Length Estimation of Rational Bézier Curves and Application to CAD Parametrization

Maharavo Randrianarivony

Abstract**—**We want to estimate the chord length Λ of a given rational Bézier curve efficiently. Since rational Bézier are nonlinear function, it is generally impossible to evaluate its length exactly. We approximate the length by using subdivision and we investigate the accuracy of the approximation Λ_n. In order to improve the efficiency, we use adaptivity with some length estimator. Additionally, we will give a rigorous theoretical analysis of the rate of convergence of Λ_n to Λ. We analyze also the required number of subdivisions in order to attain a prescribed accuracy. At the end, we briefly describe an application in CAD surface parametrization.

Keywords**—**Rational Bézier, Length, Parametrization, Adaptivity.

I. Introduction

In geometric modeling, rational Bézier curves are important CAGD entities because they can represent both the free-form setting and the algebraic one. Thus, they can exactly describe circular arcs and most interesting conic sections. On the other hand, free-form Bézier curves are special case of them. Our main contribution in this paper is as follows:

- Algorithm for length estimation of such curves,
- Theoretical investigation using subdivisions and bounds,
- Exponential convergence speed $\mathcal{O}(2^{-n})$,
- Practical computer implementation of the theory.

Related works are as follows. Roulier has proposed a length estimation algorithm but only for Bézier curves [11]. Walter *et al.* did not really evaluate lengths but they have approximated the arc length parametrization which is a very closely related task. A similar approach was proposed by Floater who used cubic spline for the approximation [5]. Subdivision technique was used by Hain who proposed some approach to stop the subdivision recursion [6]. In this paper, we use also subdivision but for the rational case. The structure of this paper is as follows. We will start by formulating the problem more accurately in the next section. The main result of this paper is found in section III where we introduce the approximation method and we analyze the error. We will see in section IV a possible improvment of the method by using adaptivity. Section V will be devoted to a brief application in CAD parametrizations. Finally, we show some numerical results at the end of the paper.

II. Problem formulation

Our objective is to design an algorithm for estimating the length of a curve $\mathbf{x}$ inside an interval $[a, b]$. That is, we want to evaluate

$$\Lambda := \int_a^b \|\mathbf{x}'(t)\| dt. \tag{1}$$

Faculty for Mathematics, University of Bonn, Germany, email: maharavo@informatik.tu-chemnitz.de

Without loss of generality, we suppose that the curve is defined on $[0, 1]$ and we compute the whole length i.e. $a = 0$, $b = 1$. The general case where $[a, b] \neq [0, 1]$ can be treated in a very similar way.

We suppose that the curve is a rational Bézier curve

$$\mathbf{x}(t) := \frac{\sum_{i=0}^{m} \omega_i \mathbf{b}_i B_i^m(t)}{\sum_{i=0}^{m} \omega_i B_i^m(t)}, \tag{2}$$

where B_i^m denotes the Bernstein polynomial [3], [2] and $\mathbf{b}_i = [b_{i,1}, b_{i,2}, b_{i,3}] \in \mathbf{R}^3$ are the control points. Additionally, we assume that the weights ω_i are uniformly bounded. That is, there exist two positive constants R_1, R_2 such that

$$R_1 < \left| \sum_{i=0}^{m} \omega_i B_i^m(t) \right| < R_2 \qquad \forall\, t \in [0, 1]. \tag{3}$$

Let us denote by $\tilde{\mathbf{x}}(t) = [\tilde{x}_1(t), \tilde{x}_2(t), \tilde{x}_3(t)]$ and $\omega(t)$ the numerator and the denominator of the above formula where $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]$. The numerator $\tilde{\mathbf{x}}$ is a Bézier curve where its control points are given by $\tilde{\mathbf{b}}_i := \omega_i \mathbf{b}_i$.

Since the formula in (2) contains rational quotient and the one in (1) has square root and derivatives such as

$$\Lambda = \int_0^1 \sqrt{x_1'(t)^2 + x_2'(t)^2 + x_3'(t)^2}\, dt, \tag{4}$$

it is very difficult to compute the integral exactly. In fact, the integrand is given by

$$\frac{1}{\omega(t)^2} \sqrt{\sum_{j=1}^{3} \left[\tilde{x}_j'(t)\omega(t) - \omega'(t)\tilde{x}_j(t)\right]^2}. \tag{5}$$

For the same reason, traditional methods using polynomial approximation of the integrand would require too high polynomial degree. Hence, we will use geometric methods for the approximation.

