


Article

# Inverse Relations for Blossoms and Parametrisations of Rational Curves and Surfaces

Leonardo Fernández-Jambrina 

Matemática e Informática Aplicadas, E.T.S.I. Navales, Universidad Politécnica de Madrid, Avenida de la Memoria 4, E-28040 Madrid, Spain; leonardo.fernandez@upm.es

**Abstract:** In this paper, we make use of an inverse formula that relates the blossom of a NURBS curve, surface or Bézier triangle with its parametrisation, with no explicit reference to the control points and weights of the parametrisation. We make use of this inverse formula to raise and lower the degree elevation and reduction.

**Keywords:** B-splines; blossom; polarisation identities; degree elevation; degree reduction

**MSC:** 65D17

## 1. Introduction

The blossoming principle has been successfully used in many contexts in Computer Aided Geometric Design (CAGD) [1–4]. Blossoms of B-splines and Bézier curves and surfaces are used for degree elevation, for obtaining new control polygons and nets on transforming the domain, for knot insertion [5] or even for computing derivatives.

The parametrisation of a curve or surface is then just a simple case of blossom for which all the values of the parameter are equal.

This paper aims to show the relation that exists between parametrisations and blossoms of NURBS curves and surfaces by expressing the blossom in terms of just the parametrisation. It may be argued that this can be accomplished by computing the control polygon or net by iterated derivation of the parametrisation and then constructing the blossom. But here, a closed formula for writing the blossom as a fairly simple combination of points on the curve or surface is shown to be useful for many purposes.

In Section 2, the construction of the blossom of a Bézier curve or surface is reviewed. The construction of  $n$ -linear polar forms in a closed formula is derived and its extension to  $n$ -affine forms is shown in Section 3. The applications of these results to the calculation of control points and degree elevation and reduction are included in Section 4.

## 2. Blossoms or Polar Forms

A Bézier curve of degree  $n$  is just a curve with polynomial parametrisation of degree  $n$ ,

$$b(t) = \sum_{i=0}^n b_i B_i^n(t),$$

where  $B_i^n(t)$  are the usual Bernstein polynomials,

$$B_i^n(t) = \frac{n!}{i!(n-i)!} (1-t)^i t^{n-i},$$

and the shape of the curve is controlled by  $n+1$  points  $b_i$ ,  $i = 0, \dots, n$ , which are the vertices of the control polygon  $\{b_0, \dots, b_n\}$  of the curve [2].



**Citation:** Fernández-Jambrina, L. Inverse Relations for Blossoms and Parametrisations of Rational Curves and Surfaces. *Mathematics* **2024**, *12*, 2841. <https://doi.org/10.3390/math12182841>

Received: 27 July 2024

Revised: 5 September 2024

Accepted: 10 September 2024

Published: 12 September 2024



**Copyright:** © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Points  $b(t)$  on the curve are obtained by iterated interpolation at parameter  $t$  of the vertices of the control polygon by the de Casteljau algorithm,

$$\begin{aligned}
 b_i^1(t) &:= (1-t)b_i + tb_{i+1}, \quad i = 0, \dots, n-1, \\
 b_i^r(t) &:= (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t), \quad i = 0, \dots, n-r, \quad r = 1, \dots, n, \\
 b(t) &:= b_0^n(t).
 \end{aligned}
 \tag{1}$$

The blossom of a Bézier curve [2] is constructed by iterated application of the linear interpolation algorithm with different values  $t_1, \dots, t_n$  of the interpolation parameter  $t$ . Its uniqueness is shown in [6].

If  $\{b_0, \dots, b_n\}$  is the control polygon of a Bézier curve of degree  $n$  defined in the interval  $[0, 1]$  and  $t_1, \dots, t_n$  are values of the parameter of the curve, the blossom at  $t_1, \dots, t_n$  is denoted by  $b[t_1, \dots, t_n]$ ,

$$\begin{aligned}
 b_i^1[t_1] &:= (1-t_1)b_i + t_1b_{i+1}, \quad i = 0, \dots, n-1, \\
 b_i^r[t_1, \dots, t_r] &:= (1-t_r)b_i^{r-1}[t_1, \dots, t_{r-1}] + t_rb_{i+1}^{r-1}[t_1, \dots, t_{r-1}], \\
 b[t_1, \dots, t_n] &:= b_0^n[t_1, \dots, t_n], \quad i = 0, \dots, n-r, \quad r = 1, \dots, n.
 \end{aligned}
 \tag{2}$$

Since both the blossom  $b[t_1, \dots, t_n]$  and the parametrisation  $b(t)$  are related to the control polygon  $\{b_0, \dots, b_n\}$ , we are using the same letter  $b$  for both functions, with squared brackets for the blossom and curly brackets for the parametrisation.

