JOURNAL OF APPROXIMATION THEORY 79, 311–346 (1994)

Dual Polynomial Bases

RONALD N. GOLDMAN

Department of Computer Science, Rice University, P.O. Box 1892, Houston, Texas 77251

Communicated by Charles K. Chui

Received July 1, 1992; accepted in revised form October 27, 1993

Formulas and procedures for B-spline and progressive polynomial bases including Marsden's identity, the blossoming and de Boor–Fix forms of the dual functionals, the Oslo algorithm, and recursive procedures for evaluation differentiation, and blossoming are extended to arbitrary polynomial, and locally linearly independent spline, bases. © 1994 Academic Press, Inc.

1. INTRODUCTION

Blossoming [15, 19, 25, 26, 27, 30], the de Boor–Fix dual functionals [12], Marsden's identity [23], the Oslo algorithm [16, 22], recursive procedures for evaluation and differentiation [10, 11]: these are topics usually associated only with B-splines. But all these formulas have local polynomial interpretations. The purpose of this paper is to extend these formulas and techniques to totally arbitrary polynomial, and locally linearly independent piecewise polynomial, bases. These extensions help to unify the theory of univariate polynomials and splines, and they also provide some additional perspective on the special status of the Bernstein and B-spline bases.

We begin in Section 2 by generalizing Marsden's identity to arbitrary polynomial bases. Then we apply Marsden's identity in Section 3 to generate the blossoming and de Boor–Fix forms of the dual functionals. From the dual functionals we derive change of basis algorithms in Section 4, and general recursive algorithms for evaluation, differentiation, and blossoming in Section 5. In Section 6 we present some concrete examples to flesh out the details of the theory, and we extend our results from polynomial bases to locally linearly independent spline bases. We revisit our formulas for Marsden's identity and the de Boor–Fix dual functions and discuss what is so special about these particular formulas in Section 7. In Section 8 we develop necessary and sufficient criteria for extending the blossoming form of the dual functionals to multivariate polynomial bases. Finally, we close in Section 9 with a brief summary of our work and a few open questions for future research.

0021-9045/94 $6.00

2. Marsden's Identity and Dual Polynomial Bases

Let $\{B_{in}(t)\}$ be the B-splines of degree n associated with the knot vector $\{t_i\}$, and let $\Psi_{in}(t) = (t_{i+1} - t) \cdots (t_{i+n} - t)$. Then Marsden's identity is the formula [23]

$$(x - t)^n = \Sigma_i \Psi_{in}(t) B_{in}(x). \tag{1}$$

Essentially Marsden's identity is an extension of the binomial theorem to B-splines (see Section 6). This formula plays a fundamental role in the theory of B-splines. Indeed most of the properties of univariate B-splines can be derived directly from this simple identity [3, 13].

We begin by extending Marsden's identity to arbitrary polynomial bases. This extension will play a central role in much of the discussion that follows.

Theorem 2.1. *Let $b_0(x), \dots, b_n(x)$ be a basis for the polynomials of degree n in x. Then there exists a unique collection of polynomials $d_0(t), \dots, d_n(t)$ of degree n in t such that*

$$(x - t)^n = \Sigma_i d_i(t) b_i(x). \tag{2}$$

Moreover $d_0(t), \dots, d_n(t)$ is a basis for the polynomials of degree n in t.

Proof. Since $b_0(x), \dots, b_n(x)$ is a basis for the polynomials of degree n in x, for each t there must be unique constants $d_0(t), \dots, d_n(t)$ independent of x such that

$$(x - t)^n = \Sigma_i d_i(t) b_i(x).$$

To show that $d_0(t), \dots, d_n(t)$ form a basis for the polynomials of degree n in t, differentiate Eq. (2) $n - k$ times with respect to x and evaluate at $x = 0$ to obtain

$$\{(-1)^k n!/(n-k)!\} t^k = \Sigma_i b_i^{(n-k)}(0) d_i(t), \qquad k = 0, 1, \dots, n.$$

Thus for $k = 0, \dots, n$, we can write t^k as a linear combination of $d_0(t), \dots, d_n(t)$. Therefore the functions $d_0(t), \dots, d_n(t)$ span the space of polynomials of degree n in t, and since there are $n + 1$ functions in this set they must necessarily form a basis for the polynomials of degree n in t.
Q.E.D.

We shall call the bases $b_0(x), \dots, b_n(x)$ and $d_0(t), \dots, d_n(t)$ that appear in Eq. (2) *dual polynomial bases*. As we shall see in subsequent sections, these bases are dual in two senses. First, various formulas and theorems remain valid when a basis is replaced by its dual. Second, the dual basis can be used to represent the dual functionals of the primary basis.

In certain applications such as computer aided geometric design (CAGD) it is desirable that the basis functions be normalized to sum to one. A pair of dual bases cannot be normalized simultaneously. The following results tell us how normalizing one basis affects it dual basis.

LEMMA 2.2. *Let $b_0(x), \dots, b_n(x)$ and $d_0(t), \dots, d_n(t)$ be dual polynomial bases. Then*

$$\Sigma_i b_i(x) = 1 \Leftrightarrow d_j^{(n)}(t) = (-1)^n n!, \qquad j = 0, 1, \dots, n.$$

Proof. Differentiating Marsden's identity (Eq. (2)) n times with respect to t, we obtain

$$(-1)^n n! = \Sigma_i d_i^{(n)}(t) b_i(x).$$

Clearly then

$$d_j^{(n)}(t) = (-1)^n n!, j = 0, 1, \dots, n \Rightarrow \Sigma_i b_i(x) = 1.$$

Conversely, if $\Sigma_i b_i(x) = 1$, then

$$\Sigma_i (-1)^n n! b_i(x) = \Sigma_i d_i^{(n)}(t) b_i(x) \Rightarrow d_j^{(n)}(t) = (-1)^n n!,$$
$$j = 0, 1, \dots, n. \quad Q.E.D.$$

THEOREM 2.3. *Let $b_0(x), \dots, b_n(x)$ and $d_0(t), \dots, d_n(t)$ be dual polynomial bases, and let $r_{j1}, \dots, r_{jn}$ be the roots of $d_j(t)$, $j = 0, 1, \dots, n$. Then*

$$\Sigma_i b_i(x) = 1 \Leftrightarrow d_j(t) = (r_{j1} - t) \cdots (r_{jn} - t), \qquad j = 0, 1, \dots, n.$$

Proof. This result follows immediately from Lemma 2.2. Q.E.D.

At the other extreme from bases normalized to sum to one are triangular bases. A basis $b_0(x), \dots, b_n(x)$ is said to be *triangular* if and only if $\mathrm{Deg}\{b_k(x)\} = k$, $k = 0, \dots, n$.

THEOREM 2.4. *A basis $b_0(x), \dots, b_n(x)$ is triangular, if and only if its dual basis $d_0(t), \dots, d_n(t)$ in reverse order is triangular. That is,*

$$\mathrm{Deg}\{b_k(x)\} = k, k = 0, 1, \dots, n \Leftrightarrow \mathrm{Deg}\{d_{n-k}(t)\} = k, k = 0, 1, \dots, n.$$

Proof. Suppose that $b_0(x), \dots, b_n(x)$ is triangular. Then by differentiating the Marsden identity (Eq. (2)) n times with respect to x, we obtain

$$n! = d_n(t) b_n^{(n)}(x).$$

Hence clearly $\mathrm{Deg}\{d_n(t)\} = 0$. Now suppose that $\mathrm{Deg}\{d_{n-j}(t)\} = j$ for $j < k$ and proceed by induction on k. Differentiating the Marsden identity (Eq. (2)) $n - k$ times with respect to x, we obtain

$$\{n!/(n-k)!\}(x-t)^k = \Sigma_{j \geq n-k} d_j(t) b_j^{(n-k)}(x).$$

Since the left hand side is a polynomial of degree k in t, the right hand side must also be a polynomial of degree k in t. Hence by the inductive hypothesis, $\mathrm{Deg}\{d_{n-k}(t)\} = k$. The proof of the converse is essentially the same. Q.E.D.

3. Dual Functionals: Blossoming and the de Boor–Fix Formula

Dual functionals are important in the theory of polynomials and splines because they allow us to calculate the coefficients of an arbitrary polynomial or spline with respect to a fixed basis. They are also important because the dual basis may be simpler than the original basis. Thus it is sometimes easier to work in the dual space than in the primal space.

Paralleling two standard approaches to dual functionals for B-splines, we are going to develop two approaches to dual functionals for arbitrary polynomial bases: the blossoming method [25–27] and the de Boor–Fix formula [12]. Both techniques will be derived directly from our version of the Marsden identity (Eq. (2)).

3.1. *Blossoming*

Let $p(t)$ be a polynomial of degree n. The blossom or polar form of $p(t)$ is the unique, symmetric, multiaffine polynomial $B[p](u_1, \ldots, u_n)$ which reduces to $p(t)$ along the diagonal. That is, the blossom $B[p](u_1, \ldots, u_n)$ is independent of the order of the variables $u_1, \ldots, u_n$, each variable appears to at most the first power, and $B[p](t, \ldots, t) = p(t)$.

We shall give three explicit formulas for the blossom of a polynomial $p(t)$ depending on the information available for $p(t)$. Most commonly, if $p(t)$ is represented in monomial form, then

$$p(t) = \Sigma_k a_k t^k \Rightarrow B[p](u_1, \ldots, u_n) = \Sigma_k a_k s_k(u_1, \ldots, u_n) \Big/ \binom{n}{k}$$

where $s_k(u_1, \ldots, u_n)$ is the kth elementary symmetric function on $u_1, \ldots, u_n$. This formula establishes the existence and uniqueness of the blossom of $p(t)$.

Another approach, closely related to the de Boor–Fix formula for the dual functionals which we shall discuss in Subsection 3.3, is to let $\Psi[u_1,\ldots,u_n](t) = (u_1 - t)\cdots(u_n - t)$. Then

$$B[p](u_1,\ldots,u_n) = \Sigma_k\left[(-1)^{(n-k)}/n!\right]p^{(k)}(\tau)\Psi^{(n-k)}[u_1,\ldots,u_n](\tau). \tag{3}$$

Notice that the right hand side of Eq. (3) is independent of τ because its derivative with respect to τ is zero. Moreover, it is easy to check that the right hand side is symmetric, multiaffine, and by Taylor's theorem reduces to $p(t)$ when $u_i = t$ for $i = 1,\ldots,n$. A detailed discussion of this blossoming formula and its relationship to the de Boor–Fix dual functionals is given in [2]. A derivation of the blossoming form of the dual functionals from the de Boor–Fix formula is provided in [21].

Our first two approaches to blossoming assume that we know the derivatives of $p(t)$ at some parameter τ. Suppose instead that we know the roots of $p(t)$ as well as its highest order coefficient. That is, $p(t) = \{(-1)^n p^{(n)}(t)/n!\}(r_1 - t)\ldots(r_n - t)$. Then we can express the blossom of $p(t)$ using permanents [24].

The permanent of an $n \times n$ matrix $M = (M_{ij})$ is defined by

$$\text{Perm}(M) = \Sigma_\sigma m_{1\sigma(1)} \cdots m_{n\sigma(n)}.$$

Thus the permanent is similar to the determinant but without the alternating signs. Hence whereas the determinant is an antisymmetric function of its rows (columns), the permanent is a symmetric function of its rows (columns). Now the blossoming formula is simply

$$B[p](u_1,\ldots,u_n) = \left\{(-1)^n p^{(n)}(t)/(n!)^2\right\}\text{Perm}(r_i - u_j). \tag{4}$$

Again it is easy to check that the right hand side of Eq. (4) is symmetric, multiaffine, and reduces to $p(t)$ when $u_i = t$ for $i = 1,\ldots,n$.

