An Introduction to Blossoming of BSplines

Xianming Chen

School of Computing, University of Utah, Salt Lake City, UT 84112

1 Introduction

Blossoming is an alternative representation to B-splines, and is well understood in the CAD community. See [Ram87, Ram89, DG91, Sei93, Gal98] for more details.

2 Affine Spaces and Affine Mappings

2.1 Affine Spaces

An affine space $\mathbf{A}$, has two types of entities, i.e., points and vectors. All the vectors form a vector space, denoted as $\mathbf{V_A}$, on its own. $\mathbf{A}$ has the following properties.

(1) There is a unique vector related to a pair of points, defining two operations,
   (a) Add a vector to a point to get another point.

$$p + \vec{v} = q. \tag{1}$$

   (b) Subtracting two points to get a vector,

$$q - p = \vec{v}. \tag{2}$$

(2) Affine combination of points into another point.

$$\sum_{i=0}^{n} \lambda_i p_i = q, \tag{3}$$

when

$$\sum_{i=0}^{n} \lambda_i = 1.$$

In NURBS community, we actually almost always put an extra restriction on equation (3). That is, all the coefficients λ_i are required to be non-negative, and such an affine combination is called convex hull combination, or more commonly interpolation (contrast to *extrapolation¡*).

(3) Combination of points into a vector.

$$\sum_{i=0}^{n} \lambda_i p_i = \vec{v}, \tag{4}$$

when

$$\sum_{i=0}^{n} \lambda_i = 0.$$

Based on these properties, it can be proved that an affine space has (addition and subtraction) algebra for its points and vectors in the common sense.

The usual Euclidean spaces are affine spaces with the affine operations defined component-wise. We use $\mathbf{A}^r$, to represent an affine space of Euclidean r-space. The related vector space, $\mathbf{V_A}$ is just $\mathbf{R}^r$. Also in this tutorial, $\mathbf{A}$ and $\mathbf{A_i}$ is used to represent a general affine space of any dimension; similarly $\mathbf{V}$ and $\mathbf{V_i}$ for a general vector space.

Example 1 *In Euclidean 1-space, the same notation* 1*, could be a point, a vector, or just a scalar(i.e. an element of the field on which the vector space is defined). The ambiguity can usually be resolved based on the context in most cases. When this is impossible, vector* 1 *is explicitly written as* $\vec{1}$*.*

2.2 Linear and Affine Mappings

Mapping $f : \mathbf{V}_1 \to \mathbf{V}_2$ is linear if, for $x_0, x_1 \in \mathbf{V}_1$,

$$f(\lambda_0 x_0 + \lambda_1 x_1) = \lambda_0 f(x_0) + \lambda_1 f(x_1).$$

Mapping $f : \mathbf{A}_1 \to \mathbf{A}_2$ is affine if, for $x_0, x_1 \in \mathbf{A}_1$,

$$f(\lambda_0 x_0 + \lambda_1 x_1) = \lambda_0 f(x_0) + \lambda_1 f(x_1), \quad \text{when } \lambda_0 + \lambda_1 = 1.$$

Example 2 $f(x) = 3x$ *is a linear mapping from vector space* $\mathbf{R}^1$ *to* $\mathbf{R}^1$*, since,*

$$\begin{aligned} f(\lambda_0 x_0 + \lambda_1 x_1) &= 3(\lambda_0 x_0 + \lambda_1 x_1) \\ &= 3\lambda_0 x_0 + 3\lambda_0 x_1 \\ &= \lambda_0 f(x_0) + \lambda_1 f(x_1). \end{aligned}$$

On the other hand, $g(x) = 3x+2$ *is affine from* $\mathbf{A}^1$ *to* $\mathbf{A}^1$*, since, when* $\lambda_0+\lambda_1 =$

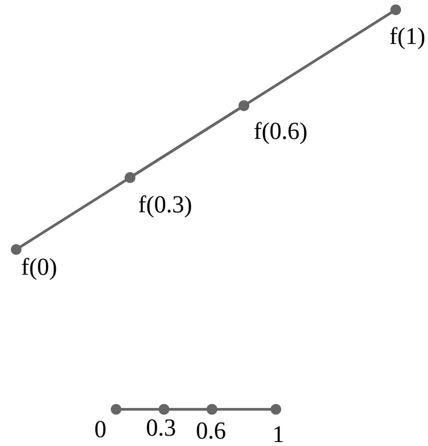

Fig. 1. Affine Mapping and Its Control Points. *Interpolate affine mapping (from* $\mathbf{A^1}$ *to* $\mathbf{A^3}$*) values from control points* $f(0)$ *and* $f(1)$*. For example,* $f(0.3) = f(0.3 * 1 + (1 - 0.3) * 0) = 0.3f(1) + 0.7f(0)$*, and similarly for* $f(0.6)$*.*

1,

$$\begin{aligned} g(\lambda_0 x_0 + \lambda_1 x_1) &= 3(\lambda_0 x_0 + \lambda_1 x_1) + 2 \\ &= 3\lambda_0 x_0 + 3\lambda_1 x_1 + 2(\lambda_0 + \lambda_1) \\ &= \lambda_0(3x_0 + 2) + \lambda_1(3x_1 + 2) \\ &= \lambda_0 g(x_0) + \lambda_1 g(x_1), \end{aligned}$$

