Parametric Curves Variations with Respect to a Given Direction

Abdelouahad BAYAR
Ecole Supérieure de Technologie, Safi
Cadi Ayyad University
Rue Sidi Aissa, P.O.Box 89
46000, Safi, Morocco
bayar@ucam.ac.ma

Khalid SAMI
Faculty of Sciences, Marrakesh
Cadi Ayyad University
12 BD. My Abdellah, P.B 2390
40000, Marrakesh, Morocco
k_sami@ucam.ac.ma

ABSTRACT

Modeling dynamic characters, in observance of calligraphic rules can be done with the help of parametric curves. The class of Bézier curves is a powerful tool for modeling the outlines of the surface razed by a nib's calligrapher. Unfortunately, after capturing the character's hand-drawn outlines, the surface delimited by the outlines doesn't correspond to the space to shade when writing the character. Some of the curves pieces of the outlines are to be subdivided at some appropriate points. In order to identify these points, decompositions of the Bézier curves involved, with respect to particular directions, are necessary. This paper presents a simple and effective mathematical method for doing such decompositions.The proposed method helps to determine the points as extrema of certain functions. Then, the method is implemented and used to design some Arabic dynamical characters. This will help in building dynamic fonts respecting calligraphic strong rules.

Keywords

Bézier curves, decompositions with respect to a given direction, Arabic dynamical characters.

1. OVERVIEW

One of the big challenges in typesetting Arabic alphabet based texts consists on designing Arabic dynamical fonts respecting strong rules of a very well established calligraphy. The characters are context dependent. Some of them are to be stretched and inter-connected with small curves called *Kashida* [MEss 87].

Traditional software assisting to develop fonts, such as FontCreator [HiLo 05] and others don't provide enough support for developing Arabic fonts which can respect the calligraphic rules. In particular, stretching characters, with small curves, according to the context is not available. The general features and characteristics of a software to support these special needs are out of the scope of this paper. One of the preliminary problems to solve before designing applications for generating fonts can be stated as follows: after capturing hand written characters, or more precisely the nib's motion outlines, blacken the surface delimited by the Bézier curves [Bézi 77] representing the movement of the nib's vertices will not produce the original character. In order to reproduce the original characters, these Bézier curves have to be decomposed in some particular points. Then the nib's motion have to be considered according to the resulting curves separately. How to identify these points? This paper proposes a mathematical method to do so. finding out these points leads to the study of the *variations of parametric curves with respect to a given direction.* The method presented in this paper concerns parametric curves in general. Cubic Bézier curves are particular cases. Such curves are used in font development languages such as PostScript [Adob 99] and Metafont [Knut 86] .

After showing the need for a method for studying variations of parametric curves according to a given vector, the proposed mathematical method will be presented in detail. Then the way to decompose some Bézier curves will be presented and illustrated by an example.

2. PROBLEM

One of the ways to build a program that can help in developing Arabic fonts can be described as follows: After displaying a scanned image of the character, the font developer will seek the draw with it's mouse pointer as if it were a nib's head. Therefore, the program will generate the PostScript encoding corresponding to the character in the font. The program will have to take into account the curvilinear stretching, or *Kashida* [MEss 87], of the characters .

In the Arabic writing, the nib's head can be represented as a shaded rectangle whose width is one/a sixth of the length. The character is materialized by the surface razed by this rectangle. According to the writing style, Naskh, Roqaa, Diwany etc..., the angle between the rectangle's length and the baseline can vary a little or remain unchanged while writing. For instance, in the Naskh style this angle remains at about 70° with the baseline [MEss 87, MEss 94] (See Figure 1).

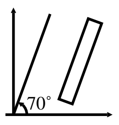

Figure 1. Nib's head in Naskh style, in 12mm size.

In the Arabic writing, many characters are dynamic, the shape and size of the character are context dependent. The stretching or change of size of the character in a given context is not a linear scaling nor a simple enlargement. Therefore, no suitable vectorial modeling can be found of a character through focusing on the outlines of the surface razed by the nib's head. The optimal solution consists on the curves modeling the nib's head *motion* instead of those representing the razed surface outlines. Then, the draw modeling the character is to be shaded taking into account the nib's head movement.

The rectangle modeling the nib's head can be considered as a rigid body (the length, width and angles between sides are invariant). The movement of one of the rectangle vertices determine entirely the movement of the other vertices through simple translations. The rectangle represented in Figure 2 has a length $M_{10}M_{20}$ and a width $M_{10}M_{40}$. It makes an angle α with the baseline.

