_i\mathbf{x}_i$. The rational curve $F_{[*]}$ is given, after projection into Q, by

$$F_{[*]}(\bar{u}) = \frac{\displaystyle\sum_{0\le i\le n} \binom{n}{i}\left(\frac{t-u}{t-s}\right)^{n-i}\left(\frac{u-s}{t-s}\right)^{i} w_i\mathbf{x}_i}{\displaystyle\sum_{0\le i\le n} \binom{n}{i}\left(\frac{t-u}{t-s}\right)^{n-i}\left(\frac{u-s}{t-s}\right)^{i} w_i},$$

and its multiprojective blossom $f_{[*]}$ is given by

$$f_{[*]}(\bar{u}_1, \ldots, \bar{u}_n) = \frac{\displaystyle\sum_{\substack{I\cap J=\emptyset\\ I\cup J=\{1,\ldots,n\}}} \prod_{i\in I}\left(\frac{t-u_i}{t-s}\right)\prod_{j\in J}\left(\frac{u_j-s}{t-s}\right) w_{|J|}\mathbf{x}_{|J|}}{\displaystyle\sum_{\substack{I\cap J=\emptyset\\ I\cup J=\{1,\ldots,n\}}} \prod_{i\in I}\left(\frac{t-u_i}{t-s}\right)\prod_{j\in J}\left(\frac{u_j-s}{t-s}\right) w_{|J|}}.$$

Going from a rational curve to its Bézier tensors involves choosing a scale factor, as we discussed in Section 24. The scalar scale factor represented by the second thin vertical line in Fig. 24.3 manifests itself in the fact that we can multiply all of the weights w_i of all of the Bézier tensors by a common scalar without affecting the rational curve $F_{[*]}$. The polynomial scale factor represented by the thick black line around the rational box in Fig. 24.3 arises only when a degeneracy of degree is involved. When it arises, however, it affects the positions of the Bézier points as well as their associated weights. This phenomenon is precisely the import of

Warning 24.4, and the example given their works here as well; perhaps that example is worth reviewing. Suppose that we want to describe the x-axis of the plane Q under the trivial parameterization $F_{[*]}(\bar{u}) = \langle u, 0\rangle$ as a degenerate case of a quadratic rational curve, and suppose that we choose the standard reference interval $[\bar{0}, \bar{1}]$ for the parameter line L. The first and last Bézier points of $F_{[*]}([\bar{0}, \bar{1}])$ are clearly determined: $f_{[*]}(\bar{0}, \bar{0}) = F_{[*]}(\bar{0}) = \langle 0, 0\rangle$ and $f_{[*]}(\bar{1}, \bar{1}) = F_{[*]}(\bar{1}) = \langle 1, 0\rangle$. But the location of the middle Bézier point along the x-axis and the weights of the three Bézier tensors depend upon the common denominator $(au + b)$ in the formula

$$F_{[*]}(\bar{u}) = \left\langle \frac{u(au+b)}{au+b}, \frac{0}{au+b} \right\rangle.$$

Choosing the denominator $(au + b)$ results in the scaled parabola $F(\bar{u}) := \langle au^2 + bu, 0; au + b\rangle$ in Q_*, whose biaffine blossom is

$$f(\bar{u}_1, \bar{u}_2) := \left\langle au_1u_2 + b\left(\frac{u_1+u_2}{2}\right),\ 0;\ a\left(\frac{u_1+u_2}{2}\right) + b \right\rangle.$$

Thus, the middle Bézier point will be located at $f_{[*]}(\bar{0}, \bar{1}) = \langle b/(a+2b), 0\rangle$, and the weights of the three Bézier tensors will be b, $b + a/2$, and $b + a$.

There is another type of redundancy involved in the choice of the weights of the Bézier tensors, if the details of the parameterization of the curve segment $F_{[*]}([\bar{s}, \bar{t}])$ don't matter to us. Note that there is a one-parameter family of projective maps $\varphi\colon L_{[*]} \to L_{[*]}$, from the projective line $L_{[*]}$ to itself, that fix the two points $\bar{s}$ and $\bar{t}$. In particular, the map

$$\varphi(\bar{u}) := \frac{\left(\frac{t-u}{t-s}\right)a\bar{s} + \left(\frac{u-s}{t-s}\right)b\bar{t}}{\left(\frac{t-u}{t-s}\right)a + \left(\frac{u-s}{t-s}\right)b}$$

fixes $\bar{s}$ and $\bar{t}$ while taking the midpoint $(\bar{s} + \bar{t})/2$ to the point $(a\bar{s} + b\bar{t})/(a + b)$, which can be anywhere in the interval $(\bar{s}, \bar{t})$, depending upon the ratio $a : b$. We can think of the numbers a and b as the weights of the endpoints $\bar{s}$ and $\bar{t}$ of the reference interval, viewed as tensors in L_* rather than as points in L. If $F_{[*]}$ is a rational n-ic, then the composite map $G_{[*]}\colon \bar{u} \mapsto F_{[*]}(\varphi(\bar{u}))$ will also be a rational n-ic, and the segment $G_{[*]}([\bar{s}, \bar{t}])$ will be the same as $F_{[*]}([\bar{s}, \bar{t}])$ except for the details of the interior parameterization. The Bézier points of $G_{[*]}$ will be the same as those of $F_{[*]}$, with the weights readjusted as follows:

$$g^{\otimes}(\bar{s}^{n-i}\bar{t}^{i}) = a^{n-i}b^{i}f^{\otimes}(\bar{s}^{n-i}\bar{t}^{i}).$$

There was no phenomenon analogous to this redundancy in the polynomial case, since the only affine map from the line L to itself that preserves the endpoints of the reference interval $[\bar{s}, \bar{t}]$ is the identity.

The case of rational quadratic curves deserves some special attention, since the rational quadratics are precisely the well-known conic sections. A segment of a rational quadratic has three Bézier points, say $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$, each with a corresponding weight, say u, v, and w. Scaling all three weights by the same factor doesn't change anything. More generally, replacing the three weights by a^2u, abv, and b^2v for $a \neq b$

changes only the parameterization of the conic segment. Thus, the sole effect of the weights on the shape of the segment is contained in the ratio $v : \sqrt{uw}$. For simplicity, let us assume that the weights are all positive (or all negative, it doesn't matter). If $v^2 = uw$, the conic segment is the parabolic arc whose Bézier points, in the polynomial sense, are $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. If $v^2 > uw$, we have an arc of a hyperbola; if $v^2 < uw$, less than half of an ellipse. This behavior helps to justify the name "weight" for the flavor of a Bézier tensor, since increasing the weight v associated with the apex $\mathbf{y}$ of the control triangle $\triangle\mathbf{xyz}$ pulls the conic segment up towards that apex. In fact, let $\mathbf{m}$ denote the midpoint of the base $\mathbf{xz}$ of the control triangle. The parameter $v/\sqrt{uw}$ is projectively related to the point $\mathbf{r}$ where the conic segment cuts the median line $\mathbf{ym}$ by the formula

$$\frac{v}{\sqrt{uw}} = \frac{\mathbf{r} - \mathbf{m}}{\mathbf{y} - \mathbf{r}}.$$

If u and w have the same sign, but v has the opposite sign, we get the complements of the segments above: a parabolic arc that includes the point at infinity, a hyperbolic arc including both points at infinity, or more than half of an ellipse. When u and w have opposite signs, the endpoints $\mathbf{x}$ and $\mathbf{z}$ are on different branches of a hyperbola (and the conic does not intersect the median line, over the real numbers).