2.3 Affine Basis and Control Points of an Affine Mapping: $f : \mathbf{A}^1 \to \mathbf{A}$

Point 0 and point 1 form a basis of the affine space $\mathbf{A}^1$ since every point in $\mathbf{A}^1$ can be expressed as an affine combination of 0 and 1 as,

$$t = (1-t)\ 0 + t\ 1 \ \text{[1]},$$

And consequently, we have

$$f(t) = f((1-t)\ 0 + t\ 1) = (1-t)f(0) + tf(1).$$

Hence, every affine mapping can be constructively defined by its two image point $f(0)$ and $f(1)$. To have a numerically stable constructive algorithm, the mapping is actually only defined in the domain $[0, 1]$ so that the affine combinations from $f(0)$ and $f(1)$ are convex. See Figure 1 for an example.

[1] Notice that the left hand side t is a *point* in $\mathbf{A^1}$, while the right hand side ones are scalars.

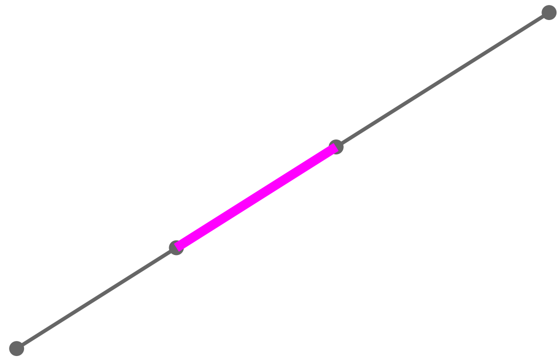

Fig. 2. Affine Map on Vectors $\quad f(\vec{0.3}) = f(0.6 - 0.3) = f(0.6) - f(0.3)$

Notice that the pair 0 and 1 is not the only basis of $\mathbf{A}^1$. Actually any two distinct points in $\mathbf{A}^1$ form a basis. For example, if we choose another basis of $\mathbf{A}^1$, say 1.5 and 2.1, then every affine mapping (from $\mathbf{A^1}$) can be uniquely decided by its two image points (control points) $f(1.5)$ and $f(2.1)$.

2.4 Affine Mappings on Vectors in $\mathbf{R}^1$

By equation (4), an affine mapping on a vector in $\mathbf{R}^1$ is defined as a vector-combination of the affine mapping values on the corresponding affine points, which combine into the given vector. Specifically, if

$$\vec{v} = \sum \lambda_i p_i,$$

then

$$f(\vec{v}) = \lambda_i f(p_i)$$

Notice that f is linear w.r.t. its vector argument as proved below.

$$\begin{aligned} \vec{v} &= \sum \lambda_i p_i \\ \vec{u} &= \sum \sigma_j q_j \\ f(a\vec{v} + b\vec{u}) &= a\ \lambda_i f(p_i) + b\ \sigma_j f(q_j) = af(\vec{v}) + bf(\vec{u}) \end{aligned}$$

Specifically, an affine map on unit vector of $\mathbf{R}^1$ can be defined canonically as

$$f(\vec{1}) = f(1 - 0) = f(1) - f(0).$$

See Figure 2 for an example.

3 Symmetric Multi-Affine Mappings

$f : \mathbf{A}_1 \times \cdots \times \mathbf{A}_1 \to \mathbf{A}_2$ is called a symmetric multi-affine mapping(on $\mathbf{A}_1$), if it is symmetric,

$$f(x_0, \cdots, x_n) = f(\pi(x_0, \cdots, x_n)),$$

where π is any permutation, and multi-affine,

$$f(x_0, \cdots, \sum_{k=0}^{m} a_{i_k} x_{i_k}, \cdots, x_n) = \sum_{k=0}^{m} a_{i_k} f(x_0, \cdots, x_{i_k}, \cdots, x_n), \quad \text{for } i = 0, \cdots, n.$$

Example 3 *$f(x, y) = xy + 3$ is a symmetric affine mapping from $\mathbf{A}^1 \times \mathbf{A}^1$ to $\mathbf{A}^1$, since*

$$f(x, y) = xy + 3 = yx + 3 = f(y, x),$$

and

$$\begin{aligned} f(ax_0 + bx_1, y) &= (ax_0 + bx_1)y + 3 \\ &= ax_0y + bx_1y + (a + b)3 \\ &= af(x_0, y) + bf(x_1, y), \end{aligned}$$

provided

$$a + b = 1.$$

Similarly as in affine mappings, any symmetric n-affine mapping on $\mathbf{A}^1$ can be constructively defined from its $n+1$ control points. For any $t_i \in [0, 1], i = 0, 1, \cdots, n-1$, we have

$$f(t_0, \cdots, t_{n-1}) = f((1 - t_0)\ 0 + t_0\ 1, \cdots, (1 - t_{n-1})\ 0 + t_{n-1}\ 1) = \sum_{i=0}^{n} e_i b_i,$$

where for $i = 0, 1, \cdots, n$,

$$\begin{aligned} e_i &= \prod_{k=i}^{n-1} (1 - t_k) \prod_{k=0}^{i-1} t_k, \\ b_i &= f(0^{n-i} 1^i). \end{aligned}$$

The b_is are the $n+1$ control points, and the e_is are the non-negative coefficients to affine combine these control points into $f(t_0, \cdots, t_{n-1})$.

