Comparing Bézier curves and surfaces for coincidence

Krassimira Vlachkova

Abstract It is known that Bézier curves and surfaces may have multiple representations by different control polygons. The polygons may have different number of control points and may even be disjoint. This phenomenon causes difficulties in variety of applications where it is important to recognize cases where different representations define same curve (surface) or partially coincident curves (surfaces). The problem of finding whether two arbitrary parametric polynomial curves are the same has been addressed in Pekerman et al. [Are two curves the same?, *Comput.-Aided Geom. Des. and Appl.*, 2(1-4)(2005), pp. 85-94]. There the curves are reduced into canonical irreducible forms using the monomial basis, then they are compared and their shared domains, if any, are identified. Here we present an alternative geometric algorithm based on subdivision that compares two input control polygons and reports the coincidences between the corresponding Bézier curves if they are present. We generalize the algorithm for tensor product Bézier surfaces. The algorithms are implemented and tested using Mathematica package. The experimental results are presented.

1 Introduction

Comparing Bézier curves and surfaces for coincidence is an important problem in computer-aided design (CAD) which arises in various applications. Suppose we are given the control polygons of two curves (surfaces) which are obtained by different sources, e. g. they can be generated by different software packages. The curves (surfaces) have to be stitched together to obtain a new curve (surface) which is continuous. It is possible that the two control polygons may have different number of control points and may even be disjoint but nevertheless they represent same

Krassimira Vlachkova
Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski"
Blvd. James Bourchier 5, 1164 Sofia, Bulgaria, e-mail: `krassivl@fmi.uni-sofia.bg`

curve (surface), see Fig. 2. Therefore it is important to find out whether the control polygons represent same curve (surface) and in this case to determine the coincident part of the two curves (surfaces). By *coincident* curves (surfaces) we mean that they occupy the same locus of points in $\mathbb{R}^3$, i. e. they are *geometrically equivalent* as defined by Denker and Heron [4]. A *shared domain* of two curves (surfaces) with same parametrization is the intersection of their domains of definition.

Different representations of a parametric polynomial curve occur if it has been degree elevated and/or reparameterized by a composition with a polynomial. While the degree elevation is comparatively easy to be detected, it is usually hard to find out whether the curve has undergone polynomial composition. Decomposition of polynomials is a classical problem in computer algebra. The first algorithm for polynomial decomposition was proposed by Barton and Zippel [1, 2]. An alternative decomposition algorithm was proposed by Kozen and Landau [6]. Later, it has been improved by von zur Gathen [5]. Today polynomial decomposition is a standard built-in function in the computer algebra systems as Maple (the function compoly), Matlab (the function polylib:decompose), Mathematica (the function Decompose). The applied algorithms are known for some of these systems. For example, Matlab applies Barton and Zippel's algorithm [2]. Currently, Mathematica does not disclose such information to the public.

A curve is *irreducible* if it is not a result of a polynomial composition and has not been degree elevated. Sánchez-Reyes [8] has showed uniqueness of the control points (up to reverse order) of an irreducible Bézier curve of arbitrary degree. He pointed out that this result is a straightforward consequence of a previous and more general result by Berry and Patterson [3] for rational Bézier curves. The uniqueness of the control points for cubic curves has also been studied by Wang et al. [10].

The problem of finding whether two arbitrary polynomial curves are the same has been considered by Pekerman et al. [7] where an algorithm for a polynomial decomposition is proposed and used. The authors do not compare the algorithm to any of the previously known algorithms as [1, 2, 5, 6]. They point out that the computations can be done efficiently since the degree of the polynomial to be decomposed is usually low for practical purposes. In [7] the curves are reduced into canonical irreducible forms, then they are compared and their shared domain, if any, is identified.

Here we present an alternative geometric algorithm for comparing Bézier curves for coincidence. Our algorithm works in two phases. In the first phase the control polygons are tested for reducibility. We adopt a set of routines from Mathematica package and transform the control polygons to an irreducible form which is unique as shown in [8]. In the second phase the obtained irreducible control polygons are checked for coincidences. We use a new geometric approach based on subdivision. In this approach the usage of a monomial basis and consequent conversion into canonical form are avoided. In addition, simple geometric conditions for checking whether two control polygons define different irreducible curves are obtained. Given two irreducible Bézier curves our Algorithm 1 reports whether they are different, disjoint, or coincident and in that case reports the control points of the coincident part.

We generalize Algorithm 1 for the case of tensor product Bézier surfaces. Since computer algebra systems currently do not support bivariate polynomial decomposition, our aim was to reduce the problem for surfaces to the problem for curves.