This last blossoming formula requires some further explanation as well as a word of caution. First, the roots of the polynomial $p(t)$ may be complex. This anomaly causes no difficulty in the blossoming formula (Eq. (4)). Indeed for polynomials with real coefficients, complex roots come in conjugate pairs and the permanent in Eq. (4) evaluated at conjugate pairs is real.

More troublesome are polynomials with roots at infinity—that is, polynomials of degree $m < n$. In this case to apply Eq. (4), we must homogenize the permanent with respect to the roots, replace the coefficient $(-1)^n p^{(n)}(t)/(n!)^2$ by $(-1)^m p^{(m)}(t)/m!n!$, replace the finite roots r_i by $(r_i, 1)$, and replace the $n - m$ roots at infinity by $n - m$ copies of

$\delta = (1,0)$. With these alterations, Eq. (4) remains valid. From here on whenever a polynomial has roots at infinity, we shall adopt the preceding conventions without further comment; however, for simplicity, all our theorems and formulas are stated only for polynomials with finite roots.

Similarly, whenever we need to evaluate a blossom at an infinite parameter value, we shall first homogenize the blossom, or equivalently replace the multiaffine blossom by the multilinear blossom [25], and then replace the infinite value by $\delta = (1,0)$. In addition, whenever $p(t)$ has $n - m$ roots at infinity, we must replace the leading coefficient $(-1)^n p^{(n)}(t)/n!$ by $(-1)^m p^{(m)}(t)/m!$. Again, we shall adopt the preceding conventions without further comment, though for simplicity all our theorems and formulas are stated only for blossoms evaluated at finite values.

Now we can prove the following interesting identity which we will have occasion to apply in Section 5.

THEOREM 3.1. *Let $p(t)$ be a polynomial of degree n with roots $P_1, \dots, P_n$, and let $q(t)$ be a polynomial of degree n with roots $Q_1, \dots, Q_n$. Then*

$$q^{(n)}(t) B[p](Q_1, \dots, Q_n) = (-1)^n p^{(n)}(t) B[q](P_1, \dots, P_n).$$

Proof. By Eq. (4) we have

$$B[p](Q_1, \dots, Q_n) = \{(-1)^n p^{(n)}(t)/(n!)^2\} \mathrm{Perm}(P_i - Q_j)$$

$$B[q](P_1, \dots, P_n) = \{(-1)^n q^{(n)}(t)/(n!)^2\} \mathrm{Perm}(Q_i - P_j)$$

from which the desired result easily follows. Q.E.D

For further information on blossoms and polar forms as well as other explicit formulas for the blossom, see [27].

We can generate various useful extensions of Marsden's identity by blossoming both sides of Eq. (2) with respect to either t or x or both. Starting with Marsden's identity, this approach yields the identities

$$(x - t)^n = \Sigma_i d_i(t) b_i(x) \tag{2}$$

$$(x - t_1) \cdots (x - t_n) = \Sigma_i B[d_i](t_1, \dots, t_n) b_i(x) \tag{5}$$

$$(x_1 - t) \cdots (x_n - t) = \Sigma_i B[b_i](x_1, \dots, x_n) d_i(t) \tag{6}$$

$$(1/n!)\mathrm{Perm}(x_j - t_k) = \Sigma_i B[d_i](t_1, \dots, t_n) B[b_i](x_1, \dots, x_n), \tag{7}$$

where in Eq. (7) we have used the fact that the blossom of $(x_1 - t) \cdots (x_n - t)$ evaluated at $t_1, \dots, t_n$ is $(1/n!)\mathrm{Perm}(x_j - t_k)$. By Eq. (4), the blossoms on the right hand side of Eqs. (5)–(7) are also constants times permanents, so these formulas are actually identities involving permanents. Notice too that all of these identities are essentially equivalent, and that each identity holds for all pairs of dual bases. For B-splines these formulas appear in [34] where they were derived by blossoming the original version of Marsden's identity.

3.2. *Blossoming and Dual Functionals*

Blossoming can be used to find the B-spline coefficients of any spline. Indeed, if $\{B_{in}(t)\}$ are the B-splines of degree n associated with the knot vector $\{t_i\}$ and if $S(t)$ is a spline of degree n with the same knots, then there are constants c_i such that

$$S(t) = \Sigma_i c_i B_{in}(t).$$

Moreover the coefficients c_i are given by the formula

$$c_i = B[S](t_{i+1}, \ldots, t_{i+n}),$$

where the blossom of S on the right hand side is the blossom of the polynomial which represents S on any interval affected by c_i [25–27]. In this sense blossoming provides the dual functionals for the B-splines since it furnishes a technique for calculating the B-spline coefficients of any spline. Notice that to find c_i, we blossom $S(t)$ and evaluate at the roots of $\Psi_{in}(t)$, the function that multiplies $B_{in}(t)$ in the original version of Marsden's identity (Eq. (1)). We now extend this blossoming formula for the B-spline coefficients to arbitrary polynomial bases.

LEMMA 3.2. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases, and let $r_{j1}, \ldots, r_{jn}$ be the roots of $d_j(t)$. Then*

$$\begin{aligned} B[b_i](r_{j1}, \ldots, r_{jn}) &= (-1)^n n!/d_j^{(n)}(t), \qquad & i = j \\ &= 0, & i \neq j. \end{aligned}$$

Proof. From Eq. (6)

$$(r_{j1} - t) \cdots (r_{jn} - t) = \Sigma_i B[b_i](r_{j1}, \ldots, r_{jn}) d_i(t).$$

Since $r_{j1}, \ldots, r_{jn}$ are the roots of $d_j(t)$, the left hand side is just a constant multiple $d_j(t)$. In fact, we have

$$\left\{(-1)^n n!/d_j^{(n)}(t)\right\} d_n(t) = \Sigma_i B[b_i](r_{j1}, \ldots, r_{jn}) d_i(t).$$

But $d_0(t), \ldots, d_n(t)$ is a polynomial basis; therefore the coefficients of $d_i(t)$ on the left hand side and the right hand side must be identical, so the result follows. Q.E.D.

THEOREM 3.3. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases, and let $r_{j1}, \ldots, r_{jn}$ be the roots of $d_j(t)$. If $p(x)$ is any polynomial in x of degree n, then*

$$p(x) = \Sigma_i c_i b_i(x), \qquad \text{where } c_j = \left\{(-1)^n d_j^{(n)}(t)/n!\right\} B[p](r_{j1}, \ldots, r_{jn}).$$

That is, the coefficient of $b_j(x)$ is given by a constant times the blossom of $p(x)$ evaluated at the roots of $d_j(t)$.

Proof. Since $b_0(x), \ldots, b_n(x)$ is a polynomial basis, there exist constants $c_0, \ldots, c_n$ such that

$$p(x) = \Sigma_i c_i b_i(x).$$

Blossoming both sides with respect to x, evaluating at $r_{j1}, \ldots, r_{jn}$, and applying Lemma 3.2, we obtain

$$B[p](r_{j1}, \ldots, r_{jn}) = c_j(-1)^n n!/d_j^{(n)}(t)$$

and the result follows by solving for c_j. Q.E.D.

By Theorem 3.3, given a polynomial basis $b_0(x), \ldots, b_n(x)$ there exist constants $\gamma_0, \ldots, \gamma_n$ and parameters $s_{j1}, \ldots, s_{jn}$ such that if $p(x) = \Sigma_i c_i b_i(x)$, then

$$c_j = \gamma_j B[p](s_{j1}, \ldots, s_{jn}).$$

We now show that the constants $\gamma_0, \ldots, \gamma_n$ and the parameters $s_{j1}, \ldots, s_{jn}$ are unique.

THEOREM 3.4. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases, and let $r_{j1}, \ldots, r_{jn}$ be the roots of $d_j(t)$. Suppose that there exist constants $\gamma_0, \ldots, \gamma_n$ and parameters $s_{j1}, \ldots, s_{jn}$ such that*

$$p(x) = \Sigma_i c_i b_i(x) \qquad \text{implies} \qquad c_j = \gamma_j B[p](s_{j1}, \ldots, s_{jn}).$$

Then

$$\gamma_j = (-1)^n d_n^{(n)}(t)/n! \qquad \text{and} \qquad \{s_{j1}, \ldots, s_{jn}\} = \{r_{j1}, \ldots, r_{jn}\}.$$

Proof. Consider the polynomial $p(x) = (x - t)^n$. Since $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ are dual polynomial bases, it follows by Marsden's identity (Eq. (2)) that

$$(x - t)^n = \Sigma_i d_i(t) b_i(x).$$

Now by hypothesis

$$d_j(t) = \gamma_j B[p](s_{j1}, \ldots, s_{jn}).$$

Therefore for all t

$$\left\{(-1)^n d_j^{(n)}(t)/n!\right\}(r_{j1} - t) \cdots (r_{jn} - t) = \gamma_j (s_{j1} - t) \cdots (s_{jn} - t)$$

so the result follows. Q.E.D.

The constant factors $(-1)^n d_j^{(n)}(t)/n!$ that appear in Theorems 3.3, 3.4 are an annoyance. Indeed if $\operatorname{Deg}\{d_j(t)\} = m < n$, we must replace $(-1)^n d_j^{(n)}(t)/n!$ by $(-1)^m d_j^{(m)}(t)/m!$. In addition, we must homogenize the blossom of $p(x)$ and replace the $n - m$ roots at infinity by $n - m$ copies of $\delta = (1, 0)$ (see Subsection 3.1). However, when the basis functions $b_0(x), \ldots, b_n(x)$ are normalized to sum to one, then by Theorem 2.3 these factors disappear and we have the following elegant result.

THEOREM 3.5. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases, and let $r_{j1}, \ldots, r_{jn}$ be the roots of $d_j(t)$. If $\Sigma_i b_i(x) = 1$, then for any polynomial $p(x)$ of degree n*

$$p(x) = \Sigma_i c_i b_i(x), \qquad \text{where} \qquad c_j = B[p](r_{j1}, \ldots, r_{jn}).$$

That is, the coefficient of $b_j(x)$ is given by the blossom of $p(x)$ evaluated at the roots of $d_j(t)$.

Proof. This result follows immediately from Theorem 3.3 and Lemma 2.2. Q.E.D.

3.3. *The de Boor–Fix Formula*

The blossoming form of the dual functionals requires us to calculate the roots of the dual basis. But even for relatively low degree polynomials, these roots may be hard to compute. Fortunately, only symmetric functions in the roots are actually essential for the dual functionals. The coefficients of the dual basis with respect to the Taylor expansion are also symmetric functions of the roots. Moreover by applying the techniques in the proof of Theorem 2.1, these coefficients can be computed using only linear algebra. Therefore in the formulas for the dual functionals it would be desirable to replace the roots by the derivatives of the dual basis. This we now proceed to do by adopting an alternative approach to the construction of the dual functionals analogous to the de Boor–Fix construction of the dual functionals for the B-splines.