Figure 3 shows the constructive definition of a symmetric 2-affine mapping.

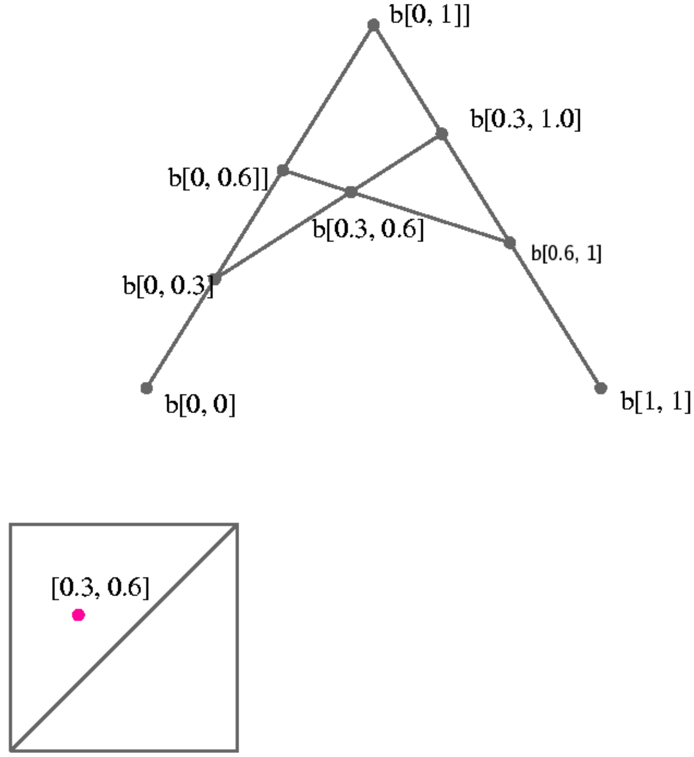

Fig. 3. Symmetric Multi Affine Map *Interpolate symmetric 2-affine mapping values from its control points* $f(0.3, 0.6)$ *can either be interpolated from* $f(0, 0.3)$ *and* $f(0.3, 1.0)$*, or from* $f(0, 0.6)$ *and* $f(0.6, 1.0)$*. Recursively,* $f(0, 0.3)$ *is interpolated from* $f(0, 0)$ *and* $f(0, 1)$*, etc.*

4 Diagonalization and Blossoming: Polynomials and Symmetric Multi-Affine Mappings

In this tutorial, capital letters, like B and F, G, H, represent polynomials, which are the diagonalization of the corresponding symmetric multi-affine mappings, like b and f, g, h. And for piecewise case, a subscript of i means the i-th piece polynomial or symmetric multi-affine mapping (cf. Section **??**.

4.1 Diagonalization

When all the n variables in a symmetric n-affine mapping $f(x_0, \cdots, x_{n-1})$ are the same, $f(x_0, \cdots, x_{n-1})$ is identified with a degree n polynomial $F(x)$. This is the diagonalization of symmetric multi-affine mapping. If $f(x_0, \cdots, x_{n-1})$

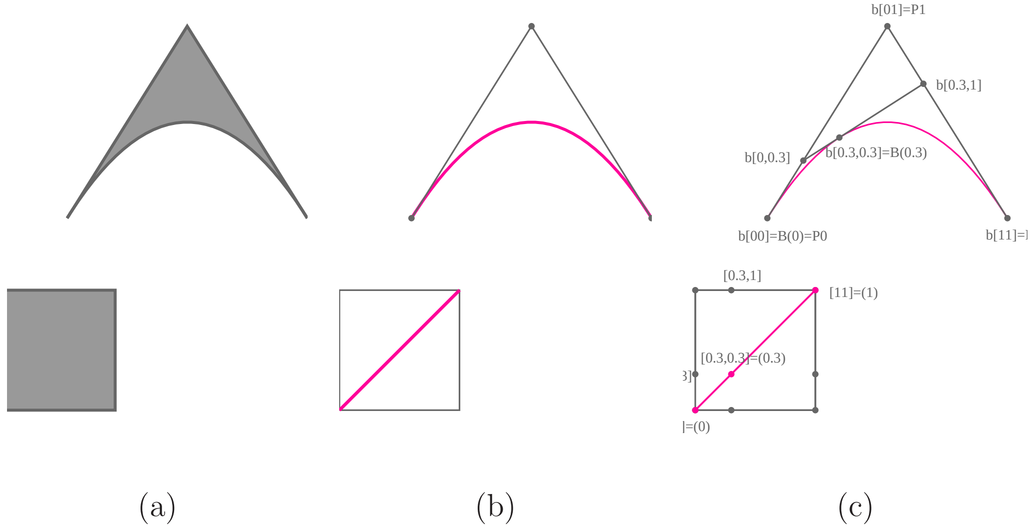

Fig. 4. A Symmetric Multi-Affine Mapping and Its Diagonalization *(a) shows the square-shaped domain and the range of a symmetric 2-affine mapping. (b)shows the diagonal domain and its corresponding range. (c)shows how to compute a polynomial value through interpolation.*

is defined on a hyper-cube $[0,1]^n$, then $F(x)$ is defined on the corresponding diagonal, i.e. $[0,1]$. See Figure 4 for an example.

