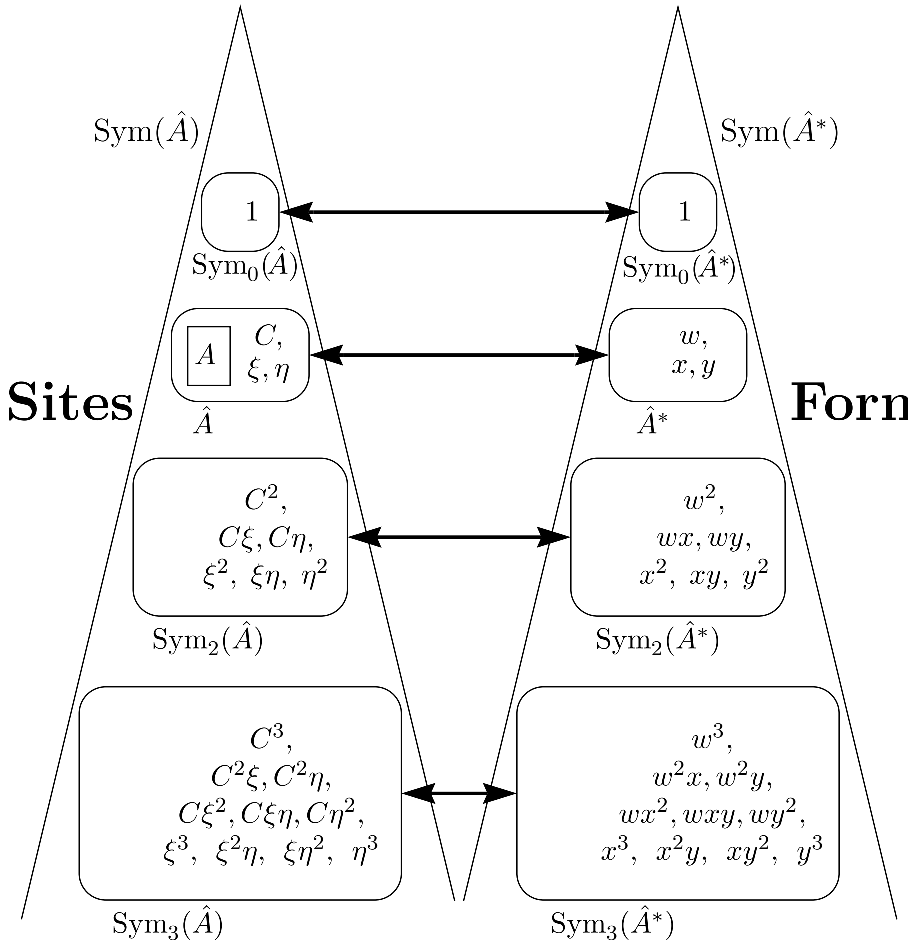
On Multiplying Points: The Paired Algebras of Forms and Sites

Lyle Ramshaw

May 1, 2001



Contents

List of Figures

Teaser

There is a multiplication operation on points that your teachers failed to tell you about, either because they didn't know about it or because they judged it to be unimportant. But that multiplication turns out to have important applications in computer-aided geometric design (CAGD). Among other things, it provides the best labels for Bézier control points — better even than the labels provided by polar forms (a.k.a. blossoms).

Let V be a finite-dimensional vector space. Everyone understands that it makes sense to multiply covectors, the elements of the dual space $V^* = \mathrm{Lin}(V, \mathbf{R})$. For example, if x, y, and z are covectors, then the expression $x^2 - 5yz$ denotes a *quadratic form* on the space V. Forms have lots of applications; for example, to put a Euclidean metric on V, we would choose a positive definite quadratic form as our measure of squared length.

But most people don't yet realize that it also makes sense to multiply vectors, the elements of V itself. If ρ, σ, and τ are vectors, then the expression $\rho^2 - 5\sigma\tau$ denotes an object that is the dual analog of a quadratic form. Let's call such an object a *quadratic site* over V. The sites over V of all degrees form an algebra, dual to the well-known algebra of forms on V.

What are sites good for? Consider, say, a cubic Bézier curve segment. It is the image, under a cubic function, of a closed interval on the parameter line, say the interval $[R \mathrel{..} S]$. The best labels for the Bézier points of that cubic segment are the cubic sites R^3, R^2S, RS^2, and S^3.

Preface

In computer-aided geometric design (CAGD), a beautiful technology has emerged for manipulating algebraic curves and surfaces, associated with the names Bernstein, Bézier, de Casteljau, and de Boor. I have spent an embarrassing fraction of the last fifteen years exploring the roots of that technology, trying to clarify the mathematics at its core.

I made some progress in the late 1980s by exploiting functions that I christened *blossoms*. I later learned that those functions already had a name: *polar forms*. I also learned that, in much of my work with polar forms, I had been following in de Casteljau's footsteps. But putting aside issues of terminology and priority, the key point is that polar forms make things clearer. They give us a labeling scheme for Bézier control points in which the labels perspicuously encode the geometry. This sparked new discoveries: Dahmen, Micchelli, and Seidel used polar forms to construct elegant bases for multivariate spline spaces over arbitrary triangulations [11].

But I suspected early on that polar forms were not the whole truth in this area. To evaluate the polar form of an n-ic, we take n input points and combine them, with concatenation, into a sequence of length n. Surely it would be better to combine those n points with some flavor of multiplication, rather than concatenation; but what flavor? That is, how should we multiply two points in this context? For some years, I mistakenly believed that tensors would be essential in constructing the proper flavor of multiplication. I wasn't far wrong; one way to think of the proper multiplication is as a symmetrized variant of the tensor product. But there is a better way to think of it.

Over the last few years, I finally realized that duality is the key to the proper multiplication on points — the duality of finite-dimensional linear spaces, where every linear space has an associated dual space and where the relationship between primal and dual is a symmetric one. How is duality relevant? Well, the dual of a point is a linear form; and we all know how to multiply two linear forms, producing a quadratic form. Suppose that we multiply two points using that same technique, but on the other side of the duality. We produce a quadratic object that is the dual analog of a quadratic form. Aha! That is the proper way to multiply two points in this context.

Site is the name that I propose for the dual analog of a form. So a point is a linear site. The product of two linear sites is a quadratic site, just as the product of two linear forms is a quadratic form. Indeed, we have two whole algebras, each the dual of the other; the algebra of forms is familiar, but the algebra of sites has been heretofore unfairly ignored. By recognizing and exploiting both algebras, we repair the flawed symmetry between primal and dual in CAGD and we finally arrive at an explanation of the Bézier technology that feels, to me, like a whole truth.

Lyle Ramshaw
lyle.ramshaw@compaq.com
May 1, 2001

Chapter 1

Introduction

This monograph repairs a flaw, quite low in the conventional mathematical underpinnings of CAGD. Such flaws show up only rarely; so I want to begin by pointing out the flaw, using as little machinery as possible.

1.1 Multiplying points

Let A be a finite-dimensional affine space, equipped with a Cartesian coordinate system. For concreteness, let's focus on the case in which A is an affine plane and let's refer to the two axes in the Cartesian coordinate system for A as x and y. So there is a one-to-one correspondence between points in the plane A and pairs of real numbers. If we think of x and y as functions from the plane A to the reals, the coordinates of any point P in A are the real numbers $x(P) = x_P$ and $y(P) = y_P$.

1.1.1 Question 1

Does it make sense to multiply x and y? People typically answer yes. Given the real-valued functions $x\colon A \to \mathbf{R}$ and $y\colon A \to \mathbf{R}$, we can multiply them pointwise to get the function $xy\colon A \to \mathbf{R}$ defined by $xy(P) := x(P)y(P)$.

Indeed, objects like the product xy are familiar enough to have acquired their own name; they are *quadratic forms* on the plane A. Each quadratic form on A can be written $ax^2 + bxy + cy^2 + dx + ey + f$, for some six real coefficients a through f.[†] Recall that a conic section in the plane A is the zero-set of a quadratic form on A. Even simpler, a line in A is the zero-set of a *linear form* on A, which can be written $ax + by + c$.[‡] Just as we can

[†]Please pardon my temporary sloppiness. More precisely, an n-form is a polynomial that is homogeneous of degree n, so the true quadratic form here is the homogenized polynomial $ax^2 + bxy + cy^2 + dxw + eyw + fw^2$. See Section 1.4 and Chapter 4.

[‡]I am being analogously sloppy; the true linear form here is $ax + by + cw$.

multiply the two coordinate functionals x and y, we can multiply any two linear forms:

(1.1-1)
$$\begin{pmatrix} ax & +by \\ & +c \end{pmatrix} \begin{pmatrix} dx & +ey \\ & +f \end{pmatrix} = \begin{pmatrix} adx^2 & +(ae+bd)xy & +bey^2 \\ & +(af+cd)x & +(bf+ce)y \\ & & +cf \end{pmatrix}$$

A quadratic form that is produced by multiplication in this way is rather special, of course; its zero-set is a reducible conic, the union of two lines.

1.1.2 Question 2

Given two points P and Q in A, does it make sense to multiply P and Q? People typically answer no. Of course, there are various special flavors of multiplication that arise in special contexts.

- If P and Q were actually the complex numbers $P = x_P + iy_P$ and $Q = x_Q + iy_Q$, their complex product would be the complex number $PQ = (x_P x_Q - y_P y_Q) + i(x_P y_Q + y_P x_Q)$.
- If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ were vectors in a 2-dimensional inner-product space, then their dot product would be the real number $P \cdot Q = x_P x_Q + y_P y_Q$.
- If $P = (x_P, y_P, z_P)$ and $Q = (x_Q, y_Q, z_Q)$ were vectors in a 3-dimensional Euclidean space, then their cross product would be the vector $P \times Q = (y_P z_Q - z_P y_Q,\ z_P x_Q - x_P z_Q,\ x_P y_Q - y_P x_Q)$.

Outside of such special situations, however, people typically don't assign any meaning to the product PQ of two points.

1.1.3 The Flaw

If you answered yes to Question 1 and no to Question 2, then the duality of linear algebra is broken for you. The points P and Q are elements of a certain linear space $\hat{A}$ that we discuss in Section 1.4, while the linear forms $ax + by + c$ and $dx + ey + f$ are elements of the dual space $\hat{A}^*$. If it makes sense to multiply two linear forms — and it manifestly does — then, by the symmetry of duality, it must make equal sense to multiply two points.

1.1.4 The Repair

It does indeed make perfect sense to multiply points. It was a regrettable oversight that your teacher failed to explain this to you. Fortunately, we can correct that oversight without inventing any new mathematical techniques. The proper technique to use for multiplying the points P and Q is the same technique that we are already familiar with for multiplying linear forms:

(1.1-2)
$$\begin{pmatrix} x_P & & y_P \\ & 1 & \end{pmatrix} \begin{pmatrix} x_Q & & y_Q \\ & 1 & \end{pmatrix} = \begin{pmatrix} x_P x_Q & & (x_P y_Q + y_P x_Q) & & y_P y_Q \\ & (x_P + x_Q) & & (y_P + y_Q) & \\ & & 1 & & \end{pmatrix}.$$

This rule for points is dual to the rule for linear forms in Equation 1.1-1. The plus signs that are missing from this rule are explained in Section 1.4, as are the extra 1's that appear on the left-hand side, acting like third coordinates for the points P and Q.

While we don't need a new mathematical technique to multiply points, we do need a new name; let's refer to the dual analog of a form as a *site*. So the object denoted by Equation 1.1-2, the product PQ of the two points P and Q, is a *quadratic site* over the plane A. Points in the plane A are linear sites over A. There are constant sites, linear sites, quadratic sites, cubic sites, and so forth: a whole algebra of sites over A, dual to the well-known algebra of forms on A.

Note that a quadratic site over the plane A has six coordinates, just as a quadratic form on the plane A has six coefficients. Thus, the product of two points is not itself a point, nor is it a scalar, nor a vector; rather, it is an object of a new type. "Quadratic site" is a name for that new type.

1.2 Labeling Bézier control points

By constructing the algebra of sites in parallel with the algebra of forms, we restore symmetry to duality, repairing the flaw pointed out in Section 1.1.3. But sites have another important benefit for us in CAGD: They are the key to the clearest labeling scheme that I know of for Bézier control points. Since we have been talking about multiplying two points in a plane A, let's first consider a quadratic Bézier triangle $F(\triangle QRS)$, as shown in Figure 1.1.

Our modeled objects will sit in some affine space; let's refer to that space as our *object space* and denote it O. Suppose that the function $F\colon A \to O$ maps the parameter plane A to some surface in O, with each coordinate of the varying point $F(P)$ being given by a polynomial in the coordinates $x(P)$ and $y(P)$ of total degree at most 2 — that is, being given by a quadratic form on the plane A. The image $F(A)$ is then a parametric surface in the object space O, out of which we are cutting a triangular surface patch $F(\triangle QRS)$. Such a surface patch is called a *quadratic Bézier triangle.*

The Bézier triangle $F(\triangle QRS)$ has six control points, which are most

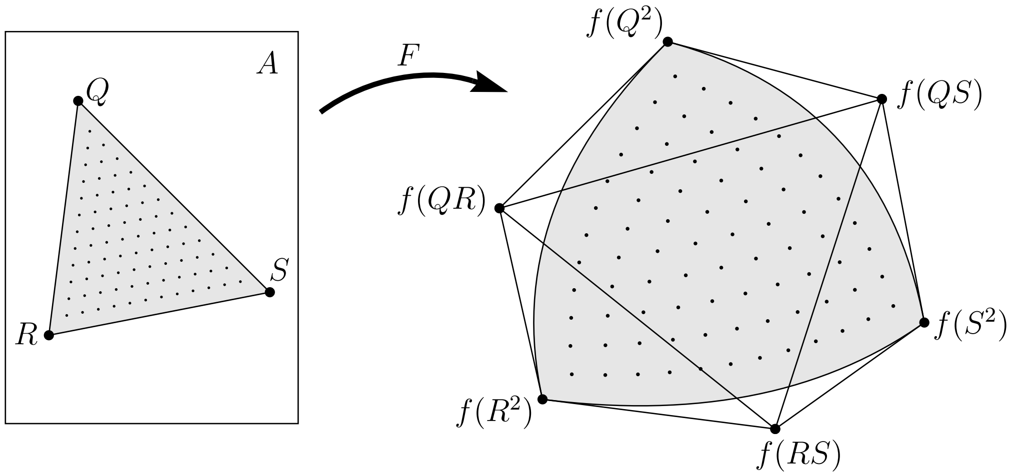


Figure 1.1: A quadratic Bézier triangle

clearly labeled

$$\begin{array}{ccccc} & & f(Q^2) & & \\ & f(QR) & & f(QS) & \\ f(R^2) & & f(RS) & & f(S^2). \end{array}$$

In these labels, the arguments to the function f are quadratic sites over the plane A; for example, Q^2 is the square of the point Q. Let σ_2 denote the squaring map on A, the map that takes each point P in the plane A to its square: $\sigma_2(P) := P^2$. (The projective completion of this map σ_2 is called the *Veronese surface* in algebraic geometry.) Any quadratic polynomial surface can be written in a unique way as an affine transform of the prototypical surface σ_2, and the affine map f in our labels is the instancing transformation involved when we so write the particular surface F. We thus have $F(P) = f(\sigma_2(P)) = f(P^2)$, for all points P in A.

These labels encode the geometric relationships among the Bézier points in a way that makes the de Casteljau Algorithm almost obvious. For the example point $T := Q/6 + R/3 + S/2$ in the triangle $\triangle QRS$, Figure 1.2 shows how the de Casteljau Algorithm computes the point $F(T) = f(T^2)$ from the six Bézier points of the patch $F(\triangle QRS)$ by doing four 2-dimensional affine interpolations. Consider the quadratic sites Q^2, QR, QS, and QT. You may not be too sure, as yet, what sites really are. But surely it must follow from $T = Q/6 + R/3 + S/2$ that $QT = Q(Q/6 + R/3 + S/2) = Q^2/6 + QR/3 + QS/2$. Since the map f is affine, we then have $f(QT) = f(Q^2)/6 + f(QR)/3 + f(QS)/2$, which justifies the uppermost interpolation. The other interpolations are justified similarly, multiplying the equation $T = Q/6 + R/3 + S/2$ by R, by S, and, for the final interpolation, by T.

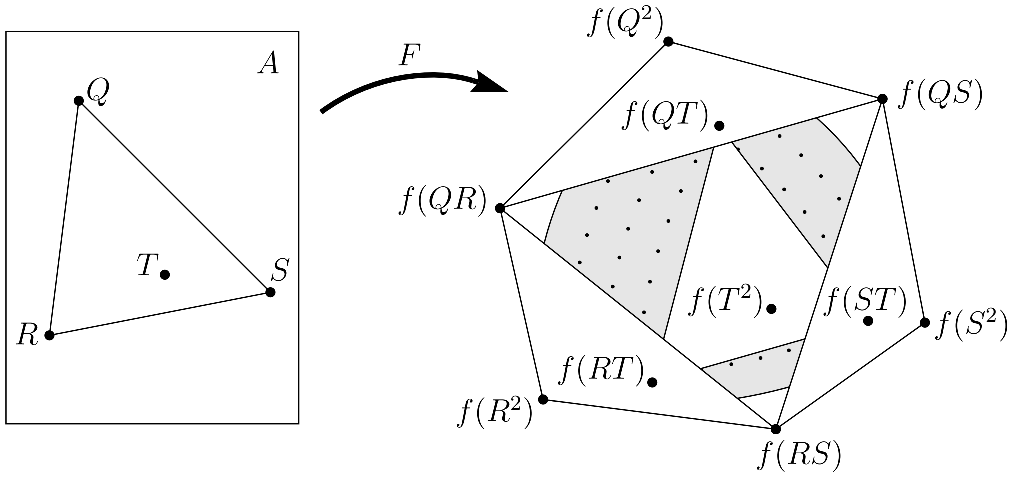


Figure 1.2: Computing a point on a quadratic Bézier triangle

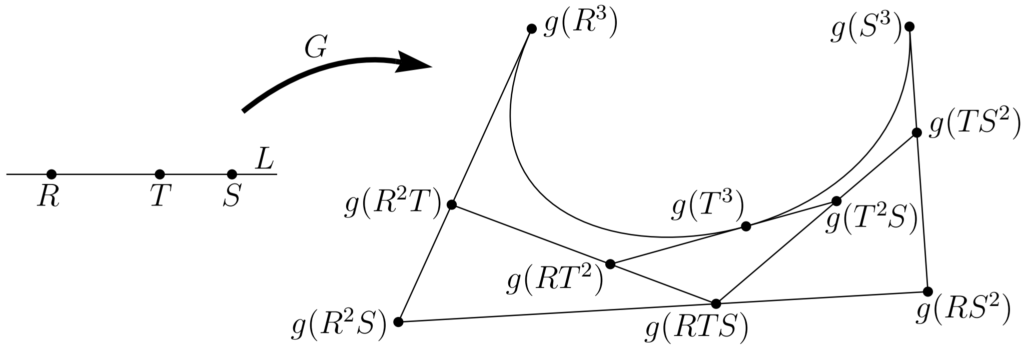


Figure 1.3: Computing a point on a cubic Bézier segment

For a Bézier curve, the parameter plane A is replaced by a parameter line L, and we get analogous labels by multiplying the points on L. Let the function $G\colon L \to O$ have the property that each coordinate of the varying point $G(P)$ is given by, say, a cubic form on L. The image $G(L)$ is then a cubic curve in the object space O, typically twisted. From that curve, we cut out the Bézier cubic segment $G([R \mathinner{..} S])$. Letting T be the example point $T := \frac{2}{5}R + \frac{3}{5}S$ on the parameter line L, Figure 1.3 shows the de Casteljau Algorithm computing the point $G(T) = g(T^3)$ from the four Bézier points $G(R) = g(R^3)$, $g(R^2S)$, $g(RS^2)$, and $G(S) = g(S^3)$ of the cubic segment $G([R \mathinner{..} S])$. The arguments to the map g are cubic sites over the line L, while the affine map g itself is the instancing transformation that realizes the particular curve G as an affine image of the prototypical cubic curve, the curve κ_3 given by $\kappa_3(P) := P^3$ for all points P on L.

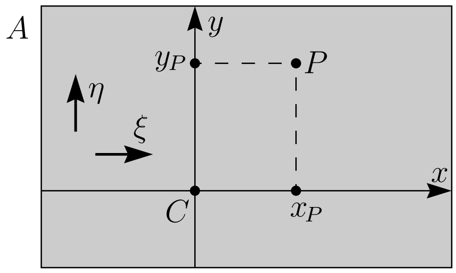


Figure 1.4: A point P in the affine plane A

1.3 Adding points

Let's return to the plane A of Section 1.1, with its Cartesian coordinate system (x, y) and its points P and Q. In Equation 1.1-2, we proposed a rule for multiplying P and Q, a rule in which each of the points P and Q is given three coordinates, rather than two; the point P, for example, has the coordinates $(x_P, y_P, 1)$. To explain where that third coordinate of 1 comes from, let's put aside the question of how to multiply points for a moment and take up the more elementary question of how to add them. Given two points P and Q, does their sum $P + Q$ make sense? The answer is tied up with the distinction between linear spaces and affine spaces.

If the plane A were a linear space (a.k.a. a vector space), with the point $(0, 0)$ as its origin, then we could add two vectors in A simply by adding their x and y coordinates separately; we would have $P+Q = (x_P, y_P)+(x_Q, y_Q) = (x_P + x_Q, y_P + y_Q)$. More precisely, letting $\xi := (1, 0)$ and $\eta := (0, 1)$ denote the unit vectors in the x and y directions, we would have $P = x_P\xi + y_P\eta$ and $Q = x_Q\xi + y_Q\eta$, and hence $P + Q = (x_P + x_Q)\xi + (y_P + y_Q)\eta$.

But we introduced the plane A as an affine space, and we referred to its elements P and Q as points, rather than vectors. Recall that an affine space is like a linear space, but without an origin. In an affine space, the midpoint $P/2 + Q/2$ of a line segment $\overline{PQ}$ is a well-defined point, as is, for any t, the point $(1 - t)P + tQ$ that lies t of the way from P to Q. But the sum $P + Q$ of two points is not well-defined.

To see why not, let $C := (0, 0)$ be the point at the center of our Cartesian coordinate system for the plane A, as shown in Figure 1.4. Since the plane A is affine, the center point C is a point like any other; it has no special role. In particular, we do not have $C = 0$. For any point P in A, however, we do have the equation $P - C = x_P\xi + y_P\eta$; that is, the point P differs from the center point C by the vector $P - C$, and the coordinates x_P and y_P of P are the coefficients that express that vector $P - C$ as a linear combination of the unit vectors ξ and η. So any point P in A can be expressed in the

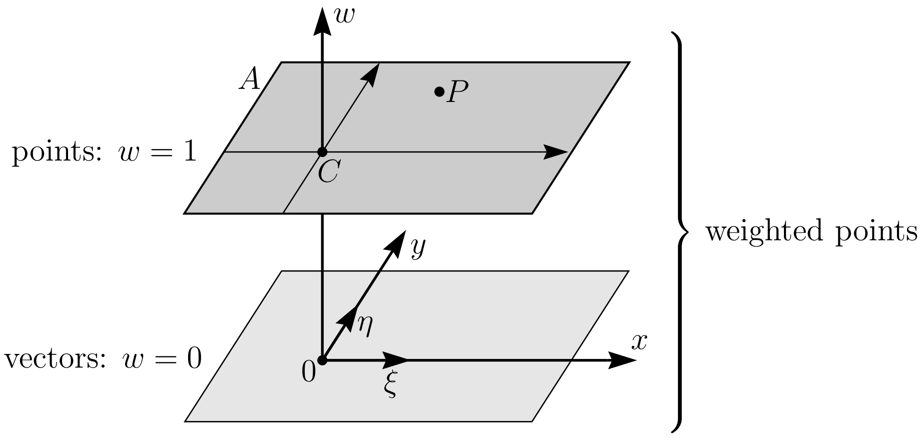


Figure 1.5: The linearization $\hat{A}$ of the affine plane A

form $P = x_P\xi + y_P\eta + C$, as a linear combination of ξ, η, and C in which the coefficient of C is 1. That restriction on the coefficient of C explains the difficulty that arises when we add two points in an affine space. In the affine plane A, for example, we have $P = x_P\xi + y_P\eta + C$ and $Q = x_Q\xi + y_Q\eta + C$, and hence $P + Q = (x_P + x_Q)\xi + (y_P + y_Q)\eta + 2C$. The sum $P + Q$ is not a point in A because the coefficient of C is not 1.

1.4 Linearization

The spaces that arise in CAGD are often affine. For example, the parameter space of a polynomial Bézier curve or surface is affine. Note that, when we used our plane A as the parameter plane of a quadratic Bézier surface in Section 1.2, the center point $C = (0, 0)$ played no special role.

But linear spaces have simpler algebraic properties. They are closed under addition and scalar multiplication as separate operations. Also, it is linear spaces that have associated dual spaces. So it is worth considering whether we can somehow convert an affine space into a linear space.

Fortunately, there is a well-known conversion method, called *linearization* (a.k.a. *homogenization*). My teachers told me about it, and I hope that your teachers told you as well. When we linearize an affine space, we embed it as an affine hyperplane in a linear space of the next larger dimension. For example, we linearize the affine plane $A = \{x\xi + y\eta + C \mid x, y \in \mathbf{R}\}$ by extending it into the linear 3-space $\hat{A} = \{x\xi + y\eta + wC \mid x, y, w \in \mathbf{R}\}$, as shown in Figure 1.5, thereby removing the restriction that the coefficient of C be 1. An element p of the linearized space $\hat{A}$ has three coordinates $p = (x_p, y_p, w_p) = x_p\xi + y_p\eta + w_pC$. Such an element p is typically called a *weighted point*, where w_p is the *weight*. (I prefer the term *anchor*; but let's save that discussion for later.) A point, such as P, is a weighted point of weight 1; a vector, such as ξ or $P - C$, is a weighted point of weight 0; and a sum of two points, such as $P + Q$, is a weighted point of weight 2.

Linearization enlarges an affine space A of points into a linear space $\hat{A}$ of weighted points, thereby making it legal to add two points and legal to multiply a point by a scalar. But linearization, by itself, does not make it legal to multiply two points. To do that, we must enlarge the space A even further: into the algebra $\mathrm{Sym}(\hat{A})$, as we discuss in Section 1.6.

While linearization doesn't take us all the way to the algebra $\mathrm{Sym}(\hat{A})$, it does clear up some issues that we left dangling in Section 1.1; in particular, it supplies the linear space to which we apply duality. Recall that we were discussing how to multiply two points P and Q in the affine plane A. Let $\hat{A}$ be the linearization of the plane A, which is the linear 3-space of weighted points shown in Figure 1.5. The dual of $\hat{A}$ is the space $\hat{A}^* = \mathrm{Lin}(\hat{A}, \mathbf{R}) = \mathrm{Aff}(A, \mathbf{R})$ of linear forms on A. This dual space $\hat{A}^*$ is also 3-dimensional, a typical element of it being written either as $ax + by + c$, when viewed as belonging to $\mathrm{Aff}(A, \mathbf{R})$, or as $ax + by + cw$, when viewed as belonging to $\mathrm{Lin}(\hat{A}, \mathbf{R})$. Whichever way the forms in $\hat{A}^*$ are written, everyone agrees that it makes sense to multiply them as polynomials, using the obvious rule

$$\begin{pmatrix} ax & & + by \\ & + cw & \end{pmatrix} \begin{pmatrix} dx & & + ey \\ & + fw & \end{pmatrix} = \begin{pmatrix} adx^2 & & + (ae + bd)xy & & + bey^2 \\ & + (af + cd)xw & & + (bf + ec)yw & \\ & & + cfw^2 & & \end{pmatrix}.$$

By duality, it makes equal sense to multiply two weighted points p and q in $\hat{A}$ as polynomials, using the analogous rule

$$\begin{aligned} &\begin{pmatrix} x_p\,\xi & & + y_p\,\eta \\ & + w_p\,C & \end{pmatrix} \begin{pmatrix} x_q\,\xi & & + y_q\,\eta \\ & + w_q\,C & \end{pmatrix} \\ &\qquad = \begin{pmatrix} x_p x_q\,\xi^2 & & + (x_p y_q + y_p x_q)\xi\eta & & + y_p y_q\,\eta^2 \\ & + (x_p w_q + w_p x_q)\xi C & & + (y_p w_q + w_p y_q)\eta C & \\ & & + w_p w_q\,C^2 & & \end{pmatrix}. \end{aligned}$$

In particular, for two points P and Q in A, with weights $w_P = w_Q = 1$, we compute the quadratic site PQ that is their product via the rule

$$\begin{aligned} &\begin{pmatrix} x_P\xi & & + y_P\eta \\ & + C & \end{pmatrix} \begin{pmatrix} x_Q\xi & & + y_Q\eta \\ & + C & \end{pmatrix} \\ &\qquad = \begin{pmatrix} x_P x_Q\xi^2 & & + (x_P y_Q + y_P x_Q)\xi\eta & & + y_P y_Q\eta^2 \\ & + (x_P + x_Q)\xi C & & + (y_P + y_Q)\eta C & \\ & & + C^2 & & \end{pmatrix}. \end{aligned}$$

This is precisely the rule for the product of the two points P and Q that we first saw in Equation 1.1-2; but the missing plus signs have now been filled in, revealing polynomials in ξ, η, and C, while the extra 1's have been revealed to be weight coordinates.

1.5 Multiplying vectors

Linearization solves the problems of addition and scalar multiplication, but not the problem of multiplication. To make sure that we understand the multiplication problem on its own, free from any extraneous issues associated with linearization, let's go through Section 1.1 again, but considering linear spaces from the outset. That is, instead of multiplying the points in an affine space, let's consider multiplying the vectors in a linear space.

Let X be a finite-dimensional real linear space, say of dimension k and with basis $(\xi_1, \ldots, \xi_k)$. So every vector ζ in X is a linear combination $\zeta = x_1\xi_1 + \cdots + x_k\xi_k$ of the basis vectors ξ_1 through ξ_k. Furthermore, the real coefficients x_1 through x_k in that linear combination are uniquely determined by ζ. Writing them as functions of ζ, we have

$$\zeta = x_1(\zeta)\xi_1 + \cdots + x_k(\zeta)\xi_k. \tag{1.5-1}$$

For each i in $[1 \mathinner{..} k]$, the function $x_i \colon X \to \mathbf{R}$ is linear; hence, x_i is a *covector*, an element of the dual space $X^* = \mathrm{Lin}(X, \mathbf{R})$. Indeed, the covectors $(x_1, \ldots, x_k)$ form the basis for X^* that is dual to the basis $(\xi_1, \ldots, \xi_k)$ for X; that is, we have the duality constraints

$$x_i(\xi_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

It follows that any covector z in X^* can be written uniquely as a linear combination of the basis covectors x_1 through x_k as follows:

$$z = z(\xi_1)x_1 + \cdots + z(\xi_k)x_k. \tag{1.5-2}$$

Equation 1.5-2 is dual to Equation 1.5-1. The two would look more alike if we had written Equation 1.5-2 as $z = \xi_1(z)x_1 + \cdots + \xi_k(z)x_k$. We chose to write the i^{th} coefficient as $z(\xi_i)$, rather than as $\xi_i(z)$, because people typically prefer to think of covectors as functions that take vectors as their arguments, rather than vice versa. In fact, those two points of view are equally valid. To avoid choosing between them, we can view both the vector and the covector, more symmetrically, as arguments to a *pairing map*, the bilinear map $\langle\ ,\ \rangle \colon X^* \times X \to \mathbf{R}$ that takes a covector z and a vector ζ to the real number $\langle z, \zeta\rangle = z(\zeta) = \zeta(z)$; we discuss this more symmetric point of view in Section 2.3.

Just as in Section 1.1, we are now faced with two questions:

Question 1: Does it make sense to multiply two covectors, say x_1 and x_2? Sure. The product x_1x_2 is a quadratic form on the vector space X. We can think of forms on X either syntactically or semantically. Syntactically, a form is simply a polynomial in the variables x_1 through x_k. Semantically, it is the real-valued function on X that results when the multiplications in such a polynomial are interpreted as pointwise multiplication of functions. For example, the quadratic form x_1x_2 on X is interpreted semantically as the function $x_1x_2 \colon X \to \mathbf{R}$ given, for all ζ in X, by $x_1x_2(\zeta) := x_1(\zeta)x_2(\zeta)$.

Question 2: Does it make sense to multiply two vectors, say ξ_1 and ξ_2? By duality, the answer must be yes. The product $\xi_1\xi_2$ is a *quadratic site* over X. We can think of sites over X either syntactically or semantically. Syntactically, a site is simply a polynomial in the variables ξ_1 through ξ_k. Semantically, it is the real-valued function on X^* that results when the multiplications in such a polynomial are interpreted as pointwise multiplication of functions. For example, the quadratic site $\xi_1\xi_2$ is interpreted semantically as the function $\xi_1\xi_2 \colon X^* \to \mathbf{R}$ whose value, at any covector z in X^*, is given by $\xi_1\xi_2(z) := \xi_1(z)\xi_2(z) = z(\xi_1)z(\xi_2)$.

When these theories get applied to practical situations, people tend to be more interested in real-valued functions on X than they are in real-valued functions on X^*, that is, more interested in forms than in sites. Indeed, forms are used extensively in many fields; in CAGD, for example, each coordinate of a parametric curve or surface is a form on the parameter space. Sites, on the other hand, have been used so little that they do not yet have a standard name. This monograph argues that sites are important in CAGD because they give us the best labels for Bézier points.

1.6 The paired algebras

We are going to repair the flaw in the underpinnings of CAGD pointed out in Section 1.1.3 as follows: Given any affine space A, we are going to supplement the well-known algebra of forms on A with the algebra of sites over A, thus producing a dual pair of algebras. Doing this takes three steps.

1.6.1 Linearization

The first step is the familiar process of linearization, as we discussed in Section 1.4. Linearizing the affine space A extends it into a linear space $\hat{A}$ of the next larger dimension. Once we have produced the linear space $\hat{A}$, we get its dual space $\hat{A}^*$ automatically, since every linear space has a dual.

1.6.2 Algebrization

The second step is similar in structure; let's call it *algebrization*.[§] Just as linearization extends an affine space A into a naturally associated linear space $\hat{A}$, so algebrization extends a linear space X into a naturally associated commutative algebra. Viewed abstractly, this algebra is called the *symmetric algebra* of X and is denoted $\mathrm{Sym}(X)$.

Actually, multilinear algebra provides at least four ways to algebrize a linear space X, that is, to extend X into a naturally associated algebra of some flavor [25, 36]:

- the tensor algebra $T(X) = \bigotimes X$,
- the symmetric algebra $\mathrm{Sym}(X) = S(X)$,
- the alternating (a.k.a. skew-symmetric, exterior, or Grassmann) algebra $\mathrm{Alt}(X) = \bigwedge X$,
- and, if a quadratic form on X has been chosen, thereby giving X a metric structure, the Clifford algebra $\mathrm{Clif}(X)$.

This monograph uses the symmetric algebra. Luckily, that one is the simplest of the four: the only one whose multiplication commutes and the only one that can be constructed using just polynomials, with no need for tensors.

By the way, many of these algebras have important applications in CAGD. Starting at the bottom, Clifford algebras have proven helpful in analyzing Pythagorean-hodograph (PH) curves — a problem in which the Euclidean metric plays a central role [10]. Alternating algebras have long been widely used; they give a good naming scheme for the subspaces of a linear space, and they underlie calculus on manifolds. Symmetric algebras have an even longer history, though they are seldom referred to by name; for example, the algebra of all forms on an affine space A is the symmetric algebra $\mathrm{Sym}(\hat{A}^*)$. This monograph argues that we in CAGD should supplement that famous algebra with its dual, the symmetric algebra $\mathrm{Sym}(\hat{A})$ of sites over A. As for the tensor algebra, I can't think of any application of the full tensor algebra in CAGD; but the multiplication in the tensor algebra is the asymmetric tensor product, which is what the phrase "tensor-product surface" refers to.

The symmetric algebra $\mathrm{Sym}(X)$ can be constructed in various ways, as we discuss in Chapter 5. One simple way uses polynomials: We choose a basis for the linear space X, say $(\xi_1, \ldots, \xi_k)$, and we then construct the symmetric algebra $\mathrm{Sym}(X)$ as the algebra $\mathbf{R}[\xi_1, \ldots, \xi_k]$ of all polynomials

[§] The verb should mean "to convert something into an algebra", rather than "to make something more algebraic"; this argues in favor of "algebrize" or "algebratize", rather than "algebraicize" or "algebraify". I am not fond of "algebrize"; but I like "algebrization", and I can't justify forming "algebratize" in the absence of the adjective "algebratic".

in the symbols ξ_1 through ξ_k, treated as variables. But choosing a basis as part of the construction, in this way, raises the fear that different choices might lead to algebras that differed in some important way. A more abstract and basis-independent way to construct the symmetric algebra $\mathrm{Sym}(X)$ is as the algebra $\mathrm{Poly}(X^*, \mathbf{R})$ of all real-valued, polynomial functions on the dual space X^*. A vector in X can be thought of as a linear functional on X^*, so a polynomial whose variables are vectors in X gives rise to a real-valued, polynomial function on X^*.

As our second step in building the paired algebras, we apply this process of algebrization, independently, to the linear spaces $\hat{A}$ and $\hat{A}^*$. We get two algebras with the same structure, one of which is an old friend:

$\mathrm{Sym}(\hat{A}) = \mathrm{Poly}(\hat{A}^*, \mathbf{R})$, the algebra of sites over A	and	$\mathrm{Sym}(\hat{A}^*) = \mathrm{Poly}(\hat{A}, \mathbf{R})$, the algebra of forms on A.

1.6.3 Choosing the pairing maps

Only one step remains, in building the paired algebras; but first, we need to discuss homogeneity. Given any vector space X, an element of the symmetric algebra $\mathrm{Sym}(X)$ is called *homogeneous of degree n* when the corresponding polynomial in $\mathbf{R}[\xi_1, \ldots, \xi_k]$ has every term of total degree precisely n or, equivalently, when the corresponding real-valued function $f\colon X^* \to \mathbf{R}$ satisfies $f(tz) = t^n f(z)$, for all covectors z in X^* and all real numbers t. In the symmetric algebra $\mathrm{Sym}(X)$, the elements that are homogeneous of degree n form a linear subspace, which we denote $\mathrm{Sym}_n(X)$. (Some authors use a superscript: $\mathrm{Sym}^n(X)$.) In our situation, those forms on A that are homogeneous of degree n constitute the linear space $\mathrm{Sym}_n(\hat{A}^*)$, while those sites over A that are homogeneous of degree n constitute $\mathrm{Sym}_n(\hat{A})$. We refer to the elements of these spaces as *n-forms* on A and *n-sites* over A.

So far, we have constructed the algebra of forms $\mathrm{Sym}(\hat{A}^*)$ and the algebra of sites $\mathrm{Sym}(\hat{A})$ as separate algebras, built from the dual linear spaces $\hat{A}^*$ and $\hat{A}$. Our third step makes those two algebras themselves into a dual pair by choosing, for each n, a pairing

$$\langle\ ,\ \rangle\colon\ \mathrm{Sym}_n(\hat{A}^*) \times \mathrm{Sym}_n(\hat{A}) \to \mathbf{R}$$

between n-forms and n-sites. Fixing such a pairing lets us represent a linear functional on n-forms as an n-site, and vice versa. For example, consider the evaluate-an-n-form-at-P functional, the dual functional ϵ_P in $\mathrm{Sym}_n(\hat{A}^*)^*$ that takes an n-form f in $\mathrm{Sym}_n(\hat{A}^*)$ as its argument and returns the real number $\epsilon_P(f) := f(P)$. With the pairing maps that I recommend, that linear functional is represented by the n-site $\epsilon_P = P^n/n!$. Warning: Some authors scale their pairing maps differently, so as to eliminate that denominator of $n!$. By doing so, they simplify their formulas for evaluation, but complicate their formulas for differentiation — as we discuss at length in Appendix B.

1.7 Piecewise models with smooth joints

Once we have built the paired algebras of forms and sites, what good are they? In a nutshell, they provide a tool for analyzing functions defined by polynomials. This tool is particularly effective at constraining two such functions so that they agree to a certain order somewhere. And that, in turn, is a key problem in spline theory, as we here review.

Computer-aided geometric design (CAGD) is a field of applied mathematics that studies ways to model and manipulate smooth, synthetic shapes. The standard techniques in CAGD involve breaking a shape up into pieces and modeling each piece algebraically. The word *spline* originally meant a flexible strip of wood; but it now refers to a great variety of clever ways to arrange that the joints between the pieces end up sufficiently smooth.

Suppose that O is our *object space*, the space in which we want our modeled shapes to sit. In CAGD, we typically model shapes in O either *parametrically* or *implicitly*. To model a shape S in O parametrically, we invent for ourselves an auxiliary space A, called the *parameter space*; we choose a function $\mathcal{F}\colon A \to O$, typically piecewise rational; and we then model S as the image $S := \mathcal{F}(A)$. To model a shape S in O implicitly, we invent an auxiliary space B, which might be called the *gauge space*; we choose a function $\mathcal{G}\colon O \to B$, typically piecewise polynomial; and we model S as the inverse image $S := \mathcal{G}^{-1}(0)$ of the origin in B. The ideas in this monograph are applicable to both parametric and implicit modeling. But parametric models of shapes are more common in CAGD today, so we shall use parametric models as our examples. When the parameter space A is 1-dimensional, the resulting shape $\mathcal{F}(A)$ is a *parametric curve*; when $\dim(A) = 2$, it is a *parametric surface*; when $\dim(A) = d$, it is a *parametric d-fold.*

The parametric d-folds used in CAGD are typically either piecewise-rational or piecewise-polynomial, the latter being a special case of the former. If $\mathcal{F}\colon A \to O$ is a piecewise-polynomial parametric d-fold, then the spaces A and O are taken to be affine. For a polynomial piece F of $\mathcal{F}$, say of degree n, each coordinate of the output point $F(P)$ in O is given by an n-form on A. The piecewise-rational case is similar, except that we add one additional n-form, serving as a common denominator. More precisely, the spaces A and O are taken to be projective and each homogeneous coordinate of the output point $F(P)$ is given by an n-form on A. In Section 4.6, we mention how one completes an affine space A into its projective closure by ignoring scalar multiples in the linearized space $\hat{A}$. For the bulk of this monograph, however, we restrict ourselves to the polynomial case.

Let $\mathcal{F}\colon A \to O$ be a piecewise-polynomial parametric d-fold and let F be one of its pieces. Since F is given by polynomials, it extends to a polynomial function $F\colon A \to O$, defined on all of A. For example, Figure 1.6 shows the graph of a function from $\mathbf{R}$ to $\mathbf{R}$ that is built up from four pieces, each a

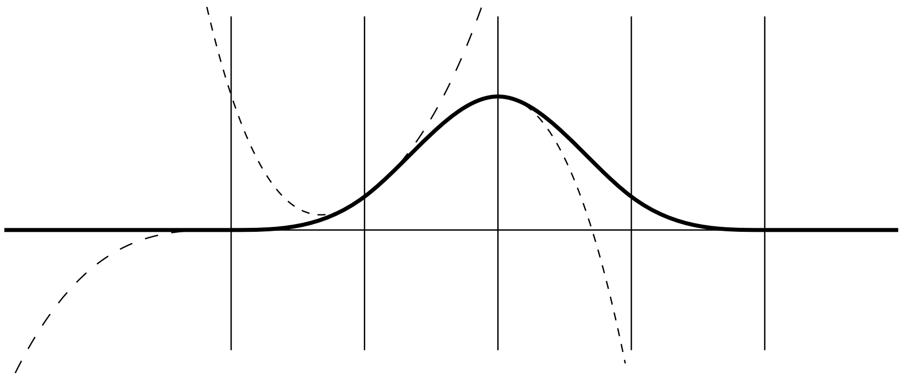


Figure 1.6: A cubic spline curve with four segments

segment of a cubic polynomial. It also shows what happens when the first two of those segments are extended beyond their endpoints. (The other two are symmetric.) In this example, each adjacent pair of polynomials agrees to second order at the joint between them, leading to an overall curve that is twice continuously differentiable.

One of the key problems in spline theory is achieving smooth joints. The paired algebras assist in that quest through the following result, which we discuss in Sections 6.7 and 7.11:

> For any k in $[0..n]$, two n-forms f and g on an affine space A agree to k^{th} order at a point P in A just when we have $\langle f, s\rangle = \langle g, s\rangle$ for all n-sites s over A that are multiples of P^{n-k}. (The angle brackets here denote the pairing between n-forms and n-sites.)

The full implications of this result are subtle, but we can easily check out the extreme cases. Letting $k := n$, the two n-forms f and g agree to n^{th} order at P just when $\langle f, s\rangle = \langle g, s\rangle$ for all n-sites s over A that are multiples of $P^0 = 1$, that is, for all n-sites s over A. Thus, f and g agree to n^{th} order at P just when they coincide. Letting $k := 0$, the forms f and g agree to 0^{th} order at P just when $\langle f, s\rangle = \langle g, s\rangle$ for all n-sites s that are multiples of P^n, that is, when $\langle f, P^n\rangle = \langle g, P^n\rangle$. This also makes sense, since we have seen that $\langle f, P^n\rangle/n! = \langle f, P^n/n!\,\rangle = \epsilon_P(f) = f(P)$, and similarly for g.

1.8 Cubic Bézier triangles

It turns out that polynomial parametric surfaces in 3-space of total degree at most 3 are general enough to motivate much of what we do. In what follows, we shall often use that class of surfaces, called *cubic Bézier triangles*, as a convenient example. Indeed, we have already used quadratic Bézier triangles as an example several times, including the one shown in Figure 1.1. But quadratics are a bit too special; cubics are more generic.

Let the object space O be affine 3-space, say with (x, y, z) as a Cartesian coordinate system, and suppose that we want to design a cubic polynomial parametric surface in O. We invent for ourselves an affine parameter plane A. Since we just agreed to use x, y, and z as the coordinates in the object space O, let's use u and v from now on as the names of the Cartesian coordinates in the parameter plane A. We define the x, y, and z coordinates of the varying point $F(u, v) = (F_x(u, v), F_y(u, v), F_z(u, v))$ as polynomials F_x, F_y, and F_z of total degree at most 3 in the variables u and v. The resulting function $F\colon A \to O$ is called a *cubic polynomial parametric surface*. The piece typically cut out of such a surface is a *cubic Bézier triangle*, the image $F(\triangle QRS)$ of a triangle $\triangle QRS$ in the plane A. The analog, for arbitrary degree n and parametric dimension d, is an *n-ic polynomial parametric d-fold*, out of which we cut an *n-ic Bézier d-simplex*.

1.9 Related work

We now discuss how the paired algebras relate to other work in CAGD, using a cubic Bézier triangle $F(\triangle QRS)$ as our example.

1.9.1 Bernstein bases and Bézier points

Bernstein bases and Bézier control points provide the common foundation for much of CAGD. Any point P in the parameter plane A can be written uniquely as a barycentric combination of the three points Q, R, and S; that is, we have $P = q(P)Q + r(P)R + s(P)S$, where q, r, and s are affine, real-valued functions on A and where $q(P) + r(P) + s(P) = 1$, for all points P in A. It then transpires that every cubic polynomial parametric surface $F\colon A \to O$ can be written uniquely in the form

$$F(P) = \sum_{i+j+k=3} \binom{3}{i\ j\ k} q(P)^i r(P)^j s(P)^k\, F_{i,j,k}, \tag{1.9-1}$$

for some ten control points

$$\begin{array}{ccccccc} & & & F_{3,0,0} & & & \\ & & F_{2,1,0} & & F_{2,0,1} & & \\ & F_{1,2,0} & & F_{1,1,1} & & F_{1,0,2} & \\ F_{0,3,0} & & F_{0,2,1} & & F_{0,1,2} & & F_{0,0,3} \end{array}$$

in O. The factor of $\binom{3}{i\ j\ k}$ in the summand is the *trinomial coefficient* given, for $i+j+k = n$, by $\binom{n}{i\ j\ k} = n!/(i!\ j!\ k!)$. The ten control points $(F_{i,j,k})$ are the

Bézier points of the Bézier triangle $F(\triangle QRS)$, while the ten corresponding coefficient functions $B_{i,j,k}\colon A \to \mathbf{R}$ given by

$$B_{i,j,k}(P) := \binom{3}{i\ j\ k} q(P)^i r(P)^j s(P)^k$$

constitute the *Bernstein basis* for the cubic polynomial functions on the plane A with the reference triangle $\triangle QRS$.

Things are much the same for any degree n and parametric dimension d. Given a reference d-simplex for a d-dimensional parameter space A, we get a Bernstein basis for the n-ic polynomial functions on A, and the coefficients of the functionals in that Bernstein basis are the Bézier points of the resulting n-ic Bézier d-simplex.

Assembling a spline d-fold out of pieces cut from polynomial d-folds is quite a subtle problem, once d exceeds 1. In the case of $d = 1$, however, de Casteljau, de Boor, and others built *B-splines*, a thoroughly satisfactory theory of spline parametric curves. Indeed, this theory of spline curves is so attractive that it is tempting to construct spline surfaces as curves of curves. The resulting surfaces are built from functions $F\colon A \to O$ whose defining polynomials obey separate degree bounds in u and in v; we discuss these *tensor-product surfaces* in Section 6.8. Because B-splines are such an effective way of dealing with spline curves, tensor-product spline surfaces have become the most popular surfaces in CAGD.

1.9.2 Bézier points as polar values

In de Casteljau's development of the theory of B-splines [14], he made good use of the classical notion of *polar forms*, referring to his B-spline control points as *poles*. I popularized his ideas under the name *blossoming* [42, 43]. The *polar form*, a.k.a. *blossom*, of a cubic Bézier triangle $F\colon A \to O$ is the unique symmetric, triaffine function $\tilde{F}\colon A^3 \to O$ that agrees with F on the diagonal, that is, that satisfies $\tilde{F}(P, P, P) = F(P)$, for all points P in A.

Polar forms are valuable in this context because they give us perspicuous names for many of the points that are associated with the surface F, but that don't lie on that surface itself. In particular, the ten Bézier control points $(F_{i,j,k})_{i+j+k=3}$ of the triangular patch $F(\triangle QRS)$ are the following values of its polar form $\tilde{F}$:

$$\begin{array}{ccccccc} & & & \tilde{F}(Q,Q,Q) & & & \\ & & \tilde{F}(Q,Q,R) & & \tilde{F}(Q,Q,S) & & \\ & \tilde{F}(Q,R,R) & & \tilde{F}(Q,R,S) & & \tilde{F}(Q,S,S) & \\ \tilde{F}(R,R,R) & & \tilde{F}(R,R,S) & & \tilde{F}(R,S,S) & & \tilde{F}(S,S,S) \end{array}$$

Furthermore, the symmetric, multiaffine nature of the polar form $\tilde{F}$ neatly encodes the geometry that underlies the essential algorithms, such as the de Casteljau Algorithm for subdivision.

Polar forms have proven useful to researchers in spline theory, as well as to teachers of it. By using polar forms, Dahmen, Micchelli, and Seidel [11] constructed elegant bases for multivariate spline spaces over arbitrary triangulations, thereby taking an important step toward generalizing the theory of B-splines from curves to surfaces.

By the way, the name "polar form" is good, because it points out the connection with the other places in mathematics where the technique of polarization is exploited. But the word "form" is used quite heavily already in this area of mathematics, most notably for the objects — quadratic forms, cubic forms, and the like — that make up the algebra of forms. In this monograph, simply to reduce our overloading of the word "form", let's refer to $\tilde{F}$ as the *blossom* of F.

1.9.3 From polarization to the paired algebras

Even in my early work on blossoming, I suspected that the n arguments to the blossom should be combined using some flavor of multiplication, rather than simply being concatenated into a sequence. But I wrongly believed that the symmetrized variant of the tensor-product construction would be an essential tool in defining the proper way to multiply points.

Ron Goldman's pioneering work on dual bases [23, 38] pushed me to think harder about duality, since I found it disturbing when he referred to two different bases for the same linear space as dual. I then realized that I had been mistaken: You don't need tensors to multiply points. Points are dual to linear forms, and you certainly don't need tensors to multiply forms. Rather, forms are essentially polynomials, and you multiply them as you would polynomials. This monograph argues that points are also essentially polynomials — to wit, linear sites. And the proper way to multiply sites is as you would multiply polynomials.

The ability to multiply points together to form sites provides a framework for the Bernstein/Bézier theory that is clearer and more convenient than blossoming. For example, the function that maps each point P in the plane A to the cubic site P^3 over A becomes a prototype for all possible cubic parametric surfaces. In algebraic geometry, that prototype is called the *Veronese surface of parametric degree* 3 [29]. That Veronese surface sits in a space of fairly high dimension — in fact, in a 9-space. But every cubic surface $F\colon A \to O$ in the 3-space O is simply an affine transform of that prototype; that is, we have $F(P) = f(P^3)$, for all points P in A, where f is an affine transformation from the 9-dimensional space $\mathrm{Sym}_3(\hat{A})^{\downarrow}$ of unit-weight 3-sites over A to the 3-space O.