Let $f(t)$, $g(t)$ be polynomials of degree n and define

$$[f(t), g(t)]_n = \Sigma_k\left[(-1)^{(n-k)}/n!\right] f^{(k)}(\tau) g^{(n-k)}(\tau). \tag{8}$$

The operator $[f(t), g(t)]n$ is a bilinear form on the vector space of polynomials of degree n; that is,

(i) $[f(t), g(t)]_n$ is a constant independent of τ,

(ii) $[f(t), g(t)]_n$ is bilinear.

The first property follows because the derivative with respect to τ of the right hand side of Eq. (8) is zero. The second property is a consequence of the linearity of differentiation.

In addition to these two basic properties, the bilinear form $[f(t), g(t)]_n$ satisfies the following three important identities:

(iii) $[f(t), (x-t)^n]_n = f(x)$,

(iv) $[f(t), (x-t)^{n-k}]_n = \{(n-k)!/n!\} f^{(k)}(x)$,

(v) $[f(t), (u_1 - t) \cdots (u_n - t)]_n = B[f](u_1, \ldots, u_n)$.

Properties (iii) and (iv) follow directly from Taylor's theorem or more simply by evaluating the right hand side of Eq. (8) at $\tau = x$; property (v) is simply a restatement of Eq. (3). Notice that properties (iii) and (iv) are special cases of property (v) with $(u_1, \ldots, u_n) = (x, \ldots, x)$ or $(u_1, \ldots, u_n) = (\delta, \ldots, \delta, x^*, \ldots, x^*)$. where $\delta = (1, 0)$ and $x^* = (x, 1)$.

Let $\{B_{in}(t)\}$ be the B-splines of degree n associated with the knot vector $\{t_i\}$, and let $\Psi_{in}(t) = (t_{i+1} - t) \cdots (t_{i+n} - t)$. Then the de Boor–Fix formula [12] is

$$\left[B_{in}(t), \Psi_{jn}(t)\right]_n = \delta_{ij}, \tag{9}$$

where $[B_{in}(t), \Psi_{jn}(t)]_n$ is evaluated at any parameter τ such that $t_{j+1} < \tau < t_{j+n}$. (Strictly speaking we have only defined $[f(t), g(t)]_n$ for polynomials. However, since $B_{in}(t)$ is a piecewise polynomial, there is certainly no difficulty as long as we do not evaluate $[B_{in}(t), \Psi_{jn}(t)]_n$ at a knot. On the other hand, at a knot, $\Psi_{jn}(t)$ has a zero of multiplicity at least as high as the discontinuity in the derivatives of $B_{in}(t)$. Hence the terms on the right hand side of Eq. (8) where the derivatives of $B_{in}(t)$ jump are annihilated by the corresponding derivative of $\Psi_{jn}(t)$. Thus $[B_{in}(t), \Psi_{jn}(t)]_n$ is well-defined for $t_{j+1} < \tau < t_{j+n}$.) If $S(t)$ is a spline of degree n with knots $\{t_i\}$, then certainly

$$S(t) = \Sigma_i c_i B_{in}(t).$$

Now it follows immediately from Eq. (9) and the bilinearity of $[f(t), g(t)]_n$

that the coefficients c_i are given by the formula

$$c_j = \left[S(t), \Psi_{jn}(t)\right].$$

In this sense the de Boor–Fix formula provides the dual functionals for the B-splines since it furnishes an alternative technique for calculating the B-spline coefficients. It is this formula for the dual functionals that we shall now extend to arbitrary polynomial bases.

THEOREM 3.6. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases. Then*

$$\left[b_i(t), d_j(t)\right]_n = \delta_{ij}.$$

Proof. By Marsden's identity (Eq. (2)) and properties (ii) and (iii) of $[f(t), g(t)]_n$

$$\begin{aligned} b_i(x) &= \left[b_i(t), (x-t)^n\right]_n = \left[b_i(t), \Sigma_j d_j(t) b_j(x)\right]_n \\ &= \Sigma_j \left[b_i(t), d_j(t)\right]_n b_j(x). \end{aligned}$$

Since the polynomials $b_0(x), \ldots, b_n(x)$ are linearly independent, the coefficients of $b_j(x)$ on both sides of this equation must be identical, so the result follows. Q.E.D.

THEOREM 3.7. *Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases. If $p(x)$ is any polynomial in x of degree n, then*

$$p(x) = \Sigma_i c_i b_i(x), \qquad \textit{where} \qquad c_j = \left[p(t), d_j(t)\right]_n.$$

Proof. This result follows immediately from Theorem 3.6 and the bilinearity of $[f(t), g(t)]_n$. Q.E.D.

Theorem 3.7 extends the de Boor–Fix form of the dual functionals to arbitrary polynomial bases. This result achieves our goal of replacing the roots of the dual basis by the more readily accessible Taylor coefficients in the formulas for the dual functionals.

Finally combining the blossoming and de Boor–Fix forms of the dual functionals, we have the following corollary.

COROLLARY 3.8. *Let $r_{j1}, \ldots, r_{jn}$ be the roots of $d_j(t)$. If $p(x)$ is any polynomial in x of degree n, then*

$$\left\{(-1)^n d_n^{(n)}(t)/n!\right\} B[p](r_{j1}, \ldots, r_{jn}) = \left[p(t), d_j(t)\right]_n.$$

Proof. This equality follows immediately from Theorem 3.3 and Theorem 3.7. Notice too that this result is essentially equivalent to the blossoming formula in Eq. (3). Q.E.D.

4. Change of Basis

We now apply the dual functionals developed in Section 3 to derive change of basis algorithms for converting B-spline curves to piecewise polynomials with respect to arbitrary polynomial bases. We shall also use the dual functionals to extend a duality principle for transformations from progressive and Polya bases to arbitrary pairs of dual polynomial bases.

4.1. *The Oslo Algorithm*

The original version of the Oslo algorithm was a knot insertion procedure for B-spline curves [16]. However in [7] it was shown that a local variant of the Oslo algorithm can also be used to convert a polynomial from any progressive basis to any other progressive basis. Here we shall show how to apply a local variant of this procedure to convert from any progressive basis to any arbitrary polynomial basis. Thus we shall establish that the Oslo algorithm can be used to convert a B-spline curve to a piecewise polynomial with respect to any arbitrary polynomial basis.

A degree n polynomial basis $b_0(x), \dots, b_n(x)$ is said to be *progressive* if there are $2n$ parameters $t_1, t_2, \dots, t_{2n}$, $t_{j+n} \neq t_i$ for $1 \leq i \leq j \leq n$, called *knots* such that for any polynomial $p(x)$

$$p(x) = \Sigma_i c_i b_i(x) \Leftrightarrow c_i = B[p](t_{i+1}, \dots, t_{i+n}).$$

That is, the coefficients of $p(x)$ are given by its blossom evaluated at consecutive knots. By Theorem 3.5 this condition is equivalent to the dual basis $d_0(t), \dots, d_n(t)$ being given by

$$d_i(t) = (t_{i+1} - t) \cdots (t_{i+n} - t), \qquad i = 0, 1, \dots, n.$$

This dual basis is sometimes called a Polya basis [3, 5, 6]. The Bernstein basis is the progressive basis with knots $(0, \dots, 0, 1, \dots, 1)$, and the monomial basis is the progressive basis with knots $(0, \dots, 0, \delta, \dots, \delta)$ [7, 20]. Any B-spline basis is locally (over a single knot interval) a progressive basis, but not every progressive basis can be extended to a B-spline [7].

Given the $n+1$ values $B[p](t_1, \dots, t_n), \dots, B[p](t_{n+1}, \dots, t_{2n})$, we can compute any blossom value $B[p](u_1, \dots, u_n)$ recursively by the following algorithm due to Ramshaw [25].

Recursive Algorithm for Computing Blossom Values.

Let

$$B_{i0} = B[p](t_{i+1}, \ldots, t_{i+n}) \qquad i = 0, \ldots, n$$

$$B_{ir}(u_1, \ldots, u_r) = [(t_{i+n+1-r} - u_r)/(t_{i+n+1-r} - t_i)]\, B_{i-1r-1}(u_1, \ldots, u_{r-1}) + [(u_r - t_i)/(t_{i+n+1-r} - t_i)]\, B_{ir-1}(u_1, \ldots, u_{r-1})$$

where $r = 1, \ldots, n$ and $i = r, \ldots, n$.

Then

$$B_{ir}(u_1, \ldots, u_r) = B[p](u_1, \ldots, u_r, t_{i+1}, \ldots, t_{i+n-r})$$

$$B_{nn}(u_1, \ldots, u_n) = B[p](u_1, \ldots, u_n).$$

Notice that this algorithm simply inserts u_i at the ith level of the recursion. This algorithm can be derived easily from the symmetry and multiaffine properties of the blossom [25]. We illustrate this procedure for cubic polynomials in Fig. 1. When the progressive basis is the restriction of a B-spline basis to the knot interval $[t_n, t_{n+1}]$ and the new knots $u_1, \ldots, u_n$

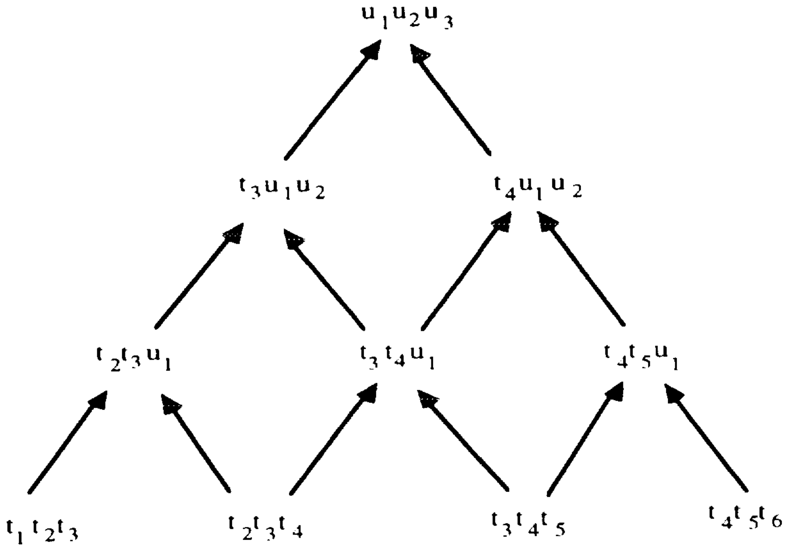

FIG. 1. A schematic view of the Oslo algorithm for cubic polynomials. Here a triple uvw represents the blossom value $B[p](u, v, w)$, and the arrows represent the affine combinations in the recurrence. Notice that this algorithm simply inserts u_i at the ith level of the recursion.

lie in this interval, then this algorithm is a variant of the Oslo algorithm [19].

Now given the coefficients of a polynomial $p(x)$ with respect to the progressive basis $b_0(x), \ldots, b_n(x)$, we can apply this recursive algorithm $n+1$ times to find the coefficients of $p(x)$ relative to any other basis because by Theorem 3.3 these coefficients are (up to constant multiples) the blossom of $p(x)$ evaluated at the roots of the dual basis. Thus the Oslo algorithm is a general change of basis procedure from progressive to arbitrary polynomial bases provided that we know the roots of the dual basis. For example, we can apply this variant of the Oslo algorithm to convert a polynomial from the Bernstein basis to any other polynomial basis by replacing the parameter t by different roots of the dual basis on each level of the de Casteljau algorithm. More generally, we can use the Oslo algorithm to convert a B-spline curve to a piecewise polynomial with respect to any arbitrary polynomial basis.