4.2 Blossoming

The diagonalization of symmetric multi-affine mapping has an inverse operation, called the blossoming, which turns a degree n polynomial into a symmetric n-affine mapping.

Example 4 (Blossoming of a linear polynomial.)

$$F = ax + b,$$

$$f = ax + b.$$

Example 5 (Blossoming of a quadric polynomial.)

$$\begin{aligned} G &= ax^2 + bx^1 + c, \\ g &= ax_0x_1 + b(x_0 + x_1)/2 + c. \end{aligned} \tag{5}$$

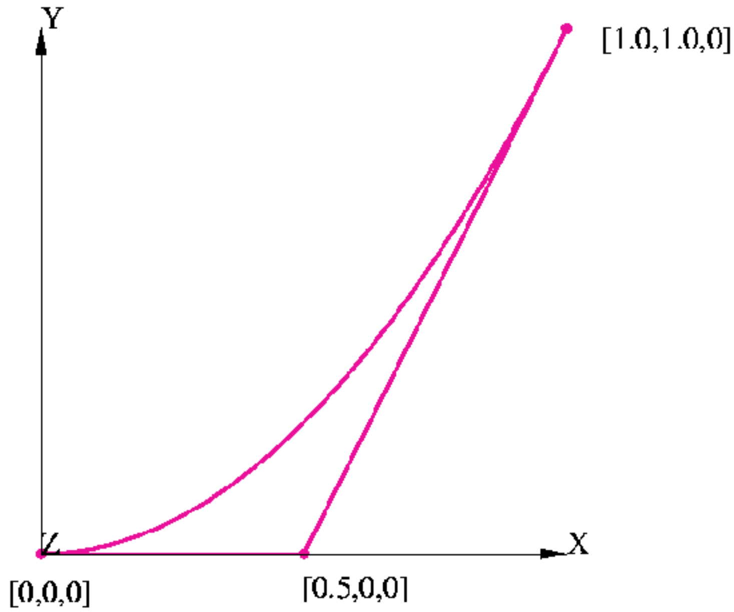

Fig. 5. Construct Parabola $y = x^2$ from Its Blossoming

Example 6 (Blossoming of a cubic polynomial.)

$$H = ax^3 + bx^2 + cx^1 + d,$$
$$h = ax_0x_1x_2 + b(x_0x_1 + x_0x_2 + x_1x_2)/3 + c(x_0 + x_1 + x_2)/3 + d. \tag{6}$$

Example 7 (Contruct a parabola by its control points) *Suppose we want to construct a parabola,* $y = x^2$ *with the simple parametrization (for all* 3 *components) of,*

$$F_0(t) = t,$$
$$F_1(t) = t^2,$$
$$F_2(t) = 0.$$

First we find $f_0(t_0, t_1), f_1(t_0, t_1), f_2(t_0, t_1)$*, the blossom of* Fs*.*

$$f_0(t_0, t_1) = (t_0 + t_1)/2,$$
$$f_1(t_0, t_1) = t_0t_1,$$
$$f_2(t_0, t_1) = 0.$$

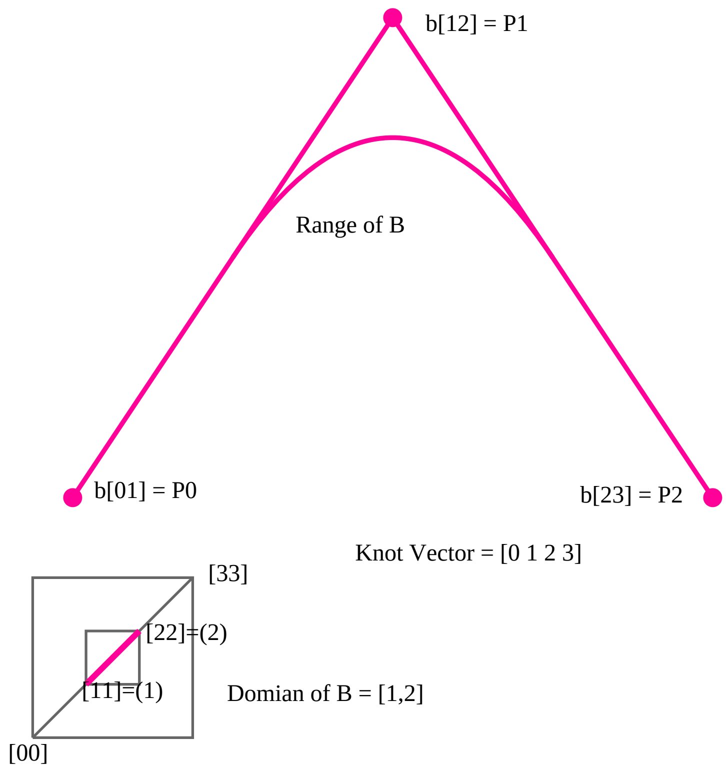

Fig. 6. Floating End Conditions

Next, we find the control points defining the 3 2-affine mappings.

$$P_0 = (f_0(0,0), f_1(0,0), f_2(0,0)) = (0.0, 0, 0),$$
$$P_1 = (f_0(0,1), f_1(0,1), f_2(0,1)) = (0.5, 0, 0),$$
$$P_2 = (f_0(1,1), f_1(1,1), f_2(1,1)) = (1.0, 1, 0),$$

And finally we get the constructive parabola as in Figure 5.