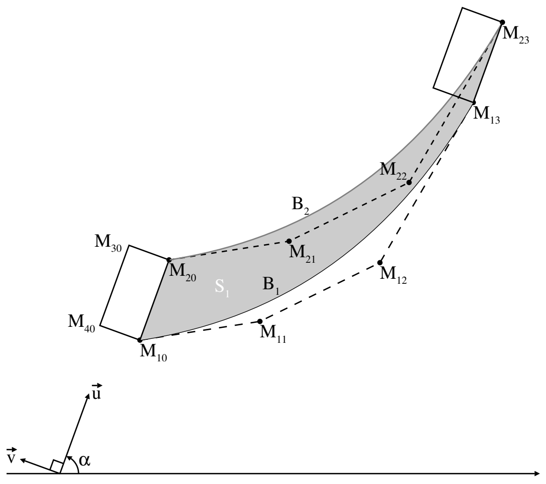

Figure 2. Surface S1 razed by the side $[M_{10}M_{20}]$

The movement of the vertex M_{10} is represented by the Bézier curve with four control points M_{10}, M_{11}, M_{12} and M_{13}. From now, we make the assumption that the angle α remains invariant while the rectangle is in motion (all movements are reduced to simple translations, no rotations are performed). Taking into account rotations will need supplementary efforts on approximation. Now, it is out of the scope of this paper. The three vertices will seek a similar movement through translations of vectors $\overrightarrow{u}$, $\overrightarrow{u} + \overrightarrow{v}$ and $\overrightarrow{v}$ respectively ($\overrightarrow{M_{10}M_{20}}$, $\overrightarrow{M_{10}M_{30}}$ and $\overrightarrow{M_{10}M_{40}}$). How to shade the surface razed by the rectangle? A way to do so consists on shading the surface razed by each of the rectangle's sides. So for the segment $[M_{10}, M_{20}]$, we will have to shade the surface delimited by the Bézier curve B_1 with control points M_{10}, M_{11}, M_{12} and M_{13}, the segment $[M_{13}, M_{23}]$, the Bézier curve B_2 with control points M_{23} , M_{22}, M_{21} and M_{20}, and finally the segment $[M_{20}, M_{10}]$. The points M_{20}, M_{21}, M_{22} and M_{23} are derived from M_{10}, M_{11}, M_{12} and M_{13} trough the translation of vector $\overrightarrow{u}$. Consider the movement of the segment $[M_{10}, M_{20}]$, then we define the Bézier outlines associated to this movement to be the *multi-curve* $(B_1, [M_{13}, M_{23}], B_2, [M_{20}, M_{10}])$. This way is meaningful. Actually, the PostScript language supports operators for shading outlines as with graphic development languages such as Java [Java 05].

This method of shading surfaces doesn't provide always the desired results. Actually, consider a very smart nib (a nib with a negligible thickness (width)) or simply a segment in motion (in general case, the surface razed by the nib's head is obtained through considering the surfaces razed by all the edges). As in Figure 3 where the segment $[M_{10}, M_{20}]$ is in movement, the extremity M_{10} follows the cubic Bézier's curve B. The surface razed by the segment is presented in Figure 4 and the surface delimited with the outlines associated to the movement doesn't fit exactly with the surface razed by the segment $[M_{10}, M_{20}]$ (See Figure 5).

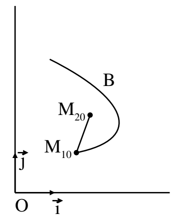

Figure 3. M_{10} path

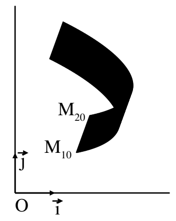

Figure 4. The Surface razed by the segment $[M_{10}, M_{20}]$

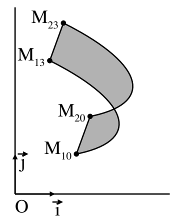

Figure 5. Surface delimited by the outlines associated to the movement

Now, let us decompose the curve B with the generalized algorithm of refinement [Bars 85, Jeff 81, Hosa 80, Gold 82] (case of none median decomposition) with respect to the coefficient $T = 0.3613981051$. We get two Bézier curves B_1 and B_2 that can represent the vertex's M_{10} movement (See Figure 6). Let S_1 and S_2 be the surfaces delimited by the outlines associated to the curves B_1 and B_2 (See Figure 7 and 8). Superposing S_1 and S_2 will give a shaded surface that fits exactly with the surface razed by the segment (See Figure 9).