4.2. *Dual Transformations*

The change of basis transformation between any pair of progressive bases is known to be the transpose of the transformation between the corresponding dual bases in the opposite direction [3, 5, 20]. Here we generalize this duality principle to arbitrary pairs of dual polynomial bases.

THEOREM 4.1. *Let $\{b_k(x), d_k(t)\}$ and $\{B_k(x), D_k(t)\}$ be two pairs of dual polynomial bases of degree n. Then*

$$b_i(x) = \Sigma_j M_{ij} B_j(x) \Leftrightarrow D_i(t) = \Sigma_j M_{ji} d_j(t).$$

Proof. Let $b_i(x) = \Sigma_j M_{ij} B_j(x)$ and let $D_i(t) = \Sigma_j N_{ij} d_j(t)$. Then by Theorem 3.5

$$M_{ik} = \left[\Sigma_i M_{ij} B_j(x), D_k(x)\right]_n = \left[b_i(x), D_k(x)\right]_n$$

$$N_{ik} = \left[b_k(x), \Sigma_{ij} N_{ij} d_j(t)\right]_n = \left[b_k(x), D_i(x)\right]_n.$$

Hence $N_{ik} = M_{ki}$. Q.E.D.

We can give an alternative proof of this result using blossoming by applying Theorem 3.3 and Eq. (4). Let $R_{k1}, \ldots, R_{kn}$ be the roots of $D_k(x)$, and let $r_{i1}, \ldots, r_{in}$ be the roots of $b_i(x)$. Then

$$\begin{aligned} M_{ik} &= \left\{(-1)^n D_k^{(n)}(t)/n!\right\} B[b_i](R_{k1}, \ldots, R_{kn}) \\ &= \left\{b_i^{(n)}(t) D_k^{(n)}(t)/(n!)^3\right\} \text{Perm}(r_{ip} - R_{kq}) \end{aligned}$$

and similarly

$$N_{ik} = \{b_k^{(n)}(t)/n!\}B[D_i](r_{k1},\ldots,r_{kn})$$
$$= \{(-1)^n D_i^{(n)}(t)b_k^{(n)}(t)/(n!)^3\}\mathrm{Perm}(R_{iq} - r_{kp}).$$

Hence again we conclude that $N_{ik} = M_{ki}$.

Since the change of basis matrix is also the matrix which transforms the coefficients of any polynomial with respect to the one basis to the coefficients with respect to the other basis, we have the following result.

COROLLARY 4.2. *Let $\{b_k(x), d_k(t)\}$ and $\{B_k(x), D_k(t)\}$ be two pairs of dual polynomial bases of degree n. If M is the matrix which transforms the coefficients of a polynomial with respect to the basis $\{b_k(x)\}$ to the coefficients with respect to the basis $\{B_k(x)\}$, then M^T is the matrix which transforms the coefficients of a polynomial with respect to the basis $\{D_k(x)\}$ to the coefficients with respect to the basis $\{d_k(x)\}$. That is, the transformation between two polynomial bases is equivalent to the transpose of the transformation between the corresponding dual bases in the opposite direction.*

The Oslo algorithm can be used to find the change of basis matrix from any progressive basis to any polynomial basis. Therefore, by taking the transpose of this transformation matrix, we can use the Oslo algorithm to find the change of basis matrix from any polynomial basis to any Polya (progressive dual) basis. In particular, since the Bernstein basis is self dual [7, 20], we can use this approach to find the change of basis matrix in either direction between the Bernstein basis and any polynomial basis.

5. RATIONAL RECURRENCES FOR BLOSSOMING, EVALUATION, AND DIFFERENTIATION

In Subsection 4.1 we saw that one variant of the Oslo algorithm is a recursive procedure for evaluating the blossom of a polynomial given its coefficients with respect to a progressive basis. Using this variant of the Oslo algorithm to compute the blossom along the diagonal provides us with a recursive evaluation procedure for polynomials in progressive form. For B-spline curves this recursive procedure is the de Boor evaluation algorithm [10]. We can also compute derivatives recursively using the Oslo algorithm. Let $p(t)$ be a polynomial of degree n. If we replace the multiaffine blossom by the multilinear blossom and evaluate this blossom at $(\delta,\ldots,\delta,t^*,\ldots,t^*)$, where $t^* = (t,1)$, $\delta = (1,0)$, and δ is repeated k time, then it can be shown that [7, 27]

$$\{(n-k)!/n!\}p^{(k)}(t) = B[p](\delta,\ldots,\delta,t^*,\ldots,t^*). \qquad (10).$$

(We shall give a simple derivation of this formula at the end of this section.) Thus the Oslo algorithm can be used to compute the derivatives of $p(t)$ given its coefficients with respect to a progressive basis.

We shall now extend these recursive procedures which are well-known for B-spline and progressive bases to arbitrary polynomial bases. One difference we shall find is that for B-spline and progressive bases these recursive procedures are linear in the parameters at each stage of the recurrence while for arbitrary polynomial bases these algorithms are rational of high degree. Another difference is that for B-spline and progressive bases these recursive procedures consist of real affine combinations while for arbitrary polynomial bases the affine combinations may be complex.

Let $b_0(x), \ldots, b_n(x)$ and $d_0(t), \ldots, d_n(t)$ be dual polynomial bases. The key to our approach is to work with the dual basis rather than with the primary basis. To simplify this discussion, we shall assume that the primary basis $b_0(x), \ldots, b_n(x)$ is normalized to sum to one so that by Theorem 2.3 the dual basis functions $d_0(t), \ldots, d_n(t)$ have the form

$$d_j(t) = (r_{j1} - t) \cdots (r_{jn} - t), \qquad j = 0, 1, \ldots, n.$$

To begin, we will describe a recurrence that computes the polynomial $u(t) = (u_1 - t) \cdots (u_n - t)$ from the polynomials $d_0(t), \ldots, d_n(t)$.

The Polynomial Recurrence (Root Insertion Algorithm).

Let

$$B_{i0}(t) = d_i(t) \qquad i = 0, 1, \ldots, n$$

$$\begin{aligned} B_{ir}[u_1, \ldots, u_r](t) &= c_{ir-1}(u_1, \ldots, u_r) B_{ir-1}[u_1, \ldots, u_{r-1}](t) - c_{i-1r-1}(u_1, \ldots, u_r) \\ &\quad \times B_{i-1r-1}[u_1, \ldots, u_{r-1}](t) \end{aligned}$$

where

$$\begin{aligned} c_{ir-1}(u_1, \ldots, u_r) &= B_{i-1r-1}[u_1, \ldots, u_{r-1}](u_r) / \{B_{i-1r-1}[u_1, \ldots, u_{r-1}](u_r) \\ &\qquad - B_{ir-1}[u_1, \ldots, u_{r-1}](u_r)\} \\ c_{i-1r-1}(u_1, \ldots, u_r) &= B_{ir-1}[u_1, \ldots, u_{r-1}](u_r) / \{B_{i-1r-1}[u_1, \ldots, u_{r-1}](u_r) \\ &\qquad - B_{ir-1}[u_1, \ldots, u_{r-1}](u_r)\} \end{aligned}$$

for $r = 1, \ldots, n$, $i = r, \ldots, n$.

Then

$$B_{nn}[u_1, \ldots, u_n](t) = u(t).$$

The key to establishing this recurrence is to observe that by construction u_r is a root of $B_{ir}[u_1, \ldots, u_r](t)$ for $i = r, \ldots, n$. Moreover if $u_1, \ldots, u_{r-1}$ are roots of $B_{jr-1}[u_1, \ldots, u_{r-1}](t)$ for $j = r-1, \ldots, n$, then $u_1, \ldots, u_{r-1}$ are also roots of $B_{ir}[u_1, \ldots, u_r](t)$. Therefore it follows by induction on r that $u_1, \ldots, u_n$ are roots of $B_{nn}[u_1, \ldots, u_n](t)$. Thus the polynomial recurrence is actually a root insertion algorithm since it inserts the root u_r at the r^{th} stage of the recurrence. Finally, it also follows easily by induction on r that the coefficient of t^n in $B_{ir}[u_1, \ldots, u_r](t)$ is $(-1)^n$—this is the reason for the normalizing constant in the denominator of $c_{ir-1}(u_1, \ldots, u_r)$. Hence since $\mathrm{Deg}\{B_{ir}[u_1, \ldots, u_r](t)\} = n$, we conclude that

$$B_{nn}[u_1, \ldots, u_n](t) = u(t).$$

Notice that if some of the roots u_r are complex, then the affine combinations in the polynomial recurrence—that is, the constants $c_{ir-1}(u_1, \ldots, u_r)$—may also be complex. Nevertheless, if $u(t)$ is a polynomial with real coefficients, the result of the polynomial recurrence is a polynomial with real coefficients.

The polynomials $B_{ir}[u_1, \ldots, u_r](t)$, $i = r, \ldots, n$, at the rth level of the recurrence are linearly independent. This property also follows by induction on r. Indeed, it is certainly true for $r = 0$. Moreover, the polynomials at the rth level are simple two term combinations of the polynomials on the $(r-1)$st level. Thus it is easy to show by induction on i that

$$\Sigma_i a_i B_{ir}[u_1, \ldots, u_r](t) = 0 \Rightarrow a_i = 0.$$

We have described the polynomial recurrence for the general setting, but many special situations may arise. We now discuss what to do in these special cases.

First, if the leading coefficient of $d_i(t)$ is not $(-1)^n$, then we set

$$B_{i0}(t) = \left\{(-1)^m / d_i^{(m)}(t)\right\} d_i(t),$$

where m is the degree of $d_i(t)$.

Second, if the denominator of $c_{ir-1}(u_1, \ldots, u_r)$ is zero, then we must replace

$$B_{i-1r-1}[u_1, \ldots, u_{r-1}](u_r) - B_{ir-1}[u_1, \ldots, u_{r-1}](u_r) = 0$$

by

$$B_{i-1r-1}{}^{(p)}[u_1, \ldots, u_{r-1}](u_r) - B_{ir-1}{}^{(p)}[u_1, \ldots, u_{r-1}](u_r),$$

where p is the first integer such that

$$B_{i-1r-1}^{(p)}[u_1,\ldots,u_{r-1}](u_r) - B_{ir-1}^{(p)}[u_1,\ldots,u_{r-1}](u_r) \neq 0.$$

Such an integer p must exist because we know that

$$B_{i-1r-1}[u_1,\ldots,u_{r-1}](t) \neq B_{ir-1}[u_1,\ldots,u_{r-1}](t)$$

since all the polynomials at the $(r-1)$st level are linearly independent.

Finally, if

$$\text{Deg}\{B_{i-1r-1}[u_1,\ldots,u_{r-1}](t)\} < \text{Deg}\{B_{ir-1}[u_1,\ldots,u_{r-1}](t)\}$$

then we must replace the denominator

$$B_{i-1r-1}[u_1,\ldots,u_{r-1}](u_r) - B_{ir-1}[u_1,\ldots,u_{r-1}](u_r)$$
$$\text{by } B_{i-1r-1}[u_1,\ldots,u_{r-1}](u_r).$$

If, in addition,

$$B_{i-1r-1}[u_1,\ldots,u_{r-1}](u_r) = 0,$$

then simply set

$$B_{ir}[u_1,\ldots,u_{r-1}](t) = B_{i-1r-1}[u_1,\ldots,u_{r-1}](t).$$

Notice that this recurrence is rational in the parameters $u_1,\ldots,u_n$. However, even though the polynomials $B_{ir-1}[u_1,\ldots,u_{r-1}](u_r)$ are, in general, degree n in u_r, because $B_{i-1r-1}[u_1,\ldots,u_{r-1}](t)$ and $B_{ir-1}[u_1,\ldots,u_{r-1}](t)$ share $r-1$ common factors, the rth level of this recurrence is only degree $n-r+1$ in the numerator and degree $n-r$ in the denominator. The degree is one lower in the denominator because the highest order terms cancel. We illustrate the entire algorithm for cubic polynomials in Fig. 2, and we focus in on one step of this algorithm in Fig. 3.