III. Approximation and its accuracy

In this section, we are going to approximate the exact length Λ by a sequence Λ_n and we analyze the error $|\Lambda - \Lambda_n|$. Before going into the technical details, let us give a short motivation about our approach. We compute Λ_n by finding a lower bound $\mathcal{L}_n$ and an upper bound $\mathcal{U}_n$ such that

$$\mathcal{L}_n \leq \Lambda \leq \mathcal{U}_n. \tag{6}$$

If those bounds have the property that their difference $|\mathcal{L}_n - \mathcal{U}_n|$ converges to zero, then a good choice for the approximation is the average $\Lambda_n := 0.5(\mathcal{L}_n + \mathcal{U}_n)$ or any other convex combination:

$$\Lambda_n := \alpha\mathcal{L}_n + (1-\alpha)\mathcal{U}_n \qquad \text{for} \quad 0 < \alpha < 1. \tag{7}$$

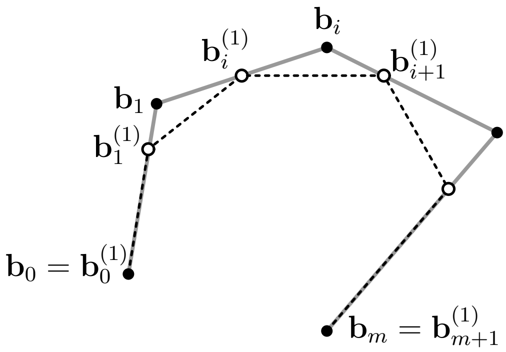

Fig. 1. Degree elevation of rational Bézier where $L[\mathbf{b}_i^{(1)}, \mathbf{b}_i, \mathbf{b}_{i+1}^{(1)}] \leq L[\mathbf{b}_i^{(1)}, \mathbf{b}_{i+1}^{(1)}]$.

Thus, the speed of convergence of Λ_n to Λ depends on that of the difference $|\mathcal{L}_n - \mathcal{U}_n|$ of the bounds. The main advantage of our method is that the bounds $\mathcal{L}_n$ and $\mathcal{U}_n$ can be computed algorithmically and the rate of convergence is exponential.

We will not need to use quadrature rules to estimate the integral in (1) because the structure of the function $\mathbf{x}$ is known [11], [13]. Our preferred method is to apply subdivision recursively while using some flatness criterion [6], [4] in order to know if the curve is close to be linear.

A. Preliminary results

Before we state the main theorem, let us look at the following simple lemma. At first glance the lemma seems evident because of the famous convex hull property [3]. But a closer look reveals that the convex hull property alone cannot justify the claim, especially if we take the weights into consideration. We prove the lemma by using rational degree elevation. Note that the degree elevation is not used in practice for that it is exclusively for proving purpose. Before going any further, note that the following bound is not yet the upper bound $\mathcal{U}_n$ which we are searching for.

Lemma

For any rational Bézier curve of the form (2), the length is smaller than

$$\sum_{i=0}^{m-1} \|\mathbf{b}_i - \mathbf{b}_{i+1}\|. \tag{8}$$

Proof

For a finite sequence of 3D points $\mathbf{P} = \{\mathbf{p}_i\}_{i=0}^{n}$, we denote

$$L[\mathbf{P}] = L[\mathbf{p}_0, ..., \mathbf{p}_n] := \sum_{i=0}^{n-1} \|\mathbf{p}_i - \mathbf{p}_{i+1}\|. \tag{9}$$

The degree elevated of a rational Bézier curve $\mathbf{x}$ is given by

$$\mathbf{x}(t) = \frac{\sum_{i=0}^{m+1} \omega_i^{(1)} \mathbf{b}_i^{(1)} B_i^m(t)}{\sum_{i=0}^{m} \omega_i^{(1)} B_i^m(t)}, \tag{10}$$

where the new weights are

$$\omega_i^{(1)} := c_{i,m}\omega_{i-1} + (1 - c_{i,m})\omega_i \quad \text{with} \quad c_{i,m} := i/(m+1),$$

