International Journal of Shape Modeling, Vol. 2, No. 2&3 (1996) 103–116

REPARAMETRIZATION OF NURBS CURVES*

GIULIO CASCIOLA† and SERENA MORIGI‡
Department of Mathematics , University of Bologna
P.zza di Porta San Donato 5, 40127 Bologna, Italy

Received 1 March 1996
Revised 9 September 1996
Communicated by B. Falcidieno

In geometric design, it is often useful to be able to give an arc length reparametrization for NURBS curves, that keeps the curve a NURBS too. Since parametric rational curves, except for straight lines, cannot be parametrized by arc length, we developed a numerical method of approximating the arc length parametrization function. In this way it was possible to obtain a good parametrization of a NURBS curve with respect to arc length. Numerical results show a good behaviour of the proposed method on several test curves.

Keywords: Reparametrization; NURBS; Arc length parametrization; Approximation.

1. Introduction

We are concerned with the problem of parametric NURBS curve reparametrization. This consists in changing the current parameter of a curve with another parameter using a reparametrization function. It should be noted that the shape of the curve remains unchanged during this process; only the way the curve is described is altered.

In our case, the fact that the curve is defined as a ratio of two polynomial splines enables us to apply a rational polynomial function that will keep the curve a NURBS.

In particular, if it is important that the degree of the curve should be kept unchanged, we may choose a rational linear reparametrization function.

In this case [1] and [2] give an explicit expression for the reparametrized NURBS curve.

The most common and useful parametrization, from a computational point of view, is the arc length, where a unit change in the parametrizing variable results in a unit change in curve length. Such a parametrization is unfortunately almost never possible for NURBS, unless for straight lines (see [3]). These considerations lead us to search for a parametrization that best fits the arc length.

*This research was supported by CNR-Italy, Contract n.95.00730.CT01.
†e-mail: casciola@dm.unibo.it
‡e-mail: serena@csr.unibo.it

To our knowledge no author in the literature has considered this problem, while in several papers numerical techniques to compute arc length were developed (see, for example, [4] and [5]).

This paper is essentially in two parts. In the first, we consider the reparametrization problem on a single interval, and in the second part we extend this result to deal with adaptive piecewise reparametrization.

In the next section we consider some essential motivations that led us to consider this topic.

In section 3 a formulation of arc length reparametrization is given; while in section 4 we introduce the concept of approximate reparametrization. Reparametrization on a single interval is considered in section 5, studying the particular reparametrization function used in this paper and the numerical method applied to compute it. In order to establish the quality of the proposed reparametrization method, we present two evaluation parameters in section 6, and in section 7 we report the results obtained on several test curves. The second part of the paper is introduced by a description of a C^0 adaptive piecewise reparametrization technique in section 8, followed by the respective experimental results in section 9. Finally, an extension to C^1 reparametrization is considered and some concluding remarks are given.

2. Motivation

To date, NURBS are the most general parametric representation in geometric modelling. The most frequently used NURBS design techniques are the specification of a control polygon, and the interpolation or approximation of data points to generate the initial shape. This is then refined into the desired final shape through the interactive adjustment of control points and weights.

A possible shortcoming of this process is that small changes in the shape of the curve can also lead to a bad parametrization with respect to the arc length.

Figure 1 shows a curve modified in shape by increasing weights w_3 and w_4. The curves, before and after modification, are evaluated and drawn at parameter values uniformly spaced in the parametric domain (see, the drawing in dots in Fig. 1). We can observe the effects of significant changes in the parametrization; note that the arc length parametrization would yield points spaced uniformly on the curve.

Moreover, when a badly parametrized curve is used in the construction of surfaces (cross-sectional techniques) badly parametrized surfaces are obtained.

Any numerical method, or simply the rendering procedure, applied to curves or surfaces, are affected by their particular parametrization in terms of computational complexity and numerical stability, as shown in the following example.