Given a fixed control triangle, it makes an interesting geometric problem of the old school to compute the value of the parameter $v/\sqrt{uw}$ that gives the ellipse of minimum eccentricity, as described in the following pair of exercises. Of course, eccentricity is not even an affine invariant, so both exercises end up having to compute a ratio between distances in different directions.

Exercise 25.1. Let D be a diameter of an ellipse, and consider the chords of the ellipse that are parallel to D. Show that the locus of the midpoints of these chords is another diameter D' of the ellipse, and show that the chords parallel to D' have their midpoints on D. The diameters D and D' are called *conjugate*. The major and minor axes of an ellipse form one pair of conjugate diameters, whose lengths are unequal. Show that each ellipse has a unique pair of conjugate diameters that are equal in length; the axes of the ellipse bisect the angles between this pair of conjugate diameters. Let $\triangle\mathbf{xyz}$ be a control triangle for a conic segment, and let $\mathbf{m}$ be the midpoint of the base $\mathbf{xz}$. Show that every ellipse tangent to $\mathbf{xy}$ at $\mathbf{x}$ and tangent to $\mathbf{yz}$ at $\mathbf{z}$ has a pair of conjugate diameters with one parallel to the base $\mathbf{xz}$ and the other lying along the median line $\mathbf{ym}$. Finally, show that the ellipse of this form with minumum eccentricity is the one in which those two conjugate diameters have equal length. See Fig. 25.2. Hint: Problem 54 in Heinrich Dörrie's book *100 Great Problems of Elementary Mathematics* [20] considers the more general situation in which the ellipses are constrained to circumscribe a quadrilateral. In our case, the quadrilateral is degenerate: its four vertices, in order, are $\mathbf{x}$, $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{z}$, while $\mathbf{y}$ is the point of intersection of its two opposite zero-length sides.

Exercise 25.3. Algebra and geometry are both paths to the truth; show that the

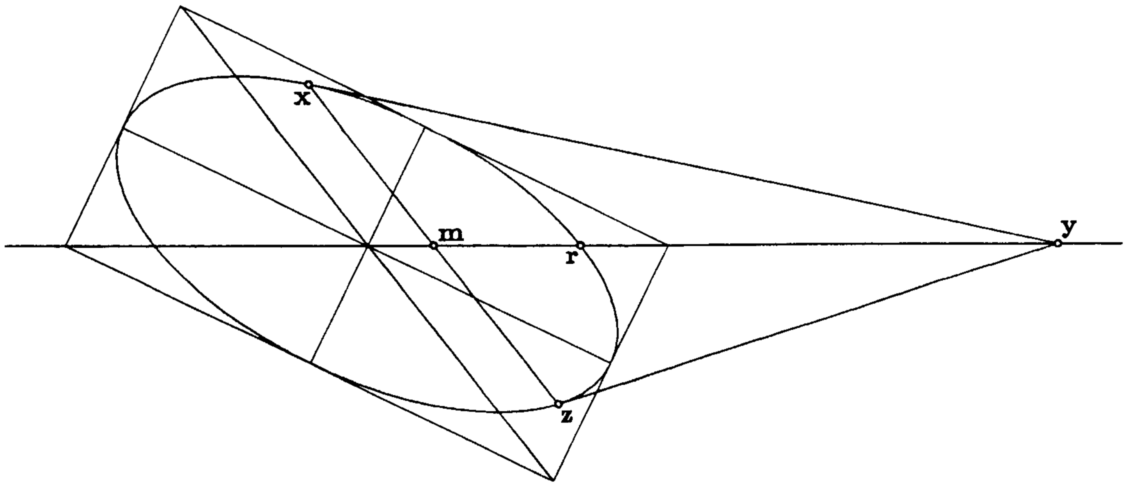


Fig. 25.2. The ellipse of minimum eccentricity in the control triangle $\Delta\mathbf{xyz}$

ellipse of minimum eccentricity is characterized algebraically by the formula

$$\frac{v^2}{uw} = \frac{\|\mathbf{z}-\mathbf{x}\|^2}{2\left(\|\mathbf{y}-\mathbf{x}\|^2 + \|\mathbf{z}-\mathbf{y}\|^2\right)}.$$

A rational curve segment $F_{[*]}([\bar{s},\bar{t}])$ of degree n for $n \geq 3$ has $n+1$ Bézier tensors, each with its own weight. Multiplying all of the weights by a common scalar has no effect. Multiplying the weights by a sequence of scalars in geometric progression changes the parameterization, but not the shape. The remaining $n-1$ degrees of freedom in the weights actually give different shapes for $F_{[*]}([\bar{s},\bar{t}])$.

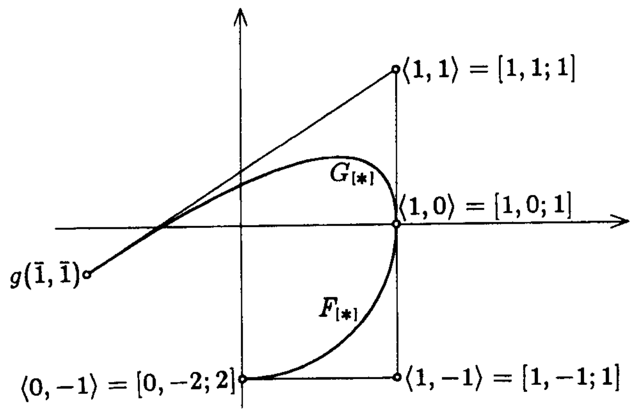
The convex-hull property carries over to the rational case, as long as all of the weights are positive. If we think of the scaled form $F([\bar{s},\bar{t}])$ as a polynomial curve in Q_*, the points $F(\bar{u})$ for $\bar{u}$ in $[\bar{s},\bar{t}]$ must remain in the convex hull of the Bézier tensors in Q_*. If all of the weights are positive, the projection in Q of this convex hull will be the same as the convex hull in Q of the Bézier points, which are the projections of the Bézier tensors.

The weight-based approach generalizes to rational surfaces of degree n or to birational surfaces of degree $(m;n)$ with no difficulty. We just draw a polynomial or bipolynomial surface in the linearized space Q_* by using weights to turn Bézier points into Bézier tensors, and then project down into the actual object space Q. For example, a quadratic rational surface has six Bézier tensors with respect to any reference triangle $\Delta\mathbf{rst}$ for its parameter plane. The six weight coordinates correspond to one irrelevant uniform scalar factor, two degrees of freedom in the choice of parameterization, and three degrees of freedom in the shape of the surface. Again, the convex-hull property will continue to hold, as long as all of the weights have the same sign.

In most cases, the weight-based approach is an effective compromise between thinking of the affine geometry of the polynomial scaled form F in Q_* versus thinking of the projective geometry of the rational map $F_{[*]}$ in $Q \subset Q_{[*]}$. The projection operation that transforms the former into the latter is simple enough that much of the structure that we analyzed in the polynomial case carries over to the rational case essentially intact. For example, since the projection of an osculating flat is the same as the flat that osculates the projection, we can apply the differential perspective on the meaning of a blossom's n arguments to compute a value of the multiprojective blossom of a rational map by intersecting osculating flats.