Notice that the algorithm shown in Fig. 2 simply inserts the factor $(u_i - t)$ at the ith level of the recursion.

Now this polynomial recurrence can be used to generate recursive procedures for blossoming, evaluating, and differentiating polynomials represented with respect to an arbitrary polynomial basis. There are two approaches to deriving these algorithms from the polynomial recurrence, both based on dual functionals: the first uses the de Boor–Fix formula, the second the blossoming method. We shall look at each approach in turn.

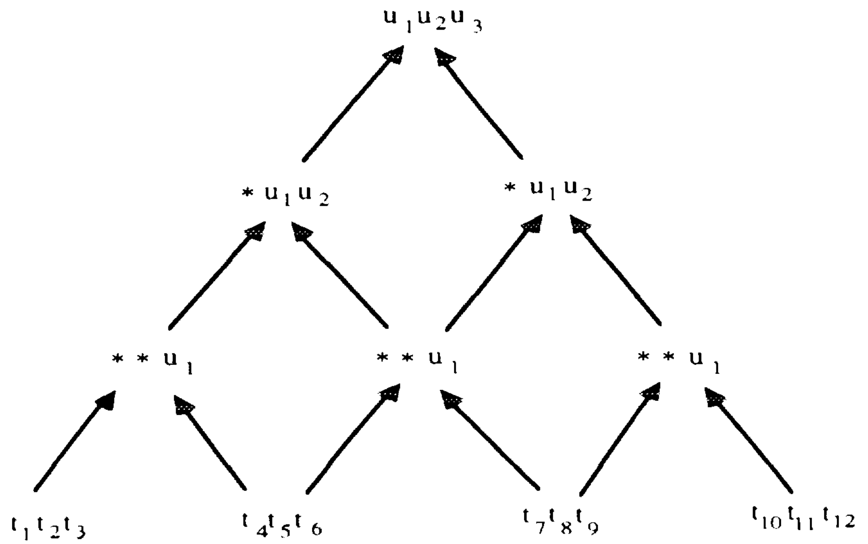

FIG. 2. A schematic view of the polynomial recurrence (root insertion algorithm) for cubic polynomials. Here a triple uvw represents the polynomial $(u-t)(v-t)(w-t)$, asterisks represent unknown linear factors, and the arrows represent the rational coefficients, c_{ir}, in the recurrence.

First, consider the de Boor–Fix formula. Let $p(t)=\sum_i c_i b_i(t)$ be a polynomial of degree n. Then by Eq. (3)

$$B[p](u_1,\ldots,u_n)=[p(t),(u_1-t)\cdots(u_n-t)]_n.$$

Now let us apply the operator $[p(t),]_n$ to every element $B_{ir}[u_1,\ldots,u_r](t)$ in the polynomial recurrence. Since $[f(t),g(t)]_n$ is bilinear, the polynomial

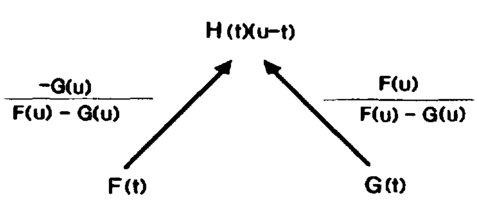

FIG. 3. One step of the polynomial recurrence (root insertion algorithm). The polynomial at the apex is computed by multiplying the polynomials at the base by the expressions along the corresponding arrows and adding the results. When $t=u$, the result is zero, so $(u-t)$ must be a factor of the polynomial at the apex. Moreover common roots of $F(t)$ and $G(t)$ are also roots of $H(t)$. Notice that the labels along the arrows are rational functions of u.

recurrence now becomes a recursive procedure for computing $[p(t), u(t)]_n$ from $[p(t), d_0(t)]_n, \dots, [p(t), d_n(t)]_n$. But by Theorem 3.7

$$[p(t), d_j(t)]_n = c_j$$

and by Eq. (3)

$$[p(t), u(t)]_n = B[p](u_1, \dots, u_n).$$

Therefore when the polynomials $d_0(t), \dots, d_n(t)$ at the base of the recurrence are replaced by the coefficients $c_0, \dots, c_n$ of $p(t)$, the polynomial recurrence becomes an algorithm for computing the blossom values $B[p](u_1, \dots, u_n)$ from the coefficients of $p(t)$.

We could also derive this blossoming recurrence directly by blossoming arguments. Let $P_1, \dots, P_n$ be the roots of $p(t)$. Now blossom every element $B_{ir}[u_1, \dots, u_r](t)$ in the polynomial recurrence and evaluate these blossoms at the roots $P_1, \dots, P_n$. Since blossoming is a linear operator, the polynomial recurrence now becomes a recursive procedure for computing $B[u](P_1, \dots, P_n)$ from $B[d_0](P_1, \dots, P_n), \dots, B[d_n](P_1, \dots, P_n)$. But it follows from Theorem 3.1 that

$$B[u](P_1, \dots, P_n) = \{n!/p^{(n)}(t)\}B[p](u_1, \dots, u_n)$$
$$B[d_j](P_1, \dots, P_n) = \{n!/p^{(n)}(t)\}B[p](r_{j1}, \dots, r_{jn}),$$

where $r_{j1}, \dots, r_{jn}$ are the roots of $d_j(t)$. Moreover by Theorem 3.5

$$B[p](r_{j1}, \dots, r_{jn}) = c_j.$$

Removing the common factor $n!/p^{(n)}(t)$, we find again that when the polynomials $d_0(t), \dots, d_n(t)$ at the base of the polynomial recurrence are replaced by the coefficients $c_0, \dots, c_n$ of $p(t)$, the polynomial recurrence becomes an algorithm for computing the blossom values $B[p](u_1, \dots, u_n)$ from the coefficients of $p(t)$.

Once we have established this blossoming recurrence, it is an easy matter to derive recursive procedures for evaluation and differentiation. If we replace u_i by x for $i = 1, \dots, n$, then the recurrence for computing $B[p](u_1, \dots, u_n)$ becomes a recurrence for computing $B[p](x, \dots, x) = p(x)$. Thus we get a recursive evaluation algorithm for $p(x)$. Similarly, if we homogenize the blossoming recurrence and replace $(u_1, \dots, u_n)$ by

$(\delta, \ldots, \delta, x^*, \ldots, x^*)$, then by Eq. (10) the recurrence for computing $B[p](u_1, \ldots, u_n)$ becomes a recurrence for computing $B[p](\delta, \ldots, \delta, x^*, \ldots, x^*) = \{(n-k)!/n!\}p^{(k)}(x)$. Homogenizing the blossoming recurrence and evaluating at $(\delta, \ldots, \delta, x^*, \ldots, x^*)$ is equivalent to setting

$$c_{ir-1}(u_1, \ldots, u_r) = (-1)^n/\{B_{i-1r-1}[\delta, \ldots, \delta](\delta) - B_{ir-1}[\delta, \ldots, \delta](\delta)\}$$

on the first k levels of the recurrence. Thus the derivative recurrence is often considerably simpler than the evaluation recurrence. Notice that, in general, these recurrences for blossoming, evaluation, and differentiation have complex-valued, high degree, rational coefficients c_{ir}. Nevertheless, at the final stage these algorithms collapse to real-valued polynomial functions.

The evaluation recurrence provides a recurrence for the basis functions $b_0(x), \ldots, b_n(x)$. If we set $c_{i0} = \delta_{ij}$ at the base of the evaluation recurrence, then the basis function $b_j(x)$ emerges at the apex. Thus $b_j(x)$ is the sum of all paths from the base to the apex of the triangle (see Fig. 2). We can also run this recurrence in reverse by placing a 1 at the apex of the triangle and reversing the arrows in Fig. 2. The basis functions $b_0(x), \ldots, b_n(x)$ will then emerge at the nth level of the recurrence at the base of the triangle. In a similar manner we can compute the blossoms or the derivatives of the basis functions.

Given the coefficients of a polynomial $p(x)$ with respect to any basis $b_0(x), \ldots, b_n(x)$, we can apply the blossoming recurrence to find the coefficients of $p(x)$ relative to any other basis because by Theorem 3.3 these new coefficients are (up to constant multiples) the blossom of $p(x)$ evaluated at the roots of the dual basis. Thus the blossoming recurrence can be used as a general change of basis procedure; from this perspective the blossoming recurrence is yet another generalization of the Oslo algorithm.

Because these recursive procedures for evaluation, differentiation, blossoming, and change of basis are high degree rational functions in the parameters, they may not be too useful in the general case. But when consecutive dual basis functions share common roots, these degrees may be substantially reduced. For B-spline and progressive bases, these algorithms reduce to the standard recursive procedures for evaluation, differentiation, blossoming, and change of basis. These standard procedures are linear at every stage of the recurrence precisely because consecutive polynomials in the dual basis share $n-1$ common roots.

One final observation about the origins of these recursive procedures for blossoming, evaluation, and differentiation. These three recursive procedures are actually consequences of the following three identities from Subsection 3.3:

(iii) $[p(t), (x - t)^n]_n = p(x) = B[p](x, \ldots, x)$,

(iv) $[p(t), (x - t)^{n-k}]_n = \{(n - k)!/n!\}p^{(k)}(x) = B[p](\delta, \ldots, \delta, x^*, \ldots, x^*)$,

(v) $[p(t), (u_1 - t) \cdots (u_n - t)]_n = B[p](u_1, \ldots, u_n)$.

Now our three recursive procedures have the following interpretations: To generate the blossoming recurrence, we place $(u_1 - t) \cdots (u_n - t)$ at the last stage of the polynomial recurrence and we insert the factor $u_i - t$ at the i^{th} level; to generate the evaluation recurrence, we place $(x - t)^n$ at the last stage of the polynomial recurrence and we insert the factor $x - t$ at each level; and to generate the derivative recurrence, we place $(x - t)^{n-k}$ at the last stage of the polynomial recurrence and we insert the factor 1 at k levels and the factor $x - t$ at $n - k$ levels. We then replace the dual basis functions $d_0(t), \ldots, d_n(t)$ at the base of the polynomial recurrence by the coefficients $c_0, \ldots, c_n$ of the polynomial $p(t)$ with respect to the basis $b_0(x), \ldots, b_n(x)$. These three recurrences can be derived, as above, by applying the operator $[p(t),]_n$ to every polynomial in the polynomial recurrence or by blossoming every polynomial in the polynomial recurrence and evaluating at the roots of $p(t)$.

Notice that we never actually need to compute the dual basis to find the coefficients of $p(t)$ relative to the basis b since

$$\text{coefficients} = B[p](\text{roots of } d) = B[d](\text{roots of } p).$$

But the last expression can be computed directly from b by running the Oslo algorithm for d (i.e., place b at the base of the triangle) and evaluating at the roots of p.