We prove in Theorem 1 that two irreducible tensor product Bézier surfaces of same degree coincide if and only if their control polygons coincide (up to different enumeration of the control points). Given two irreducible tensor product Bézier curves of same degree (m,n) our Algorithm 2 reports whether they are different, disjoint, or coincident and in that case reports the control points of the coincident part. Coincidences also can occur in the case where the degrees of the two irreducible surfaces are (n,m) and $(n+m,n+m)$, respectively. Our algorithm doesn't cover this case.

We implemented and tested our algorithms using Mathematica package. Real examples illustrating our presentation are included.

The paper is organized as follows. In Sect. 2 we consider the problem for Bézier curves and propose Algorithm 1 based on subdivision that reports two irreducible Bézier curves as different, disjoint, or coincident. In case of coincidence Algorithm 1 reports the control points of the coincident part. The problem for tensor product Bézier surfaces is considered in Sect. 3 and Algorithm 2 that compares two irreducible tensor product Bézier surfaces of same degree for coincidence is proposed. Algorithm 2 reports these surfaces as different, disjoint, or coincident and in that case reports the control points of the coincident part.

2 Coincidence of Bézier curves

In this section we present an algorithm based on subdivision for comparing Bézier curves for coincidence. Our algorithm takes as input two irreducible Bézier curves of same degree $\mathbf{b}^1(t)$, $t \in [0,1]$, and $\mathbf{b}^2(u)$, $u \in [0,1]$, and reports if they are different. If not the algorithm reports them as disjoint or coincident. In the case of coincidence the algorithm reports the control points of their coincident part.

Let $\mathbf{b}(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$, where $\mathbf{b}_i$, $i = 0,\ldots,m$, are points in $\mathbb{R}^3$ and $B_i^m(t)$ are the Bernstein polynomials defined for $0 \le t \le 1$ by $B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}$, where $\binom{m}{i}$ for $0 \le i \le m$ are the binomial coefficients. We assume that $\binom{m}{i} = 0$ if $i < 0$ or $i > m$.

The curve $\mathbf{b}$ is degree elevated if and only if $\Delta^m \mathbf{b}_0 = \mathbf{0}$, where Δ^r is the forward finite difference of order r, $r \ge 1$, defined recursively by $\Delta^0 \mathbf{b}_i = \mathbf{b}_i$, $\Delta^r \mathbf{b}_i = \Delta^{r-1}\mathbf{b}_{i+1} - \Delta^{r-1}\mathbf{b}_i$. We note that $\Delta^m \mathbf{b}_0$ is the coefficient of t^m in the canonical form of $\mathbf{b}(t)$. If $\mathbf{b}$ has been degree elevated then the control points $\hat{\mathbf{b}}_i$ of the degree reduced curve are computed recursively as

$$\hat{\mathbf{b}}_i = \frac{m}{m-i}\mathbf{b}_i - \frac{i}{m-i}\hat{\mathbf{b}}_{i-1}, \qquad i = 0,\ldots,m-1.$$

In order to check $\mathbf{b}$ for a composition by a polynomial we use the built-in function Decompose in Mathematica. This function returns the decomposed curve $\tilde{\mathbf{b}}(t)$ in its canonical form $\tilde{\mathbf{b}}(t)=\sum_{i=0}^{m_1}\mathbf{a}_i t^i$, where $m_1\le m$, $\mathbf{a}_i\in\mathbb{R}^3$. Then the control points of $\tilde{\mathbf{b}}(t)$ are

$$\tilde{\mathbf{b}}_i=\sum_{j=0}^{i}\frac{\binom{i}{j}}{\binom{m_1}{j}}\mathbf{a}_j,\qquad i=0,\dots,m_1.$$

Degree elevation and decomposition are non-commutative operations, as pointed out in [7]. Hence, in order to check the Bézier curve $\mathbf{b}$ for irreducibility we have to perform alternate check for degree elevation and decomposition while both attempts fail. Finally, we obtain the control points $\mathbf{p}_i$, $i=0,\dots,n$ of the irreducible form of $\mathbf{b}$ and we store the last two finite differences $\Delta^{n-1}\mathbf{b}_0$ and $\Delta^n\mathbf{b}_0$. Note that $\Delta^n\mathbf{b}_0\neq\mathbf{0}$.