In this notation, we readily check that the vertices of the control polygon are

$$b_0 = b[0, \dots, 0], \quad \dots, \quad b_i = b[0, \dots, 0, 1, \dots, 1], \quad \dots, \quad b_n = b[1, \dots, 1].$$

In the case of degree  $n = 2$ , for a control polygon  $\{b_0, b_1, b_2\}$ , we get its explicit form,

$$\begin{aligned}
 b_0^1[t_1] &= (1-t_1)b_0 + t_1b_1, \quad b_1^1[t_1] = (1-t_1)b_1 + t_1b_2, \\
 b[t_1, t_2] &:= (1-t_2)b_0^1[t_1] + t_2b_1^1[t_1] \\
 &= (1-t_1)(1-t_2)b_0 + (t_1(1-t_2) + t_2(1-t_1))b_1 + t_1t_2b_2,
 \end{aligned}$$

and we see it is symmetric,  $b[t_1, t_2] = b[t_2, t_1]$ .

This is a pyramidal algorithm in which each value is obtained from just two correlative ones in the previous step. For instance, for degree three,

$$\begin{array}{ccccccc}
 & & b_0 & & b_1 & & b_2 & & b_3 \\
 & & b_0^1[t_1] & & & & b_1^1[t_1] & & b_2^1[t_1] \\
 & & & & b_0^2[t_1, t_2] & & & & b_1^2[t_1, t_2] \\
 & & & & & & b_0^3[t_1, t_2, t_3] & & & & 
 \end{array}$$

Obviously,  $b[t_1, \dots, t_n]$  is not a point on the curve unless all values of the parameter  $t_1, \dots, t_n$  are equal. In this case, the original de Casteljau algorithm is recovered,  $b[t, \dots, t] = b(t)$ .

This construction has nice properties [2]. It is symmetric under any permutation  $\sigma$  of the order of the parameters  $b[t_1, \dots, t_n] = b[t_{\sigma(1)}, \dots, t_{\sigma(n)}]$  and is multiaffine. If one of the parameters is a barycentric combination of two numbers,  $\lambda t_i + \mu s_i$  with  $\lambda + \mu = 1$ , then

$$\begin{aligned}
 b[t_1, \dots, t_{i-1}, \lambda t_i + \mu s_i, t_{i+1}, \dots, t_n] &= \lambda b[t_1, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_n] \\
 &+ \mu b[t_1, \dots, t_{i-1}, s_i, t_{i+1}, \dots, t_n],
 \end{aligned}
 \tag{3}$$

that is, the blossom is affine in every variable.

Due to these properties the blossom is just a symmetric multiaffine polar form.



The blossoming principle is also extended to tensor product patches of surfaces of degree  $(m, n)$  parametrised by  $b(u, v)$ ,  $u, v \in [0, 1]$ , defined by a control net  $\{b_{00}, \dots, b_{0n}, \dots, b_{m0}, \dots, b_{mn}\}$ ,

$$b(u, v) = (B_0^m(u), \dots, B_m^m(u)) \begin{pmatrix} b_{00} & \dots & b_{0n} \\ \vdots & \ddots & \vdots \\ b_{m0} & \dots & b_{mn} \end{pmatrix} \begin{pmatrix} B_0^n(v) \\ \vdots \\ B_n^n(v) \end{pmatrix}.$$

Note that for surface patches, the parametrisation  $b(u, v)$  depends on two variables; however, we are keeping the same notation  $b$  for both of them.

The formulas are applied in two steps, one for each of the variables,

$$b[u_1, \dots, u_m; v_1, \dots, v_n] := b[u_1, \dots, u_m][v_1, \dots, v_n], \tag{9}$$

that is, the blossom is evaluated first on  $u_1, \dots, u_m$  for each of the columns of the control net and the resulting polygon  $\{b_0[u_1, \dots, u_m], \dots, b_n[u_1, \dots, u_m]\}$  is used to evaluate the blossom on  $v_1, \dots, v_n$ . Or, conversely, the blossom may be evaluated on  $v_1, \dots, v_n$  for each of the rows of the control net and the resulting polygon  $\{\tilde{b}_0[v_1, \dots, v_n], \dots, \tilde{b}_m[v_1, \dots, v_n]\}$  is used to evaluate the blossom on  $u_1, \dots, u_m$ . A similar construction is used to extend the blossoming principle to B-spline surfaces.

Finally, the extension to triangular Bézier patches (cfr. for instance [7] for a review) of surfaces of degree  $n$  parametrised by

$$b(u, v, w), \quad u, v, w \in [0, 1], \quad u + v + w = 1,$$

defined by a triangular control net  $\{b_{ijk} \in \mathbb{R}^3 : 0 \leq i, j, k \leq n, i + j + k = n\}$ ,

$$b(u, v, w) = \sum_{i+j+k=n} b_{ijk} \frac{n!}{i!j!k!} u^i v^j w^k,$$

is achieved in a similar fashion.