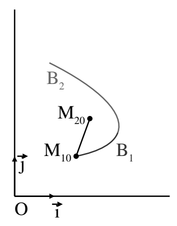

Figure 6. M_{10} path Decomposition

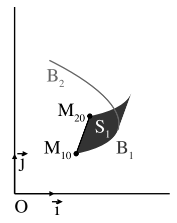

Figure 7. Area corresponding to B_1

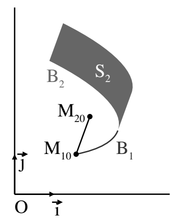

Figure 8. Area corresponding to B_2

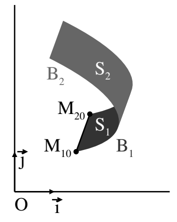

Figure 9. Superposing areas S_1 and S_2

The coefficient of decomposition T is an extrema of the Bézier parametric curve. Of course, there is no order relation in an affine plane. Next, we'll define comparison operators that will make the study of parametric curves as easy as the study of real functions with a single variable.

3. ORDER IN AN AFFINE PLANE IN $\mathbb{R}^3$

Consider $\mathcal{P}$ an affine plane in $\mathbb{R}^3$ containing the origin O. Let $\overrightarrow{u}$ be a given vector in $\overrightarrow{\mathcal{P}}$, the director vectorial plane associated to $\mathcal{P}$. We will define in $\mathcal{P}$ an *equality operator* $\underset{\overrightarrow{u}}{=}$ and an *order operator* $\underset{\overrightarrow{u}}{\leq}$ with respect to the vector $\overrightarrow{u}$.

In the sequel, we consider:

- The affine space $\mathbb{R}^3$ with the rectangular Cartesian system of reference $R\left(O, \overrightarrow{\imath}, \overrightarrow{\jmath}, \overrightarrow{k}\right)$.
- An affine plane $\mathcal{P}$ in $\mathbb{R}^3$ containing the origin O. Then, $\overrightarrow{\mathcal{P}}$ stands for the vectorial space of translations in $\mathcal{P}$. A direct orthonormal basis of $\overrightarrow{\mathcal{P}}$ will be denoted by $p = \left(\overrightarrow{P_1}, \overrightarrow{P_2}\right)$.
- Let $\overrightarrow{u} \in \overrightarrow{\mathcal{P}}$.

Definition 1 (Equality with respect to a vector in an affine plane)
We define an equality relation $\underset{\overrightarrow{u}}{=}$ *, among points in* $\mathcal{P}$*, with respect to the vector* $\overrightarrow{u}$ *by:*
$\forall M_1, M_2 \in \mathcal{P}$,
$M_1 \underset{\overrightarrow{u}}{=} M_2 \Leftrightarrow \left(\overrightarrow{u} \wedge \overrightarrow{OM_1}\right) = \left(\overrightarrow{u} \wedge \overrightarrow{OM_2}\right)$

Definition 2 (Direct orthogonal range with respect to a vector and a basis)
Let M *in* $\mathcal{P}$*, the value* $\left(\overrightarrow{u} \wedge \overrightarrow{OM}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right)$*, denoted by* $\mathcal{R}_{(\overrightarrow{u},p)}(M)$*, is called the direct orthogonal range of* M *with respect to a given vector* $\overrightarrow{u}$ *and a normed direct basis* p*.*

Remark 1
The direct orthogonal range is the component of the surface vector with respect to $\overrightarrow{P_1} \wedge \overrightarrow{P_2}$.

Property 1 *Let* $M_1, M_2 \in \mathcal{P}$*, then:*
$M_1 \underset{\overrightarrow{u}}{=} M_2 \Leftrightarrow \mathcal{R}_{(\overrightarrow{u},p)}(M_1) = \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$.
This property allows to establish a relation of equivalence in the set of points $\mathcal{P}$*. Classes of equivalence are identified as parallel lines in* $\mathcal{P}$ *with the same direction* $\overrightarrow{u}$*.*