Despite the power of the weight-based approach, some designers are so wedded to the geometry of Q rather than that of Q_* that they aren't comfortable dealing with the flavor components of the Bézier tensors directly, even if those components are referred to as weights. Gerald Farin suggested an alternative approach for the case of curves [22]. Suppose that the designer begins by specifying all of the Bézier points $\mathbf{x}_i := f^{\otimes}_{[*]}(\bar{s}^{n-i}\bar{t}^{i})$, but none of the weights. At this point, we know $\mathbf{x}_0 = f^{\otimes}_{[*]}(\bar{s}^{n})$, and we know $\mathbf{x}_1 = f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{t})$. The line connecting $\mathbf{x}_0$ and $\mathbf{x}_1$ in $Q_{[*]}$ must consist of all points of the form $f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{u})$ for $\bar{u}$ in $L_{[*]}$, and the correspondence $\bar{u} \mapsto f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{u})$ must be a projective map. But we don't yet have enough information to compute the value $f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{u})$ for any $\bar{u}$ other than $\bar{s}$ and $\bar{t}$. One way to supply the missing information is to specify the ratio $w_0 : w_1$, where w_0 and w_1 are the weights of the Bézier tensors $e_0 = w_0\mathbf{x}_0$ and $e_1 = w_1\mathbf{x}_1$. Farin's alternative approach is to specify the blossom value $f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{u})$ for some third chosen value $\bar{q}$ of $\bar{u}$, perhaps $\bar{q} := (\bar{s}+\bar{t})/2$. Once three values of the projective map $\bar{u} \mapsto f^{\otimes}_{[*]}(\bar{s}^{n-1}\bar{u})$ are known, all other values can be computed by exploiting the fact that projective maps preserve cross ratios; equivalently, the ratio $w_0 : w_1$ can be computed. In a similar way, Farin suggests that the designer specify a third point somewhere along the ith edge of the control polygon to be the value $f^{\otimes}_{[*]}(\bar{s}^{n-i}\bar{q}\bar{t}^{i-1})$, for i in $[1,n]$. Sliding this point along the edge between $\mathbf{x}_{i-1} = f^{\otimes}_{[*]}(\bar{s}^{n-i}\bar{t}^{i})$ and $\mathbf{x}_i = f^{\otimes}_{[*]}(\bar{s}^{n-i-1}\bar{t}^{i+1})$ corresponds to adjusting the ratio w_i/w_{i-1} over the interval $(0,\infty)$. Once all the ratios $w_{i-1} : w_i$ are known for i in $[1,n]$, then all the weights are known up to a uniform constant multiple, which doesn't matter.

It is interesting to observe that Farin's technique does not generalize gracefully to surfaces. Let $F_{[*]}$ be a quadratic rational surface in Q; let F, with values in Q_*, be a quadratic polynomial scaled form of $F_{[*]}$; let $w(\mathbf{u},\mathbf{v}) := \mathrm{Flav}(f(\mathbf{u},\mathbf{v}))$ be the biaffine blossom of the weight coordinate of the scaled form F; and let $\triangle\mathbf{rst}$ be a reference triangle in the parameter plane P. Consider the three Bézier points $f_{[*]}(\mathbf{r},\mathbf{r})$, $f_{[*]}(\mathbf{r},\mathbf{s})$, and $f_{[*]}(\mathbf{r},\mathbf{t})$. Knowing those three points as points in Q is enough to determine the plane $\{f_{[*]}(\mathbf{r},\mathbf{u}) \mid \mathbf{u} \in P_{[*]}\}$ as a whole, but we need extra information to compute any particular point $f_{[*]}(\mathbf{r},\mathbf{u})$, unless $\mathbf{u} \in \{\mathbf{r},\mathbf{s},\mathbf{t}\}$. In the weight-based approach, that extra information comes from the ratios of the three corresponding weights $w(\mathbf{r},\mathbf{r})$, $w(\mathbf{r},\mathbf{s})$, and $w(\mathbf{r},\mathbf{t})$. By analogy with Farin's technique for curves, the designer could also supply that information by specifying the location of $f_{[*]}(\mathbf{r},\mathbf{q})$, where $\mathbf{q}$ is a fixed point in $\triangle\mathbf{rst}$, perhaps $\mathbf{q} := (\mathbf{r}+\mathbf{s}+\mathbf{t})/3$. But we get into trouble if try to apply this same trick to all three of the triangles that account for the first stage of the de Casteljau Algorithm. Specifying the location of $f_{[*]}(\mathbf{r},\mathbf{q})$ determines, among other things, the ratio $w(\mathbf{r},\mathbf{s}) : w(\mathbf{r},\mathbf{t})$; specifying



$\langle 1,1\rangle = [1,1;1]$
$G_{[*]}$
$\langle 1,0\rangle = [1,0;1]$
$g(\bar{1},\bar{1})$
$F_{[*]}$
$\langle 0,-1\rangle = [0,-2;2]$
$\langle 1,-1\rangle = [1,-1;1]$

Fig. 25.4. A two-segment conic spline curve

the location of $f_{[*]}(\mathbf{s},\mathbf{q})$ determines the ratio $w(\mathbf{r},\mathbf{s}) : w(\mathbf{s},\mathbf{t})$; and specifying the location of $f_{[*]}(\mathbf{t},\mathbf{q})$ determines $w(\mathbf{r},\mathbf{t}) : w(\mathbf{s},\mathbf{t})$. Hence, the locations of the three points $f_{[*]}(\mathbf{r},\mathbf{q})$, $f_{[*]}(\mathbf{s},\mathbf{q})$, and $f_{[*]}(\mathbf{t},\mathbf{q})$ are not independent. Once two of those three points are chosen, the third is constrained to lie along a line, not just in a plane.

There is one important area where the existence of the projection that takes a polynomial thing into a rational thing makes life more complicated, by opening up new possibilities: continuity conditions for rational splines. When deciding how to assemble pieces of polynomial maps to form a polynomial spline, we considered two different types of continuity constraints: parametric and geometric. Geometric continuity was a more appropriate constraint, in many cases. But parametric continuity was technically easier, since it interacted very neatly with blossoming. In the rational case, there are four different types of continuity constraints, since we can demand either parametric or geometric continuity either before or after the projection has happened.

Let's consider a simple case: assembling a two-segment conic spline in a plane A. Suppose that, for $\bar{u}$ in $[\bar{-1},\bar{0}]$, we have decided to follow the unit circle with its standard rational parameterization. That is, in the 3-space A_*, we shall follow the parabolic arc $F([\bar{-1},\bar{0}])$, where $F(\bar{u}) := \langle 1-u^2, 2u; 1+u^2\rangle$. The biaffine blossom f of F is given by $f(\bar{u},\bar{v}) := \langle 1-uv, u+v; 1+uv\rangle$. When F and f are projected down from Q_* into Q, they become the quarter-circle $F_{[*]}([\bar{-1},\bar{0}])$ shown in Fig. 25.4 and its biprojective blossom $f_{[*]}$. The three Bézier points of this quarter-circle are $f_{[*]}(\bar{-1},\bar{-1}) = \langle 0,-1\rangle$, $f_{[*]}(\bar{-1},\bar{0}) = \langle 1,-1\rangle$, and $f_{[*]}(\bar{0},\bar{0}) = \langle 1,0\rangle$, and the associated weights are 2, 1, and 1. (Plugging these values into the result of Ex. 25.3 shows that a circle, with eccentricity zero, is indeed an ellipse of minimum eccentricity.) Our task is to determine what conic segments $G_{[*]}([\bar{0},\bar{1}])$ can follow $F_{[*]}([\bar{-1},\bar{0}])$.