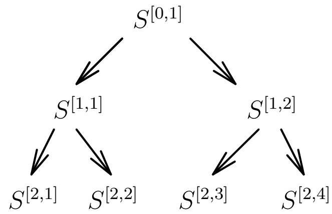

Fig. 2. Recursive subdivisions

and the new control points are

$$\mathbf{b}_i^{(1)} := \frac{c_{i,m}\omega_{i-1}\mathbf{b}_{i-1} + (1 - c_{i,m})\omega_i \mathbf{b}_i}{c_{i,m}\omega_{i-1} + (1 - c_{i,m})\omega_i}. \tag{11}$$

Thus, $\mathbf{b}_i^{(1)}$ is a convex combination of $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ because the weights are positive. Therefore, we have (see Fig. 1)

$$L[\mathbf{b}_i^{(1)}, \mathbf{b}_i, \mathbf{b}_{i+1}^{(1)}] \leq L[\mathbf{b}_i^{(1)}, \mathbf{b}_{i+1}^{(1)}]. \tag{12}$$

Denote by $\mathbf{B}^{(0)}$ the initial control polygon and by $\mathbf{B}^{(p)}$ ($p \geq 1$) the next control polygons after repeated degree elevations. Since $\mathbf{b}_i^{(1)}$ is a convex combination of $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$, we have

$$L[\mathbf{B}^{(0)}] = L[\mathbf{b}_0, \mathbf{b}_1^{(1)}, \mathbf{b}_1, \mathbf{b}_2^{(1)}, \mathbf{b}_2, ..., \mathbf{b}_{m-1}, \mathbf{b}_m^{(1)}, \mathbf{b}_m].$$

As a consequence to (12), we have $L[\mathbf{B}^{(1)}] \leq L[\mathbf{B}^{(0)}]$. By doing that repeatedly, we have

$$L[\mathbf{B}^{(p)}] \leq L[\mathbf{B}^{(p-1)}] \leq \cdots \leq L[\mathbf{B}^{(1)}] \leq L[\mathbf{B}^{(0)}]. \tag{13}$$

Since it is well known [3] that the control polygon of the curve tends to the curve itself, we have $\Lambda = L[\mathbf{B}^{(\infty)}] \leq L[\mathbf{B}^{(0)}]$.

Q.E.D.

B. Rational Bézier subdivision

Let us first recall some notions related to the successive subdivision [7] of an arbitrary Bézier function

$$C(t) = \sum_{i=0}^{m} \mathbf{s}_i B_i^m(t). \tag{14}$$

Let $\mathbf{s}_i^{(j)}$ be the points which are found by using the de Casteljau [3] algorithm at $t = 0.5$, i.e. $\mathbf{s}_i^{(j+1)} := 0.5(\mathbf{s}_i^{(j)} + \mathbf{s}_{i+1}^{(j)})$ and $\mathbf{s}_i^{(0)} := \mathbf{s}_i$. The function $C^{[0,1]} := S$ can be split into two Bézier functions $C^{[1,1]}$ and $C^{[1,2]}$ (see Fig. 2) which have respectively the control points $\mathbf{s}_i^{[1,1]} := \mathbf{s}_0^{(i)}$ and $\mathbf{s}_i^{[1,2]} := \mathbf{s}_i^{(m-i)}$ and such that

$$C^{[0,1]}(t) = \begin{cases} C^{[1,1]}(t) & \forall\, t \in [0, 0.5], \\ C^{[1,2]}(t) & \forall\, t \in [0.5, 1]. \end{cases} \tag{15}$$

We can apply that process successively in order to obtain from each Bézier function $C^{[p-1,i]}$ two Bézier functions $C^{[p,2i-1]}$ and $C^{[p,2i]}$. That is, after applying subdivisions n times we have the curves $C^{[n,1]}, C^{[n,2]}, ..., C^{[n,2^n]}$ as explained in Fig. 2. Each function $C^{[n,k]}$ coincides with C on the interval $[p_{k-1}, p_k]$ where $p_k := k/2^n$ and its control points are denoted by $\mathbf{s}_i^{[n,k]}$ for $k = 1, ..., 2^n$ and $i = 0, ..., m$

Now, we want to apply the above subdivision technique to the numerator and denominator. The functions $\tilde{\mathbf{x}}(\cdot)$ and $\omega(\cdot)$

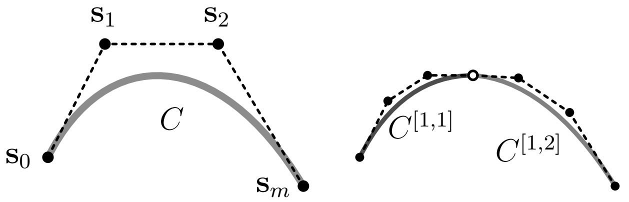

Fig. 3. Subdivision of a Bézier curve.