4.3 Floating End Conditions

So far we have been talking about open end conditions, i.e., for degree n polynomial, the first and the last control points are defined by $f(0^n)$ and $f(1^n)$.

Figure 6 shows a general end condition.

5 Basic B-spline Operations via Blossoming

5.1 Differentiation

Let $B(t)$ be a degree n polynomial, with blossom of symmetric n-affine mapping b. We have,

$$\begin{aligned}
B'(t) &= \lim_{h\to 0} \frac{B(t+h\vec{1}) - B(t)}{h} \\
&= \lim_{h\to 0} \frac{b((t+h\vec{1})^n) - b(t^n)}{h} \\
&= \lim_{h\to 0} \frac{b(t^n) + n * b(t^{n-1}h\vec{1}) + n*(n-1)/2 * b(t^{n-2}h^2\vec{1}) + \cdots + b(h^n\vec{1}) - b(t^n)}{h} \\
&= \lim_{h\to 0} \frac{n * b(t^{n-1}h\vec{1})}{h} \\
&= n * b(t^{n-1}\vec{1}).
\end{aligned}$$

And by recursion, we have, for any $q = 0, \cdots, n$,

$$B^{(q)}(t) = n * (n-1) * \cdots * (n-q+1) * b[t^{n-q}\vec{1}^q]. \tag{7}$$

Example 8 *Compute the derivative of $y = x^2$ at $x = 0.5$ with the parametrization as earlier.*

$$\begin{aligned}
F_0(t) &= t, \\
F_1(t) &= t^2, \\
F_2(t) &= 0. \\
F'(0.5) &= (1, 2t, 0)_{t=0.5} = (1, 1, 0),
\end{aligned}$$

Blossoming $F = (F_0, F_1, F_2)$, we have,

$$\begin{aligned}
f_0(t_0, t_1) &= (t_0 + t_1)/2, \\
f_1(t_0, t_1) &= t_0 t_1, \\
f_2(t_0, t_1) &= 0.
\end{aligned}$$

and differentiating the multi-affine mapping, we get,

$$\begin{aligned}
2 * f_0(0.5, \vec{1}) &= 2 * f_0(0.5, 1-0) = 2f_0(0.5, 1) - 2f_0(0.5, 0) = 1.0, \\
2 * f_1(0.5, \vec{1}) &= 2 * f_1(0.5, 1-0) = 2f_1(0.5, 1) - 2f_1(0.5, 0) = 1.0, \\
2 * f_2(0.5, \vec{1}) &= 2 * f_2(0.5, 1-0) = 2f_2(0.5, 1) - 2f_2(0.5, 0) = 0.0,
\end{aligned}$$

the same result as above.

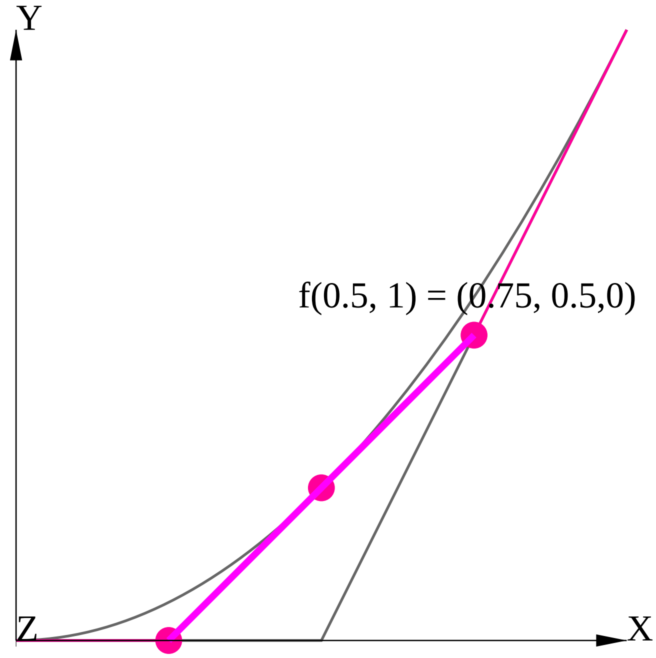

Fig. 7. Derivative of Polynomial Via Blossoming $F'(0.5) = f'(0.5, 0.5) = 2f(0.5, \vec{1}) = 2(f(0.5, 1.0) - f(0, 0.5))$. *The tangent segment clipped by the control mesh is half of the derivative of $y = x^2$ at $x = 0.5$. This result actually holds for any point on the curve.*

The important thing to note is that the derivative of a symmetric n-affine mapping is still a symmetric affine mapping (i.e., a closed operation), with varity of $n-1$; and the new blossom can again be defined constructively, with its control pints (vectors actually) to be the scaled (by n) differences of the original consecutive control points. See Figure 8

5.2 Degree Elevation

Using blossoming, it is straight forward to degree raise a polynomial, and any amount of degree elevation can be done in one step.