Let $\mathbf{b}^1(t)=\sum_{i=0}^{n}\mathbf{b}_iB_i^n(t)$, $t\in[0,1]$, and $\mathbf{b}^2(u)=\sum_{i=0}^{n}\mathbf{p}_iB_i^n(u)$, $u\in[0,1]$, $n\in\mathbb{N}$, $n\ge 2$, be two irreducible Bézier curves such that their control polygons do not coincide. Our aim is to check out whether $\mathbf{b}^1$ and $\mathbf{b}^2$ represent same curve and in this case to find their coincident part, if any. Suppose that $\mathbf{b}^1$ and $\mathbf{b}^2$ represent same curve. Then we can consider one of them, say $\mathbf{b}^2$, as obtained from $\mathbf{b}^1$ by subdivision at two parameters a and b, $a,b\in\mathbb{R}$, see Fig. 1. Hence, $\mathbf{b}^2$ is defined in the interval

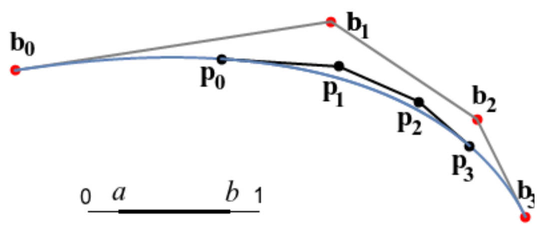


Fig. 1: Curve $\mathbf{b}^2$ with control points $\mathbf{p}_i$, $i=1,\dots,n$ is obtained from curve $\mathbf{b}^1$ with control points $\mathbf{b}_i$, $i=1,\dots,n$, by subdivision at a and b, where $a,b\in\mathbb{R}$.

with endpoints a and b which are uniquely defined. This interval is an image of the interval $[0,1]$ by the affine map $t=(b-a)u+a$, $0\le u\le 1$. We have

$$\sum_{i=0}^{n}\mathbf{p}_iB_i^n(u)=\sum_{i=0}^{n}\mathbf{b}_iB_i^n((b-a)u+a),\qquad 0\le u\le 1. \tag{1}$$

Next, we express a and b in terms of the control points $\mathbf{p}_i,\mathbf{b}_i$, $i=0,\dots,n$. We differentiate $n-1$ times both sides of (1). The derivative of order $n-1$ is obtained by applying consecutively $n-1$ finite differences and one evaluation by de Casteljau algorithm. We have

$$n!\sum_{i=0}^{n-(n-1)}\Delta^{n-1}\mathbf{p}_iB_i^{n-(n-1)}(u)=n!(b-a)^{n-1}\sum_{i=0}^{n-(n-1)}\Delta^{n-1}\mathbf{b}_iB_i^{n-(n-1)}((b-a)u+a). \tag{2}$$

Hence, from (2) we have

$$(1-u)\Delta^{n-1}\mathbf{p}_0 + u\Delta^{n-1}\mathbf{p}_1 = (b-a)^{n-1}\big((1-a-(b-a)u)\Delta^{n-1}\mathbf{b}_0 + (a+(b-a)u)\Delta^{n-1}\mathbf{b}_1\big). \quad (3)$$

For the derivatives of order n in both sides of (1) we obtain

$$\Delta^n\mathbf{p}_0 = (b-a)^n\Delta^n\mathbf{b}_0. \quad (4)$$

From (4) we have $b-a = \sqrt[n]{|\Delta^n\mathbf{p}_0|/|\Delta^n\mathbf{b}_0|}$ for n odd, and $b-a = \pm\sqrt[n]{|\Delta^n\mathbf{p}_0|/|\Delta^n\mathbf{b}_0|}$ for n even. Note that we have stored in advance $\Delta^{n-1}\mathbf{p}_0$, $\Delta^n\mathbf{p}_0$, $\Delta^{n-1}\mathbf{b}_0$, $\Delta^n\mathbf{b}_0$ while performing check for degree elevation and we have $\Delta^n\mathbf{p}_0 \neq \mathbf{0}$ and $\Delta^n\mathbf{b}_0 \neq \mathbf{0}$.

We have from (3) for $u=0$ and $u=1$, respectively

$$\Delta^{n-1}\mathbf{p}_0 = (b-a)^{n-1}\big((1-a)\Delta^{n-1}\mathbf{b}_0 + a\Delta^{n-1}\mathbf{b}_1\big), \quad (5)$$

$$\Delta^{n-1}\mathbf{p}_1 = (b-a)^{n-1}\big((1-b)\Delta^{n-1}\mathbf{b}_0 + b\Delta^{n-1}\mathbf{b}_1\big). \quad (6)$$

Then we compute

$$A = \frac{\Delta^{n-1}\mathbf{p}_0}{(b-a)^{n-1}} - \Delta^{n-1}\mathbf{b}_0, \quad B = \frac{\Delta^{n-1}\mathbf{p}_1}{(b-a)^{n-1}} - \Delta^{n-1}\mathbf{b}_0 = A + \frac{\Delta^n\mathbf{p}_0}{(b-a)^{n-1}}. \quad (7)$$

From (5) and (6) we obtain

$$a\Delta^n\mathbf{b}_0 = A, \quad b\Delta^n\mathbf{b}_0 = B. \quad (8)$$

In the case where n is even there are two possibilities for A and B in (8) but only one of them is correct. For the other possibility there exist no a and b that satisfy (8).