The blossom is evaluated on triplets of values of the parameters  $\mathbf{u}_1, \dots, \mathbf{u}_n$ ,  $\mathbf{u}_i = (u_i, v_i, w_i)$ , such that  $u_i + v_i + w_i = 1$ ,

$$\begin{aligned} b_{ijk}^1[\mathbf{u}_1] &:= u_1 b_{i+1jk} + v_1 b_{ij+1k} + w_1 b_{ijk+1}, \\ &0 \leq i, j, k \leq n-1, \quad i + j + k = n-1, \\ b_i^{(r)}[\mathbf{u}_1, \dots, \mathbf{u}_r] &:= u_r b_{i+1jk}^{(r-1)}[\mathbf{u}_1, \dots, \mathbf{u}_{r-1}] + v_r b_{ij+1k}^{(r-1)}[\mathbf{u}_1, \dots, \mathbf{u}_{r-1}] \\ &+ w_r b_{ijk+1}^{(r-1)}[\mathbf{u}_1, \dots, \mathbf{u}_{r-1}], \\ &0 \leq i, j, k \leq n-r, \quad i + j + k = n-r, \\ b[\mathbf{u}_1, \dots, \mathbf{u}_n] &:= b_{000}^{(n)}[\mathbf{u}_1, \dots, \mathbf{u}_n]. \end{aligned}$$

Written in its de Casteljau form, the blossom requires knowing the control polygon of the curve beforehand; that is, the curve must be written in Bézier or B-spline form. If we just have the parametrisation of the curve, we have to obtain the control points before calculating the blossom.

This could be achieved in principle taking into account the relation that exists between derivatives of the parametrisations at the endpoints. For instance, for Bézier curves over the interval  $[0, 1]$ ,

$$\frac{d^r b(0)}{du^r} = \frac{n! \Delta^r b_0}{(n-r)!}, \quad \Delta^1 b_i = b_{i+1} - b_i, \quad \Delta^s b_i = \Delta^{s-1} b_{i+1} - \Delta^{s-1} b_i, \tag{10}$$

but derivation is not an algebraic operation and it is numerically unstable.

However, an algebraic approach may be used for relating the parametrisation and the blossom directly.

A simple example of this arises naturally in Euclidean geometry. The scalar product and the length or norm of a vector  $\mathbf{v}$  in a real vector space are related by the usual expression,

$$\|\mathbf{v}\|^2 = \langle \mathbf{v}, \mathbf{v} \rangle . \tag{11}$$

This relation may be inverted provided that the norm comes from a scalar product,

$$\langle \mathbf{v}, \mathbf{w} \rangle = \frac{\|\mathbf{v} + \mathbf{w}\|^2 - \|\mathbf{v}\|^2 - \|\mathbf{w}\|^2}{2} . \tag{12}$$

Of course, this inverse expression is not unique. One can resort to the polarisation identity,

$$\langle \mathbf{v}, \mathbf{w} \rangle = \frac{\|\mathbf{v} + \mathbf{w}\|^2 - \|\mathbf{v} - \mathbf{w}\|^2}{4} , \tag{13}$$

with the same result.

Therefore, it is possible to relate a quadratic form, the norm, with a symmetric bilinear form, the scalar product. This is pretty obvious, since they share the same matrix representation, the Gram matrix.

In our case, we are not dealing with symmetric multilinear forms but with symmetric multiaffine forms. However, this example suggests that we may exchange the role of the square of the length for the parametrisation, and the scalar product for the blossom.

### 3. Multilinear Polar Forms

As a previous step before tackling the issue of writing the blossom of a parametrisation of degree  $n$  in terms of the parametrisation itself, we shall handle a related problem: the inversion formula, which expresses a symmetric multilinear form in terms of the associated  $n$ -atic form.

Let us denote by  $f(x_1, \dots, x_n)$  a symmetric  $n$ -linear form on a real linear space and  $q(x)$ , the associated  $n$ -atic form,

$$q(x) := f(x, \dots, x) , \tag{14}$$

which is a homogeneous function of degree  $n$ ,

$$q(\lambda x) = \lambda^n q(x) , \quad \lambda \in \mathbb{R} . \tag{15}$$

Also, we denote by  $q_N(S)$  the sum of the values of  $q$  acting on all ordered subsets of  $N$  elements of  $S$ ,

$$q_N(S) := \sum_{1 \leq i_1 < \dots < i_N \leq n} q(x_{i_1} + \dots + x_{i_N}) .$$

For instance,

$$\begin{aligned} q_1(S) &:= q(x_1) + \dots + q(x_n) , \\ q_2(S) &:= q(x_1 + x_2) + \dots + q(x_1 + x_n) + q(x_2 + x_3) + \dots + q(x_2 + x_n) \\ &\quad + \dots + q(x_{n-1} + x_n) , \\ q_n(S) &:= q(x_1 + \dots + x_n) . \end{aligned}$$

We would like to split the previous expression for  $q_n(S)$  in terms with just one, two, ...,  $n$  different values of  $x_i$ :

$$q_n(S) = f(x_1 + \dots + x_n, \dots, x_1 + \dots + x_n) = \sum_{N=1}^n f_N(S) ,$$

that is, we denote by  $f_N(S)$  the sum of all values of  $f(x_{i_1}, \dots, x_{i_n})$  where  $\{x_{i_1}, \dots, x_{i_n}\}$  are permutations, allowing for repetition, of just  $N$  different elements of  $S = \{x_1, \dots, x_n\}$ . For instance, the first and last terms are

$$\begin{aligned} f_1(S) &:= f(x_1, \dots, x_1) + \dots + f(x_n, \dots, x_n), \\ f_n(S) &:= n!f(x_1, \dots, x_n). \end{aligned} \tag{16}$$