Here is the general principle. Given any degree n polynomial F, and its blossom n-affine mapping f, we can define a symmetric $(n+d)$-affine mapping g as,

$$g(t_1 \cdots t_{n+d}) = \frac{\sum_{\{k_1 \cdots k_n\} \subset \{1,2,\cdots n+d\}} f(t_{k_1} \cdots t_{k_n})}{\binom{n+d}{n}} \tag{8}$$

The diagonalization of $g(t_1 \cdots t_{n+d})$, G, is the degree raised polynomial we want. That is, $G(x) = F(x)$ for any x, and $G(x)$ has d higher degrees than F has. This is really true, because

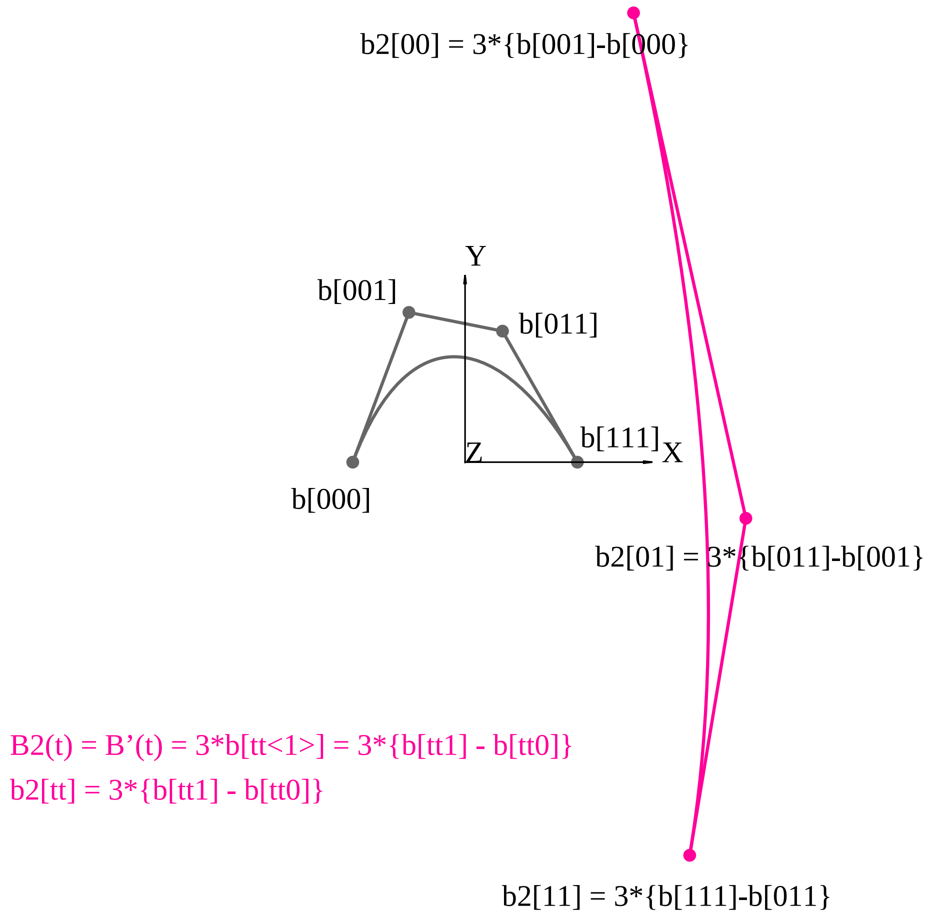

Fig. 8. B-spline Differentiation

(1) $g(t_1 \cdots t_{n+d})$ has varity $n + d$.
(2) $g(t_1 \cdots t_{n+d})$ is symmetric because every combination of d variables out of $d+n$ variables are taken into account in the nominator of equation (8).
(3) $g(t_1 \cdots t_{n+d})$ is affine since it is an affine combination of $\binom{n+d}{n}$ affine mappings.
(4) When diagonalizing, all $\binom{n+d}{n}$ terms in the nominator of equation (8) are the same as the diagonalization of the original affine mapping. Dividing by $\binom{n+d}{n}$, the two diagonalizations are exactly the same.

Example 9 (Degree Raise A Degree One Polynomial To A Cubic Polynomial) *Let $F(t) = f(t) = t$, from A^1 to A^1. Suppose they are defined by control points $F(0) = 0$ and $F(1) = 1$.*

Define 3-affine mapping g as,

$$g(t_1, t_2, t_3) = \frac{f(t_1) + f(t_2) + f(t_3)}{3}$$

with domain of $[0,1]^3$.