Exploiting sites and the affine transformation f, we get the following simple formulas for the ten Bézier points of the cubic patch $F(\triangle QRS)$:

$$\begin{array}{ccccccc} & & & f(Q^3) & & & \\ & & f(Q^2R) & & f(Q^2S) & & \\ & f(QR^2) & & f(QRS) & & f(QS^2) & \\ f(R^3) & & f(R^2S) & & f(RS^2) & & f(S^3) \end{array}$$

Comparing this notation for the Bézier points to our previous two notations, we have

$$F_{i,j,k} = \tilde{F}(\underbrace{Q,\ldots,Q}_{i}, \underbrace{R,\ldots,R}_{j}, \underbrace{S,\ldots,S}_{k}) = f(Q^i R^j S^k)$$

whenever $i + j + k = 3$. The right-hand, site-based notation preserves all of the symmetric, multiaffine strengths of the middle, blossom-based notation, while restoring the brevity of the left-hand notation, in which the Bézier points are simply numbered.

The notation and concepts of the paired algebras are more powerful than their predecessors, as well as more concise. As an example of this power, consider Equation 1.9-1, the basic formula that expresses a point $F(P)$ on a cubic Bézier triangle as an affine combination of the ten Bézier points. Using sites, we can prove that formula with elementary algebra:

$$\begin{aligned} F(P) = f(P^3) &= f\big((q(P)Q + r(P)R + s(P)S)^3\big) \\ &= f\Big(\sum_{i+j+k=3} \binom{3}{i\ j\ k} q(P)^i r(P)^j s(P)^k\, Q^i R^j S^k\Big) \\ &= \sum_{i+j+k=3} \binom{3}{i\ j\ k} q(P)^i r(P)^j s(P)^k\, f(Q^i R^j S^k). \end{aligned}$$

1.9.4 Vegter exploits the contraction operators

The basic operators that interconnect the algebra of forms with the algebra of sites are the pairing maps; for each n, we can pair an n-form with an n-site to produce a real number. But there is also a richer family of interconnecting operators that can be defined from the pairing maps: the *contraction* operators [22, 25], which we discuss in Section 7.8. For any k in $[0 \mathinner{..} n]$, we can contract an n-form on a k-site to produce an $(n-k)$-form. Symmetrically, we can contract an n-site on a k-form to produce an $(n-k)$-site. Pairing is the special case $k = n$ of either of these flavors of contraction, since both 0-forms and 0-sites are simply real numbers.

Recently, Gert Vegter has been applying these contraction operators to problems in CAGD [47]; at least, that is how I would describe what he has been doing. He describes his work as applying the *apolar bilinear form*, an inner product on spaces of homogeneous multivariate polynomials that was used in 19^{th}-century invariant theory. I hope that Vegter will come to view the paired algebras as providing a cleaner foundation and a simpler notation for his fine work. Meanwhile, I view his work as encouraging evidence that the paired algebras will be broadly useful in CAGD, above and beyond giving us the clearest names for Bézier points.

Warning: I regret to report that Vegter uses the family of pairing maps that I first used, as opposed to the family that I now recommend. In the language of Appendix B, he uses the *averaged pairing*, while I currently recommend the *summed pairing*. The field of CAGD will avoid a lot of confusion if a clear winner emerges soon on this annoying question of where to put the factor of $n!$.

1.10 The four frameworks

The bulk of this monograph analyzes four different frameworks that can be used when studying problems in CAGD — for example, when devising new spline methods. Each of those frameworks gives names to the relevant linear spaces, stipulates various relationships between those spaces, and provides certain operators that interconnect those spaces.

The nested-spaces framework: After some mathematical preliminaries in Chapter 2, Chapter 3 discusses the *nested-spaces framework*, shown schematically on page 29. This is the naive framework that people often adopt when they first start working in CAGD. Such people typically view a quadratic polynomial, say in the variables u and v, as being a degenerate case of a cubic polynomial — degenerate in the sense that the coefficients of the u^3, u^2v, uv^2, and v^3 terms all happen to be zero. Thus, in this framework, the 6-dimensional linear space of all quadratic polynomials in u and v is viewed as a subspace of the 10-dimensional linear space of all cubics in u and v. That is the sense in which the spaces in this framework are nested.

The homogenized framework: Linearization leads to the *homogenized framework*, discussed in Chapter 4 and shown on page 41. This is the framework commonly used by researchers in CAGD today. They homogenize their polynomials; for example, rather than dealing with $u^2 - 3uv + 7v$ either as a quadratic polynomial in u and v or as a degenerate cubic in u and v, they instead deal either with the quadratic form $u^2 - 3uv + 7vw$ or with the cubic form $u^2w - 3uvw + 7vw^2$, where the weight variable w lets them express the point (u, v) in homogeneous coordinates as $[u : v : w]$. The resulting forms make up an algebra, the symmetric algebra $\mathrm{Sym}(\hat{A}^*)$ of forms on A.

The separate-algebras framework: In the *separate-algebras framework*, discussed in Chapter 5 and shown on page 50, the algebra $\mathrm{Sym}(\hat{A}^*)$ of forms on A is joined by its dual, the algebra $\mathrm{Sym}(\hat{A})$ of sites over A. But these two algebras remain separate, in the sense that we have not yet chosen a family of pairing maps; so we can't combine an n-form on A with an n-site over A to produce a real number. Even without such pairing maps, a framework that embraces the algebra of sites along with the algebra of forms has significant advantages. Chapter 6 attacks various questions in CAGD by analyzing the *Veronese prototypes*, the images of the perfect-power maps $P \mapsto P^n$, as geometric objects sitting inside the algebra of sites.

The paired-algebras framework: By choosing, for each n, a pairing between n-forms and n-sites, we arrive at our final goal: the *paired-algebras framework*, discussed in Chapter 7 and shown on page 82. The pairing maps and the contraction maps defined from them give us simple formulas for the evaluation and differentiation of n-forms. Chapter 8 reviews some basic concepts of CAGD in the light of the paired algebras, showing, among other things, that the dual of a Bernstein basis for the linear space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms is a *Bézier basis* for the linear space $\mathrm{Sym}_n(\hat{A})$ of n-sites.

Unfortunately, a question of convention raises its ugly head in defining the pairing maps: Do we divide by $n!$ or not? Each choice makes some formulas pretty, at the price of cluttering up others. I recommend not dividing by $n!$, which makes differentiation pretty at the price of cluttering up evaluation. Appendix B discusses the tradeoffs at length.

Universal mapping conditions: By this point in your reading, I hope to have convinced you that sites are important and useful in CAGD. But you may still feel uneasy about what sites really are — that is, what it really means to multiply points. One way to address that uneasiness is to apply duality strictly: If you are happy thinking of forms on A as real-valued functions on $\hat{A}$ of a certain type, then you should also be happy thinking of sites over A as real-valued functions on $\hat{A}^*$ of the analogous type. But universal mapping conditions provide a truer and deeper answer to the question of what sites really are. In Chapter 9, we discuss how to construct symmetric algebras, as well as tensor algebras, alternating algebras, and Clifford algebras, by means of universal mapping conditions.

If universal mapping conditions don't scare you off by being too abstract and formal, you might consider the next step toward formalized abstraction, which is category theory. Viewed from the perspective of category theory, both the linearization of affine spaces and the algebrization of linear spaces are left adjoints of forgetful functors. Appendix A discusses the mathematics that underlies this monograph from that still more abstract perspective.

Chapter 2

Mathematical Preliminaries

2.1 On the words "affine" and "linear"

The word "linear" is used inconsistently in mathematics: It sometimes implies homogeneity and sometimes doesn't. For example, the polynomial $f(x) := ax + b$ is called linear even when b is nonzero; but we must have $b = 0$ in order for the function $f\colon \mathbf{R} \to \mathbf{R}$ defined by $f(x) := ax + b$ to qualify as a linear map. We here adopt the convention that *linear* always implies homogeneous; when we mean "of degree 1, but not necessarily homogeneous", we use the term *affine*. For example, we say *affine interpolation*, where most people would say "linear interpolation".

We use the term *linear space* for the mathematical structure that is often called a *vector space*. While some linear spaces do indeed have vectors as their elements, many linear spaces have elements of other types: covectors, polynomials, or functions, for example.

An *affine space* is like a linear space, but without an origin. If P_1 through P_m are points in an affine space A, the linear combination $t_1P_1 + \cdots + t_mP_m$ denotes a point in A only when $t_1 + \cdots + t_m = 1$. Linear combinations whose coefficients sum to 1 in this way are called *affine combinations*. If A and B are affine spaces, a map $f\colon A \to B$ is *affine* when it preserves affine combinations, that is, when $t_1 + \cdots + t_m = 1$ implies $f(t_1P_1 + \cdots + t_mP_m) = t_1f(P_1) + \cdots + t_mf(P_m)$.

2.2 Finite dimensionality

For simplicity in our mathematical constructions, we restrict ourselves to the case of finite-dimensional spaces, either affine or linear, over the real numbers. That is the case of primary interest in CAGD, and it is also the case in which the theories of duality and of the symmetric algebra are at their simplest and prettiest. Much of those theories carries over to more general contexts:

linear spaces that are infinite-dimensional, linear spaces over fields of prime characteristic, even modules over commutative rings. But various intriguing subtleties arise in those wilder contexts, as we discuss in paragraphs labeled "Math remark" and in Appendix C.

Actually, finite-dimensional linear spaces over the complex numbers are, in some ways, even better behaved than those over the real numbers, particularly when factoring is involved. For example, before investigating whether a form or a site factors over the real numbers, it is often helpful to consider the easier question of whether it factors over the complex numbers.

2.3 Linear-space duality

Recall that the duality of linear algebra, in the finite-dimensional case, is a symmetric relationship between pairs of spaces. Let X and Y be linear spaces (a.k.a. vector spaces). The set of all linear maps $f\colon X \to Y$ is another linear space, written $\mathrm{Lin}(X, Y)$. In the particular case $Y = \mathbf{R}$, linear maps $f\colon X \to \mathbf{R}$ are called *linear functionals* on X (a.k.a. *dual functionals*), and the space $\mathrm{Lin}(X, \mathbf{R})$ of all such linear functionals is called the *dual* of X and written X^*. Repeating that same construction, the linear space of all second-order maps $\sigma\colon X^* \to \mathbf{R}$ is $X^{**} = \mathrm{Lin}(\mathrm{Lin}(X, \mathbf{R}), \mathbf{R})$, the dual of the dual of X. There is a natural map from X to X^{**} that takes an element x of X to the second-order functional ϵ_x defined by $\epsilon_x(f) := f(x)$, that is, to the functional that evaluates its first-order argument f at the datum x. This natural map is always injective. When $\dim(X)$ is finite, the equality $\dim(X) = \dim(X^*) = \dim(X^{**})$ implies that it must be surjective as well; that is, every second-order functional σ is the evaluate-at-x functional $\sigma = \epsilon_x$, for a unique x in X. The spaces X and X^{**} thus being isomorphic in a natural way, it does no harm to identify them. So, in the finite-dimensional case, duality is a symmetric relationship between pairs of spaces. For example, if we represent the elements of X using column vectors, we then represent the elements of X^* using row vectors and the elements of X^{**} using column vectors once again.

That explanation of duality is standard in the textbooks; but it has the drawback that it treats the spaces X and X^* somewhat asymmetrically. We viewed an element f of the dual space X^* as a first-order functional, while we viewed an element $x = \epsilon_x$ of the primal space $X = X^{**}$ sometimes as the datum x and sometimes as the corresponding second-order functional ϵ_x.

The concept of a *pairing map* puts the primal and dual spaces on a more equal footing. Suppose that X and Y are linear spaces of the same finite dimension k; so, choosing bases, we can think of an element x in X as a column vector of length k, and the same for an element y in Y. Any bilinear map $B\colon X \times Y \to \mathbf{R}$ then has an associated k-by-k matrix M, under the convention that the scalar $B(x, y)$ is the matrix product $B(x, y) := x^{\mathbf{t}}My$. The map B is called a *pairing* between X and Y when its associated matrix M is invertible. Note that every linear functional on X has the form $x \mapsto x^{\mathbf{t}}p$, for a unique column vector p of length k. If the matrix M is invertible, then we have $x^{\mathbf{t}}p = x^{\mathbf{t}}M(M^{-1}p)$, so we can describe that functional equally well as $x \mapsto B(x, y)$ where $y := M^{-1}p$. Thus, we can use the space Y to represent the dual space X^*. Symmetrically, each linear functional on Y has the form $y \mapsto q^{\mathbf{t}}y$, for a unique column vector q. If M is invertible, we can describe that functional equally well as $y \mapsto B(x, y)$ where $x := M^{-\mathbf{t}}q$; so we can use the space X to represent Y^*. Thus, once we fix a pairing between two spaces, we can treat each of them as the dual of the other, without committing ourselves as to which of the two spaces is the primal and which is the dual.

On the other hand, making a temporary convention about that can be pedagogically helpful. For example, suppose that we have fixed a pairing B between two linear spaces X and Y. By the way, it is conventional to denote pairings using angle brackets, so let's switch from writing $B(x, y)$ to writing $\langle x, y\rangle$. We might call the elements of the space X *vectors* and think of them as data, while we call the elements of Y *covectors* and think of them as functions. The pairing map then produces the scalar $\langle x, y\rangle$ from the vector x and the covector y by applying the function y to the datum x, so we have $\langle x, y\rangle = y(x)$. Having broken the symmetry in this direction, we would call X the *primal space* and Y the *dual space* of the pair. But keep in mind that we could equally well have broken the symmetry in the opposite direction, treating x as the function and y as the input datum, with $\langle x, y\rangle = x(y)$. The underlying reality is symmetric, with both x and y as input data; we break that symmetry only because an asymmetric situation, with a datum on one side and a function on the other, is often easier to talk about.

Warning: Different frameworks for CAGD make it pedagogically natural to break the symmetry between primal and dual in different directions. A linear space of n-forms, for example, is typically thought of as a primal space in the homogenized framework. In the paired-algebras framework, on the other hand, it is more natural to view the space of n-sites as primal, which forces the space of n-forms to be dual. When the words "primal" and "dual" are used as identifiers in this way, only the context can clarify which is which — that is, can clarify the direction in which the symmetry is being broken.

2.4 Algebras

In this monograph, we are going to be extending affine spaces into linear spaces and linear spaces into commutative algebras. We here review the mathematical concept of an algebra. Feel free to skip this section on first reading, referring back to it only as needed.

Fix some field of scalars. In this monograph, that field will always be the real numbers $\mathbf{R}$, but any field would do. An *algebra* over that field is a set G with three operations defined on it: addition, multiplication, and scalar multiplication. The addition and the multiplication must make G into a ring. (In keeping with modern practice, we require that any ring — and hence any algebra — have a multiplicative identity.) The addition and the scalar multiplication must make G into a linear space. And the two multiplications must satisfy $t(xy) = (tx)y = x(ty)$, for all scalars t and all elements x and y of the algebra G. An algebra is *commutative* when its multiplication is commutative, that is, when $xy = yx$.

For example, for any fixed n, the set of all n-by-n real matrices forms an algebra. The dimension of this algebra, viewed as a linear space, is n^2. Once n exceeds 1, this algebra is noncommutative.

Exercise 2.4-1 Another way to think of an algebra is as a ring that includes the field of scalars as a central subring. More precisely, show that the definition of an algebra given above is equivalent to the following: An *algebra* is a ring G together with a ring homomorphism $g \colon \mathbf{R} \to G$ with the property that $g(t)x = xg(t)$, for all x in G and t in $\mathbf{R}$.

Hint: Given a ring G and a homomorphism $g \colon \mathbf{R} \to G$, we can define a scalar multiplication in G by the rule $tx := g(t)x$. Conversely, given an algebra as described above, we can define a ring homomorphism $g \colon \mathbf{R} \to G$ by setting $g(t) := t\mathbf{1}$, where $\mathbf{1}$ denotes the multiplicative identity in G.

For our purposes, polynomial algebras are the most important examples. If V is some set of symbols, then all polynomials with real coefficients and with variables drawn from V form a commutative algebra, written $\mathbf{R}[V]$. The dimension of this algebra, as a linear space, is infinite whenever V is nonempty, since we can form polynomials of arbitrarily high degree. In this monograph, the set V will usually be finite; but the polynomial algebra $\mathbf{R}[V]$ makes sense even when V is infinite. Keep in mind, though, that any single polynomial is a sum of finitely many terms, each of finite total degree, and hence any single polynomial involves only finitely many variables.

An algebra is *graded* when it is expressed as a linear-space direct sum $G = \bigoplus_{n \geq 0} G_n$ in such a way that the ring multiplication takes $G_i \times G_j$ into G_{i+j}, for all nonnegative i and j. The linear subspace G_n is called the n^{th} *graded slice* of the algebra G, and the elements of G_n are called *homogeneous of grade* n or *of degree* n. Every element x of a graded algebra can be written uniquely as a sum $x = \sum_{n \geq 0} (x)_n$ of its *graded components*, where the n^{th} graded component $(x)_n$ is homogeneous of grade n and where only finitely many of the components are nonzero.

The key example of a graded algebra, for our purposes, is the polynomial algebra $\mathbf{R}[V]$, graded by total degree. Let's denote by $\mathbf{R}_n[V]$ the linear space of all polynomials that are homogeneous of total degree n in the variables in V. That space is the n^{th} graded slice of the polynomial algebra $\mathbf{R}[V]$ (and hence might also be denoted $\mathbf{R}[V]_n$). Note that multiplication maps $\mathbf{R}_i[V] \times \mathbf{R}_j[V]$ into $\mathbf{R}_{i+j}[V]$. We can group the terms of any polynomial f by their total degree and hence decompose f uniquely as the sum $f = \sum_{n \geq 0} (f)_n$ of its graded components. If the number of variables $v := |V|$ is finite, then each graded slice of the polynomial algebra $\mathbf{R}[V]$ is finite-dimensional; in fact, by the formula for choosing with repetition, we have

$$\dim(\mathbf{R}_n[V]) = \binom{n+v-1}{n}.$$

An *algebra homomorphism* is a linear map that is also a ring homomorphism. Thus, an algebra homomorphism $f\colon G \to H$ must satisfy $f(x+y) = f(x)+f(y)$, $f(tx) = tf(x)$, $f(xy) = f(x)f(y)$, and $f(\mathbf{1}) = \mathbf{1}$, for all elements x and y of G and all scalars t.

Exercise 2.4-2 If $g\colon \mathbf{R} \to G$ and $h\colon \mathbf{R} \to H$ are the ring homomorphisms that describe two algebras G and H as in Exercise 2.4-1, show that an algebra homomorphism $f\colon G \to H$ is the same thing as a ring homomorphism that preserves scalars, in the sense that $f \circ g = h$.

Chapter 3

The Nested-Spaces Framework

The paired algebras are the cornerstones of a new framework for studying polynomial functions in CAGD. Before we construct that new framework, however, we should discuss the frameworks that are currently used for this purpose — of which there are two. In the first of those existing frameworks, the one that underlies the thinking of a high-school student, the real-valued, polynomial functions of various degrees defined on a common affine domain space form a nested family of linear spaces. In this chapter, we review that framework, which we'll call the *nested-spaces framework*.

As we consider various frameworks for CAGD, we shall use cubic Bézier triangles as our motivating problem. So suppose that we want to design a surface in a 3-dimensional object space O, with (x, y, z) coordinates. We divide that surface into triangular patches, and we specify each such patch parametrically as follows: We invent an affine parameter plane A, with (u, v) coordinates, and we define the x, y, and z coordinates of the patch to be real-valued, polynomial functions on A of total degree at most 3 in u and v. Any framework for CAGD that we consider must provide the linear spaces that are appropriate for studying a problem of that sort.

3.1 Choosing a Cartesian coordinate system

In the nested-spaces framework, we begin by setting up a Cartesian coordinate system in the affine parameter plane A. That is, we choose some point C in A to act as the center of our Cartesian coordinate grid, and we choose two vectors φ and ψ over A to be the unit vectors in the u and v directions. Each point P in A can then be uniquely expressed in the form

$$P = C + u(P)\varphi + v(P)\psi, \tag{3.1-1}$$

for certain real numbers $u(P)$ and $v(P)$, called the *Cartesian coordinates* of P. Note that u itself is an affine function $u\colon A \to \mathbf{R}$, and the same for v.

We intend to use, as the x, y, and z coordinates of our surface patch, polynomial functions $f\colon A \to \mathbf{R}$ of total degree at most 3. The set of all such cubic functions forms a linear space of dimension 10, and we often adopt, as our basis for that linear space, the ten functions defined by the monomials $u^i v^j$, for $i + j \le 3$. Any cubic function $f\colon A \to \mathbf{R}$ can be uniquely expressed as a linear combination of the ten functions in that *power basis*:

$$\begin{aligned} f = {} & f_{30}u^3 + f_{21}u^2v + f_{12}uv^2 + f_{03}v^3 \\ & + f_{20}u^2 + f_{11}uv + f_{02}v^2 \\ & + f_{10}u + f_{01}v \\ & + f_{00}. \end{aligned} \tag{3.1-2}$$

In general, let's denote by $\mathrm{Poly}_{\le n}(A, \mathbf{R})$ the linear space of all functions $f\colon A \to \mathbf{R}$ that can be defined by polynomials of degree at most n in the variables u and v. Note that, if we adopted some other Cartesian coordinate system (u', v') for the plane A, based on an origin point C' and unit vectors φ' and ψ', the two systems (u, v) and (u', v') would be affinely related, so we would end up with the same space of functions $\mathrm{Poly}_{\le n}(A, \mathbf{R})$.

3.2 Picturing the nested-spaces framework

Figure 3.1 depicts the nested-spaces framework graphically. On the left, we have the affine parameter plane A, all by itself. Each of the other shapes represents a linear space of interest in CAGD. In the infinite nest of triangles, the n^{th} triangle represents the linear space $\mathrm{Poly}_{\le n}(A, \mathbf{R})$. The space $\mathrm{Poly}_{\le 0}(A, \mathbf{R})$ of constant, real-valued functions on A has the constant function 1 as a basis. The space $\mathrm{Poly}_{\le 1}(A, \mathbf{R})$ of affine, real-valued functions on A has the three functions u, v, and 1 as a basis. The space $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$ has the six functions u^2, uv, v^2, u, v, and 1 as a basis. And so forth; in general, we have $\dim(\mathrm{Poly}_{\le n}(A, \mathbf{R})) = \binom{n+2}{n} = \binom{n+2}{2}$.

The union $\bigcup_{n \ge 0} \mathrm{Poly}_{\le n}(A, \mathbf{R})$ of the nested triangles is an algebra, which we shall denote $\mathrm{Poly}(A, \mathbf{R})$: the algebra of all real-valued functions on A that can be defined by polynomials in u and v of any degree. By mapping each such function to its defining polynomial, we see that the algebra of functions $\mathrm{Poly}(A, \mathbf{R})$ is isomorphic to the polynomial algebra $\mathbf{R}[u, v]$.

Warning: The polynomial algebra $\mathbf{R}[u, v]$ is graded by total degree; but the algebra $\mathrm{Poly}(A, \mathbf{R})$ of polynomial functions has no natural grading. For example, it makes sense to distinguish those polynomials in $\mathbf{R}[u, v]$ that are homogeneous of total degree 3, that is, the linear combinations of u^3, u^2v, uv^2, and v^3. Indeed, that 4-dimensional linear space of homogeneous cubics is precisely $\mathbf{R}_3[u, v]$, the third graded slice of the algebra $\mathbf{R}[u, v]$. But it wouldn't make sense to distinguish the corresponding functions in the algebra

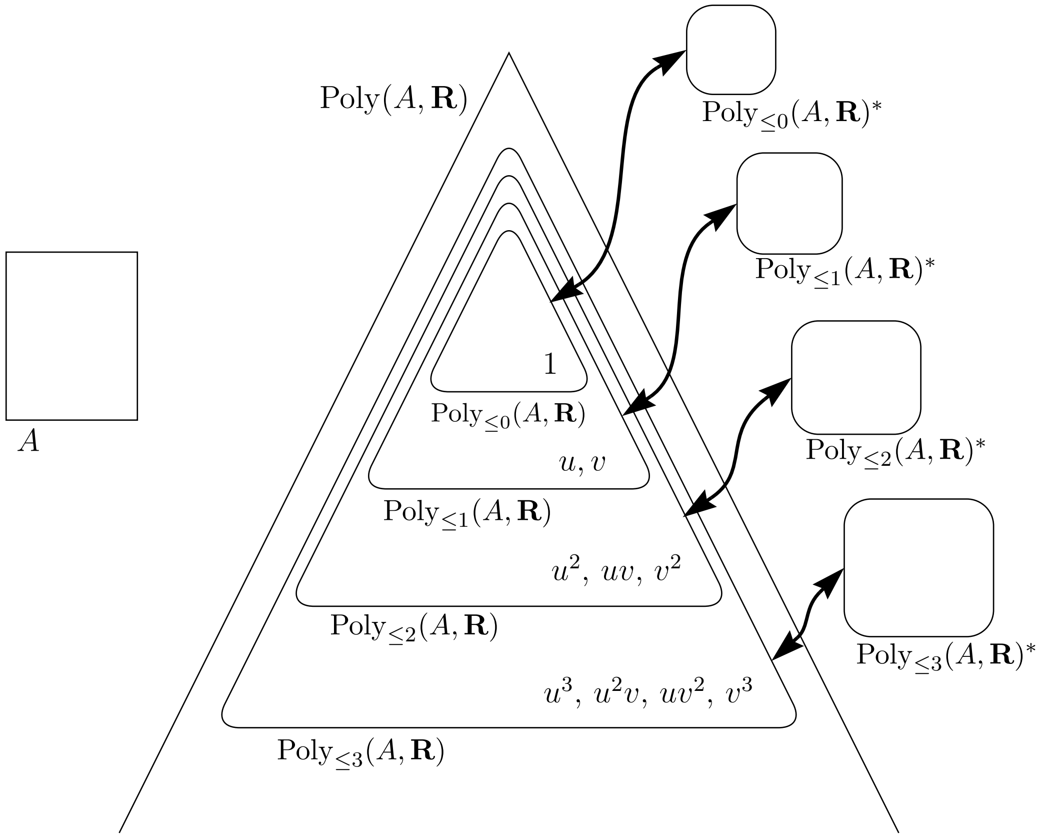


Figure 3.1: The nested-spaces framework

Poly$(A, \mathbf{R})$. The functions in Poly$(A, \mathbf{R})$ whose defining polynomials lie in $\mathbf{R}_3[u, v]$ are the ones that vary as a cubic function of the distance from the point C, that is, the functions f that satisfy $f(C + t(P - C)) = t^3 f(P)$, for all points P and real numbers t. But the center point C of our coordinate system for the affine plane A was an arbitrary choice — that's part of what it means for the plane A to be affine. Thus, requiring a function $f\colon A \to \mathbf{R}$ to be a homogeneous cubic doesn't make sense, since the affine plane A has no preferred origin to be homogeneous around.

3.3 The dual spaces

In the nested-spaces framework, the linear spaces Poly$_{\leq n}(A, \mathbf{R})$ are thought of as primal; that is, the symmetry discussed in Section 2.3 is broken in the direction that views a polynomial function on A to be a primal object. The duals of those primal spaces are shown in Figure 3.1 as rounded rectangles, each linked by a double-headed arrow to its primal partner.

Consider the space Poly$_{\leq 3}(A, \mathbf{R})^*$, for example. An element σ of this space is a dual functional defined on cubic functions, that is, it is a linear map $\sigma\colon \mathrm{Poly}_{\le 3}(A, \mathbf{R}) \to \mathbf{R}$. If we adopt the power basis $(u^i v^j)_{i+j\le 3}$ for the space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})$ of cubic functions, it is natural to adopt the dual basis for the space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})^*$. That dual basis consists of the ten functionals $(\tau_{ij})_{i+j\le 3}$ determined by the duality constraints

$$\tau_{ij}(u^k v^l) = \begin{cases} 1 & \text{if } i = k \text{ and } j = l \\ 0 & \text{otherwise.} \end{cases}$$

Given any cubic function f in $\mathrm{Poly}_{\le 3}(A, \mathbf{R})$, we can use the functionals in this dual basis to compute the coefficients that are needed to expand f in the power basis; that is, Equation 3.1-2 holds just when the ten coefficients $(f_{ij})_{i+j\le 3}$ are determined by the equations $f_{ij} = \tau_{ij}(f)$.

Of course, since duality is symmetric, the same holds the other way around. An arbitrary dual functional σ in the space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})^*$ can be uniquely expressed as a linear combination of the elements of the dual basis,

$$\begin{aligned} \sigma = \; & \sigma_{30}\tau_{30} + \sigma_{21}\tau_{21} + \sigma_{12}\tau_{12} + \sigma_{03}\tau_{03} \\ & + \sigma_{20}\tau_{20} + \sigma_{11}\tau_{11} + \sigma_{02}\tau_{02} \\ & + \sigma_{10}\tau_{10} + \sigma_{01}\tau_{01} \\ & + \sigma_{00}\tau_{00}, \end{aligned} \tag{3.3-1}$$

and the ten coefficients $(\sigma_{ij})_{i+j\le 3}$ in this expansion are given by $\sigma_{ij} = \sigma(u^i v^j)$.

3.4 Interpreting elements of the dual spaces

Of the ten dual functionals in our basis $(\tau_{ij})_{i+j\le 3}$ for the space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})^*$, one has a particularly simple interpretation. Evaluating Equation 3.1-2 at the center point C, the point in the domain plane A with coordinates $u(C) = v(C) = 0$, we see that $f(C) = f_{00} = \tau_{00}(f)$. So the functional τ_{00} evaluates its argument at the center point C.

Evaluation at any fixed point constitutes a dual functional. If we evaluate Equation 3.1-2 at $P = C + u(P)\varphi + v(P)\psi$, we find that

$$\begin{aligned} f(P) = \; & f_{30}\, u(P)^3 + f_{21}\, u(P)^2 v(P) + f_{12}\, u(P) v(P)^2 + f_{03}\, v(P)^3 \\ & + f_{20}\, u(P)^2 + f_{11}\, u(P) v(P) + f_{02}\, v(P)^2 \\ & + f_{10}\, u(P) + f_{01}\, v(P) \\ & + f_{00}. \end{aligned}$$

It follows that evaluation at the point P is the dual functional ϵ_P given by

$$\begin{aligned}\epsilon_P = \; & u(P)^3\,\tau_{30} + u(P)^2 v(P)\,\tau_{21} + u(P)v(P)^2\,\tau_{12} + v(P)^3\,\tau_{03}\\ & + u(P)^2\,\tau_{20} + u(P)v(P)\,\tau_{11} + v(P)^2\,\tau_{02}\\ & + u(P)\,\tau_{10} + v(P)\,\tau_{01}\\ & + \tau_{00}.\end{aligned}$$

Typically, though, when we expand a dual functional in terms of our chosen basis $(\tau_{ij})_{i+j\le 3}$, the expansion won't have that special form for any two scalars $u(P)$ and $v(P)$. Thus, a typical dual functional does not correspond to evaluation at any point. We can express any dual functional as a linear combination of point evaluations (as follows from Lemma 7.2-2). Alternatively, we can express any dual functional as a certain differential operator (as discussed in Section 7.10). But let's put those topics aside for now.

3.5 Are the dual spaces nested?

The primal spaces $\mathrm{Poly}_{\le n}(A, \mathbf{R})$, the triangles in Figure 3.1, are nested. It would be nice if the dual spaces $\mathrm{Poly}_{\le n}(A, \mathbf{R})^*$ were also nested; so it is important to thoroughly understand that they are not. They would be if the concept "subspace" in linear algebra were self-dual. But the concept that is dual to "subspace" is "quotient space": When X and Y are linear spaces, X is a subspace of Y if and only if X^* is a quotient space of Y^*. This is standard linear algebra, but we review it here for completeness.

Consider the space $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$ of quadratic functions on A, sitting as a subspace inside the space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})$ of cubic functions on A. We have $\dim(\mathrm{Poly}_{\le 2}(A, \mathbf{R})) = 6$, while $\dim(\mathrm{Poly}_{\le 3}(A, \mathbf{R})) = 10$. Suppose that we expand a cubic function f in terms of the power basis $(u^i v^j)_{i+j\le 3}$, as shown in Equation 3.1-2. The function f will also be quadratic — that is, will lie in the subspace $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$ — just when the four cubic coefficients f_{30}, f_{21}, f_{12}, and f_{03} are all zero.

What happens in the dual spaces? Because we have singled out the subspace $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$ of the primal space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})$, there is a certain subset of the dual space $\mathrm{Poly}_{\le 3}(A, \mathbf{R})^*$ that we can single out in a natural way: the *annihilator* of $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$, written $\mathrm{Ann}(\mathrm{Poly}_{\le 2}(A, \mathbf{R}))$. A dual functional σ in $\mathrm{Poly}_{\le 3}(A, \mathbf{R})^*$ belongs to $\mathrm{Ann}(\mathrm{Poly}_{\le 2}(A, \mathbf{R}))$ just when $\sigma(f) = 0$, for all f in $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$. Unfortunately, there is no hope that this annihilator subspace will coincide with or can even somehow represent the smaller dual space $\mathrm{Poly}_{\le 2}(A, \mathbf{R})^*$, since the dimensions are wrong. We have $\dim(\mathrm{Poly}_{\le 2}(A, \mathbf{R})^*) = 6$; but a dual functional annihilates $\mathrm{Poly}_{\le 2}(A, \mathbf{R})$ just when it can be written as a linear combination of the four functionals τ_{30}, τ_{21}, τ_{12}, and τ_{03}, so $\dim(\mathrm{Ann}(\mathrm{Poly}_{\le 2}(A))) = 4$. Here is what is true instead:

The smaller dual space $\mathrm{Poly}_{\leq 2}(A, \mathbf{R})^*$ is isomorphic in a natural way to the quotient space $\mathrm{Poly}_{\leq 3}(A, \mathbf{R})^* / \mathrm{Ann}(\mathrm{Poly}_{\leq 2}(A, \mathbf{R}))$. But there is no natural way to single out one 6-dimensional subspace of $\mathrm{Poly}_{\leq 3}(A, \mathbf{R})^*$ to represent $\mathrm{Poly}_{\leq 2}(A, \mathbf{R})^*$; that is, singling out one such subspace would involve making an arbitrary choice.

As it happens, we have already made an adequate arbitrary choice: our choice of the point C as the center of our Cartesian coordinate system in the plane A. As we discussed at the end of Section 3.2, the choice of C determines a 4-dimensional subspace of $\mathrm{Poly}_{\leq 3}(A, \mathbf{R})$: the functions given by polynomials that are homogeneous of degree 3 in u and v, that is, the functions that are homogeneous cubics around C. Call that space H_C. The annihilator $\mathrm{Ann}(H_C)$ is a 6-dimensional subspace of $\mathrm{Poly}_{\leq 3}(A, \mathbf{R})^*$ that we could use to represent $\mathrm{Poly}_{\leq 2}(A, \mathbf{R})^*$. But we want the structures in our frameworks to be independent of the coordinate system that we choose for the affine space A, so this path to nested dual spaces is closed to us.

Chapter 4

The Homogenized Framework

In the nested-spaces framework, the primal spaces $\mathrm{Poly}_{\leq n}(A, \mathbf{R})$ are nested, but the dual spaces $\mathrm{Poly}_{\leq n}(A, \mathbf{R})^*$ are not. Since duality is a symmetric relationship, that lack of symmetry constitutes a flaw. Furthermore, forcing the dual spaces to be nested as well would involve making arbitrary choices, as we discussed in Section 3.5; so we aren't willing to repair the flaw that way. The only other way to repair the flaw is to eliminate the nesting of the primal spaces. Fortunately, we can eliminate the primal nesting by making a simple change in our framework, to wit, by *homogenizing*. Indeed, if we adopt Bernstein bases for our primal spaces, rather than power bases, this homogenization happens automatically. The resulting *homogenized framework* is the framework for studying polynomial functions that underlies most current research in CAGD.

4.1 To n-forms via barycentric coordinates

As an easy introduction to the homogenized framework, let's study cubic Bézier triangles once again, but using a Bernstein basis, rather than a power basis. Let $\triangle QRS$ be a reference triangle in the parameter plane A. Any point P in A can be uniquely represented as an affine combination of the three vertices Q, R, and S:

$$P = q(P)Q + r(P)R + s(P)S \qquad \text{where } q(P) + r(P) + s(P) = 1. \tag{4.1-1}$$

The numbers $(q(P), r(P), s(P))$ are called the *barycentric coordinates* of P, while the triple of affine functions (q, r, s) is called a *barycentric coordinate system* for the plane A. Every cubic function $f \colon A \to \mathbf{R}$ can be uniquely expressed as a homogeneous cubic polynomial in the variables q, r, and s. The *Bernstein basis* for the space of such functions consists of the ten cubic monomials in q, r and s, scaled by trinomial coefficients:

$$\left(\binom{3}{i\ j\ k} q^i r^j s^k\right)_{i+j+k=3}.$$

Every cubic function $f\colon A \to \mathbf{R}$ can be expanded uniquely as a linear combination of those basis functions,

$$\begin{aligned}
f = {} & f_{300}\, q^3 \\
& + f_{210}\, 3q^2 r + f_{201}\, 3q^2 s \\
& + f_{120}\, 3qr^2 + f_{111}\, 6qrs + f_{102}\, 3qs^2 \\
& + f_{030}\, r^3 + f_{021}\, 3r^2 s + f_{012}\, 3rs^2 + f_{003}\, s^3,
\end{aligned} \tag{4.1-2}$$

and the coefficients $(f_{ijk})_{i+j+k=3}$ in this expansion are known as the *Bézier ordinates* of the function f.

In general, let's denote by $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$ the linear space of all functions that can be defined by polynomials that are homogeneous of degree n in the variables q, r, and s. Such a function is called an *n-form* on the plane A. Note that an n-form has a well-defined value for any triple of scalars (q, r, s), even when the sum $q+r+s$ differs from 1. As we discuss shortly, it is for that reason that we write $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$, with a hat accent over the A. (Indeed, it wouldn't make sense to write "$\mathrm{Poly}_n(A, \mathbf{R})$", without the hat accent. As we discussed in Section 3.2, the affine space A has no preferred center point around which to require a polynomial function to be homogeneous.)

Have we pulled apart the nested spaces? The nesting arose because we considered the constant function 1, for example, to be a function of degree at most n, for every nonnegative n. But the single function 1 has now given rise to an infinite sequence of distinct forms: the constant form 1, the linear form $q+r+s$, the quadratic form $(q+r+s)^2$, and so forth. In this way, we have converted the nested spaces $\mathrm{Poly}_{\le 0}(A, \mathbf{R}) \subset \mathrm{Poly}_{\le 1}(A, \mathbf{R}) \subset \ldots$ into disjoint spaces $\mathrm{Poly}_0(\hat{A}, \mathbf{R})$, $\mathrm{Poly}_1(\hat{A}, \mathbf{R})$, and so forth. (To be picky, those latter spaces are only almost disjoint: They share a common origin, since the zero function on A is an n-form for every $n \ge 0$.)

4.2 To n-forms via a weight coordinate

While Bernstein bases lead naturally to the homogenized framework, we can start with power bases and still end up homogenized as follows. Given the center point C and the unit vectors φ and ψ in the plane A, we write each point P in A in the form

$$P = w(P)C + u(P)\varphi + v(P)\psi \qquad \text{where } w(P) = 1. \tag{4.2-1}$$

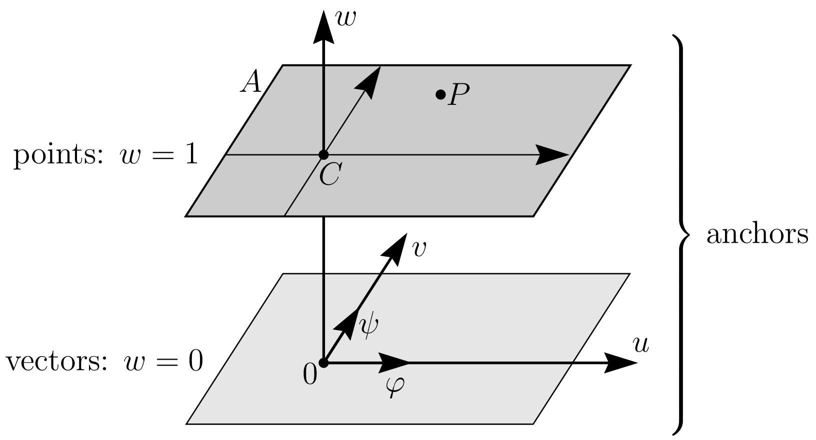


Figure 4.1: The linearization $\hat{A}$ of the affine plane A

Like Equation 4.1-1, but unlike Equation 3.1-1, this equation gives each point P three coordinates, subject to one affine constraint: the coordinates $(w(P), u(P), v(P))$, subject to the constraint $w(P) = 1$. Any polynomial of total degree at most n in the variables u and v can be converted into an equivalent polynomial that is homogeneous of degree n in the variables w, u, and v simply by adding factors of w to each term as appropriate, the term $tu^i v^j$ becoming $tw^{n-i-j}u^i v^j$. This process is called *homogenization.* Homogenizing each function in the space $\mathrm{Poly}_{\leq n}(A, \mathbf{R})$ leads to the same linear space $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$ of n-forms that we arrived at via the Bernstein basis, since the coordinate systems (q, r, s) and (w, u, v) are linearly related.

4.3 The linearization of an affine space

But wait a minute: What space is it that has (q, r, s) and (w, u, v) as two possible coordinate systems? Well, it is surely a 3-dimensional space, and we want it be linear, rather than merely affine. It includes the plane A as an affine hyperplane — to wit, the hyperplane $q + r + s = 1$ or, equivalently, $w = 1$. Those properties are enough to determine it uniquely, up to a unique isomorphism, as shown in Figure 4.1. It has various names and is written in various ways; let's call it the *linearization* of A and write it $\hat{A}$.

Here is another way to describe how homogenization works. We start with polynomial functions of degree at most n, defined on an affine space A. We could choose some point in A, such as C, to act something like an origin. But different n-ic functions are homogeneous around different points in A, and we don't want to play favorites; worse yet, many n-ic functions aren't homogeneous around any point in A. Instead, we adjoin to A an origin that lies outside of A, a common origin that all n-ics can be homogeneous around.

The linear span of A with respect to this new, exterior origin is a linear space $\hat{A}$, with $\dim(\hat{A}) = \dim(A)+1$. Any polynomial function $f\colon A \to \mathbf{R}$ of degree at most n extends uniquely to a function $\hat{f}\colon \hat{A} \to \mathbf{R}$ that is homogeneous of degree n — that is, extends to an n-form. To effect that extension, let $w\colon \hat{A} \to \mathbf{R}$ be the unique linear functional on $\hat{A}$ that takes the constant value 1 on the hyperplane A, so $A = w^{-1}(1)$; a value of this functional w is often called a *weight*. To compute $\hat{f}$, we take the polynomial that defines f and we add factors of the weight functional w to each term, as needed, to bring that term up to the proper total degree. Going back from $\hat{f}$ to f is even easier: We simply substitute $w := 1$. As a result, we can treat the function f and the n-form $\hat{f}$ as two aspects of the same underlying reality.

In the particular case $n = 1$, the affine function q on A extends uniquely to a 1-form on A, that is, to a linear functional on $\hat{A}$. For simplicity, we shall use the same symbol q to denote that linear functional, rather than writing $\hat{q}$. The same goes for the functionals r, s, w, u, and v; indeed, we wrote w already in the last paragraph, rather than $\hat{w}$. In this way, each of the triples (q, r, s) and (w, u, v) now constitutes a linear coordinate system on the linear space $\hat{A}$, and those two systems are related by some invertible 3-by-3 matrix.

Marcel Berger [3] gives a thorough explanation of linearization. He (or perhaps his translator, Silvio Levy) refers to the space $\hat{A}$ as the *universal space* of A, since $\hat{A}$ satisfies a certain universal mapping condition, as we discuss in Section 9.1. But lots of things satisfy universal mapping conditions; it seems more specific to refer to $\hat{A}$ as the *linearization* of A.

4.4 A new term: "anchor"

Linearization is a central technique in CAGD, but it is not understood as clearly as it should be. One reason is people's reluctance to add one more dimension — especially to move from 3 dimensions, which they can visualize, to 4 dimensions, which they cannot. But a simpler stumbling block is the lack of good terminology. Given an affine space A sitting inside its linearization $\hat{A}$, we want to reserve the term "point" for the elements of A, that is, the elements P of $\hat{A}$ that have weight 1, that satisfy $w(P) = 1$. In the same spirit, we want to reserve the term "vector" for the elements π of $\hat{A}$ that have weight 0. We then have the familiar equations "point − point = vector" and "point ± vector = point". But what name should we use for an arbitrary element p of $\hat{A}$? No good term has yet taken hold. Most authors use a phrase like "weighted point", "mass point", or "punctual mass". The justification for such names is that any element p of $\hat{A}$ whose weight is nonzero can be written as a scalar multiple of a point: We have $p = w(p)(p/w(p))$, where $p/w(p)$ is a point because $w(p/w(p)) = w(p)/w(p) = 1$. But note that nonzero vectors over A are elements of the linearization $\hat{A}$ also, and they can't be written as scalar multiples of points. On that basis, Fiorot and Jeannin [21] proposed the term "massic vector".

But none of those phrases is adequate. We should dignify the elements of the linearization by giving them a single-word name. Here is my proposal: Given any affine space A, let's refer to an element of its linearization $\hat{A}$ as an *anchor* over A. So a point in A is an anchor over A of weight 1, while a *vector* over A is an anchor over A of weight 0. Every anchor is either a vector or a scalar multiple of a point.

(In defense of the word "anchor", it connotes a fixed point and something weighty, both of which are appropriate. Indeed, control points in computer drawing systems are sometimes called "anchors". Also, there is no established mathematical meaning of "anchor" with which this new sense might be confused. Finally, it is quite convenient that the noun "anchor" has two syllables and ends in "-or", like "vector" and "tensor".)

Consider the domain plane A of a cubic Bézier triangle $F\colon A \to O$, as in our recurring example. Every anchor p over the plane A can be written uniquely as a linear combination

$$p = w(p)C + u(p)\varphi + v(p)\psi, \tag{4.4-1}$$

where we no longer place any constraint on the weight $w(p)$. Equivalently, every anchor p can be written uniquely as a linear combination

$$p = q(p)Q + r(p)R + s(p)S, \tag{4.4-2}$$

with no constraint on the sum $q(p) + r(p) + s(p)$.

4.5 Coanchors

Now that an element of the linearized space $\hat{A}$ is an anchor over A, an element of the dual space $\hat{A}^*$ — that is, a linear functional on anchors — is a *coanchor* on A (nothing to do with a co-anchor of a television newscast). In particular, the linear functionals q, r, s, w, u, and v are coanchors on A. The *weight coanchor* is the coanchor $w = q+r+s$ that satisfies $A = w^{-1}(1)$. A Cartesian coordinate system for A, such as (w, u, v), is a basis of $\hat{A}^*$ that contains the weight coanchor as one basis element. A barycentric coordinate system, such as (q, r, s), is a basis of $\hat{A}^*$ whose coanchors sum to the weight.

Using Cartesian coordinates, every coanchor h on the plane A can be written uniquely as a linear combination of the coanchors w, u, and v:

$$\begin{aligned} h &= C(h)w + \varphi(h)u + \psi(h)v \\ &= h(C)w + h(\varphi)u + h(\psi)v \\ &= \langle h, C\rangle w + \langle h, \varphi\rangle u + \langle h, \psi\rangle v. \end{aligned}$$

We wrote the right-hand side of that equation three times because there is an issue about how to write it. On the first line, we wrote the exact dual of Equation 4.4-1. The coefficients on that first line look strange because we aren't used to treating an anchor as a function that gets applied to a coanchor as its input datum. We typically prefer to break the symmetry in the opposite direction, treating the coanchor as the function and the anchor as the datum, as on the second line. Of course, the underlying reality is symmetric, as we discussed in Section 2.3: We are really pairing the coanchor with the anchor, however we choose to write it.

In barycentric coordinates, the story is much the same. We can write any coanchor h on the plane A uniquely as a linear combination of the coanchors q, r, and s:

$$h = \langle h, Q\rangle q + \langle h, R\rangle r + \langle h, S\rangle s.$$

The dual of a coordinate system is a *reference frame*. For example, the reference frame for the plane A that is dual to the Cartesian coordinate system (w, u, v) consists of the center point C and the unit vectors φ and ψ. The three anchors (C, φ, ψ) form a basis for the linear space $\hat{A}$ of anchors over A, and we have the duality constraints

$$\begin{pmatrix} w \\ u \\ v \end{pmatrix} \begin{pmatrix} C & \varphi & \psi \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In general, a *Cartesian reference frame* for an affine space is a basis for its linearization that is all vectors, except for a single point.

The reference frame that is dual to the barycentric coordinate system (q, r, s) consists of the three points Q, R, and S. Those three points also form a basis for the linearization $\hat{A}$, and they satisfy the duality constraints

$$\begin{pmatrix} q \\ r \\ s \end{pmatrix} \begin{pmatrix} Q & R & S \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In general, a *barycentric reference frame* for an affine space is a basis for its linearization that consists entirely of points.

A comment about notation: We are denoting the fundamental pairing between the linear space $\hat{A}$ of anchors and the linear space $\hat{A}^*$ of coanchors as a function $\langle\ ,\ \rangle\colon \hat{A}^* \times \hat{A} \to \mathbf{R}$. In particular, given an anchor p and a coanchor h, we shall pair them by writing $\langle h, p\rangle$, with h on the left and p on the right. We adopt that convention for two related reasons. First, when breaking the symmetry, people more often think of the coanchor h as the function and the anchor p as its input datum, and it is convenient to end up with the function on the left. Second, people typically represent an anchor (or vector) in coordinates as a column of numbers, while they represent a coanchor (or covector) as a row of numbers; it is convenient to end up with the row to the left of the column, so that the dot product that effects the pairing follows the standard rules for matrix multiplication. Hence, we prefer to write our pairings with their arguments in the order $\langle \text{dual}, \text{primal} \rangle$.

4.6 The benefits of linearization

There are many reasons why linearization and homogenization are beneficial in CAGD. While this monograph is not about linearization, let's pause to recall some of those benefits.

- Linearization simplifies the algebra that underlies geometric operations. If P and Q are points in an affine space A, the points on the line joining them have the form $(1-t)P+tQ$. Before we linearize, we must treat that entire affine combination as a single algebraic operation, since the individual summands $(1-t)P$ and tQ are not points. After linearizing, however, we recognize the summands as anchors. The linear space $\hat{A}$ of anchors is closed under addition and scalar multiplication as independent operations. The overall affine combination $(1-t)P+tQ$ denotes a point because its weight is 1; we calculate

$$w\big((1-t)P+tQ\big) = (1-t)w(P) + tw(Q) = (1-t) + t = 1.$$

- Linearization converts the geometric notions of collinearity, coplanarity, and the like into rank tests. Suppose that the affine space A is of dimension d. Given points P_0 through P_k in A, the coordinates of the corresponding anchors form a $(k+1)$-by-$(d+1)$ matrix. The points $(P_0, \ldots, P_k)$ are affinely independent, spanning a flat of the maximum possible dimension k, just when that matrix has rank $k+1$.

- Linearization lets us encode an affine transformation as a single matrix. Before we linearize, we implement an affine transformation of an affine d-space A as a linear transformation of A followed by a translation, that is, as a d-by-d matrix together with a vector of length d. After linearizing, we instead use a single matrix of size $(d+1)$-by-$(d+1)$. Assuming Cartesian coordinate systems, that larger matrix is produced by pasting the vector onto the smaller matrix and then adding a new first (or last) column (or row) that is all zeros, except for a single one. Combining all of the data that describes an affine transformation into a single matrix in this way is particularly helpful when we want to compose affine transformations; after linearizing, we can compose affine transformations simply by multiplying their matrices.

- Linearization provides an elegant way to generalize from polynomial curves and surfaces to rational ones. To specify a cubic polynomial Bézier triangle in the affine object space O, we have been choosing three cubic forms on the affine parameter plane A to determine the x, y, and z coordinates of the surface. To allow that surface to be rational, it suffices to choose one additional cubic form, playing the role of a common denominator. An elegant way to achieve that effect is to draw a polynomial surface in the linearized object space $\hat{O}$, which is a linear 4-space, say with coordinate system (w_O, x, y, z). (We write the weight coanchor on O as w_O only to distinguish it from the weight coanchor $w = w_A$ on A.) Projecting the resulting polynomial surface down into O from the origin of $\hat{O}$ gives us a rational surface in O. Thus, while a polynomial curve or surface has Bézier points, a rational Bézier curve or surface has *Bézier anchors*. Typically, those anchors are positive scalar multiples of points; but vectors and negative scalar multiples of points also make sense as Bézier anchors.

- Linearization is the first step on the road to projective geometry. In projective geometry, we identify any two anchors that differ by a scalar multiple and we treat the resulting equivalence class, a line through the origin of the linearized space $\hat{A}$, as a "point" in a new space: the projective closure of A. The coordinates of any nonzero anchor on such a line are *homogeneous coordinates* of the corresponding "point" in the projective closure. Each point in A, together with all of its scalar multiples, becomes a "finite point" in this projective closure. But the projective closure also contains "points at infinity", which are lines through the origin of $\hat{A}$ that consist entirely of vectors over A. We can then represent a projective transformation of a d-dimensional space A using a matrix of size $(d+1)$-by-$(d+1)$: the same matrix that we used above to encode an affine transformation, except with the constraint on the first (or last) column (or row) removed and with the understanding that matrices that differ by a scalar multiple are identified.