will be subdivided into functions $\tilde{\mathbf{x}}^{[n,k]}$ and $\omega^{[n,k]}$ having the control points $\tilde{\mathbf{b}}_i^{[n,k]}$ and $\omega_i^{[n,k]}$. On each subinterval $[p_k, p_{k+1}]$ we introduce the rational Bézier $\mathbf{x}^{[n,k]} := \tilde{\mathbf{x}}^{[n,k]}/\omega^{[n,k]}$. Thus, by defining $\mathbf{b}_i^{[n,k]} := \tilde{\mathbf{b}}_i^{[n,k]}/\omega_i^{[n,k]}$, we have for all $\tau \in [p_k, p_{k+1}]$:

$$\mathbf{x}^{[n,k]}(\tau) = \frac{\sum_{i=0}^m \omega_i^{[n,k]} \mathbf{b}_i^{[n,k]} B_i^m(s)}{\sum_{i=0}^m \omega_i^{[n,k]} B_i^m(s)} \quad \text{where} \quad s = \frac{\tau - p_k}{p_{k+1} - p_k}. \tag{16}$$

Furthermore, we have the restriction property:

$$\tilde{\mathbf{x}}^{[n,k]} = \tilde{\mathbf{x}}_{|[p_{k-1},p_k]}, \quad \text{and} \quad \omega^{[n,k]} = \omega_{|[p_{k-1},p_k]} \tag{17}$$

By considering the interval $[p_{k-1}, p_k]$, we can introduce for $i = 0, ..., m$

$$\theta_{i,k} := (i/m)p_k + (1 - i/m)p_{k-1} \quad \text{with} \quad p_k = k/2^n. \tag{18}$$

Theorem

Suppose that the rational Bézier in (2) has been subdivided n times. Then, we have the following accuracy order for all $k = 1, ..., 2^n$ and $i = 0, ..., m$:

$$\|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{b}_i^{[n,k]}\| = \mathcal{O}(2^{-2n}). \tag{19}$$

Proof

Due to boundedness (3) and restriction property (17), we have $|\omega^{[n,k]}(t)| = |\omega(t)| < R_2$. Hence, there exists K_1 such that

$$\left\|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \frac{\omega_i^{[n,k]}\mathbf{b}_i^{[n,k]}}{\omega^{[n,k]}(\theta_{i,k})}\right\| = \left\|\frac{\tilde{\mathbf{x}}^{[n,k]}(\theta_{i,k}) - \omega_i^{[n,k]}\mathbf{b}_i^{[n,k]}}{\omega^{[n,k]}(\theta_{i,k})}\right\|$$

$$\le K_1 \left\|\tilde{\mathbf{x}}^{[n,k]}(\theta_{i,k}) - \omega_i^{[n,k]}\mathbf{b}_i^{[n,k]}\right\|, \tag{20}$$

Similarly,

$$\left\|\mathbf{b}_i^{[n,k]} - \frac{\omega_i^{[n,k]}\mathbf{b}_i^{[n,k]}}{\omega^{[n,k]}(\theta_{i,k})}\right\| \le K_2 \left|\omega^{[n,k]}(\theta_{i,k}) - \omega_i^{[n,k]}\right|. \tag{21}$$

As a consequence, we obtain

$$\|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{b}_i^{[n,k]}\| \le K_1 \left\|\tilde{\mathbf{x}}^{[n,k]}(\theta_{i,k}) - \omega_i^{[n,k]}\mathbf{b}_i^{[n,k]}\right\| + K_2 \left|\omega^{[n,k]}(\theta_{i,k}) - \omega_i^{[n,k]}\right|. \tag{22}$$

On the other hand, let us consider the blossom function $\mathcal{P}$ of the polynomial $\tilde{\mathbf{x}}^{[n,k]}$. We have the relation with the control points [12]:

$$\tilde{\mathbf{b}}_i^{[n,k]} = \mathcal{P}(\underbrace{p_{k-1}, ..., p_{k-1}}_{m-i}, \underbrace{p_k, ..., p_k}_{i}). \tag{23}$$