To keep things simple, let us demand that the curves F and G join with C^1 continuity at $\bar{u} = 0$, even before they are projected from A_* into A. This requirement fixes the first two Bézier tensors of $G([\bar{0},\bar{1}])$ at the values $g(\bar{0},\bar{0}) := \langle 1,0;1\rangle$ and $g(\bar{0},\bar{1}) := \langle 1,1;1\rangle$. We shall consider what various second-order continuity conditions imply about the value of the third Bézier tensor $g(\bar{1},\bar{1})$.

The strictest choice is to demand C^2 continuity before projection, which forces the parabolas F and G to be equal, giving $g(\bar{1},\bar{1}) = f(\bar{1},\bar{1}) = \langle 0,2;2\rangle$. After projection, the arc $G_{[*]}([\bar{0},\bar{1}])$ simply continues around the next quarter of the unit circle. In this case, the joint between F and G in A_* is surrounded by a de Casteljau Diagram with two shells. This diagram gets distorted by the projection, however, since the center vertex $f(\overline{-1},\bar{1}) = g(\overline{-1},\bar{1}) = \langle 2,0;0\rangle$ of the outer shell happens to be a vector on A, and hence projects to a point at infinity in $A_{[*]}$.

One possible weaker notion would be to demand G^2 continuity before projection. From the theory in Section 20, we can calculate that this leads to the constraint $g(\bar{1},\bar{1}) = \langle 0,2+a;2\rangle$, where the scalar a denotes the shape parameter β_2 at the joint. In A_*, the joint between F and G is guarded by an A-frame with $\lambda = 2/(a+2)$, whose apex is located at $\langle (a+4)/(a+2), -a/(a+2); a/(a+2)\rangle$. When such a G is projected down into A, the result is an arc of an ellipse that ends on the y-axis at the point $\langle 0, 1+a/2\rangle$.

We get weaker continuity constraints if we ask for continuity only after projection, instead of before. Suppose that we want the curves $F_{[*]}$ and $G_{[*]}$, after projection, to join with C^2 continuity. That is, we will allow the planes containing F and G in A_* to be tilted with respect to each other, hence precluding any chance of second-order continuity before projection, as long as that tilt is washed out by the projection down into A. A little algebra reveals that this constrains the third Bézier tensor be of the form $g(\bar{1},\bar{1}) = \langle b,2;2+b\rangle$, where b is a parameter that measures the amount of tilt.

The weakest of the four notions is to demand G^2 continuity after projection, which is just curvature continuity of the resulting conic spline. In this case, the third Bézier tensor $g(\bar{1},\bar{1}) := \langle b, 2+a; 2+b\rangle$ is affected both by the shape parameter a and by the tilt parameter b. After projection, the resulting conic segment $G_{[*]}([\bar{0},\bar{1}])$ will end at the point $g_{[*]}(\bar{1},\bar{1}) = \langle b/(2+b), (2+a)/(2+b)\rangle$, which could be anywhere in the plane. Choosing b in the range $(-1,\infty)$ gives arcs of ellipses that end with x in $(-1,1)$. Choosing $b=-1$ gives arcs of parabolas that end with $x=-1$; the example drawn in Fig. 25.4 happens to have $b=-1$. Choosing b in $(-2,-1)$ gives arcs of hyperbolas that end with x in $(-\infty,-1)$. Choosing b in $(-\infty,-2)$ causes the points $g_{[*]}(\bar{0},\bar{0})$ and $g_{[*]}(\bar{1},\bar{1})$ to be on opposite branches of a hyperbola, since the third Bézier tensor $g(\bar{1},\bar{1})$ then has a negative weight.

Whenever the tilt parameter b is nonzero, there is no A-frame in the space A_*, since the tilt causes the tangent lines $\mathrm{Osc}_1 F(\overline{-1})$ and $\mathrm{Osc}_1 G(\bar{1})$ to be skew. The corresponding lines after projection, which are $\mathrm{Osc}_1 F_{[*]}(\overline{-1})$ and $\mathrm{Osc}_1 G_{[*]}(\bar{1})$, do intersect, giving something that looks rather like an A-frame except that its apex has two different weights associated with it.

The expedient approach to drawing a rational spline curve or a rational spline surface in a space Q is to draw a polynomial spline curve or spline surface with C^k continuity in Q_*, and then project it. But this approach is somewhat sloppy. Even if the parameterization of the resulting rational spline is important to us, C^k continuity after projection would presumably be sufficient, rather than before projection. If we are simply modeling a smooth shape, then G^k continuity after projection should be enough. The problem with these weaker notions of continuity is that they don't interact well with blossoming.

Challenge 25.5. Use the differential perspective on the meaning of a blossom's n arguments to define the blossom of a rational spline curve that has G^k continuity after projection. In particular, if $h: L_{[*]} \times L_{[*]} \to A_{[*]}$ is the blossom of the two-segment conic spline in Fig. 25.4, the point in $A_{[*]}$ that forms the apex of the apparent A-frame should be labeled $h(\overline{-1}, \bar{1})$, even if a nonzero tilt causes that point to have two different weights. The resulting blossoms will fail to be multiprojective for two different reasons: nonzero shape parameters, as in Challenge 20.6, and nonzero tilts. Construct an enlightening theory of such blossoms.

Challenge 25.6. Develop useful techniques for controlling rational spline curves and surfaces whose joints are G^k continuous after projection.

26. Multiprojectivity in the weak sense

In Section 24, we defined multiprojectivity by saying that a map $f_{[*]}: (P_{[*]})^n \to Q_{[*]}$ is multiprojective if it has a multilinear scaled form, that is, if there exists a multilinear map $f_*: (P_*)^n \to Q_*$ with the property that the identity $f_{[*]}([e_1], \ldots, [e_n]) = [f_*(e_1, \ldots, e_n)]$ holds for almost all n-tuples $(e_1, \ldots, e_n)$ in $(P_*)^n$. The analogy between the words "multiprojective" and either "multilinear" or "multiaffine" suggests a definition along the following lines instead. A map $f_{[*]}$ should be multiprojective if the value $f_{[*]}([e_1], \ldots, [e_n])$ is a projective function of each argument whenever all of the other arguments are held fixed at arbitrary values. For example, for any fixed values of $[e_2]$ through $[e_n]$, the map $[e_1] \mapsto f_{[*]}([e_1], \ldots, [e_n])$ should be projective, meaning that it should have a linear scaled form: a linear map $g: P_* \to Q_*$ that makes the identity $f_{[*]}([e_1], \ldots, [e_n]) = [g(e_1)]$ hold for almost all e_1. The definition suggested by this analogy is more liberal than the definition that we gave in Section 24, because the scaled form g is allowed to depend in an arbitrary way on the fixed arguments $[e_2]$ through $[e_n]$. In this section, we shall explore this weaker notion of multiprojectivity, contrasting it with the stronger notion of Section 24. Multiprojectivity in the strong sense is the interesting concept from the point of view of practical applications in computer-aided geometric design. But there is intriguing mathematics involved in exploring the weak sense. The exploration of this mathematics was joint work with my colleague James B. Saxe.