While linearization is quite valuable, our goal is to take the next step, which is algebrization. Linearization embeds an affine space of points in a linear space of anchors, thereby defining addition and scalar multiplication as separate operations. Algebrization embeds that linear space, in turn, in an algebra of sites, thereby defining a new operation of multiplication. Each step along this road, from point to anchor to site, from affine space to linear space to algebra, brings us new tools to exploit in CAGD.

It is not clear, by the way, that sites are the end of this road. Section 8.5 speculates about taking one more step, from sites to *locations*, so that division by points will be legal, as well as multiplication. Thus, the road may go on:

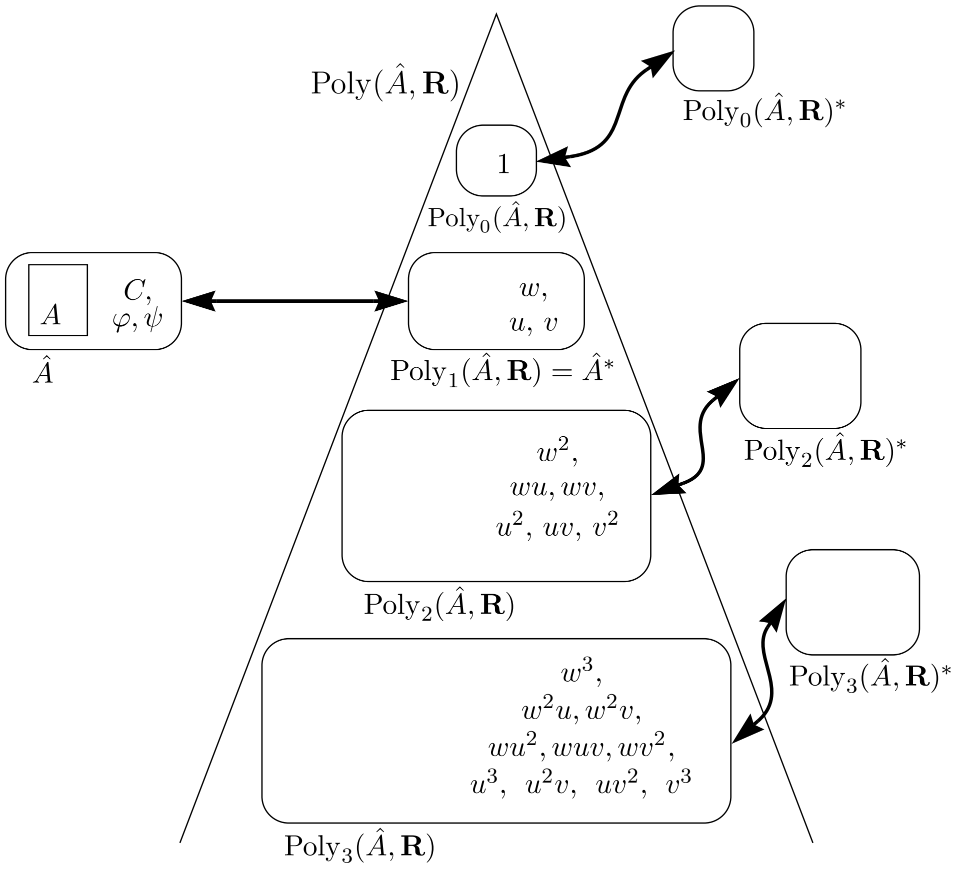


Figure 4.2: The homogenized framework

from point to anchor to site to location, and from affine space to linear space to algebra to field.

4.7 The algebra of forms

Figure 4.2 depicts the homogenized framework in the same schematic style in which Figure 3.1 depicted the nested-spaces framework. On the left, we have the linear 3-space $\hat{A}$ of anchors, with Cartesian basis (C, φ, ψ) and with the domain space A sitting inside it as the affine plane $w = 1$. On the right, for each nonnegative n, we have a rounded rectangle representing the linear space $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$ of n-forms on A. Note that an n-form on A can be evaluated at any anchor over A, whether or not that anchor is a point, and hence n-forms on A have all of $\hat{A}$ as their domain. The space $\mathrm{Poly}_0(\hat{A}, \mathbf{R})$ of constant forms is essentially the real numbers, with (1) as its obvious basis. The space $\mathrm{Poly}_1(\hat{A}, \mathbf{R})$ of linear forms has (w, u, v) as one possible basis — so that space is the same as the space $\hat{A}^*$ of coanchors on A. Indeed, for any linear space X, we have $\mathrm{Poly}_1(X, \mathbf{R}) = \mathrm{Lin}(X, \mathbf{R}) = X^*$, since 1-forms are the same thing as linear functionals. Next come quadratic forms on A, cubic forms, and so forth, where $\dim(\mathrm{Poly}_n(\hat{A}, \mathbf{R})) = \dim(\mathrm{Poly}_{\leq n}(A, \mathbf{R})) = \binom{n+2}{2}$.

Homogenization has pulled apart the primal spaces, so that they are no longer nested inside of one another; but they do fit together as the slices of a graded algebra. In particular, if f is an n-form on A and g is an m-form, the product fg is a form on A of degree $n + m$. The big triangle in Figure 4.2 represents this *algebra of forms*, which we denote $\mathrm{Poly}(\hat{A}, \mathbf{R})$. The algebra $\mathrm{Poly}(\hat{A}, \mathbf{R})$ is essentially a polynomial algebra; using our Cartesian basis (w, u, v) for the space of coanchors $\hat{A}^*$, we can think of it as the polynomial algebra $\mathbf{R}[w, u, v]$. It would be equally valid to use some other basis for the space $\hat{A}^*$, such as the barycentric basis (q, r, s); but let's stick with our chosen Cartesian basis until Section 4.9.

Note that the big triangle in Figure 4.2 is bigger than the union of the rectangles that it contains; the extra area represents inhomogeneous forms. If we add an m-form to an n-form, the sum is again a form on A, but it typically is not homogeneous. Such inhomogeneous forms don't seem to be good for anything, as far as CAGD is concerned; we shan't use them in this monograph. But they don't do much harm either. We view them as valid forms because we want the set of all forms on A to constitute an algebra — in particular, to be closed under addition. This problem of ending up with more primal objects than we really want did not arise in the nested-spaces framework; in that sense, inhomogeneous forms are a cost of homogenization. But the benefits of homogenization far outweigh its costs.

4.8 The dual spaces

For each nonnegative n, the linear space $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$ of n-forms on A has a dual $\mathrm{Poly}_n(\hat{A}, \mathbf{R})^*$, also shown as a rounded rectangle in Figure 4.2. As in the nested-spaces framework, the elements of the dual space $\mathrm{Poly}_n(\hat{A}, \mathbf{R})^*$ are typically called *dual functionals*. Evaluation at a point is one flavor of dual functional, as is evaluation at a vector or evaluation at any anchor. But there are also many dual functionals that don't correspond to evaluation.

The case $n = 1$ is special, since every dual functional on 1-forms does correspond to evaluation at a fixed anchor. Indeed, since the space $\mathrm{Poly}_1(\hat{A}, \mathbf{R})$ of 1-forms on A is the same as the space $\hat{A}^*$ of coanchors on A, it follows that the dual space $\mathrm{Poly}_1(\hat{A}, \mathbf{R})^*$ is simply $\hat{A}^{**} = \hat{A}$. This is reflected in Figure 4.2 by having the arrow leaving $\mathrm{Poly}_1(\hat{A}, \mathbf{R})$ point back to the domain space $\hat{A}$. We could make Figure 4.2 look more uniform if we moved the space $\hat{A}$ over to the right of the big triangle; but we leave $\hat{A}$ on the left, since that it where we are going to want it in later figures.

This confusion about left versus right arises because of a confusion in the homogenized framework about the direction in which to break the symmetry between primal and dual. Once the degree n exceeds 1, people using this framework typically think of an n-form in $\mathrm{Poly}_n(\hat{A}, \mathbf{R})$ as a primal object, suitable to be passed as an input datum to a dual functional in $\mathrm{Poly}_n(\hat{A}, \mathbf{R})^*$. In the special case $n = 1$, however, we surely want to think of a 1-form — that is, of a coanchor — as a dual object, since an anchor in $\hat{A}$, of which a point in A is a special case, seems quintessentially primal.

4.9 Linearization revisited

The key to the homogenized framework is the process of *linearization*, which takes an affine space A and constructs for us a naturally associated linear space $\hat{A}$ — in some sense, the free linear space generated by A. Let's take a moment to revisit how linearization works mathematically. There is an analogous, but less familiar, process of *algebrization*, which takes a linear space X and constructs for us a naturally associated commutative algebra $\mathrm{Sym}(X)$ — in some sense, the free commutative algebra generated by X. People in CAGD are already familiar with the algebra of forms, which is produced by algebrizing the linear space $\hat{A}^*$ of coanchors. In this monograph, we also algebrize the linear space $\hat{A}$ of anchors, thereby producing the algebra of sites. We can give ourselves a leg up on understanding algebrization if we polish our understanding of linearization.

In particular, we shall consider four approaches to linearization: fixing a frame, defining an equivalence relation, exploiting duality, and imposing a universal mapping condition. Those four approaches give us four different answers to the basic question, "What is an anchor?"

4.9.1 Fixing a frame

Let A be an affine space of dimension d that we want to linearize; that is, we want to construct the naturally associated linear space $\hat{A}$. The simplest approach involves choosing one particular reference frame for A and letting our construction of the linear space $\hat{A}$ depend upon that choice of frame.

For example, we might choose a Cartesian reference frame for A, say consisting of the point C in A and the d vectors $(\varphi_1, \ldots, \varphi_d)$ over A. Every point P in A can be written uniquely as $P = C + u_1(P)\varphi_1 + \cdots + u_d(P)\varphi_d$, where the coefficients $(u_1(P), \ldots, u_d(P))$ are the Cartesian coordinates of P. Having fixed that reference frame, we can construct the linearization $\hat{A}$ as the unique linear space that has $(C, \varphi_1, \ldots, \varphi_d)$ as a basis. An *anchor* over A is then, by definition, a linear combination $p = w(p)C + u_1(p)\varphi_1 + \cdots + u_d(p)\varphi_d$ of those $d + 1$ basis elements.

Of course, a barycentric reference frame would work just as well. Suppose that the points $(R_0, \ldots, R_d)$ are the vertices of a nondegenerate d-simplex in A, so that every point P in A can be written uniquely as an affine combination of the (R_i). Having fixed that frame, we could define the linearization $\hat{A}$ to be the unique linear space that has $(R_0, \ldots, R_d)$ as a basis. So an anchor, under this definition, is any linear combination $p = r_0(p)R_0 + \cdots + r_d(p)R_d$ of those $d+1$ points.

This process of fixing a frame is the simplest approach to linearization; it is both easy to understand and easy to prove theorems about. But it is unsettling to have the concept of an anchor over A appear to depend upon which reference frame for A we happen to have chosen.

Vacant remark: An affine space of dimension 0 is a single point. We could stop there, but let's go one more step, calling the empty set the unique affine space of dimension -1. There are a few anomalies in the case $d = -1$, and we'll comment about them in paragraphs, like this one, labeled "Vacant remark"; feel free to skip them.

One anomaly in the case $d = -1$ is that the empty affine space doesn't have any Cartesian reference frames; such a frame would have to consist of one point (of which there aren't any) and minus one vectors. But the empty affine space does have a barycentric reference frame, in fact, a unique one: the sequence with 0 points. The linearization of the empty affine space is the zero linear space, the space whose only element is 0. And the rule $\dim(\hat{A}) = \dim(A) + 1$ holds also when A is empty.

4.9.2 Defining an equivalence relation

Mathematicians have a standard technique for avoiding choices such as the choice of a reference frame: They make all choices simultaneously and then use an equivalence relation to collapse out the superfluous structure that results. Here's how we would linearize using that technique.

Given the affine space A, we first construct the unique linear space $L(A)$ that has A itself as a basis. The space $L(A)$ is huge — indeed, has uncountable dimension, one dimension for each point P in A. All of those extra dimensions allow us to draw too many distinctions. For example, suppose that $M := (P+Q)/2$ is the midpoint of the segment from P to Q in A. The expression $P/2 + Q/2 - M$ denotes a certain element of $L(A)$: the unique element with coordinate $1/2$ on the P axis, $1/2$ on the Q axis, -1 on the M axis, and 0 on all other axes. We don't want any such element in the linearization $\hat{A}$; more precisely, we want the expression $P/2 + Q/2 - M$ to denote 0 in $\hat{A}$ — that's what it means to say that $M = (P+Q)/2$. More generally, for every way of expressing a point Q in A as an affine combination of other points, say $Q = b_1P_1 + \cdots + b_mP_m$ with $b_1 + \cdots + b_m = 1$, we want to have $b_1P_1 + \cdots + b_mP_m - Q = 0$ in the linearization $\hat{A}$. To achieve that, let

$E(A)$ denote the smallest linear subspace of $L(A)$ that contains all elements of the form $b_1P_1 + \cdots + b_mP_m - Q$, where $Q = b_1P_1 + \cdots + b_mP_m$ in A. We then define the linearization $\hat{A}$ to be the quotient space $L(A)/E(A)$.

So, what is an anchor in this approach? An anchor over A is a huge equivalence class of linear combinations of points in A, under a certain equivalence relation. And how do we test whether two linear combinations, say $b_1P_1 + \cdots + b_nP_n$ and $c_1Q_1 + \cdots + c_mQ_m$, are equivalent? We first choose some reference frame for the affine space A. Having chosen such a frame, we expand each point P_i and each point Q_j as a linear combination of our frame elements. The two linear combinations of points that we started with are equivalent just when the two linear combinations of frame elements that result from this rewriting are equal. Note that we have to choose a frame in order to carry out this test, but the result of the test doesn't depend upon which frame we choose.

The advantage of this approach, over the fixed-frame approach, is that it gives us a notion of "anchor" that is independent of reference frame. But we pay a high price in mathematical complexity for that frame-independence: taking a quotient of linear spaces of uncountable dimension.

4.9.3 Exploiting duality

Defining an equivalence relation, as above, is the standard way that a mathematician would achieve frame-independence; but there are other ways. A more specialized trick that is available in this case exploits duality.

Given the affine space A, consider the space $\mathrm{Aff}(A, \mathbf{R})$ of all affine, real-valued maps on A. Since the space of real numbers $\mathbf{R}$ is linear, as well as affine, the space of maps $\mathrm{Aff}(A, \mathbf{R})$ is also linear, with addition and scalar multiplication defined pointwise. Hence, it makes sense to talk about the dual space $\mathrm{Aff}(A, \mathbf{R})^*$. And that dual space turns out to be a perfectly fine model for the linearization $\hat{A}$. Note that $\dim(\mathrm{Aff}(A, \mathbf{R})) = d + 1$, the extra $+1$ coming, in Cartesian coordinates, from the constant term. So $\dim(\mathrm{Aff}(A, \mathbf{R})^*) = d+1$ also, which is what we want for the linearization $\hat{A}$. We also want the affine space A to sit, in its linearization $\hat{A}$, as a hyperplane not containing the origin. If we make the definition $\hat{A} := \mathrm{Aff}(A, \mathbf{R})^*$, then A won't actually be a subspace of its linearization $\hat{A}$. But there will be a natural isomorphism from A to an affine hyperplane in $\hat{A}$: the map that takes a point P in A to the evaluate-at-P functional ϵ_P, the second-order functional defined, for all f in $\mathrm{Aff}(A, \mathbf{R})$, by $\epsilon_P(f) := f(P)$.

The reason that this trick works is that there is a natural one-to-one correspondence between $\mathrm{Aff}(A, \mathbf{R})$ and $\mathrm{Lin}(\hat{A}, \mathbf{R})$. In the forward direction, that's just a special case of homogenization. Recall from Section 4.3 that a polynomial map $f\colon A \to \mathbf{R}$ of degree at most n extends uniquely to an n-form $\hat{f}\colon \hat{A} \to \mathbf{R}$. Letting $n = 1$, we deduce that any f in $\mathrm{Aff}(A, \mathbf{R})$

extends uniquely to a 1-form $\hat{f}$ in $\mathrm{Lin}(\hat{A}, \mathbf{R})$. The reverse direction is even easier: We produce f from $\hat{f}$ by restricting the domain from $\hat{A}$ to A. Since $\mathrm{Aff}(A, \mathbf{R}) \approx \mathrm{Lin}(\hat{A}, \mathbf{R}) = \hat{A}^*$, it then follows that $\mathrm{Aff}(A, \mathbf{R})^* \approx \hat{A}^{**} = \hat{A}$.

In this approach, an anchor over A is a linear functional on $\mathrm{Aff}(A, \mathbf{R})$, which, given the one-to-one correspondence that we just discussed, is essentially the same thing as a linear functional on $\mathrm{Lin}(\hat{A}, \mathbf{R}) = \hat{A}^*$. So an anchor is essentially a linear functional on coanchors! That is pleasantly symmetric, since a coanchor is, of course, precisely a linear functional on anchors. It is the qualifier "essentially" that allows the resulting pair of definitions to avoid circularity. Indeed, the real work of linearization is in showing that the spaces $\mathrm{Aff}(A, \mathbf{R})$ and $\mathrm{Lin}(\hat{A}, \mathbf{R})$ are in natural one-to-one correspondence.

Math remark: Exploiting duality in this way requires that A be finite-dimensional, since we need the isomorphism $\hat{A}^{**} \approx \hat{A}$. In contrast, the first two approaches work fine to linearize affine spaces even of infinite dimension.

4.9.4 Imposing a universal mapping condition

We've now seen three concrete constructions of anchors, one dependent on a choice of reference frame and the other two frame-independent. There are further possibilities. For example, Berger [3] gives a frame-independent construction with a geometric flavor, in which an anchor over A turns out to be a vector field on A of a certain type.

Why don't multiple concrete constructions lead to chaos? Because we can characterize the linearization $\hat{A}$ abstractly, using a universal mapping condition. That condition does not determine the linearization uniquely, but does determine it up to a unique isomorphism. So any concrete construction must produce a result that is uniquely isomorphic to every other such result. We need to verify that one of the concrete constructions succeeds, in order to show that the universal mapping condition is satisfiable. But, once we have done that, it doesn't matter which concrete construction we employ, since they all give essentially the same result.

We shall return to these issues in greater depth in Chapter 9. But here, in brief, is how to characterize the linearization abstractly. A *linearization* of an affine space A is a pair (X, i) consisting of a linear space X and an affine map $i\colon A \to X$ that satisfies the following *universal mapping condition*:

> For every pair (Y, j) consisting of a linear space Y and an affine map $j\colon A \to Y$, there exists a unique linear map $f\colon X \to Y$ with $j = f \circ i$.

As it turns out, this universal mapping condition can be satisfied. Choose, for the space X, some linear space with $\dim(X) = \dim(A)+1$ and choose, for the map $i\colon A \to X$, some affine injection whose image, which will be an affine hyperplane in X, does not include the origin. Given any pair (Y, j), the values of f on the image of i are determined by the equation $j = f \circ i$; and those values extend uniquely to a linear map $f \colon X \to Y$. Furthermore, we shall see in Section 9.1 that, whenever some two pairs (X_1, i_1) and (X_2, i_2) both satisfy the universal mapping condition, there is a unique linear isomorphism between X_1 and X_2 that makes this diagram commute:

$$\begin{tikzcd} & A \arrow[dl, "i_1"'] \arrow[dr, "i_2"] & \\ X_1 \arrow[rr, leftrightarrow] & & X_2 \end{tikzcd}$$

Since any pair that satisfies the universal mapping condition is uniquely isomorphic to any other, we are justified in choosing any satisfying pair (X, i) that we like and referring to the space X in that pair as "the" linearization of A, denoting it $\hat{A}$. The affine injection $i \colon A \to \hat{A}$ allows us to identify A with the image of i, which is an affine hyperplane in $\hat{A}$ not containing the origin. Thus, it is also safe for us to pretend that the linearization $\hat{A}$ of A actually includes A as a subset: the set of unit-weight anchors.

4.9.5 So what is an anchor, really?

With this universal mapping condition in mind, we can now give the truest and deepest answer to the question, "What is an anchor?" Answer: An anchor over A is an element of some concrete linearization of A, but with the understanding that, if two different linearizations of A ever get involved in the same argument, we are required to use the unique isomorphism between them to identify each element of one with the corresponding element of the other. That is, we agree not to distinguish between different linearizations. So all of our earlier answers were correct simultaneously. An anchor over A is a linear combination of $(C, \varphi_1, \ldots, \varphi_d)$. It's also a linear combination of $(R_0, \ldots, R_d)$. It's also a huge equivalence class of linear combinations of points of A. And it's a linear functional on coanchors, and it's a vector field of a certain type, and so on. Speaking loosely, an anchor over A is an element of "the" linearization $\hat{A}$ of A.

Keep in mind that these same issues are going to arise again in defining sites. A site over A is, speaking loosely, an element of "the" algebrization $\mathrm{Sym}(\hat{A})$ of the linear space $\hat{A}$ of anchors. Given any linear space X, there is a universal mapping condition that determines when a commutative algebra is *an algebrization* of X. Since it follows from this condition that any two algebrizations of X are isomorphic in a unique way, we typically pretend that the algebrization $\mathrm{Sym}(X)$ is uniquely determined.

In fact, the same issues arose already in defining forms, although we didn't comment about them at the time. The algebra of forms is, we claim,

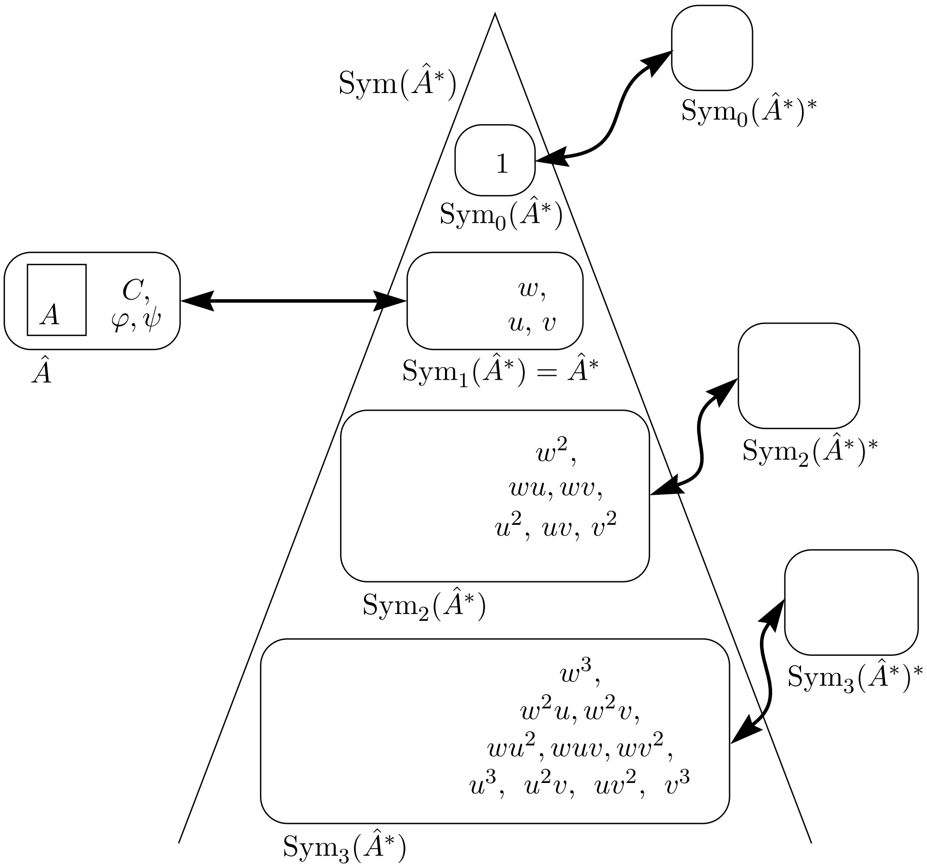


Figure 4.3: The homogenized framework with abstract labels

the algebrization $\operatorname{Sym}(\hat{A}^*)$ of the linear space $\hat{A}^*$ of coanchors. That claim should be plausible, because a form is, roughly speaking, a polynomial whose variables are coanchors. We introduced the algebra of forms in Section 4.7 as the algebra $\operatorname{Poly}(\hat{A}, \mathbf{R})$ of all polynomial, real-valued functions on $\hat{A}$. But that is simply one concrete construction of the abstract algebra $\operatorname{Sym}(\hat{A}^*)$. Indeed, for any linear space X, it turns out that we can exploit duality to construct the algebrization $\operatorname{Sym}(X)$ concretely as $\operatorname{Poly}(X^*, \mathbf{R})$. So one concrete model for the algebra of forms $\operatorname{Sym}(\hat{A}^*)$ is the algebra of functions $\operatorname{Poly}(\hat{A}^{**}, \mathbf{R}) = \operatorname{Poly}(\hat{A}, \mathbf{R})$.

Figure 4.3 shows the homogenized framework again, just as in Figure 4.2, except that the spaces are now labeled abstractly. For example, the space of n-forms on A, which used to be labeled $\operatorname{Poly}_n(\hat{A}, \mathbf{R})$, is now labeled $\operatorname{Sym}_n(\hat{A}^*)$. (Many authors write $\operatorname{Sym}^n(X)$ for the n^{th} graded slice of the algebra $\operatorname{Sym}(X)$, with the n as a superscript. We make the n a subscript just for consistency with the notations $\operatorname{Poly}_n(\hat{A}, \mathbf{R})$ and $\mathbf{R}_n[w, u, v]$.)

Chapter 5

The Separate-Algebras Framework

The homogenized framework has brought us closer to symmetry, in the sense that, in Figure 4.3, neither the primal spaces $(\mathrm{Sym}_n(\hat{A}^*))_{n\geq 0}$ nor the dual spaces $(\mathrm{Sym}_n(\hat{A}^*)^*)_{n\geq 0}$ are nested. But we still haven't achieved symmetry. The primal spaces fit together to make up the algebra of forms, while each dual space stands alone. Our eventual goal is the *paired-algebras framework*, in which the dual spaces fit together, in similar way, to make up the algebra of sites. But it is going to take us two steps to get there.

In the first of those two steps, we achieve symmetry in a brute-force way by treating the linear space $\hat{A}$ of anchors exactly as the homogenized framework treats the space $\hat{A}^*$ of coanchors. The result is the *separate-algebras framework*, shown in Figure 5.1. This framework has the serious drawback that there are four linear spaces associated with each degree n, the space of n-forms $\mathrm{Sym}_n(\hat{A}^*)$ and its dual $\mathrm{Sym}_n(\hat{A}^*)^*$ being joined by the space of n-sites $\mathrm{Sym}_n(\hat{A})$ and its dual $\mathrm{Sym}_n(\hat{A})^*$.

In the second step, we shall choose a sequence of pairing maps, the n^{th} of which pairs the space of n-forms with the space of n-sites, thereby allowing us to use each of those spaces to represent the dual of the other. This yields the paired-algebras framework, with just two spaces on the n^{th} level once again, rather than four. The reason that we delay taking this second step until Chapter 7 is that it entails a contentious choice about an annoying factor of $n!$. There are two sequences of pairing maps, in which the n^{th} maps differ by a factor of $n!$. Consider an n-form and an n-site, both of which happen to be perfect powers — say the n-form h^n and the n-site p^n, where h is a coanchor and p is an anchor. With one pairing, we have $\langle h^n, p^n\rangle = \langle h, p\rangle^n$; with the other, we have $\langle h^n, p^n\rangle = n!\,\langle h, p\rangle^n$. Sad to say, adopting either convention leaves us with annoying factors in many of our formulas, as we discuss in Appendix B. For now, let's get as far as we can using the separate-algebras framework, before tackling the annoying $n!$.

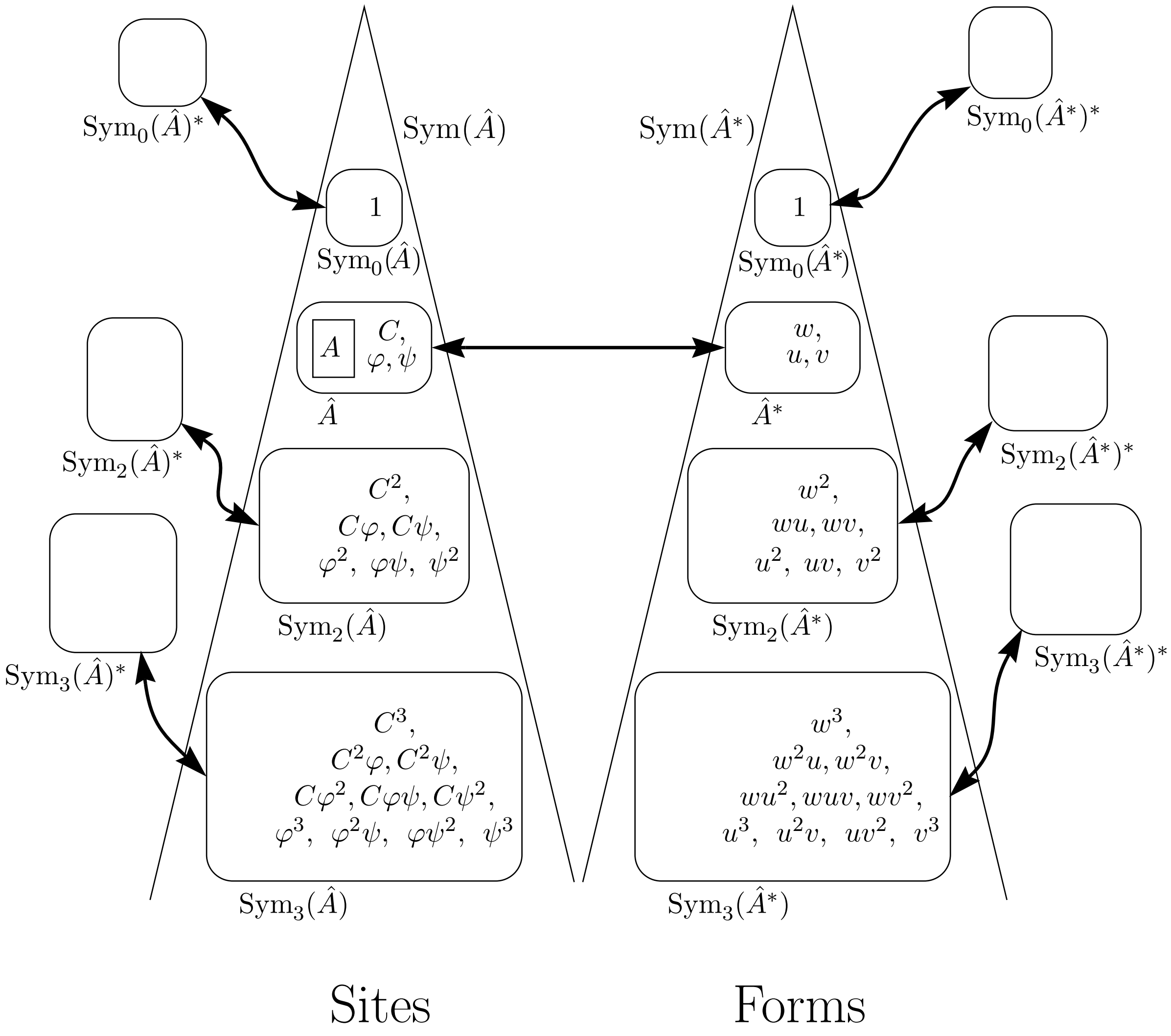


Figure 5.1: The separate-algebras framework

5.1 The algebra of sites

The triangle on the right in Figure 5.1 is the algebra of forms $\mathrm{Sym}(\hat{A}^*) = \mathrm{Poly}(\hat{A}, \mathbf{R})$, just as in the homogenized framework. In a completely symmetric way, the triangle on the left is the algebra $\mathrm{Sym}(\hat{A}) = \mathrm{Poly}(\hat{A}^*, \mathbf{R})$, which we christen the *algebra of sites*.

This may be as good a time as any to discuss why I chose the word "site". I wanted a noun that would accept a numeric prefix, so that I could talk about n-sites as being dual to n-forms; that strongly suggested a one-syllable noun. I also wanted a noun that means something like "point". The nouns "place" and "site" met those criteria. Unfortunately, both of those words have preexisting meanings in algebraic geometry. A *place* on a curve is an equivalence class of irreducible parameterizations — roughly speaking, a point on a branch of the curve [1, 48]. From the Encyclopedic Dictionary of Mathematics [35], I learned that a *site* is a category in which each object comes equipped with a covering family of morphisms that fit together to form a Grothendieck topology. I hope that Grothendieck topologies are high-powered enough that no confusion will arise between that meaning of "site" and sites as the duals of forms.

The equality $\mathrm{Sym}(\hat{A}) = \mathrm{Poly}(\hat{A}^*, \mathbf{R})$ suggests that a site is a real-valued, polynomial function on coanchors; and indeed, that is one of various equivalent ways to define a site. But viewing sites from that perspective is not the best way to get to know them. Keep in mind that, roughly speaking, sites are polynomials whose variables are anchors, just as forms are polynomials whose variables are coanchors. Let's refer to such polynomials as *anchor polynomials* and *coanchor polynomials*.

For definiteness, let's assume once again that A is an affine plane, of dimension $d = 2$. What is a site over A, more precisely? For that matter, what is a form on A? Both questions have an abstract answer, given on the first line of Table 5.1, and a variety of concrete answers, three of which are given on the following lines. Table 5.2 shows the four different ways in which we shall denote the linear space of all n-sites over A and the linear space of all n-forms on A, corresponding to the four lines in Table 5.1. So each of the bottom three lines names a concrete construction for the linear space that the top line names abstractly.

5.1.1 Imposing a universal mapping condition

The linearization $\hat{A}$ of an affine space A is a linear space that satisfies a certain universal mapping condition. In a similar way, the algebrization $\mathrm{Sym}(X)$ of a linear space X is a commutative algebra that satisfies a universal mapping condition. We shall discuss that condition and related issues in Chapter 9. Until then, just keep in mind that there is an abstract characterization of

site	form
an element of the algebrization $\mathrm{Sym}(\hat{A})$ of the linear space $\hat{A}$	an element of the algebrization $\mathrm{Sym}(\hat{A}^*)$ of the linear space $\hat{A}^*$
a polynomial in the anchor variables (C, φ, ψ)	a polynomial in the coanchor variables (w, u, v)
an equivalence class of anchor polynomials whose variables are arbitrary anchors	an equivalence class of coanchor polynomials whose variables are arbitrary coanchors
a real-valued function on coanchors that can be defined by some anchor polynomial — in fact, by an equivalence class of anchor polynomials	a real-valued function on anchors that can be defined by some coanchor polynomial — in fact, by an equivalence class of coanchor polynomials

Table 5.1: What are sites and forms?

the algebra of sites that determines it up to a unique isomorphism. So which concrete construction we adopt for that algebra doesn't matter.

5.1.2 Fixing a basis

Given a linear space X of dimension k, the simplest way to construct the symmetric algebra $\mathrm{Sym}(X)$ is to fix a basis $(\xi_1, \ldots, \xi_k)$ for X and then to construct $\mathrm{Sym}(X)$ as the algebra $\mathbf{R}[\xi_1, \ldots, \xi_k]$ of all polynomials in those k basis elements, treated as variables. Using this approach, we can construct the algebra of forms $\mathrm{Sym}(\hat{A}^*)$ as $\mathbf{R}[w, u, v]$, and we can construct the algebra of sites $\mathrm{Sym}(\hat{A})$ as $\mathbf{R}[C, \varphi, \psi]$. While this approach is delightfully simple, it might seem to unfairly favor the fixed basis.

5.1.3 Defining an equivalence relation

We would prefer to use different bases at different times and, even better, to use multiple bases simultaneously. For example, we would like any coanchor polynomial to denote a form, even if its variables don't all come from any single basis for the space $\hat{A}^*$ of coanchors. Once we allow coanchors that are linearly dependent, however, distinct polynomials may denote the same form. For example, the linear dependence $q + r + s = w$ tells us that the two polynomials $q + r + s$ and w denote the same 1-form — to wit, the weight coanchor. It follows that the quadratic polynomials $qu+ru+su = (q+r+s)u$ and wu must denote the same 2-form. Thus, we can think of a form as an

space of n-sites	space of n-forms
$\mathrm{Sym}_n(\hat{A})$	$\mathrm{Sym}_n(\hat{A}^*)$
$\mathbf{R}_n[C, \varphi, \psi]$	$\mathbf{R}_n[w, u, v]$
$\mathbf{R}_n[\hat{A}]/\approx_{\hat{A}}$	$\mathbf{R}_n[\hat{A}^*]/\approx_{\hat{A}^*}$
$\mathrm{Poly}_n(\hat{A}^*, \mathbf{R})$	$\mathrm{Poly}_n(\hat{A}, \mathbf{R})$

Table 5.2: Formulas for the spaces of n-sites and n-forms

equivalence class of coanchor polynomials.

Given two coanchor polynomials, elements of the huge algebra $\mathbf{R}[\hat{A}^*]$, how do we test whether they are equivalent? Answer: We rewrite all of the coanchors in both of them as linear combinations of w, u, and v and check whether the rewritten polynomials coincide. Of course, there is nothing special about the basis (w, u, v); we can adopt any basis for $\hat{A}^*$ in performing this equivalence test without affecting the result.

The same goes for sites. We would like any anchor polynomial in $\mathbf{R}[\hat{A}]$ to denote a site, even if its anchors don't all come from any single basis for $\hat{A}$. But once we allow anchors that are linearly dependent, distinct polynomials may denote the same site. For example, let $E := (Q + R + S)/3$ be the centroid of the reference triangle $\triangle QRS$. The linear polynomials $3E$ and $Q + R + S$ denote the same 1-site — an anchor of weight 3. It follows that the quadratic polynomials $3E\psi$ and $Q\psi + R\psi + S\psi = (Q + R + S)\psi$ denote the same 2-site. We can think of a site as an equivalence class of anchor polynomials, where two such polynomials are equivalent when rewriting all of the anchors in both of them as linear combinations of the anchors in a common basis for $\hat{A}$ would make them coincide; and which common basis we adopt in this test of equivalence doesn't matter.

5.1.4 Exploiting duality

One thing that you can do with a coanchor polynomial is to define a real-valued function on anchors; for example, the coanchor polynomial wu defines the function that takes an anchor p to the real number $w(p)u(p) = \langle w, p\rangle\langle u, p\rangle$. As it happens, we are quite interested in real-valued functions on anchors, since we intend to use three of them to define the x, y, and z coordinates of our Bézier triangle. Note that the coanchor polynomial $qu+ru+su$ defines the same real-valued function as does wu; so each real-valued function actually arises from some equivalence class of coanchor polynomials. In fact, these equivalence classes are the same ones that we introduced in the third row of Table 5.1. Thus, we can think of a form either as an equivalence class of coanchor polynomials or as the common real-valued function on anchors that any one of those equivalent polynomials defines.

The same goes for sites, except that CAGD doesn't give us any particular reason to be interested in the resulting real-valued functions. An anchor polynomial defines a real-valued function on coanchors; for example, the anchor polynomial $3E\psi$ defines the function that takes a coanchor h to the real number $3E(h)\psi(h) = 3h(E)h(\psi) = 3\langle h, E\rangle\langle h, \psi\rangle$. Two anchor polynomials define the same real-valued function just when they are equivalent, in the sense of the third row of Table 5.1. For example, the equivalent polynomials $3E\psi$ and $Q\psi + R\psi + S\psi$ define the same real-valued function. Thus, we can think of a site either as an equivalence class of anchor polynomials or as the common real-valued function on coanchors that any one of those equivalent polynomials defines.

Mathematically, forms and sites are completely symmetric; but their applications to CAGD are not. Since CAGD gives us good uses for real-valued functions on anchors, the definition of forms in the fourth row seems more attractive than the one in the third row. Indeed, when we first defined forms in Chapter 4, we talked only about real-valued functions on anchors, leaving implicit the equivalence relation on coanchor polynomials. But CAGD does not give us similarly good uses for real-valued functions on coanchors. Hence, in defining sites, the third row seems more attractive than the fourth.

Math remark: Why does it work to algebrize by exploiting duality, that is, to construct $\mathrm{Sym}(X)$ as $\mathrm{Poly}(X^*, \mathbf{R})$? It works because, over the real numbers, the coefficients of a polynomial are uniquely determined by that polynomial's values. The same works over any infinite field, even infinite fields of prime characteristic. But not over finite fields. Let p be a prime. Over the Galois field of order p^k, the two polynomials ξ^{p^k} and ξ have all of the same values; thus, two distinct elements of $\mathrm{Sym}(X)$ would be indistinguishable as elements of $\mathrm{Poly}(X^*, \mathbf{R})$. For more on this, see Section C.2.3.

5.2 The weight of a site

While we are not typically interested in sites as real-valued functions on coanchors — they are going to be useful to us for other reasons — there is one coanchor at which we do want to evaluate our sites: w, the weight coanchor. If s is any site over A, we define the real number $s(w)$ to be the *weight* of s. If s is a 1-site over A, that is, if $s = p$ is an anchor, we have $s(w) = p(w) = w(p) = \langle w, p\rangle$, so this definition agrees with our former notion for the weight of an anchor. More generally, suppose that we are given an n-site s explicitly, in terms of our Cartesian basis, as a linear combination of the n-sites $(C^{n-i-j}\varphi^i\psi^j)_{i+j\leq n}$. Since C is a point, of weight 1, while φ and ψ are vectors, of weight 0, the weight of s is simply the coefficient of C^n in this expansion.

Exercise 5.2-1 If we think of a real number t as a 0-site over A, what is its weight $t(w)$? (Answer: Working from the fourth row of Table 5.1, a 0-site is a real-valued function on coanchors that is homogeneous of degree 0, that is, a constant function. So $t(h) = t$ for all coanchors h, including $t(w) = t$. Working from the second row gives the same result: The coefficient of C^0 in the expansion $t = tC^0$ is t.)

Now that we have defined a notion of weight for n-sites, let's introduce the notation $\mathrm{Sym}_n(\hat{A})^{\downarrow}$ for the affine space of all n-sites over A that are of unit weight. Thus, if we construct our sites by fixing the Cartesian reference frame (C, φ, ψ), then the affine subspace $\mathrm{Sym}_n(\hat{A})^{\downarrow} = \mathbf{R}_n[C, \varphi, \psi]^{\downarrow}$ consists of all polynomials that are homogeneous of degree n in C, φ, and ψ and in which the coefficient of the term C^n is 1. The intuition behind the downward arrow is that restricting to unit weight is like undoing linearization. If linearization lifts us from the affine space A to the linear space $\hat{A} = A^{\uparrow}$, with its associated weight functional, then restricting to unit weight takes us back again: $\hat{A}^{\downarrow} = A^{\uparrow\downarrow} = A$. Note that it makes sense to restrict to unit weight only when the context makes clear which weight functional is intended; it wouldn't make sense to write $\mathrm{Sym}_n(\hat{A}^*)^{\downarrow}$, for example, since we haven't defined a notion of weight for n-forms.

5.3 The dual spaces

Each graded slice of the algebra of forms is a linear space, which has a dual, and the same is true for the algebra of sites. Thus, Figure 5.1 has four linear spaces on the n^{th} level, for $n \neq 1$. But the case $n = 1$ is special. The linear space $\mathrm{Sym}_1(\hat{A}) = \mathrm{Poly}_1(\hat{A}^*, \mathbf{R})$ of 1-sites is simply the space $\hat{A} = \hat{A}^{**}$ of anchors, just as the linear space $\mathrm{Sym}_1(\hat{A}^*) = \mathrm{Poly}_1(\hat{A}, \mathbf{R})$ of 1-forms is the space $\hat{A}^*$ of coanchors. The two spaces $\hat{A}$ and $\hat{A}^*$ formed a dual pair already in the homogenized framework, and we have no reason to break up that pairing now. Thus, there are only two spaces on the first level in Figure 5.1, and the double-headed arrow on that level simply links those spaces to each other. When we are ready to tackle the annoying $n!$ in Chapter 7, we'll be able to cut back to only two spaces on every level.

While the separate-algebras framework of Figure 5.1 has twice as many spaces as it should, at least it comes close to achieving perfect symmetry between forms and sites. Indeed, had we started with an arbitrary dual pair of linear spaces (X, Y) in building the algebras of forms and sites, $\mathrm{Sym}(X)$ and $\mathrm{Sym}(Y)$, the symmetry would have been perfect. But the spaces in the pair

$(\hat{A}^*, \hat{A})$ are not arbitrary. In the space $\hat{A}$ of anchors, we have distinguished the hyperplane A of points as being of special interest. There is no analogous hyperplane in the space $\hat{A}^*$ of coanchors; instead, it is a particular coanchor, the weight coanchor w, that is distinguished by the equation $A = w^{-1}(1)$. Those two distinguished structures, each of which determines the other, are the sole source of mathematical asymmetry in the separate-algebras framework. But keep in mind that there is also some motivational asymmetry; for example, we have practical applications in mind for real-valued functions on anchors, but none for real-valued functions on coanchors.

5.4 Flavors of evaluation

Before we leave Figure 5.1, let's review the three flavors of evaluation that we have defined. Consider a site-like object $\mathcal{S}$ and a form-like object $\mathcal{F}$. In what situations can we evaluate one of them at the other?

$\mathcal{S}$	$\mathcal{F}$
anchor	coanchor
anchor	form
site	coanchor

All three of our evaluations have, at their core, the pairing between the fundamental spaces $\hat{A}$ and $\hat{A}^*$, between anchors and coanchors. Given any anchor p over A and coanchor h on A, we can view pairing the two of them as evaluating either of them at the other, depending upon how we are breaking the symmetry at the moment: $p(h) = h(p) = \langle h, p\rangle$.

Let p and q be anchors, while g and h are coanchors. In building the algebra of forms, we learned how to evaluate a form at an anchor. For example, we have $(g^3 + 5gh)(p) = g(p)^3 + 5g(p)h(p)$. In building the algebra of sites, we learned how to evaluate a site at a coanchor in a completely analogous way, modulo a little notational confusion. For example, we have $(p^3 + 5pq)(h) = p(h)^3 + 5p(h)q(h)$ — though we might prefer to write that value as $h(p)^3 + 5h(p)h(q)$.

Later on, we shall study various other ways in which to combine forms and sites: pairing an n-form with an n-site to produce a scalar, contracting an n-form on a k-site to produce an $(n-k)$-form, and so on. But we shall reserve the term "evaluation" for these three flavors of combination.

Chapter 6

The Veronese Prototypes

Let A be an affine parameter space of dimension d. We have constructed the algebra $\mathrm{Sym}(\hat{A})$ of sites over A, in parallel with the well-known algebra $\mathrm{Sym}(\hat{A}^*)$ of forms on A. In Chapter 7, we are going to pair up, for each n, the linear spaces $\mathrm{Sym}_n(\hat{A})$ and $\mathrm{Sym}_n(\hat{A}^*)$ of n-sites and n-forms, so that each of them can represent the dual of the other. But the algebra of sites has important applications in CAGD, even without those pairing maps.

The key to those applications is the set of n-sites that are perfect n^{th} powers. The linear space $\mathrm{Sym}_n(\hat{A})$ of n-sites over A has dimension $\binom{n+d}{n}$. Inside that big space, we focus on those n-sites s that have the special form $s = P^n$, for some point P in A. Those n-sites make up a certain d-fold — a curve when $d = 1$, a surface when $d = 2$. This d-fold is important to CAGD because it can serve as a prototype for all polynomial, parametric d-folds of degree at most n. For example, when $d = 1$, the geometric structure of an n-ic Bézier curve is best understood by viewing that curve as an affine transform of the *moment curve of degree* n, the curve κ_n that maps a point P on its parameter line to the n-site $\kappa_n(P) := P^n$. In a similar way, when $d = 2$, an n-ic Bézier triangular surface is best understood by viewing it as an affine transform of the *Veronese surface of parametric degree* n, the surface σ_n that maps a point P on its parameter plane to the n-site $\sigma_n(P) := P^n$.

6.1 Quadratic sites over the line

We begin with the case $n = 2$ and $d = 1$, looking for those quadratic sites over the line that are perfect squares.

As a convenient line to work with, let's take the u axis of the plane $A = \{C + u\varphi + v\psi \mid u, v \in \mathbf{R}\}$ shown in Figure 4.1; that is, let's take the affine line L given by $L := \{C + u\varphi \mid u \in \mathbf{R}\}$. When our parameter space is a single line, it is convenient to adopt some scheme that names a point on that line using a single real number. So, for any real number t, let $\bar{t}$ denote

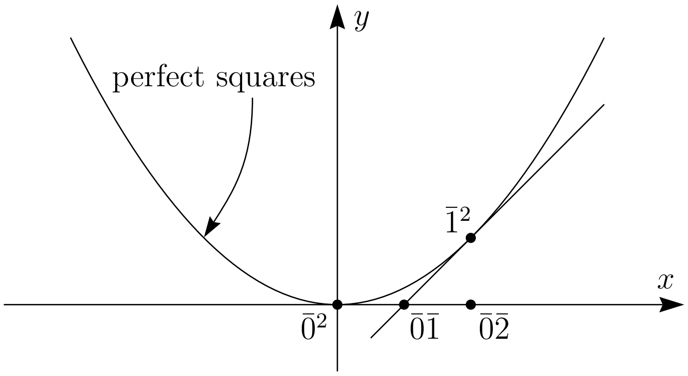


Figure 6.1: The plane of unit-weight 2-sites over the line L.

the point on the line L whose u-coordinate is t; that is, we set $\bar{t} := C + t\varphi$.

We want to study the linear space $\mathrm{Sym}_2(\hat{L})$ of quadratic sites over the line L. We know of three different concrete constructions for that space, corresponding to the last three lines in Table 5.1: $\mathrm{Sym}_2(\hat{L}) = \mathbf{R}_2[C, \varphi] = (\mathbf{R}_2[\hat{L}]/\approx_{\hat{L}}) = \mathrm{Poly}_2(\hat{L}^*, \mathbf{R})$. In this section, we'll stick to the first of those three and study the space $\mathbf{R}_2[C, \varphi]$, a linear 3-space of quadratic polynomials:

$$\mathrm{Sym}_2(\hat{L}) = \mathbf{R}_2[C, \varphi] = \{r\,C^2 + x\,C\varphi + y\,\varphi^2 \mid r, x, y \in \mathbf{R}\}$$

We can save one dimension by restricting our attention to those quadratic sites over L that have unit weight, that is, to the case $r = 1$; such sites form the affine plane

$$\mathrm{Sym}_2(\hat{L})^{\downarrow} = \mathbf{R}_2[C, \varphi]^{\downarrow} = \{C^2 + x\,C\varphi + y\,\varphi^2 \mid x, y \in \mathbf{R}\}.$$

That plane is pictured, using x and y as coordinates, in Figure 6.1.

By elementary algebra, we can plot various unit-weight 2-sites over L as points in Figure 6.1. For example, we have $\bar{1}^2 = (C + \varphi)^2 = C^2 + 2\,C\varphi + \varphi^2$, so we plot the 2-site $\bar{1}^2$ at the spot $x = 2$ and $y = 1$. More generally, for any real number t, we have $\bar{t}^2 = (C + t\varphi)^2 = C^2 + 2t\,C\varphi + t^2\,\varphi^2$, with $x = 2t$ and $y = t^2$. Thus, the squares of the points on L form a parabola in the plane of Figure 6.1: the parabola with equation $x^2 - 4y = 0$. Note that $x^2 - 4y$ is the discriminant of the quadratic polynomial $C^2 + x\,C\varphi + y\,\varphi^2$; this makes sense, since the algebra of sites is essentially a polynomial algebra.

Let's also plot the sites of the form $\bar{0}\bar{t}$. We calculate $\bar{0}\bar{t} = C(C + t\varphi) = C^2 + t\,C\varphi$. Thus, those sites constitute the x-axis in Figure 6.1, which is the tangent line to the parabola of squares at the 2-site $\bar{0}^2$.

The point $\bar{0}$ on the parameter line L is no different from any other point on L; so we suspect that, for any real number a, the tangent line to the parabola of squares at the 2-site $\bar{a}^2$ should comprise those 2-sites of the form

$\bar{a}\bar{b}$, for varying b. The latter sites clearly form some line, and that line clearly passes through the site $\bar{a}^2$ when $b := a$. All that remains to verify is the tangency. So consider how the 2-site $\overline{a+h}^2$ behaves, as the real number h tends to zero; we have

$$\begin{aligned}\overline{a+h}^2 &= (\bar{a} + h\,\varphi)^2 = \bar{a}^2 + 2h\,\bar{a}\varphi + h^2\,\varphi^2\\ &= \bar{a}\,(\bar{a} + 2h\,\varphi) + h^2\,\varphi^2 = \bar{a}\;\overline{a+2h} + h^2\,\varphi^2.\end{aligned}$$

Since h^2 goes to zero faster than h, the tangent line to the parabola at $\bar{a}^2$ is indeed the line whose 2-sites have the form $\bar{a}\bar{b}$. (Note that we carried out that analysis using $(\bar{a}, \varphi)$ as our basis for the linear space $\hat{L}$ of anchors over L, rather than the standard basis $(C, \varphi) = (\bar{0}, \varphi)$. There are many such situations where it is convenient to use some non-standard basis.)