Proof (Property 1)
The implication from left to right is obvious.
Conversely,
suppose that $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) = \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$
it means $\left(\overrightarrow{u} \wedge \overrightarrow{OM_1}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right) = \left(\overrightarrow{u} \wedge \overrightarrow{OM_2}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right)$
We have $\overrightarrow{u} \in \overrightarrow{P}$, $\overrightarrow{OM_1} \in \overrightarrow{P}$ and $\overrightarrow{OM_2} \in \overrightarrow{P}$.
Therefore $\left(\overrightarrow{u} \wedge \overrightarrow{OM_1}\right) = \lambda \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right)$ and $\left(\overrightarrow{u} \wedge \overrightarrow{OM_2}\right) = \beta \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right)$.
It follows that
$\left(\overrightarrow{u} \wedge \overrightarrow{OM_1}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right) = \lambda \cdot \left\|\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right\|^2$ and
$\left(\overrightarrow{u} \wedge \overrightarrow{OM_2}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right) = \beta \cdot \left\|\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right\|^2$
But $\left\|\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right\| \neq 0$.
Consequently $\lambda = \beta$.
We then get $\left(\overrightarrow{u} \wedge \overrightarrow{OM_1}\right) = \left(\overrightarrow{u} \wedge \overrightarrow{OM_2}\right)$
and finally $M_1 \underset{\overrightarrow{u}}{=} M_2$. ■

Definition 3 (Order with respect to a vector in an affine plane)
The relation $\underset{\overrightarrow{u}}{\leq}$ *defined in the set* $\mathcal{P}$*, with respect to the direction* $\overrightarrow{u}$*, by:*
$\forall M_1, M_2 \in \mathcal{P}$*, we have:*
$M_1 \underset{\overrightarrow{u}}{\leq} M_2 \Leftrightarrow \mathcal{R}_{(\overrightarrow{u},p)}(M_1) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$
is a total order.

Proof (Definition 3)

- Reflexivity: obvious
- Antisymmetry:
  Let $M_1, M_2 \in \mathcal{P}$ such that $M_1 \underset{\overrightarrow{u}}{\leq} M_2$ and $M_2 \underset{\overrightarrow{u}}{\leq} M_1$.
  then $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$ and $\mathcal{R}_{(\overrightarrow{u},p)}(M_2) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_1)$.
  So $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) = \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$
  according to the property 1, it follows
  $M_1 \underset{\overrightarrow{u}}{=} M_2$
- Transitivity:
  Let $M_1, M_2, M_3 \in \mathcal{P}$ such that $M_1 \underset{\overrightarrow{u}}{\leq} M_2$ and $M_2 \underset{\overrightarrow{u}}{\leq} M_3$.
  Then $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$ and $\mathcal{R}_{(\overrightarrow{u},p)}(M_2) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_3)$.
  It follows $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_3)$.
  So $M_1 \underset{\overrightarrow{u}}{\leq} M_3$
- Total order:
  Let $M_1, M_2 \in \mathcal{P}$, then we have $\mathcal{R}_{(\overrightarrow{u},p)}(M_1) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_2)$ or $\mathcal{R}_{(\overrightarrow{u},p)}(M_2) \leq \mathcal{R}_{(\overrightarrow{u},p)}(M_1)$
  and therefore, $M_1 \underset{\overrightarrow{u}}{\leq} M_2$ or $M_2 \underset{\overrightarrow{u}}{\leq} M_1$

Thus $\underset{\overrightarrow{u}}{\leq}$ is a total order relation. ■

4. FUNCTIONS FROM $\mathbb{R}$ INTO AN AFFINE PLANE IN $\mathbb{R}^3$

The equality and order relations defined before will allow studying parametric functions defined from an interval in $\mathbb{R}$ into an affine plane in the space $\mathbb{R}^3$ reported to a rectangular Cartesian system of reference $R\left(O, \overrightarrow{\imath}, \overrightarrow{\jmath}, \overrightarrow{k}\right)$ in terms of extrema and variations.

In the following, let $\mathcal{P}$ be an affine plane contained in $\mathbb{R}^3$, containing the origin O, with director vectorial plane associated $\overrightarrow{\mathcal{P}}$ and a normed direct basis $\left(\overrightarrow{P_1}, \overrightarrow{P_2}\right)$.

Consider a given vector $\overrightarrow{u} \in \overrightarrow{\mathcal{P}}$ and a closed interval $[a, b] \subset \mathbb{R}$.
Let f be a parametric function (parametric curve) such that:

$$\begin{array}{rcccl} f & : & [a,b] & \longrightarrow & \mathcal{P} \\ & & t & \longmapsto & (f_1(t), f_2(t)) \end{array}$$

where f_1 and f_2 are two real functions defined on $[a, b]$.

In the following, we will establish some lemmas and generalize theorems about real functions in order to take into account parametric functions with respect to a given vector. Before that, let us give an obvious property that will next be used.