We note that in the case where $a,b \notin [0,1]$ it is better to subdivide $\mathbf{b}^2$ instead of $\mathbf{b}^1$ to avoid extrapolation which is not numerically stable for large parameter values.

The geometric meaning of (4) and (8) is that the four vectors $\Delta^n\mathbf{b}_0$, $\Delta^n\mathbf{p}_0$, A, and B are collinear. The corresponding coefficients of proportion are $(b-a)^n$ in (4), a and b in (8). Another interpretation of the geometric meaning of the parameters a and b was proposed by Sánchez-Reyes in [9].

The next lemma provides simple sufficient geometric conditions for two irreducible Bézier curves to be different.

Lemma 1. *The irreducible Bézier curves* $\mathbf{b}^1$ *and* $\mathbf{b}^2$ *are different if any of the next three statements is true.*

(i) $\Delta^n\mathbf{b}_0$ *and* $\Delta^n\mathbf{p}_0$ *are not collinear;*
(ii) n *is even and* $\Delta^n\mathbf{b}_0$ *and* $\Delta^n\mathbf{p}_0$ *have opposite directions;*
(ii) $\Delta^n\mathbf{b}_0$ *is collinear with at most one of* A *and* B *defined by (7).*

Proof. Statements (i) and (ii) follows from (4). Statement (iii) follows from (8). □

Next, we propose Algorithm 1 for comparing two irreducible Bézier curves for coincidence.

Algorithm 1 Comparison for Coincidence of two Irreducible Bézier Curves

Input: Irreducible Bézier curves $\mathbf{b}^1$ and $\mathbf{b}^2$ represented by their control points $\{\mathbf{b}_i\}_{i=0}^n$, $\{\mathbf{p}_i\}_{i=0}^n$, respectively; $\Delta^{n-1}\mathbf{b}_0$, $\Delta^n\mathbf{b}_0$; $\Delta^{n-1}\mathbf{p}_0$, $\Delta^n\mathbf{p}_0$.

Output: (i) $\mathbf{b}^1$ and $\mathbf{b}^2$ are different;
(ii) $\mathbf{b}^1$ and $\mathbf{b}^2$ are disjoint;
(iii) $\mathbf{b}^1$ and $\mathbf{b}^2$ have coincident part $\bar{\mathbf{b}}$. Report the control points of $\bar{\mathbf{b}}$.

Step 1. **if** $\Delta^n\mathbf{b}_0$ and $\Delta^n\mathbf{p}_0$ are not collinear
  then output (i) and **stop**;
  else if n is even and $\Delta^n\mathbf{b}_0$ and $\Delta^n\mathbf{p}_0$ have opposite directions
    then output (i) and **stop**;
    else compute $l = \sqrt[n]{|\Delta^n\mathbf{p}_0|/|\Delta^n\mathbf{b}_0|}$, $A = \Delta^{n-1}\mathbf{p}_0/l^{n-1} - \Delta^{n-1}\mathbf{b}_0$, $B = A + \Delta^n\mathbf{p}_0/l^{n-1}$
    end if
end if

Step 2. **if** both A and B are collinear to $\Delta^n\mathbf{b}_0$
  then go to Step 4;
  else if n is odd
    then output (i) and **stop**;
    else compute $A_1 = -A - 2\Delta^{n-1}\mathbf{b}_0$ and $B_1 = -B - 2\Delta^{n-1}\mathbf{b}_0$
    end if
end if

Step 3. **if** $\Delta^n\mathbf{b}_0$ is collinear with at most one of A_1 and B_1
  then output (i) and **stop**;
  else go to Step 4
end if

Step 4. Compute a and b such that $a\Delta^n\mathbf{b}_0 = A$, $b\Delta^n\mathbf{b}_0 = B$.

Step 5. Subdivide $\mathbf{b}^1$ at a and b and compare the obtained control points to $\mathbf{p}_i$, $i = 0, \ldots, n$.
if there are at least two corresponding non-coincident control points
  then output (i) and **stop**;
  else compute the intersection I of $[0,1]$ with the interval with endpoints a and b.
end if
if $I = \emptyset$ **then** output (ii) and **stop**;
  else output (iii) with the control points computed in Step 5 and **stop**
end if

In Example 1 the curves $\mathbf{b}^1$ and $\mathbf{b}^2$ are compared for coincidence using Algorithm 1.