Thus, we have the following Taylor development:

$$\begin{aligned}\tilde{\mathbf{b}}_i^{[n,k]} \;=\; & \mathcal{P}(\theta_{i,k}, ..., \theta_{i,k}) + \\ & \sum_{p=1}^{m-i}(p_{k-1} - \theta_{i,k})\frac{\partial}{\partial x_p}\mathcal{P}(\theta_{i,k}, ..., \theta_{i,k}) + \\ & \sum_{p=m-i+1}^{m}(p_k - \theta_{i,k})\frac{\partial}{\partial x_p}\mathcal{P}(\theta_{i,k}, ..., \theta_{i,k}) + \\ & \mathcal{O}(|p_k - p_{k-1}|^2).\end{aligned}$$

Since $\mathcal{P}$ is symmetric, all partial derivatives in the above relation are the same. Due to the fact that $(m-i)(p_k - \theta_{i,k}) + i(p_{k-1} - \theta_{i,k}) = 0$, we obtain $\tilde{\mathbf{b}}_i^{[n,k]} = \mathcal{P}(\theta_{i,k}, ..., \theta_{i,k}) + \mathcal{O}(|p_k - p_{k-1}|^2)$. As a consequence, we deduce $\tilde{\mathbf{b}}_i^{[n,k]} = \tilde{\mathbf{x}}^{[n,k]}(\theta_{i,k}) + \mathcal{O}(2^{-2n})$. The same analysis can be repeated to the blossom of the polynomial ω in order to obtain $\omega_i^{[n,k]} = \omega(\theta_{i,k}) + \mathcal{O}(2^{-2n})$. Therefore, we can deduce from (22) that $\|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{b}_i^{[n,k]}\| = \mathcal{O}(2^{-2n})$. **Q.E.D.**

C. Upper and Lower Bounds

At the n-th subdivision, the true length Λ is the sum of the lengths $\lambda(k,n)$ of the subcurves $\mathbf{x}^{[n,k]}$ such as

$$\Lambda = \sum_k \lambda(k,n). \tag{24}$$

We can now use that approximation result in order to deduce the accuracy in length computation.

Theorem

Define for all $k = 0, ..., 2^n - 1$

$$l(k,n) := \sum_{i=0}^{m-1} \|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{x}^{[n,k]}(\theta_{i+1,k})\|, \tag{25}$$

$$u(k,n) := \sum_{i=0}^{m-1} \|\mathbf{b}_i^{[n,k]} - \mathbf{b}_{i+1}^{[n,k]}\|. \tag{26}$$

We claim that for any $\alpha \in]0,1[$, the sequence $\Lambda_n := \sum_k(\alpha l(k,n) + (1-\alpha)u(k,n))$ converges to the exact length Λ in dyadic order:

$$|\Lambda - \Lambda_n| = \mathcal{O}(2^{-n}). \tag{27}$$

Proof

Consider the length $\lambda(k,n)$ of the curve $\mathbf{x}^{[n,k]}$. We have

$$l(k,n) \le \lambda(k,n) \le u(k,n), \tag{28}$$

where the second inequality is due to the preceding Lemma and the first one is obvious. On the other hand, the difference $D(k,n) := |u(k,n) - l(k,n)|$ of those bounds can be estimated as follows

$$\begin{aligned} D(k,n) &= \sum_{i=0}^{m-1} \; \|\mathbf{b}_i^{[n,k]} - \mathbf{b}_{i+1}^{[n,k]}\| - \\ &\qquad \|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{x}^{[n,k]}(\theta_{i+1,k})\| \\ &\le \sum_{i=0}^{m-1} \; \|[\mathbf{b}_i^{[n,k]} - \mathbf{x}^{[n,k]}(\theta_{i,k})] - \\ &\qquad [\mathbf{b}_{i+1}^{[n,k]} - \mathbf{x}^{[n,k]}(\theta_{i+1,k})] + \\ &\qquad [\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{x}^{[n,k]}(\theta_{i+1,k})]\| - \\ &\qquad \|\mathbf{x}^{[n,k]}(\theta_{i,k}) - \mathbf{x}^{[n,k]}(\theta_{i+1,k})\| \\ &\le \sum_{i=0}^{m-1} \; \|[\mathbf{b}_i^{[n,k]} - \mathbf{x}^{[n,k]}(\theta_{i,k})]\| + \\ &\qquad \|[\mathbf{b}_{i+1}^{[n,k]} - \mathbf{x}^{[n,k]}(\theta_{i+1,k})]\|. \end{aligned}$$