Property 2 *Let f be a continuous function defined on the interval $[a, b]$. Suppose that f is differentiable on $]a, b[$. Then $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$ is continuous on $[a, b]$, differentiable on $]a, b[$ and $\left(\mathcal{R}_{(\overrightarrow{u},p)} \circ f\right)' = \mathcal{R}_{(\overrightarrow{u},p)} \circ f'$.*

Theorem 1 (Rolle's theorem) *If f is a continuous function on $[a, b]$ such that:*

- *$f(a) \underset{\overrightarrow{u}}{=} f(b)$ and*
- *f is differentiable on $]a, b[$*

then $\exists\, c \in\,]a, b[$ such that $f'(c) \underset{\overrightarrow{u}}{=} O$ (O is the point with null coordinates in $\mathcal{P}$).

<u>Proof (Theorem 1)</u>
The function $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$ is continuous on $[a, b]$, and differentiable on $]a, b[$ (from the property 2) and $\left(\mathcal{R}_{(\overrightarrow{u},p)} \circ f\right)(a) = \left(\mathcal{R}_{(\overrightarrow{u},p)} \circ f\right)(b)$.
from Rolle's theorem for real functions of one variable we get:
$\exists\, c \in\,]a, b[$ such that $\left(\mathcal{R}_{(\overrightarrow{u},p)} \circ f\right)'(c) = 0$.
Consequently
$\exists\, c \in\,]a, b[$ such that $\mathcal{R}_{(\overrightarrow{u},p)}(f'(c)) = 0$.
Thus, $\exists\, c \in\,]a, b[$ such that
$\left(\overrightarrow{u} \wedge \overrightarrow{Of'(c)}\right) \cdot \left(\overrightarrow{P_1} \wedge \overrightarrow{P_2}\right) = 0$.
It follows $\exists\, c \in\,]a, b[$ such that
$\left(\overrightarrow{u} \wedge \overrightarrow{Of'(c)}\right) = \overrightarrow{(0, 0, 0)}$,
because $\overrightarrow{u} \wedge \overrightarrow{Of'(c)}$ and $\overrightarrow{P_1} \wedge \overrightarrow{P_2}$ are collinear and $\overrightarrow{P_1} \wedge \overrightarrow{P_2}$ is not null.
Then, we will have
$\exists\, c \in\,]a, b[$ such that $\overrightarrow{u} \wedge \overrightarrow{Of'(c)} = \overrightarrow{u} \wedge \overrightarrow{OO}$.
finally $\exists\, c \in\,]a, b[$ such that $f'(c) \underset{\overrightarrow{u}}{=} O$. ■

Theorem 2 (Mean value theorem) *If f is a continuous function on $[a, b]$, differentiable on $]a, b[$, then $\exists\, c \in\,]a, b[$ such that $f'(c) \underset{\overrightarrow{u}}{=} \frac{f(b)-f(a)}{b-a}$.*

<u>Proof (Theorem 2)</u>
As in the previous proof (Theorem 1), consider the real function $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$.
By the Mean value theorem for real functions of one variable, we get:
$\exists\, c \in\,]a, b[$ such that $f'(c) \underset{\overrightarrow{u}}{=} \frac{f(b)-f(a)}{b-a}$. ■

Definition 4 (Monotony) *A function f is monotone increasing (respectively decreasing) on $[a, b]$ with respect to the direction $\overrightarrow{u}$ if and only if:*
$\forall t_1, t_2 \in [a, b]$, $t_1 \le t_2 \Rightarrow f(t_1) \underset{\overrightarrow{u}}{\le} f(t_2)$ (respectively $\forall t_1, t_2 \in [a, b]$, $t_1 \le t_2 \Rightarrow f(t_2) \underset{\overrightarrow{u}}{\le} f(t_1)$).

Remark 2

The monotony is *strict* whenever the order relation $\underset{\overrightarrow{u}}{<}$ is used.