With these definitions, we may split every expression for  $q_N(S)$  in terms with one to  $N$  different elements of  $S$

$$\begin{aligned} q_1(S) &= f_1(S), \\ q_{n-1}(S) &= (n-1)f_1(S) + (n-2)f_2(S) + \dots + f_{n-1}(S), \\ q_N(S) &= \sum_{M=1}^N \binom{n-M}{N-M} f_M(S). \end{aligned} \tag{17}$$

The reasoning leading to these expressions is clear. A term with just  $M$  different values of  $x_i$  as the ones in  $f_M(S)$ , for instance  $f(x_1, \dots, x_M)$ , appears in  $q_N(S)$  in all terms of the sort  $q(x_1 + \dots + x_N)$ , where  $\{x_1, \dots, x_M\}$  is a subset of  $\{x_1, \dots, x_N\}$ . These terms arise in the sum  $q_N(S)$  by fixing  $M$  values of  $x_i$ , which leaves free  $n - M$  values, among which we are to choose the remaining  $N - M$  ones.

The polar form  $f(x_1, \dots, x_n)$  appears just in  $q(x_1 + \dots + x_n)$ ; however, we may use the other expressions to simplify the unwanted terms. The alternating combination

$$\begin{aligned} \sum_{N=1}^n (-1)^{n-N} q_N(S) &= \sum_{N=1}^n (-1)^{n-N} \sum_{M=1}^N \binom{n-M}{N-M} f_M(S) \\ &= \sum_{M=1}^n \sum_{N=M}^n (-1)^{n-N} \binom{n-M}{N-M} f_M(S) \\ &= \sum_{M=1}^n (1-1)^{n-M} f_M(S) = f_n(S), \end{aligned}$$

is the required one for isolating the expression for the polar form in terms of the  $n$ -atic form. This result may be summarised as follows:

**Theorem 1.** *A symmetric  $n$ -linear form  $f$  on a real linear space may be written in terms of its associated  $n$ -atic form  $q$ ,*

$$f(x_1, \dots, x_n) = \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} \sum_{1 \leq i_1 < \dots < i_N \leq n} q(x_{i_1} + \dots + x_{i_N}). \tag{18}$$

This result is concise and useful; however, in CAGD, it deals with multiaffine instead of multilinear forms. This is not much of a problem, since we may consider barycentric combinations as linear combinations in which the sum of the coefficients is one.

Now, terms like  $q(x_1 + \dots + x_N)$  are not allowed since they are not evaluated on a barycentric combination. However, this difficulty can be easily overcome, since we may substitute this term in Theorem 1 by  $N^n q(\frac{x_1 + \dots + x_N}{N})$ , because  $q$  is homogeneous of degree  $n$ . This expression is suitable for the multiaffine case.

We keep the previous notations, but now  $f$  is a symmetric multiaffine form and  $q$  is the associated  $n$ -atic form.

**Theorem 2.** A symmetric  $n$ -affine form  $f$  on a real affine space may be written in terms of its associated  $n$ -atic form  $q$ ,

$$f(x_1, \dots, x_n) = \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} N^n \sum_{1 \leq i_1 < \dots < i_N \leq n} q\left(\frac{x_{i_1} + \dots + x_{i_N}}{N}\right). \tag{19}$$

Therefore, the problem of writing the blossom of a Bézier curve in terms of its parametrisation has a simple solution in closed form for every degree [3]:

**Corollary 1.** The blossom of a curve of degree  $n$  may be written in terms of its parametrisation,

$$b[t_1, \dots, t_n] = \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} N^n \sum_{1 \leq i_1 < \dots < i_N \leq n} b\left(\frac{t_{i_1} + \dots + t_{i_N}}{N}\right). \tag{20}$$

We may write the first examples:

$n = 2$ :

$$b[t_1, t_2] = \frac{1}{2} \left( 4b\left(\frac{t_1 + t_2}{2}\right) - b(t_1) - b(t_2) \right).$$

$n = 3$ :

$$b[t_1, t_2, t_3] = \frac{1}{6} \left( 27b\left(\frac{t_1 + t_2 + t_3}{3}\right) + b(t_1) + b(t_2) + b(t_3) - 8 \left( b\left(\frac{t_1 + t_2}{2}\right) + b\left(\frac{t_2 + t_3}{2}\right) + b\left(\frac{t_1 + t_3}{2}\right) \right) \right).$$

We see that the calculation of the blossom of a curve of degree  $n$  involves evaluating the parametrisation  $2^n - 1$  times and adding them. This means that the calculation increases exponentially with the degree, but in CAGD the standard case is  $n = 3$ , since interpolation with high degree polynomials causes undesired problems, such as the Runge phenomenon.

It must be noted that these formulas are valid also for  $B$ -splines, provided that  $t_1, \dots, t_n$  lie in the same segment of the spline, since splines are piecewise functions of degree  $n$ .