6. Examples of Dual Polynomial Bases

Many examples of dual polynomial bases are known. A progressive basis $b_0(x), \ldots, b_n(x)$ with knots $t_1, t_2, \ldots, t_{2n}$, $t_{j+n} \neq t_i$ for $1 \leq i \leq j \leq n$, has the dual Polya basis given by

$$d_i(t) = (t_{i+1} - t) \ldots (t_{i+n} - t), \qquad i = 0, 1, \ldots, n.$$

Any B-spline basis is locally (over a single knot interval) a progressive basis. The Bernstein basis is the progressive basis with knots $(0, \ldots, 0, 1, \ldots, 1)$, and the monomial basis is progressive with knots $(0, \ldots, 0, \delta, \ldots, \delta)$ [7, 20].

Another interesting dual progressive/Polya pair is the power basis and the Lagrange basis. Let

$$L_k(t) = \Pi_{j \neq k}(t - t_j)/\Pi_{j \neq k}(t_k - t_j), \qquad k = 0, 1, \ldots, n$$

be the Lagrange basis for the nodes $t_0, \ldots, t_n$, and let

$$P_k(x) = (x - t_k)^n, \qquad k = 0, 1, \ldots, n$$

be the power basis at $t_0, \ldots, t_n$. Then

$$(x - t)^n = \Sigma_k L_k(t) P_k(x)$$

since both sides of this equation are degree n polynomials in t which agree at the nodes $t_0, \ldots, t_n$. Notice that up to normalizing constants, the power and Lagrange dual bases are the progressive/Polya pair for the knot sequence $t_1, \ldots, t_n, t_0, \ldots, t_{n-1}$ [7, 20].

Along these same lines, let us consider some dual bases which are not progressive/Polya pairs. Let $t_0, \ldots, t_m$ be a sequence of real numbers. Associate with each node t_j a positive integer μ_j such that $\Sigma_j \mu_j = n + 1$. Let $H_{jk}(t)$, $j = 0, \ldots, m$, $k = 0, \ldots, \mu_j - 1$, be the degree n Hermite basis functions associated with the nodes $t_0, \ldots, t_m$ [14]. That is,

$$H_{jk}^{(l)}(t_i) = \delta_{ij}\delta_{kl}, \qquad l = 0, \ldots, \mu_j - 1.$$

Let $P_{jk}(x)$, $j = 0, \ldots, m$, $k = 0, \ldots, \mu_j - 1$, be the generalized power basis defined by

$$P_{jk}(x) = \{(-1)^k n!/(n - k)!\}(x - t_j)^{n-k}.$$

Then again

$$(x - t)^n = \Sigma_{jk} H_{jk}(t) P_{jk}(x)$$

since both sides of this equation are degree n polynomials in t which agree at the nodes t_j with multiplicities μ_j. In general, the Hermite basis $H_{jk}(t)$ is not a Polya basis so the generalized power basis is not a progressive basis.

Another interesting example is Ball's cubic basis [1]:

$$b_0(x) = (1 - x)^2, \qquad b_1(x) = 2x(1 - x)^2,$$
$$b_2(x) = 2x^2(1 - x), \qquad b_3(x) = x^2.$$

The dual to Ball's cubic basis is

$$d_0(t) = -t^3, \qquad d_1(t) = t^2(3/2 - t),$$
$$d_2(t) = t^2(-1/2 - t), \qquad d_3(t) = (1 - t)^3.$$

Since the roots of the dual basis are readily available and since Ball's basis is normalized to sum to one, the coefficients of any cubic polynomial $p(x)$ relative to Ball's cubic basis are simply

$$P_0 = B[p](0,0,0), \qquad P_1 = B[p](0,0,3/2),$$
$$P_2 = B[p](-1/2,1,1), \qquad P_3 = B[p](1,1,1).$$

Notice that these formulas for the Ball coefficients differ only slightly from the formulas for the Bernstein coefficients.

As one last example, let us consider the cubic Chebysheff polynomials [14]:

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_2(x) = 2x^2 - 1, \qquad T_3(x) = 4x^3 - 3x.$$

Since the Chebysheff basis is triangular, we know by Theorem 2.4 that the dual basis in reverse order is also triangular. Thus we can calculate this dual basis by solving a triangular system of equations. Explicitly, the cubic basis dual to the cubic Chebysheff basis is given by

$$D_0(t) = -t(2t^2 + 3)/2, \qquad D_1(t) = 3(4t^2 + 1)/4,$$
$$D_2(t) = -3t/2, \qquad D_3(t) = 1/4.$$

The Chebysheff basis is orthogonal relative to the inner product [14]

$$(f, g) = \int_{-1}^{1} \left\{ f(x)g(x)/(1 - x^2)^{1/2} \right\} dx.$$

That is,

$$(T_i, T_j) = \delta_{ij}.$$

Therefore the coefficients of any polynomial $p(x)$ relative to the Chebysheff basis are given by (p, T_k). On the other hand, we have shown that these coefficients are also given by a constant times the blossom of $p(x)$ evaluated at the roots of the dual basis. Thus we can calculate these integrals from the blossom of $p(x)$ and the roots of the dual basis. Alternatively, we could dispense entirely with the roots of the dual basis and simply apply the de Boor–Fix dual functionals which require only the derivatives of the dual basis. That is, by Theorem 3.7 we have

$$(p, T_k) = [p, D_k]_n$$

or equivalently in expanded form

$$\int_{-1}^{1} \left\{ p(x) T_k(x)/(1-x^2)^{1/2} \right\} dx = \Sigma_i \left[(-1)^{(n-i)}/n! \right] p^{(i)}(\tau) D_k^{(n-i)}(\tau).$$

This observation allows us to calculate definite integrals as finite sums by solving a triangular system of linear equations for the coefficients of $D_0(t), \ldots, D_n(t)$ since the monomial coefficients are essentially the derivatives at zero. Similar observations hold for other sets of orthogonal polynomials relative to different inner products.

Finally, we note that all the results we have derived so far for polynomial bases $\{b_j(t)\}$ are valid as well for locally linearly independent spline basis $\{M_i(t)\}$. In particular, Marsden's identity, the blossoming and de Boor–Fix forms of the dual functionals, the Oslo algorithm, and the recursive procedures for evaluating, differentiating, and blossoming all extend readily to locally linearly independent spline bases. Thus if $S(t)$ is a spline in the space spanned by $\{M_i(t)\}$, that is, if

$$S(t) = \Sigma_i c_i M_{in}(t),$$

then we can find the coefficients c_i by applying the blossoming or de Boor–Fix dual functionals on any interval affected by c_i. Similarly, there are local recursive procedures for evaluating, differentiating, and blossoming $S(t)$.

There exist many important examples of locally linearly independent spline bases. Consider, for instance, the space of piecewise polynomials determined by connection matrices [18]. If the connection matrices $\{C_i\}$ are totally positive [29], then the space of piecewise polynomials of degree n associated with the knot vector $\{t_i\}$ and the connection matrices $\{C_i\}$ has a locally linearly independent spline basis [4]. A global extension of Marsden's identity to such spaces is discussed in [4].

Under certain mild restrictions, the space of piecewise polynomials that are C^{n-1} under reparameterization—often called Beta-splines—is an example of a space of piecewise polynomials determined by a set of totally positive connection matrices [18]. Boehm [9] discusses a local rational quadratic recursive evaluation algorithm for cubic Beta-splines. From the analysis in Section 5 one would normally expect the first stage of the general recursive evaluation algorithm for a cubic spline to be rational cubic in the parameter, so there is a little bit of unexpected simplification in Boehm's evaluation algorithm.

7. Marsden's Identity and Dual Functionals—Revisited

In Marsden's identity (Eq. (2)), we express the function $(x - t)^n$ in terms of a given basis $b_0(x), \ldots, b_n(x)$. What is so special about the function $(x - t)^n$? Well if we blossom $(x - t)^n$ with respect to x and evaluate at the roots of a degree n polynomial $d(t)$, then, up to constant multiples, we retrieve the polynomial $d(t)$. That is,

$$B\left[(x - t)^n\right](\text{roots of } d(t)) = \text{constant} \times d(t).$$

It is this identity which leads directly to the blossoming form of the dual functionals expressed in Theorems 3.3 and 3.5: that we can find the coefficients of an arbitrary degree n polynomial $p(x)$ with respect to a given basis $b_0(x), \ldots, b_n(x)$ by evaluating the blossom of $p(x)$ at the roots of the dual basis, i.e.,

$$p(x) = \Sigma_i c_i b_i(x), \qquad \text{where } c_j = \text{constant} \times B[p](r_{j1}, \ldots, r_{jn})$$

and $r_{j1}, \ldots, r_{jn}$ are the roots of the dual basis function $d_j(t)$.

We could choose to use some other degree n polynomial $E(x, t)$ on the left hand side of Marsden's identity. If we did, then we would still obtain a blossoming formula for the dual functionals in terms of the roots of the dual basis but the formula would involve a more complicated expression in the roots. For example, if we let $E(x, t) = (tx + 1)^n$, then for any degree n polynomial $d(t)$

$$B\left[(tx + 1)^n\right](\text{negative reciprocals of the roots of } d(t)) = \text{constant} \times d(t).$$

If we take Marsden's identity to be

$$(tx + 1)^n = \Sigma_i d_i^*(t) b_i(x),$$

then we can find the coefficients of an arbitrary degree n polynomial $p(x)$ with respect to a given basis $b_0(x), \ldots, b_n(x)$ by evaluating the blossom of $p(x)$ at the negative reciprocal of the roots of this new dual basis $d_0^*(t), \ldots, d_n^*(t)$, i.e.,

$$p(x) = \Sigma_i c_i b_i(x), \qquad \text{where } c_j = \text{constant} \times B[p](-1/r_{j1}, \ldots, -1/r_{jn})$$

and $r_{j_1}, \ldots, r_{j_n}$ are the roots of the dual basis function $d_j^*(t)$. More complicated polynomials $E(x, t)$ on the left hand side of Marsden's identity would yield still more complicated blossoming formulas for the dual functionals involving yet more complicated expressions in the roots of the dual basis functions. What is special then about the function $(x - t)^n$

is that it is very easy to blossom, and the blossoming form of the dual functionals is a particularly simple expression in the roots of the dual basis. Nevertheless, the expression $E(x,t)$ that we ultimately choose for the left hand side of Marsden's identity for a particular basis $b_0(x),\dots,b_n(x)$ should depend both on how simple the dual basis functions $d_0^*(t),\dots,d_n^*(t)$ become as well as on how complicated an expression we need to take in their roots to express the blossoming form of the dual functionals.

We can ask much the same question about the de Boor–Fix form of the dual functionals. What is so special about the bilinear form $[f(t), g(t)]_n$? Each bilinear form $[v,w]$ on a vector space V is represented by a matrix M with respect to a basis $v_1,\dots,v_n$. That is,

$$[v,w] = vMw^T.$$

For the de Boor–Fix bilinear form the vector space is the space of degree n univariate polynomials in t, the basis is $1,(t-\tau),\dots,(t-\tau)^n/n!$, and the matrix is the $(n+1)\times(n+1)$ matrix

$$R = \begin{vmatrix} 0 & 0\cdots & & (-1)^n/n! \\ & & \vdots & \\ 1/n! & 0\cdots & & 0 \end{vmatrix}.$$

(It is a lucky accident that by property (i) of Subsection 3.3 the matrix R is independent of the choice of τ. That is, R is invariant under a change of basis from the Taylor basis at τ to the Taylor basis at τ^*.) What would happen if we chose a different matrix M and a different bilinear form $[f(t), g(t)]_n^*$?