Lemma 1 *The functions f and $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$ have the same monotony.*

Theorem 3 (variations sens) *If f is continuous on $[a, b]$ and differentiable on $]a, b[$ then:*

- *$\forall t \in\,]a, b[$, $f'(t) \underset{\overrightarrow{u}}{\ge} O \Leftrightarrow$ f is monotone increasing on $[a, b]$ with respect to the direction $\overrightarrow{u}$.*
- *$\forall t \in\,]a, b[$, $f'(t) \underset{\overrightarrow{u}}{\le} O \Leftrightarrow$ f is monotone decreasing on $[a, b]$ with respect to the direction $\overrightarrow{u}$.*

<u>Proof (Theorem 3)</u>
That's an obvious corollary of the lemma 1. ■

Definition 5 (Extrema) *The function f admits a maximum (resp a minimum) at the point $t_0 \in [a, b]$ with respect to the direction $\overrightarrow{u}$ if and only if there exist $t_1, t_2 \in [a, b]$, $t_1 \ne t_2$ such that:*
$t_0 \in\,]t_1, t_2[$ and $\forall t \in [t_1, t_2]$ $f(t) \underset{\overrightarrow{u}}{\le} f(t_0)$ (resp $f(t_0) \underset{\overrightarrow{u}}{\le} f(t)$).

Lemma 2 *The functions f and $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$ have the same extrema.*

Theorem 4 (Extrema and derivative)
Suppose that f is continuous on $[a,b]$ and differentiable on $]a,b[$. Let $t_0 \in\ ]a,b[$. Then if $\overrightarrow{u} \wedge \overrightarrow{Of'(t)}$ vanishes at t_0 and changes its direction, f admits an extrema in t_0 with respect to the direction $\overrightarrow{u}$.

Proof (Theorem 4)
As previously, consider the real function $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$.
The vector $\overrightarrow{u} \wedge \overrightarrow{Of'(t)}$ is collinear with $\overrightarrow{P_1} \wedge \overrightarrow{P_2}$, and $\overrightarrow{u} \wedge \overrightarrow{Of'(t)}$ is null at t_0 and changes in direction.
This will imply that $\mathcal{R}_{(\overrightarrow{u},p)} \circ f'$ is null at t_0 and changes in sign,
it follows that t_0 is an extrema of $\mathcal{R}_{(\overrightarrow{u},p)} \circ f$ on $[a,b]$
We will have from the lemma 2, that t_0 is an extrema of f on $[a,b]$. ■

Remark 3

At $t_0 \in\ ]a,b[$ which is an extrema according to the theorem 4, the directional tangent $\overrightarrow{Of'(t_0)}$ is collinear with the vector $\overrightarrow{u}$.
In the Arabic document processing area, there is no system that typeset Arabic documents with a quality that is like the one in the calligrapher handwritten texts. The simple cause is that the development of fonts respecting the Arabic calligraphy rules isn't a simple continuation of the works done concerning the latin documents. The problem is not simple as the majority had thought in the beginning. The first step and mandatory in this area is to offer to the community a good formalization of the problem. So the goal of the paper is not to *invent a new theory but to give an adequate and direct mathematical formalism.* The theory studied here would be extended in some coming works to offer mathematical ways to develop dynamic fonts of Arabic letters in stretchable outlines when the nib's head is translated without and with rotations.

5. BÉZIER CURVES DECOMPOSITION

Up till now, a way to decompose parametric curves characterizing a segment motion has been developed. So, the surface razed by the segment can be obtained. It is the same for any rigid body. Now, we will focus on the polar Bézier curves. In the following, we'll give a method for decomposing Bézier curves of any degree. An example with a cubic Bézier curve will be given as illustration.

- Let $\mathbb{R}^3$ be the affine space with the rectangular Cartesian system of reference $R\left(O, \overrightarrow{\imath}, \overrightarrow{\jmath}, \overrightarrow{k}\right)$.
- Let $\mathcal{P}$ denote the $\mathbb{R}^2$ affine plane with direct normed basis $p = (\overrightarrow{\imath}, \overrightarrow{\jmath})$ passing through the origin O.
- Let $[M_{10}, M_{20}]$ be a segment to be translated in $\mathcal{P}$.
- $[a,b] = [0,1]$.
- The Bézier curve B describes the movement of M_{10}. Suppose that their control points M_{10}, M_{11}, $\ldots, M_{1n}$ are such that $M_{1i} = (x_i, y_i, 0)$.