We shall concentrate on the simplest case of multiprojectivity: given a function $f_{[*]}: L_{[*]} \times L_{[*]} \to L_{[*]}$, whose two arguments and result value all lie in the projective line $L_{[*]}$, we shall ask whether $f_{[*]}$ is biprojective in either the weak or the strong sense. For brevity, let us refer to a function, like $f_{[*]}$, from the cross product of the projective line with itself to the projective line, as a ***pairing***. If $f_{[*]}(x, y)$ is a pairing, the maps $x \mapsto f_{[*]}(x, y)$ for fixed y and $y \mapsto f_{[*]}(x, y)$ for fixed x are called ***sections*** of $f_{[*]}$. The basic intuitions behind the two types of biprojectivity are as follows: a pairing is ***weakly biprojective*** if all of its sections are projective, that is, have linear scaled forms; a pairing is ***strongly biprojective*** if it has a bilinear scaled form.

Much of the complexity of weak multiprojectivity comes from the handling of undefined values. In Section 24, we adopted a particular approach towards undefinedness in projective and rational functions, which might be called the *almost-everywhere approach.* In this approach, a projective function is considered to be defined only almost everywhere, in the sense of integration theory. The values of such a function can be changed arbitrarily on any set of measure zero without affecting the identity of the function. As a result, the formula that relates a projective function $F_{[*]}$ to one of its scaled forms F_* is required to hold only almost everywhere; that is, we require $F_{[*]}([e]) = [F_*(e)]$ only for almost all e. Unfortunately, if we follow the almost-everywhere approach, it turns out that the concept of weak biprojectivity is much weaker than one might think, as demonstrated by the monstrosity $g_{[*]}$ in the upcoming Example 26.1.

We shall use x, y, and z to denote points in the line $L_{[*]}$, each of which is given in homogeneous coordinates in the form $x = [x_1; x_0]$, where x_0 and x_1 are real numbers, not both zero. From the affine-world point of view, we have $x = x_1/x_0$. In the almost-everywhere approach, a pairing $f_{[*]}(x,y)$ is weakly biprojective if and only if, for each fixed y, there exist four constants a, b, c, and d, not all zero, such that the equation $f_{[*]}(x,y) = [ax_1 + bx_0; cx_1 + dx_0]$ holds for almost all x, and similarly with x and y interchanged. A pairing $f_{[*]}(x,y)$ is strongly biprojective if and only if there exist eight absolute constants a, b, c, d, p, q, r, and s, not all zero, such that the equation

$$f(x,y) = [ax_1y_1 + bx_1y_0 + cx_0y_1 + dx_0y_0;\ px_1y_1 + qx_1y_0 + rx_0y_1 + sx_0y_0]$$

holds for almost all pairs (x,y).

Example 26.1. To construct this example, we assume the truth of the Continuum Hypothesis and of the Axiom of Choice; in particular, we assume that the real numbers can be well-ordered by the elements of Ω, where Ω denotes the smallest uncountable ordinal. If the real numbers can be so well-ordered, certainly the projective line $L_{[*]}$ over the reals can be also. Pick such a well-ordering, and consider the set S of all pairs (x,y) in $L_{[*]} \times L_{[*]}$ with the property that the index of x is less than the index of y. For each fixed y, the set of x values with (x,y) in S is countable, and hence a set of measure zero in $L_{[*]}$. On the other hand, for each fixed x, the set of y values with (x,y) in S has a countable complement. Define a pairing $g_{[*]}$ by the rules

$$g_{[*]}(x,y) := \begin{cases} [1;1] & \text{if } (x,y) \text{ is in } S \\ [2;1] & \text{otherwise.} \end{cases}$$

For each fixed x, we have $g_{[*]}(x,y) = [1;1]$ for almost all y; for each fixed y, we have $g_{[*]}(x,y) = [2;1]$ for almost all x. Hence, in the almost-everywhere approach, $g_{[*]}$ is weakly biprojective—in fact, weakly biconstant. But $g_{[*]}$ certainly isn't strongly biprojective; it isn't even measurable, as a bivariate function.

The pairing $g_{[*]}$ of Example 26.1 shows that, in the almost-everywhere approach, the notion of weak biprojectivity is too weak to be very interesting. We could try to strengthen the notion by adding a second clause to the definition, which would somehow constrain the behavior of the pairing $f_{[*]}(x,y)$ as a function of two variables. But, once we are forced to constrain $f_{[*]}(x, y)$ as a bivariate function, rather than just constraining its sections, strong biprojectivity becomes the natural notion.

There is an entirely different approach to the problem of undefinedness in projective and rational functions, which we could have adopted in Section 24 instead of the almost-everywhere approach; we shall call it the ***indeterminate-point approach.*** In this approach, we invent a special point, called the ***indeterminate point,*** and we add it by fiat to every projective space. The formerly undefined symbol $[0]$ is then reinterpreted as naming the indeterminate point. With projective spaces augmented in this way, we can demand that the relation $F_{[*]}([e]) = [F_*(e)]$ between a projective map and its linear scaled form must hold for all e, without exception. In particular, the case $e = 0$ demands that $F_{[*]}([0]) = [0]$; that is, a projective map $F_{[*]}: P_{[*]} \rightarrow Q_{[*]}$ must take the indeterminate point of its domain $P_{[*]}$ to the indeterminate point of its codomain $Q_{[*]}$. For $e \neq 0$, the relation $F_{[*]}([e]) = [F_*(e)]$ demands that $F_{[*]}$ carry a ***determinate point*** $[e]$ to $[0]$ precisely when the scaled form F_* collapses the line $[e]$ through the origin of P_* into the origin $[0]$ of Q_*. In the indeterminate-point approach, we similarly demand that the identity relating a strongly multiprojective map to its multilinear scaled form must hold without exception.

While the almost-everywhere approach tries to ignore undefined values as much as possible, the indeterminate-point approach builds undefinedness, and the precise locations of the undefined values, into the concept of a projective map. For example, a function $F_{[*]}: L_{[*]} \rightarrow L_{[*]}$ that is equal to a constant $F_{[*]}(x) := w = [w_1; w_0]$ almost everywhere is as constant as a function can be in the almost-everywhere approach, and is also projective in that approach. In the indeterminate-point approach, in order for $F_{[*]}$ to be projective, it must satisfy the identity $F_{[*]}(x) = [F_*(x)]$ for some linear scaled form F_* and for all x. If we want $F_{[*]}$ to be as constant as possible, our best choice is to have $F_{[*]}(x) := [w_1(ax_1 + bx_0); w_0(ax_1 + bx_0)]$ for some real constants a and b, not both zero. This gives us $F_{[*]}(x) = w$ for all x except for the two special cases $x = [-b; a]$ and $x = [0; 0] = [0]$, where we have $F_{[*]}([-b; a]) = F_{[*]}([0]) = [0]$. Thus, in order to be considered projective in the indeterminate-point approach, an almost-everywhere-constant map $F_{[*]}$ must carry precisely one determinate argument point to the indeterminate result point; furthermore, the identity of that special argument point is a distinguishing characteristic of the function $F_{[*]}$.