Example 1. The control points of the curve $\mathbf{b}^1$ of degree 4 and $\mathbf{b}^2$ of degree 8 are shown in Table 1. The curve $\mathbf{b}^1$ is irreducible. After decomposition we obtain that $\mathbf{b}^2$ is a result of a composition by the polynomial $t^2 + t$ of the curve

$$\mathbf{b}(t) = \big(-1 + 12t - 30t^2 + 24.8t^3 - 4.6t^4, -1 + 4.8t - 4.2t^2 - 0.4t^3 - 0.1t^4\big).$$

The irreducible form of $\mathbf{b}^2$ has degree $n = 4$, its control points are shown in Table 1 and the control polygons of both forms of $\mathbf{b}^2$ are shown in Fig. 2. The control polygons of the irreducible curves $\mathbf{b}^1$ and $\mathbf{b}^2$ are shown in Fig. 3**a.** By applying Algorithm 1 we obtain that $\mathbf{b}^2$ is a result of subdivision of $\mathbf{b}^1$ at $a = -.05$ and $b = .4$. The control points of the coincident part $\bar{\mathbf{b}}$ of $\mathbf{b}^1$ and $\mathbf{b}^2$ are shown in Table 1. The curve $\bar{\mathbf{b}}$ and its control polygon are shown in Fig. 3**b.**

Table 1: Comparison of Bézier curves $\mathbf{b}^1$ and $\mathbf{b}^2$ for coincidence

$\mathbf{b}^1$	$\mathbf{b}^2$	irreducible $\mathbf{b}^2$	coincident part $\bar{\mathbf{b}}$
(-1, -1)	(-1.67813, -1.25045)	(-1.67813, -1.25045)	(-1, -1)
(2, .2)	(-1.06849, -1.04105)	(0.030555, -0.663532)	(0.2, -0.52)
(0, .7)	(-0.513367, -0.822876)	(0.59886, -0.21639)	(0.6, -0.152)
(-.8, .4)	(-0.0474957, -0.602289)	(0.61272, 0.08232)	(0.5968, 0.0976)
(1.2, -.9)	(0.300317, -0.386845)	(0.46944, 0.21984)	(0.46944, 0.21984)
	(0.513146, -0.18546)		
	(0.59217, -0.0086048)		
	(0.5616, 0.131384)		
	(0.469442, 0.219838)		

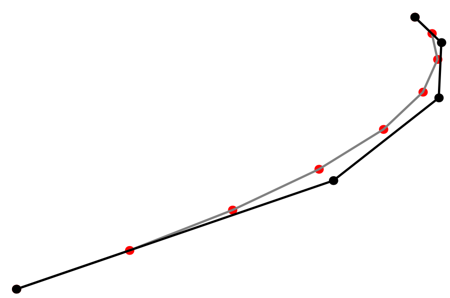


Fig. 2: The control polygons of curve $\mathbf{b}^2$ and its irreducible form. The corresponding control points are shown in Table 1.

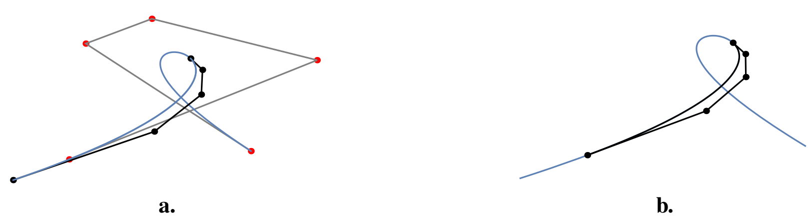


Fig. 3: **a.** Curve $\mathbf{b}$ and the control polygons of the irreducible form of $\mathbf{b}^1$ and $\mathbf{b}^2$; **b.** Curve $\mathbf{b}$ and the control polygon of the coincident part $\bar{\mathbf{b}}$ of $\mathbf{b}^1$ and $\mathbf{b}^2$.

3 Coincidence of Tensor Product Bézier Surfaces

In this section we present an algorithm for comparing tensor product Bézier surfaces for coincidence. Our algorithm takes as input two irreducible tensor product Bézier surfaces of same degree $\mathbf{b}^1(s,t)$, $(s,t) \in [0,1]\times[0,1]$, and $\mathbf{b}^2(u,v)$, $(u,v) \in [0,1]\times[0,1]$, and reports if they are different. If not then the algorithm reports them as disjoint or coincident. In the case of coincidence the algorithm reports the control points of their coincident part.

We start with a definition of irreducibility of tensor product Bézier surface which is consistent with the analogous definition for Bézier curve.

Definition 1. The tensor product Bézier surface $\mathbf{b}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{b}_{ij} B_i^n(u) B_j^m(v)$, $0 \le u \le 1$, $0 \le v \le 1$, $m,n \in \mathbb{N}$, is *irreducible* if the curves $\mathbf{c}_j^1(u)$ with control points $\{\mathbf{b}_{ij}\}_{i=0}^{n}$, $j = 0,\dots,m$, and $\mathbf{c}_i^2(v)$ with control points $\{\mathbf{b}_{ij}\}_{j=0}^{m}$, $i = 0,\dots,n$, are irreducible.