By using the previous theorem with the last inequality, we deduce

$$D(k,n) = |u(k,n) - l(k,n)| = \mathcal{O}(2^{-2n}). \qquad (29)$$

As a consequence, we obtain $|u(k,n) - \lambda(k,n)| = \mathcal{O}(2^{-2n})$ and $|l(k,n)-\lambda(k,n)| = \mathcal{O}(2^{-2n})$. Hence, the accuracy of the length estimation is given as

$$\begin{aligned} |\Lambda - \Lambda_n| &= \left|\sum_{k=0}^{2^n} \lambda(k,n) - [\alpha l(k,n) + (1-\alpha)u(k,n)]\right| \\ &\le \sum_{k=0}^{2^n} |\alpha(\lambda(k,n) - l(k,n)) + \\ &\quad (1-\alpha)(\lambda(k,n) - u(k,n))| \\ &= 2^n \mathcal{O}(2^{-2n}) = \mathcal{O}(2^{-n}). \qquad \textbf{Q.E.D.} \end{aligned}$$

By using relation (24), the lower and upper bounds $\mathcal{L}_n$ and $\mathcal{U}_n$ that we introduced in the beginning are

$$\mathcal{L}_n := \sum_k l(k,n) \le \Lambda \le \mathcal{U}_n := \sum_k u(k,n). \qquad (30)$$

Corollary

For any prescribed accuracy $\varepsilon > 0$, the expected number n of subdivisions to have an accuracy $|\Lambda - \Lambda_n| < \varepsilon$ is of order

$$\left\lceil \log_2 \frac{1}{\varepsilon} \right\rceil, \qquad (31)$$

where $\lceil x \rceil$ denotes the smallest integer larger than x.

IV. Improvment by using adaptivity

In the preceding section, we have developed a method which always subdivides each rational Bézier curve into two everywhere. In this section, we would like to discuss about an improvement of that approach. In fact, we will show how to develop adaptive strategy in order to only apply subdivisions at positions where they are necessary. In practice, when the rational Bézier curve is almost linear, we do not need to subdivided it any more. Our goal is then to identify positions where we need further subdivision without deteriorating the accuracy. As a consequence, we need a certain metric to quantify the error inside a subcurve. The quantities $l(k,n)$ and $u(k,n)$ of relation (28) are very good values for evaluating the flatness of the curve. We have proven in (29) that the difference between $l(k,n)$ and $u(k,n)$ tends to zero. That is, we should only apply subdivision at positions where $D(k,n) = |l(k,n) - u(k,n)|$ is large. One can even devise an adaptive strategy where we only refine the rational Bézier curves corresponding to

$$D(k,n) \ge \frac{\varepsilon}{2^n}. \qquad (32)$$

It is because if $D(k,n) < \varepsilon/2^n$, then all subcurves have error smaller than $\varepsilon/2^n$ so that refinement is unnecessary. By doing that, we need only to use subdivisions at subintervals where the flatness metric $D(k,n)$ indicates that the local upper bound $l(k,n)$ and lower bound $u(k,n)$ are still very different from one another.

Our method is summarized in the next algorithm. We update a list $\mathcal{S}$ of subcurves $\mathcal{S} = \{\mathbf{x}_0, \mathbf{x}_1, ...\}$. We denote by ESTIM($\mathbf{x}_p$) the value $D(k,n)$ of the subcurve $\mathbf{x}_p \in \mathcal{S}$.

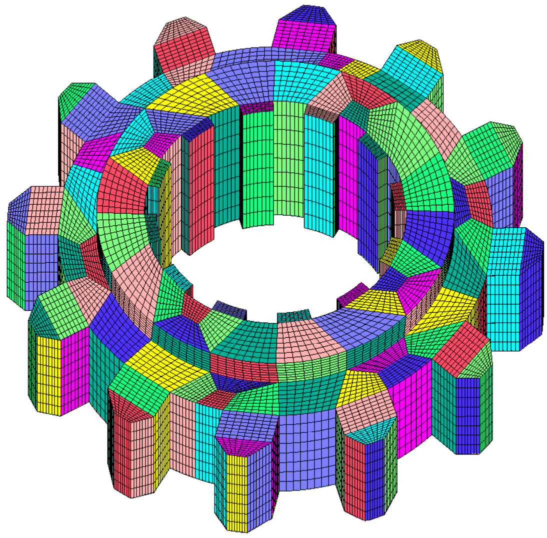

Fig. 4. Globally continuos mappings on a CAD surface.