Remark 26.2. The restrictions placed on almost-everywhere-constant functions $F_{[*]}: L_{[*]} \rightarrow L_{[*]}$ if they want to be projective in the indeterminate-point approach have a certain charm. The facts that there is a unique x with $x \neq [0]$ but $F_{[*]}(x) = [0]$ and that this x must be specified mean essentially that the polynomial scale factor that converts the constant function $F_{[*]}$ from degree zero to degree one must be specified, up to a constant multiple. If converting to the indeterminate-point approach meant that all polynomial scale factors were automatically specified up to a constant multiple, we could eliminate the thick dividing line around the "rational map" box in Fig. 24.3. Unfortunately, that hope is not realized, because of the problem of multiple roots. Suppose that $F_{[*]}(x) := [0]$ for x in $\{[0], u, v\}$, while $F_{[*]}(x) := w \neq [0]$ for all other x; and suppose that we want to consider $F_{[*]}$ as a degenerate case of a rational cubic curve. The cubic polynomial scale factor must have either u as a single root and v as a double root, or vice versa; but there is no way to tell which. Thus, we would still need the thick dividing line in Fig. 24.3, even if we converted completely to the indeterminate-point approach. (This example would work even over the complex numbers. Over the reals, things are worse: the map defined by $F_{[*]}([0]) := [0]$ and $F_{[*]}(x) := w \neq [0]$ for $x \neq [0]$ can be viewed as a degenerate quadratic rational curve by using, as scale factor, any quadratic polynomial that has no real roots.)

The good news is that, in the indeterminate-point approach, the monstrous pairing $g_{[*]}$ of Example 26.1 is neither weakly nor strongly biprojective. The bad news is that the indeterminate-point approach has its own monstrosities, like the following example, invented by Jim Saxe.

Example 26.3. Let $\pi\colon L_{[*]} \to L_{[*]}$ be an arbitrary bijective map, preferably rather chaotic. Define the pairing $h_{[*]}$ as follows. If either x or y is the indeterminate point $[0]$, then $h_{[*]}(x,y) := [0]$. Also, if $y = \pi(x)$, then $h_{[*]}(x,y) := [0]$. Otherwise, $h_{[*]}(x,y) := [1;1]$. The function $h_{[*]}$ is weakly biprojective in the indeterminate-point approach, since each section parallel to either the x or y axis is the constant $[1;1]$ except for a single indeterminate value. For example, if we fix x, we can write $h_{[*]}(x,y) = [\pi(x)_0 y_1 - \pi(x)_1 y_0; \pi(x)_0 y_1 - \pi(x)_1 y_0]$, where $\pi(x) = [\pi(x)_1; \pi(x)_0]$; the affine-world version of this formula might be more perspicuous:

$$h_{[*]}(x,y) = \frac{y - \pi(x)}{y - \pi(x)}.$$

If y is fixed, we have the similar formula

$$h_{[*]}(x,y) = \frac{x - \pi^{-1}(y)}{x - \pi^{-1}(y)}.$$

On the other hand, the function $h_{[*]}$ is not strongly biprojective, since there is too much information in π to be repesented by eight coefficients. (Note that, in the almost-everywhere approach, the pairing $h_{[*]}$ is both weakly and strongly biprojective—in fact, both weakly and strongly biconstant.)

We have tried two different approaches to handling undefinedness, and, in both cases, the natural notion of weak biprojectivity turned out to be too weak to be interesting: monstrous pairings existed that were weakly, but not strongly, biprojective. If the only difference between weak and strong multiprojectivity is that the weak case includes various monstrosities, then it probably isn't worth bothering with the weak case. We could continue to test this hypothesis by exploring still more approaches to handling undefinedness. One interesting possibility would be to call a pairing weakly biprojective if almost all of its sections were projective in the indeterminate-point sense. But we won't; instead, we shall try to tackle the question from a different perspective. What happens, in the indeterminate-point approach, if we restrict our attention to the special class of pairings where no undefinedness arises at all? Let us call a pairing $f_{[*]}$ *total* if the value $f_{[*]}(x,y)$ is a determinate point whenever both x and y are determinate points.

Exercise 26.4. Prove that strongly-biprojective total pairings exist. In particular, prove that the pairing given affinely by the formula

$$f_{[*]}(x,y) := \frac{x+y}{1-xy}$$

is such a pairing. The projective version of this formula is $f_{[*]}(x,y) := [x_1 y_0 + x_0 y_1; x_0 y_0 - x_1 y_1]$; show that the two homogeneous coordinates on the right-hand side can be simultaneously zero over the real numbers only if either $x_0 = x_1 = 0$ or $y_0 = y_1 = 0$. Note that this result depends on the fact that the coefficient field is the real numbers. Over the complex numbers, no strongly-biprojective pairing can be total.

Restricting our attention to total pairings simplifies things substantially. In particular, consider a section of a weakly-biprojective total pairing $f_{[*]}$; say that we fix x and consider the section $y \mapsto f_{[*]}(x,y)$. By weak biprojectivity in the indeterminate approach, there must exist four constants a, b, c, and d, not all zero, with $f_{[*]}(x,y) = [ay_1 + by_0; cy_1 + dy_0]$ for all y. The equivalent formula in the affine world is

$$f_{[*]}(x,y) = \frac{ay+b}{cy+d}.$$

Note that this section will be total if and only if it is a *fractional linear map*, that is, if and only if $ad - bc \neq 0$. The weakly-biprojective pairing $f_{[*]}$ as a whole will be total if and only if all of its sections are fractional linear maps.

Exercise 26.5. Over the real numbers, prove that every weakly-biprojective total pairing is actually strongly-biprojective. Warning: the proof that I know is rather intricate, and exploits both the topological and algebraic properties of the real numbers.

The result in Exercise 26.5 deals yet another blow to the concept of weak multiprojectivity, by suggesting that there are no interesting pairings that are weakly, but not strongly, biprojective. While that may be true over the real numbers, Jim Saxe discovered that it fails in a fascinating way over another coefficient field, in particular, over the field $k = \mathbf{Z}/3\mathbf{Z}$. Over this field, there are precisely $4 \cdot 144 = 576$ weakly-biprojective total pairings, of which only $3 \cdot 144 = 432$ are strongly biprojective. We shall close this section (and this paper) by studying the case $k = \mathbf{Z}/3\mathbf{Z}$ in a series of exercises.

For the rest of this section, let k denote the coefficient field $\mathbf{Z}/3\mathbf{Z}$, which consists of $\{0, 1, -1\}$ under arithmetic modulo 3. Before we consider biprojective pairings over k, it is worth pointing out that the discrete nature and finite characteristic of k cause other theories in this paper to work out rather differently. One example is that the almost-everywhere approach doesn't make any sense when the coefficient field is finite and discrete. As a second example, note that computing an n-blossom involves dividing by various multinomial coefficients with n as their upper index. If n equals or exceeds the characteristic of the coefficient field k, then blossoming won't be possible in general.

Exercise 26.6. There are 27 set-theoretic functions $F\colon k \to k$, which are precisely the 27 polynomials of the form $F(x) := ax^2+bx+c$ for a, b, and c in k. Polynomials of degree 3 or more are not needed, because $x^3 \equiv x \pmod 3$. There are $3^8 = 6561$ triaffine functions $f\colon k^3 \to k$, of which 81 are symmetric. These 81 symmetric triaffine functions, however, have only 9 different diagonals among them. For example, the diagonal of the function $f(x_1, x_2, x_3) := x_1 + x_2 + x_3$ is identically zero, and hence agrees with the diagonal of the zero function $f(x_1, x_2, x_3) := 0$. Deduce that some of the quadratic functions over k don't have multiaffine 3-blossoms. In fact, the function $F(x) := ax^2+bx+c$ has a 3-blossom if and only if $a = 0$, in which case it has 9 different 3-blossoms.