Formula (20) is also valid for Bézier triangles by replacing the values of  $t$  with the barycentric coordinates  $\mathbf{u}_i = (u_i, v_i, w_i)$ ,  $u_i + v_i + w_i = 1$ , of points in the parametric plane,

$$b[\mathbf{u}_1, \dots, \mathbf{u}_n] = \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} N^n \sum_{1 \leq i_1 < \dots < i_N \leq n} b\left(\frac{\mathbf{u}_{i_1} + \dots + \mathbf{u}_{i_N}}{N}\right). \tag{21}$$

For tensor product patches, the formulas may be applied in two steps, since

$$b[u_1, \dots, u_m; v_1, \dots, v_n] = b[u_1, \dots, u_m][v_1, \dots, v_n].$$

Therefore, there are simple inverse relations for blossoms and parametrisations for both curves and surfaces.

#### 4. Restriction, Degree Elevation and Reduction

One of the straightforward applications of the inverse formula is the possibility of calculating control and  $B$ -spline polygons and nets and weights directly from the parametrisation of the curve or surface. This is useful for instance to restrict the parametrisation of a curve to an interval  $[\alpha, \beta] \subset [0, 1]$ .

According to (4), we may calculate the vertices of the control polygon of a curve of degree  $n$ :

**Corollary 2.** The vertices  $\{b_0, \dots, b_n\}$  of a parametrised curve of degree  $n$ ,  $b(t)$ , defined on an interval  $[\alpha, \beta]$  are

$$b_i = \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} N^n \sum_{I=0}^i \binom{i}{I} \binom{n-i}{N-I} b\left(\frac{(N-I)\alpha + I\beta}{N}\right), \tag{22}$$

for  $i = 0, \dots, n$ , where the binomial coefficients are non-zero if and only if  $0 \leq N - I \leq n - i$ .

We may write the first instances of this formula as follows:

$n = 2$ :

$$\begin{aligned} b_0 &= b(\alpha), \\ b_1 &= -\frac{b(\alpha)}{2} - \frac{b(\beta)}{2} + 2b\left(\frac{\alpha + \beta}{2}\right), \\ b_2 &= b(\beta). \end{aligned}$$

$n = 3$ :

$$\begin{aligned} b_0 &= b(\alpha), \\ b_1 &= -b(\alpha) + \frac{b(\beta)}{6} - \frac{8}{3}b\left(\frac{\alpha + \beta}{2}\right) + \frac{9}{2}b\left(\frac{2\alpha + \beta}{3}\right), \\ b_2 &= \frac{b(\alpha)}{6} - b(\beta) - \frac{8}{3}b\left(\frac{\alpha + \beta}{2}\right) + \frac{9}{2}b\left(\frac{\alpha + 2\beta}{3}\right), \\ b_3 &= b(\beta). \end{aligned}$$

$n = 4$ :

$$\begin{aligned} b_0 &= b(\alpha), \\ b_1 &= -\frac{3}{2}b(\alpha) - \frac{b(\beta)}{24} + 2b\left(\frac{\alpha + \beta}{2}\right) - \frac{81}{8}b\left(\frac{2\alpha + \beta}{3}\right) + \frac{32}{3}b\left(\frac{3\alpha + \beta}{4}\right), \\ b_2 &= 7\frac{b(\alpha) + b(\beta)}{12} + \frac{40}{3}b\left(\frac{\alpha + \beta}{2}\right) - \frac{27}{4}\left(b\left(\frac{2\alpha + \beta}{3}\right) + b\left(\frac{\alpha + 2\beta}{3}\right)\right), \\ b_3 &= -\frac{b(\alpha)}{24} - \frac{3}{2}b(\beta) + 2b\left(\frac{\alpha + \beta}{2}\right) - \frac{81}{8}b\left(\frac{\alpha + 2\beta}{3}\right) + \frac{32}{3}b\left(\frac{\alpha + 3\beta}{4}\right), \\ b_4 &= b(\beta). \end{aligned}$$

Similarly, for Bézier triangles, we have

**Corollary 3.** The vertices  $\{b_{ijk} : 1 \leq i, j, k \leq n, i + j + k = n\}$  of a Bézier triangle of degree  $n$ ,  $b(u, w, w)$ , such that  $0 \leq u, v, w \leq 1, u + v + w = 1$ , restricted to a triangle of vertices  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  are

$$\begin{aligned} b_{ijk} &= \frac{1}{n!} \sum_{N=1}^n (-1)^{n-N} N^n \sum_{J=0}^j \sum_{K=0}^k \binom{j}{J} \binom{k}{K} \binom{i}{N-J-K} \\ &\times b\left(\frac{(N-J-K)\mathbf{a} + J\mathbf{b} + K\mathbf{c}}{N}\right), \end{aligned} \tag{23}$$

where the binomial coefficients are non-zero if and only if  $0 \leq N - J - K \leq i$ .

Another nice and interesting property of these formulas is that the concept of degree elevation is already built-in. Since a parametrisation  $b(t)$  of degree  $n$  is formally of any degree  $m$  greater than  $n$ , application of the blossom formula for degree  $m$  provides the degree-elevated blossom of  $b(t)$  directly.