The de Boor–Fix bilinear form is closely tied to the function $(x-t)^n$ that appears on the left hand side of the Marsden identity. Indeed by property (iii) of Subsection 3.3

$$\left[f(t),(x-t)^n\right]_n = f(x).$$

From this formula and the Marsden identity it follows that

$$\left[b_i(t), d_j(t)\right]_n = \delta_{ij}$$

and this identity, in turn, leads to the de Boor–Fix form of the dual functionals, i.e.,

$$p(x) = \Sigma_i c_i b_i(x) \quad \Rightarrow \quad c_j = \left[p(t), d_j(t)\right]_n.$$

If we choose a different bilinear form $[f(t), g(t)]_n^*$ represented by a different matrix M, then to obtain the corresponding dual functionals we would have to choose a different expression $E(x, t)$ on the left hand side of the Marsden identity. Let

$$E(x,t)^T = M^{-1}R(x-t)^{nT}.$$

(Here and in subsequent formulas to keep the notation simple, we shall slightly abuse notation by not representing $(x-t)^n$, or any other polynomial, in terms of its Taylor expansion, although it is the Taylor coefficients that we must use for the matrix multiplications to make sense.) Then

$$\begin{aligned}[f(t), E(x,t)]_n^* &= f(t)ME(x,t)^T \\ &= f(t)R(x-t)^{nT} \\ &= \left[f(t), (x-t)^n\right]_n \\ &= f(x).\end{aligned}$$

From this formula and the new Marsden identity

$$E(x,t) = \Sigma_i d_i^*(t)b_i(x)$$

it follows again that

$$\left[b_i(t), d_j^*(t)\right]_n^* = \delta_{ij}$$

and this identity, in turn, leads to the new de Boor–Fix form of the dual functionals, i.e.,

$$p(x) = \Sigma_i c_i b_i(x) \quad \Rightarrow \quad c_j = \left[p(t), d_j^*(t)\right]_n^*.$$

For example, if $M = I$, then

$$\begin{aligned}[f(t), g(t)]_n^* &= \Sigma_k f^{(k)}(0)g^{(k)}(0) \\ E(x,t) &= (tx+1)^n.\end{aligned}$$

Again the bilinear form $[f(t), g(t)]_n^*$ that we ultimately choose for a particular basis $b_0(x), \ldots, b_n(x)$ should depend both on how simple the dual basis functions $d_0^*(t), \ldots, d_n^*(t)$ become as well as on how complicated an expression we need to take in their derivatives to express the de Boor–Fix form of the dual functionals. Choosing the appropriate left hand side of Marsden's identity and choosing the appropriate bilinear form for the dual functionals are intimately related; indeed ultimately they are the same problem. Several cases where it makes good sense to use more complicated expressions for the left hand side of Marsden's identity and to take more complicated bilinear forms to express the dual functionals are discussed in [33].

This problem comes up again with piecewise polynomials determined by connection matrices. With Beta-splines we have two choices: we can represent the dual functionals using the usual blossom evaluated at complicated expressions in the knots or we can use a complicated symmetric expression evaluated at the original knots. Both points of view have been adopted; see [32] for the first and [8] for the second.

8. Extensions to Multivariate Polynomial Bases

By Theorems 3.3, 3.4 given any arbitrary univariate polynomial basis $b_0(x), \ldots, b_n(x)$ there exist unique constants $\gamma_0, \ldots, \gamma_n$ and unique parameters $s_{j1}, \ldots, s_{jn}$ such that if $p(x) = \Sigma_i c_i b_i(x)$, then

$$c_j = \gamma_j B[p](s_{j1}, \ldots, s_{jn}).$$

Here we shall show that this result does not extend to arbitrary multivariate polynomial bases. In addition, we shall derive a necessary and sufficient criteria under which this result does remain valid in the multivariate setting.

To simplify our notation, let us restrict our attention to bivariate polynomials. Consider then a bivariate basis $\{b_{ij}(x, y) | 0 \le i + j \le n\}$ of $\binom{n+2}{2}$ polynomials of total degree n. We begin by extending Marsden's identity to the bivariate setting. Recall from Section 8 that we have a good deal of flexibility in our choice for the left hand side, $E(x, y, s, t)$, of the Marsden identity. We choose to set

$$E(x, y, s, t) = (sx + ty + 1)^n$$

because, as we shall see shortly, this expression is easy to blossom. (This expression is analogous to setting $E(x, t) = (tx + 1)^n$ in the univariate setting. There seems to be no useful direct bivariate analogue to the univariate formula $E(x, t) = (x - t)^n$.) Now in analogy with Theorem 2.1 for every bivariate basis $\{b_{ij}(x, y) | 0 \le i + j \le n\}$ there exists a unique dual basis $\{d_{ij}(s, t) | 0 \le i + j \le n\}$ such that

$$(sx + ty + 1)^n = \Sigma_{ij} d_{ij}(s, t) b_{ij}(x, y). \qquad (11)$$

The proof of this bivariate Marsden identity is much the same as the proof of Theorem 2.1 so we omit the details here.

In addition to Marsden's identity, we need to extend the notation of blossoming to the bivariate setting. Again this is easy to do. Let $p(s,t)$ be a polynomial of total degree n. The blossom or polar form of $p(s,t)$ is the unique, symmetric, multiaffine polynomial $B[p]\{(u_1,v_1),\dots,(u_n,v_n)\}$ which reduces to $p(s,t)$ along the diagonal. That is, the blossom $B[p]\{(u_1,v_1),\dots,(u_n,v_n)\}$ is independent of the order of the pairs of variables $(u_1,v_1),\dots,(u_n,v_n)$, each pair of variables (u_j,v_j) appears to at most the first power, and $B[p]\{(s,t),\dots,(s,t)\} = p(s,t)$. Again the existence and uniqueness of the bivariate blossom are quite straightforward [25]. As in the univariate case, the multilinear blossom $B[p]\{(u_1,v_1,w_1),\dots,(u_n,v_n,w_n)\}$ is just the homogenized version of the multiaffine blossom.

Consider, for example, the polynomial on the left hand side of Marsden's identity (Eq. (11)). This polynomial is especially simple to blossom with respect to the variables (x,y). Indeed it is easy to see that

$$B\left[(sx + ty + 1)^n\right]\{(u_1,v_1,w_1),\dots,(u_n,v_n,w_n)\}$$
$$= (u_1 s + v_1 t + w_1)\cdots(u_n s + v_n t + w_n) \tag{12}$$

because the right hand side is symmetric, multilinear, and reduces to $(sx + ty + 1)^n$ when $(u_i,v_i,w_i) = (x,y,1)$ for $i = 1,\dots,n$.

Every univariate polynomial $p(t)$ is a constant multiple of the blossom of $(x-t)^n$ evaluated at its roots. However, not every bivariate polynomial $p(s,t)$ is a constant multiple of the blossom of $(sx + ty + 1)^n$ evaluated at some appropriate parameters because there are bivariate polynomials which do not factor into linear factors. We shall see shortly that this inability to factor bivariate polynomials into linear factors is the main obstruction to representing the coefficients of an arbitrary bivariate polynomial with respect to a fixed bivariate basis as a multiple of its blossom evaluated at fixed parameter values.

Given a univariate basis $\{b_k(t)\}$, Lemma 3.2 establishes that the blossoms $\{B[b_k](u_1,\dots,u_n) \mid k \neq l\}$ always have a common root. This result is no longer valid for arbitrary bivariate bases. Given a bivariate basis $\{b_{ij}(x,y) \mid 0 \le i+j \le n\}$, the following result tells us precisely when the blossoms $\{B[b_{ij}]\{(u_1,v_1,w_1),\dots,(u_n,v_n,w_n)\} \mid (i,j) \neq (k,l)\}$ have a common root.

LEMMA 8.1. *Let $\{b_{ij}(x,y) \mid 0 \le i+j \le n\}$ and $\{d_{ij}(s,t) \mid 0 \le i+j \le n\}$ be dual polynomial bases. Then*

$$B\left[b_{ij}\right]\{(u_{kl1},v_{kl1},w_{kl1}),\dots,(u_{kln},v_{kln},w_{kln})\} = 0, \qquad (i,j) \neq (k,l)$$

if and only if

$$d_{kl}(x,y) = \gamma_{kl}(u_{kl1}x + v_{kl1}y + w_{kl1}) \cdots (u_{kln}x + v_{kln}y + w_{kln}).$$

That is, the blossoms $\{B[b_{ij}] | (i,j) \neq (k,l)\}$ *have a common root if and only if the polynomial* $d_{kl}(x,y)$ *factors into linear factors.*

Proof. Suppose that

$$B[b_{ij}]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} = 0, \qquad (i,j) \neq (k,l).$$

Then blossoming Eq. (11) with respect to the variables (x, y), evaluating at $(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})$, and applying Eq. (12), we obtain

$$\begin{aligned}(u_{kl1}s + v_{kl1}t + w_{kl1}) &\cdots (u_{kln}s + v_{kln}t + w_{kln}) \\ &= B[b_{kl}]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} d_{kl}(s,t).\end{aligned}$$

Thus $d_{kl}(s,t)$ factors into linear factors. Conversely, suppose that

$$d_{kl}(s,t) = \gamma_{kl}(u_{kl1}s + v_{kl1}t + w_{kl1}) \cdots (u_{kln}s + v_{kln}t + w_{kln}).$$

Again blossoming Eq. (11) with respect to the variables (x, y) and evaluating at $(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})$, we obtain

$$d_{kl}(s,t)/\gamma_{kl} = \Sigma_{ij} B[b_{ij}^{n}]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} d_{ij}(s,t).$$

But $\{d_{ij}(s,t)\}$ is a polynomial basis; therefore the coefficients of $d_{ij}(s,t)$ on the left hand side and the right hand side must be identical, so

$$B[b_{ij}]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} = 0, \qquad (i,j) \neq (k,l). \quad \text{Q.E.D.}$$

THEOREM 8.2. *Let* $\{b_{ij}(x,y) | 0 \leq i + j \leq n\}$ *and* $\{d_{ij}(s,t) | 0 \leq i + j \leq n\}$ *be dual polynomial bases, and let* $p(x,y)$ *be an arbitrary polynomial of total degree n. Then the following two statements are equivalent:*

(1) $d_{kl}(s,t) = \gamma_{kl}(u_{kl1}s + v_{kl1}t + w_{kl1}) \cdots (u_{kln}s + v_{kln}t + w_{kln})$;

(2) $p(x,y) = \Sigma_{ij} c_{ij} b_{ij}(x,y)$ *where* $c_{kl} = \gamma_{kl} B[p]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\}$.

That is, the coefficient of $b_{kl}(x,y)$ *is given by a constant times the blossom of* $p(x,y)$ *evaluated at certain fixed parameter values if and only if* $d_{kl}(s,t)$ *factors into linear factors.*

Proof. Suppose that

$$d_{kl}(s,t) = \gamma_{kl}(u_{kl1}s + v_{kl1}t + w_{kl1}) \cdots (u_{kln}s + v_{kln}t + w_{kln}).$$

Since $\{b_{ij}(x, y)\}$ is a polynomial basis, there exist constants $\{c_{ij}\}$ such that

$$p(x, y) = \Sigma_{ij} c_{ij} b_{ij}(x, y).$$

Blossoming both sides with respect to (x, y), evaluating at $(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})$, and applying Lemma 8.1, we obtain

$$\begin{aligned} B[p]\{&(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} \\ &= c_{kl} B[b_{kl}]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} \end{aligned}$$

and the result follows by solving for c_{kl}. Conversely, suppose that for any polynomial $p(x, y)$ of total degree n

$$p(x, y) = \Sigma_{ij} c_{ij} b_{ij}(x, y),$$

where $c_{kl} = \gamma_{kl} B[p]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\}$.