Exercise 26.7. We now turn to pairings over k. An affine line L over k contains three points: $\bar{0}$, $\bar{1}$, and $\overline{-1}$. The linearization L_* of L is a vector space of dimension two over k, which contains 9 points. The projective line $L_{[*]}$ over k consists of four points, which we shall refer to sloppily as 0, 1, ∞, and -1, where:

$$\begin{aligned} 0 &= [0;1] = [0;-1] \\ 1 &= [1;1] = [-1;-1] \\ \infty &= [1;0] = [-1;0] \\ -1 &= [-1;1] = [1;-1] \end{aligned}$$

Prove that a set-theoretic map $H\colon L_{[*]} \to L_{[*]}$ is a fractional linear mapping if and only if H is a permutation on the four letters $\{0, 1, \infty, -1\}$. Conclude that a function $f_{[*]}\colon L_{[*]} \times L_{[*]} \to L_{[*]}$ is a weakly-biprojective total pairing if and only if the values of $f_{[*]}$ form a 4-by-4 Latin square.

Exercise 26.8. The structure of a pairing $f_{[*]}(x, y)$ is not changed in any essential way if we permute the names of the four values of x, or those of y, since every permutation is a fractional linear map. Show that such renamings can be used to partition the set of all weakly-biprojective total pairings into equivalence classes of size 144, where each equivalence class contains precisely one member that satisfies the normalization conditions $f_{[*]}(w, 0) = f_{[*]}(0, w) = w$ for w in $L_{[*]}$. Enumerate the 4-by-4 Latin squares by showing that there are precisely four of them that satisfy the normalization conditions, the four shown in Fig. 26.9.

-1	-1	0	1	∞
∞	∞	-1	0	1
1	1	∞	-1	0
0	0	1	∞	-1
$y \backslash x$	0	1	∞	-1

-1	-1	∞	1	0
∞	∞	0	-1	1
1	1	-1	0	∞
0	0	1	∞	-1
$y \backslash x$	0	1	∞	-1

-1	-1	∞	0	1
∞	∞	-1	1	0
1	1	0	-1	∞
0	0	1	∞	-1
$y \backslash x$	0	1	∞	-1

-1	-1	∞	1	0
∞	∞	-1	0	1
1	1	0	-1	∞
0	0	1	∞	-1
$y \backslash x$	0	1	∞	-1

Fig. 26.9. The four normalized Latin squares of order 4.

Exercise 26.10. By the rules of the indeterminate-point approach, a pairing $f_{[*]}$ is strongly biprojective if and only if there exist eight constants, a through d and p through s, not all zero, such that

$$f_{[*]}(x,y) = [ax_1y_1 + bx_1y_0 + cx_0y_1 + dx_0y_0;\ px_1y_1 + qx_1y_0 + rx_0y_1 + sx_0y_0]$$

for all x and y. Show that, if the normalization conditions also hold for $f_{[*]}$, then there must exist two constants, a and p, such that

$$f_{[*]}(x,y) = [ax_1y_1 + x_1y_0 + x_0y_1;\ px_1y_1 + x_0y_0]$$

for all x and y. The values of such a pairing are given in Fig. 26.11 as functions of a and p.

Exercise 26.12. If we assume that the pairing $f_{[*]}$ is total, as well as being strongly biprojective and normalized, an explicit check of cases reveals that only three possibilities for the two parameters a and p remain: we can have $p = 1$ and $a = \pm 1$ or we can have $p = -1$ and $a = 0$. These three cases correspond to three of the four weakly-biprojective total pairings given in Fig. 26.9. The fourth (bottom-right) pairing in Fig. 26.9 represents an equivalence class of 144 total pairings over $k = \mathbf{Z}/3\mathbf{Z}$ that are weakly biprojective, but not strongly biprojective.

Exercise 26.13. Find a simple property of a weakly-biprojective total pairing over k that can serve as a test of strong biprojectivity. Hint: Consider the cycle structures of the six permutations formed by picking two rows out of the Latin square.

y \ x	0	1	∞	-1
-1	-1	$\frac{a}{p-1}$	$\frac{a-1}{p}$	$\frac{a+1}{p+1}$
∞	∞	$\frac{a+1}{p}$	$\frac{a}{p}$	$\frac{a-1}{p}$
1	1	$\frac{a-1}{p+1}$	$\frac{a+1}{p}$	$\frac{a}{p-1}$
0	0	1	∞	-1

Fig. 26.11. The normalized, strongly-biprojective pairings over $\mathbf{Z}/3\mathbf{Z}$

Acknowledgments

In an effort of this magnitude, there are many people to thank.

Jim Saxe and Jorge Stolfi were my invaluable and reliable comrades in the development of these ideas, ever ready to listen to my latest half-baked idea, to indicate its problems, and to point me in the right direction. Jim Saxe has formidable courage, tenacity, and insight; Jorge Stolfi has a formidable instinct for the perfect definition and the best notation.

Carl de Boor taught me about splines in his wonderful book and, in the process, gave me a standard to strive for in clarity of explanation. He also told me about his corner-sectioning technique during a provocative conversation, and he sent me a copy of de Casteljau's notes.

Andrew Odlyzko showed me how to persevere on the factoring problem at the end of Section 10, even after I had to give up on the obvious approach. Bill Gosper coaxed Macsyma into solving that same factoring problem while I was still persevering, although the proof that Macsyma found would have been difficult to publish.

Tony DeRose intrigued me by his work on geometric continuity and sparked my interest in other spline problems.

Leo Guibas, Greg Nelson, and Neil Wilhelm served as valuable test audiences for many of these ideas.

Mary-Claire van Leunen answered countless small questions about writing and the English language that occurred to me as I wrote, and did so with clarity and insight.

Cynthia Hibbard produced the index and worried about the many details of production and printing. She believed in me when I said that I would finish the manuscript someday. She and Frances Maurier both did some editing of the text, and I had great fun reading and pondering their suggestions.

Jorge Stolfi drew the cartoon on the title page, and also lent me his drafting equipment for me to use on the figures.

John Warnock and the other folks at Adobe Systems Incorporated devised and implemented PostScript, which served me quite well in the production of the figures. Fig. 9.6 is a good case in point.

Donald Knuth taught me how to do research and also wrote TeX, the system that compiled this document.

Gaston Gonnet and his colleagues at Waterloo wrote Maple, the symbolic algebraic manipulation system that I used for such purposes as computing the formulas in Section 23. Andrei Broder installed and maintained Maple at SRC.

Bob Taylor, my boss at SRC, built a research laboratory where a one-day search for a good way to label some diagrams of splines could blossom into a three-year exploration of some interesting mathematics. He also looked the other way—in fact, even applauded—while I slighted my other responsibilities unconscionably in order to work on this paper.

Jim Saxe, Marc Brown, and Leo Guibas provided many helpful comments on drafts of the manuscript.

Thank you all.

References

[1] Brian A. Barsky and John C. Beatty. "Local control of bias and tension in beta-splines." *ACM Transactions on Graphics* **2**, 2 (April 1983), 109–134.

[2] Richard H. Bartels and John C. Beatty. "Beta-splines with a difference." Technical Report CS-83-40, Computer Graphics Laboratory, University of Waterloo, Ontario, Canada (May 1984).

[3] Wolfgang Böhm. "Generating the Bézier points of triangular splines." In Robert E. Barnhill and Wolfgang Böhm, editors, *Surfaces in Computer Aided Geometric Design*, pages 77–91, particularly page 77. North-Holland (1983).