Let $s = C^2 + x\,C\varphi + y\,\varphi^2$ be any 2-site in the plane $\mathrm{Sym}_2(\hat{L})^{\downarrow}$ of Figure 6.1. The site s lies outside the parabola just when its discriminant $x^2 - 4y$ is positive. In that case, we can see geometrically that s is the intersection of the tangent lines to the parabola at $\bar{a}^2$ and $\bar{b}^2$, for some two distinct real numbers a and b. So the 2-site s must have both $\bar{a}$ and $\bar{b}$ as factors, which means that $s = \bar{a}\bar{b}$ must be the product of the points $\bar{a}$ and $\bar{b}$ on L. If we like, we can use the Quadratic Formula to compute a and b; we have

$$C^2 + x\,C\varphi + y\,\varphi^2 = \left(C + \frac{x + \sqrt{x^2 - 4y}}{2}\,\varphi\right)\left(C + \frac{x - \sqrt{x^2 - 4y}}{2}\,\varphi\right),$$

and hence

$$s = \overline{\left(\frac{x + \sqrt{x^2 - 4y}}{2}\right)}\;\overline{\left(\frac{x - \sqrt{x^2 - 4y}}{2}\right)}.$$

Those 2-sites $s = C^2 + x\,C\varphi + y\,\varphi^2$ that lie inside the parabola, with $x^2 - 4y$ negative, don't factor as the product of two points over the real numbers. They would factor over the complex numbers; but the real numbers are the scalars of primary interest in CAGD.

Warning: While the product of any n anchors is an n-site, it is by no means the case that every n-site splits as the product of n anchors. Here already, in studying Figure 6.1, we have examples of 2-sites that don't split: the ones inside the parabola of squares. Once the parametric dimension d exceeds 1, the sites that do split, even over the complex numbers, become a tiny minority of all sites; see Exercise 6.7-1.

For any affine space A and for any nonnegative n, we shall say that an n-form on A is *real-lineal* when it splits — that is, factors completely — over the reals as the product of n coanchors on A. When the n factors are allowed to be complex, we'll use the term *complex-lineal*. Similarly, an n-site is *real-lineal* or *complex-lineal* when it factors, over the reals or complexes, as the product of n anchors. The 2-sites in Figure 6.1 that are real-lineal are those that lie either on the parabola of perfect squares or outside it. All of the 2-sites in Figure 6.1 — indeed, all n-sites over the line L for every n — are complex-lineal.

More generally, given any graded algebra (not necessarily commutative), a homogeneous element of grade n in that algebra is called *lineal* when it can be expressed as the product of n linear elements, that is, of n elements that are homogeneous of grade 1. Some synonyms for this sense of "lineal" are "simple", "totally decomposable", and "completely reducible". (A fine point: Under this definition, the multiplicative identity 1 is the only homogeneous element of grade 0 that is lineal, since 1 is the value of the unique empty product. So, for example, 1 is the only 0-form that is lineal and also the only 0-site that is lineal. Some authors are more generous, calling an n-ic thing lineal whenever it can be written as a scalar multiple of a product of n linear things; under that definition, all scalars are lineal.)

Note that the same geometric and algebraic properties that hold of sites hold also of forms, the only exceptions being those that involve the weight coanchor. For example, consider a quadratic form on the line L, an element $f := a\,w^2 + b\,wu + c\,u^2$ of the linear 3-space $\mathrm{Sym}_2(\hat{L}^*) = \mathbf{R}_2[w,u]$. The forms f with $b^2 - 4ac = 0$ are plus or minus the square of a coanchor; they form a quadratic cone in the space $\mathrm{Sym}_2(\hat{L}^*)$. The forms f that lie on or outside that cone have $b^2 - 4ac \geq 0$, and they factor as the product of two coanchors. Returning to the algebra of sites, we have precisely similar structures. A quadratic site $s := r\,C^2 + x\,C\varphi + y\,\varphi^2$ of weight r is plus or minus the square of an anchor just when $x^2 - 4ry = 0$, and the sites s that lie on or outside that cone, with $x^2 - 4ry \geq 0$, are the ones that factor as the product of two anchors. In our analysis above, we restricted ourselves to the plane $r = 1$ of unit-weight sites, where that plane cuts the cone of squares in the parabola of Figure 6.1. The analogous restriction for quadratic forms would require the coefficient a of w^2 to be 1, that is, would restrict our attention to forms whose value at the point C happens to be 1. But such a restriction would be unnatural, since we made an arbitrary choice when we selected C as the center point of our coordinate system for the line L.

6.2 The prototypical parabola

The parabola in Figure 6.1 is the image of the affine line L under the squaring map, the map $\kappa_2\colon L \to \mathrm{Sym}_2(\hat{L})^{\downarrow}$ that takes a point P on L as its argument and squares it: $\kappa_2(P) := P^2$. Expressing P in terms of our standard basis $P = \bar{u} = C + u(P)\varphi$, we have $\kappa_2(\bar{u}) = C^2 + 2u\,C\varphi + u^2\,\varphi^2$, as above.

Now, suppose that we want to use an arc of a parabola as part of a spline curve that we are designing. Let $F\colon L \to O$ be that parabola, sitting in some plane in our object space O, and suppose we want to use the arc $F([\bar{a} \mathinner{..} \bar{b}])$. No matter what design methodology we employ, the parabola F will be an affine transform of the particular parabola κ_2. That is, there will exist an *instancing transformation*, an affine map $f\colon \mathrm{Sym}_2(\hat{L})^{\downarrow} \to O$ with $F(P) = f(\kappa_2(P)) = f(P^2)$, for all points P on L. In this way, the particular parabola κ_2 can serve as a prototype for all parabolas.

To add the parabolic arc $F([\bar{a} \mathinner{..} \bar{b}])$ to our design, it suffices to specify the instancing transformation f. And one simple way to specify the map f is to specify the images under f of the three sites $\bar{a}^2$, $\bar{a}\bar{b}$, and $\bar{b}^2$; note that, whenever the real numbers a and b are distinct, those three sites constitute an affine frame for the plane $\mathrm{Sym}_2(\hat{L})^{\downarrow}$ of Figure 6.1. The images of those three sites under f are, of course, the three Bézier points of the parabolic segment $F([\bar{a} \mathinner{..} \bar{b}])$. To see this algebraically, suppose that the point P on L is located t of the way from $\bar{a}$ to $\bar{b}$, so that $P = (1-t)\bar{a} + t\bar{b}$. We then have

$$\begin{aligned} F(P) &= f(\kappa_2(P)) = f(P^2) \\ &= f\bigl(((1-t)\bar{a} + t\bar{b}\,)^2\bigr) \\ &= f\bigl((1-t)^2\,\bar{a}^2 + 2t(1-t)\,\bar{a}\bar{b} + t^2\,\bar{b}^2\bigr) \\ &= (1-t)^2 f(\bar{a}^2) + 2t(1-t) f(\bar{a}\bar{b}) + t^2 f(\bar{b}^2). \end{aligned}$$

In degenerate cases, we might choose the three Bézier points $f(\bar{a}^2)$, $f(\bar{a}\bar{b})$, and $f(\bar{b}^2)$ to be collinear, or even choose all three to coincide. The instancing transformation f would then collapse the plane of Figure 6.1 down either to a line or to a point. But this collapsing happens only to our parabolic instance F, not to the prototypical parabola κ_2.

6.3 The moment curves

In a similar way, the n^{th}-power map $\kappa_n\colon L \to \mathrm{Sym}_n(\hat{L})^{\downarrow}$ defined by $\kappa_n(P) := P^n$ provides a prototype for all polynomial curves of degree at most n. For example, when $n = 3$, we have $\bar{t}^{\,3} = (C + t\varphi)^3 = C^3 + 3t\,C^2\varphi + 3t^2\,C\varphi^2 + t^3\,\varphi^3$, so our prototypical cubic is the twisted cubic curve $(x, y, z) := (3t, 3t^2, t^3)$, sitting in the affine 3-space

$$\mathrm{Sym}_3(\hat{L})^{\downarrow} = \mathbf{R}_3[C, \varphi]^{\downarrow} = \{C^3 + x\,C^2\varphi + y\,C\varphi^2 + z\,\varphi^3 \mid x, y, z \in \mathbf{R}\}.$$

In projective geometry, the curve that results from the analogous construction is called the *rational normal curve of degree n*. Since we are working in an affine space, instead of in its projective closure, we'll refer to κ_n by its other name: the *moment curve of degree n*.

The tangent lines, osculating planes, and so forth of the moment curve κ_n are related to the multiplication in the algebra of sites as follows.

Proposition 6.3-1 *Let $\kappa_n \colon L \to \mathrm{Sym}_n(\hat{L})^{\downarrow}$ be the moment curve of degree n given by $\kappa_n(P) := P^n$, for all points P on the affine line L. A unit-weight n-site s over L lies in the affine k-flat that osculates the curve κ_n to k^{th} order at P^n just when the $(n-k)$-site P^{n-k} divides s.*

Proof As the real number h tends to 0, we have

$$\begin{aligned}\overline{t+h}^{\,n} = (\bar{t}+h\varphi)^n &= \sum_{0\le i\le n}\binom{n}{i}\bar{t}^{\,n-i}h^i\varphi^i\\ &= \sum_{0\le i\le k}\binom{n}{i}\bar{t}^{\,n-i}h^i\varphi^i + O(h^{k+1})\\ &= \bar{t}^{\,n-k}\sum_{0\le i\le k}\binom{n}{i}\bar{t}^{\,k-i}h^i\varphi^i + O(h^{k+1}).\end{aligned}$$

Thus, the moment curve κ_n is approximated to k^{th} order, near the n-site $\bar{t}^{\,n}$, by the k-flat that consists of all multiples of $\bar{t}^{\,n-k}$. □

For example, consider the 3-site $\bar{a}\bar{b}^2$. The one factor of $\bar{a}$ puts us in the osculating plane to the twisted cubic κ_3 at $\bar{a}^3$, while the two factors of $\bar{b}$ put us in the osculating plane at $\bar{b}^3$ twice, that is, on the tangent line at $\bar{b}^3$. So the 3-site $\bar{a}\bar{b}^2$ sits where that tangent line cuts that osculating plane.

The moment curve κ_n can serve as a prototype for all n-ic polynomial parametric curves. Given any such curve $F \colon L \to O$, sitting in some object space O, there exists a unique affine map $f\colon \mathrm{Sym}_n(\hat{L})^{\downarrow} \to O$ that realizes F as an instance of the prototype κ_n, that is, that satisfies $F(P) = f(\kappa_n(P)) = f(P^n)$, for all points P on L. Given some parameter interval $[\bar{a} \mathinner{..} \bar{b}]$ on L, one convenient way to determine which n-ic segment $F([\bar{a} \mathinner{..} \bar{b}])$ we want in some design of ours is to specify the instancing transformation f by specifying the images under f of the n-sites $\bar{a}^n$, $\bar{a}^{n-1}\bar{b}$, through $\bar{b}^n$, those images being the Bézier points of the segment $F([\bar{a} \mathinner{..} \bar{b}])$.

Note that the instancing transformation f may well fail to be injective. Indeed, the prototypical cubic κ_3 is twisted, spanning the affine 3-space $\mathrm{Sym}_3(\hat{L})^{\downarrow}$. So, the instancing transformation for any planar cubic segment will definitely fail to be injective; its four Bézier points will be coplanar. When the instancing transformation f fails to be injective in this way, the differential geometry in the object space gets affected. For example, all of the osculating planes of a planar cubic coincide, so we can't construct the point $f(\bar{a}\bar{b}\bar{c})$ geometrically by intersecting the osculating planes to F at the parameter values $\bar{a}$, $\bar{b}$, and $\bar{c}$. But the differential geometry of the prototype is not affected. We can still intersect the osculating planes to κ_3 at $\bar{a}^3$, $\bar{b}^3$, and $\bar{c}^3$ to find the 3-site $\bar{a}\bar{b}\bar{c}$ and then apply the instancing transformation f. The resulting point $f(\bar{a}\bar{b}\bar{c})$ is the blossom value $\tilde{F}(\bar{a}, \bar{b}, \bar{c})$, as we discuss next.

6.4 The relationship to blossoming

Let $F\colon L \to O$ be an n-ic, parametric curve in some affine object space O. The *polar form* or *blossom* of F is the unique symmetric, n-affine function $\tilde{F}\colon L^n \to O$ that agrees with F on the diagonal, that is, that satisfies

$$\tilde{F}(\underbrace{P,\ldots,P}_{n}) = F(P).$$

The term "polar form" points out the relationship to polarization in other contexts; but we are using the term "blossom" in this monograph, to alleviate overuse of the word "form". The blossom gives us an enlightening way to name the Bézier points of any segment of F, the k^{th} Bézier point of the segment $F([\bar{a} \mathinner{..} \bar{b}])$ being the blossom value

$$\tilde{F}(\underbrace{\bar{a},\ldots,\bar{a}}_{n-k},\underbrace{\bar{b},\ldots,\bar{b}}_{k}).$$

But the algebra of sites gives us an even better naming scheme for Bézier points. We realize the particular n-ic curve F as an affine transform of the prototype κ_n; that is, for some affine map $f\colon \mathrm{Sym}_n(\hat{L})^{\downarrow} \to O$, we have $F(P) = f(\kappa_n(P)) = f(P^n)$, for all points P in L. It immediately follows that the blossom of F is given by

$$\tilde{F}(P_1,\ldots,P_n) = f(P_1 \cdots P_n),$$

since that right-hand side is clearly symmetric, n-affine, and agrees with F on the diagonal. In particular, we can now write the k^{th} Bézier point of the segment $F([\bar{a} \mathinner{..} \bar{b}])$ simply as $f(\bar{a}^{\,n-k}\,\bar{b}^{\,k})$.

This is a significant notational improvement. By exploiting exponential notation, we can now write down the k^{th} Bézier point in running text, without requiring a displayed formula and horizontal braces.

Furthermore, that improved notation is just one of the rewards for an underlying conceptual advance: replacing concatenation with multiplication. When computing a blossom value $\tilde{F}(P_1,\ldots,P_n)$, we assemble the points P_1 through P_n by concatenating them into a sequence. Concatenation is automatically associative; but we must explicitly require the blossom $\tilde{F}$ to be symmetric in order to get commutativity.† Multiplication is better in every way: It is automatically both associative and commutative; it also distributes

†We could build in commutativity by using a multiset (a.k.a. bag or suite) as the input to the blossom, rather than a sequence. But multisets are unfamiliar, and they introduce their own notational challenges. For example, the domain space of the blossom would then be the set of all multisets of size n whose n elements are points on the parameter line L — a set for which there is no standard notation.

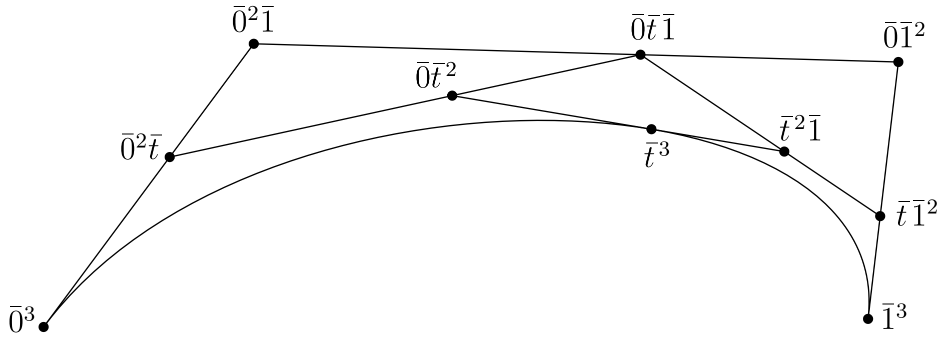


Figure 6.2: The de Casteljau Algorithm on the prototypical cubic κ_3

over addition; and the notations associated with it are more concise, to boot. Thus, moving from concatenation to multiplication is a big win.

Two things combine to make that win available to us: First, the curve κ_n is rich enough to serve as a prototype for all polynomial n-ics; and second, that prototype κ_n is defined, not in some arbitrary way, but by exploiting the multiplication in an algebra.

Why is κ_n rich enough to serve as a prototype? Because all of the higher-order, nonaffine stuff that has to happen as part of evaluating an n-ic curve F at an argument point P already happens as part of computing $\kappa_n(P)$. Once we know the n-site $\kappa_n(P)$, we can compute $F(P) = f(\kappa_n(P))$ by applying the instancing transformation f, which is an affine map. But that property of κ_n is shared by lots of other curves — indeed, by every n-ic polynomial curve that is twisted in n different dimensions, so as to have an affine span that is n-dimensional.

The second key point about the prototype κ_n is that we have defined it using the multiplication in the algebra of sites, setting $\kappa_n(P) := P^n$. The geometric structure of the de Casteljau Algorithm simply reflects the multiplicative structure of that algebra. Figure 6.2 shows the de Casteljau Algorithm working on the moment cubic κ_3.

We don't cover the details in this monograph, but a similar result holds in the rational case. Every parametric rational n-ic curve is a projective transform of the *n-ic rational normal curve* of algebraic geometry, which is the projective closure of the n-ic moment curve κ_n.

Exercise 6.4-1 Which cubic sites over the affine line are real-lineal? What does this question have to do with the twisted cubic κ_3 of perfect cubes?

Answer: The discriminant of the 3-site $s = r\,C^3 + x\,C^2\varphi + y\,C\varphi^2 + z\,\varphi^3$ is $18rxyz + x^2y^2 - 4ry^3 - 4x^3z - 27r^2z^2$. This discriminant is zero precisely on the ruled surface that is swept out by the tangent lines to the twisted cubic κ_3, that is, precisely for those sites s that are divisible by a perfect square. When the discriminant is positive, which it is on one side of that quartic ruled surface, the site s splits, over the reals, as the product of three distinct anchors. When the discriminant is negative, the site s has, as its factors, one real anchor and one pair of conjugate, complex anchors.

Exercise 6.4-2 What does the discriminant of a quartic site over the affine line tell us about whether or not that site is real-lineal?

Answer: The 4-site $r\,C^4 + w\,C^3\varphi + x\,C^2\varphi^2 + y\,C\varphi^3 + z\,\varphi^4$ has, as its discriminant, a homogeneous sextic polynomial in the coefficients (r, w, x, y, z) with sixteen terms. Readers who want the details can type this to Maple [8]:

```
discrim(r*t^4 + w*t^3 + x*t^2 + y*t + z, t);
```

The zero-set of this discriminant is the 3-fold in the 4-space $\mathrm{Sym}_4(\hat{L})^{\downarrow}$ whose 4-sites are divisible by a perfect square. Such 4-sites may have the form P^2QR, $P^2E\bar{E}$, or $E^2\bar{E}^2$, where P, Q, and R are real points on the line L, while E is a complex point on L and $\bar{E}$ is its conjugate. Thus, 4-sites with zero discriminant may have 4, 2, or 0 real anchors as factors. Note that the sites of the first two types are swept out by the osculating planes to the moment quartic κ_4, as P moves along L. A 4-site with negative discriminant has, as its factors, two distinct real anchors and one pair of conjugate, complex anchors. A 4-site with positive discriminant may have either four distinct real factors or else two pairs of conjugate, complex factors.

6.5 The Veronese surface

Now that we have some intuition for how the n^{th}-power map behaves on the line L, let's consider how the squaring map behaves on the plane A; that is, let's return to the case $n = 2$, but now with $d = 2$. Let $\sigma_2\colon A \to \mathrm{Sym}_2(\hat{A})^{\downarrow}$ be the map defined by $\sigma_2(P) := P^2$, for each point P on the plane A. The image $\sigma_2(A)$ is a curved surface (that is, a 2-fold) sitting in the affine 5-space

$$\begin{aligned}\mathrm{Sym}_2(\hat{A})^{\downarrow} &= \mathbf{R}_2[C, \varphi, \psi]^{\downarrow} \\ &= \{C^2 + b\,\varphi^2 + c\,\psi^2 + x\,\varphi\psi + y\,C\psi + z\,C\varphi \mid b, c, x, y, z \in \mathbf{R}\}.\end{aligned}$$

The projective completion of the image $\sigma_2(A)$ is called the *Veronese surface* in algebraic geometry. To allow for higher degrees, we'll call it the *Veronese surface of parametric degree* 2. (Warning: The parametric degree is different from the degree of the surface itself. Indeed, the Veronese surface $\sigma_2(A)$, as a variety in 5-space, actually has degree 4. More generally, the Veronese d-fold of parametric degree n, as a variety in projective space of dimension $\binom{n+d}{n} - 1$, turns out [31] to have degree n^d.)

Like the moment curve κ_n, the Veronese surface σ_2 is a prototype — a prototype for all parametric polynomial surfaces of degree at most 2. Let

$F\colon A \to O$ be any parametric surface in an affine object space O whose coordinates are given by polynomials of total degree at most 2 in u and v, the coordinates on A. Then, there exists a unique affine transformation $f\colon \operatorname{Sym}_2(\hat{A})^{\downarrow} \to O$ with $F(P) = f(\sigma_2(P)) = f(P^2)$. Note that the domain of the instancing transformation f here is 5-dimensional. If our surface instance is to sit in 3-space, as is typically the case in CAGD, then the instancing transformation f can't possible be injective.

Bézier points and blossoming for quadratic surfaces hold no surprises. The blossom of the quadratic surface F is given by $\tilde{F}(P,Q) = f(PQ)$, for all points P and Q in the plane A. Given a reference triangle $\triangle RST$ in A, the six 2-sites R^2, RS, RT, S^2, ST, and T^2 form an affine frame for the 5-space $\operatorname{Sym}_2(\hat{A})^{\downarrow}$, and we often specify an affine instancing transformation f by giving the images of these six frame points under f, those images being the Bézier points of the quadratic triangular patch $F(\triangle RST)$.

Which 2-sites in the affine 5-space $\operatorname{Sym}_2(\hat{A})^{\downarrow} = \mathbf{R}_2[C,\varphi,\psi]^{\downarrow}$ are lineal? That is, for which coefficients (b,c,x,y,z) do there exist coefficients (u_1,v_1) and (u_2,v_2), either real or complex, with

$$\begin{aligned} s &= C^2 + b\,\varphi^2 + c\,\psi^2 + x\,\varphi\psi + y\,C\psi + z\,C\varphi \\ &= (C + u_1\varphi + v_1\psi)(C + u_2\varphi + v_2\psi)? \end{aligned}$$

Since there are five parameters on the first line and only four on the second, it is clear that a typical 2-site s is not even complex-lineal. Instead, the five coefficients (b,c,x,y,z) must satisfy one constraint in order for the resulting 2-site s to have any hope of factoring. That constraint is encoded by a polynomial, which is again referred to as the *discriminant*. To write that discriminant more symmetrically, let's abandon the constraint of unit weight, replacing the term C^2 with aC^2. It turns out that the 2-site

$$\begin{array}{rccccc} s = & & & a\,C^2 & & \\ & & +\,z\,C\varphi & & +\,y\,C\psi & \\ & +\,b\,\varphi^2 & & +\,x\,\varphi\psi & & +\,c\,\psi^2 \end{array}$$

factors over the complexes as the product of two anchors just when

$$\Delta(a,b,c,x,y,z) := 4abc + xyz - ax^2 - by^2 - cz^2 = 0. \tag{6.5-1}$$

Polynomials that have no nontrivial factors, even over the complex numbers, are called *absolutely irreducible*, and we shall apply that term also to forms and sites. Thus, a quadratic site s over the plane A is either complex-lineal or absolutely irreducible, according as its discriminant Δ is zero or nonzero. If s is complex-lineal, it may or may not factor also over the reals.

Warning: The word "discriminant" is used whenever some condition can be tested by a single polynomial, regardless of what condition that might be.

In the case $d = 1$ of sites over a line (as for univariate polynomials), every n-site is complex-lineal, and the discriminant tests whether some two of those n factors coincide. In the case $d = 2$ of sites over a plane (as for bivariate polynomials), most n-sites are absolutely irreducible. For the particular case $n = d = 2$, there is a single polynomial that tests for absolute irreducibility: the discriminant polynomial Δ in Equation 6.5-1.

Exercise 6.5-2 Assuming that the coefficients (a, b, c, x, y, z) of the 2-site $s = a\,C^2 + z\,C\varphi + y\,C\psi + b\,\varphi^2 + x\,\varphi\psi + c\,\psi^2$ are real and that the discriminant $\Delta(a, b, c, x, y, z)$ is zero, when will s be real-lineal?

Hint: In order for s to factor over the reals, the three inequalities

$$x^2 - 4bc \geq 0$$
$$y^2 - 4ac \geq 0$$
$$z^2 - 4ab \geq 0$$

are clearly necessary, and it turns out that they are also sufficient.

A fine point: Given that $\Delta = 0$, any two of those three inequalities are almost enough to imply the third. For example, the last two imply the first except in the degenerate case $a = y = z = 0$, where $\Delta = 0$ holds automatically and the last two inequalities hold automatically as equalities.

Exercise 6.5-3 Which 3-sites over the plane A are complex-lineal?

Answer: The space $\mathrm{Sym}_3(\hat{A})^{\downarrow}$ of unit-weight 3-sites is 9-dimensional, while the three factors of a lineal 3-site have only 6 degrees of freedom among them. Hence, there are $9 - 6 = 3$ dimension's worth of algebraic constraints that must hold, among the coefficients of a 3-site, in order for that site to be complex-lineal. Unfortunately, while one algebraic constraint can always be encoded by a single polynomial, it typically takes more than k polynomials to encode k dimension's worth of constraints, the extra polynomials being required to eliminate spurious roots. To say the same thing in more modern language, most varieties of codimension $k > 1$ are not complete intersections. In this exercise, requiring a 3-site over the plane A to be complex-lineal involves 3 dimension's worth of constraints; but the most efficient encoding of those constraints that I know of uses 45 polynomials, each of which is a quartic in the ten coefficients of a 3-site of arbitrary weight [45].

6.6 Degen's analysis of quadratic surfaces

While this monograph proposes a new framework for research in CAGD, most of the results from CAGD that we discuss — such as Bézier points —

are quite basic. In contrast, this section comes closer to current research. We here use the multiplication in the algebra of sites to illuminate Wendelin Degen's 1994 analysis [15] of quadratic surfaces.

In that paper, Degen analyzes the various types of surfaces that can occur as parametric rational quadratics (a.k.a. quadratic Bézier triangles) in 3-space. The prototype for such a surface is the Veronese surface $\sigma_2(A) = \{P^2 \mid P \in A\}$, consisting of the squares of the points in the plane A. That surface of squares, we recall, is a quartic surface, sitting in the 5-space $\mathrm{Sym}_2(\hat{A})^{\downarrow}$. The instancing transformation for a quadratic Bézier triangle projects the surface $\sigma_2(A)$, sitting in this 5-space, down into an object 3-space; so the dimension goes down by 2. A projection that reduces the dimension by 1 projects along those lines that pass through a certain *center point*. To reduce the dimension by 2, we must project, instead, along those planes that pass through a certain *center line*, a line λ in 5-space. The heart of Degen's analysis considers the various geometric relationships that the center line λ can have, both with the Veronese surface $\sigma_2(A)$ itself and with the cubic 4-fold of complex-lineal sites, that is, the 4-fold $\Delta = 0$ characterized by the vanishing of the discriminant Δ in Equation 6.5-1.

Degen's analysis is correct, complete, and pretty; but Degen missed some opportunities because he worked purely geometrically with the 5-space in which the Veronese surface sits. In fact, that 5-space $\mathrm{Sym}_2(\hat{A})^{\downarrow}$ lies in the algebra of sites $\mathrm{Sym}(A)$. We here exploit the multiplication of that algebra to give us a new, more algebraic perspective on Degen's results.

By the way, Degen used this same *method of projection* to tackle other problems [18], in each case studying the geometric relationships between a Veronese prototype and the central flat of the instancing transformation that projects that prototype down into some object space. Whenever Veronese prototypes are exploited in this way, the multiplication of the algebra of sites may be a helpful algebraic adjunct to more geometric reasoning.

Warning: This section is rather technical; indeed, one of its goals is to show how the algebra of sites performs when put to a significant test. Some readers may prefer to skip on to Section 6.7.

We are trying not to rely on projective geometry in this monograph. So the only quadratic surfaces that we can handle are the polynomial ones, which are produced by projecting the Veronese surface $\sigma_2(A)$ down from 5-space into 3-space along a family of parallel planes. Speaking projectively, the center line λ of such a projection is a line at infinity: the line at infinity where all of those parallel planes intersect. In discussing Degen's analysis, however, we don't want to restrict λ to be a line at infinity. Fortunately, we don't need projective geometry in order to discuss how an arbitrary line λ in 5-space relates to the Veronese surface $\sigma_2(A)$ and to the cubic 4-fold $\Delta = 0$. We would need projective geometry to perform a projection from a line λ that wasn't at infinity; but we won't discuss that projection.

6.6.1 The planes in the 4-fold $\Delta = 0$

There turn out to be two families of planes that lie entirely within the cubic hypersurface $\Delta = 0$, and those planes are important in analyzing the various degenerate ways in which the center line λ can interact with the hypersurface $\Delta = 0$. What such planes can we think of?

First, let L denote any line in the affine plane A. The image of L under the Veronese map σ_2 will be a parabola, lying in some plane I_L — a plane that looks, in fact, just like Figure 6.1. All of the 2-sites in the plane I_L factor over the complexes. The 2-sites on the parabola itself are the squares of the points on L; the 2-sites outside the parabola factor as PQ, where P and Q are distinct, real points on L; and the 2-sites inside the parabola factor as the product of a pair of conjugate complex points on L. Thus, the entire plane I_L lies inside the hypersurface $\Delta = 0$ of complex-lineal sites.

Second, let P be some point in A, and consider the plane $T_P := \{PQ \mid Q \in A\}$. Every site in the plane T_P factors over the reals, so T_P lies inside the hypersurface $\Delta = 0$. In fact, the plane T_P is the tangent plane to the Veronese surface σ_2 at P^2, as we shall see in Section 6.7.

So we have two families of planes, the first indexed by lines in A and the second by points in A. How do planes from those two families intersect?

Consider first two planes I_L and I_M from the first family. If the lines L and M coincide, then the image planes I_L and I_M also coincide, obviously. If the lines L and M are distinct, they typically intersect in a unique point P. (In affine geometry, L and M might be parallel; but let's not worry about that case, since projective geometry stands ready to deal with all of the special cases caused by parallelism.) The 2-sites that lie in I_L factor as the product of two points — possibly conjugate complex — along L, and the analogous claim holds for I_M. Thus, the 2-site P^2 is the unique site that belongs to both I_L and I_M. So two distinct planes from the first family intersect in a unique site, and that site lies on the Veronese surface.

It's a similar story for two planes T_P and T_Q from the second family. If $P = Q$, then the tangent planes T_P and T_Q coincide. Otherwise, the 2-site PQ is the unique site that belongs to both T_P and T_Q. But note that the site PQ lies off of the Veronese surface, rather than on it; in that detail, the second family differs from the first.

What about a plane I_L from the first family and a plane T_P from the second? If P lies on L, then the planes I_L and T_P intersect in the entire line of 2-sites $\{PQ \mid Q \in L\}$. But, if P does not lie on L, then the planes I_L and T_P are skew.

The cubic 4-fold $\Delta = 0$ thus has two 2-parameter families of planes that lie inside it, where two planes drawn from the same family intersect in a flat of even dimension, while two planes drawn from different families intersect in a flat of odd dimension. That geometry is hardly surprising, since the situation for a nonsingular quadric 4-fold in 5-space is the same, except that each of the two families of planes lying inside a nonsingular quadric is a 3-parameter family. The moral of the story is not that the geometry is surprising, but rather that we can uncover that geometry easily by exploiting the multiplication in the algebra of sites.

One place where these two families of planes arise is in Degen's handling of degenerate cases. Recall that Degen classifies the different ways in which the center line λ of a proposed projection can intersect the cubic hypersurface $\Delta = 0$. The most degenerate thing that can happen is for λ to lie entirely within the hypersurface $\Delta = 0$. Degen proves [16] that any such line λ lies, in fact, either entirely inside the image plane I_L, for a unique line L in A, or entirely inside the tangent plane T_P, for a unique point P in A. (Those two options can happen simultaneously, if λ is the line where I_L intersects T_P, for some point P on some line L.)

Exercise 6.6-1 Given two distinct points P and Q in A, on how many planes of each family does the site PQ lie? What about the site P^2?

Answer: The site PQ lies on one plane of the first family, the plane I_L where L is the line joining P to Q. It lies on two planes of the second family, the planes T_P and T_Q. As for the site P^2, it lies on a one-parameter family of planes of the first family: on the plane I_L, for each line L passing through P. It lies on just one plane of the second family, the plane T_P.

6.6.2 The typical complete quadrilateral

Let's now turn from the most degenerate things that can happen to the thing that happens typically: Our proposed center line λ typically intersects the cubic hypersurface $\Delta = 0$ at three distinct sites. Let's further assume that all three of those sites of intersection are real; this is the case that Degen [17] refers to as (**Aa**). And finally, for simplicity, let's assume that each of the three sites of intersection factors, not only over the complexes, but actually over the reals. So, for some three pairs of points $\{P_1, Q_1\}$, $\{P_2, Q_2\}$, and $\{P_3, Q_3\}$ in A, our center line λ intersects the hypersurface $\Delta = 0$ precisely at the three sites P_1Q_1, P_2Q_2, and P_3Q_3. What geometric relationships hold among the P's and Q's?

Given two of the three pairs, say $\{P_1, Q_1\}$ and $\{P_2, Q_2\}$, the third pair $\{P_3, Q_3\}$ is determined, since the line joining the site P_1Q_1 to P_2Q_2 intersects the hypersurface $\Delta = 0$ at those two sites and at one more site, which must be P_3Q_3. But we can also locate P_3 and Q_3 by carrying out a geometric construction in the plane A. To do so, note that the four points P_1, Q_1, P_2, and Q_2 are coplanar in A, so they must be linearly dependent; that is, there must exist real numbers a_1, b_1, $-a_2$, and b_2 with

$$a_1P_1 + b_1Q_1 - a_2P_2 + b_2Q_2 = 0.$$

(We write $-a_2$ rather than a_2 in order to make what follows symmetric under cyclic permutations of the subscripts 1, 2, and 3.) To save writing, let's introduce p_1 as an abbreviation for the anchor $p_1 := a_1P_1$, and similarly for $q_1 := b_1Q_1$, $p_2 := a_2P_2$, and $q_2 := b_2Q_2$. In terms of those anchors, we have the dependence

$$p_1 + q_1 - p_2 + q_2 = 0. \tag{6.6-2}$$

Using that dependence, we find that

$$\begin{aligned} p_1q_1 + p_2q_2 &= p_1q_1 + p_2q_2 + (p_1 + q_1 - p_2 + q_2)q_2 \\ &= p_1q_1 + p_1q_2 + q_1q_2 + q_2^2 \\ &= (p_1 + q_2)(q_1 + q_2). \end{aligned}$$

Thus, the line joining P_1Q_1 to P_2Q_2 passes through the site $(p_1+q_2)(q_1+q_2) = (a_1P_1 + b_2Q_2)(b_1Q_1 + b_2Q_2)$; so the two anchors in that product must be scaled versions of the points P_3 and Q_3, in some order. Choosing an order and undoing the scaling, we set

$$P_3 := \frac{a_1P_1 + b_2Q_2}{a_1 + b_2} \qquad \text{and} \qquad Q_3 := \frac{b_1Q_1 + b_2Q_2}{b_1 + b_2}.$$

(If either of those denominators is zero, the corresponding point is at infinity; let's not worry about that possibility.) If we now set $a_3 := a_1 + b_2$, $b_3 := -(b_1 + b_2)$, $p_3 := a_3P_3$, and $q_3 := b_3Q_3$, we have established the following four equations, the first three of which are cyclically symmetric:

$$\begin{aligned} q_1 - p_2 + p_3 &= 0 \\ p_1 + q_2 - p_3 &= 0 \\ -p_1 + p_2 + q_3 &= 0 \\ q_1 + q_2 + q_3 &= 0 \end{aligned}$$

The last two equations reveal that Q_3 is the point where the line $P_1 \vee P_2$ intersects the line $Q_1 \vee Q_2$, while the first two reveal that P_3 is given by $P_3 = (P_2 \vee Q_1) \wedge (P_1 \vee Q_2)$. (Note that we are writing $S \vee T$ to denote the line joining S to T, to avoid confusion with the 2-site ST.) To say the same thing more symmetrically, of the eight possible combinations of one point from each pair, the four collinear triples are the ones with an even number of P's and an odd number of Q's. So the three pairs of points $\{P_1, Q_1\}$, $\{P_2, Q_2\}$, and $\{P_3, Q_3\}$ are the pairs of opposite vertices of a *complete quadrilateral*, as shown in Figure 6.3. Thus, a center line λ in 5-space typically corresponds to a complete quadrilateral in the plane A in the sense that the three intersections of λ with the hypersurface $\Delta = 0$ are the products of that quadrilateral's three opposite pairs of vertices. That's neat.

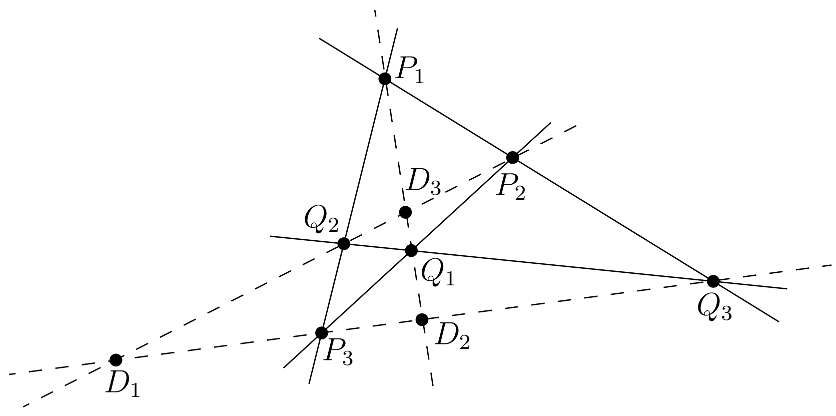


Figure 6.3: The complete quadrilateral from a typical center line λ

Geometry remark: The complete quadrilateral captures the most nondegenerate way in which three lineal 2-sites P_1Q_1, P_2Q_2, and P_3Q_3 over a plane A can be collinear in the space $\mathrm{Sym}_2(\hat{A})^{\downarrow}$. For other values of n, is there an analogous geometric configuration that captures the most nondegenerate way in which $n+1$ lineal n-sites over an n-space A can span a flat in the space $\mathrm{Sym}_n(\hat{A})^{\downarrow}$ whose dimension is less than n? When $n = 3$, the answer is yes. Ramshaw and Saxe [44] analyze the solution for $n = 3$, a configuration that captures the coplanarity of the four 3-sites $P_1Q_1R_1$, $P_2Q_2R_2$, $P_3Q_3R_3$, and $P_4Q_4R_4$ by constraining those twelve points in 3-space to be incident to two lines and thirteen planes, in a pattern described by the budget matroid $B_{2,1,1}$. Since complete quadrilaterals, which are the solution for $n = 2$, are representations of the budget matroid $B_{2,1}$, this suggests a pattern. Unfortunately, it seems that the representations of the budget matroid $B_{2,1,1,1}$ cannot provide an analogous solution when $n = 4$, since it seems that there are only 30 dimensions' worth of such representations, rather than the 34 dimensions' worth that would be required.

Warning: Ramshaw and Saxe [44] don't exploit the multiplication in the algebra of sites; hence, they don't prove that the four products $(P_iQ_iR_i)_{1\le i\le 4}$ are coplanar. (Indeed, I was only beginning to realize, back then, that it makes sense to multiply points.) But they do prove a certain property of the slopes of the twelve planes that result when those twelve points are projected from an arbitrary line in 3-space, and that slope property turns out to be equivalent to the coplanarity of the four products.

The dashed lines $P_1 \vee Q_1$, $P_2 \vee Q_2$, and $P_3 \vee R_3$ in Figure 6.3 are called the *diagonals* of the complete quadrilateral, and they have a role to play in Degen's analysis as well. Let D_i, for i from 1 to 3, be the vertex of the dashed triangle that is opposite the side $P_i \vee Q_i$, as shown in Figure 6.3. Equation 6.6-2 tells us that $p_1 + q_1 = p_2 - q_2$. The point D_3 is the intersection of the diagonals $P_1 \vee Q_1$ and $P_2 \vee Q_2$, so D_3 must be a scalar multiple of the

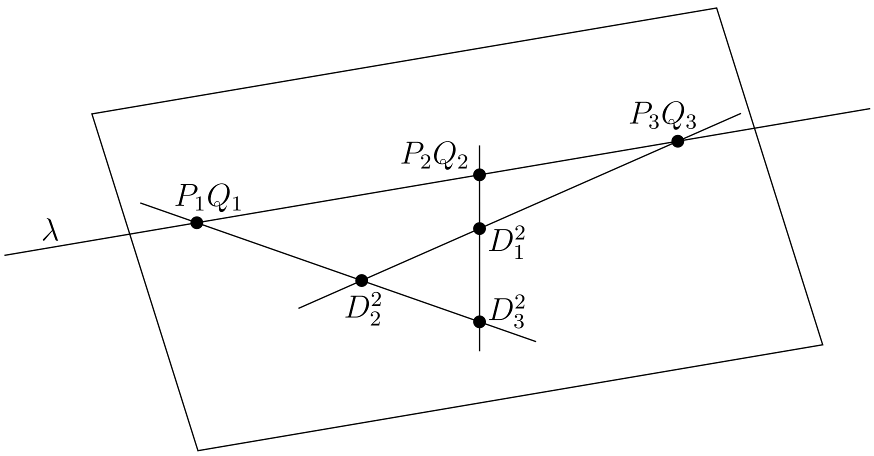


Figure 6.4: The plane projector that corresponds to the triple point

anchor $d_3 := p_1 + q_1 = p_2 - q_2$. Introducing scalar multiples of D_1 and D_2 in a similar way, we have

$$\begin{aligned} d_1 &:= p_2 + q_2 = p_3 - q_3 \\ d_2 &:= p_3 + q_3 = p_1 - q_1 \\ d_3 &:= p_1 + q_1 = p_2 - q_2. \end{aligned}$$

These formulas reveal some interesting collinearities in the 5-space $\mathrm{Sym}_2(\hat{A})^{\downarrow}$, as shown in Figure 6.4. The simple identity

$$(p_1 - q_1)^2 + 4p_1q_1 = (p_1 + q_1)^2$$

shows that the 2-sites D_2^2, P_1Q_1, and D_3^2 are collinear; and the same holds with the subscripts cyclically permuted. Any two of those three collinearities suffice to show that the plane in 5-space spanned by the three perfect squares D_1^2, D_2^2, and D_3^2 contains our entire proposed center line λ, the line through the collinear 2-sites P_1Q_1, P_2Q_2, and P_3Q_3. Thus, when Degen takes the planes in 5-space through the line λ as the points of his object 3-space, the particular plane shown in Figure 6.4 will belong to the resulting surface instance — the projected image of the Veronese surface $\sigma_2(A)$ — for three different reasons. In fact, that point in 3-space is the *triple point* of the resulting quartic Steiner surface.

Recall what happens when we project the twisted cubic curve $\kappa_3(L)$ in 3-space, from some center point, to get a rational cubic curve in the plane. For a typical choice of the center point C, there is a unique line through C that intersects $\kappa_3(L)$ twice. That is, most points lie on a unique chord of the twisted cubic, that chord giving rise to a double point on the projected, planar cubic. If we define a 2-*chord* of the Veronese surface $\sigma_2(A)$ to be the plane spanned by the squares of three noncollinear points in the plane A, Degen's analysis shows that most lines λ in 5-space lie on a unique 2-chord of the Veronese surface, that 2-chord giving rise to a triple point on the projected surface in 3-space.

Exercise 6.6-3 The triple point of the Steiner surface in case (**Aa**) is the intersection of three noncoplanar double lines, that is, lines where the Steiner surface intersects itself. A point on one of those double lines corresponds to a plane projector through λ that contains both S^2 and T^2, for some two distinct points S and T in the plane A. Where do the points S and T in such a pair lie in Figure 6.3?

Answer: For some i between 1 and 3, the points S and T are harmonic conjugates of P_i and Q_i, along the dashed line $P_i \vee Q_i$. At the triple point itself, any two of the three points D_1, D_2, and D_3 can play the roles of S and T; for example, D_1 and D_2 are harmonic conjugates of P_3 and Q_3.

6.6.3 Extending Degen's analysis to cubics

Extending Degen's analysis from quadratics to cubics would be a challenging endeavor in which the algebra of sites might well prove useful.

The prototype for a cubic Bézier triangle is the Veronese surface of parametric degree 3, the surface $\sigma_3(A) = \{P^3 \mid P \in A\}$. This surface has degree $3^2 = 9$ and sits in the 9-space $\mathrm{Sym}_3(\hat{A})^{\downarrow}$. To end up with a surface in an object 3-space, the instancing transformation must reduce the dimension by 6; thus, it will project from some central 5-flat H, sitting in 9-space, down into the 3-space of all 6-flats that include H. The character of the Bézier triangle that results from this projection will presumably depend upon how the 5-flat H sits in 9-space, in relation to such structures as the following:

- the surface $\sigma_3(A)$ itself, the 2-fold of 3-sites that are perfect cubes;
- the 4-fold of 3-sites that are divisible by a perfect square;
- the 6-fold of complex-lineal 3-sites, the sites that are divisible by some three anchors, typically all distinct;
- and the 7-fold of reducible 3-sites, the sites that have some anchor as a factor, but where the quadratic cofactor may be absolutely irreducible.

Indeed, some of these dependencies are straightforward; for example, the degree of the projected Bézier triangle will be 9 minus the number of points where the central 5-flat H intersects the Veronese surface $\sigma_3(A)$ itself. But other dependencies will be more subtle. We leave those questions open, returning in a moment to the main thread of this monograph.

By the way, it will follow from Proposition 6.7-2 that the second structure in the list above, the 4-fold, is the union of the tangent planes to the Veronese surface σ_3; that is, for a point P in the plane A and a 3-site s over A, the site s is divisible by P^2 just when s lies in the tangent plane to the surface σ_3 at P^3. In a similar way, the fourth variety in the list above, the 7-fold, is the union, for all points P in the plane A, of the 5-flat that osculates σ_3 to first order at P^3. Note that each such osculating flat is 5-dimensional, since it is spanned by the two first-derivatives $\partial P^3/\partial u$ and $\partial P^3/\partial v$ and the three second-derivatives $\partial^2 P^3/\partial u^2$, $\partial^2 P^3/\partial u \partial v$, and $\partial^2 P^3/\partial v^2$.

Degen himself extended his work in a different direction by classifying the types of surfaces that can arise as tensor-product surfaces of bidegree $(2;1)$ in 3-space — that is, by classifying the Bézier rectangles that are quadratic in one parameter direction and affine in the other [18]. We discuss tensor-product surfaces in Section 6.8.

6.7 Polynomial d-folds of degree at most n

The theory of Veronese prototypes generalizes to n-ic d-folds, for any bound n on the total degree and any parametric dimension d. Let A be our affine domain space, now of dimension d. The linearization $\hat{A}$ has dimension $d+1$, so the algebras $\mathrm{Sym}(\hat{A})$ and $\mathrm{Sym}(\hat{A}^*)$ of sites over A and forms on A are essentially polynomial algebras with $d+1$ variables. An n-ic polynomial parametric d-fold, sitting in some affine object space O, is a map $F\colon A \to O$ that can be given by polynomials of total degree at most n in the coordinates on A. There is a prototype for all such d-folds F: the n^{th}-power map $\theta_{d,n}\colon A \to \mathrm{Sym}_n(\hat{A})^{\downarrow}$ given by $\theta_{d,n}(P) := P^n$, for all points P in the d-space A. In particular, for any n-ic parametric d-fold $F\colon A \to O$, there is a unique affine transformation $f\colon \mathrm{Sym}_n(\hat{A})^{\downarrow} \to O$ with $F(P) = f(\theta_{d,n}(P)) = f(P^n)$, for all points P in A. We'll refer to $\theta_{d,n}$ as the *Veronese d-fold of parametric degree n*; the map $\theta_{d,n}$ is also known as the *n-uple embedding of d-space* [33]. The moment curve κ_n is the Veronese 1-fold $\kappa_n = \theta_{1,n}$, while the Veronese surface σ_n is the Veronese 2-fold $\sigma_n = \theta_{2,n}$.

What is a Bézier point in this context? We choose some d-simplex of reference, say $[R_0, \ldots, R_d]$ in A. The points $(R_0, \ldots, R_d)$ form an affine frame for the affine space A and also form a basis for its linearization $\hat{A}$. Hence, we can view the algebra of sites $\mathrm{Sym}(\hat{A})$ as the polynomial algebra $\mathbf{R}[R_0, \ldots, R_d]$. Consider the n-sites $R_0^{i_0} \ldots R_d^{i_d}$, where i_0 through i_d are any nonnegative integers with $i_0 + \cdots + i_d = n$. There are $\binom{n+d}{n}$ such sites, and they form an affine frame for the affine space $\mathrm{Sym}_n(\hat{A})^{\downarrow}$ of all unit-weight n-sites over A. One convenient way to specify which affine instancing transformation $f\colon \mathrm{Sym}_n(\hat{A})^{\downarrow} \to O$ we have in mind is to specify the images of the sites in that frame, each image $f(R_0^{i_0} \ldots R_d^{i_d})$ being a Bézier point of the resulting d-fold.

Exercise 6.7-1 Consider (real) n-sites of unit weight on an affine d-space. How many dimension's worth of such sites are there altogether? How many are complex-lineal? Are real-lineal? Are perfect n^{th} powers?

Answers: $\binom{n+d}{n} - 1$, nd, nd, and d.

What about the flat that osculates the Veronese d-fold $\theta_{d,n}$ to k^{th} order at the n-site $\theta_{d,n}(P) = P^n$, for some $k \leq n$? The only surprising thing about that osculating flat is its dimension, which is $\binom{k+d}{k} - 1$. Let $(C, \varphi_1, \ldots, \varphi_d)$ be some Cartesian reference frame for the space A, and let $(w, u_1, \ldots, u_d)$ be the dual basis for the linear space $\hat{A}^*$ of coanchors. If $F\colon A \to O$ is any parametric d-fold, the flat that osculates F to 0^{th} order at P is the point $F(P)$. For the flat that osculates F to 1^{st} order at P, we expand to include the d vectors $\partial F/\partial u_1(P)$ through $\partial F/\partial u_d(P)$. For osculation to 2^{nd} order, we include $\binom{d+1}{2}$ second-order partials, either pure or mixed: $\partial^2 F/\partial u_1^2(P)$, $\partial^2 F/\partial u_1 \partial u_2(P)$, and so on. Osculation to 3^{rd} order adds in $\binom{d+2}{3}$ third-order partials. For osculation to k^{th} order, we have a total of

$$\binom{d}{1} + \binom{d+1}{2} + \cdots + \binom{d+k-1}{k} = \binom{d+k}{k} - 1$$

vectors. In the particular case where $F = \theta_{d,n}$ is the Veronese d-fold of some degree $n \geq k$, all of these vectors will be linearly independent, since $\theta_{d,n}$ is a prototype for any n-ic d-fold.

Proposition 6.7-2 *Let A be an affine d-space, let $\theta_{d,n}\colon A \to \operatorname{Sym}_n(\hat{A})^{\downarrow}$ be the Veronese d-fold of degree n, and let P be a point in A. The flat that osculates $\theta_{d,n}$ to k^{th} order at $\theta_{d,n}(P) = P^n$ is $P^{n-k}\operatorname{Sym}_k(\hat{A})^{\downarrow}$, the flat of dimension $\binom{k+d}{k} - 1$ that consists of all unit-weight multiples of P^{n-k}.*

Proof The only subtlety, in comparison with the proof of the case $d = 1$ in Proposition 6.3-1, is that we must consider approaching the point P in some arbitrary way, not necessarily along a straight line. Let $\mathbf{b}$ denote some vector in $\mathbf{R}^d$ and let $\|\mathbf{b}\|$ be the norm of $\mathbf{b}$ in some fixed norm for $\mathbf{R}^d$ — it doesn't matter which. We analyze the n^{th} power $(P + \mathbf{b} \cdot \varphi)^n$ as $\|\mathbf{b}\|$ tends to zero, where $\mathbf{b} \cdot \varphi$ denotes the vector $\mathbf{b} \cdot \varphi := b_1\varphi_1 + \cdots + b_d\varphi_d$. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_d)$ denote a multi-index of nonnegative integers with $|\alpha| = n$, and let $\alpha_+ := (\alpha_1, \ldots, \alpha_d)$ denote the dehomogenized version of α, with α_0 removed. By the Multinomial Theorem, we have

$$\begin{aligned}
(P + \mathbf{b} \cdot \varphi)^n &= \sum_{|\alpha|=n} \binom{n}{\alpha} P^{\alpha_0}\, \mathbf{b}^{\alpha_+} \varphi^{\alpha_+} \\
&= \sum_{\substack{|\alpha|=n \\ \alpha_0 \geq n-k}} \binom{n}{\alpha} P^{\alpha_0}\, \mathbf{b}^{\alpha_+} \varphi^{\alpha_+} + O(\|\mathbf{b}\|^{k+1}),
\end{aligned}$$

as $\|\mathbf{b}\|$ goes to zero. The sum here is a multiple of P^{n-k} and also has weight 1, since all terms have weight 0 except for P^n. Thus, the flat $P^{n-k}\operatorname{Sym}_k(\hat{A})^{\downarrow}$ osculates the Veronese d-fold $\theta_{d,n}$ to k^{th} order at P^n. □

6.8 Tensor-product surfaces

Despite the generality of the preceding section, there is yet another case to consider. So far, the degree bounds that we have been imposing are bounds on the total degree in all of the variables. Another option is to impose separate bounds on the degrees in disjoint sets of variables. In particular, the most common way to define a polynomial surface in CAGD is to impose separate bounds on the degrees of its defining polynomials in the two variables u and v. The resulting surfaces are the *tensor-product surfaces*, which can be thought of as curves of curves.