Let $p(x, y) = (sx + ty + 1)^n$. By Marsden's identity (Eq. (11)),

$$(sx + ty + 1)^n = \Sigma_{ij} d_{ij}(s,t) b_{ij}(x, y).$$

Therefore by assumption

$$\begin{aligned} d_{kl}(s,t) &= \gamma_{kl} B[p]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\} \\ &= \gamma_{kl}(u_{kl1}s + v_{kl1}t + w_{kl1}) \cdots (u_{kln}s + v_{kln}t + w_{kln}). \end{aligned}$$ Q.E.D.

By Theorem 8.2, given a polynomial basis $\{b_{ij}(x, y)\}$ there exists constants $\{\gamma_{ij}\}$ and parameters $(u_{ij1}, v_{ij1}, \ldots, w_{ij1}), \ldots, (u_{ijn}, v_{ijn}, w_{ijn})$ such that if $p(x, y) = \Sigma_{ij} c_{ij} b_{ij}(x, y)$ then

$$c_{kl} = \gamma_{kl} B[p]\{(u_{kl1}, v_{kl1}, w_{kl1}), \ldots, (u_{kln}, v_{kln}, w_{kln})\}$$

if and only if the dual basis functions $d_{kl}(s, t)$ factor into linear factors. Moreover in this case

$$d_{kl}(s,t) = \gamma_{kl}(u_{kl1}s + v_{kl1}t + w_{kl1}) \cdots (u_{kln}s + v_{kln}t + w_{kln})$$

so the constants $\{\gamma_{ij}\}$ and the parameters $(u_{ij1}, v_{ij1}, w_{ij1}), \ldots, (u_{ijn}, v_{ijn}, w_{ijn})$ are unique up to constant multiples. If the parameters $\{w_{ijm}\}$ are non-zero, then we can normalize them so that $w_{ijm} = 1$ by an appropriate choice of the constants $\{\gamma_{ij}\}$. Thus, in this case, there exist unique constants $\{\gamma_{ij}\}$ and parameters $(u_{ij1}, v_{ij1}), \dots, (u_{ijn}, v_{ijn})$ such that if $p(x, y) = \Sigma_{ij} c_{ij} b_{ij}(x, y)$, then

$$c_{kl} = \gamma_{kl} B[p]\{(u_{kl1}, v_{kl1}), \dots, (u_{kln}, v_{kln})\}. \tag{13}$$

Here we have dehomogenized the multilinear blossom by setting $w_{ijm} = 1$; thus we have replaced the multilinear blossom by the multiaffine blossom. If, in addition, the basis functions $\{b_{ij}(x, y)\}$ are normalized so that

$$\Sigma_{ij} b_{ij}(x, y) = 1,$$

then by Eq. (13)

$$1 = \gamma_{kl} B[1]\{(u_{kl1}, v_{kl1}), \dots, (u_{kln}, v_{kln})\} = \gamma_{kl}.$$

Therefore when the basis functions are normalized to sum to one.

$$c_{kl} = B[p]\{(u_{kl1}, v_{kl1}), \dots, (u_{kln}, v_{kln})\}$$

just as in the univariate case.

Analogous results hold for multivariable polynomials with an arbitrary number of variables; the proofs are much the same. Notice that all these results always hold for univariate polynomials because every univariate polynomial factors into linear factors over the complex numbers.

9. Conclusions and Open Questions

We have presented techniques for extending Marsden's identity, the blossoming and de Boor–Fix forms of the dual functionals, the Oslo algorithm, and recursive procedures for evaluation, differentiation, and blossoming from B-spline and progressive polynomial bases to arbitrary polynomial and locally linearly independent spline bases. The key idea is to extend Marsden's identity first and then to derive the other formulas and procedures as consequences of this basic identity by working with the dual basis.

Many questions remain open. Dual bases satisfy Marsden's identity and the de Boor–Fix formula. In what other interesting ways are dual bases related? For example, many bases can be generated by recurrences that are much simpler than the general rational recurrence presented in Section 5. Indeed, the Chebysheff basis can be generated by a very simple two term recurrence [14]. Given a simple recurrence for a particular basis is there a correspondingly simple recurrence for its dual basis? Or suppose a basis satisfies Descartes' Law of Signs [29] in some interval; can we say anything about the behavior of its dual basis with respect to Descartes'

Law of Signs? This last question is especially important in CAGD where it is desirable for curves generated from blending functions to satisfy the variation diminishing property.

Multivariate polynomial and spline bases provide a rich source of questions. For multivariate spline bases such as box-splines, the dual functionals are not given simply by evaluating the blossom at the knots [25, 28]. If the dual basis factors into linear factors, then these factors will tell us precisely where to evaluate the blossom to obtain the dual fractionals. On the other hand, if the dual basis does not factor into linear factors, then we cannot obtain the dual functionals simply by evaluating the blossom at appropriate parameter values.

When the dual basis functions factor into closely related linear factors, the multivariate theory is much the same as in the univariate setting. In particular, there are simple recursive procedures for evaluation, differentiation, and blossoming for the multivariate Bernstein basis and more generally for the multivariate analogue of progressive bases—the B-weights associated with the B-patches [17, 31]. However, even when the dual basis functions factor into distinct linear factors, it is not clear, in general, how to generate even high degree rational recursive procedures for evaluation, differentiation, and blossoming. Moreover, as we have seen, fundamental problems arise because, unlike univariate polynomials, multivariate polynomials do not necessarily factor into linear factors. That is, the dual basis may contain irreducible polynomials of high degree. Can we obtain recursive procedures for evaluation, differentiation, and blossoming in this general setting? All these multivariate questions deserve further attention and we hope to return to them again sometime soon.

Acknowledgment

This research was partially supported by NSF Grant CCR-9112239.

References

1. A. A. Ball, Consurf. I. Introduction of the conic lofting tile, Comput. Aided Design **6** (1974), 243–249.
2. P. J. Barry, De Boor–Fix functionals and polar forms, *Comput. Aided Geom. Design* **7** (1990), 425–430.
3. P. J. Barry, A. D. DeRose, and R. N. Goldman, B-Splines, Polya curves, and duality, *J. Approx. Theory* **65** (1991), 3–21.
4. P. J. Barry, N. Dyn, R. N. Goldman, and C. A. Micchelli, Polynomial identities for piecewise polynomials determined by connection matrices, *Aequationes Math.* **42** (1991), 123–136.

5. P. J. Barry, R. N. Goldman, What is the natural generalization of a Bezier curve?, *in* "Mathematical Methods in Computer Aided Geometric Design" (T. Lyche and L. Schumaker, Eds.), pp. 71–85, Academic Press, San Diego, 1989.
6. P. J. Barry and R. N. Goldman, Interpolation and approximation of curves and surfaces using Polya polynomials, *CVGIP: Graphical Models Image Process.* **53** (1991), 137–148.
7. P. J. Barry and R. N. Goldman, Algorithms for progressive curves: Extending B-spline and blossoming techniques to the mononomial, power, and Newton dual bases, *in* "Knot Insertion and Deletion Algorithms for B-spline Curves and Surfaces" (R. Goldman and T. Lyche, Eds.), pp. 11–63, SIAM, Philadelphia, 1993.
8. P. J. Barry, R. N. Goldman, and C. A. Micchelli, Knot insertion algorithms for piecewise polynomial spaces determined by connection matrices, *Adv. in Comput. Math.* in press.
9. W. Boehm, Curvative continuous curves and surfaces, *Comput. Aided Geom. Design.* **2** (1985), 313–324.
10. C. de Boor, On calculating with B-splines, *J. Approx. Theory* **6** (1972), 50–62.
11. C. de Boor, "A Practical Guide to Splines," Springer-Verlag, New York, 1972.
12. C. de Boor and G. Fix, Spline approximation by quasi-interpolants, *J. Approx. Theory* **8** (1973), 19–45.
13. C. de Boor and K. Hollig, B-splines without divided differences, *in* "Geometric Modelling" (G. Farin, Ed.), pp. 21–27, SIAM, Philadelphia, 1987.
14. R. Burden and J. Faires, "Numerical Analysis," PWS Publishers, Boston, 1985.
15. P. de Casteljau, "Formes a Poles," Hermes, Paris, 1985.
16. E. Cohen, T. Lyche, and R. Riesenfeld, Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, *Comput. Graphics Image Process.*, **14** (1980), 87–111.
17. W. Dahmen, C. A. Micchelli, and H.-P. Seidel, Blossoming begets B-splines built better by B-patches, *Math. Comp.* **59**, (1992), 97–115.
18. N. Dyn and C. A. Micchelli, Piecewise polynomial spaces and geometric continuity of curves, *Numer. Math.* **54** (1988), 319–337.
19. R. N. Goldman, Blossoming and knot insertion algorithms for B-spline curves, *Comput. Aided Geom. Design* **7** (1990), 69–81.
20. R. N. Goldman and P. J. Barry, Wonderful triangle, *in* "Mathematical Methods in Computer Aided Geometric Design, II" (T. Lyche and L. Schumaker, Eds.), pp. 297–320, Academic Press, San Diego, 1992.
21. E. T.Y. Lee, A note on blossoming, *Comput. Aided Geom. Design* **6** (1989), 359–362.
22. T. Lyche and K. Morken, Making the Oslo algorithm more efficient, *SIAM J. Numer. Anal.* **23** (1986), 663–675.
23. M. J. Marsden, An identity for spline functions with applications to variation-diminishing spline approximation, *J. Approx. Theory* **3** (1970), 7–49.
24. H. Minc, "Permanents," Addison–Wesley, London, 1978.
25. L. Ramshaw, "Blossoming: A Connect the Dots Approach to Splines," Digital Systems Research Center, Rep. No. 19, Palo Alto, CA, 1987.
26. L. Ramshaw, Beziers and B-splines as multiaffine maps, *in* "Theoretical Foundations of Computer Graphics and CAD" (R. A. Earnshaw, Ed.), pp. 757–776, NATO ASI Series F, Vol. 40, Springer-Verlag, New York, 1988.
27. L. Ramshaw, Blossoms are polar forms, *Comput. Aided Geom. Design* **6** (1989), 323–358.
28. L. Ramshaw, On blossoming and box splines, *in* "Oberwolfach Conference on Computer Aided Geometric Design," Oberwolfach, Germany, 1989.

29. L. L. SCHUMAKER, "Spline Functions: Basic Theory," Wiley, New York, 1981.
30. H. P. SEIDEL, A new multiaffine approach to B-splines, *Comput. Aided Geom. Design* **6** (1989), 23–32.
31. H. P. SEIDEL, Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles, *Constr. Approx.* **7** (1991), 257–279.
32. H. P. SEIDEL, Polar forms for geometrically continuous spline curves of arbitrary degree, *Trans. Graphics* **12** (1993), 1–35.
33. Y. STEFANUS, "Blossoming off the Diagonal," Ph.D. thesis, Department of Computer Science, University of Waterloo, 1991.
34. Y. STEFANUS AND R. N. GOLDMAN, Blossoming Marsden's identity, *Comput. Aided Geom. Design* **9** (1992), 73–84.