**Example 1.** Control polygon and weights for an arc of a circle.

Let us consider an arc of a circle and its usual rational parametrisation

$$b(t) = \left( \frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2} \right), \quad t \in [\alpha, \beta] \subset [-1, 1].$$

The control points and weights for these parametrisations are readily derived

$$b_0 = \left( \frac{1-\alpha^2}{1+\alpha^2}, \frac{2\alpha}{1+\alpha^2} \right), \quad b_1 = \left( \frac{1-\alpha\beta}{1+\alpha\beta}, \frac{\alpha+\beta}{1+\alpha\beta} \right), \quad b_2 = \left( \frac{1-\beta^2}{1+\beta^2}, \frac{2\beta}{1+\beta^2} \right)$$

$$w_0 = 1 + \alpha^2, \quad w_1 = 1 + \alpha\beta, \quad w_2 = 1 + \beta^2.$$

And if we wish to write the parametrisation of the arc formally as a quartic,

$$b_0 = \left( \frac{1-\alpha^2}{1+\alpha^2}, \frac{2\alpha}{1+\alpha^2} \right), \quad w_0 = 1 + \alpha^2,$$

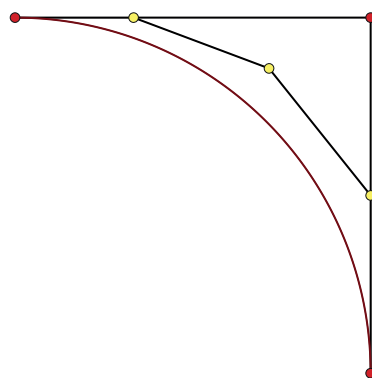
$$b_1 = \left( \frac{2-\alpha^2-\alpha\beta}{2+\alpha^2+\alpha\beta}, \frac{3\alpha+\beta}{2+\alpha^2+\alpha\beta} \right), \quad w_1 = 1 + \frac{\alpha^2+\alpha\beta}{2},$$

$$b_2 = \left( \frac{6-\alpha^2-4\alpha\beta-\beta^2}{6+\alpha^2+4\alpha\beta+\beta^2}, \frac{6(\alpha+\beta)}{6+\alpha^2+4\alpha\beta+\beta^2} \right), \quad w_2 = 1 + \frac{\alpha^2+4\alpha\beta+\beta^2}{6},$$

$$b_3 = \left( \frac{2-\alpha\beta-\beta^2}{2+\alpha\beta+\beta^2}, \frac{\alpha+3\beta}{2+\alpha\beta+\beta^2} \right), \quad w_3 = 1 + \frac{\alpha\beta+\beta^2}{2},$$

$$b_4 = \left( \frac{1-\beta^2}{1+\beta^2}, \frac{2\beta}{1+\beta^2} \right), \quad w_4 = 1 + \beta^2.$$

For instance, if we start from an arc of a circle with  $\alpha = 0, \beta = 1$ , we get  $\{(1, 0), (1, 1), (0, 1)\}$  as the control polygon and  $\{1, 1, 2\}$  as weights. As a quartic, the control polygon is  $\{(1, 0), (1, 1/2), (5/7, 6/7), (1/3, 1), (0, 1)\}$  and the list of weights is  $\{1, 1, 7/6, 3/2, 2\}$ . Both polygons for this arc may be seen in Figure 1.



**Figure 1.** Circle arc as a formally quartic curve.

The inversion formula is not only helpful to elevate the degree of a parametrisation, but also provides an approximation method for degree reduction.

There are several procedures in the literature for degree reduction. According to [8], degree reduction of a polynomial curve  $b(t)$  of degree  $n$  by another one of lower degree  $\tilde{b}(t)$  is a special case of polynomial approximation of lower degree. One approach makes use of Chebychev polynomials to obtain the uniform approximation of  $b(t)$ ; that is, the one for which

$$\|b - \tilde{b}\|_{\infty} := \max_{t \in [0,1]} |b(t) - \tilde{b}(t)|,$$

is smaller [9–11]. The use of dual Bernstein basis is shown in [12].

One could use another norm to minimize the error produced by degree reduction. For instance, any of the  $L^p$  norms for  $p \in \mathbb{N}$ ,

$$\|b - \tilde{b}\|_p := \sqrt[p]{\int_0^1 |b(t) - \tilde{b}(t)|^p dt}.$$

In [13], it is shown that the degree reduction that minimizes the error in the  $L^2$  norm (least squares approximation) is given by the best Euclidean approximation to the degree-raised control polygon. One of the problems of least squares approximations is that, unless we impose it, the resulting curve does not pass through the first and last vertices of the control polygon [2]. Extensions of the result by [13] to constrained degree reduction are to be found in [14] and to the Jacobi  $L^2$ -norm in [15].

A unified approach for degree reduction methods may be found in [16]. Degree reduction for spline curves is discussed in [17–19]. An extension to composite curves may be found in [20], considering geometric continuity [21–23].

Another approach would be a discrete approximation for which we impose some points on the curve instead of minimizing the distance from the approximating curve to the original curve. For achieving this, we just have to apply the blossom formula to the parametrisation with a degree lower than the actual one.

We must take into account that, although degree reduction is approximating a curve by one of a lower degree, we are conducting a standard Lagrange interpolation of points on the curve by a polynomial curve, with the nice feature of having closed formulas in terms of the parametrisation of the original curve. It has, of course, the same problems as Lagrange interpolation, such as the undesired Runge’s phenomenon when the degree is high; however, it can be managed using standard spline interpolation, for which the previous formulas are also valid.