Next, we prove a theorem that provides necessary and sufficient condition for coincidence of two irreducible tensor product Bézier surfaces.

Theorem 1. *Let* $\mathbf{b}^1(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{b}_{ij} B_i^n(s) B_j^m(t)$ *defined for* $(s,t) \in [0,1]\times[0,1]$, *and* $\mathbf{b}^2(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{p}_{ij} B_i^n(u) B_j^m(v)$ *defined for* $(u,v) \in [0,1]\times[0,1]$, *be irreducible Bézier surfaces. Then* $\mathbf{b}^1$ *and* $\mathbf{b}^2$ *coincide if and only if their control polygons coincide (up to different enumeration of the control points).*

Proof. $\Rightarrow$ Suppose that $m \ne n$. Since $\mathbf{b}^1$ and $\mathbf{b}^2$ coincide then there exist smooth functions $s = \varphi(u,v)$ and $t = \psi(u,v)$ such that $\mathbf{b}^1(\varphi(u,v),\psi(u,v)) \equiv \mathbf{b}^2(u,v)$ and the boundaries of $\mathbf{b}^1$ and $\mathbf{b}^2$ coincide. Suppose that $\mathbf{b}_{00} \equiv \mathbf{p}_{00}$. Hence, $\mathbf{b}^1(s,0)$ coincides with $\mathbf{b}^2(u,0)$. Therefore $\mathbf{b}^1(\varphi(u,0),\psi(u,0)) \equiv \mathbf{b}^2(u,0)$ for $0 \le u \le 1$ and since both curves are irreducible it follows

$$\varphi(u,0) = u, \quad \psi(u,0) = 0. \tag{9}$$

Moreover, $\mathbf{b}^1(\varphi(0,v),\psi(0,v)) \equiv \mathbf{b}^2(0,v)$ for $0 \le v \le 1$, thus

$$\varphi(0,v) = 0, \quad \psi(0,v) = v. \tag{10}$$

We expand φ and ψ as Taylor series,

$$\varphi(u,v) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_{ij} u^i v^j, \quad \psi(u,v) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \beta_{ij} u^i v^j.$$

From (9) and (10) it follows $\varphi(u,v) = u + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_{ij} u^i v^j$ and $\psi(u,v) = v + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \beta_{ij} u^i v^j$. Therefore,

$$\mathbf{b}^1(\varphi(u,v),\psi(u,v)) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{b}_{ij} B_i^n\big(u + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_{ij} u^i v^j\big) B_j^m\big(v + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \beta_{ij} u^i v^j\big). \tag{11}$$

Suppose that $\varphi(u,v) \neq u$. Then there exists $\alpha_{ij} \neq 0$ for some i, j, $1 \leq i, j < \infty$. Hence

$$\deg\big(\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\alpha_{ij}u^i v^j\big) \geq 2 \;\Rightarrow\; \deg\big(B_i^n(u+\cdots)\big) \geq 2n.$$

We obtain that the degree of the polynomial in the right hand side in (11) is $\geq 2n+m$ which is not possible since $\mathbf{b}^1$ has degree $n+m$. Therefore $\varphi(u,v) = u$. Similarly we obtain $\psi(u,v) = v$ and hence $\mathbf{b}^1(u,v) \equiv \mathbf{b}^2(u,v)$.

The other possibilities for the boundaries are treated analogously. In the case where $\mathbf{b}_{00} = \mathbf{p}_{n0}$ we obtain $\varphi(u,v) = 1-u$, $\psi(u,v) = v$, and $\mathbf{b}^1(1-u,v) = \mathbf{b}^2(u,v)$. If $\mathbf{b}_{00} = \mathbf{p}_{0m}$ then $\mathbf{b}^1(u,1-v) = \mathbf{b}^2(u,v)$, and if $\mathbf{b}_{00} = \mathbf{p}_{nm}$ then $\mathbf{b}^1(1-u,1-v) = \mathbf{b}^2(u,v)$. In the case where $m = n$ there are four more possibilities:

$$\begin{aligned}
&\mathbf{b}_{00} = \mathbf{p}_{00},\ \mathbf{b}_{n0} = \mathbf{p}_{0m} \;\Rightarrow\; \mathbf{b}^1(v,u) = \mathbf{b}^2(u,v),\\
&\mathbf{b}_{00} = \mathbf{p}_{n0},\ \mathbf{b}_{n0} = \mathbf{p}_{nm} \;\Rightarrow\; \mathbf{b}^1(1-v,u) = \mathbf{b}^2(u,v),\\
&\mathbf{b}_{00} = \mathbf{p}_{0m},\ \mathbf{b}_{n0} = \mathbf{p}_{00} \;\Rightarrow\; \mathbf{b}^1(v,1-u) = \mathbf{b}^2(u,v),\\
&\mathbf{b}_{00} = \mathbf{p}_{nm},\ \mathbf{b}_{n0} = \mathbf{p}_{n0} \;\Rightarrow\; \mathbf{b}^1(1-v,1-u) = \mathbf{b}^2(u,v).
\end{aligned}$$

In all cases the control polygons of $\mathbf{b}^1$ and $\mathbf{b}^2$ coincide up to different enumeration of the control points.