For tensor-product surfaces, we decompose the parameter plane A as the product of two lines, say $A = L_1 \times L_2$, and we linearize each of those lines separately. Suppose that we choose C_1 and φ_1 to be a center point and a unit vector for the line L_1, while C_2 and φ_2 are the same for L_2. Linearizing L_1 gives us the linear 2-space $\hat{L}_1$ of anchors over L_1, where each such anchor p_1 can be written uniquely as a linear combination $p_1 = w_1(p_1)C_1 + u_1(p_1)\varphi_1$. The coanchors (w_1, u_1) here are the basis for $\hat{L}_1^*$ that is dual to the basis (C_1, φ_1) for $\hat{L}_1$. All the same goes for L_2.

Both forms on $L_1 \times L_2$ and sites over $L_1 \times L_2$ have separate degrees n_1 and n_2 in the two parameters L_1 and L_2, the pair $(n_1; n_2)$ being called the *bidegree* of the form or site. Table 6.1 gives the formulas by which we shall denote the spaces of forms and sites of bidegree $(n_1; n_2)$ when characterized abstractly or when constructed by one of our three concrete constructions. On the first line, we are abstractly characterizing the algebrization of the linear space $\hat{L}_1 \oplus \hat{L}_2$ or $\hat{L}_1^* \oplus \hat{L}_2^*$ using a universal mapping condition, as discussed in Chapter 9. On the second line, fixing our chosen bases, an $(n_1; n_2)$-form on $L_1 \times L_2$ is a polynomial in the four variables w_1, u_1, w_2, and u_2 that is homogeneous of degree n_1 in w_1 and u_1 and separately homogeneous of degree n_2 in w_2 and u_2. That space of polynomials is most simply written $\mathbf{R}_{n_1;n_2}[w_1, u_1; w_2, u_2]$; but people who understand the tensor-product construction will see that it can equally well be written $\mathbf{R}_{n_1}[w_1, u_1] \otimes \mathbf{R}_{n_2}[w_2, u_2]$, and that is how the term "tensor-product surface" arose. An $(n_1; n_2)$-site over $L_1 \times L_2$ is analogous, but with the anchor variables C_1, φ_1, C_2, and φ_2. Moving to the third line, we can allow anchors or coanchors that are linearly dependent into our polynomials, as long as we realize that any given $(n_1; n_2)$-site or $(n_1; n_2)$-form will then have multiple, equivalent names, so we must mod out by an equivalence relation. The fourth line exploits duality to interpret each of those equivalence classes as a recipe for a real-valued function. We view an $(n_1; n_2)$-form as defining

space of $(n_1; n_2)$-sites	space of $(n_1; n_2)$-forms
$\mathrm{Sym}_{n_1;n_2}(\hat{L}_1 \oplus \hat{L}_2)$ $= \mathrm{Sym}_{n_1}(\hat{L}_1) \otimes \mathrm{Sym}_{n_2}(\hat{L}_2)$	$\mathrm{Sym}_{n_1;n_2}(\hat{L}_1^* \oplus \hat{L}_2^*)$ $= \mathrm{Sym}_{n_1}(\hat{L}_1^*) \otimes \mathrm{Sym}_{n_2}(\hat{L}_2^*)$
$\mathbf{R}_{n_1;n_2}[C_1, \varphi_1; C_2, \varphi_2]$ $= \mathbf{R}_{n_1}[C_1, \varphi_1] \otimes \mathbf{R}_{n_2}[C_2, \varphi_2]$	$\mathbf{R}_{n_1;n_2}[w_1, u_1; w_2, u_2]$ $= \mathbf{R}_{n_1}[w_1, u_1] \otimes \mathbf{R}_{n_2}[w_2, u_2]$
$\mathbf{R}_{n_1;n_2}[\hat{L}_1; \hat{L}_2]/ \approx_{(\hat{L}_1;\hat{L}_2)}$ $(\mathbf{R}_{n_1}[\hat{L}_1]/ \approx_{\hat{L}_1}) \otimes (\mathbf{R}_{n_2}[\hat{L}_2]/ \approx_{\hat{L}_2})$	$\mathbf{R}_{n_1;n_2}[\hat{L}_1^*; \hat{L}_2^*]/ \approx_{(\hat{L}_1^*;\hat{L}_2^*)}$ $(\mathbf{R}_{n_1}[\hat{L}_1^*]/ \approx_{\hat{L}_1^*}) \otimes (\mathbf{R}_{n_2}[\hat{L}_2^*]/ \approx_{\hat{L}_2^*})$
$\mathrm{Bipoly}_{n_1;n_2}(\hat{L}_1^* \times \hat{L}_2^*, \mathbf{R})$ $= \mathrm{Poly}_{n_1}(\hat{L}_1^*, \mathbf{R}) \otimes \mathrm{Poly}_{n_2}(\hat{L}_2^*, \mathbf{R})$	$\mathrm{Bipoly}_{n_1;n_2}(\hat{L}_1 \times \hat{L}_2, \mathbf{R})$ $= \mathrm{Poly}_{n_1}(\hat{L}_1, \mathbf{R}) \otimes \mathrm{Poly}_{n_2}(\hat{L}_2, \mathbf{R})$

Table 6.1: Formulas for the space of sites or forms of bidegree $(n_1; n_2)$

a real-valued function of bidegree $(n_1; n_2)$ on $\hat{L}_1 \times \hat{L}_2$, while an $(n_1; n_2)$-site defines a function on $\hat{L}_1^* \times \hat{L}_2^*$. But keep in mind that real-valued functions on coanchors don't have the obvious applications in CAGD that real-valued functions on anchors have.

Given any site s over $L_1 \times L_2$, we define the *weight* of s to be the real number $s(w_1, w_2)$ that results from evaluating s at the weight coanchors w_1 and w_2 of L_1 and L_2. That is, given any expression for s as a polynomial whose variables are anchors over L_1 or L_2, we replace each anchor p_1 on L_1 by its weight $w_1(p_1)$, we replace each anchor p_2 on L_2 by its weight $w_2(p_2)$, and we then simplify to get the weight $s(w_1, w_2)$. Going back to the second line in Table 6.1, if a site s of bidegree $(n_1; n_2)$ has been represented as a polynomial in $\mathbf{R}_{n_1;n_2}[C_1, \varphi_1; C_2, \varphi_2]$, then its weight is simply the coefficient of the term $C_1^{m_1} C_2^{m_2}$.

Since L_1 is an affine line, it is convenient to name the points on L_1 using real numbers. But the same holds for L_2, and we don't want to get the two lines confused; hence, we shall use two different accents. Let's denote by $\grave{t}$ the point $\grave{t} := C_1 + t\varphi_1$ with coordinate t on the line L_1, while $\acute{t} := C_2 + t\varphi_2$ is the point with that same coordinate on L_2. (I suggest reading the formulas $\grave{t}$ and $\acute{t}$ as "t in" and "t out".) For example, the formula $\grave{0}\grave{1}\acute{2}\acute{3}\acute{4}$ denotes a site over $L_1 \times L_2$ of bidegree $(2; 3)$, and the formula $\grave{0}^2\acute{0}^3$ denotes another such. In fact, we have $\grave{0}^2\acute{0}^3 = C_1^2 C_2^3$. When an $(n_1; n_2)$-site over $L_1 \times L_2$ is *real-lineal*, that is, splits as the product of n_1 anchors over L_1 and n_2 anchors on L_2, we'll typically write it with its L_1 factors to the left of its L_2 factors, by convention. But we could equally well write the factors in any order. Like the algebra of sites over A, the algebra of sites over $L_1 \times L_2$ is commutative, so $\grave{x}\acute{y} = \acute{y}\grave{x}$, for any real numbers x and y.

If we didn't distinguish between points on L_1 and points on L_2 using slanted accents, we would have to use some other technique to keep track of which points lay on which lines. For example, some authors would denote the $(2;3)$-site $\grave{0}\grave{1}\acute{2}\acute{3}\acute{4}$ as $\bar{0}\bar{1} \otimes \bar{2}\bar{3}\bar{4}$, where the points to the left of the symbol "$\otimes$" are presumed to lie on L_1, while those to the right lie on L_2. With this notation, the sites $\bar{t} \otimes 1$ and $1 \otimes \bar{t}$ are distinct, the former being the point $\grave{t}$ on L_1, while the latter is the point $\acute{t}$ on L_2. But don't be confused by this notation into thinking that the multiplication on sites somehow fails to be commutative. No matter how we write things, we still have $\grave{x}\acute{y} = (\bar{x} \otimes 1)(1 \otimes \bar{y}) = (1 \otimes \bar{y})(\bar{x} \otimes 1) = \acute{y}\grave{x}$.

More generally, the tensor-product construction combines n_1-forms on an affine space A_1 of dimension d_1 with n_2-forms on a space A_2 of dimension d_2 to produce tensor-product forms of bidegree $(n_1; n_2)$ on the product space $A_1 \times A_2$, that product space having $(d_1; d_2)$ as its *bidimension*. In a similar way, it combines n_1-sites over A_1 with n_2-sites over A_2 to produce $(n_1; n_2)$-sites over $A_1 \times A_2$. Even more generally, we could consider triple tensor products, such as tensor-product forms with tridegree $(n_1; n_2; n_3)$ on a product space $A_1 \times A_2 \times A_3$ of tridimension $(d_1; d_2; d_3)$. But the most important case in CAGD is tensor-product surfaces, where the parameter space is the product $L_1 \times L_2$ of two lines.

6.9 Tensor-product prototypes

The theory of Veronese prototypes extends to the tensor-product case, with the help of another concept from algebraic geometry: the *Segre embedding*.

Let's first consider the example of biquadratic tensor-product surfaces. The prototype of such surfaces is the surface $\sigma_{2;2}$ that takes the point (P_1, P_2) in the product space $L_1 \times L_2$ to the site $\sigma_{2;2}(P_1, P_2) := P_1^2 P_2^2$, lying in the affine space $(\mathrm{Sym}_2(\hat{L}_1) \otimes \mathrm{Sym}_2(\hat{L}_2))^{\downarrow} = \mathbf{R}_{2;2}[C_1, \varphi_1; C_2, \varphi_2]^{\downarrow}$. That affine space is 8-dimensional, a typical element of it, a unit-weight $(2;2)$-site s on $L_1 \times L_2$, being uniquely expressible in the form

$$\begin{aligned} s = {} & C_1^2 C_2^2 && + s_{01} C_1^2 C_2 \varphi_2 && + s_{02} C_1^2 \varphi_2^2 \\ & + s_{10} C_1 \varphi_1 C_2^2 && + s_{11} C_1 \varphi_1 C_2 \varphi_2 && + s_{12} C_1 \varphi_1 \varphi_2^2 \\ & + s_{20} \varphi_1^2 C_2^2 && + s_{21} \varphi_1^2 C_2 \varphi_2 && + s_{22} \varphi_1^2 \varphi_2^2. \end{aligned} \tag{6.9-1}$$

The sites s in that 8-space that lie on the prototypical surface $\sigma_{2;2}$ are those that factor as the product of two perfect squares: $s = \sigma_{2;2}(\grave{u}_1, \acute{u}_2) = \grave{u}_1^2 \acute{u}_2^2 = (C_1 + u_1\varphi_1)^2 (C_2 + u_2\varphi_2)^2$. When we use a biquadratic surface patch in one of our designs, say parameterized over the rectangle $[\grave{a} \,..\, \grave{b}] \times [\acute{c} \,..\, \acute{d}\,]$, we can view that patch as an affine transform of the prototypical patch $\sigma_{2;2}([\grave{a} \,..\, \grave{b}] \times [\acute{c} \,..\, \acute{d}\,])$.

And one convenient way to specify the instancing transformation that we intend is to specify the images of the nine sites $(\grave{a}^{2-i}\grave{b}^{i}\acute{c}^{2-j}\acute{d}^{j})_{0\le i,j\le 2}$, those nine images being the Bézier points of the resulting biquadratic patch.

Viewed more abstractly, the prototype $\sigma_{n_1;n_2}$ for tensor-product surfaces of bidegree $(n_1;n_2)$ can be thought of as $\sigma_{n_1;n_2} = \theta_{(1;1),(n_1;n_2)}$, a Veronese prototype for forms of bidegree $(n_1;n_2)$ on a product space of bidimension $(1;1)$. The *Segre embedding* is a construction in algebraic geometry that is relevant here, since it lets us combine two Veronese prototypes, say θ_{d_1,n_1} and θ_{d_2,n_2}, into a tensor-product prototype $\theta_{(d_1;d_2),(n_2;n_2)}$. The Segre embedding [30, 34] maps the product of (k_1-1)-space and (k_2-1)-space into (k_1k_2-1)-space. For example, if A is an affine plane and B is an affine 3-space, so $k_1=3$ and $k_2=4$, the Segre embedding maps $A\times B$ into an affine space of dimension $3\cdot 4-1=11$ by the rule

$$\bigl((1,u,v),(1,x,y,z)\bigr)\mapsto \begin{pmatrix}1, & x, & y, & z,\\ u, & ux, & uy, & uz,\\ v, & vx, & vy, & vz\end{pmatrix},$$

where we have written all three weight coordinates as explicit 1's to make the pattern clearer. Note that a point in the target 11-space, that is, a 3-by-4 matrix of real numbers with a 1 in the upper-left corner, lies in the image of this Segre embedding just when that matrix has rank 1.

A tensor-product Veronese prototype is the image, under the appropriate Segre embedding, of the Cartesian product of two separate Veronese prototypes. For example, suppose that we want to construct $\theta_{(d_1;d_2),(n_2;n_2)}$, the prototype for forms of bidegree $(n_1;n_2)$ on a product space of bidimension $(d_1;d_2)$. We begin with the separate Veronese prototypes θ_{d_1,n_1} and θ_{d_2,n_2}, which sit in affine spaces of dimensions $\binom{n_1+d_1}{n_1}-1$ and $\binom{n_2+d_2}{n_2}-1$. Setting $k_i := \binom{n_i+d_i}{n_i}$ for i in $\{1,2\}$, we then use the Segre embedding with parameters $(k_1;k_2)$ to embed the Cartesian product of those separate Veronese prototypes into an affine space of dimension $k_1k_2-1=\binom{n_1+d_1}{n_1}\binom{n_2+d_2}{n_2}-1$.

Chapter 7

The Paired-Algebras Framework

We have built the algebras of forms and sites as separate algebras, as shown in Figure 5.1. But the two algebras realize their full power only when we pair up the space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms on A with the space $\mathrm{Sym}_n(\hat{A})$ of n-sites over A, for each n, so that each can represent the dual of the other. For example, let P be a point in A and consider the evaluate-at-P functional on n-forms, that is, the linear functional ϵ_P that takes an n-form f as its argument and returns the real number $\epsilon_P(f) := f(P)$. Once we choose a pairing between n-forms and n-sites, we can represent that linear functional ϵ_P as a certain n-site. It turns out that there are two reasonable choices: Either $\epsilon_P = P^n$ or $\epsilon_P = P^n/n!$.

Warning: We are about to take the only mathematical step in the entire construction of the paired algebras where there is a real choice about what to do. There are two candidate pairings between the spaces $\mathrm{Sym}_n(\hat{A}^*)$ and $\mathrm{Sym}_n(\hat{A})$, the *summed pairing* and the *averaged pairing*, and they differ by a factor of $n!$. Adopting the summed pairing leads to an annoying factor of $n!$ in any formula that evaluates an n-ic — for example, the denominator in the formula $\epsilon_P = P^n/n!$. But it leads to simple, powerful, and familiar formulas for differentiation. Adopting the averaged pairing would simplify evaluation a bit, at the price of complicating differentiation a lot. I argue in Appendix B that the summed pairing is the wiser overall choice, and this monograph follows my advice. I hope that other researchers in CAGD will find my arguments convincing, lest we all find ourselves bedeviled by conflicting conventions. Sad to say, there is no way to make the formulas for evaluation and for differentiation both come out pretty. Indeed, evaluating an n-form is essentially the same process as differentiating that form n times in the same direction — except that the latter result exceeds the former by that annoying factor of $n!$.

This controversy about how to scale the pairing maps is unfortunate,

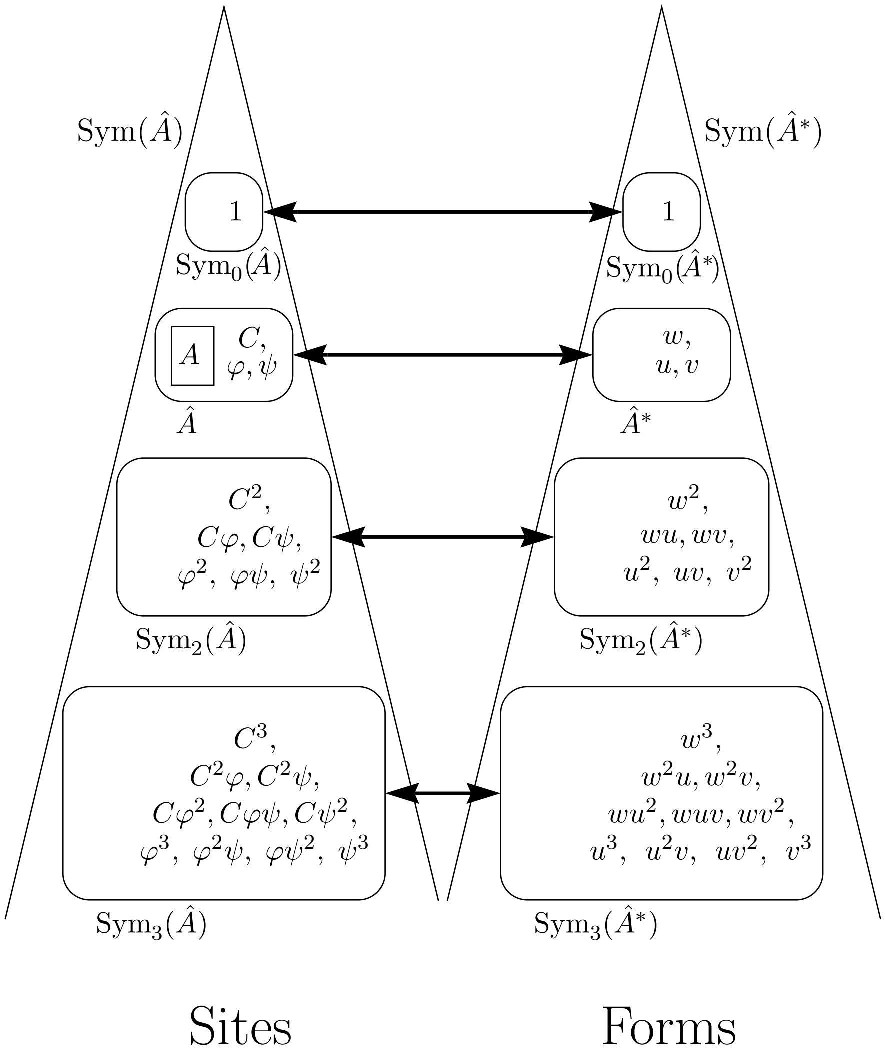


Figure 7.1: The paired-algebras framework

but don't be too downcast: The controversy concerns only a numeric factor. The deep benefit of the pairing maps is that they allow us to exploit the multiplication in the algebra of sites as a new tool with which to study linear functionals on forms, and vice versa. For example, the formula $\epsilon_P = P^n/n!$ tells us that the evaluate-at-P functional ϵ_P is a perfect n^{th} power. So, in the linear space $\mathrm{Sym}_n(\hat{A})$ of n-sites, what is the geometry of those n-sites that represent point evaluations? Answer: Except for the annoying factor of $n!$, that set is precisely the Veronese d-fold of parametric degree n, the set $\theta_{d,n}(A)$ of all perfect n^{th} powers of points.

7.1 Picturing our goal

Figure 7.1 shows our final goal at last, the *paired-algebras framework*. The double-headed arrow on the n^{th} level denotes the pairing that we shall adopt between the spaces $\mathrm{Sym}_n(\hat{A})$ and $\mathrm{Sym}_n(\hat{A}^*)$, thereby allowing us to use each to represent the dual of the other.

The issue about which pairings to adopt, the summed or the averaged, leaves the lower levels in Figure 7.1 somewhat fuzzy. But note that there is no issue about levels 0 and 1. On level 1, we want the fundamental pairing between the spaces $\hat{A}$ and $\hat{A}^*$ of anchors and coanchors. On level 0, we want the pairing that combines two real numbers by multiplying them. The annoying factor of $n!$ becomes an issue only once n exceeds 1.

Math remark: In the paired-algebras framework, the linear space $\mathrm{Sym}_n(\hat{A})^*$ of dual functionals on n-sites is represented by the space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms, and the same is true with forms and sites reversed. Is a similar representation possible for the algebras in their entirety? The whole algebra of sites $\mathrm{Sym}(\hat{A})$ is also a linear space, albeit of infinite dimension, so it has a dual $\mathrm{Sym}(\hat{A})^*$. Can we use the whole algebra of forms $\mathrm{Sym}(\hat{A}^*)$ to represent that dual, all at once? That is, can we combine all of the separate double-headed arrows in Figure 7.1 into one fat double-headed arrow?

No, because of the blow-up in dimension that happens when we take the dual of an infinite-dimensional space. The dual space $\mathrm{Sym}(\hat{A})^*$ is huge. We can think of an element F in that dual space as a sum $F = \sum_{n\geq 0} f_n$, where each f_n is an n-form, but with no requirement that all but finitely many of the (f_n) must be zero. Instead, all of the (f_n) may be nonzero simultaneously. The infinite sum F still determines a linear map from sites to real numbers as follows. Any site s can be uniquely expanded as a sum $s = \sum_{n\geq 0} s_n$ of its graded components (s_n), where s_n is an n-site and where all but finitely many of the (s_n) are zero. So we can define $F(s)$ by the rule $F(s) := \sum_{n\geq 0}\langle f_n, s_n\rangle$, and the resulting sum of real numbers will always be a finite sum. But the algebra of all such infinite sums $F = \sum_{n\geq 0} f_n$ is vastly larger than the algebra of forms, in which only finitely many of the graded components (f_n) are allowed to be nonzero.

Bourbaki [4] introduces the concept of the *graded dual* of a graded algebra, which is the direct sum of the duals of the graded slices. By exploiting that concept, we could combine all of the double-headed arrows in Figure 7.1 into one fat arrow: a fat arrow asserting that each of the graded algebras $\mathrm{Sym}(\hat{A})$ and $\mathrm{Sym}(\hat{A}^*)$ can represent the graded dual of the other.

7.2 Lineals and perfect powers

To prepare for defining the pairing maps, we go over some easy lemmas about the algebras of forms and sites.

Lemma 7.2-1 *Given any affine space A of finite dimension $d := \dim(A)$ and any nonnegative integer n, every n-form on A is a linear combination of real-lineal n-forms. The same goes for sites over A.*

Proof Fix some basis $(c_0, \ldots, c_d)$ for the space $\hat{A}^*$ of coanchors on A. As on the second line of Table 5.1, we can then concretely construct the linear space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms on A as the space $\mathbf{R}_n[c_0, \ldots, c_d]$ of all polynomials that are homogeneous of degree n in the variables $(c_0, \ldots, c_d)$.

Let α denote a multi-index $\alpha = (\alpha_0, \ldots, \alpha_d)$, where each α_k is nonnegative and where $|\alpha| := \alpha_0 + \cdots + \alpha_d$ satisfies $|\alpha| = n$. We then denote by c^α the n-form $c^\alpha := c_0^{\alpha_0} \cdots c_d^{\alpha_d}$. The n-forms $(c^\alpha)_{|\alpha|=n}$ form a basis for the space $\mathbf{R}_n[c_0, \ldots, c_d]$ of n-forms; let's call it the *monomial basis*.

Every n-form on A is a linear combination of the $\binom{n+d}{n}$ monomials in this basis. And each of those monomials is clearly real-lineal — indeed, splits as the product of n elements of our chosen basis $(c_0, \ldots, c_d)$ for $\hat{A}^*$. So every n-form is a linear combination of real-lineal n-forms. $\square$

In fact, more is true.

Lemma 7.2-2 *Given any affine space A of finite dimension $d := \dim(A)$ and any nonnegative n, every n-form on A is a linear combination of n-forms that are perfect n^{th} powers. Again, the same goes for sites over A.*

Proof We know from Lemma 7.2-1 that every n-form is a linear combination of real-lineal n-forms. So it suffices to prove that every real-lineal n-form is a linear combination of perfect n^{th} powers.

Let $f = h_1 \cdots h_n$ be a real-lineal n-form on A. In the case $n = 2$, we have

$$h_1 h_2 = \frac{(h_1 + h_2)^2 - h_1^2 - h_2^2}{2}. \tag{7.2-3}$$

This is quadratic case of the Polarization Identity. An inclusion-exclusion argument establishes the general case

$$h_1 \cdots h_n = \frac{1}{n!} \sum_{S \subseteq \{1,\ldots,n\}} (-1)^{n-|S|} \Bigl(\sum_{i \in S} h_i\Bigr)^n, \tag{7.2-4}$$

which expresses the product $h_1 \cdots h_n$ as a linear combination of $2^n - 1$ perfect n^{th} powers. □

It might seem more natural to require the set S to be nonempty in the sum in Equation 7.2-4, since the term resulting from $S = \emptyset$ is zero in any case; that's why there are only $2^n - 1$ terms, instead of 2^n. But the resulting term is actually 0^n, which is 0 when n is positive, but is $0^0 = 1$ when $n = 0$. And that 1 is needed to make Equation 7.2-4 correct in the case $n = 0$.

Math remark: Both Lemma 7.2-1 and Lemma 7.2-2 hold equally well in the algebrization of any linear space X; they don't use any special properties of the spaces $\hat{A}^*$ or $\hat{A}$ of coanchors or anchors. Indeed, those lemmas hold even when X is infinite dimensional, with the same proofs. But Lemma 7.2-2 does not hold over fields of finite characteristic, because of the division by $n!$.

In the case of sites, Lemma 7.2-2 can be strengthened a bit further still.

Lemma 7.2-5 *Given any affine space A of finite dimension $d := \dim(A)$ and any nonnegative n, every n-site over A is a linear combination of n-sites that are n^{th} powers of points — that is, n^{th} powers of anchors of unit weight.*

Vacant remark: We make the convention that the real number 1 is the 0^{th} power of a point. This convention isn't controversial when $d \geq 0$, since $P^0 = 1$, for all points P in A. When $d = -1$, however, there are no points P in A to raise to the 0^{th} power. We argue that 1 is a 0^{th} power of a point anyway, since we can write 1 as a product of 0 factors, a product in which all factors are equal and all factors — of which there aren't any — are points.

Proof When $n = 0$, an n-site over A is a real number and hence a scalar multiple of 1, which is a 0^{th} power of a point by the convention that we just adopted. So we may suppose that n is positive.

Since $(tp)^n = t^n p^n$ for any real number t and anchor p, it suffices to show that every n-site is a linear combination of n^{th} powers of anchors whose weights are not zero. To see this, choose a barycentric reference frame $(R_0, \ldots, R_d)$ for A, consisting entirely of points. Every n-site is then a linear combination of the monomials $(R^\alpha)_{|\alpha|=n}$, which form a basis for the space $\operatorname{Sym}_n(\hat{A})$ of n-sites. When we apply Equation 7.2-4 to such a monomial, all of the n^{th} powers that arise will have the form $(\beta_0 R_0 + \cdots + \beta_d R_d)^n$, for some multi-index β with $|\beta| \leq n$. If $|\beta| = 0$, then we are talking about 0^n, which is zero since n is positive. So we end up with n^{th} powers of anchors whose weights are positive integers. □

Let's use $\beta \cdot R$ to denote the sum $\beta \cdot R := \beta_0 R_0 + \cdots + \beta_d R_d$. We saw, in proving Lemma 7.2-5, that the perfect n^{th} powers $((\beta \cdot R)^n)_{|\beta| \leq n}$ span the entire space $\mathrm{Sym}_n(\hat{A})$ of n-sites. We shall show, in Proposition 8.4-1, that the subset $((\beta \cdot R)^n)_{|\beta| = n}$ actually forms a basis.

Exercise 7.2-6 Equation 7.2-3 expresses the product $h_1 h_2$ as a linear combination of three perfect squares; but the similar identity

$$h_1 h_2 = \frac{(h_1 + h_2)^2 - (h_1 - h_2)^2}{4}$$

uses only two squares, which is clearly the fewest possible. By generalizing this latter identity, write the product $h_1 \cdots h_n$ as a linear combination of only 2^{n-1} perfect n^{th} powers.

Answer: We have

$$h_1 \cdots h_n = \frac{1}{2^{n-1} n!} \sum_{\substack{S \cup T = \{2, \ldots, n\} \\ S \cap T = \emptyset}} (-1)^{|T|} \left(h_1 + \sum_{i \in S} h_i - \sum_{j \in T} h_j \right)^n .$$

Could we get by with even fewer than 2^{n-1} perfect n^{th} powers? In the case $n = 3$, it turns out that four cubes are necessary; a Gröbner-basis calculation with Maple [9] or Singular [27] establishes that it is impossible to write the product xyz as the sum of three terms, each of which is the cube of a linear combination of x, y, and z. But I don't know whether eight fourth powers are necessary, when $n = 4$; perhaps fewer would suffice?

7.3 The Permanent Identity

With those lemmas under our belt, it is time to define the pairing, for each n, between n-forms on A and n-sites over A. That is, we want to pair the linear spaces $\mathrm{Sym}_n(\hat{A}^*)$ and $\mathrm{Sym}_n(\hat{A})$. Those two linear spaces have the same finite dimension; so there are lots of pairings between them. The key to a useful theory is to find a pairing that interacts well with the multiplications in the algebras of forms and sites.

By Lemma 7.2-1, defining such a pairing on n-forms and n-sites that are real-lineal would suffice to define it everywhere. So let's think about an n-form $f = h_1 \cdots h_n$ that is the product of n coanchors and an n-site $s = p_1 \cdots p_n$ that is the product of n anchors. What value should we assign to $\langle f, s \rangle = \langle h_1 \cdots h_n, p_1 \cdots p_n \rangle$? One value that might be relevant is the product $\langle h_1, p_1\rangle\langle h_2, p_2\rangle \cdots \langle h_n, p_n\rangle$. But the order in which we numbered the n factors h_1 through h_n of f was arbitrary, and the same for s; so there is no reason why matching h_i with p_i, for i from 1 to n, makes more sense than any other way of matching up the n coanchors with the n anchors. To be symmetric, we should consider all possible such matchings and then — and here is where the $n!$ issue arises — we either sum or average the results.

Let's say that a pairing map between n-forms and n-sites satisfies the *Summed Permanent Identity* when, for all coanchors h_1 through h_n and for all anchors p_1 through p_n, we have

$$\langle h_1 \cdots h_n, p_1 \cdots p_n\rangle = \sum_{\nu \in \mathbf{S}_n} \prod_{k \in [1..n]} \langle h_k, p_{\nu(k)}\rangle, \tag{7.3-1}$$

where the summation index ν varies over the symmetric group $\mathbf{S}_n$ of all $n!$ permutations of the integers from 1 to n. The *Averaged Permanent Identity* is the same, except that we divide the sum by $n!$:

$$\langle h_1 \cdots h_n, p_1 \cdots p_n\rangle = \frac{1}{n!} \sum_{\nu \in \mathbf{S}_n} \prod_{k \in [1..n]} \langle h_k, p_{\nu(k)}\rangle. \tag{7.3-2}$$

The term "permanent" is appropriate because that sum is the permanent of the n-by-n matrix whose $(i,j)^{\text{th}}$ entry is $\langle h_i, p_j\rangle$. Recall that the *permanent* of a matrix is like the determinant, except that all products are added; in the determinant, of course, products from even permutations ν are added, but those from odd permutations are subtracted.† In the next section, we show that there exists a unique pairing between n-forms and n-sites that satisfies the Summed Permanent Identity; dividing that pairing by $n!$ gives the unique pairing that satisfies the Averaged Permanent Identity.

When we wrote the Permanent Identities, we used angle brackets both for the pairing on the left-hand side, the new pairing between n-forms and n-sites, and also for the pairing on the right-hand side, the fundamental pairing between coanchors and anchors. In the particular case $n = 1$, that could potentially lead to confusion. Fortunately, both of the Permanent Identities reduce, in the case $n = 1$, to the identity $\langle h_1, p_1\rangle = \langle h_1, p_1\rangle$. Since the new pairing is thus required to agree with the old wherever the old is defined, it causes no confusion to use the same angle brackets for both.

Exercise 7.3-3 What values do the two Permanent Identities mandate for the pairing value $\langle f, s\rangle$ in the case $n = 0$, when the 0-form f and the 0-site s are simply real numbers?

†Math remark: The permanent arises because we are studying the symmetric algebra $\mathrm{Sym}(X)$. The analogous formula for the alternating algebra $\mathrm{Alt}(X)$ has the determinant instead. That explains why the alternating algebra is appropriate for multivariate calculus, where the determinant of a Jacobian measures the ratio of two signed volumes.

Answer: The sum has a single term, and that single term is an empty product; so both of the Permanent Identities mandate that $\langle 1, 1\rangle = 1$. By bilinearity, it follows that $\langle f, s\rangle = fs$, for all real numbers f and s.

We started this section by using Lemma 7.2-1 to restrict our attention to n-forms and n-sites that are real-lineal. By using the stronger Lemma 7.2-2, we could have gone further and restricted our attention to n-forms and n-sites that are perfect n^{th} powers. What do the Permanent Identities say about that special case? When $h_1 = \cdots = h_n = h$ and $p_1 = \cdots = p_n = p$, the Summed Permanent Identity mandates that $\langle h^n, p^n\rangle = n!\,\langle h, p\rangle^n$, the $n!$ arising from the $n!$ different ways of matching up the n identical h's with the n identical p's. The Averaged Permanent Identity divides out that $n!$, giving the simpler formula $\langle h^n, p^n\rangle = \langle h, p\rangle^n$.

7.4 Defining the pairing

Recall, from Section 2.3, that every bilinear form $B\colon X \times Y \to \mathbf{R}$ has an associated matrix M, under the convention that the scalar $B(x, y)$ is given by the matrix product $x^{\mathbf{t}}My$. Furthermore, the bilinear form B is a pairing just when its matrix M is invertible.

Proposition 7.4-1 *Let A be an affine space of finite dimension $d := \dim(A)$, and let n be nonnegative. There is a unique pairing between the space $\operatorname{Sym}_n(\hat{A}^*)$ of n-forms on A and the space $\operatorname{Sym}_n(\hat{A})$ of n-sites over A that satisfies the Summed Permanent Identity. We christen it the* summed pairing. *Dividing the summed pairing by $n!$ gives the* averaged pairing, *the unique pairing that satisfies the Averaged Permanent Identity.*

Proof We construct the spaces $\operatorname{Sym}_n(\hat{A}^*)$ and $\operatorname{Sym}_n(\hat{A})$ concretely by fixing bases. Let $(a_0, \ldots, a_d)$ be some basis for the linearized space $\hat{A}$ of anchors over A, and let $(c_0, \ldots, c_d)$ be the basis for the space $\hat{A}^*$ of coanchors on A that is dual to $(a_0, \ldots, a_d)$. The duality constraints tell us that

$$\langle c_i, a_j\rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

We can then represent the space $\operatorname{Sym}_n(\hat{A}^*)$ of n-forms as the space of polynomials $\mathbf{R}_n[c_0, \ldots, c_d]$, while the space $\operatorname{Sym}_n(\hat{A})$ of n-sites is $\mathbf{R}_n[a_0, \ldots, a_d]$. Note that both spaces have dimension $\binom{n+d}{n}$.

Let γ denote a multi-index $\gamma = (\gamma_0, \ldots, \gamma_d)$ with $|\gamma| := \gamma_0 + \cdots + \gamma_d = n$. The n-forms $(c^\gamma)_{|\gamma|=n}$ form the monomial basis for the space $\mathbf{R}_n[c_0, \ldots, c_d]$ of n-forms. In a similar way, the n-sites $(a^\alpha)_{|\alpha|=n}$ form the monomial basis for the space $\mathbf{R}_n[a_0, \ldots, a_d]$ of n-sites. In order to define any bilinear map

$B\colon \mathbf{R}_n[c_0,\ldots,c_d] \times \mathbf{R}_n[a_0,\ldots,a_d] \to \mathbf{R}$, it suffices to specify the values of B on those n-forms and n-sites that are monomials, that is, to define the $\binom{n+d}{n}$-by-$\binom{n+d}{n}$ matrix of real numbers $(B(c^\gamma, a^\alpha))_{|\gamma|=|\alpha|=n}$. For the map B to be a pairing, that matrix must have full rank.

Each n-form c^γ and each n-site a^α is real-lineal, so the Summed Permanent Identity 7.3-1 leaves us no choice about how to fill in the matrix of B. What value does it mandate for $\langle c^\gamma, a^\alpha\rangle$? If the multi-indices γ and α are distinct, then all $n!$ ways of matching the n coanchor factors with the n anchor factors will involve at least one match-up $\langle c_k, a_l\rangle$ with $k \neq l$. So all $n!$ terms in the resulting sum will be zero; for $\gamma \neq \alpha$, we have $\langle c^\gamma, a^\alpha\rangle = 0$. What about when $\gamma = \alpha$? To get a nonzero term in the sum, we must match up, for each k from 0 to d, the γ_k copies of c_k in c^γ with the $\gamma_k = \alpha_k$ copies of a_k in a^γ, and we can do that in $\gamma_k!$ ways. So the number of nonzero terms is the product $\gamma_0!\,\gamma_1!\cdots\gamma_d!$, which we shall abbreviate as $\gamma!$. Since each nonzero term contributes 1, we conclude that the Summed Permanent Identity mandates:

$$\langle c^\gamma, a^\alpha\rangle = \begin{cases} \gamma! & \text{if } \gamma = \alpha \\ 0 & \text{otherwise.} \end{cases}$$

The matrix that we have just constructed is diagonal, with all of its diagonal entries nonzero; so it has full rank. We conclude that there exists a unique pairing that satisfies all *monomial instances* of the Summed Permanent Identity, that is, all instances in which each coanchor factor h_k lies in our chosen basis $(c_0,\ldots,c_d)$ for $\hat{A}^*$ and each anchor factor p_k lies in our chosen basis $(a_0,\ldots,a_d)$ for $\hat{A}$.

It remains to verify that this unique pairing in fact satisfies all instances of the Summed Permanent Identity. To see that, note that both sides of the Summed Permanent Identity are $(2n)$-linear functions from $(\hat{A}^*)^n \times (\hat{A})^n$ to the reals; that is, both sides vary linearly as a function of each h_k if the other h's and all of the p's are held fixed, and the same for each p_k. Thus, the validity of the Summed Permanent Identity extends by linearity from monomial instances to all instances. □

The matrix that we construct in this proof has zeros off the diagonal and has the positive integer $\gamma!$ in the (γ,γ) slot on the diagonal. Once $n \geq 2$, the diagonal entries are not all ones, and that engenders a warning. We chose the bases $(c_0,\ldots,c_d)$ and $(a_0,\ldots,a_d)$ for the spaces $\hat{A}^*$ of coanchors and $\hat{A}$ of anchors to be dual to each other. But the monomial bases $(c^\gamma)_{|\gamma|=n}$ and $(a^\alpha)_{|\alpha|=n}$ that we then constructed for the spaces $\mathbf{R}_n[c_0,\ldots,c_d]$ of n-forms and $\mathbf{R}_n[a_0,\ldots,a_d]$ of n-sites are not dual to each other — that is, they are not dual under the unique pairing that satisfies the Summed Permanent Identity.

Math remark: The existence of diagonal entries $\gamma!$ that exceed 1 makes the theory of symmetric algebras somewhat subtle. Among other things, over a field of prime characteristic, the analog of the summed pairing may fail to be a pairing; once n is at least the characteristic, the matrix that we get from the Summed Permanent Identity has zeros on the diagonal, and hence fails to have full rank. Alternating algebras are better behaved in this respect. In the alternating algebras $\mathrm{Alt}(\hat{A})$ and $\mathrm{Alt}(\hat{A}^*)$, the analogs of the monomial basis elements are $a_{i_1} \wedge \cdots \wedge a_{i_n}$ and $c_{i_1} \wedge \cdots \wedge c_{i_n}$ for $i_1 < \cdots < i_n$, with no repeated factors allowed. So the Summed Determinant Identity produces a matrix in which all of the diagonal entries are ones.

Averaging, rather than summing, divides everything by $n!$, so the (γ, γ) slot on the diagonal is $\gamma!/n!$. Recall that $\gamma!$ divides $n!$, for all γ with $|\gamma| = n$, since the quotient $\binom{n}{\gamma} := n!/\gamma!$ is a multinomial coefficient, and hence an integer. So the diagonal entries $\gamma!/n! = 1/\binom{n}{\gamma}$ are reciprocals of integers.

It's bad news that the diagonal entries of these matrices are not all equal. But it's good news that all of the off-diagonal entries are zero. It follows that the dual of a monomial basis differs only by some factorial scale factors from being itself a monomial basis.

Proposition 7.4-2 *Given a d-dimensional affine space A, let $(a_0, \ldots, a_d)$ and $(c_0, \ldots, c_d)$ be dual bases for the spaces $\hat{A}$ and $\hat{A}^*$ of anchors and coanchors, and consider the corresponding monomial bases $(a^\alpha)_{|\alpha|=n}$ and $(c^\gamma)_{|\gamma|=n}$ for the spaces $\mathrm{Sym}_n(\hat{A})$ and $\mathrm{Sym}_n(\hat{A}^*)$ of n-sites and n-forms. Under the summed pairing, the dual of the basis $(a^\alpha)_{|\alpha|=n}$ is the scaled monomial basis $(c^\gamma/\gamma!)_{|\gamma|=n}$. Alternatively, putting the scale factors on the other side, the dual of the basis $(c^\gamma)_{|\gamma|=n}$ is the basis $(a^\alpha/\alpha!)_{|\alpha|=n}$. Under the averaged pairing, the scale factors are larger by a factor of $n!$; so the dual of (a^α) is $(\binom{n}{\gamma} c^\gamma)$ and the dual of (c^γ) is $(\binom{n}{\alpha} a^\alpha)$.*

Proof In proving Proposition 7.4-1, we saw that the value $\langle c^\gamma, a^\alpha \rangle$ given by the summed pairing is

$$\langle c^\gamma, a^\alpha \rangle = \begin{cases} \gamma! & \text{if } \gamma = \alpha \\ 0 & \text{otherwise.} \end{cases}$$

Thus, to end up with dual bases, it suffices to divide one basis or the other by the factorial of its multi-index. Under the averaged pairing, we must also multiply one basis or the other by $n!$. □

7.5 Summing is better — trust me

We have reached an unpleasant juncture. We want to pair n-forms with n-sites, and we have found two candidate pairings, differing by a factor of $n!$.

Perfectly adequate theories can be erected based on either of those pairings, and reasonable people could prefer either theory.

So far, we have been laying the groundwork for the two theories in parallel. If we continue to develop the two in parallel, however, we shall be faced with two versions of every new formula, which seems a recipe for confusion. Better, instead, to develop one theory in isolation. We can return later to discuss how the other theory would differ from the one that we then understand.

Until further notice, therefore, we are going to pair n-forms with n-sites using the summed pairing, the unique pairing map that satisfies the Summed Permanent Identity 7.3-1. We won't reopen the summing-versus-averaging debate until Appendix B, where we analyze how all of our formulas would change if we averaged, instead of summing. Some things would get prettier, others would get uglier. But Appendix B argues that, in the context of CAGD, summing beats averaging overall.

Unfortunately, the costs of summing show up before its benefits; that is, summing clutters up some formulas that you learn right away, thereby enabling some formulas that you don't learn until later to be cleaner. So we are going to run across annoying factors of $n!$ quite soon. Please grant summing the benefit of the doubt until Appendix B.

7.6 Evaluating an n-form

Having chosen the summed pairing, we now have a slew of formulas to cover, formulas that relate pairing to other operations on forms and sites. The first of those operations is evaluation, and the basic rule for evaluation is this:

> To evaluate an n-ic under the summed pairing, pair it with an n^{th} power and divide by $n!$.

Proposition 7.6-1 *If f is any n-form on an affine space A and if P is any point in A, evaluation of f at P is related to pairing by the formula*

$$f(P) = \langle f, P^n/n!\,\rangle. \tag{7.6-2}$$

More generally, the same formula holds with the point P replaced by any anchor p over A:

$$f(p) = \langle f, p^n/n!\,\rangle. \tag{7.6-3}$$

Dually, if s is any n-site over A and h is any coanchor on A, we have

$$s(h) = \langle h^n/n!\,, s\rangle. \tag{7.6-4}$$

Proof Lemma 7.2-2 tells us that every n-form on A is a linear combination of perfect n^{th} powers. Since evaluation at a fixed point P is a linear process, it suffices to prove Formula 7.6-2 when $f = h^n$ is the n^{th} power of some coanchor h. In that case, the left-hand side is $h^n(P) = (h(P))^n = \langle h, P\rangle^n$. On the right-hand side, the Summed Permanent Identity 7.3-1 tells us that $\langle h^n, P^n/n!\,\rangle = \langle h^n, P^n\rangle/n! = \bigl(n!\,\langle h, P\rangle^n\bigr)/n! = \langle h, P\rangle^n$. The same proof applies to any anchor p over A, and the proof of the dual result is symmetric. □

This correspondence between evaluating an n-ic and pairing it with an n^{th} power gives us a new perspective from which to view Lemma 7.2-5. That lemma tells us that every n-site is a linear combination of perfect n^{th} powers of points. So, if we know the value $\langle f, P^n\rangle$ for every point P in A, we can compute the value $\langle f, s\rangle$ for any n-site s, which determines the n-form f completely. That is no surprise, since knowing the value $\langle f, P^n\rangle$ is the same as knowing the value $f(P) = \langle f, P^n/n!\,\rangle$; and an n-form f is determined by its values $f(P)$ at all points P. Indeed, the process of determining the polynomial f from a sufficiently large and sufficiently independent set of its values $f(P)$ is the familiar process of polynomial interpolation.

Back in Chapter 4, when we adopted the homogenized framework, we noted that it makes sense to evaluate an n-form at an anchor that isn't a point. What does that process mean geometrically? There are two cases.

Consider first an anchor p whose weight $w(p)$ is nonzero. Such an anchor is a scalar multiple of a point; so we have $p = w(p)P$, where the point P is given by $P := p/w(p)$. Since an n-form f is homogeneous of degree n, it follows that $f(p) = f(w(p)P) = w(p)^n f(P)$.

The remaining case is more subtle: the case of a vector π over A. What is the value $f(\pi)$? It turns out that

$$f(\pi) = \frac{1}{n!}(D_\pi)^n f. \tag{7.6-5}$$

That is, evaluating an n-form at the vector π is the same as differentiating that n-form n times, each time in the direction of the vector π — and then dividing by $n!$. Note that the n^{th} derivative of an n-form is a constant, so the right-hand side of Formula 7.6-5 needs no further evaluation. We won't pause to verify Formula 7.6-5 now, since it will follow easily once we can differentiate, as well as evaluate, by pairing with an appropriate site. But Formula 7.6-5 should at least seem plausible, on the following grounds. A vector is a scaled version of a point at infinity. So evaluating an n-form at a vector means finding out the leading term of what happens as we go to infinity in that direction. Fix any point Q in A and consider the function $g(t) := f(Q+t\pi)$. Taylor's Theorem tells us that $g(t) = \sum_{0\le k\le n} g^{(k)}(0)t^k/k!$. As t tends to infinity, the dominant term is the last, in which the coefficient of t^n is $g^{(n)}(0)/n! = ((D_\pi)^n f)/n! = f(\pi)$. Thus, the value of an n-form at a vector tells us, in a natural way, about what happens as we go to infinity in the direction of that vector. Note also that the denominator of $n!$ in Formula 7.6-5 is closely related to the denominators in a Taylor series.

So, evaluating an n-ic is essentially the same process as differentiating it n times, always in the same direction — the same process, that is, except for the annoying factor of $n!$. Our choice of the summed pairing, over the averaged pairing, means that our formulas for evaluation are cluttered with factors of $n!$, while our formulas for differentiation, coming soon, are pretty. For the tradeoff between the two pairings, see Appendix B.

7.6.1 Evaluating the blossom of an n-form

Blossoming replaces n-ic dependence on a single parameter with n-affine dependence on n symmetric parameters. By exploiting the paired algebras of forms and sites, we have learned how to represent n-ic dependence on a single parameter p as the process of pairing with the n-site $p^n/n!$. This makes it trivial to blossom: We merely pair, instead, with $p_1 \cdots p_n/n!$.

Proposition 7.6-6 *Let f be any n-form on an affine space A, let $\tilde{f}\colon A^n \to \mathbf{R}$ be its multiaffine blossom (a.k.a. polar form), and let P_1 through P_n be any points in A. We then have*

$$\tilde{f}(P_1, \ldots, P_n) = \langle f, P_1 \cdots P_n/n!\,\rangle. \tag{7.6-7}$$

The blossom $\tilde{f}$ extends uniquely to a multilinear function $\tilde{f}\colon \hat{A}^n \to \mathbf{R}$, which satisfies

$$\tilde{f}(p_1, \ldots, p_n) = \langle f, p_1 \cdots p_n/n!\,\rangle. \tag{7.6-8}$$

for all anchors p_1 through p_n on A.

Proof The product $p_1 \cdots p_n$ is a linear function of each factor, is symmetric, and reduces to the n^{th} power p^n when $p_1 = \cdots = p_n = p$. $\square$

7.7 Formulas for differentiation

Relating differentiation to pairing is more subtle, because differentiating an n-form produces an $(n-1)$-form, rather than a scalar. Here is the basic story: Differentiating an n-form f in the direction of a vector π corresponds to setting to π one of the n factors of the n-site with which f eventually gets paired. The other factors of that n-site are set only later, when the derivative $D_\pi f$ is itself evaluated or further differentiated. This process is best understood from a few examples.

Proposition 7.7-1 *Let f be an n-form on the affine space A, let π be a vector over A, and let R be a point in A. Differentiating f in the direction π and then evaluating the resulting $(n-1)$-form at R is related to pairing by the formula:*

$$D_\pi f(R) = \langle f, \pi R^{n-1}/(n-1)!\,\rangle. \tag{7.7-2}$$

Proof Formula 7.6-2 tells us how to evaluate by pairing, so we can simply calculate:

$$\begin{aligned} D_\pi f(R) &= \lim_{t\to 0} \frac{f(R+t\pi)-f(R)}{t} \\ &= \lim_{t\to 0} \frac{\langle f, (R+t\pi)^n/n!\,\rangle - \langle f, R^n/n!\,\rangle}{t} \\ &= \lim_{t\to 0} \langle f, (R+t\pi)^n - R^n\rangle/n!\,t \\ &= \lim_{t\to 0} \langle f, (R^n + nt\pi R^{n-1} + O(t^2)) - R^n\rangle/n!\,t \\ &= \lim_{t\to 0} \langle f, nt\pi R^{n-1} + O(t^2)\rangle/n!\,t \\ &= \langle f, \pi R^{n-1}/(n-1)!\,\rangle. \end{aligned}$$

Note that $R^{n-1}/(n-1)!$ is the site with which we would pair the $(n-1)$-form $D_\pi f$, in order to evaluate it at the point R. So the differentiation merely sets to π one of the factors of the n-site with which f eventually gets paired. □

The fact that π is a vector, that is, that its weight is 0, plays no role in that proof. Indeed, we shall use the standard limit formula to define the derivative of a n-form f on A in the direction of any anchor p over A:

$$D_p f(R) := \lim_{t\to 0} \frac{f(R+tp)-f(R)}{t}.$$

With this definition, Formula 7.7-2 extends from vectors to arbitrary anchors; we have $D_p f(R) = \langle f, pR^{n-1}/(n-1)!\,\rangle$. But we have to be a bit careful. In some contexts, the directions in which it is legal to differentiate are restricted to be vectors. For example, when we say that two forms f and g, possibly of different degrees, agree to k^{th} order at a point R, we are saying that $D_{\pi_1}\cdots D_{\pi_j} f(R) = D_{\pi_1}\cdots D_{\pi_j} g(R)$, for any $j \le k$ and any vectors π_1 through π_j. But only vectors are permissible as directions in this context, not arbitrary anchors, as discussed in Section 7.11.

Differentiating multiple times is an easy generalization.