Some advantages of this procedure are that the approximating curve passes through the endpoints of the original curve and the data are taken symmetrically along the curve. On the contrary, we may lose control of the mean distance to the original curve.

**Example 2.** Degree reduction of a polynomial curve parametrised by  $b(t)$ .

$$b(t) = (t^{10} - 3t^8 + 2t^7 - t^5 + t^2 + t - 3, -2t^{10} + t^9 - t^7 + t^5 + t^3 + t^2).$$

The control polygon for this curve of degree 10 is formed by the points

$$\left\{ (-3, 0), \left(-\frac{29}{10}, 0\right), \left(-\frac{25}{9}, \frac{1}{45}\right), \left(-\frac{79}{30}, \frac{3}{40}\right), \left(-\frac{37}{15}, \frac{1}{6}\right), \left(-\frac{575}{252}, \frac{13}{42}\right), \right. \\ \left. \left(-\frac{439}{210}, \frac{11}{21}\right), \left(-\frac{19}{10}, \frac{5}{6}\right), \left(-\frac{26}{15}, \frac{56}{45}\right), \left(-\frac{9}{5}, \frac{9}{5}\right), (-2, 1) \right\};$$

however, we may reduce the degree to 6 directly with the blossom inverse formula or with the least squares formalism. The results are compared in Figures 2–4. The blue curve is the original one and the red curve is the degree-reduced one. The last case corresponds to least squares approximation, but imposing the values of  $b(0)$  and  $b(1)$  as the right ones.

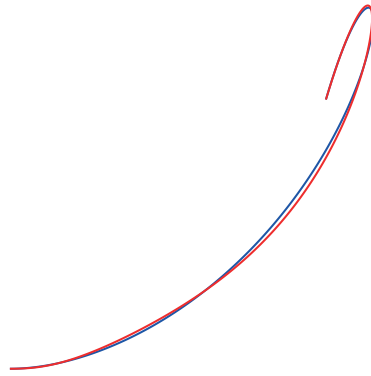


Figure 2. Degree reduction with blossom inverse formula.

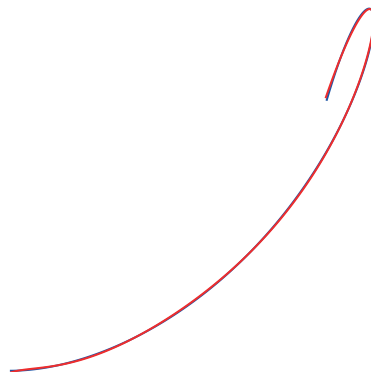


Figure 3. Degree reduction with least squares formalism.

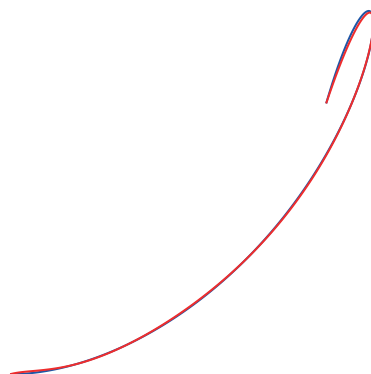


Figure 4. Degree reduction with least squares formalism, enforcing the first and last vertices.

It is true that the least squares formalism provides a globally closer fitting; however, on the contrary, the blossom inverse formula provides a control polygon that oscillates less and keeps the endpoints of the original one.

The degree reduction procedure need not be applied only to polynomial or rational curves, but can be used for approximating non-algebraic curves. With this procedure the approximating curve passes through the endpoints and is exact for polynomial curves of the same degree.

**Example 3.** Approximating curve for the logarithm function,  $b(t) = (t, \ln t)$ ,  $t \in [1/2, 1]$ .

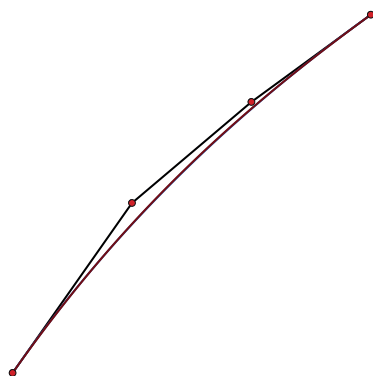
If we approximate  $b(t)$  with a cubic curve,  $\tilde{b}(t)$ , with control polygon

$$\left\{ \left( \frac{1}{2}, -\ln 2 \right), \left( \frac{2}{3}, \frac{65 \ln 2 - 43 \ln 3}{6} \right), \left( \frac{5}{6}, \frac{27 \ln 5 + 4 \ln 2 - 43 \ln 3}{6} \right), (1, 0) \right\},$$

the  $L^2$ -error committed,  $2 \cdot 10^{-7}$ ,

$$\int_{1/2}^1 \|\tilde{b}(t) - b(t)\|^2 dt,$$

is of the same order as the best  $L^2$ -cubic approximation,  $10^{-7}$ , as can be seen in Figure 5, where the logarithm and the approximating curve are nearly indistinguishable.