$\Leftarrow$ Straightforward □

Next, we propose Algorithm 2 that compares for coincidence two irreducible tensor product Bézier surfaces of degree (n,m). In Example 2 tensor product Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ are compared for coincidence using Algorithm 2.

Example 2. The control points of the irreducible tensor product Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ of degree $(2,3)$ are shown in Table 2. The two surfaces and their control polygons are shown in Fig. 4**a.** and Fig. 4**b.**, respectively. In Fig. 5**a.** $\mathbf{b}^1$ and $\mathbf{b}^2$ are shown together. Their coincident part $\bar{\mathbf{b}}$ with its control polygon is shown in Fig. 5**b.**. This surface is obtained by subdivision of $\mathbf{b}^1$ in direction u at $a = 0$, $b = 1/2$ and direction v at $c = 1/6$, $d = 3/4$. The corresponding control points are shown in Table 2.

4 Conclusion

In this paper we present a new geometric algorithm based on subdivision that compares irreducible Bézier curves for coincidence and reports their coincident part if it is present. We generalize the algorithm to pairs of irreducible tensor product Bézier surfaces of degree (n,m), $m,n \in \mathbb{N}$. We believe that our approach can be successfully applied to the open problem for comparing pairs of irreducible tensor product Bézier surfaces of degree (n,m) and $(n+m,n+m)$, respectively. Another task for future research is to develop and implement an algorithm for comparing triangular Bézier surfaces for coincidence.

Algorithm 2 Comparison for Coincidence of two Irreducible Tensor Product Bézier Surfaces

Input: Irreducible Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ represented by their control points $\{\mathbf{b}_{ij}\}_{i=0,j=0}^{n,m}$, $\{\mathbf{p}_{ij}\}_{i=0,j=0}^{n,m}$, respectively

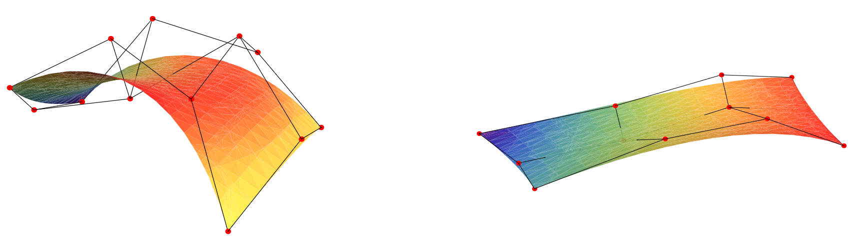
Output: (i) $\mathbf{b}^1$ and $\mathbf{b}^2$ are different;
(ii) $\mathbf{b}^1$ and $\mathbf{b}^2$ are disjoint;
(iii) $\mathbf{b}^1$ and $\mathbf{b}^2$ have coincident part $\bar{\mathbf{b}}$. Report the control points of $\bar{\mathbf{b}}$.

Step 1. Set $M_0 := [0,1]$.
for $j = 0,\dots,m$ apply Algorithm 1 to curves $\mathbf{c}_j^1$ and $\mathbf{c}_j^2$ with control points $\{\mathbf{b}_{ij}\}_{i=0}^{n}$ and $\{\mathbf{p}_{ij}\}_{i=0}^{n}$, respectively.
  if $\mathbf{c}_j^1$ and $\mathbf{c}_j^2$ are different
    then output (i) and **stop**;
      else find the corresponding coincidence interval I_j and set $M_{j+1} = M_j \cap I_j$
  end if
end for

Step 2. **if** $M_{m+1} := [a,b] \neq \emptyset$
  then for $j = 0,\dots,m$ subdivide $\mathbf{c}_j^1$ and $\mathbf{c}_j^2$ at a and b. Use the same notation for the new control points **end for**;
    else go to Step 3
end if

Step 3. Set $N_0 := [0,1]$.
for $i = 0,\dots,n$ apply Algorithm 1 to curves $\bar{\mathbf{c}}_i^1$ and $\bar{\mathbf{c}}_i^2$ with control points $\{\mathbf{b}_{ij}\}_{j=0}^{m}$ and $\{\mathbf{p}_{ij}\}_{j=0}^{m}$, respectively.
  if $\bar{\mathbf{c}}_i^1$ and $\bar{\mathbf{c}}_i^2$ are different
    then output (i) and **stop**;
      else find the corresponding coincidence interval J_i and set $N_{i+1} = N_i \cap J_i$
  end if
end for