The curve B can be defined in $\mathbb{R}^3$ through considering a null component with respect to the vector $\overrightarrow{k}$:

$$\begin{array}{rcccl} B & : & [0,1] & \longrightarrow & \mathcal{P} \\ & & t & \longmapsto & \displaystyle\sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} M_{1i} \end{array}$$

In order to decompose a Bézier's curve into monotone curves, with respect to the direction $\overrightarrow{M_{10}M_{20}}$, we should determine the set of extrema of the direct orthogonal range $\mathcal{R}_{(\overrightarrow{M_{10}M_{20}},p)} \circ B$ on $[0,1]$. Then, these extrema will be classified by increasing order. Let $T_0 = \{t_{01}, t_{02}, \ldots, t_{0m}\}$ be this ordered set, with $m \leq n-1$. Thus, we decompose B with respect to t_{01}, we get two Bézier curves B_0 and B_1 such that B_0 has a constant monotony with respect to $\overrightarrow{M_{10}M_{20}}$ on $[0,1]$ and B_1 is a curve to decompose. We don't need to determine the extrema of B_1 with respect to $\overrightarrow{M_{10}M_{20}}$ because the set of these extrema is $T_1 = \left\{t_{11}, t_{12}, \ldots, t_{1(m-1)}\right\}$ where $t_{1i} = \frac{t_{0(i+1)} - t_{01}}{1 - t_{01}}$ (the formula can be established by a an easy calculation). We continue the process until obtaining the last monotone curve. This method is presented as an automatic algorithm in the following.
Suppose that we have already defined the following data structures and algorithms:
`Point` a data structure modeling a point in $\mathbb{R}^3$,
`Vector` a data structure modeling a vector in $\overrightarrow{\mathbb{R}}^3$, (the Point data structure can be used instead of this one),
`Bezier` a data structure modeling a Bézier's curve,
`PtVect(M10:Point,M20:Point):Vector` a function that results in the vector defined by two points.

ExtremaNbr(B:Bezier,U:Vector):INTEGER a function that results in the number of extrema of B with respect to U,

Extremas(B:Bezier,U:Vector): array[1..NMax]of REAL a function that results in an array of extrema of B with respect to U,

LDecp(B:Bezier,t:REAL):Bezier a function that results in the left Bézier curve by decomposing B with respect to t,

RDecp(B:Bezier,t:REAL):Bezier a function that results in the right Bézier curve by decomposing B with respect to t.

Now, we will give the meaning of some variables and algorithms.

M10, M20 are the points which determine the segment,

U is the decomposition direction,

B is the curve to decompose,

Nx is the B extrema number according to U,

AExt is the array displaying all the extrema of B according to U,

DBC is the array displaying the Bézier curves obtained by decomposition of B according to U,

Ctrd and *Ctre*: are indexes for accessing DBC and AExt arrays

```
 ⋮
M10,M20 :  Point;
U : Vector;
B : Bezier;
Nx :  INTEGER;
AExt :  array[1..Nmax] fo REAL;
DBC : array[1..Nmax+1] of Bezier;
Ctrd, Ctre :  INTEGER;
BEGIN
 /* B, M10, M20 initialising
 ⋮
 U ← PointVector(M10,M20);
 Nx ← ExtremaNumber(B,U);
 AExt ← Extremas(B,U);
 FOR Ctrd=1, Nx, +1
 BEGIN
  DBC[Ctrd]←LDecp(B,AExt[Ctrd]);
  B←RDecp(B,AExt[Ctrd]);
  FOR Ctre=Ctrd+1, Nx, +1
  AExt[Ctre]←
   (AExt[Ctre]-AExt[Ctrd])/(1-AExt[Ctrd]);
 END
 DBC[Nx+1]←B;
END
```

We can remark that there is no need for two separate functions LDecp and RDecp since both parts of the subdivided curve can be computed by one execution of the algorithm of de Casteljau. We use two functions for better clarity and also when we use the algorithm of de Casteljau we must extract the decomposition parts separately.

The Method will be illustrated through an example modeling the movement of a segment $[M_{10}, M_{20}]$ with a cubic Bézier curve B with two extrema with respect to the vector $\overrightarrow{M_{10}M_{20}}$ (See Figure 10).

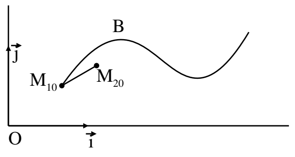

Figure 10. A cubic Bézier curve of a movement

The surface razed by the segment $[M_{10}, M_{20}]$ (See Figure 11) is different from the surface delimited by the outlines associated to the movement (See Figure 12). The reason behind this is that B is not monotone with respect to $\overrightarrow{M_{10}M_{20}}$.