[4] Wolfgang Böhm. "Smooth curves and surfaces." In Gerald E. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 175–184. SIAM (1987).

[5] Wolfgang Böhm, Gerald Farin, and Jürgen Kahmann. "A survey of curve and surface methods in CAGD." *Computer Aided Geometric Design* **1**, 1 (July 1984), 1–60.

[6] Elaine Cohen, Tom Lyche, and Larry L. Schumaker. "Algorithms for degree-raising of splines." *ACM Transactions on Graphics* **4**, 3 (July 1985), 171–181.

[7] Wolfgang Dahmen and Charles A. Micchelli. "Multivariate splines: a new constructive approach." In Robert E. Barnhill and Wolfgang Böhm, editors, *Surfaces in Computer Aided Geometric Design*, pages 191–215, particularly page 204. North-Holland (1983).

[8] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag (1978).

[9] Pages 4–12.

[10] Page 131.

[11] Pages 146–147.

[12] C. de Boor and R. DeVore. "Approximation by smooth multivariate splines." *Transactions of the AMS* **276**, 2 (April 1983) 775–788, particularly 783.

[13] Carl de Boor and Klaus Höllig. "B-splines without divided differences." In Gerald E. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 21–27. SIAM (1987).

[14] Paul de Faget de Casteljau. "Formes à Pôles: Courbes et Surfaces." Photocopied manuscript (June 1984).

[15] Anthony D. DeRose. *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design*. PhD thesis, University of California at Berkeley (1985). Also available from Berkeley as Technical Report UCB/CSD 86/255.

[16] Tony D. DeRose and Brian A. Barsky. "An intuitive approach to geometric continuity for parametric curves and surfaces." In *Proceedings of Graphics Interface '85*, Montreal (May 1985), 343-351.

[17] C. T. J. Dodson and T. Poston. *Tensor Geometry: The Geometric Viewpoint and its Uses*. Pitman, London (paperback edition, 1979).

[18] Pages 83 and 262.

[19] Exercise 7d, page 104.

[20] Heinrich Dörrie. *100 Great Problems of Elementary Mathematics: Their History and Solution*, pages 231–236. Translated by David Antin. Dover (1965).

[21] Gerald Farin. "Visually C^2 cubic splines." *Computer-aided design* **14**, 3 (May 1982) 137–139.

[22] Gerald Farin. "Algorithms for rational Bézier curves." *Computer-aided design* **15**, 2 (March 1983) 73–77.

[23] A. R. Forrest. "Computational geometry: achievements and problems." In Robert E. Barnhill and Richard F. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 17–44, particularly page 27. Academic Press (1974).

[24] P. O. Frederickson. "Generalized triangular splines." Math Report 7–71, Lakehead University, 1971. Cited in [12].

[25] Ronald N. Goldman. "Subdivision algorithms for Bézier triangles." *Computer-Aided Design* **15**, 3 (May 1983), 159–166. See also the letter to the editor from Wolfgang Böhm and Gerald Farin in **15**, 5 (September 1983), 260–261.

[26] T. N. T. Goodman and K. Unsworth. "Generation of β-spline curves using a recurrence relation." Computer Science Report 85/02, University of Dundee, Scotland (February 1985).

[27] Paul R. Halmos. *Finite-Dimensional Vector Spaces.* Springer-Verlag (1974).

[28] Exercise 6, page 38.

[29] Pages 138–139.

[30] Donald E. Knuth. *Fundamental Algorithms*, pages 50 and 480–482. Volume 1 of *The Art of Computer Programming.* Addison-Wesley (second edition, 1973).

[31] Serge Lang. *Algebra.* Addison-Wesley (second edition, 1984).

[32] Pages 501–502.

[33] Page 520.

[34] Pages 586–588.

[35] Lyle Ramshaw. "Béziers and B-splines as multiaffine maps." To appear in *Theoretical Foundations of Computer Graphics and CAD*, the proceedings of a NATO International Advanced Study Institute in Lucca, Italy during July, 1987. Springer-Verlag (1987).

[36] M. A. Sabin. *The use of piecewise forms for the numerical representation of shape.* Dissertation, Hungarian Academy of Sciences (1977). Cited in [3, 5].

[37] P. Sablonniére. "Spline and Bézier polygons associated with a polynomial spline curve." *Computer-Aided Design* **10**, 4 (July 1978), 257–261.

[38] Jorge Stolfi. "Oriented projective geometry." To appear in the proceedings of the ACM Symposium on Computational Geometry, Waterloo, Canada (June 1987).

[39] E. C. Zeeman. "The umbilic bracelet and the double-cusp catastrophe." In P. Hilton, editor, *Structural Stability, the Theory of Catastrophes, and Applications in the Sciences*, published as volume 525 of *Lecture Notes in Mathematics*, pages 328–366. Springer-Verlag (1976).

Index

Index of Notations

Degree bounds

n	the degree bound on a polynomial map	1
$(m;n)$	the degree bound on a bipolynomial map	69

Points and coordinates

$\langle u_1,\ldots,u_p\rangle$	the Cartesian coordinates of a point $\mathbf{u}$ in an affine space	10
$\langle u_1,\ldots,u_p;u_0\rangle$	the Cartesian coordinates of a point $\mathbf{u}$ in a linear space	27
$[u_1,\ldots,u_p;u_0]$	the homogeneous coordinates of a point in a projective space	142
$\bar{r}$	the point $\langle r\rangle$ with real coordinate r in the affine line L	13
$\grave{r}$ and $\acute{s}$	the points with coordinates r and s in the lines U and V	70

Maps and blossoms

F	a polynomial map between affine spaces	10
f	the multiaffine blossom of F	10
$f^{\otimes}$	the affine blossom of F	23
F_*	the homogeneous polynomial form of F	29
f_*	the multilinear blossom of F	29
$f_*^{\otimes}$	the linear blossom of F	28
$F_{[*]}$	a rational map between projective spaces	144
$f_{[*]}$	the multiprojective blossom of $F_{[*]}$	145
$f_{[*]}^{\otimes}$	the projective blossom of $F_{[*]}$	145
f_S	the shared values of the blossoms f_i for i in the set S	81
$f_{(r,s)}$	the shared values of those f_i for which $(\bar{t}_i,\bar{t}_{i+1})\cap(\bar{r},\bar{s})\neq\emptyset$	81

Spaces and tensors

P	an affine space	8
Φ	an X-polynomial on P	18
$X[P]$	the algebra of X-polynomials on P	18
$X_c^k[P]$	the k-homogeneous c-flavored X-polynomials on P	19
$\sim$	affine equivalence between X-polynomials	21
$[\Phi]$	an affine equivalence class of X-polynomials, that is, a tensor	21
$\mathcal{X}[P]$	the symmetric tensor algebra on P	21
$\mathcal{X}_c^k[P]$	the k-homogeneous c-flavored tensors on P	22
ρ	the normal-form map associated with an affine frame for P	21
β	the normal-form map associated with a linear basis for P_*	30
P_*	the linearization $\mathcal{X}_*^1[P]$ of P, formed by all 1-tensors on P	26
P_1	the affine space $P=\mathcal{X}_1^1[P]$ formed by all points in P	26
P_0	the linear space $\mathcal{X}_0^1[P]$ formed by all vectors on P	26
δ	the unit vector on the affine line L	40

SRC Reports