**Figure 5.** Approximating curve for the logarithm function.

## 5. Conclusions

In this paper, a formula that furnishes the blossom of a NURBS parametrisation with no explicit reference to control points, just in terms of points on the NURBS, is provided for all cases of polynomial, rational, NURBS curves and surfaces. The provided solution is not unique, but it is the most symmetric.

It has been shown that this inverse formula may be useful for obtaining control points and weights for any domain of the parametrisation. As applications, it has been used for restricting the domain of a parametrisation, for elevating and reducing its degree and for approximating non-rational parametrisations.

**Funding:** This research received no external funding.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. de Casteljau, P. *Shape Mathematics and CAD*; Kogan Page: London, UK, 1986.
2. Farin, G. *Curves and Surfaces for CAD*, 5th ed.; Morgan Kaufmann: San Francisco, CA, USA, 2001.
3. Ramshaw, L. Blossoms are polar forms. *Comput. Aided Geom. Des.* **1989**, *6*, 323–359. [[CrossRef](#)]
4. Seidel, H.-P. A new multiaffine approach to B-splines. *Comput. Aided Geom. Des.* **1989**, *6*, 23–32. [[CrossRef](#)]
5. Seidel, H.-P. Knot insertion from a blossoming point of view. *Comput. Aided Geom. Des.* **1988**, *5*, 81–86. [[CrossRef](#)]
6. Tuncer, O.O.; Simeonov, P.; Goldman, R. On the uniqueness of the multirational blossom. *Comput. Aided Geom. Des.* **2023**, *107*, 102252. [[CrossRef](#)]
7. Farin, G. Triangular Bernstein-Bézier patches. *Comput. Aided Geom. Des.* **1986**, *3*, 83–127. [[CrossRef](#)]
8. Farouki, R.T. The Bernstein polynomial basis: A centennial retrospective. *Comput. Aided Geometric Des.* **2012**, *29*, 379–419. [[CrossRef](#)]
9. Eck, M. Degree reduction of Bézier curves. *Comput. Aided Geom. Des.* **1993**, *10*, 237–252.
10. Peterson, J. Degree reduction of Bézier curves. *Comput.-Aided Des.* **1991**, *23*, 460–461.
11. Watkins, M.A.; Worsey, A.J. Degree reduction of Bézier curves. *Comput.-Aided Des.* **1988**, *20*, 398–405. [[CrossRef](#)]
12. Woźny, P.; Lewanowicz, S. Multi-degree reduction of Bézier curves with constraints, using dual Bernstein basis polynomials. *Comput. Aided Geometric Des.* **2009**, *26*, 566–579. [[CrossRef](#)]
13. Lutterkort, D.; Peters, J.; Reif, U. Polynomial degree reduction in the  $L^2$ -norm equals best Euclidean approximation of Bézier coefficients. *Comput. Aided Geom. Des.* **2000**, *16*, 607–612. [[CrossRef](#)]
14. Ahn, Y.J.; Lee, B.; Park, Y.; Yoo, J. Constrained polynomial degree reduction in the  $L^2$ -norm equals best weighted Euclidean approximation of Bézier coefficients. *Comput. Aided Geom. Des.* **2004**, *21*, 181–191. [[CrossRef](#)]

15. Ait-Haddou, R.; Bartoň, M. Constrained multi-degree reduction with respect to Jacobi norms. *Comput. Aided Geom. Des.* **2016**, *42*, 23–30. [[CrossRef](#)]
16. Szafnicki, B. A unified approach for degree reduction of polynomials in the Bernstein basis. Part I: Real polynomials. *J. Comput. Appl. Math.* **2002**, *142*, 287–312. [[CrossRef](#)]
17. Piegl, L.; Tiller, W. Algorithm for degree reduction of B-spline curves. *Comput. Aided Design* **1995**, *27*, 101–110. [[CrossRef](#)]
18. Wolters, H.; Wu, G.; Farin, G. Degree reduction of B-spline curves. In *Geometric Modelling*; Farin, G., Bierl, H., Brunnett, G., DeRose, T., Eds.; Springer: Berlin/Heidelberg, Germany, 1998.
19. Yong, J.; Hu, S.; Sun, J.; Tan, X. Degree reduction of B-spline curves. *Comput. Aided Geometric Des.* **2001**, *18*, 117–128. [[CrossRef](#)]
20. Gospodarczyk, P.; Lewanowicz, S.; Woźny, P. Degree reduction of composite Bézier curves. *Appl. Math. Comput.* **2017**, *293*, 40–48. [[CrossRef](#)]
21. Lu, L. Explicit  $G^2$ -constrained degree reduction of Bézier curves by quadratic optimization. *J. Comput. Appl. Math.* **2013**, *253*, 80–88. [[CrossRef](#)]
22. Lu, L.; Wang, G. Optimal multi-degree reduction of Bézier curves with  $G^2$ -continuity. *Comput. Geom. Des.* **2006**, *23*, 673–683. [[CrossRef](#)]
23. Rababah, A.; Mann, S. Linear methods for  $G^1$ ,  $G^2$  and  $G^3$ —Multi-degree reduction of Bézier curves. *Comput.-Aided Des.* **2013**, *45*, 405–414. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.