The control points of g *are,*

$$g(0,0,0) = 0.0, \quad g(0,0,1) = 1.0/3, \quad g(0,1,1) = 2.0/3, \quad g(1,1,1) = 1.0,$$

and it is obvious that these four control points define a cubic which is actually linear.

Example 10 (Degree Raise A Quadric To Quartic Bezier Curve) *Given a quadric Bezier curve, defined by 3 control points* p_0*,* p_1*, and* p_2*. That is, the blossom has,*

$$f(0,0) = p_0, \quad f(0,1) = p_1, \quad f(1,1) = p_2.$$

Define symmetric 4-affine mapping g *as,*

$$g(t_0,t_1,t_2,t_3) = \frac{f(t_0,t_1) + f(t_0,t_2) + f(t_0,t_3) + f(t_1,t_2) + f(t_1,t_3) + f(t_2,t_3)}{\binom{4}{2}}$$

Therefore the degree raised quartic curve has control points,

$$\begin{aligned}
g(0,0,0,0) &= 6p_0/6 = p_0, \\
g(0,0,0,1) &= (3p_0 + 3p_1)/6 = (p_0 + p_1)/2, \\
g(0,0,1,1) &= (p_0 + 4p_1 + p_2)/6, \\
g(0,1,1,1) &= (3p_1 + 3p_2)/6 = (p_1 + p_2)/2, \\
g(1,1,1,1) &= 6p_2/6 = p_2.
\end{aligned}$$

This is illustrated in Figure 9.

5.3 Insertion and Subdivision

Figure 10 shows insertion of a knot value, and the resulting subdivision of the curve and the new control points of the new segment curves.

5.4 B-spline Product

Direct multiplication of two B-splines can be achieved using the blossom symmetrizing strategy used earlier for degree elevation.

First, let us observe that the distinct knot values of the product B-spline is the set union of those of the two factor B-splines. And based on continuity consideration, the multiplicity of any knot that is only from one of the factor B-splines is raised by the degree of the other factor B-spline, and the multiplicity

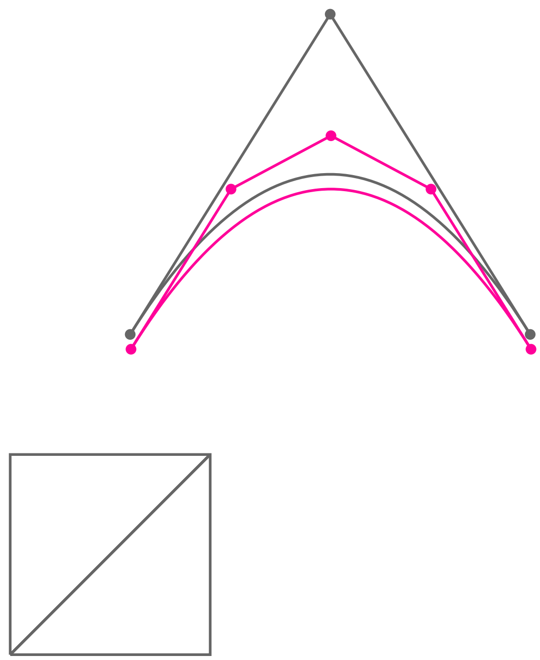

Fig. 9. Degree Raise a Quadric to a Cubic *the degree raised curve is translated a little bit for the comparison.*

of any knot that exists in both factor B-splines is raised from the original multiplicity in the first factor B-spline by the degree of the second B-spline, or is raised from the original multiplicity in the second factor B-spline by the degree of the first B-spline, depending on which one gives the higher final multiplicity.

Now, suppose the product knot vector is $u_0, u_1, u_2, \cdots$, and the blossoms of f_1 and f_2 be, $b_1(u_0, \cdots, u_{d_1-1})$ and $b_1(u_{d_1}, \cdots, u_{d_1+d_2-1})$. Define a symmetric multi-affine mapping as,

$$b(u_I) = \frac{\sum_{I_1 \oplus I_2 = I, \|I_1\|=d_1, \|I_2\|=d_2} b_1(u_{I_1}) * b_2(u_{I_2})}{\binom{d_1}{d_1+d_2}}, \tag{9}$$

where I is the index set $\{0, 1, \cdots, d_1 + d_2 - 1\}$, and $\{I_1, I_2\}$ is any partition of I, with fixed cardinality of d_1 and d_2 respectively; also, u_I (similarly for u_{I_1} and u_{I_2}) stands for $(u_0, u_1, \cdots, u_{d_1+d_2-1})$, i.e. all u-s with the subscripts in I. Obviously the diagonalization of b is the mulitplication of those of b_1 and b_2. That is, b is the blossom of $f(t) = f_1(t)f_2(t)$.

To gain some intuition, let us work out an example.