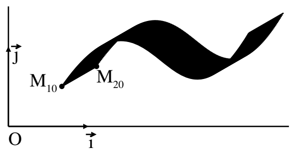

Figure 11. Surface razed by the segment $[M_{10}, M_{20}]$

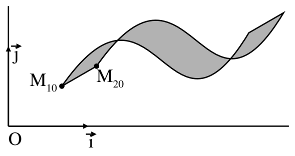

Figure 12. Surface delimited by the associated outlines

We will have to decompose the curve B into monotone sub-curves with respect to the direction $\overrightarrow{M_{10}M_{20}}$. The B control points are $M_{10} = (95, 50)$, $M_{11} = (130, 100)$, $M_{12} = (135, 20)$ and $M_{13} = (165, 70)$. We determine the direct orthogonal range of B. In order to find the extrema on $]0, 1[$, we derive this range. Then, we find that the direct orthogonal range has two extrema $T_{01} = 0.1571942250$ and $T_{02} = 0.8301512030$. We decompose B with respect to T_{01} to obtain two curves; B_1 monotone, and a curve B_2 to decompose again (See Figure 13).

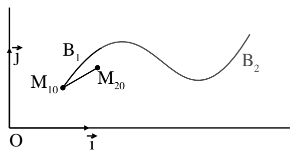

Figure 13. First decomposition

Then, the curve B_2 is decomposed with respect to $T_{11} = \frac{T_{02}-T_{01}}{1-T_{01}}$, say for instance, $T_{11} = 0.7984721960$. The decomposition gives two curves B_{21} and B_{22}. Both of them are monotone with respect to $\overrightarrow{M_{10}M_{20}}$ (See Figure 14).

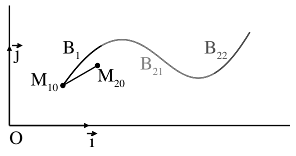

Figure 14. Second decomposition

If we superpose all the surfaces delimited by the outlines associated to the three curves B_1, B_{21} and B_{22} (See Figure 15), we get exactly the surface in Figure 11.

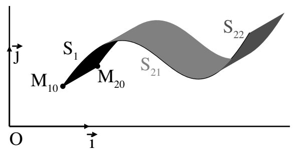

Figure 15. Superposition of the surfaces of decomposition

6. CONCLUSIONS

A method for the determination of the extrema of a parametric curve is now ready for use. This method has been used to find the decomposition's coefficients of cubic parametric curves. This allows generating the PostScript code which supports the movement of an Arabic nib that remains with a constant angle of inclination with the baseline. The case where this angle is changing will be presented in a next work. The study of both the Bézier curves variations and angle's variations will lead to use also approximations with polar curves.

7. REFERENCES

[Adob 99] Adobe. PostScript Language reference. Third edition, Adobe Systems Incorporated, Library of Congress Cataloging-in-Publication Data, 1999.

[Bars 85] Brian A. Barsky, Arbitrary Subdivision of Bézier Curves, Technical Report UCB.CSD 85/265, Computer Science Division, University of California, Berkeley, California 94720, November, 1985.

[Bézi 77] Pierre E. Bézier, Essai de Définition Numérique des Courbes et des Surfaces Expérimentales, PhD dissertation, Université Pierre et Marie Curie, Paris, 1977.

[HiLo 05] High-Logic, Font Creator Manual, http://www.high-logic.com, 2005 .

[Java 05] Sun, Java 2D Programmer's Guide, http://java.sun.com/products/jdk/1.2/docs/guide/2d/spec/j2d-title.fm.HTML,2005.

[Knut 86] D.E. Knuth, The Metafont Book, Computers and Typesetting, Vol. C. Reading MA: Addison-Wesley. 1986.

[MEss 87] Mahdi Essayed Mahmoud, Learning Arabic Calligraphy: Naskh, Roqaa, Farsi, Tholoth, Diwany, Ibn Sina, Publisher, Cairo, Egypt, 1987.

[MEss 94] Mahdi Essayed Mahmoud, Learn your self Arabic Calligraphy: Naskh, Roqaa, Farsi, Tholoth, Diwany, Ibn Sinaa publisher Cairo, Egypt, 1994.

[Jeff 81] Jeffrey M. Lane, Richard F. Riesenfelf, Bounds on Polynomial, BIT, Vol 21, No 1, pp. 112-117, 1981.

[Hosa 80] Mamoru Hosaka, Fumihiko Kimura, A Theory and Methods for Free Form Shape Construction, Journal of Information Processing, Vol 3, No 3, pp. 140-151, 1980.

[Gold 82] Ronald N. Goldman, Using Degenerate Bézier Triangles and Tetrahedra to Subdivide Bézier Curves, Computer-Aided Design, Vol 14, No 6, pp. 307-311, November 1982.