Algorithm: Adaptive length computation of $\mathbf{x}$

1: Choose accuracy $\epsilon > 0$.
2: Estimate n by using (31)
3: Initialize the set of subcurves as $\mathcal{S} := \{\mathbf{x}\}$
4: for $(i = 1, \cdots, n)$
5: Find all $\mathbf{x}_p \in \mathcal{S}$ with ESTIM($\mathbf{x}_p$) $\ge \varepsilon/2^n$.
6: Subdivide $\mathbf{x}_p$ and compute $l(k,n)$ and $u(k,n)$.
7: $\Lambda_n := \Lambda_n + 0.5(l(k,n) + u(k,n))$. Update $\mathcal{S}$.
8: end for

V. Brief Application to CAD Parametrization

In our earlier works [8], [9], [10], we were interested in splitting a given model into four-sided patches $\mathcal{P}_i$. The generation of the mappings was mainly performed by using Coons maps [3] which are defined on the unit square $[0,1]^2$. We needed some functions

$$\psi_i(u,v) \qquad (u,v) \in [0,1]^2 \quad \text{with} \quad \mathcal{P}_i = \mathrm{Im}(\psi_i). \qquad (33)$$

Our main goal was that that mappings are globally continuous. Such a task can be illustrated by Fig. 4. For two incident four-sided patches $\mathcal{P}_i$ and $\mathcal{P}_j$, the image of u-constant or v-constant isoline of ψ_i should matched that of ψ_j at the interface. We have demonstrated that if we use the chord length reparametrization of the boundary curves then two adjacent Coons patches verify such matching conditions. That is, we have to reparametrize a boundary curve κ by $\tilde{\kappa}$ where $\kappa = \tilde{\kappa} \circ \chi$ in which

$$\chi(t) = \int_a^t \left\| \frac{d\rho}{dt}(\theta) \right\| d\theta \qquad (34)$$

where ρ is a well chosen function. The work presented in this paper is important when you want to generate the chord length reparametrization. A complete detail of such a reparametrization using curve length could be found in [10].

VI. Numerical results

In order to observe the practical efficiency of the former theory, we have implemented it in C/C++. We want to numerically

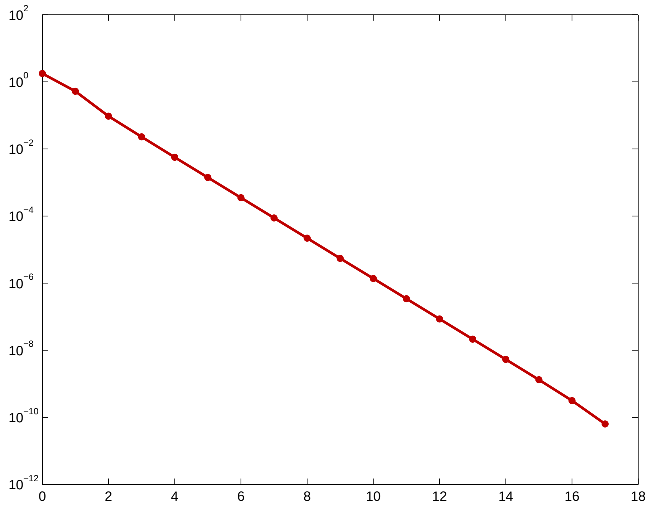

Fig. 5. Logarithmically scaled error $|\Lambda - \Lambda_n|$ in function of n.

TABLE I
BEHAVIOR OF THE UPPER AND LOWER BOUNDS WITH RESPECT TO n.

n	Lower bound $\mathcal{L}_n$	Upper bound $\mathcal{U}_n$	Difference $\lvert\mathcal{L}_n - \mathcal{U}_n\rvert$
0	4.5703476367	8.9798491325	4.4095e+00
2	4.9693082543	5.2046867676	2.3538e-01
4	4.9904688238	5.0045723724	1.4104e-02
5	4.9915226799	4.9950382924	3.5156e-03
7	4.9918519637	4.9920714912	2.1953e-04
9	4.9918725432	4.9918862629	1.3720e-05
10	4.9918735722	4.9918770021	3.4299e-06
11	4.9918738294	4.9918746869	8.5748e-07
13	4.9918739098	4.9918739634	5.3592e-08
15	4.9918739148	4.9918739182	3.3495e-09
16	4.9918739151	4.9918739159	8.3744e-10