Proposition 7.7-3 *Let f be an n-form on the affine space A and let p_1 through p_k and r be anchors over A, for some $k \le n$. Taking f and differentiating k times, in the directions of the anchors p_1 through p_k, and then evaluating the resulting $(n-k)$-form at r is related to pairing by the formula*

$$D_{p_1}\cdots D_{p_k} f(r) = \langle f, p_1\cdots p_k r^{n-k}/(n-k)!\,\rangle. \tag{7.7-4}$$

Proof The argument for $k = 2$ should make the pattern clear:

$$\begin{aligned} D_p D_q f(r) &= \lim_{t\to 0} \frac{D_q f(r+tp) - D_q f(r)}{t} \\ &= \lim_{t\to 0} \frac{\langle f, q(r+tp)^{n-1}/(n-1)!\,\rangle - \langle f, qr^{n-1}/(n-1)!\,\rangle}{t} \\ &= \lim_{t\to 0} \langle f, q(n-1)r^{n-2}tp + O(t^2)\rangle/(n-1)!\,t \\ &= \langle f, pq\,r^{n-2}/(n-2)!\,\rangle. \qquad \square \end{aligned}$$

Each differentiation thus sets one factor of the n-site s with which the n-form f eventually gets paired. Differentiating k times leaves us with an $(n-k)$-form, which we can then evaluate at an anchor r by setting the remaining $n-k$ factors of s to $r^{n-k}/(n-k)!$. If we differentiate n times, we get the constant $D_{p_1}\cdots D_{p_n} f(r) = \langle f, p_1 \cdots p_n\rangle$, independent of r. If we further specialize to the case $p_1 = \cdots = p_n = p$ in which all n directions are the same, we find that $(D_p)^n f = \langle f, p^n\rangle$. Since the rule for evaluation at p is $f(p) = \langle f, p^n/n!\,\rangle$, we see that the relationship between evaluating an n-ic and differentiating it n times is indeed as we claimed in Formula 7.6-5.

For the record, here is the formula for differentiating an n-form k times and then evaluating the blossom of the resulting $(n-k)$-form at the anchors r_1 through r_{n-k}:

$$(D_{p_1}\cdots D_{p_k} f)^{\sim}(r_1, \ldots, r_{n-k}) = \langle f, p_1 \cdots p_k\, r_1 \cdots r_{n-k}/(n-k)!\,\rangle.$$

7.8 The contraction operators

If we set k of the factors of the n-site with which an n-form will eventually get paired, we have essentially converted that n-form into an $(n-k)$-form. The operator that does that conversion is called *contraction.*

Let A be an affine space, let f be an n-form on A, and let s be a k-site over A, where $k \le n$. In the special case $k = n$, we know how to combine f with s to produce a real number: the pairing value $\langle f, s\rangle$. When $k < n$, we can't get a real number. But we can produce, from f and s, a mapping that takes $(n-k)$-sites to real numbers: the mapping $t \mapsto \langle f, st\rangle$, for any $(n-k)$-site t. This mapping is an element of the dual space $\mathrm{Sym}_{n-k}(\hat{A})^*$, which we are representing as the space $\mathrm{Sym}_{n-k}(\hat{A}^*)$ of $(n-k)$-forms. Thus, the n-form f and the k-site s together determine an $(n-k)$-form, which is written $f \llcorner s$ and called the *contraction of f on s* or the *s-contraction of f*. The terms "internal product" and "inner product" are also used. Note that, in the expression $f \llcorner s$, the vertical stroke of the operator symbol is next to the operand of higher degree.

Formally speaking, the contraction $f \llcorner s$ of f on s is completely defined by the equation

$$\langle f \llcorner s, t\rangle = \langle f, st\rangle. \tag{7.8-1}$$

Intuitively, contraction is a flavor of partial evaluation. We can think of our n-form f as a function that accepts an n-site as its input and returns a real number. When we contract f on a k-site s, we are declaring that we are interested in the values of that function only on those n-sites that are multiples of s.

In the special case $k = n$, contracting an n-form f on a k-site s results in a 0-form $f \llcorner s$, that is, in a scalar. By setting t in Equation 7.8-1 to be the 0-site $t := 1$, we find that $\langle f \llcorner s, 1\rangle = \langle f, s \cdot 1\rangle = \langle f, s\rangle$. Since pairing a 0-form with a 0-site simply multiplies the two scalars, as discussed in Exercise 7.3-3, we conclude that $f \llcorner s = \langle f \llcorner s, 1\rangle = \langle f, s\rangle$. Thus, when $k = n$, contraction reduces to pairing.

It is convenient to extend the contraction operator $f \llcorner s$ to the case $k > n$ by setting $f \llcorner s = 0$. To support this, we make the convention that 0, which we have already agreed is an m-form for every nonnegative m, is also an m-form — in fact, is the unique m-form — when $m = n-k$ is negative. Extending the contraction operator in this way makes the value $f \llcorner s$ well-defined whenever f and s are homogeneous, whatever their degrees. We further extend to those cases where the arguments f and s are inhomogeneous in the unique way that preserves linearity. Having done so, the site $f \llcorner s$ is now well-defined for any form f and any site s — even inhomogeneous ones.

Successive contractions commute with each other. Indeed, we have the identity $(f \llcorner s) \llcorner t = (f \llcorner t) \llcorner s = f \llcorner (st)$. When f, s, and t, are all homogeneous, this follows because all three expressions denote the unique form of degree $\deg(f) - \deg(s) - \deg(t)$ that, when paired with any site u of that degree, returns the real number $\langle f, stu\rangle$. When f, s, or t are inhomogeneous, the result follows by linearity.

Just as we can contract a form on a site, we can contract a site on a form. If s is an m-site and g is a k-form, the expression $g \lrcorner s$ denotes an $(m-k)$-site called the *g-contraction of s* or the *contraction of s on g*. It is the unique $(m-k)$-site that makes $\langle f, g \lrcorner s\rangle = \langle fg, s\rangle$, for all $(m-k)$-forms f on A. We extend this dual contraction operator also to return zero when $k > m$, and we further extend it by linearity to the inhomogeneous case.

If f is an n-form and s is an m-site, don't get the two contractions $f \llcorner s$ and $f \lrcorner s$ confused. The operator with its vertical bar on the left, the form side, produces an $(n - m)$-form, while the one with its vertical bar on the right, the site side, produces an $(m-n)$-site. If n and m are distinct, at least one of the two results will have negative degree and hence will perforce be zero. If $n = m$, we have $f \llcorner s = f \lrcorner s = \langle f, s\rangle$.

7.9 Differentiation as contraction

We are interested in contractions primarily because they give us a more concise way to write down the rule for how to differentiate in the paired algebras. In particular, Formula 7.7-2 tells us that

$$D_p f(r) = \langle f, pr^{n-1}/(n-1)!\,\rangle = \langle f \llcorner p, r^{n-1}/(n-1)!\,\rangle.$$

Since this holds for all anchors r, we conclude that the $(n-1)$-form $D_p f$ coincides with the contraction $f \llcorner p$; that is, we have the simpler formula

$$D_p f = f \llcorner p.$$

Rephrasing that in English, we finally have a rule for differentiation that is worthy to stand alongside our evaluation rule:

> To differentiate under the summed pairing, simply contract.

This rule is deliciously simple; in particular, note that the degree of the form being differentiated is irrelevant. Indeed, the formula $D_p f = f \llcorner p$ holds even for forms f that are inhomogeneous. That delicious simplicity is the reward that we have earned by tolerating the annoying factor of $n!$ in our evaluation rule. The rules for evaluation and differentiation under the averaged pairing are different, as discussed in Appendix B.

Contracting on an anchor p, even one that isn't a vector, corresponds to differentiating in the direction p. Thus, all of the standard formulas for differentiation carry over, including the product rule,

$$(fg) \llcorner p = (f \llcorner p)g + f(g \llcorner p),$$

and the rule for perfect powers,

$$f^k \llcorner p = kf^{k-1}(f \llcorner p).$$

These identities hold for any forms f and g, regardless of their degrees, and without even any requirement of homogeneity. But it is critical that p be an anchor, that is, a 1-site. Contracting on a 0-site, that is, on a real number b, is simply scalar multiplication; so we get the simpler rules $(fg) \llcorner b = f(g \llcorner b) = (f \llcorner b)g = bfg$ and $f^k \llcorner b = bf^k$. Contracting on m-sites for $m > 1$ is more complicated, like differentiating m times; for example, if f and g are forms and p and q are anchors, we can calculate that

$$fg \llcorner pq = (f \llcorner pq)g + (f \llcorner p)(g \llcorner q) + (f \llcorner q)(g \llcorner p) + f(g \llcorner pq).$$

7.10 Derivations and differential operators

Let $G = \bigoplus_{n\geq 0} G_n$ be any graded algebra. A linear map $\delta\colon G \to G$ is called a *derivation* when it satisfies the product rule

$$\delta(xy) = \delta(x)y + x\delta(y),$$

for all x and y in G. A derivation δ is said to *lower degree by* 1 when $\delta(G_n) \subseteq G_{n-1}$, that is, when δ maps every element x that is homogeneous of degree n to an element $\delta(x)$ that is homogeneous of degree $n-1$.

We have been studying the algebra of forms $\mathrm{Sym}(\hat{A}^*)$, which is a graded algebra. For any anchor p over A, let $\delta_p\colon \mathrm{Sym}(\hat{A}^*) \to \mathrm{Sym}(\hat{A}^*)$ be the map that contracts on p, so that $\delta_p(f) := f \llcorner p$. For any fixed anchor p over A, the map δ_p is a derivation that lowers degree by 1.

It turns out that every derivation $\delta\colon \mathrm{Sym}(\hat{A}^*) \to \mathrm{Sym}(\hat{A}^*)$ that lowers degree by 1 is of the form $\delta = \delta_p$, for some anchor p over A. Here is why. Since δ lowers degree by 1, δ must map 1-forms to real numbers; so δ restricts to a linear functional on coanchors. But every such linear functional corresponds to pairing with some anchor. So there exists some anchor p with $\delta(h) = \langle h, p\rangle$, for all coanchors h. Rephrasing this, we have $\delta(f) = f \llcorner p = \delta_p(f)$ for every 1-form f on A. We also have $\delta(1) = 1 \llcorner p = \delta_p(1) = 0$, since the only way that δ can lower the degree of the 0-form 1 is by taking it to 0, the unique (-1)-form. (See also Exercise 7.10-1.) But every form f on A can be written as a linear combination of products of zero or more coanchors. The derivations δ and δ_p agree on the empty product 1, they agree on all coanchors, they are both linear, and they both satisfy the product rule; so we can conclude that $\delta(f) = \delta_p(f)$ for all forms f.

Exercise 7.10-1 Let $\delta\colon G \to G$ be any derivation of an algebra G. Without any assumption about what the derivation δ does to degrees, show that $\delta(1) = 0$. (Hint: Substitute $x := y := 1$ in the product rule.)

Exercise 7.10-2 Define a map $\delta\colon \mathrm{Sym}(\hat{A}^*) \to \mathrm{Sym}(\hat{A}^*)$ by setting $\delta(f) := nf$, for every n-form f. Show that δ is a derivation of the algebra of forms that leaves degree unchanged.

Answer: For an n-form f and an m-form g, we have $\delta(fg) = (n+m)fg = nfg + mfg = \delta(f)g + f\delta(g)$.

So every derivation of the algebra of forms that lowers degree by 1 simply contracts on some anchor. Those derivations have the additional pleasant property that they all commute with each other. For any anchors p and q and any form f, we have $(f \llcorner p) \llcorner q = (f \llcorner q) \llcorner p = f \llcorner (pq)$; so we have $\delta_p \circ \delta_q = \delta_q \circ \delta_p$. Indeed, the differential operators D_p and D_q actually commute with each other more generally; we have $D_p(D_q(f)) = D_q(D_p(f))$, not just for the functions f in $\mathrm{Poly}(\hat{A}, \mathbf{R})$, but at least for all real-valued functions $f\colon \hat{A} \to \mathbf{R}$ that are twice continuously differentiable.

Since the derivations δ_p, for anchors p in $\hat{A}$, commute with each other, we can use them to build up a commutative algebra: the algebra of all differential operators that can be expressed as polynomials in the derivations $(\delta_p)_{p\in\hat{A}}$. Note that two different polynomials in the variables $(\delta_p)_{p\in\hat{A}}$ may denote the same operator. For example, if $E := (Q + R + S)/3$ is the centroid of a reference triangle $\triangle QRS$ in A, then $3\delta_E = \delta_{3E} = \delta_Q + \delta_R + \delta_S$. Sound familiar? Indeed, this algebra of differential operators is simply the algebra of sites in disguise. Any site s on A gives us such a differential operator by contraction, by the rule $f \mapsto f \llcorner s$.

Thus, if we already understand the algebra of forms, one way to construct the algebra of sites is as a certain algebra of differential operators on forms. For example, suppose that A is an affine plane. Working in Cartesian coordinates, we could define an n-site over A to be a polynomial that is homogeneous of degree n, not in the three anchors (C, φ, ψ), but in the three derivations $(\partial/\partial w, \partial/\partial u, \partial/\partial v)$. We would then pair an n-site s with an n-form f by applying, to the form f, the differential operator that s denotes.

I've never seen anyone do so, but it would make equal sense to treat sites as basic and to define forms as certain differential operators on sites, replacing the three coanchors (w, u, v) by the derivations $(\partial/\partial C, \partial/\partial\varphi, \partial/\partial\psi)$.

People who define sites to be differential operators on forms get the right answers, but they obscure the fundamental symmetry between forms and sites. Suppose that we have somehow defined the algebra of forms $\mathrm{Sym}(\hat{A}^*)$. Whatever technique we used to algebrize the linear space $\hat{A}^*$ of coanchors would surely work, equally well, to algebrize the space $\hat{A}$ of anchors, thus producing the algebra of sites $\mathrm{Sym}(\hat{A})$. It seems more natural to produce forms and sites via the same technology, rather than to exploit differential operators to define one of them in terms of the other.

It also seems strange, when talking about differential operators, to restrict ourselves to operators that are polynomials in the three derivations $\partial/\partial w$, $\partial/\partial u$, and $\partial/\partial v$. Typically, when defining differential operators, we also allow multiplying by w, u, or v; for example, $u(\partial/\partial u)$ is a common differential operator that preserves degree. Of course, differential operators of this more general type typically don't commute; for example, the operator $u(\partial/\partial u)$ first partials with respect to u and then multiplies by u, not the reverse.

Exercise 7.10-3 People who define sites to be differential operators on forms are naturally led to one of the two possible pairings. Which one is it, the summed pairing or the averaged pairing?

Answer: The summed pairing. For example, they compute the real number $\langle w^n, C^n\rangle = \langle w^n, (\partial/\partial w)^n\rangle$ by applying the operator $(\partial/\partial w)^n$ to the n-form w^n, getting $(\partial/\partial w)^n(w^n) = n!$, rather than 1.

Math remark: The derivations that we have defined, maps from an algebra to itself, are a special case. There is a more general notion of a derivation as a linear map $\delta\colon G \to H$ that satisfies the product rule, where G is a commutative algebra and H is a G-module. Generalizing the notion of a derivation in this way lets us construct, for any commutative algebra, a universal derivation of that algebra. For a concrete example, suppose that $G = \mathrm{Sym}(\hat{A}^*) = \mathbf{R}[w, u, v]$ is a polynomial algebra in three variables. The universal derivation of G is the map $d\colon G \to H$ defined by

$$d(f) := \frac{\partial f}{\partial w}dw + \frac{\partial f}{\partial u}du + \frac{\partial f}{\partial v}dv,$$

where H is the free G-module with basis (dw, du, dv). Any derivation of G can then be achieved by substituting appropriate values for the three symbols dw, du, and dv. For example, if we substitute scalars p_w, p_u, and p_v for dw, du, and dv, we get a derivation $\delta\colon G \to G$ that lowers degree by 1; in fact, we get the derivation δ_p associated with the anchor $p = p_wC + p_u\varphi + p_v\psi = p_w(\partial/\partial w) + p_u(\partial/\partial u) + p_v(\partial/\partial v)$. For another example, if we substitute w, u, and v for dw, du, and dv, we get $w(\partial/\partial w) + u(\partial/\partial u) + v(\partial/\partial v)$, the degree-preserving derivation of Exercise 7.10-2.

7.11 Agreement to k^{th} order

An n-form f on A can be evaluated, not only at points in A, but at any anchor over A. As a consequence, f can also be differentiated, not only in the directions of vectors over A, but in the direction of any anchor over A. Evaluating a form at arbitrary anchors doesn't lead to confusion. But differentiating a form in the directions of anchors that aren't vectors leads to a subtlety that is worth discussing.

By the way, these generalized flavors of evaluation and differentiation became available to us as soon as we homogenized. We started out, in the nested-spaces framework, with a polynomial function $f\colon A \to \mathbf{R}$ of degree at most n. In converting to the homogenized approach, we linearized the domain space A into $\hat{A}$ and we homogenized f into the n-form $f\colon \hat{A} \to \mathbf{R}$. Already at this point, it started making sense to use arbitrary anchors in evaluation, and hence also in differentiation. Thus, the subtlety that this section discusses has nothing to do with the algebra of sites.

The subtlety involves the naive concept of "all possible derivatives". In some cases, what this turns out to mean, precisely, is the derivatives in all possible directions that are vectors — but not the derivatives in directions that are anchors of nonzero weight.

For example, consider the notion of "agreement to k^{th} order". Two smooth, real-valued functions f and g defined on A are said to *agree to*

k^{th} *order* at a point P in A when

$$D_{\pi_1} \cdots D_{\pi_j} f(P) = D_{\pi_1} \cdots D_{\pi_j} g(P),$$

for all j in $[0\,..\,k]$ and all vectors π_1 through π_j over A. That is, all derivatives of f and g of order at most k agree at P.

Suppose now that f is given by a polynomial of degree at most n, and let that same symbol f denote the resulting n-form; and similarly for g, an m-form. The identity above then reduces to the identity

$$\langle f, \pi_1 \cdots \pi_j P^{(n-j)}/(n-j)!\,\rangle = \langle g, \pi_1 \cdots \pi_j P^{(m-j)}/(m-j)!\,\rangle.$$

Might this identity hold with the vectors π_1 through π_j generalized to become arbitrary anchors?

If $n = m$, then that generalized identity does hold. A more concise way to phrase the situation is as follows: Two n-forms f and g agree to k^{th} order at P just when $\langle f, s\rangle = \langle g, s\rangle$ for all n-sites s that are multiples of P^{n-k}, as we essentially saw in Proposition 6.7-2.

If $k = 0$ and hence $j = 0$, the generalized identity holds trivially, since there are no parameters π_i to remove restrictions from.

But, if n and m are distinct and $k \geq 1$, there is no hope. Substituting $\pi_1 := \cdots := \pi_j := P$, we find that we must have $\langle f, P^n/(n-j)!\,\rangle = \langle g, P^m/(m-j)!\,\rangle$ for all j from 0 to k, and that is possible only if both f and g are zero to k^{th} order at P. Thus, when we require two forms of differing degrees to agree to some order at some point, we must restrict the directions of differentiation (π_i) to be vectors.

Suppose that f is a fixed n-form and that we want to determine g to be the unique k-form that agrees with f to k^{th} order at P. How do we construct that unique g via the paired algebras? We must arrange that

$$\langle f, \pi_1 \cdots \pi_j P^{(n-j)}/(n-j)!\,\rangle = \langle g, \pi_1 \cdots \pi_j P^{(k-j)}/(k-j)!\,\rangle, \tag{7.11-1}$$

for all j in $[0\,..\,k]$ and all vectors π_1 through π_j over A. Let $(P, \varphi_1, \ldots, \varphi_d)$ be some Cartesian reference frame for the affine d-space A that uses the point P as the center of its coordinate system. Every k-site over A can be expanded as a linear combination of monomials of the form $P^{k-j}\varphi^\alpha$, where j is in $[0\,..\,k]$ and $\alpha = (\alpha_1, \ldots, \alpha_d)$ is a multi-index with $|\alpha| := \alpha_1 + \cdots + \alpha_d = j$. For any monomial k-site of the form $P^{k-j}\varphi^\alpha$, Equation 7.11-1 tells us the value that we must assign to $\langle g, P^{k-j}\varphi^\alpha\rangle$. Assigning arbitrary values to those pairings determines a unique k-form g, since those monomials form a basis for the space $\mathrm{Sym}_k(\hat{A})$ of k-sites. And the k-form g that is so determined will, in fact, satisfy Equation 7.11-1 for all vectors π_1 through π_j, since each π_i is a linear combination of $(\varphi_1, \ldots, \varphi_d)$.

In Section 8.4.2, we shall analyze a differencing algorithm for computing n^{th} derivatives of an n-form, when that n-form is given to us by its values at the points of an evenly n-divided d-simplex. That algorithm is another example where we must restrict the directions of differentiation to be vectors.

Chapter 8

Exploiting the Pairing

We have built the algebra of sites, in parallel with the algebra of forms; and we have chosen, for each n, a pairing map between n-sites and n-forms. So each dual functional on n-forms is now represented, for us, by an n-site. Symmetrically, each dual functional on n-sites is represented by an n-form. In this chapter, we study several ways in which those representations clarify and simplify CAGD.

8.1 The duals of popular monomial bases

Several of the most popular bases for the linear space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms on A are monomial bases. For example, a power basis is the monomial basis associated with a Cartesian reference frame for A, while a Bernstein basis is a rescaling of the monomial basis associated with a barycentric reference frame. In the paired-algebras framework, Proposition 7.4-2 tells us that the duals of these popular bases are also rescalings of monomial bases.

8.1.1 Power-basis forms and Taylor-basis sites

Let A be an affine space of finite dimension d, and let $(C, \varphi_1, \ldots, \varphi_d)$ be a Cartesian reference frame for A. The point C together with the vectors φ_1 through φ_d form a basis $(C, \varphi_1, \ldots, \varphi_d)$ for the linearized space $\hat{A}$. Let $(w, u_1, \ldots, u_d)$ be the dual basis for $\hat{A}^*$. The monomials of total degree n in the variables $(w, u_1, \ldots, u_d)$ form the *power basis* for n-forms on A associated with this reference frame. To denote those monomials, let $\alpha := (\alpha_0, \ldots, \alpha_d)$ be a multi-index with $|\alpha| = n$, and let α_+ denote the dehomogenized multi-index $\alpha_+ := (\alpha_1, \ldots, \alpha_d)$, so that $\alpha_0 + |\alpha_+| = |\alpha| = n$. The power basis consists of the n-forms $(w^{\alpha_0}\, u^{\alpha_+})_{|\alpha|=n}$.

We now apply Proposition 7.4-2. Since we have adopted the summed pairing, we conclude that the dual basis for n-sites is the rescaled monomial basis $(C^{\alpha_0}\varphi^{\alpha_+}/\alpha!)_{|\alpha|=n}$. We shall refer to this basis as the *Taylor basis* for n-sites associated with the reference frame $(C,\varphi_1,\ldots,\varphi_d)$, since pairing an n-form with the n-sites in this Taylor basis $(C^{\alpha_0}\varphi^{\alpha_+}/\alpha!)_{|\alpha|=n}$ precisely corresponds to expanding that n-form in a Taylor series around C.

Take the case $d=2$ and $n=3$, for a concrete example. Here, listed on successive lines, are the power-basis cubic forms, the Taylor-basis cubic sites, and the results of pairing those cubic sites with an arbitrary cubic form f:

w^3	w^2u	w^2v	wu^2	wuv	wv^2	u^3	u^2v	uv^2	v^3
$\frac{C^3}{6}$	$\frac{C^2\varphi}{2}$	$\frac{C^2\psi}{2}$	$\frac{C\varphi^2}{2}$	$C\varphi\psi$	$\frac{C\psi^2}{2}$	$\frac{\varphi^3}{6}$	$\frac{\varphi^2\psi}{2}$	$\frac{\varphi\psi^2}{2}$	$\frac{\psi^3}{6}$
$f(C)$	$D_\varphi f(C)$	$D_\psi f(C)$	$\frac{(D_\varphi)^2 f(C)}{2}$	$D_\varphi D_\psi f(C)$	$\frac{(D_\psi)^2 f(C)}{2}$	$\frac{(D_\varphi)^3 f}{6}$	$\frac{(D_\varphi)^2 D_\psi f}{2}$	$\frac{D_\varphi (D_\psi)^2 f}{2}$	$\frac{(D_\psi)^3 f}{6}$

Note that all three of the corner sites in this example represent evaluations. The site $C^3/6=\epsilon_C$ represents evaluation at the center point C, clearly. But the site $\varphi^3/6=\epsilon_\varphi$ also represents evaluation — evaluation at the vector φ; we have $f(\varphi)=\langle f,\varphi^3/6\rangle=(D_\varphi)^3 f/6$.

8.1.2 Bernstein-basis forms and Bézier-basis sites

Let's consider Bernstein bases for n-forms next. Let $(R_0,\ldots,R_d)$ be a barycentric reference frame for the d-dimensional affine space A. And let $(r_0,\ldots,r_d)$ be the basis for $\hat{A}^*$ that is dual to the basis $(R_0,\ldots,R_d)$ for $\hat{A}$. The *Bernstein basis* for n-forms on A associated with the reference frame $(R_0,\ldots,R_d)$ (or with the reference d-simplex $[R_0,\ldots,R_d]$) consists of the n-forms $\left(\binom{n}{\alpha}r^\alpha\right)_{|\alpha|=n}$. The multinomial scaling factor $\binom{n}{\alpha}$ makes the Bernstein n-forms a partition of unity; that is, we have

$$\begin{aligned}\sum_{|\alpha|=n}\binom{n}{\alpha}r^\alpha(P) &= \sum_{|\alpha|=n}\binom{n}{\alpha}r_0(P)^{\alpha_0}\cdots r_d(P)^{\alpha_d}\\ &= (r_0(P)+\cdots+r_d(P))^n = 1^n = 1,\end{aligned}$$

for all points P in A.

What basis is dual to the Bernstein basis? In traditional approaches to CAGD, that dual basis consisted of certain dual functionals $(\rho_\alpha)_{|\alpha|=n}$. The $d+1$ functionals at the corners, from $\rho_{(n,0,\ldots,0)}$ through $\rho_{(0,\ldots,0,n)}$, were recognized as being the point evaluations ϵ_{R_0} through ϵ_{R_d}. But the remaining dual functionals were not typically viewed as having any simple form.

The paired-algebras framework lets us represent every one of those dual functionals quite simply, as a monomial in the points $(R_0,\ldots,R_d)$. By Proposition 7.4-2, the basis dual to the Bernstein basis consists of the n-sites $(R^\alpha/n!)_{|\alpha|=n}$. Note that the corner n-sites are again point evaluations, from

$\epsilon_{R_0} = R_0^n/n!$ through $\epsilon_{R_d} = R_d^n/n!$. The internal n-sites represent evaluations of the blossom; we have

$$\langle f, R^\alpha/n!\,\rangle = \tilde{f}(\underbrace{R_0, \ldots, R_0}_{\alpha_0}, \ldots, \underbrace{R_d, \ldots, R_d}_{\alpha_d}).$$

Indeed, for any n-form f, the pairing values $\langle f, R^\alpha/n!\,\rangle$ are precisely the *Bézier ordinates* of f, the coefficients that are needed to expand f as a linear combination of the Bernstein n-forms. As a result, it seems natural to refer to the basis $(R^\alpha/n!\,)_{|\alpha|=n}$ as the *Bézier basis* for n-sites that is associated with the barycentric reference frame $(R_0, \ldots, R_d)$.

8.2 The de Casteljau Algorithm

The de Casteljau Algorithm can be thought of in various ways. From a blossoming perspective, it starts with the Bézier ordinates of f, the blossom values

$$\tilde{f}(\underbrace{R_0, \ldots, R_0}_{\alpha_0}, \ldots, \underbrace{R_d, \ldots, R_d}_{\alpha_d})$$

for $|\alpha| = n$ and, by taking repeated linear combinations, it computes an arbitrary blossom value $\tilde{f}(p_1, \ldots, p_n)$, where p_1 through p_n are any anchors over A. Now that we understand about n-sites, we can avoid all mention of the n-form f as follows: The de Casteljau Algorithm computes an arbitrary real-lineal n-site $p_1 \cdots p_n/n!$ as a linear combination of the Bézier n-sites $(R^\alpha/n!\,)_{|\alpha|=n}$.

Here's how the de Casteljau Algorithm works. Let e_i, for i in $[0\,..\,d]$, denote the multi-index that has a one in the i^{th} place and zeros everywhere else, so that $\alpha = \alpha \cdot e = \alpha_0 e_0 + \cdots + \alpha_d e_d$. The de Casteljau Algorithm computes an n-site over A that we shall denote I_α, for all $|\alpha| \leq n$:

> **for** $|\alpha| = n$ **do** $I_\alpha := R^\alpha/n!$ **od**;
> **for** k **from** 1 **to** n **do**
>   **for** $|\alpha| = n - k$ **do**
> $I_\alpha := r_0(p_k) I_{\alpha+e_0} + \cdots + r_d(p_k) I_{\alpha+e_d}$
>   **od**;
> **od**;
> **output** $I_{(0,\ldots,0)} = p_1 \cdots p_n/n!$

This works because, for all α with $|\alpha| = n - k$, we have $I_\alpha = p_1 \cdots p_k R^\alpha/n!$. The first statement establishes this invariant for $k = 0$. To analyze the assignment in the inner loop inductively, consider the barycentric expansion $p = r_0(p)R_0 + \ldots r_d(p)R_d$, which is valid for all anchors p over A. Setting $p := p_k$ in that expansion and then multiplying by $p_1 \cdots p_{k-1} R_\alpha/n!$ shows that I_α is set correctly.

8.3 Degree raising

Given the Bézier points of a polynomial curve or surface of degree at most n, there are well-known rules for computing the Bézier points of that same curve or surface, viewed as being of degree at most $n+1$. The process is called *degree raising* (a.k.a. *degree elevation*). As an exercise in the use of the paired algebras, let's rederive those well-known rules.

Each coordinate of the original curve or surface is a real-valued function on the parameter space A, and homogenization converts each of them into an n-form on A. Let f be one of those n-forms. If we homogenized up to degree $n+1$, rather than to degree n, the result would be wf, where w is the weight coanchor on A. This suggests that degree raising corresponds to multiplication by w. Indeed, since any point P in A has $w(P) = 1$, we have $(wf)(P) = w(P)f(P) = f(P)$; so the n-form f and the $(n+1)$-form wf agree at all points. Note that they don't agree at arbitrary anchors, however.

So our task is to compute the Bézier ordinates of wf in terms of those of f. Letting $[R_0, \ldots, R_d]$ be a reference d-simplex in A, the Bézier ordinates of f are the values $\langle f, R^\alpha/n!\,\rangle$, where α varies over all multi-indices with $|\alpha| = n$. Similarly, the Bézier ordinates of wf are the values $\langle wf, R^\beta/(n+1)!\,\rangle$, for $|\beta| = n+1$. We can compute the latter in terms of the former by using the less popular contraction operator "$\lrcorner$", the one that contracts a site on a form to produce a site of lower degree.

From the definition of contraction, we have

$$\langle wf, R^\beta/(n+1)!\,\rangle = \langle f, w \lrcorner R^\beta/(n+1)!\,\rangle.$$

Contracting on a coanchor obeys the product rule, just like contracting on an anchor, which is differentiation. Since $w(R_0) = \cdots = w(R_d) = 1$, we have $w \lrcorner R^\beta = \beta_0 R^{\beta-e_0} + \cdots + \beta_d R^{\beta-e_d}$. Thus, for any β with $|\beta| = n+1$, we get the familiar formula for degree raising

(8.3-1)

$$\begin{aligned}\langle wf, R^\beta/(n+1)!\,\rangle &= \langle f, w \lrcorner R^\beta/(n+1)!\,\rangle \\ &= \langle f, (\beta_0 R^{\beta-e_0} + \cdots + \beta_d R^{\beta-e_d})/(n+1)!\,\rangle \\ &= \frac{\beta_0}{n+1}\langle f, R^{\beta-e_0}/n!\,\rangle + \cdots + \frac{\beta_d}{n+1}\langle f, R^{\beta-e_d}/n!\,\rangle.\end{aligned}$$

If $\beta_i = 0$ for some i, then the exponent $\beta - e_i$ will have a negative entry, which might seem like trouble; while the paired-algebras framework does allow us to multiply by points, it doesn't allow us to divide by them. (Though it could; see Section 8.5.) But any such troublesome term is multiplied by $\beta_i = 0$, and hence drops out of the sum. This is just like differentiating the constant 1 via the power-law, where we have $D_\pi 1 = D_\pi g^0 = 0\, g^{-1} D_\pi g = 0$ for any g, whether or not g is invertible.

Formula 8.3-1 expresses the β^{th} Bézier ordinate of wf as an affine combination of $d+1$ of the Bézier ordinates of f. This process actually has nothing to do with the particular n-form f. Here is that same formula, rewritten to express the w-contraction of the Bézier $(n+1)$-site $R^\beta/(n+1)!$ as an affine combination of $d+1$ Bézier n-sites:

$$w \lrcorner \frac{R^\beta}{(n+1)!} = \frac{\beta_0}{n+1}\left(\frac{R^{\beta-e_0}}{n!}\right) + \cdots + \frac{\beta_d}{n+1}\left(\frac{R^{\beta-e_d}}{n!}\right).$$

8.4 The geometry of point evaluations

Representing dual functionals on n-forms as n-sites is helpful also when studying those dual functionals that evaluate at points. Let ϵ_P be the dual functional that evaluates n-forms at the point P, so that $\epsilon_P(f) := f(P)$. In traditional approaches to CAGD, this functional is just one element of a space of linear functionals — rather abstract and disembodied. But we have forged the connection $f(P) = \langle f, P^n/n!\,\rangle$; so the functional ϵ_P is represented, in the paired-algebras framework, by the n-site $\epsilon_P = P^n/n!$. Why is such a concrete, algebraic formula possible and what does it buy us?

Such a formula is possible because the Veronese map $\theta_{d,n}$, the map that takes each point P in the d-space A to its n^{th} power $\theta_{d,n}(P) := P^n$, turns out to encapsulate precisely the nonlinear stuff that has to happen as part of evaluating an n-form. Evaluating an n-form f at a point P is not a linear process; while the value $f(P)$ is a linear function of f, it is a nonlinear function of P. But — and here is the key insight on which the paired-algebras framework is built — the value $f(P)$ varies linearly as a function of P^n. We can evaluate any n-form f at the point P by plugging f and P^n into the bilinear map $(f, s) \mapsto \langle f, s/n!\,\rangle$. Thus, raising P to the n^{th} power precomputes precisely enough information about P so that the subsequent evaluation of any n-form at P is a linear process.

Math remark: Raising P to the n^{th} power is linearly equivalent to evaluating n-forms at P only over fields of characteristic zero. Over a field of prime characteristic, raising P to the n^{th} power may not give us enough information to evaluate an arbitrary n-form at P. The problem shows up already for functions of a single variable, say $f(\bar{t}\,) = f(C + t\varphi)$. Raising $\bar{t}$ to the n^{th} power means computing the coefficients in the binomial expansion

$$\bar{t}^n = (C + t\varphi)^n = \sum_{0 \le i \le n} \binom{n}{i} t^i\, C^{n-i} \varphi^i.$$

In characteristic zero, knowing the values $(\binom{n}{i}\, t^i)_{0 \le i \le n}$ is linearly equivalent to knowing the values $(t^i)_{0 \le i \le n}$, the latter being what we need to evaluate any n-form f at $\bar{t}$. But in prime characteristic, we may have $\binom{n}{i} = 0$ for some i in the range $0 < i < n$, so we may not know enough. The fact that the evaluate-at-P functional ϵ_P is represented by the n-site $P^n/n!$ serves as a warning that we would be in trouble were $n!$ to be 0.

Once we have the formula $\epsilon_P = P^n/n!$, we can use that correspondence as a source of insight into the geometry of the point-evaluation functionals. Indeed, given any points P_1 through P_m in A, the geometric relationships that hold among the functionals ϵ_{P_i} are precisely those that hold among the corresponding n-sites $P_i^n/n!$. And since most geometric relationships aren't affected by a uniform scaling, we can often drop the annoying $n!$ and consider simply the n^{th} powers P_1^n through P_m^n.

For example, let $d := \dim(A)$ and let $m := \binom{n+d}{n}$ denote the dimension of the space $\mathrm{Sym}_n(\hat{A}^*)$ of n-forms on A. An n-form on A then involves m degrees of freedom. So we might hope that we could specify an n-form f on A by requiring that f interpolate arbitrary specified values at m fixed points, say $(P_1, \ldots, P_m)$. Whether that scheme works or not depends upon the geometric structure of the points (P_i). The points $(P_1, \ldots, P_m)$ are called *good for interpolation by d-variate n-ics* when specifying arbitrary real values for $f(P_1)$ through $f(P_m)$ determines a unique n-form f. (Since the n-form f has been homogenized, it is actually a polynomial in $d + 1$ variables; but it is still referred to as d-variate.) In the univariate case $d = 1$, the points $(P_1, \ldots, P_{n+1})$ are good for interpolation by n-ics whenever they are distinct, as follows from the Vandermonde determinant. But the multivariate case is more subtle. Note, for example, that we certainly can't allow more than $n+1$ of our $m = \binom{n+d}{n}$ points (P_i) to be collinear, since the n-form restricted to that line is a univariate n-ic.

From basic linear algebra, the points $(P_1, \ldots, P_m)$ will be good for interpolation by n-ics just when the point-evaluation functionals $(\epsilon_{P_1}, \ldots, \epsilon_{P_m})$ are linearly independent, hence forming a basis for the dual space $\mathrm{Sym}_n(\hat{A}^*)^*$. Now that we have the correspondence $\epsilon_P = P^n/n!$, we can make this criterion more primal and concrete: The points $(P_1, \ldots, P_m)$ are good for interpolation by n-ics just when the n-sites $(P_1^n, \ldots, P_m^n)$ are linearly independent, hence forming a basis for the space $\mathrm{Sym}_n(\hat{A})$ of n-sites over A.

8.4.1 Evenly n-divided d-simplices

One standard example of a configuration of points that is good for interpolation by d-variate n-ics is the vertices of an evenly n-divided d-simplex. In this section, we introduce that configuration of points. For completeness, we also verify that those points are indeed good for interpolation by n-ics. Figure 8.1 shows an evenly 4-divided 2-simplex.

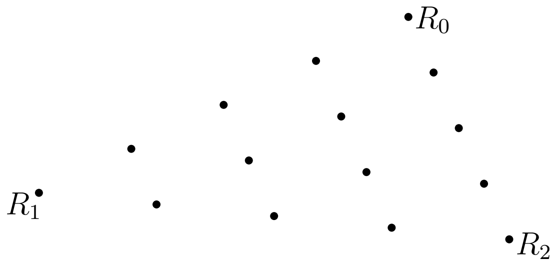


Figure 8.1: An evenly 4-divided 2-simplex

Let $(R_0, \ldots, R_d)$ be a barycentric reference frame for a d-dimensional affine space A. The points R_0 through R_d are then the vertices of a d-simplex $[R_0, \ldots, R_d]$ in A. We want to subdivide that simplex evenly into subsimplices whose linear dimensions are n times smaller. To do that, we let $\beta \cdot R$ denote the linear combination $\beta \cdot R := \beta_0 R_0 + \cdots + \beta_d R_d$, where $\beta = (\beta_0, \ldots, \beta_d)$ is a multi-index. Since R_0 through R_d are all points, the dot product $\beta \cdot R$ is an anchor over A of weight $|\beta|$. Consider the set of points $(\beta \cdot R/n)_{|\beta|=n}$. We shall say that those $m := \binom{n+d}{n}$ points *evenly n-divide* the d-simplex $[R_0, \ldots, R_d]$. Note that the edge from R_i to R_j is divided into n segments of equal length by the points

$$R_i, \frac{(n-1)R_i + R_j}{n}, \frac{(n-2)R_i + 2R_j}{n}, \ldots, R_j.$$

By the way, we must restrict n to be positive when evenly n-dividing a d-simplex, since we are dividing by n to get points. The theory would perhaps be cleaner if we dealt directly with the weight-n anchors $(\beta \cdot R)_{|\beta|=n}$. We might refer to those anchors as *evenly n-replicating* the d-simplex. An evenly 0-replicated d-simplex would consist of the single anchor 0, for any dimension d (vacant remark: including $d = -1$).

Proposition 8.4-1 *Let $(R_0, \ldots, R_d)$ be a barycentric reference frame for the d-dimensional affine space A. For any positive n, the points $(\beta \cdot R/n)_{|\beta|=n}$ that result from evenly n-dividing the d-simplex $[R_0, \ldots, R_d]$ are good for interpolation by n-ics on A.*

Proof We must show that the evaluation functionals $(\epsilon_{\beta \cdot R/n})_{|\beta|=n}$ form a basis for the dual space $\mathrm{Sym}_n(\hat{A}^*)^*$ or, equivalently, letting s_β denote the n-site

$$s_\beta := \frac{1}{n!}\left(\frac{\beta \cdot R}{n}\right)^n,$$

that the n-sites $(s_\beta)_{|\beta|=n}$ form a basis for the space $\mathrm{Sym}_n(\hat{A})$. We'll use the latter language for practice, even though this proof doesn't exploit the multiplication in the algebra of sites.

How can we prove that the particular family of n-sites $(s_\beta)_{|\beta|=n}$ forms a basis? Let $(f_\alpha)_{|\alpha|=n}$ be some family of n-forms on A. Letting $m := \binom{n+d}{n}$, we can construct an m-by-m matrix whose (α, β) entry is $\langle f_\alpha, s_\beta \rangle$. If that matrix is invertible, it follows that both the n-forms (f_α) and the n-sites (s_β) must constitute bases. So it suffices to construct some family of n-forms (f_α) which, with the n-sites (s_β), generate an invertible matrix.

We effect that construction in a nonsymmetric way, letting the vertex R_0 play a special role. For i from 1 to d, let $\varphi_i := (R_i - R_0)/n$ be the vector that separates two adjacent subdivision points along the edge joining R_0 to R_i. The sequence $(R_0, \varphi_1, \ldots, \varphi_d)$ constitutes a Cartesian reference frame for the space A. In that reference frame, the point $\beta \cdot R/n$ can be rewritten as $R_0 + (\beta_+ \cdot \varphi)$, where $\beta_+ := (\beta_1, \ldots, \beta_d)$ denotes the multi-index β with its zeroth component β_0 removed. Let $(w, u_1, \ldots, u_d)$ be the basis for the space $\hat{A}^*$ of coanchors that is dual to the basis $(R_0, \varphi_1, \ldots, \varphi_d)$ for $\hat{A}$. From the duality constraints, we deduce that $\langle w, R_0 + (\beta_+ \cdot \varphi)\rangle = 1$, while $\langle u_i, R_0 + (\beta_+ \cdot \varphi)\rangle = \beta_i$. It follows that, for any integer k, we have $\langle u_i - kw, R_0 + (\beta_+ \cdot \varphi)\rangle = (u_i - kw)\big(R_0 + (\beta_+ \cdot \varphi)\big) = \beta_i - k$. This value is zero, of course, precisely when $k = \beta_i$.

To generate lots of such zeros, we use falling-factorial powers. For any i in $[1\,..\,d]$ and any nonnegative k, we define the k-form $u_i^{\underline{k}}$ by

$$u_i^{\underline{k}} := u_i(u_i - w)(u_i - 2w)\cdots(u_i - (k-1)w).$$

These are analogs of the *falling-factorial powers* in combinatorics [24], but homogenized. Note that $u_i^{\underline{k}}\big(R_0 + (\beta_+ \cdot \varphi)\big)$ will be zero just when $k > \beta_i$, because of the factor of $u_i - \beta_i w$ in $u_i^{\underline{k}}$.

For any multi-index α, we denote by $u^{\underline{\alpha_+}}$ the product $u^{\underline{\alpha_+}} := u_1^{\underline{\alpha_1}} \cdots u_d^{\underline{\alpha_d}}$. We then choose our family of n-forms $(f_\alpha)_{|\alpha|=n}$ to be

$$f_\alpha := w^{\alpha_0} u^{\underline{\alpha_+}} = w^{\alpha_0} u_1^{\underline{\alpha_1}} \cdots u_d^{\underline{\alpha_d}}.$$

Pairing any n-form f with the n-site s_β corresponds to evaluating f at the subdivision point $R_0 + (\beta_+ \cdot \varphi)$. So, if $\alpha_i > \beta_i$ for any i from 1 to d, we conclude that $\langle f_\alpha, s_\beta \rangle = f_\alpha\big(R_0 + (\beta_+ \cdot \varphi)\big) = 0$. On the other hand, the diagonal entry $\langle f_\alpha, s_\alpha \rangle$ definitely won't be zero.

We now order both the rows and columns of our m-by-m matrix using any common total ordering $\prec$ of the multi-indices with the property that $\alpha_0 > \beta_0$ implies $\alpha \prec \beta$; that is, larger values of α_0 get listed first. It follows that, whenever $\alpha \succeq \beta$, we have $\alpha_0 \le \beta_0$. So either $\alpha = \beta$ or else we have $\alpha_i > \beta_i$, for some i in $[1\,..\,d]$. For example, Figure 8.2 shows the matrix that arises for bivariate cubics, under a certain ordering, using the abbreviations $(C, \varphi, \psi) := (R_0, \varphi_1, \varphi_2)$ and $(w, u, v) := (w, u_1, u_2)$. The matrix that results from any such ordering will be upper-triangular with nonzero entries on the diagonal, and will hence be invertible. $\square$

$$
\begin{array}{ll}
 & \begin{array}{cccccccccc} C^3/6 & (C{+}\varphi)^3/6 & (C{+}\psi)^3/6 & (C{+}2\varphi)^3/6 & (C{+}\varphi{+}\psi)^3/6 & (C{+}2\psi)^3/6 & (C{+}3\varphi)^3/6 & (C{+}2\varphi{+}\psi)^3/6 & (C{+}\varphi{+}2\psi)^3/6 & (C{+}3\psi)^3/6 \end{array} \\
\begin{array}{l} w^3 \\ w^2u \\ w^2v \\ wu(u-w) \\ wuv \\ wv(v-w) \\ u(u-w)(u-2w) \\ u(u-w)v \\ uv(v-w) \\ v(v-w)(v-2w) \end{array} &
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 2 & 1 & 0 & 3 & 2 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 2 & 0 & 1 & 2 & 3 \\
0 & 0 & 0 & 2 & 0 & 0 & 6 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6
\end{pmatrix}
\end{array}
$$

Figure 8.2: Proving Proposition 8.4-1 for bivariate cubics

Vacant remark: The proof of Proposition 8.4-1 used a Cartesian reference frame, so it required $d \geq 0$. But the result holds also for $d = -1$, trivially since n is positive. If we were working with anchors instead of points, the analogous result would hold as well for evenly 0-replicated (-1)-simplices, and the proof would be only slightly less trivial.

8.4.2 The Differencing Algorithm

Suppose that we have specified an n-form f on A by choosing the values of f at the points that result from evenly n-dividing the d-simplex $[R_0, \ldots, R_d]$, that is, by choosing the values $\big(f(\alpha \cdot R/n)\big)_{|\alpha|=n}$. Proposition 8.4-1 tells us that f is uniquely determined by those values, so we can compute anything that we like about f. There turns out to be a surprisingly easy way to compute n^{th} derivatives of f. That is, the computation itself is easy; we do n stages of differencing, quite similar to the n stages of linear combinations in the de Casteljau Algorithm. But the reason why this Differencing Algorithm works is more subtle. In this section, as another example of the benefits of the paired-algebras framework, we verify this Differencing Algorithm by exploiting the multiplication in the algebra of sites.

Rephrasing this without mentioning the n-form f, Proposition 8.4-1 says that the n-sites $\big((\alpha \cdot R/n)^n/n!\big)_{|\alpha|=n}$ form a basis for the linear space $\mathrm{Sym}_n(\hat{A})$ of n-sites. So we can expand any n-site as a linear combination of those basis elements. The Differencing Algorithm is a particularly simple way to expand certain n-sites: the products $\pi_1 \cdots \pi_n$ of n vectors over A.

The Differencing Algorithm is quite similar to the de Casteljau Algorithm in computational structure — surprisingly similar, given that the two algorithms take quite different inputs. The de Casteljau Algorithm starts with the sites in a barycentric monomial basis, while the Differencing Algorithm starts with the n^{th} powers of the points in an evenly n-divided d-simplex:

> **for** $|\alpha| = n$ **do** $J_\alpha := (\alpha \cdot R/n)^n/n!$ **od**;
> **for** k **from** 1 **to** n **do**
>   **for** $|\alpha| = n - k$ **do**
> $J_\alpha := n\bigl(r_0(\pi_k)J_{\alpha+e_0} + \cdots + r_d(\pi_k)J_{\alpha+e_d}\bigr)$
>   **od**;
> **od**;
> **output** $J_{(0,\ldots,0)} = \pi_1 \cdots \pi_n$

Recall that e_i, for i in $[0 \mathinner{..} d]$, denotes the multi-index that has a one in the i^{th} place and zeros elsewhere. The factor of n in the inner loop is justified as follows. The numbers $(r_0(\pi_k), \ldots, r_d(\pi_k))$ are the barycentric coordinates of the vector π_k in the reference frame $(R_0, \ldots, R_d)$. But we want to take differences with respect to one of the small simplices into which that large simplex is divided. Viewed with respect to one of those small simplices, the vector π_k looks n times longer.

We verified the de Casteljau Algorithm quite easily, but several things indicate that the Differencing Algorithm is more subtle. For one thing, the Differencing Algorithm requires π_1 through π_n to be vectors, that is, to have barycentric coordinates that sum to zero. If we try to use the Differencing Algorithm to compute an n-site $p_1 \cdots p_n$ whose factors are not vectors, it gets the wrong answer. For another thing, the factor of $n!$ that divides the input sites has mysteriously disappeared in the output site; the result is $\pi_1 \cdots \pi_n$, with no $n!$ in the denominator.

If we substitute the definitions of the sites (J_α) computed earlier into the formulas for those computed later, we can capture the correctness of the Differencing Algorithm as the following hoped-for algebraic identity:

$$\sum_{\substack{0 \le i_1 \le d \\ \vdots \quad \vdots \\ 0 \le i_n \le d}} r_{i_1}(\pi_1) r_{i_2}(\pi_2) \cdots r_{i_n}(\pi_n) \frac{(R_{i_1} + R_{i_2} + \cdots + R_{i_n})^n}{n!} = \pi_1 \cdots \pi_n.$$

Note that $(e_{i_1} + \cdots + e_{i_n}) \cdot R = R_{i_1} + \cdots + R_{i_n}$. Note also that the factors of n in the inner-loop differences, nested n levels deep, cancel against the n^n in the denominators of the input sites.

We shall generalize that identity slightly and then prove it by induction on n. The generalization reflects the fact that n^{th} differences of polynomials of degree less than n are zero:

(8.4-2)

$$\sum_{\substack{0\le i_1\le d\\ \vdots\quad \vdots\\ 0\le i_n\le d}} r_{i_1}(\pi_1)\cdots r_{i_n}(\pi_n)\frac{(R_{i_1}+\cdots+R_{i_n})^m}{n!} = \begin{cases}\pi_1\cdots\pi_n & \text{when } m=n\\ 0 & \text{when } 0\le m<n.\end{cases}$$

The base case of the induction is the trivial case $n = m = 0$. The sum on the left has $(d+1)^n = (d+1)^0 = 1$ term, that single term being $0^m/n! = 1$. The empty product on the right is also 1.

So assume that Equation 8.4-2 holds up to n and consider what happens for some m with $0 \le m \le n+1$. We start off by pulling the sum over i_{n+1} outside and then applying the Binomial Theorem:

$$\sum_{\substack{0\le i_1\le d\\ \vdots\quad \vdots\\ 0\le i_{n+1}\le d}} r_{i_1}(\pi_1)\cdots r_{i_{n+1}}(\pi_{n+1})\frac{(R_{i_1}+\cdots+R_{i_{n+1}})^m}{(n+1)!}$$

$$= \sum_{0\le i_{n+1}\le d} r_{i_{n+1}}(\pi_{n+1}) \sum_{\substack{0\le i_1\le d\\ \vdots\quad \vdots\\ 0\le i_n\le d}} r_{i_1}(\pi_1)\cdots r_{i_n}(\pi_n)\frac{((R_{i_1}+\cdots+R_{i_n})+R_{i_{n+1}})^m}{(n+1)!}$$

$$= \sum_{0\le i_{n+1}\le d} r_{i_{n+1}}(\pi_{n+1}) \sum_{0\le k\le m}\binom{m}{k}R_{i_{n+1}}^{m-k} \sum_{\substack{0\le i_1\le d\\ \vdots\quad \vdots\\ 0\le i_n\le d}} r_{i_1}(\pi_1)\cdots r_{i_n}(\pi_n)\frac{(R_{i_1}+\cdots+R_{i_n})^k}{(n+1)!}$$

The innermost of these three nested sums is zero by induction when $k < n$; it doesn't matter that the denominator is $(n+1)!$ instead of $n!$. So we can raise the lower bound in the sum on k from 0 to n. We can also lower the upper bound from m to $m-1$, as follows. When $k = m$, the factor $R_{i_{n+1}}^{m-k}$ drops out, leaving nothing that depends upon i_{n+1}. So the terms with $k = m$ contribute some constant multiple of the outer sum $r_0(\pi_{n+1})+\cdots+r_d(\pi_{n+1})$. But that sum is zero, since π_{n+1} is a vector.