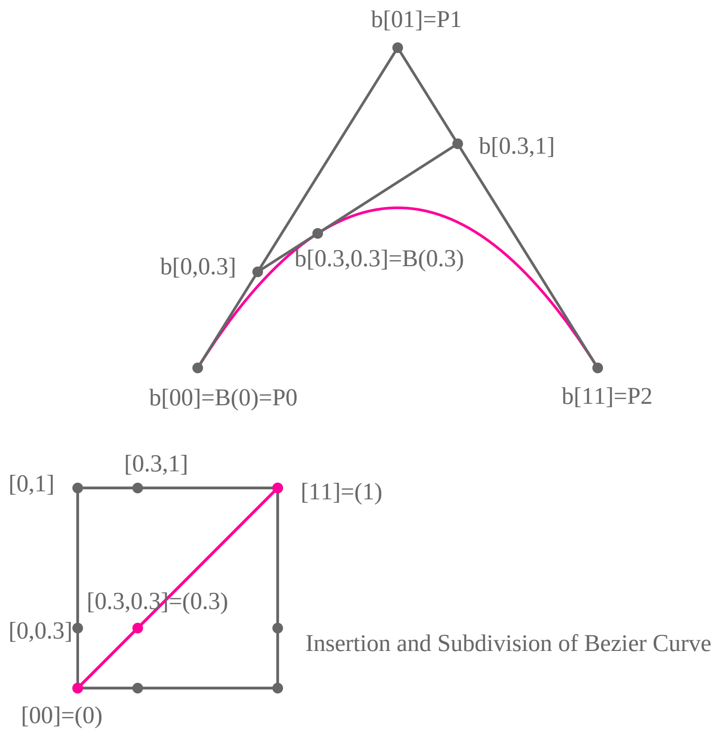

Fig. 10. Insertion and Subdivision

Example 11 *Suppose a degree 4 NURBS function* f_1 *has knot vector*[2] *,*

$$0, 1^2, 2, 3, 4^2, 5, 6^4, 7^2,$$

and another degree 5 NURBS function f_2*,*

$$0, 1, 2^3, 3, 4^2, 5^5, 6^2, 7.$$

The mutliplied degree 9 NURBS function $f = f_1 f_2$ *has knot vector,*

$$0^6, 1^7, 2^7, 3^6, 4^7, 5^9, 6^9, 7^7,$$

Let us focus on the segment of interval $[3, 4)$*, with knot vector of*

$$2^3, 3^6, 4^7, 5^2.$$

The blossom values (control points) of f *can be interpolated from the those of*

[2] The two NURBS in this example donot have open end conditions. Using the same method in this example, it is easy to see that the end control points of the multiplied NURBS have to be extrapolated, instead of interpolated as shown in this example for the inner control points. Therefore, this also demonstrate the necessity of making the two NURBS open end conditioned before doing any multiplication.

f_1 and f_2, as

$$b(2^3 3^6) \leftarrow \begin{pmatrix} b_1(2^3 3^1) * b_2(3^5) \\ b_1(2^2 3^2) * b_2(2^1 3^4) \\ b_1(2^1 3^3) * b_2(2^2 3^3) \\ b_1(2^0 3^4) * b_2(2^3 3^2) \end{pmatrix}$$

$$b(2^2 3^6 4^1) \leftarrow \begin{pmatrix} b_1(2^2 3^2 4^0) * b_2(2^0 3^4 4^1) \\ b_1(2^2 3^1 4^1) * b_2(2^0 3^5 4^0) \\ \\ b_1(2^1 3^3 4^0) * b_2(2^1 3^3 4^1) \\ b_1(2^1 3^2 4^1) * b_2(2^1 3^4 4^0) \\ \\ b_1(2^0 3^4 4^0) * b_2(2^2 3^2 4^1) \\ b_1(2^0 3^3 4^1) * b_2(2^2 3^3 4^0) \end{pmatrix}$$

$$b(2^1 3^6 4^2) \leftarrow \begin{pmatrix} b_1(2^1 3^3 4^0) * b_2(2^0 3^3 4^2) \\ b_1(2^1 3^2 4^1) * b_2(2^0 3^4 4^1) \\ b_1(2^1 3^1 4^2) * b_2(2^0 3^3 4^0) \\ \\ b_1(2^0 3^4 4^0) * b_2(2^1 3^2 4^2) \\ b_1(2^0 3^3 4^1) * b_2(2^1 3^3 4^1) \\ b_1(2^0 3^2 4^2) * b_2(2^1 3^4 4^0) \end{pmatrix}$$

etc, where the left arrow $\leftarrow$ means the left side is an affine combination of all rows at the right side (the weights are omitted here). All blossom terms in the right side, can in turn be interpolated from the original control points of f_1 and f_2.

For the acutall multiplication aglorithm, see [CRC06].

References

[CRC06] Xianming Chen, Richard Riesenfeld, and Elaine Cohen. Nurbs multiplication revisited. *Under Submission*, 2006.

[DG91] T. DeRose and R. Goldman. A tutorial introduction to blossoming. In H. Hagen and D. Roller, editors, *Geometric Modeling: Methods and Applications*, pages 267–286. Springer-Verlag, 1991.

[Gal98] J. Gallier. *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan Kaufman, 2 edition, 1998.

[Ram87] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. *System Research Center, DEC*, 1987.

[Ram89] L. Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design*, 6(4):323–359, 1989.

[Sei93] Hans-Peter Seidel. An introduction to polar forms. *IEEE Computer Graphics and Applications*, 13:38–46, 1993.