investigate the dependence on n of the error $\Lambda - \Lambda_n$ and the bounds $\mathcal{L}_n$, $\mathcal{U}_n$. Thus, let us consider a rational Bézier curve where $m = 3$ and the control points with the corresponding weights are

$$\begin{aligned} \mathbf{b}_0 &= [0.143, 3.021, 2.045], & \omega_0 = 1.2, \\ \mathbf{b}_1 &= [1.945, 4.192, 2.223], & \omega_1 = 0.9, \\ \mathbf{b}_2 &= [2.043, 0.012, 2.185], & \omega_2 = 1.5, \\ \mathbf{b}_3 &= [3.543, 2.078, 2.865], & \omega_3 = 0.6, \end{aligned}$$

where the expected value of the length is 4.9918739152. A plot of the error in terms of n is depicted in Fig. 5 which confirms our theory. Note that the vertical axis is logarithmically scaled. Additionally, the numerical behavior of the difference of the lower bound $\mathcal{L}_n$ and upper bound $\mathcal{U}_n$ is seen in Table I which is also conform to the theoretical prediction.

VII. CONCLUSION

We have presented a method for estimating the lengths of a rational Bézier curve. We suppose that the weight function is uniformly bounded which is not a very restrictive assumption in practice. We have found a lower bound and an upper bound which are easy to estimate.

ACKNOWLEDGMENT

The author is very grateful to Prof. Brunnett from the Technical University of Chemnitz who has introduced him to the topic of Geometric Modeling and CAD. He thanks also Prof. Schneider from the University of Berlin for supporting his research at the Institute of Computer Science in the University of Kiel. Last but not least, the author wants to acknowledge the help of Prof. Harbrecht from the University of Bonn who allowed him to pursue his current Post-Doctoral research.

REFERENCES

[1] G. Brunnett, "Geometric design with trimmed surfaces", *Computing Supplementum*, vol. 10, pp. 101–115, 1995.

[2] C. deBoor, *A practical guide to splines*, New York: Springer Verlag, 1978.

[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Boston: Academic Press, 1997.

[4] O. Figueiredo, J. Reveilles and R. Hersch, "Digitization of Bézier curves and patches using discrete geometry", in *Proc. 8th Int. Conf. Discrete Geometry for Computer Imagery*, Marne-la-Vallée, 1999, pp. 388–398.

[5] M. Floater, "Chordal cubic spline interpolation is fourth order accurate", *IMA J. Numer. Anal.*, vol. 26, pp. 25–33, 2006.

[6] T. Hain, "Rapid termination evaluation for recursive subdivision of Bézier curves", in: *Proc. Int. Conf. on Image Science, Systems, and Technology*, Las Vegas, 2002, pp. 323–328.

[7] H. Prautzsch, W. Boehm, M. Paluszny, *Bézier and B-Spline techniques*, Berlin: Springer Verlag, 2002.

[8] M. Randrianarivony, "Geometric processing of CAD data and meshes as input of integral equation solvers". Ph.D. dissertation, Dept. Comput. Science, Chemnitz University of Technology, Chemnitz, Germany, 2006.

[9] M. Randrianarivony, G. Brunnett, "Molecular surface decomposition using geometric techniques", in *Proc. Conf. Bildverarbeitung für die Medizine*, Berlin, 2008, pp. 197–201.

[10] M. Randrianarivony, G. Brunnett, "Preparation of CAD and Molecular Surfaces for Meshfree Solvers", *Int. Workshop Meshfree Methods for PDE*, 2007 (accepted for publication).

[11] J. Roulier, "Specifying the arc length of Bézier curves", *Computer Aided Geometric Design*, Vol. 10, pp. 25–56, 1993.

[12] H. Seidel, "Polar forms for geometrically continuous spline curves of arbitrary degree", *ACM Trans. Graph.*, Vol. 12, pp. 1–34, 1993.

[13] M. Walter, A. Fournier, "Approximate arc length parametrization", in *Proc. 9th Brazilian Symposium on Computer Graphics and Image Processing*, Brazil, 1996, pp. 143–150.

Maharavo Randrianarivony obtained his Master of Science from Department of Mathematics Universtiy of Chemnitz in Germany in 2001. He obtained his PhD from the Department of Computer Science. Afterwards he has done a PostDoc for one year in Institute of Computer Science in Kiel. Now he works as an advanced researcher at the Faculty of Mathematics of the University of Bonn.