Thus, we can tighten the bounds in the sum on k from $0 \le k \le m$ to $n \le k < m$. So nothing at all remains when $m < n+1$, as we had hoped. When $m = n+1$, the single value $k = n$ gives us

$$\sum_{0\le i_{n+1}\le d} r_{i_{n+1}}(\pi_{n+1})(n+1)R_{i_{n+1}} \sum_{\substack{0\le i_1\le d\\ \vdots\quad \vdots\\ 0\le i_n\le d}} r_{i_1}(\pi_1)\cdots r_{i_n}(\pi_n)\frac{(R_{i_1}+\cdots+R_{i_n})^k}{(n+1)!}.$$

The $n+1$ that came from the $\binom{m}{k}$ converts the $(n+1)!$ in the denominator to an $n!$, after which the inductive hypothesis replaces the inner sum with

$\pi_1 \cdots \pi_n$, leaving us — again as we had hoped — with

$$\Bigl(\sum_{0 \le i_{n+1} \le d} r_{i_{n+1}}(\pi_{n+1}) R_{i_{n+1}}\Bigr)\pi_1 \cdots \pi_n = \pi_1 \cdots \pi_{n+1}.$$

8.5 Integrating over a simplex

We have talked a lot about multiplying by points; what about dividing by them? Multiplying by points makes sense because we have extended the affine space A of points into the algebra $\mathrm{Sym}(\hat{A})$ of sites. If we wanted to, we could further extend that algebra into a field: the field $\mathrm{Quo}(\mathrm{Sym}(\hat{A}))$ of quotients whose numerators and denominators are sites. We would then need yet another new term; for example, we might refer to a quotient of sites over A as a *location* over A. I don't yet see many applications for locations in CAGD, so I don't yet recommend that we in CAGD take this additional step, from sites to locations. But locations do have at least one intriguing application; this section discusses a formula for integrating real-valued functions over simplices that can be expressed more simply using locations than using sites. Perhaps, when enough other applications have been discovered, it will be time to take this further (and final?) step toward allowing arithmetic on points: from points to anchors to sites to locations.

Some words about the mathematics of locations: Let $(C, \varphi_1, \ldots, \varphi_d)$ be, say, a Cartesian reference frame for the d-space A, and let $(w, u_1, \ldots, u_d)$ be the basis for the linear space $\hat{A}^*$ of coanchors that is dual to that frame. So the algebras $\mathrm{Sym}(\hat{A})$ and $\mathrm{Sym}(\hat{A}^*)$ of sites and forms are isomorphic to the polynomial algebras $\mathbf{R}[C, \varphi_1, \ldots, \varphi_d]$ and $\mathbf{R}[w, u_1, \ldots, u_d]$. Both of those algebras are free of zero divisors; that is, for any two sites s and t over A, we have $st = 0$ only when either $s = 0$ or $t = 0$, and similarly for forms on A. Thus, each of those algebras, when viewed as a ring, is an integral domain (a.k.a. is entire). So each of those algebras has a quotient field. The quotient field $\mathrm{Quo}(\mathrm{Sym}(\hat{A}^*))$ of the algebra of forms is isomorphic to the field of rational functions $\mathbf{R}(w, u_1, \ldots, u_d)$ in the $d+1$ variables w and u_1 through u_d; we allow ourselves to divide by any form that is not identically zero. Alternatively, we can exploit duality to think of $\mathrm{Quo}(\mathrm{Sym}(\hat{A}^*))$ as the field $\mathrm{Rat}(\hat{A}, \mathbf{R})$ of real-valued, rational functions on anchors. *Locations* over A are elements of the quotient field $\mathrm{Quo}(\mathrm{Sym}(\hat{A}))$ of the algebra of sites, where we allow ourselves to divide by any site that is not identically zero. The field $\mathrm{Quo}(\mathrm{Sym}(\hat{A}))$ of locations over A is isomorphic to the field of rational functions $\mathbf{R}(C, \varphi_1, \ldots, \varphi_d)$; or we can exploit duality to think of it as the field $\mathrm{Rat}(\hat{A}^*, \mathbf{R})$ of real-valued, rational functions on coanchors.

Now, about that integration formula: Let $[R_0, \ldots, R_d]$ be a reference d-simplex for the affine space A. Recall that the Bézier ordinates of an

n-form f on A are the real numbers $\langle f, R^\alpha/n!\,\rangle$ for $|\alpha| = n$, the $\binom{n+d}{n}$ scalars that result from pairing f with each of the Bézier n-sites. Given some notion of volume in the space A, say a measure μ on A, suppose that we want to integrate the n-form $f\colon A \to \mathbf{R}$ with respect to μ over the reference simplex $[R_0, \ldots, R_d]$. Lasserre and Avrachenkov [37] give a pretty formula for this integral, based on the observation that all of the Bézier ordinates contribute to the integral with equal weight. Thus, the integral is the volume of the reference simplex times the average of the Bézier ordinates:

$$\int_{[R_0,\ldots,R_d]} f(P)\, d\mu(P) = \frac{\mu([R_0, \ldots, R_d])}{\binom{n+d}{n}} \sum_{|\alpha|=n} \langle f, R^\alpha/n!\,\rangle. \tag{8.5-1}$$

We won't bother to prove Formula 8.5-1 here, our goal being instead to use locations to simplify the sum on the right-hand side. One proof starts by showing that the average of the Bézier ordinates is not affected when the degree of f is raised. After raising the degree of f quite high, the many Bézier ordinates that result closely approximate the values of f, so their average becomes essentially a Riemann sum for the integral. Farin [20] sketches that proof in the univariate case, and it works equally well in higher dimensions.

Both sides of Formula 8.5-1 are linear functions of the n-form f, so each must correspond to pairing f with some n-site. Recalling that $f(P) = \langle f, P^n/n!\,\rangle$, we have

$$\left\langle f, \frac{1}{n!}\int_{[R_0,\ldots,R_d]} P^n\, d\mu(P)\right\rangle = \left\langle f, \frac{\mu([R_0, \ldots, R_d])}{n!\,\binom{n+d}{n}} \sum_{|\alpha|=n} R^\alpha\right\rangle.$$

So Formula 8.5-1 boils down to this relationship among the n-sites over A:

$$\int_{[R_0,\ldots,R_d]} P^n\, d\mu(P) = \frac{\mu([R_0, \ldots, R_d])}{\binom{n+d}{n}} \sum_{|\alpha|=n} R^\alpha. \tag{8.5-2}$$

Consider the special case $d = 1$, where we integrate over a line segment $[R_0 \,..\, R_1]$; and let's write that segment as $[R \,..\, S]$ for simplicity. We get

$$\int_{[R..S]} P^n\, d\mu(P) = \frac{\mu([R \,..\, S])}{n+1}(R^n + R^{n-1}S + \cdots + S^n).$$

If this were elementary calculus, we wouldn't distinguish between a point on the domain line and a real number. We could then express the length $\mu([R \,..\, S])$ simply as $(S - R)$, which would help the sum on the right to collapse, leading to the elementary formula

$$\int_R^S P^n\, dP = \frac{S^{n+1} - R^{n+1}}{n+1}.$$

In the paired-algebras framework, however, that equation is nonsense, since it alleges that an n-site equals an $(n+1)$-site. To avoid such nonsense, we must distinguish between the real number $\mu([R\,.\,.\,S])$ and the vector $S-R$.

But let's suppose that we have defined locations over A, thereby making it legal to divide by nonzero sites, as well as to multiply by them. We can then achieve much the same collapsing in a legitimate manner as follows, multiplying and dividing by the nonzero vector $S-R$:

$$\begin{aligned}\int_{[R..S]} P^n\,d\mu(P) &= \frac{\mu([R\,.\,.\,S])}{n+1}\,\frac{(S-R)(R^n+R^{n-1}S+\cdots+S^n)}{S-R}\\ &= \frac{\mu([R\,.\,.\,S])}{n+1}\,\frac{S^{n+1}-R^{n+1}}{S-R}.\end{aligned}$$

What happens when the dimension d exceeds 1? Can we still use division by sites to achieve analogous collapsings? The following identity for bivariate cubics with reference triangle $\triangle QRS$ points the way:

$$\begin{array}{ccccccc} & & & Q^3 & & & \\ & & +Q^2R & & +Q^2S & & \\ & +QR^2 & & +QRS & & +QS^2 & \\ +R^3 & & +R^2S & & +RS^2 & & +S^3 \end{array} = \frac{Q^{3+2}}{(Q-R)(Q-S)} + \frac{R^{3+2}}{(R-Q)(R-S)} + \frac{S^{3+2}}{(S-Q)(S-R)}.$$

We can make the univariate case fit that pattern by a little rewriting:

$$\int_{[R..S]} P^n\,d\mu(P) = \frac{\mu([R\,.\,.\,S])}{n+1}\left(\frac{R^{n+1}}{R-S}+\frac{S^{n+1}}{S-R}\right).$$

More generally, for any dimension d and degree n, we shall prove that

$$\sum_{|\alpha|=n} R^\alpha = \sum_{0\le k\le d} \frac{R_k^{n+d}}{\prod\limits_{\substack{0\le j\le d\\ j\ne k}} (R_k-R_j)}. \tag{8.5-3}$$

Equation 8.5-3 is a algebraic identity; as we shall prove in a moment, it holds whenever the symbols R_0 through R_d denote distinct elements of some field. By substituting the right-hand side for the left in Equation 8.5-2, we get a new integration formula, using locations, that is arguably simpler:

$$\int_{[R_0,\ldots,R_d]} P^n\,d\mu(P) = \frac{\mu([R_0,\ldots,R_d])}{\binom{n+d}{n}} \sum_{0\le k\le d} \frac{R_k^{n+d}}{\prod\limits_{\substack{0\le j\le d\\ j\ne k}} (R_k-R_j)}. \tag{8.5-4}$$

The good news about this new Formula 8.5-4 is that the sum on the right-hand side has collapsed, leaving a sum with only $d+1$ terms, instead of $\binom{n+d}{n}$. But each term is more complicated; rather than summing n-sites, we are summing locations over A whose numerators are $(n+d)$-sites and whose denominators are d-sites. The bad news about Equation 8.5-4 is that I don't know how to pair an n-form with such a location. Thus, it may well be that Formula 8.5-4 is useless for actually computing integrals, serving only as an intriguing, location-based way to abbreviate Formula 8.5-2.

It remains to prove Equation 8.5-3, which we shall do by a joint induction on n and d. When $d=0$, Equation 8.5-3 reduces to the trivial identity $R_0^n = R_0^n$. When $n=0$, the left-hand sum has an empty product as its single term, so the left-hand side reduces to 1. To handle the right-hand side in the case $n=0$, suppose that we use the Lagrange Interpolation Formula to interpolate the univariate polynomial $t \mapsto t^d$ at the $d+1$ distinct real points $t := r_0$ through $t := r_d$. The Lagrange interpolant will reconstruct the polynomial t^d exactly, so we find that

$$t^d = \sum_{0\le k\le d} \frac{r_k^d \prod_{\substack{0\le j\le d\\ j\ne k}} t - r_j}{\prod_{\substack{0\le j\le d\\ j\ne k}} r_k - r_j}.$$

Extracting the coefficients of t^d from each term in this polynomial identity, we conclude that

$$1 = \sum_{0\le k\le d} \frac{r_k^d}{\prod_{\substack{0\le j\le d\\ j\ne k}} r_k - r_j}, \tag{8.5-5}$$

for all sequences $(r_0, \ldots, r_d)$ of distinct real numbers. If we multiplied this Equation 8.5-5 through by the least common denominator $\prod_{0\le i<j\le d}(r_j - r_i)$ of the terms in the sum, however, we would be left with a polynomial identity. So Equation 8.5-5 must also hold with $(r_0, \ldots, r_d)$ replaced by any $d+1$ distinct elements of any field — in particular, by the points $(R_0, \ldots, R_d)$ in the affine space A. This establishes Equation 8.5-3 in the case $n=0$.

Suppose now that both n and d are positive, and let's rewrite the left-hand side of Equation 8.5-3 in the equivalent form

$$S(d,n) := \sum_{|\alpha|=n} R^\alpha = \sum_{0\le i_1\le\cdots\le i_n\le d} R_{i_1}\cdots R_{i_n}.$$

Partitioning this sum according as $i_n = d$ or not, we have

$$S(d,n) = R_d\, S(d, n-1) + S(d-1, n).$$

Applying the inductive hypothesis in each case, we find that

$$S(d,n) = R_d \sum_{0 \le k \le d} \frac{R_k^{n-1+d}}{\prod\limits_{\substack{0 \le j \le d \\ j \ne k}} (R_k - R_j)} + \sum_{0 \le k \le d-1} \frac{R_k^{n+d-1}}{\prod\limits_{\substack{0 \le j \le d-1 \\ j \ne k}} (R_k - R_j)}.$$

We multiply both the numerator and denominator of the right-hand summand by $R_k - R_d$, to get

$$S(d,n) = R_d \sum_{0 \le k \le d} \frac{R_k^{n-1+d}}{\prod\limits_{\substack{0 \le j \le d \\ j \ne k}} (R_k - R_j)} + \sum_{0 \le k \le d-1} \frac{R_k^{n+d-1}(R_k - R_d)}{\prod\limits_{\substack{0 \le j \le d \\ j \ne k}} (R_k - R_j)}.$$

We can now raise the upper limit on k in the right-hand sum from $d-1$ to d, since $R_k - R_d = 0$ when $k = d$. We then expand the right-hand sum into two sums, the second of which cancels the left-hand sum, and we are left with

$$S(d,n) = \sum_{0 \le k \le d} \frac{R_k^{n+d}}{\prod\limits_{\substack{0 \le j \le d \\ j \ne k}} (R_k - R_j)},$$

which completes the proof by induction.

Exercise 8.5-6 Prove Equation 8.5-3 without using induction by combining the ideas in our proof of the case $n = 0$ with Leibniz's Formula [13] from the theory of divided differences.

Hint: Find the univariate polynomial of degree at most d that interpolates the polynomial $t \mapsto t^{n+d}$ at the points $t := r_0$ through $t := r_d$. Using the Lagrange Interpolation Formula as in our proof of the base case $n = 0$, show that the right-hand side of Equation 8.5-3 gives the coefficient of t^d in that interpolant. But that same coefficient is also the divided difference $[r_0, \ldots, r_d]t^{n+d}$. Show that the left-hand side of Equation 8.5-3 gives that divided difference by using Leibniz's Formula repeatedly, expanding t^{n+d} as the product of $n + d$ copies of t.

Chapter 9

Universal Mapping Conditions

People who use the homogenized framework already have some familiarity with linearization, the process that extends an affine space A into the linear space $\hat{A}$. The paired-algebras framework also relies on algebrization, the process that extends a linear space X into the symmetric algebra $\mathrm{Sym}(X)$. Both extensions can be achieved by various concrete constructions, three of which we discussed in Section 4.9 for linearization and in Section 5.1 for algebrization. In those sections, we claimed that it didn't matter which concrete construction we employed, since we can characterize our goal abstractly using a universal mapping condition. In this section, we explore universal mapping conditions at various levels of abstraction.

9.1 Linearization via a universal condition

Let's start with linearization, since it is both simpler and more familiar. Given an affine space A, can we characterize its linearization $\hat{A}$ abstractly and uniquely? The "uniquely" part turns out to be hopeless; the closest that we can come to "unique" while remaining abstract is "unique up to a unique isomorphism". Since we can't achieve absolute uniqueness, let's temporarily refrain from talking about "the linearization" and from writing $\hat{A}$. Instead, let's try to abstractly characterize what it means for some linear space X to be "a linearization" of the affine space A.

Since linearization is a process of extension, we expect a linearization X of A to include A as a subset; indeed, we naively want A to sit, inside X, as an affine hyperplane not containing the origin. Using that subset language, here is the universal mapping condition that a linearization must satisfy.

Condition 9.1-1 For any affine space A, a linear space $X \supseteq A$ is a *linearization* of A when every affine map $j\colon A \to Y$, from A to any linear space Y, extends uniquely to a linear map $f\colon X \to Y$. That is, there must exist a unique linear map $f\colon X \to Y$ that agrees with j on the subset A of X.

While we naively expect that any linearization X of the affine space A will include A as a subset, it would be technically unfortunate to require the relationship $A \subseteq X$. For example, we exploited duality in Section 4.9 to argue that the linear space $\mathrm{Aff}(A, \mathbf{R})^*$ is a linearization of A. The space $\mathrm{Aff}(A, \mathbf{R})^*$ does not include A as a subset; but it does contain the evaluate-at-P functional ϵ_P, for every point P in A, which is almost as good. To allow for these sorts of linearizations, let's stop requiring $A \subseteq X$ and instead settle for an affine map $i\colon A \to X$ that lets us view A as sitting inside of X.

Condition 9.1-2 For any affine space A, a linear space X and an affine map $i\colon A \to X$, taken together, are a *linearization* of A when, for every linear space Y and every affine map $j\colon A \to Y$, there exists a unique linear map $f\colon X \to Y$ with $f \circ i = j$.

Condition 9.1-2 abstractly captures all of the concrete properties that we want a linearization to have. Let's first convince ourselves of that intuitively by considering those properties in turn.

1. We want the affine map $i\colon A \to X$ to be injective. (Indeed, we at first built in that requirement by demanding that $A \subseteq X$.) If there were distinct points P and Q in A with $i(P) = i(Q)$, we could construct an affine map $j\colon A \to Y$ with $j(P) \neq j(Q)$, and then no map $f\colon X \to Y$ could possibly exist — linear or not — with $f \circ i = j$.

2. We don't want the image space $i(A)$, which will be an affine subspace of X, to include the origin of X. If there were any point P in A with $i(P) = 0$, we could construct an affine map $j\colon A \to Y$ with $j(P) \neq 0$, and then, since linear maps must take zero to zero, no linear map $f\colon X \to Y$ could possibly exist with $f \circ i = j$.

3. It follows from the two previous points that the dimension of the linear space X must exceed that of A. We want $\dim(X)$ to be precisely $\dim(A) + 1$, not larger. If $\dim(X)$ were larger, we could take any space Y of positive dimension and consider the zero map $j\colon A \to Y$. The condition $f \circ i = j = 0$ would require f to be zero on a subspace of X of dimension $\dim(A) + 1$, but we would be free to map the remaining dimensions of X arbitrarily; so f would not be unique.

But much more is true. Any structure that is defined by a universal mapping condition of the same flavor as Condition 9.1-2 is always uniquely determined, up to a unique isomorphism. The following proposition shows that in the particular case of Condition 9.1-2. But the structure of the argument is quite general, so the analogous result holds, equally well, for any structure that is defined by a universal mapping condition in this way.

Proposition 9.1-3 *Let A be an affine space, and, for k equal to 1 and 2, let X_k be a linear space and $i_k\colon A \to X_k$ be an affine map. If both of the pairs (X_1, i_1) and (X_2, i_2) are linearizations of A according to Condition 9.1-2, then there exists a unique isomorphism between X_1 and X_2 that makes the following diagram commute:*

$$\begin{tikzcd} & A \arrow[dl, "i_1"'] \arrow[dr, "i_2"] & \\ X_1 \arrow[rr, shift left, "h_{21}"] & & X_2 \arrow[ll, shift left, "h_{12}"] \end{tikzcd}$$

Warning: The map h_{21} in this diagram goes from X_1 to X_2, rather than the reverse; that is, the subscripts should be read from right to left. That looks weird in the diagram, but works well in a formula such as $x_2 = h_{21}(x_1)$, where x_1 belongs to X_1 and x_2 to X_2. More generally, with the subscripts in this order, the composition $h_{ij} \circ h_{kl}$ makes sense just when the adjacent subscripts coincide, when $j = k$. Section C.1 discusses the sad choices of convention that now force either our diagrams or our formulas to look wrong.

Proof We first apply the universal mapping condition for (X_1, i_1) to the pair $(Y, j) := (X_2, i_2)$. We deduce that there exists a unique linear map $h_{21}\colon X_1 \to X_2$ with $h_{21} \circ i_1 = i_2$. Symmetrically, there exists a unique linear map $h_{12}\colon X_2 \to X_1$ with $h_{12} \circ i_2 = i_1$.

It remains to verify that h_{21} is an isomorphism with h_{12} as its inverse, that is, that the compositions $h_{12} \circ h_{21}$ and $h_{21} \circ h_{12}$ are the identity maps on X_1 and X_2, respectively. To prove the first of those claims, we apply the universal condition for (X_1, i_1) to the pair $(Y, j) := (X_1, i_1)$. So there exists a unique linear map $h_{11}\colon X_1 \to X_1$ with $h_{11} \circ i_1 = i_1$. The identity map on X_1 is clearly one candidate for h_{11}; but the composition $h_{12} \circ h_{21}$ is another such candidate, since we have $h_{12} \circ h_{21} \circ i_1 = h_{12} \circ i_2 = i_1$. Since the map h_{11} is unique, the composition $h_{12} \circ h_{21}$ must be the identity on X_1. The composition $h_{21} \circ h_{12}$ must be the identity on X_2 symmetrically. □

Thus, if we define linearization using a universal mapping condition, it follows from purely formal reasoning that linearizations are unique, up to a unique isomorphism — if they exist at all. This formal reasoning leaves open the possibility that linearizations might not exist, however. To show that they do exist, we need a more concrete argument, one that appeals to the nature of affine and linear spaces.

Proposition 9.1-4 *If A is any affine space, there exists a linear space X and an affine map $i\colon A \to X$ that are a linearization of A according to Condition 9.1-2.*

Proof We need to construct a concrete linearization (X, i), by some method. The simplest method involves fixing a reference frame for A, so that's what we'll do. We don't have to worry about the resulting linearization depending, in some bad way, on which frame we happen to choose, since all linearizations of A are uniquely isomorphic.

Let $(R_0, \ldots, R_d)$ be a barycentric reference frame for A. We construct X as the linear space that has the new atoms $(x_0, \ldots, x_d)$ as a basis. We define the map $i\colon A \to X$ by setting $i(R_k) := x_k$, for k in $[0\,..\,d]$, and then extending in the unique way that makes i affine; so we have $i(t_0R_0 + \cdots + t_dR_d) := t_0x_0 + \cdots + t_dx_d$, for all real numbers t_0 through t_d with $t_0 + \cdots + t_d = 1$.

Note that the dimension of this linear space X is $d + 1$, as it should be, and that the image $i(A)$, sitting inside X, is a hyperplane not containing the origin — to wit, the hyperplane $t_0 + \cdots + t_d = 1$. Those are indications that the pair (X, i) might be a linearization of A. But the true test comes from the universal mapping condition.

To verify Condition 9.1-2, let Y be any linear space and $j\colon A \to Y$ any affine map. If some set map $f\colon X \to Y$ is to satisfy $f \circ i = j$, we must have $f(x_k) = f(i(R_k)) = j(R_k)$, for all k in $[0\,..\,d]$. If we also require that the map f be linear, then those $d+1$ conditions determine a unique f, since the atoms $(x_0, \ldots, x_d)$ are a basis of X. Finally, with f determined in this way, the two affine maps $f \circ i$ and j agree on a barycentric reference frame for A, so they agree on all of A. Thus, the concrete pair (X, i) does satisfy the universal mapping condition and is hence a linearization of A. □

So linearizations do exist; and they are automatically essentially unique, since they are defined by a universal mapping condition. It is then convenient to pretend that they are absolutely unique, introducing the notation $\hat{A}$ to denote "the linearization" of A. As we discussed in Section 4.9.5, this abuse of language is harmless as long as, whenever two different concrete linearizations of the same affine space A appear together in any argument, we use the unique isomorphism between them to identify each element of one with the corresponding element of the other. It is also convenient to pretend that A is actually a subset of the linearization $\hat{A}$, so that we don't need to write down or even to name the underlying affine map $i\colon A \to \hat{A}$.

9.2 Algebrization via a universal condition

The theory of algebrization is very similar to that of linearization, based on the following universal mapping condition.

Condition 9.2-1 Given a linear space X, a commutative algebra G and a linear map $i\colon X \to G$ are a *commutative algebrization* of X when, for every commutative algebra H and every linear map $j\colon X \to H$, there exists a unique algebra homomorphism $f\colon G \to H$ that satisfies $f \circ i = j$.

Recall that an algebra homomorphism $f\colon G \to H$ is a linear map that is also a ring homomorphism; so we have $f(x+y) = f(x) + f(y)$, $f(tx) = tf(x)$, $f(xy) = f(x)f(y)$, and $f(1) = 1$, for all elements x and y of G and all real numbers t.

Any two pairs (G_1, i_1) and (G_2, i_2) that are both commutative algebrizations of a common linear space X will be uniquely isomorphic. The formal statement of that claim and its proof are so similar to Proposition 9.1-3 that we content ourselves with drawing the relevant commutative diagram:

$$\begin{tikzcd} & X \arrow[dl, "i_1"'] \arrow[dr, "i_2"] & \\ G_1 \arrow[rr, leftrightarrow] & & G_2 \end{tikzcd}$$

While essential uniqueness comes for free, we have to do some concrete work to show that every linear space X does have a commutative algebrization. Fortunately, that work is quite easy, since we already know a lot about polynomial algebras.

Proposition 9.2-2 *If X is any linear space, there exists an algebra G and a linear map $i\colon X \to G$ that are a commutative algebrization of X, according to Condition 9.2-1. Furthermore, that algebra G has a grading in which the image $i(X)$ coincides with the first graded slice G_1.*

Proof As in Proposition 9.1-4, any concrete construction that succeeds will suffice, so we needn't be afraid to choose a basis. Let $(x_0, \ldots, x_d)$ be a basis for X, where we set $d := \dim(X) - 1$, for consistency with our intended applications to the linear spaces $\hat{A}$ and $\hat{A}^*$, where $d = \dim(A)$. Let G denote the algebra $G := \mathbf{R}[v_0, \ldots, v_d]$ of all polynomials in the $d+1$ variables v_0 through v_d. We define a linear map $i\colon X \to G$ by setting $i(x_k) := v_k$ for k in $[0\,..\,d]$ and then extending by linearity.

Note that the algebra $G = \mathbf{R}[v_0, \ldots, v_d]$ is graded by total degree. Under this grading, the first graded slice is the linear space $\mathbf{R}_1[v_0, \ldots, v_d]$, consisting of all linear combinations of the variables. That linear space coincides with the image space $i(X)$.

It remains to verify Condition 9.2-1. So, let H be any commutative algebra and let $j\colon X \to H$ be any linear map. If some set map $f\colon G \to H$ is to satisfy $f \circ i = j$, we must have $f(v_k) = f(i(x_k)) = j(x_k)$, for all k in $[0\,..\,d]$. Those conditions are just enough to determine a unique algebra homomorphism $f\colon G \to H$. To see this, consider any element y of G, so y is a polynomial in the variables $(v_0, \ldots, v_d)$. Suppose that, for each k in $[0\,..\,d]$, we substitute $j(x_k)$ for v_k in the polynomial y. The resulting expression will simplify to some element z of the algebra H. Any algebra homomorphism f that takes v_k to $j(x_k)$, for all k in $[0\,..\,d]$, must take y to z. And performing those substitutions does give an algebra homomorphism $f\colon G \to H$. Finally, since the linear maps $f \circ i$ and j agree on the basis $(x_0, \ldots, x_d)$ of X, they must agree on all of X. □

So every linear space X does have a commutative algebrization, which is automatically essentially unique. It is a convenient standard practice to pretend that this algebrization is absolutely unique. We follow that practice by talking about "the commutative algebrization" of X, which is written $\mathrm{Sym}(X)$ and called the *symmetric algebra* of X. We also pretend that X actually coincides with the first graded slice of its symmetric algebra $\mathrm{Sym}(X)$; that is, we identify X with its image $i(X) = \mathrm{Sym}_1(X)$.

In addition to the symmetric algebra $\mathrm{Sym}(X)$, which is commutative, there are several noncommutative algebrizations of a linear space X, as we discuss in the next section. But the symmetric algebra is the simplest, so we are lucky that the algebras of forms and sites are symmetric algebras.

Exercise 9.2-3 Show that the following alternative universal mapping condition also characterizes the symmetric algebra $\mathrm{Sym}(X)$.

Condition 9.2-4 Given a linear space X, a commutative graded algebra $G = \bigoplus_{n\geq 0} G_n$ and a linear map $i\colon X \to G_1$ are a *commutative algebrization* of X when, for every commutative graded algebra $H = \bigoplus_{n\geq 0} H_n$ and every linear map $j\colon X \to H_1$, there exists a unique graded-algebra homomorphism $f\colon G \to H$ with $f \circ i = j$.

9.3 Tensors

Most math texts that construct the symmetric algebra $\mathrm{Sym}(X)$ use tensors, even though we have just seen that polynomials suffice. Polynomials suffice, in fact, even for linear spaces X of infinite dimension or over arbitrary fields. So why do math texts use tensors? There are two reasons.

First, some texts want to deal with scalars that come, not from a field, but instead from some commutative ring R. The analog of a linear space, in this more general context, is called an *R-module*. If M is any R-module, it is possible to construct a commutative R-algebra $\mathrm{Sym}(M)$ that satisfies the appropriate universal mapping condition. But polynomials don't suffice to construct $\mathrm{Sym}(M)$; you need tensors. Indeed, the first step on the road to $\mathrm{Sym}(M)$ via polynomials would be to choose a basis for M; but only the nicest modules, the *free* modules, have bases. One of the reasons that linear spaces are simpler than modules is that all linear spaces are free.

To understand the other reason, we must broaden our sights to consider noncommutative algebras. There are at least four different ways to algebrize a linear space X, each characterized by its own universal mapping condition:

- The symmetric algebra $\mathrm{Sym}(X)$ is the free commutative algebra generated by X.
- The *tensor algebra* $T(X) = \bigotimes X$ is the free algebra — not required to be commutative — generated by X.
- The *alternating* (a.k.a. *skew-symmetric*, *exterior*, or *Grassmann*) algebra $\mathrm{Alt}(X) = \bigwedge X$ is the free algebra generated by X in which elements of X skew-commute.
- Finally, if X has an associated quadratic form $Q\colon X \to \mathbf{R}$, the *Clifford* algebra $\mathrm{Clif}(X)$ is the free algebra generated by X in which every element x of X satisfies $x^2 = -Q(x)$.

Some math texts use tensors to build the symmetric algebra because they need to develop the machinery of tensors anyway, in order to construct some of these other algebras.

9.3.1 The tensor algebra

Recall that we characterized the symmetric algebra abstractly using either of two universal mapping conditions, Condition 9.2-1 or 9.2-4. The tensor algebra can be characterized abstractly using either of those conditions as well, just omitting the requirement for commutativity.

To concretely construct the tensor algebra, people typically use tensor products. Recall that the *tensor product* of two linear spaces X and Y is a linear space $X \otimes Y$ of dimension $\dim(X)\dim(Y)$. The tensor product is, itself, abstractly characterized by a universal mapping property involving bilinear maps; but it would take us too far afield to review that.

Given any linear space X, let $X^{\otimes n}$ denote the tensor product

$$X^{\otimes n} := \underbrace{X \otimes \cdots \otimes X}_{n \text{ factors}}$$

of X with itself n times. We can concretely construct the *tensor algebra* $T(X)$ as the direct sum $T(X) := \bigoplus_{n\geq 0} X^{\otimes n}$. The tensor algebra is graded, but, as soon as $\dim(X)$ exceeds 1, is not commutative. In particular, if x_1 and x_2 are two elements of X that are linearly independent, then $x_1 \otimes x_1$, $x_1 \otimes x_2$, $x_2 \otimes x_1$, and $x_2 \otimes x_2$ are four linearly independent elements of $X^{\otimes 2} = X \otimes X$. If $(b_0, \ldots, b_d)$ is a basis for X, then the products $b_{i_1} \otimes \cdots \otimes b_{i_n}$ form a basis for $X^{\otimes n}$, where the subscripts i_1 through i_n vary independently from 0 to d. Thus, an arbitrary element of $X^{\otimes n}$ has an n-dimensional matrix of coefficients.

Let $Y := X^*$ be the dual space of X. The formula

$$\langle y_1 \otimes \cdots \otimes y_n,\ x_1 \otimes \cdots \otimes x_n \rangle = \langle y_1, x_1 \rangle \cdots \langle y_n, x_n \rangle,$$

determines a natural pairing map between $Y^{\otimes n} = (X^*)^{\otimes n}$ and $X^{\otimes n}$, allowing each to represent the dual of the other. That formula is analogous to the Permanent Identity, but without any summing or averaging going on.

The elements of the space $X^{\otimes n}$ are called *n-contravariant tensors* on X, while the elements of $Y^{\otimes n} = (X^*)^{\otimes n}$ are *n-covariant tensors* on X. That is, "contravariant" here means "primal", while "covariant" means "dual"; this usage arose in physics, for reasons that Dodson and Poston explain [19]. Sad to say, this usage conflicts with category theory, where "contravariant" means "arrow-reversing", while "covariant" means "arrow-preserving".

If we have already constructed the tensor algebra $T(X)$, there are several ways to construct the symmetric algebra $\mathrm{Sym}(X)$ as a by-product. One scheme realizes an element of $\mathrm{Sym}(X)$ as an equivalence class of tensors, under an equivalence relation that makes multiplication commutative. More precisely, we construct $\mathrm{Sym}(X)$ as the quotient $T(X)/I$, where I is the smallest two-sided ideal in $T(X)$ that contains $x_1 \otimes x_2 - x_2 \otimes x_1$, for all x_1 and x_2 in X. A second scheme looks, inside $T(X)$, at the subset formed by the *symmetric tensors*, the ones whose n^{th} homogeneous components have coefficient matrices that are symmetric under all permutations of their n dimensions. The second scheme constructs $\mathrm{Sym}(X)$ by equipping that set of symmetric tensors with a new, symmetrized multiplication.

Because people read about these schemes in textbooks, they sometimes end up believing that the symmetric algebra is, in some deep sense, an algebra of tensors. Indeed, I fell into this trap myself when I claimed that the algebra of sites is built using "the symmetric variant of the tensor-product construction" [42, 43]. We can build $\mathrm{Sym}(X)$ in that way, if we like; but tensors are overkill. It is more true to say that the symmetric algebra is an algebra of polynomials. (And the real truth, of course, is that the symmetric algebra is anything that satisfies the universal mapping condition.)

Lest any confusion on this point linger, keep in mind that the algebra of sites is dual to the algebra of forms. Since we don't need tensors to build the algebra of forms, we don't need them to build the algebra of sites either.

9.3.2 The alternating algebra

Let x_1 and x_2 be linearly independent elements of the linear space X. In the tensor algebra $T(X)$, the products $x_1 \otimes x_2$ and $x_2 \otimes x_1$ are linearly independent. In the symmetric algebra $\mathrm{Sym}(X)$, the products $x_1 x_2$ and

x_2x_1 are equal. In the alternating algebra $\mathrm{Alt}(X)$, we arrange that $x_1 \wedge x_2 = -(x_2 \wedge x_1)$. That is, the multiplication in the algebra $\mathrm{Alt}(X)$ skew-commutes. Of course, we can't possibly have $y \wedge z = -(z \wedge y)$ for all elements y and z of the algebra $\mathrm{Alt}(X)$; for example, multiplication by the identity commutes in any algebra: We always have $1 \wedge y = y = y \wedge 1$. Rather, a graded algebra G is called *skew-commutative* (a.k.a. *alternating*) when its multiplication satisfies

$$yz = (-1)^{\deg(y)\deg(z)} zy, \tag{9.3-1}$$

for all homogeneous elements y and z in G. So it is homogeneous elements y and z of odd degree that satisfy $yz = -zy$.

Math remark: Over a field of characteristic 2, we have $1 = -1$, so skew-commutativity as we have just defined it would reduce to commutativity. Instead, skew-commutativity is defined to require both Identity 9.3-1 and the identity $y^2 = 0$, for all homogeneous elements y of odd degree. Note that, when y is homogeneous of odd degree, Identity 9.3-1 gives us $y^2 = -y^2$, which implies $y^2 = 0$ whenever $2 \neq 0$.

To characterize the alternating algebra abstractly, we can use a universal mapping condition similar to Condition 9.2-4; we simply replace the commutative graded algebras in that condition with skew-commutative graded algebras. But an algebra has to be graded before we can require it to be skew-commutative. Therefore, the simpler Condition 9.2-1, which doesn't mention any grading, cannot be adapted to characterize the alternating algebra.

As for constructing the alternating algebra concretely, recall that there are two methods for constructing the symmetric algebra $\mathrm{Sym}(X)$ from the tensor algebra $T(X)$, one by taking a quotient, the other by extracting a subset. Both of those methods can be adjusted to produce the alternating algebra $\mathrm{Alt}(X)$ instead. The first method constructs $\mathrm{Alt}(X)$ as the quotient $T(X)/J$, where J is the smallest two-sided ideal in $T(X)$ that contains $x \otimes x$, for all x in X. The second method looks, inside $T(X)$, at the set of *skew-symmetric tensors*, where an n-dimensional matrix of coefficients is called *skew-symmetric* (a.k.a. *alternating*) when it is skew-symmetric in all $\binom{n}{2}$ pairs of dimensions. We can construct $\mathrm{Alt}(X)$ by equipping the set of skew-symmetric tensors with a new, skew-symmetrized multiplication.

Math remark: Defining skew-symmetry in characteristic 2 has the same wrinkle that arose in defining skew-commutativity. In order to be skew-symmetric in characteristic 2, a matrix (m_{ij}) must also be zero on the diagonal; that is, we must have $m_{ii} = 0$, as well as $m_{ij} = -m_{ji}$.

Let's denote the n^{th} graded slice of the alternating algebra $\mathrm{Alt}(X)$ as $\mathrm{Alt}_n(X)$; another good notation would be $X^{\wedge n}$, but we want to emphasize the analogy between $\mathrm{Alt}(X)$ and $\mathrm{Sym}(X)$. Let's also set $d := \dim(X) - 1$, since we have in mind applying this theory when X is either $\hat{A}$ or $\hat{A}^*$, with A an affine d-space. If $(b_0, \ldots, b_d)$ is a basis for X, then the products $(b_{i_1} \wedge \cdots \wedge b_{i_n})_{0 \le i_1 < \cdots < i_n \le d}$ form a basis for $\mathrm{Alt}_n(X)$. Since equal subscripts are forbidden, the linear space $\mathrm{Alt}_n(X)$ is smaller than $\mathrm{Sym}_n(X)$; in particular, we have $\dim(\mathrm{Alt}_n(X)) = \binom{d+1}{n}$, while $\dim(\mathrm{Sym}_n(X)) = \binom{d+n}{n}$. Indeed, the whole alternating algebra is finite-dimensional; we have $\dim(\mathrm{Alt}(X)) = 2^{d+1} = 2^{\dim(X)}$.

Let $Y := X^*$ denote the dual space of X. In choosing pairing maps between $\mathrm{Alt}_n(X)$ and $\mathrm{Alt}_n(Y)$, we meet once again that contentious factor of $n!$. There is a unique pairing between $\mathrm{Alt}_n(Y) = \mathrm{Alt}_n(X^*)$ and $\mathrm{Alt}_n(X)$ that satisfies the following Summed Determinant Identity:

$$\langle y_1 \wedge \cdots \wedge y_n, x_1 \wedge \cdots \wedge x_n \rangle = \sum_{\nu \in \mathbf{S}_n} \mathrm{sgn}(\nu) \prod_{1 \le k \le n} \langle y_k, x_{\nu(k)} \rangle. \tag{9.3-2}$$

The summation index ν here varies over the symmetric group $\mathbf{S}_n$ of all $n!$ permutations of the integers from 1 to n, while $\mathrm{sgn}(\nu)$, the *sign* of ν, denotes 1 if ν is an even permutation and -1 if ν is odd. That sum is precisely the determinant of the n-by-n matrix whose $(i, j)^{\text{th}}$ entry is $\langle y_i, x_j \rangle$. There is also a unique pairing that satisfies the Averaged Determinant Identity, which is the same, except divided by $n!$. As for whether summing or averaging is better in the skew-symmetric context, let's be glad that we have no current need to decide. The tradeoffs will be different than in the symmetric case. For example, note that the skew-symmetric n^{th} power

$$P^{\wedge n} := \underbrace{P \wedge \cdots \wedge P}_{n \text{ factors}}$$

of a point P is zero as soon as n exceeds 1; so who cares whether it gets divided by $n!$ or not.

When studying the alternating algebra itself, an element of the linear space $\mathrm{Alt}_n(X)$ is typically called an *n-vector* over X, while an element of $\mathrm{Alt}_n(X^*)$ is an *n-covector*. One important application of the alternating algebra is to calculus on manifolds, however, and that application has its own nomenclature. Recall that a vector field is a map that assigns, to each point in a manifold, a vector in the tangent space at that point. If we assign an n-covector on that tangent space instead, we get a field of n-covectors; such a field is, unfortunately, called a *differential n-form*. The unfortunate aspect of this term is the further overloading of the word "form". A differential n-form on a smooth manifold has nothing to do with an n-form on a linear space; indeed, the multiplications that underlie those two types of n-forms skew-commute and commute, respectively. As for why differential n-forms are the proper things to integrate over an n-manifold, the appearance of the determinant in Equation 9.3-2 provides an indication, since that determinant expresses the ratio between two measures of volume.

The alternating algebra has other important applications to geometry. In particular, a lineal n-vector $x_1 \wedge \cdots \wedge x_n$ corresponds to an oriented volume form on the subspace of X spanned by the vectors x_1 through x_n. If we identify two such volume forms that differ by a scalar multiple, we get a rule that associates, with each n-dimensional subspace of X, a line through the origin of the space $\mathrm{Alt}_n(X)$. The set of all n-dimensional subspaces of the linear space X is a *Grassmann manifold*, and this rule gives us a way to realize that manifold as a variety in a projective space — essentially the variety formed by those n-vectors over X that are lineal, that is, that can be written as the wedge product of n vectors in X. This is why an alternating algebra is also known as a Grassmann algebra. If we identify volume forms that differ by positive scalar multiples, but distinguish between those that differ by negative scalar multiples, we get an oriented version of this theory, as Stolfi explains [46]. In the oriented theory, the product $P \wedge Q$ of two points P and Q represents the oriented line from P toward Q.

9.3.3 The Clifford algebra

Some linear spaces come to us equipped with quadratic forms. For example, a Euclidean space has a positive definite inner product, while a Minkowski space has an indefinite but non-degenerate metric tensor. When the geometry associated with that quadratic form is important to us, it may help to extend that linear space into yet another algebra: a Clifford algebra. In particular, researchers in CAGD have had good success recently explaining Pythagorean-hodograph curves using Clifford algebras [10].

Some words about quadratic forms versus bilinear forms. Given any quadratic form $Q\colon X \to \mathbf{R}$ on a linear space X, there is always a bilinear form $B\colon X \times X \to \mathbf{R}$ that satisfies the identity $Q(x) = B(x, x)$. And, since the characteristic of the real numbers $\mathbf{R}$ is not 2, we can determine the bilinear form B uniquely by requiring that B be symmetric. The inner product on a Euclidean space and the metric tensor on an Minkowski space are often thought of as symmetric, bilinear forms. In defining the Clifford algebra, however, all that we use directly is the diagonal Q of B.

Roughly speaking, the Clifford algebra $\mathrm{Clif}(X)$ is the free algebra that contains X as a subspace and in which we have $x^2 = -Q(x)$, for all elements x of X. Thus, Clifford-multiplying the vector x times itself results in the particular scalar $-Q(x)$. Note that the alternating algebra is a special case of a Clifford algebra, the special case in which $Q = 0$, so $x \wedge x = -Q(x) = 0$. Indeed, the Clifford algebra always has the same dimension as the alternating algebra; but the multiplication in the Clifford algebra has, embedded inside it in some way, the geometry associated with the form Q.

Clifford algebras don't fit well into the structures of this monograph, so we won't say much about them. In particular, beware that a Clifford algebra is not graded, under our definition of that term from Section 2.4. Like the alternating algebra $\mathrm{Alt}(X)$, the Clifford algebra $\mathrm{Clif}(x)$ is the direct sum $\mathrm{Clif}(X) = \bigoplus_{0 \le n \le d+1} \mathrm{Clif}_n(X)$ of subspaces, where $\dim(\mathrm{Clif}_n(X)) = \binom{d+1}{n}$, for a total dimension of $2^{d+1} = 2^{\dim(X)}$. But the multiplication of the Clifford algebra does not send $\mathrm{Clif}_i(X) \times \mathrm{Clif}_j(X)$ into $\mathrm{Clif}_{i+j}(X)$. Instead, the Clifford product of an element of $\mathrm{Clif}_i(X)$ with an element of $\mathrm{Clif}_j(X)$ is a linear combination of elements of $\mathrm{Clif}_k(X)$, for k in $\{i+j, i+j-2, i+j-4, \ldots, |i-j|\}$. The components of the product in these different "grades" reflect different geometric relationships among the factors. For example, if x and y are vectors in a Euclidean 3-space X, the grade-2 part of the Clifford product xy corresponds to the cross product $x \times y$ (more precisely, it is an oriented volume form on the plane that x and y span), while the grade-0, scalar part of xy is minus their dot product.

We can present one possible concrete construction of the Clifford algebra, based on ideas that we've already covered. One way to build $\mathrm{Clif}(X)$ is as $T(X)/K$, where K is the the smallest two-sided ideal of the tensor algebra $T(X)$ that contains all elements of the form $(x \otimes x) + Q(x)$, for x in X. For a more concrete way to construct the Clifford algebra and for much other wisdom, read Porteous [41].

By the way, the theory of Clifford algebras involves a contentious choice, rather like our annoying factor of $n!$, except that it's an annoying factor of -1. Some authors require $x^2 = Q(x)$, rather than $x^2 = -Q(x)$. But including the minus sign ends up singling out the positive definite forms Q as being of special interest, and that seems more appealing than the alternative of singling out the negative definite ones.

Appendix A

Some Category Theory

Category theory provides a general framework for defining things by universal mapping conditions, as happens when we linearize an affine space or algebrize a linear space. In particular, linearization and algebrization turn out to be left adjoints of forgetful functors. We here review linearization and algebrization from the enlightening perspective of category theory.

A.1 Fixing some type errors

Let's start by fixing up some type errors in our discussion of linearization. Let A be an affine space. We defined a linearization of A to be a linear space X and an affine map $i\colon A \to X$ that satisfied a certain universal mapping condition, Condition 9.1-2. But it is a type error for the codomain of an affine map to be a linear space. The expression $i\colon A \to X$ makes sense only if we implicitly perform a type conversion on the codomain, replacing the linear space X by that same set of points, but viewed as an affine space. To perform that type conversion, we take the linear space X and we forget about where the origin is; let's denote the resulting affine space as $\mathcal{T}(X)$, on the grounds that forgetting the origin is like performing some arbitrary translation. Affine combinations $t_1x_1 + \cdots t_mx_m$ with $t_1 + \cdots + t_m = 1$ still make sense in the affine space $\mathcal{T}(X)$, and they have the same values that they had in the linear space X. But linear combinations with $t_1 + \cdots + t_m \neq 1$ no longer make sense in $\mathcal{T}(X)$.

If we insert the type-conversion operator $\mathcal{T}$ appropriately, we can define what it means to be a linearization in a completely type-correct manner.

Condition A.1-1 For any affine space A, a linear space X and an affine map $i\colon A \to \mathcal{T}(X)$ are a *linearization* of A when, for every linear space Y and every affine map $j\colon A \to \mathcal{T}(Y)$, there exists a unique linear map $f\colon X \to Y$ with $\mathcal{T}(f) \circ i = j$.

To make the final equation type-correct, we had to apply the type-conversion operator $\mathcal{T}$ to the linear function $f\colon X \to Y$, thereby converting it into an affine function $\mathcal{T}(f)\colon \mathcal{T}(X) \to \mathcal{T}(Y)$. The two functions f and $\mathcal{T}(f)$ are identical as set maps; that is, they take the same input points to the same output points. The difference between them is purely type-theoretic: The map f is a linear map between linear spaces, while $\mathcal{T}(f)$ is an affine map between affine spaces.

Since $\mathcal{T}$ applies both to spaces and to maps, it isn't an operator; to say what it really is, we need some category theory. A *category* C is

1. a collection of *objects*;
2. for each pair of objects X and Y, a collection of *arrows* $C(X,Y)$, referred to as *arrows from* X *to* Y;
3. for each object X, a special arrow in $C(X,X)$, called the *identity on* X; and
4. for each triple of objects X, Y, and Z in C, a composition operation $\circ\colon C(Y,Z) \times C(X,Y) \to C(X,Z)$, taking the arrows $f\colon X \to Y$ and $g\colon Y \to Z$ to an arrow $g \circ f\colon X \to Z$. (Note the clumsiness that results from composing from right-to-left, as we discuss in Section C.1; but that clumsiness is standard in category theory.)

These structures must satisfy a few axioms:

1. Identity arrows must behave as identities for the composition operator on arrows.
2. The composition on arrows must be associative.

In many important categories, the objects are spaces with some structure, while the arrows are maps between spaces that preserve this structure. For example, there is a category **Aff**, whose objects are affine spaces and whose arrows are affine maps. Similarly, there is a category **Lin** of linear spaces and linear maps, as well as a category **Set** of sets and set maps.

A *functor* is a structure-preserving map from one category to another. So a functor $\mathcal{F}$ from C to D must map each object X of C to some object $\mathcal{F}(X)$ of D and must map each arrow $f\colon X \to Y$ of C to an arrow $\mathcal{F}(f)\colon \mathcal{F}(X) \to \mathcal{F}(Y)$ of D. A functor is also required to map identity arrows to identity arrows and to commute with composition, so that $\mathcal{F}(g \circ f) = \mathcal{F}(g) \circ \mathcal{F}(f)$.

Using this language, we can recognize the operator $\mathcal{T}$ above as a functor from **Lin** to **Aff** of a particularly simple type: It converts linear spaces into affine spaces by forgetting the origin, and it converts linear maps into affine maps by forgetting the origins in both the domain and codomain. Functors of this type are called *forgetful functors*.

A.2 Linearization as a functor

Introducing the forgetful functor $\mathcal{T}$ from **Lin** to **Aff** has allowed us to fix the type errors in our definition of a linearization. The resulting concept is perfectly respectable in category theory, where an arrow $i\colon A \to \mathcal{T}(X)$ that satisfies Condition A.1-1 would be referred to *universal from A to $\mathcal{T}$*, that is, universal from the object A to the functor $\mathcal{T}$.

But there is an important property of linearization that we have not yet captured. So far, we have been treating the parameter A as a fixed affine space. That suggests that our ability to linearize A might depend upon some special property of A; perhaps some affine spaces can be linearized, but others cannot. No. One of the good things about linearization is that we can linearize any affine space. Even more, we can extend any affine map $f\colon A \to B$ between affine spaces uniquely into a linear map $\hat{f}\colon \hat{A} \to \hat{B}$ between the corresponding linearizations. Thus, the process of linearization is actually a functor in the other direction, from **Aff** back to **Lin**. Instead of writing hat accents, let's refer to this new functor as $\mathcal{L}$. So, given any affine space A, the functor $\mathcal{L}$ produces for us a linear space $\mathcal{L}(A)$; and, given any affine map $f\colon A \to B$, we get a linear map $\mathcal{L}(f)\colon \mathcal{L}(A) \to \mathcal{L}(B)$.

The universal mapping condition is now revealed to be some flavor of pseudo-inverse relationship between the two functors $\mathcal{L}$ and $\mathcal{T}$. But note that $\mathcal{L}$ is not even a one-sided inverse of $\mathcal{T}$, on either side. We can't have $\mathcal{L}(\mathcal{T}(X)) = X$ for any linear space X, nor can we have $\mathcal{T}(\mathcal{L}(A)) = A$ for any affine space A, because $\mathcal{L}$ increases the dimension by one, while $\mathcal{T}$ preserves the dimension. We also can't have $\mathcal{L}(\mathcal{T}(\mathcal{L}(A))) = \mathcal{L}(A)$ or any other such identity in which one pair of functor applications drops out. Instead, the pseudo-inverse relationship between $\mathcal{L}$ and $\mathcal{T}$ is of a different nature.

A.3 Left adjoints

Let's restate the universal mapping condition, using both the forgetful functor $\mathcal{T}$ and the linearizing functor $\mathcal{L}$ — in particular, replacing X by $\mathcal{L}(A)$:

Condition A.3-1 For every affine space A, the linear space $\mathcal{L}(A)$ and the affine map $i_A\colon A \to \mathcal{T}(\mathcal{L}(A))$ have the property that, for every linear space Y and every affine map $j\colon A \to \mathcal{T}(Y)$, there exists a unique linear map $f\colon \mathcal{L}(A) \to Y$ with $\mathcal{T}(f) \circ i_A = j$.

Rather than defining what it means to be a linearization of a particular affine space A, this condition expresses the pseudo-inverse relationship that holds between the functors $\mathcal{T}$ and $\mathcal{L}$. The key to that pseudo-inverse relationship is a certain system of one-to-one correspondences between sets of arrows.