Step 4. **if** $N_{n+1} := [c,d] \neq \emptyset$
  then for $i = 0,\dots,n$ subdivide $\bar{\mathbf{c}}_i^1$ at c and d
      end for
  output (iii) and **stop**;
    else output (ii) and **stop**
end if

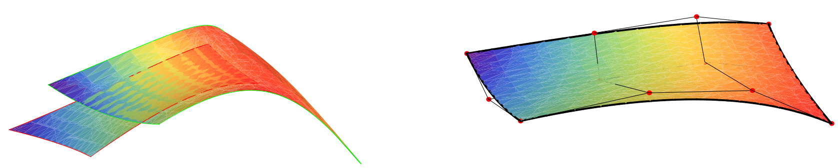


a. **b.**

Fig. 4: The irreducible tensor product Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ of degree $(2,3)$ with their control polygons: **a.** surface $\mathbf{b}^1$; **b.** surface $\mathbf{b}^2$. The corresponding control points are shown in Table 2.

Table 2: Comparison of the irreducible tensor product Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ for coincidence

$\mathbf{b}^1$	(0, 0, 0)	(0, 0.7, -0.5)	(0, 2, -.8)
	(1, 0, 1)	(1, 1, 0)	(1, 1, 1)
	(2, 0, .75)	(2, 1, 1.2)	(2, 2, 0.75)
	(3, -1, 0.15)	(3, 1, 0.4)	(3, 2, 0.2)
$\mathbf{b}^2$	(-1/2, 0.223122, -0.663304)	(-1/2, 0.685108, -0.814381)	(-1/2, 1.70446, -1.34961)
	(1/6, 0.253472, 0.0865162)	(1/6, 0.716898, -0.317969)	(1/6, 1.14913, -0.292144)
	(5/6, 0.322917, 0.517303)	(5/6, 0.799306, 0.259462)	(5/6, 1.1651, 0.332682)
	(3/2, 0.225694, 0.606424)	(3/2, 0.81875, 0.504948)	(3/2, 1.26719, 0.533203)
$\bar{\mathbf{b}}$	(0, 0.25, -0.161111)	(0, 0.716667, -0.433333)	(0, 1.3875, -0.6375)
	(1/2, 0.277778, 0.280556)	(1/2, 0.754167, -0.05)	(1/2, 1.1625, -0.00625)
	(1, 0.298611, 0.539583)	(1, 0.804167, 0.320833)	(1, 1.19062, 0.382812)
	(3/2, 0.225694, 0.606424)	(3/2, 0.81875, 0.504948)	(3/2, 1.26719, 0.533203)



a. **b.**

Fig. 5: **a.** The irreducible tensor product Bézier surfaces $\mathbf{b}^1$ and $\mathbf{b}^2$ from Example 2; **b.** The coincident part $\bar{\mathbf{b}}$ of $\mathbf{b}^1$ and $\mathbf{b}^2$ and its control polygon. The corresponding control points are shown in Table 2.

Acknowledgements This work was partially supported by the Bulgarian National Science Fund under Grant No. DFNI-T01/0001 and Sofia University Science Fund Grant No. 48/2016.

References

1. Barton, D.R., Zippel,R .: Polynomial decomposition. In: Proceedings of SYMSAC 76, pp. 356-358 (1976)
2. Barton, D.R., Zippel,R .: Polynomial decomposition algorithms. J. Symbolic Comput. **1**, 159-168 (1985)
3. Berry, T.G., Peterson, R.R.: The uniqueness of Bézier control points. Comput. Aided Geom. Des. **14**, 877-879 (1997)

4. Denker, W.A., Heron, G.J.: Generalizing rational degree elevation. Comput. Aided Geom. Des. **14**, 399-406 (1997)
5. von zur Gathen, J.: Functional decomposition of polynomials: the tame case. J. Symbolic Comput. **9**, 281-299 (1990)
6. Kozen, D, Landau, S.: Polynomial decomposition algorithms. J. Symbolic Comput. **7**, 445-456 (1989)
7. Pekerman, D., Seong, J.-K., Elber, G., Kim, M.-S.: Are two curves the same?. Comput.-Aided Des. Appl., 2(1-4), 85-94 (2005)
8. Sánchez-Reyes, J.: On the conditions for the coincidence of two cubic Bézier curves. J. of Comput. Appl. Math. **236**(6), 1675-1677 (2011)
9. Sánchez-Reyes, J.: The conditions for the coincidence or overlapping of two Bézier curves. Appl. Math. Comput. **248**, 625-630 (2014)
10. Wang, W.-K., Zhang, H., Liu, X.-M., Paul, J.-C.: Conditions for coincidence of two cubic Bézier curves. In: J. Comput. Appl. Math., **235**, 5198-5202 (2011)