Treating the affine space A and the linear space Y as free parameters, consider the set $\mathrm{Aff}(A, \mathcal{T}(Y))$ of all affine maps from A to $\mathcal{T}(Y)$ and the set $\mathrm{Lin}(\mathcal{L}(A), Y)$ of all linear maps from $\mathcal{L}(A)$ to Y. Condition A.3-1 gives us a map from the former to the latter; for every j in $\mathrm{Aff}(A, \mathcal{T}(Y))$, there exists a unique f in $\mathrm{Lin}(\mathcal{L}(A), Y)$. We also have a map the other way; given any f in $\mathrm{Lin}(\mathcal{L}(A), Y)$, applying the functor $\mathcal{T}$ to f and then composing with i_A on the right gives us a map $\mathcal{T}(f) \circ i_A$, which lies in $\mathrm{Aff}(A, \mathcal{T}(Y))$. The final equality in the universal condition tells us that those forward and backward maps are inverses. Thus, for every affine space A and every linear space Y, we have a one-to-one correspondence between

$$\mathrm{Aff}(A, \mathcal{T}(Y)) \longleftrightarrow_{A,Y} \mathrm{Lin}(\mathcal{L}(A), Y). \tag{A.3-2}$$

It is those correspondences that lie at the heart of the relevant pseudo-inverse relationship. The functor $\mathcal{L}$ is said to be *left adjoint* to $\mathcal{T}$ — or, equivalently, the functor $\mathcal{T}$ is *right adjoint* to $\mathcal{L}$ — when these correspondences exist, for all A and Y, and when they behave naturally with respect to affine and linear maps, as we'll discuss below. The resulting relationship is called an *adjunction*. Note that $\mathcal{L}$ is a left adjoint because it appears in a left-hand argument in Correspondence A.3-2, while $\mathcal{T}$ is a right adjoint because it appears in a right-hand argument. (The meanings of "left adjoint" and "right adjoint" thus depend, unfortunately, on our convention that functions compose from right to left. Perhaps more semantic terms, such as *source adjoint* and *target adjoint*, would be a better idea?)

What happens if we set the linear space Y, in Correspondence A.3-2, to be $\mathcal{L}(A)$? We then get the set $\mathrm{Lin}(\mathcal{L}(A), \mathcal{L}(A))$ on the right, which is interesting because that set has a distinguished element: the identity on $\mathcal{L}(A)$. The arrow on the left that corresponds to that identity on the right is a special affine map from A to $\mathcal{T}(\mathcal{L}(A))$. In fact, it is the map that we denoted i_A in Condition A.3-1; it shows how the affine space A sits, as an affine hyperplane, inside its linearization $\mathcal{L}(A)$. The maps i_A are called the *units* of the adjunction.

For completeness, we should mention that an adjunction has *counits* as well, which are linear maps in our example. To find the counits, we set A to $\mathcal{T}(Y)$ in Correspondence A.3-2 and then map the identity on the left over to the right. The result is a special linear map $c_Y : \mathcal{L}(\mathcal{T}(Y)) \to Y$. If we convert a linear space Y into an affine space by forgetting about its origin and then re-linearize it by adding a new, external origin, the counit c_Y projects out the newly added dimension, projecting along lines that are parallel to the line joining the new origin to the old. Note that the units in our example go from simple to complicated, while the counits go from complicated to simple. As for figuring out whether $\mathcal{L}$ or $\mathcal{T}$ gets applied first, that depends upon which is the left adjoint and which is the right adjoint.

A.4 Behaving naturally

Given two categories and two functors that go back and forth between them, an adjunction exists between the two functors when there is a system of one-to-one correspondences as in Correspondence A.3-2 that behaves naturally with respect to arrows in the two categories. For completeness, we sketch briefly in this section what it means to "behave naturally". For more details, see Mac Lane's fine text [40].

Let $g\colon Y \to Z$ be some arrow in the category **Lin**. The adjunction gives us the horizontal one-to-one correspondences in the following diagram:

$$\begin{tikzcd} \mathrm{Aff}(A, \mathcal{T}(Y)) \arrow[r, leftrightarrow, "{}_{A,Y}"'] \arrow[d, "\mathcal{T}(g)_*"] & \mathrm{Lin}(\mathcal{L}(A), Y) \arrow[d, "g_*"] \\ \mathrm{Aff}(A, \mathcal{T}(Z)) \arrow[r, leftrightarrow, "{}_{A,Z}"'] & \mathrm{Lin}(\mathcal{L}(A), Z) \end{tikzcd}$$

The vertical maps come from g. Given any linear map from $L(A)$ to Y, as on the top right, we can apply first that map and then g to get a linear map from $L(A)$ to Z, as on the bottom right; the vertical arrow labeled g_* represents that "follow up with g" process. The arrow on the left labeled $\mathcal{T}(g)_*$ represents the analogous "follow up with $\mathcal{T}(g)$" process. In order to behave properly with respect to linear maps, this diagram must commute.

Behaving properly with respect to affine maps is similar. Let $h\colon A \to B$ be some arrow in the category **Aff**. The adjunction gives us the horizontal correspondences in this diagram:

$$\begin{tikzcd} \mathrm{Aff}(B, \mathcal{T}(Y)) \arrow[r, leftrightarrow, "{}_{B,Y}"'] \arrow[d, "h^*"] & \mathrm{Lin}(\mathcal{L}(B), Y) \arrow[d, "\mathcal{L}(h)^*"] \\ \mathrm{Aff}(A, \mathcal{T}(Y)) \arrow[r, leftrightarrow, "{}_{A,Y}"'] & \mathrm{Lin}(\mathcal{L}(A), Y) \end{tikzcd}$$

Given any affine map from B to $\mathcal{T}(Y)$, as on the top left, we can apply first h and then that map to get a map from A to $\mathcal{T}(Y)$, as on the bottom left; the vertical arrow labeled h^* represents that "prepare with h" process. The vertical arrow labeled $\mathcal{L}(h)^*$ represents the analogous "prepare with $\mathcal{L}(h)$" process. That diagram must also commute.

A.5 A ladder of adjunctions

Enough, already, of category theory. What does category theory tell us about the process of algebrization that underlies the paired-algebras framework?

Let **CAlg** denote the category whose objects are commutative algebras and whose arrows are algebra homomorphisms. Given any commutative

$$\begin{tikzcd}
\mathbf{CAlg} \arrow[d, bend left, "\mathcal{M}"] \\
\mathbf{Lin} \arrow[u, bend left, "\mathcal{A}"] \arrow[d, bend left, "\mathcal{T}"] \\
\mathbf{Aff} \arrow[u, bend left, "\mathcal{L}"] \arrow[d, bend left, "\mathcal{U}"] \\
\mathbf{Set} \arrow[u, bend left, "\mathcal{G}"]
\end{tikzcd}$$

Figure A.1: A ladder of adjunctions

algebra, if we forget how to multiply, we are left with a linear space; thus, there is an obvious forgetful functor from **CAlg** to **Lin**. Let's denote that functor $\mathcal{M}$, on the grounds the it forgets how to multiply. The process of algebrization is precisely a functor $\mathcal{A}$ that is left adjoint to the forgetful functor $\mathcal{M}$. That is, writing $\mathcal{A}(X)$ for the symmetric algebra $\mathrm{Sym}(X)$, we have one-to-one correspondences

$$\mathrm{Lin}(X, \mathcal{M}(G)) \longleftrightarrow_{X,G} \mathrm{CAlg}(\mathcal{A}(X), G).$$

Figure A.1 shows a ladder of adjunctions of which we have explained the top two steps. On the right, we have forgetful functors going down, $\mathcal{M}\colon \mathbf{CAlg} \to \mathbf{Lin}$ and $\mathcal{T}\colon \mathbf{Lin} \to \mathbf{Aff}$. On the left, their left adjoints go back up, linearization $\mathcal{L}\colon \mathbf{Aff} \to \mathbf{Lin}$ and algebrization $\mathcal{A}\colon \mathbf{Lin} \to \mathbf{CAlg}$.

It is helpful to add on one more step at the bottom of the ladder. Let $\mathcal{U}$ denote the forgetful functor from **Aff** to **Set**, the functor that takes an affine space and forgets everything about it except for the underlying set of points — $\mathcal{U}$ for underlying. The left adjoint of $\mathcal{U}$ is the functor $\mathcal{G}$ that, given any set S, produces the "affinization" or "geometrization" of S, an affine space for which the points in S form a barycentric frame. Note that we again have one-to-one correspondences

$$\mathrm{Set}(S, \mathcal{U}(A)) \longleftrightarrow_{S,A} \mathrm{Aff}(\mathcal{G}(S), A).$$

So we have a ladder with four rungs, each adjacent pair of rungs connected by a forgetful functor, going down, and its left adjoint, going back up. Going down is always easy. Stepping up from **Set** to **Aff** is also conceptually easy, although the resulting affine spaces can be quite large. The other two upward steps, linearization and algebrization, are more subtle. But note that jumping up from **Set** to **Lin**, taking two steps at once, is easy; given a set S, the linear space $\mathcal{L}(\mathcal{G}(S))$ is simply a linear space that has S as a basis. And jumping all the way up the ladder from **Set** to **CAlg** is easy as well; given a set S, the commutative algebra $\mathcal{A}(\mathcal{L}(\mathcal{G}(S)))$ is simply the algebra $\mathbf{R}[S]$ of all polynomials whose variables lie in S. If we jump up from a set S to any rung, we simply get the free thing of that type that is generated by S: the free affine space with S as a barycentric reference frame, the free linear space with S as a basis, or the free commutative algebra with S as its generating variables. The subtlety comes only when we have already stepped up and we want to step farther up.

Indeed, recall that the easiest concrete construction that we found for linearizing an affine space A involved choosing a barycentric reference frame for A and then forming $\mathcal{L}(A)$ as the linear space with those frame points as a basis. In terms of the ladder in Figure A.1, choosing a barycentric frame for A means finding some set S with $\mathcal{G}(S) = A$. Note that this is not at all the same as computing $\mathcal{U}(A)$, since the functors $\mathcal{G}$ and $\mathcal{U}$ are adjoints, not inverses. Having found some S with $\mathcal{G}(S) = A$, we can then jump up from **Set** to **Lin** in one easy step, by forming the linear space that has S as a basis. Thus, jumping up from the ground is so easy that we use it as a subroutine when stepping up one rung.

Given a linear space X, jumping up from **Set** is also the easiest way to construct the symmetric algebra $\mathcal{A}(X) = \mathrm{Sym}(X)$. We choose some basis for X; that is, we find some set S with $X = \mathcal{L}(\mathcal{G}(S))$. We then construct $\mathcal{A}(X)$, in a big jump back up, as the polynomial algebra $\mathbf{R}[S]$.

This ability to step up by backing down to ground level and then jumping one higher is a special property of the particular ladder of adjunctions in Figure A.1. For example, we couldn't employ the same strategy to step up above **CAlg**, if we added a new rung at the top of the ladder, since not every commutative algebra is the polynomial algebra generated by some set of variables. One of the things that make **Aff** and **Lin** simple is that every object has a frame or basis; but that doesn't hold for **CAlg**.

Exercise A.5-1 In preparation for the day when locations over A might join sites over A as basic objects in CAGD, so that we can divide by points as well as multiply by them, extend the adjunction ladder of Figure A.1 with one more rung at the top.

Hint: First replace the former top rung **CAlg** by the smaller category **EAlg** of *entire algebras*, that is, commutative algebras that are free of zero divisors and that hence, when viewed as rings, are integral domains (a.k.a. entire rings). And verify that the functors $\mathcal{M}$ and $\mathcal{A}$ form an adjunction also between **Lin** and **EAlg**. We can then add, as a new top rung, the category **RFld** consisting of those fields that include the real numbers as a subfield. The forgetful functor $\mathcal{D}\colon$ **RFld** $\to$ **EAlg** forgets how to divide, while its left adjoint $\mathcal{Q}\colon$ **EAlg** $\to$ **RFld** forms quotients.

A.6 Injective units

All of the left adjoints that we have been studying (including the functor $\mathcal{Q}$ in Exercise A.5-1) have the special property that they only add in new, good stuff; they don't squash out any old, bad stuff. In general, left adjoints of forgetful functors may have to do some of both.

For an example of squashing out old, bad stuff, consider the category **Grp**, whose objects are groups and whose arrows are group homomorphisms. And consider the subcategory **Ab**, with only the abelian groups. There is an obvious forgetful functor $\mathcal{F}\colon \mathbf{Ab} \to \mathbf{Grp}$; it takes an abelian group H to that same group $\mathcal{F}(H) := H$, but now viewed as a general group; that is, the functor $\mathcal{F}$ forgets the axiom for commutativity.

The forgetful functor $\mathcal{F}$ has a left adjoint $\mathcal{C}$. How does $\mathcal{C}$ work? Given an arbitrary group G, we must produce, in some sense, the free abelian group generated by G. Let $[G, G]$ denote the commutator subgroup of G, the subgroup generated by all elements of the form $xyx^{-1}y^{-1}$, for x and y in G. The commutator subgroup $[G, G]$ is normal, so we can form the quotient group $G/[G, G]$, which will be abelian. That quotient is $\mathcal{C}(G)$. In particular, we have the required one-to-one correspondences:

$$\mathrm{Grp}(G, \mathcal{F}(H)) \longleftrightarrow_{G,H} \mathrm{Ab}(\mathcal{C}(G), H).$$

Thus, the left adjoint $\mathcal{C}$ produces an abelian group $\mathcal{C}(G)$ from an arbitrary group G by squashing out any bad, nonabelian stuff in G.

To determine whether a particular left adjoint adds in new, good stuff, squashes out old, bad stuff, or does some of both, we examine the units of the adjunction. An injective unit had no need to squash out any old, bad stuff, while a surjective unit had no need to add in any new, good stuff. All of the adjunctions in the ladder of Figure A.1 (even when extended as in Exercise A.5-1) have injective units, while the unit that maps an arbitrary group G to the "abelianized" group $\mathcal{F}(\mathcal{C}(G)) = G/[G, G]$ is surjective.

A.7 Right adjoints

Linearization and algebrization can be viewed as left adjoints of forgetful functors; but what about right adjoints of forgetful functors? Do they ever arise? I don't know of any examples in CAGD; but, for completeness, here is a natural forgetful functor in topology whose left adjoint and right adjoint are both useful.

Let **Top** be the category whose objects are topological spaces and whose arrows are continuous maps. There is an obvious forgetful functor $\mathcal{F}$ from **Top** to **Set**, the functor that forgets the topology. An adjoint of that forgetful functor must take an arbitrary set W and invent some topology for it.

If $\mathcal{S}$ is to be the left adjoint of $\mathcal{F}$, we must have one-to-one correspondences

$$\mathbf{Set}(W, \mathcal{F}(U)) \longleftrightarrow_{W,U} \mathrm{Top}(\mathcal{S}(W), U).$$

That is, any set map from W to any topological space U must, when we equip W with the topology that $\mathcal{S}$ chooses for it, turn out to be continuous. So $\mathcal{S}$ must equip W with the discrete topology, the topology in which every subset of W is open.

On the other hand, if $\mathcal{T}$ is to be the right adjoint of $\mathcal{F}$, we must have one-to-one correspondences

$$\mathbf{Set}(\mathcal{F}(U), W) \longleftrightarrow_{U,W} \mathrm{Top}(U, \mathcal{T}(W)).$$

That is, any set map from any topological space U to W must, when we equip W with the topology that $\mathcal{T}$ chooses for it, turn out to be continuous. So $\mathcal{T}$ must equip W with the trivial topology, the topology in which only W and the empty set are open.

A.8 Tensor-product surfaces revisited

Gentle reader, you will have to decide for yourself whether the unification, systematization, and type correctness provided by category theory are worth its conceptual overhead. But we can at least mention one instance where category theory would have assisted us in this monograph. That instance involves the tensor-product surfaces in Section 6.8, which turn out to be related to the notion of coproducts in category theory.

Products and *coproducts* are category-theoretic notions that may or may not exist in a given category. They are abstractly characterized by universal mapping conditions; indeed, the product, when it exists, is the right adjoint of a certain diagonal functor, while the coproduct, when it exists, is the left adjoint of that same functor. For the details, see Mac Lane [40].

Products and coproducts exist in all four of the categories that are the rungs on the adjunction ladder in Figure A.1. In the category **Set**, products are Cartesian products and coproducts are disjoint unions. In **Aff**, products are again Cartesian products, while coproducts are affine hulls; that is, to form the coproduct $A \amalg B$ of two affine spaces A and B, we position them in some larger affine space so as to be skew and so that no line in A is parallel to any line in B and we then take the affine hull of their union. We thus have $\dim(A \amalg B) = \dim(A) + \dim(B) + 1$. In **Lin**, products are Cartesian products, while coproducts are direct sums. Note that **Lin** is one of the unusual categories where (finite) products and (finite) coproducts are isomorphic; we have $X \times Y = X \oplus Y$, for linear spaces X and Y. Finally, in **CAlg**, products are Cartesian products while coproducts are tensor products.

On our adjunction ladder, coproducts are more interesting than products, since they correspond to different notions in each of the four categories. Those notions are tied together by the general theorem that any functor that is a left adjoint preserves coproducts. Note that the geometrization, linearization, and algebrization functors $\mathcal{G}$, $\mathcal{L}$, and $\mathcal{A}$ are all left adjoints of forgetful functors. So we get three theorems for free:

1. If S and T are sets, the affine space $\mathcal{G}(S \amalg T)$ that has the disjoint union $S \amalg T$ as a barycentric frame is the affine hull $\mathcal{G}(S) \amalg \mathcal{G}(T)$ of the affine spaces that have S and T as frames.

2. If A and B are affine spaces, the linearization $\mathcal{L}(A \amalg B)$ of their affine hull is the direct sum $\mathcal{L}(A) \oplus \mathcal{L}(B)$ of their linearizations.

3. If X and Y are linear spaces, the algebrization $\mathcal{A}(X \oplus Y)$ of their direct sum is the tensor product $\mathcal{A}(X) \otimes \mathcal{A}(Y)$ of their algebrizations.

The third of those theorems helps to explain why we build tensor-product surfaces as we do. Starting with the direct product $L_1 \times L_2$ of two affine parameter lines L_1 and L_2, the obvious thing to do would be to linearize that entire plane at once. But that result $\mathcal{L}(L_1 \times L_2)$ wouldn't be decomposable into parts associated with L_1 and L_2, since linearization doesn't preserve products; indeed, it would be 3-dimensional. So we instead linearize each affine parameter line separately, getting the linear space $\mathcal{L}(L_1) \times \mathcal{L}(L_2) = \hat{L}_1 \times \hat{L}_2$. At this point, we take advantage of the unusual property of **Lin** to convert the product into a coproduct, rewriting $\hat{L}_1 \times \hat{L}_2$ as the direct sum $\hat{L}_1 \oplus \hat{L}_2$. The third theorem from the list above then does the algebrizing for us: We have $\mathrm{Sym}(\hat{L}_1 \oplus \hat{L}_2) = \mathrm{Sym}(\hat{L}_1) \otimes \mathrm{Sym}(\hat{L}_2)$, a tensor-product algebra of sites. Tensor-product surfaces get their name from the dual algebra of forms, which is the tensor product $\mathrm{Sym}(\hat{L}_1^*) \otimes \mathrm{Sym}(\hat{L}_2^*)$ in a similar way.

Exercise A.8-1 Exercise A.5-1 extends the ladder of adjunctions, replacing **CAlg** with **EAlg** and then adding **RFld** as a new top rung. Do the categories **EAlg** of entire algebras and **RFld** of fields that include the real numbers as a subfield have products? Have coproducts?

Hint: The tensor product is a coproduct in the category **EAlg**, just as in **CAlg**. The category **RFld** also has coproducts, which can be constructed by first forming the tensor product and then forming quotients. But the direct product of two algebras always has zero divisors, since we have $(1,0)\cdot(0,1) = (0,0) = 0$; so the categories **EAlg** and **RFld** do not have products.

Appendix B

To Sum or to Average?

> (Often the coefficient $1/n!$ is put in front of the formula for ι; this makes no essential difference, but leads to awkward formulas for contractions.) – William Fulton and Joe Harris [22]

In this appendix, we analyze which pairing leads to a superior theory, the summed pairing or the averaged pairing. Fulton and Harris, the authors of the quote above, are pure mathematicians who adopt the summed pairing. When they promptly stumble across an annoying factor of $n!$, they analyze the averaged pairing as an alternative in the single pithy sentence quoted above. We here consider the issues at far greater length, on our way to the same conclusion.

B.1 Searching for pretty formulas

One way to judge which pairing is better is to see which leads to the simpler formulas. Table B.1 gives a good sample of formulas, both in their summed and their averaged versions.

The third line in Table B.1 points out a minor problem with averaging. In the averaging column, we find that $D_\pi f(P) = \langle f, n\pi P^{n-1}\rangle$. If I were teaching this theory to students, I would worry that my students would rewrite this formula as $D_\pi f(P) = \langle f, \pi n P^{n-1}\rangle$ and would then justify the annoying factor of n to themselves by recalling that

(B.1-1)
$$\frac{\partial}{\partial x}(P^n) = nP^{n-1}\frac{\partial P}{\partial x}.$$

That justification is a delusion based on a coincidence. We are differentiating the n-form f in this formula, not the n-site P^n. If you are still in any doubt, compare with the blossomed formula in the line below, where there is no P^{n-1} to remind us of the valid but irrelevant Equation B.1-1.

Table B.1: Formulas under the summed and averaged pairings

	summing	averaging
evaluate an n-form f at a point P	$f(P) = \langle f, P^n/n!\,\rangle$	$f(P) = \langle f, P^n\rangle$
evaluate the blossom $\tilde{f}$ of an n-form f at the points $(P_1, \ldots, P_n)$	$\tilde{f}(P_1, \ldots, P_n) = \langle f, P_1 \cdots P_n/n!\,\rangle$	$\tilde{f}(P_1, \ldots, P_n) = \langle f, P_1 \cdots P_n\rangle$
differentiate an n-form f in the direction of the vector π and evaluate the resulting $(n-1)$-form at P	$D_\pi f(P) = \langle f, \pi P^{n-1}/(n-1)!\,\rangle$	$D_\pi f(P) = \langle f, n\pi P^{n-1}\rangle$
differentiate an n-form f in the direction of the vector π and evaluate the blossom of the resulting $(n-1)$-form at the points $(P_1, \ldots, P_{n-1})$	$(D_\pi f)^\sim(P_1, \ldots, P_{n-1})$ $= \langle f, \pi P_1 \cdots P_{n-1}/(n-1)!\,\rangle$	$(D_\pi f)^\sim(P_1, \ldots, P_{n-1})$ $= \langle f, n\pi P_1 \cdots P_{n-1}\rangle$
differentiate an n-form f in the direction of the vector π	$D_\pi f = f \llcorner \pi$	$D_\pi f = f \llcorner n\pi$
differentiate an n-form k times, in the directions of the vectors π_1 through π_k, and then evaluate the $(n-k)$-form that results at P	$D_{\pi_1} \cdots D_{\pi_k} f(P)$ $= \langle f, \pi_1 \cdots \pi_k P^{n-k}/(n-k)!\,\rangle$	$D_{\pi_1} \cdots D_{\pi_k} f(P)$ $= \langle f, (n!/(n-k)!\,)\pi_1 \cdots \pi_k P^{n-k}\rangle$
differentiate an n-form k times, in the directions of the vectors π_1 through π_k	$D_{\pi_1} \cdots D_{\pi_k} f = f \llcorner \pi_1 \cdots \pi_k$	$D_{\pi_1} \cdots D_{\pi_k} f = f \llcorner (n!/(n-k)!\,)\pi_1 \cdots \pi_k$
differentiate an n-form n times, in the directions of the vectors π_1 through π_n, thus producing a constant	$D_{\pi_1} \cdots D_{\pi_n} f = \langle f, \pi_1 \cdots \pi_n\rangle$	$D_{\pi_1} \cdots D_{\pi_n} f = \langle f, n!\,\pi_1 \cdots \pi_n\rangle$
differentiate an n-form n times, each time in the direction of the vector π, thus producing a constant	$(D_\pi)^n f = \langle f, \pi^n\rangle$	$(D_\pi)^n f = \langle f, n!\,\pi^n\rangle$

	summing	averaging
evaluate an n-form at a point	$(E_P)^n f = \langle f, P^n/n!\,\rangle$	$(E_P)^n f = \langle f, P^n\rangle$
set one argument of the blossom of an n-form to a point	$E_P f = f \llcorner P/n$	$E_P f = f \llcorner P$
differentiate an n-form n times, each time in the direction of the same vector	$(D_\pi)^n f = \langle f, \pi^n\rangle$	$(D_\pi)^n f = \langle f, n!\,\pi^n\rangle$
differentiate an n-form once, in the direction of a vector	$D_\pi f = f \llcorner \pi$	$D_\pi f = f \llcorner n\pi$

Table B.2: Basic formulas under the summed and averaged pairings

Comparing the first line in Table B.1 to the last, we are reminded that evaluating an n-form is like differentiating n times, always in the same direction — except for that factor of $n!$. Comparing the second line to the next-to-last, we see that evaluating the blossom of an n-form is like differentiating n times, in arbitrary directions — again, except for the factor of $n!$. But is there any evaluation-like operator that corresponds to differentiating only once? We can define such an operator as follows, an operator E_P that does $(1/n)^{\text{th}}$ of the work of evaluating an n-form at P. In blossoming terms, what E_P does is to set one of the arguments of the blossom to P; that is, we require that

$$(E_P f)^{\sim}(Q_1, \ldots, Q_{n-1}) = \tilde{f}(Q_1, \ldots, Q_{n-1}, P).$$

We then have $(E_P)^n f = f(P)$. Using this new operator E_P, we can boil down Table B.1 into the more basic Table B.2, above.

Table B.2 shows that the summed pairing makes differentiation simple, hence forcing evaluation to divide, while the averaged pairing makes evaluation simple, hence forcing differentiation to multiply. Since evaluation seems more basic than differentiation, perhaps it is better to average.

On the contrary: There is a deep reason why summing is better; but it will take us a few paragraphs to develop the argument. The first step is to note that some of the numeric factors of n or $n!$ in Table B.2 are more expensive than others, the expensive ones being the ones on lines two and four, the lines where the left-hand sides don't involve n.

On lines one and three, we are evaluating an n-form completely or differentiating it n times. That is, we are applying the n^{th}-order operator $(E_P)^n$ or $(D_\pi)^n$ to set all n factors of the n-site with which that n-form gets paired. So we have to know n — we have a good reason for needing to know it. If we have to multiply or divide by $n!$ as well, that is only an annoyance.

But lines two and four are a different story. It makes sense to apply the operator E_P or D_π to an n-form f without knowing n; indeed, we can even apply those operators to inhomogeneous forms. If we are forced to require that f be an n-form solely because we have to multiply or divide by n, that is worse than annoying; it could be crippling.

For an example of those crippling effects, consider the product rule for differentiation: $D_\pi(fg) = D_\pi(f)g + fD_\pi(g)$. This rule is valid for all forms f and g, independent of their degrees; in fact, it is valid even for forms f and g that are inhomogeneous. Under the summed pairing, that rule turns into an analogous rule for contracting a product on an anchor, also valid for all forms f and g:

$$fg \llcorner \pi = (f \llcorner \pi)g + f(g \llcorner \pi) \qquad \text{[under summing]}. \tag{B.1-2}$$

But suppose that we choose averaging instead of summing, so that $D_\pi(f) = f \llcorner n\pi$. Then, in order to state the rule for contracting a product on an anchor, the two forms involved have to be homogeneous and we have to know their degrees; if f is an n-form and g is an m-form, we have

$$fg \llcorner (n+m)\pi = (f \llcorner n\pi)g + f(g \llcorner m\pi) \qquad \begin{bmatrix}\text{under averaging} \\ \text{with } \deg(f) = n \\ \text{and } \deg(g) = m.\end{bmatrix}$$

The crippling thing here is replacing a single, general rule with a two-parameter family of narrow sub-rules. It is the off-diagonal sub-rules that are particularly crippled, in this case. On the diagonal $m = n$, where both f and g are n-forms, we can divide through by $m + n = 2n$ to get

$$fg \llcorner \pi = \frac{(f \llcorner \pi)g + f(g \llcorner \pi)}{2} \qquad \begin{bmatrix}\text{under averaging} \\ \text{with } \deg(f) = \deg(g).\end{bmatrix}$$

Note that this rule differs from Rule B.1-2 as averaging differs from summing.

So the multiplication by n in the averaged formula $D_\pi f = f \llcorner n\pi$ is crippling. What about the division by n in the summed formula $E_P f = f \llcorner P/n$; is it also crippling? It has the potential to be, since it forces us to know the degree of a form f before we can apply the operator E_P to f. If there were powerful identities in which the operator E_P was applied to inhomogeneous forms, summing would cripple our ability to translate those identities into the paired-algebras framework. But there aren't. That's why you've known about the operator D_π for decades, while you learned about E_P only a minute ago — because D_π has the powerful identities, not E_P. For example, the product rule for E_P is a two-parameter family of sub-rules; when f is an n-form and g is an m-form, we have

$$E_P(fg) = \frac{n}{n+m}(E_P f)g + \frac{m}{n+m}f(E_P g).$$

We should tolerate being annoyed to avoid being crippled, so lines two and four in Table B.2 matter more than lines one and three. Since the operator D_π satisfies more powerful identities than does E_P, line four matters more than line two. To me, at least, this seems like a strong argument in favor of choosing the pairing that makes line four simple: the summed pairing.

The choice of whether to sum or to average is an issue of taste on which reasonable people may differ. And averaging definitely seems more attractive initially — I started out by averaging myself. But I now believe that CAGD will be better served in the long run if we adopt the summed pairing.

By the way, pure mathematicians often sum, rather than average, because they want their theory to apply over fields of prime characteristic. Over a field whose characteristic is a prime p, we have $n! = 0$ for all $n \geq p$, so we can't divide by $n!$. That argument is irrelevant in CAGD, of course, where the only fields of interest are the reals and perhaps the complexes. But it certainly isn't a bad thing for us in CAGD to adopt the same convention as the majority of pure mathematicians, since that will make it easier for us to read their textbooks.

B.2 Other options for avoiding annoyance

We have discussed the summed pairing and the averaged pairing as if they were our only possible options: a binary choice. There are other alternatives, of course, and several of them are worth mentioning — although not, in my opinion, worth adopting.

One way to avoid the annoying factors would introduce both the summed pairing and the averaged pairing, using different notations (perhaps $\langle\ ,\ \rangle$ and $[\ ,\ ]$). We could then use the summed pairing in formulas for differentiation and the averaged pairing in formulas for evaluation. But having two pairings around would be confusing. For example, when describing one basis as being dual to another, we would have to specify under which pairing. The cost in confusion would probably outweigh the benefit in reduced annoyance.

Several other schemes for mitigating the annoying factors involve *rescaled exponentials*, which incorporate a denominator of $n!$ into the concept of an n^{th} power. I learned of this clever idea from Greub's text [26]; I don't know who first came up with it. There are several ways to exploit this idea, so that we can adopt the summed pairing and still end up with pretty evaluation formulas. In one of those ways, the rescaling of the exponentials is explicit; in the other, it is implicit.

With explicit rescaling, we would introduce a new notation, perhaps $x^{n/!} := x^n/n!$, reading "$x^{n/!}$" as "x to the n slash bang". Exploiting this new notation, the formula for evaluating an n-form f under the summed pairing would be $f(P) = \langle f, P^{n/!}\rangle$, which is slightly prettier than $f(P) = \langle f, P^n/n!\,\rangle$.

We could also write other famous formulas more neatly using these rescaled exponentials. For example, we could expand an analytic function $f\colon \mathbf{R} \to \mathbf{R}$ in a Taylor series around b as

$$f(x) = \sum_{n \geq 0} f^{(n)}(b)(x-b)^{n/!}.$$

Even better, we could write the Binomial Theorem as

$$(x+y)^{n/!} = \sum_{0 \leq k \leq n} x^{(n-k)/!} y^{k/!}.$$

All three of the factorials that normally constitute the binomial coefficient $\binom{n}{k}$ are hidden here in the three rescaled exponentials. But other formulas are prettier using standard exponentials. For example, the rule $x^i x^j = x^{i+j}$ for multiplying by adding exponents is prettier than the alternative

$$x^{i/!} x^{j/!} = \binom{i+j}{i} x^{(i+j)/!}.$$

So we would probably end up using both rescaled and standard exponentials. I am afraid that the cost in confusion from having two different exponentials would outweigh the small benefit in reduced annoyance.

A more radical approach would implicitly rescale all exponentials of anchors. Rather than introducing a new notation, we would define p^n, for an anchor p over A, to mean

$$p^n := \frac{\overbrace{p \cdot p \cdots p}^{n \text{ factors}}}{n!},$$

leading to the summed-pairing evaluation formula $f(p) = \langle f, p^n \rangle$. But that radical approach is a nonstarter in our context, because it would destroy the symmetry between the algebras of forms and sites. In building up a form out of coanchors, it is a universal, well-established convention that the exponentials used are standard exponentials. Symmetry requires that we use standard exponentials also in building up a site out of anchors.

By the way, Greub adopts that radical approach in building his symmetric algebras. But things are a bit different for Greub because he uses the special symbol "$\vee$" to denote the multiplication in his symmetric algebra, rather than a centered dot or simple juxtaposition. The products in his symmetric algebras hence look less like standard products, so it is not so shocking that his exponentials,

$$x^n := \frac{\overbrace{x \vee x \vee \cdots \vee x}^{n \text{ factors}}}{n!},$$

are rescaled, rather than standard. (Whether rescaling or not, some authors would prefer to write $x^{\vee n}$ in this context, rather than x^n.)

Appendix C

More Math Remarks

C.1 Thoughts about functional notations

In this monograph, we follow the standard mathematical conventions for notating functions in three respects:

1. When applying a function f to a datum x, we write $f(x)$, with the function on the left; that is, we denote functions as prefix operators. The alternative would be some flavor of postfix operator, as in the expressions $(x)f$, xf, or x^f.

2. We compose functions from right to left, so that applying $f \circ g$ means applying first g and then f. The alternative, of course, would be a composition operator that worked from left to right. (If you ever need such an operator, consider using a semicolon, setting $g; f := f \circ g$. The analogy with computer programs almost requires that applying $g; f$ means applying g first.)

3. When denoting the type of a function, we give its domain first and its codomain second, connected by a rightward-pointing "to" arrow; for example, we might declare f to be "a function from A to B" and write $f \colon A \to B$. The alternative would be to give the codomain first and the domain second, connected by a leftward-pointing "from" arrow. In this alternative scheme, we would declare that same function f from A to B as "a function to B from A" and write $f \colon B \leftarrow A$.

Unfortunately, the third of those standard conventions is inconsistent with the first two. Prefix application and right-to-left composition interact better with "from" arrows than with "to" arrows. For example, it was our use of "to" arrows that made the declaration of the linear map $h_{12} \colon X_2 \to X_1$ in Proposition 9.1-3 seem backward. We would have been better off if we had declared the map h_{12} with a "from" arrow, as $h_{12} \colon X_1 \leftarrow X_2$.

Sadly, we mathematicians may be stuck with this inconsistency for a long time. Switching to "from" arrows works fine in limited contexts; but do we really want all of the arrows in our commutative diagrams to point from right to left? And, probably also, for consistency, from bottom to top? Postfix application and left-to-right composition also work fine in certain contexts. But how do you read the postfix application $(x)f$ in English? The prefix application $f(x)$ is "f of x" or "f at x". Perhaps $(x)f$ could be "x into f" or "x through f" or "x sent to f"? Also, suppose that we want to deemphasize the argument x by setting it in smaller type. If f is a prefix operator, there is no difficulty: The argument x becomes a subscript, as in f_x. Are we ready for subscripts on the left, as in ${}_xf$? How would we read that expression? What about double subscripts, as in ${}_{n\,x}f$?

By the way, duality is one context where it might be helpful to adopt both of the sets of consistent conventions simultaneously, using one for primal functions and the other for their duals. For example, we could use prefix application, right-to-left composition, and "from" arrows on the primal side, but postfix application, left-to-right composition, and "to" arrows on the dual side. Thus, we might introduce the primal linear functions $f\colon X \leftarrow Y$ and $g\colon Y \leftarrow Z$. Their composition $f \circ g$ would then be applied as a prefix operator to an element z of Z, getting $f \circ g(z) = f(g(z))$ in X. The dual functions would then be $f^*\colon X^* \to Y^*$ and $g^*\colon Y^* \to Z^*$. Their composition $f^*; g^*$ would be applied as a postfix operator to an element x^* of X^*, getting $(x^*)f^*; g^* = ((x^*)f^*)g^*$ in Z^*. One advantage of this scheme is that the primal operator f and its dual f^* are represented by the same matrix. To apply f, we multiply that matrix by a column vector on the right; to apply f^*, we multiply that same matrix by a row vector on the left. We would have the identity $(f \circ g)^* = f^*; g^*$, an analog of de Morgan's Law.

C.2 Paired algebras in wilder contexts

The paired-algebras framework for CAGD exploits some mathematics that produces a pair of algebras $(\mathrm{Sym}(X), \mathrm{Sym}(Y))$ from a dual pair (X, Y) of linear spaces. As we developed that theory in this monograph, we mentioned various things that can go wrong in contexts wilder — that is, more general — than ours. In this section, let's review that story, starting out with the most restrictive assumptions, which lead to the prettiest theory.

C.2.1 Fields of characteristic zero

For the very prettiest theory, the spaces X and Y should be finite-dimensional linear spaces over the complex numbers, or, more generally, over any algebraically closed field of characteristic zero. For example, over such a field, every univariate n-ic splits into n linear factors, and a bivariate quadratic splits into two linear factors just when its discriminant is zero.

The real numbers are the field of scalars that is most relevant in CAGD. The reals have characteristic zero, but are not algebraically closed. That makes factoring forms and sites more complicated. But the rest of the theory is unaffected, as it would be over any field of characteristic zero.

C.2.2 Fields of prime characteristic

Suppose that we start with finite-dimensional linear spaces X and Y, but over a field whose characteristic is a prime p. For simplicity, let's assume for now that the scalar field is infinite; for example, it might be the algebraic closure of the field $\mathbf{Z}/(p)$ for some prime p or the field $(\mathbf{Z}/(p))(t)$ of rational functions in an indeterminate t, where the coefficients in the numerator and denominator polynomials are taken from $\mathbf{Z}/(p)$. We can still build the symmetric algebras $\mathrm{Sym}(X)$ and $\mathrm{Sym}(Y)$. And, for each n, there is still a unique scalar-valued bilinear map on $\mathrm{Sym}_n(X) \times \mathrm{Sym}_n(Y)$ that satisfies the Summed Permanent Identity. Unfortunately, that map is no longer a pairing, in general. Indeed, as soon as n is at least p, the Summed Permanent Identity implies that $\langle x_1 \cdots x_n, y_1 \cdots y_n \rangle = 0$ when either $x_1 = \cdots = x_n$ or $y_1 = \cdots = y_n$, since the resulting sum then has $n! = 0$ identical terms. The Averaged Permanent Identity is even worse, since it tries to get rid of this factor of $n! = 0$ by dividing by it. The linear spaces $\mathrm{Sym}_n(X)$ and $\mathrm{Sym}_n(Y)$ still have the same dimension, so there still exist lots of pairing maps between them — but none of those pairing maps interact with the multiplications in the two algebras as specified by the Permanent Identity.

It is a serious blow to the theory that the bilinear map defined by the Permanent Identity is not a pairing. Furthermore, that failure cannot be repaired by, for example, rearranging the numeric factors in some clever way; rather, the case of prime characteristic really is different. To see how, consider the cubing map $\sigma\colon \hat{A} \to \mathrm{Sym}_3(\hat{A})$ given by $\sigma(q) := q^3$ for any anchor q over A and the evaluation-of-a-cubic map $\epsilon\colon \hat{A} \to \mathrm{Sym}_3(\hat{A}^*)^*$ given by $\epsilon(q)(f) = \epsilon_q(f) := f(q)$. Over a field of characteristic zero, the maps σ and ϵ behave essentially the same; so, by choosing the proper pairing, we can represent $\mathrm{Sym}_3(\hat{A}^*)^*$ as $\mathrm{Sym}_3(\hat{A})$ in such a way that $\epsilon(q)$ is represented by $\sigma(q)$, for all anchors q. (The pairing that does precisely that, of course, is the averaged pairing; under the superior summed pairing, $\epsilon(q) = \sigma(q)/6$.) Over a field of characteristic $p = 3$, however, the maps σ and ϵ behave quite differently. For example, when A is an affine plane, we have

$$\sigma(wC + u\varphi + v\psi) = (wC + u\varphi + v\psi)^3 = w^3C^3 + u^3\varphi^3 + v^3\psi^3,$$

with the remaining seven terms dropping out, due to their multinomial coefficients of 3 or 6. Thus, the cube of any anchor over A lies in the 3-dimensional subspace of $\mathrm{Sym}_3(\hat{A})$ spanned by the 3-sites C^3, φ^3, and ψ^3. Over any infinite field, however, even one of prime characteristic, the values of any polynomial determine its coefficients uniquely. Hence, the linear functionals in the set $\epsilon(\hat{A})$ span all 10 dimensions of the space $\mathrm{Sym}_3(\hat{A}^*)^*$. So there is no hope that some juggling of scale factors could allow us to represent $\mathrm{Sym}_3(\hat{A}^*)^*$ as $\mathrm{Sym}_3(\hat{A})$ in such a way that $\epsilon(q)$ is represented by $\sigma(q)$, for all q.

By the way, over fields of prime characteristic, the term "Veronese" is associated with evaluating an n-ic, rather than with raising to the n^{th} power. For example, in characteristic 3, the "Veronese surface of parametric degree 3" refers to the image of the evaluation map $\epsilon\colon \hat{A} \to \mathrm{Sym}_3(\hat{A}^*)^*$, rather than to the (degenerate) image of the cubing map $\sigma\colon \hat{A} \to \mathrm{Sym}_3(\hat{A})$.

C.2.3 Finite fields

If our field of scalars is not only of prime characteristic but is actually finite, then the values of a polynomial are no longer enough, in general, to uniquely determine the coefficients of that polynomial. This means that the evaluation-of-an-n-ic map ϵ can be degenerate, as well as the raise-to-the-n^{th}-power map σ.

For an example, let A be an affine plane once again, but now over the field $\mathbf{Z}/(3)$ of cardinality 3. So there are only 9 points in A and only 27 anchors over A. Consider the ninth-power map $\sigma\colon \hat{A} \to \mathrm{Sym}_9(\hat{A})$ and the evaluation-of-a-nonic map $\epsilon\colon \hat{A} \to \mathrm{Sym}_9(\hat{A}^*)^*$. The equation $\sigma(q) = q^9 = (q^3)^3$ shows that $\sigma(wC+u\varphi+v\psi) = w^9C^9+u^9\varphi^9+v^9\psi^9$, so the ninth powers of all anchors over A lie in a 3-dimensional subspace of $\mathrm{Sym}_9(\hat{A})$; thus, the map σ is quite degenerate. But the evaluation map ϵ must be somewhat degenerate also, just by counting. Since there are only 27 anchors over A, the linear functionals in the set $\epsilon(\hat{A})$ can't span more than 27 dimensions, while the full space $\mathrm{Sym}_9(\hat{A}^*)^*$ has dimension 55. (In fact, since $\epsilon(0) = 0$ and $\epsilon(-q) = -\epsilon(q)$, we can tighten the bound on the dimension from 27 to $(27-1)/2 = 13$, and 13 turns out to be the exact answer.)

This degeneracy of ϵ torpedoes one of the three concrete constructions of the symmetric algebra $\mathrm{Sym}(X)$ that we discussed in Section 5.1, the one that exploits duality. Over a finite field, we have to construct the algebra $\mathrm{Sym}(X)$ by dealing in some way with polynomials whose variables lie in X. We can't exploit duality to replace each such polynomial by the scalar-valued function on $Y = X^*$ that it defines, since two distinct such polynomials may define the same function.

On the other hand, some things get nicer over a finite field. A subspace, such as a line or a plane, has only finitely many points in it. So we can associate, with such a subspace, a polynomial that has all of the points in that subspace as its roots; see Macdonald [39].

C.2.4 Linear spaces of infinite dimension

If X is a linear space of infinite dimension, then its dual space $Y := X^*$ is vastly bigger than X, so the concept of a pairing breaks down already in degree 1. We can still build the algebra $\mathrm{Sym}(X)$, and it is still an algebra of polynomials — albeit polynomials in an infinite number of variables. And we can build the algebra $\mathrm{Sym}(Y)$ as well. But $\mathrm{Sym}(Y)$ is so much bigger than $\mathrm{Sym}(X)$ that those two algebras cannot be paired (except in degree 0, where $\mathrm{Sym}_0(X) = \mathrm{Sym}_0(Y) = \mathbf{R}$).

For those who are curious, here is what I mean when I say that the dual space X^* is "vastly bigger" than X. Let F be a field and let X be a linear space over F whose dimension $\kappa := \dim_F(X)$ is infinite; so κ is an infinite cardinal. It then turns out [6] that $\dim_F(X^*) = |X^*| = |F|^\kappa \geq 2^\kappa > \kappa$.

C.2.5 Modules over commutative rings

Things get still wilder when we generalize from linear spaces over a field to modules over a commutative ring R. Here are a few of the problems that can arise in that context.

First, the additive group of the ring R may have torsion. If so, we end up with all of the problems associated with fields of prime characteristic, only worse. Different elements of R may have different additive orders, and those orders need not be prime.

Second, as we mentioned in Section 9.3, only the nicest R-modules M, the *free* modules, have any bases at all. So a construction of the symmetric algebra $\mathrm{Sym}(M)$ that starts by choosing a basis may not be applicable. The field of rational numbers $\mathbf{Q}$, viewed as a module over the integers $\mathbf{Z}$, provides a simple example of a torsion-free module that is not free.

Bourbaki [4, 5] is a good source for the theory of the symmetric algebra at this level of generality and subtlety.

C.2.6 The division ring of quaternions

What about a ring R that isn't commutative? We can still define the concept of an R-module, but we must now specify whether scalars multiply from the left or from the right. Let's think about left R-modules, where the scalar multiplication takes r in R and m in M to rm in M. Things are now wilder yet; for example, a free left R-module can have bases of different cardinalities.

Exercise C.2-1 Consider real (or integral — it doesn't matter) matrices $\mathbf{m} := (m_{ij})_{i,j\geq 0}$ with infinitely many rows and columns, each of which is eventually identically zero. The product of two such matrices is another such, so the set of all such matrices forms a noncommutative ring R. Consider the ring R itself as a left R-module. The module R is free, since the identity matrix $\mathbf{i}$ forms a basis of cardinality one; for any $\mathbf{m}$ in R, the equation $\mathbf{m} = \mathbf{ri}$ has the unique solution $\mathbf{r} = \mathbf{m}$. Show that the following matrices $\mathbf{j}$ and $\mathbf{k}$ form a basis of cardinality two:

$$\mathbf{j} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \qquad \mathbf{k} := \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}.$$

That is, for any $\mathbf{m}$ in R, show that there exist unique matrices $\mathbf{r}$ and $\mathbf{s}$ in R with $\mathbf{m} = \mathbf{rj} + \mathbf{sk}$.

Left modules over division rings are much better behaved — indeed, are almost as well behaved as linear spaces over fields. A *division ring* (a.k.a. *skew field*) is a nonzero[†] ring in which every nonzero element is invertible. If R is a division ring, then every left R-module M is free and all bases for M have the same cardinality.

The division ring that is most likely to be of interest in CAGD is the quaternions $\mathbf{H}$. It would be interesting to study how much of the theory of the paired algebras survives when working with left $\mathbf{H}$-modules. One bad sign is that only a pale shadow of the theory of determinants carries over to matrices of quaternions. While we would naively hope that the determinant of a matrix $\mathbf{m}$ of quaternions was itself a quaternion, say h, the best that can be done — see Artin [2] — is to define $\det(\mathbf{m})$ to be a nonnegative real number that captures, roughly speaking, the norm $|h|$. The theory of permanents of matrices of quaternions is likely to be a shadow that is similarly pale, and that would be bad news for the paired algebras.

[†]There is a unique ring $\{0\}$, called the *zero ring*, in which $1 = 0$; we don't want the zero ring to qualify as a division ring.

Bibliography

[1] Shreeram S. Abhyankar. *Algebraic Geometry for Scientists and Engineers*. American Mathematical Society (1990), pages 125–126.

[2] E. Artin. *Geometric Algebra*. John Wiley & Sons/Interscience (1957), pages 151–158.

[3] Marcel Berger. *Geometry I*, translated from the French by M. Cole and S. Levy. Springer-Verlag (1987), chapter 3, pages 67–84.

[4] Nicolas Bourbaki. *Elements of Mathematics: Algebra I, Chapters 1–3.* Translated from the French, Springer-Verlag (1989), pages 497–506 and 587–606.

[5] Nicolas Bourbaki. *Elements of Mathematics: Algebra II, Chapters 4–7.* Translated from the French by P. M. Cohn and J. Howie, Springer-Verlag (1990), pages IV.41–IV.61.

[6] Robin J. Chapman. Solution to Problem 10468. *The American Mathematical Monthly* **105** (1998), pages 368–369.

[7] Bruce W. Char, Keith O. Geddes, Gaston H. Gonnet, Benton L. Leong, Michael B. Monagan, and Stephen M. Watt. *Maple V Library Reference Manual*. Springer-Verlag (1991).

[8] Pages 64–65.

[9] Pages 472–473.

[10] Hyeong In Choi, Doo Seok Lee, and Hwan Pyo Moon. Clifford algebra, spin representation and rational parameterization of curves and surfaces. Submitted to *Advances in Computational Mathematics*.

[11] Wolfgang Dahmen, Charles A. Micchelli, and Hans-Peter Seidel. Blossoming begets B-spline bases built better by B-patches. *Mathematics of Computation* **59** (1992), pages 97–115.

[12] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag (1978).

[13] Pages 5–6.

[14] Paul de Faget de Casteljau. *Formes à Pôles*. Hermes (1985). Also published in English translation as *Shape Mathematics and CAD*, Kogan Page Ltd. (1986).

[15] W. L. F. Degen. The types of triangular Bézier surfaces. In Glen Mullineux, editor, *The Mathematics of Surfaces VI*, proceedings of a conference held at Brunel University in 1994, sponsored by the Institute of Mathematics and its Applications, Oxford University Press (1996), pages 153–170.

[16] Proposition 3.10 on page 160.

[17] Page 163.

[18] W. L. F. Degen. The types of rational (2,1)-Bézier surfaces. *Computer Aided Geometric Design* **16** (1999), pages 639–648.

[19] C. T. J. Dodson and T. Poston. *Tensor Geometry: The Geometric Viewpoint and its Uses*, second edition. Springer-Verlag (1991), pages 57 and 61.

[20] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, third edition. Academic Press (1993), pages 65–67 and 72–74.

[21] J. C. Fiorot and P. Jeannin. *Rational Curves and Surfaces: Applications to CAD*, translated from the French by M. C. Harrison. John Wiley & Sons (1992).

[22] William Fulton and Joe Harris. *Representation Theory: A First Course.* Springer-Verlag (1991), appendix B, pages 473–477.

[23] Ronald N. Goldman. Dual polynomial bases. *Journal of Approximation Theory* **79** (1994), pages 311–346.

[24] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley (1989), pages 47–48.

[25] Werner Greub. *Multilinear Algebra*, second edition. Springer-Verlag (1978), pages 209–226.

[26] Page 212.

[27] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 2.0: A Computer Algebra System for Polynomial Computations. University of Kaiserslautern (2001). `http://www.singular.uni-kl.de`.

[28] Joe Harris. *Algebraic Geometry: A First Course*. Springer-Verlag (1992).

[29] Pages 23–25.

[30] Pages 25–26, 31.

[31] Page 231.

[32] Robin Hartshorne. *Algebraic Geometry*. Springer-Verlag (1977).

[33] Exercises 2.12 and 2.13, page 13.

[34] Exercise 2.14, page 13.

[35] Kiyosi Itô, editor. *Encyclopedic Dictionary of Mathematics*, second edition. MIT Press (1987), article 16 on Algebraic Varieties, section AA on Étale Topology, pages 85–86.

[36] Serge Lang. *Algebra*, third edition. Addison-Wesley (1993), chapter XVI, sections 6–8, pages 629–637 and chapter XIX, sections 1 and 4, pages 731–738 and 749–753.

[37] Jean B. Lasserre and Konstantin E. Avrachenkov. The multi-dimensional version of $\int_a^b x^p\,dx$. *The American Mathematical Monthly* **108** (2001), pages 151–154.

[38] Suresh Lodha and Ron Goldman. Change of basis algorithms for surfaces in CAGD. *Computer Aided Geometric Design* **12** (1995), pages 801–824.

[39] I. G. Macdonald. *Symmetric Functions and Hall Polynomials*, second edition. Oxford University Press (1995), pages 37–39.

[40] Saunders Mac Lane. *Categories for the Working Mathematician*, second edition. Springer-Verlag (1998).

[41] Ian R. Porteous. *Clifford Algebras and the Classical Groups*. Cambridge University Press (1995).

[42] Lyle Ramshaw. *Blossoming: A Connect-the-Dots Approach to Splines*. Research report 19 from the Systems Research Center of Compaq Computer Corp., then Digital Equipment Corp. (1987).

[43] Lyle Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design* **6** (1989), pages 323–358.

[44] Lyle Ramshaw and James B. Saxe. *From Quadrangular Sets to the Budget Matroids*. Research report 134a from the Systems Research Center of Compaq Computer Corp., then Digital Equipment Corp. (1995).

[45] George Salmon. *A Treatise on the Higher Plane Curves*, third edition. Chelsea (1879), articles 218 and 226 on pages 189–198.

[46] Jorge Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations*. Academic Press (1991).

[47] Gert Vegter. The apolar bilinear form in geometric modeling. *Mathematics of Computation* **69** (2000), pages 691–720.

[48] Robert J. Walker. *Algebraic Curves*. Springer-Verlag (1950), page 96.

Index