We consider the Bottle/Plane intersection computed by a Surface/Surface Intersection algorithm (SSI) based on a geometric-numerical approach (see [6]). The geometric phase is concerned with the evaluation of some Starting Points (SP), while the numerical phase consists in a marching algorithm that, starting from the SPs determines the intersection curves through a certain Number of Points (NP).

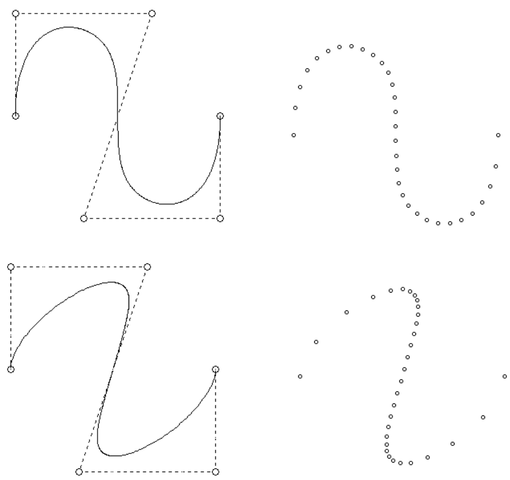

Fig. 1. The initial shape (top); the final shape after the adjustment of weights w_3 and w_4 (bottom).

We will compare case A of the intersection between the two well-parametrized surfaces (see Figure 2) with case B relating to the intersection between the same surfaces, in which the Bottle surface is badly parametrized; in fact it is built as a revolving surface with a badly parametrized circular trajectory curve (see Figure 3).

If the SSI algorithm uses the smallest number of SPs, corresponding to the pieces of the intersection curve on the parametric domain (3 in the example above), in case A 397 NPs are computed, compared with 794 NPs of case B.

This increased computational cost is due to the fact that the solution for the parametric domain in case A is given by a smoother curve than in case B. In case B the marching phase encounters problems in the spans of higher curvature making it impossible to find a solution in less than 794 NPs.

Allowing algorithm SSI to use a larger number of SPs, a solution is reached in case A with 253 NPs as against 590 NPs in case B.

Other examples where working with badly parametrized curves or surfaces can be disadvantageous are both uniform and adaptive rendering, motion control in computer animation, curve/curve intersection, ray/surface intersection in ray tracing, implicitization.

3. Arc length reparametrization of NURBS curves

Let $\mathbf{c}(t)$ be a NURBS curve of order m (degree $m-1$), expressed as

$$\mathbf{c}(t) = \sum_{i=-m+1}^{K} P_i R_{i,m}(t/w_{i-m+1}, \cdots, w_i) \qquad t \in [0,1]$$

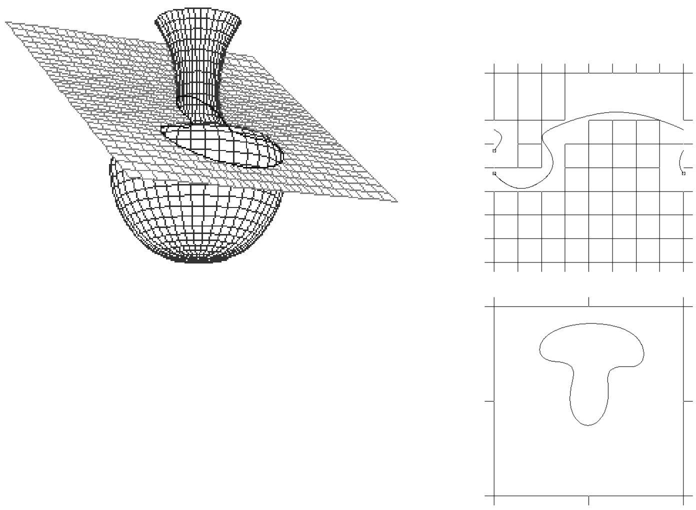

Fig. 2. Well-parametrized surfaces; Bottle domain (top), Plane domain (bottom).

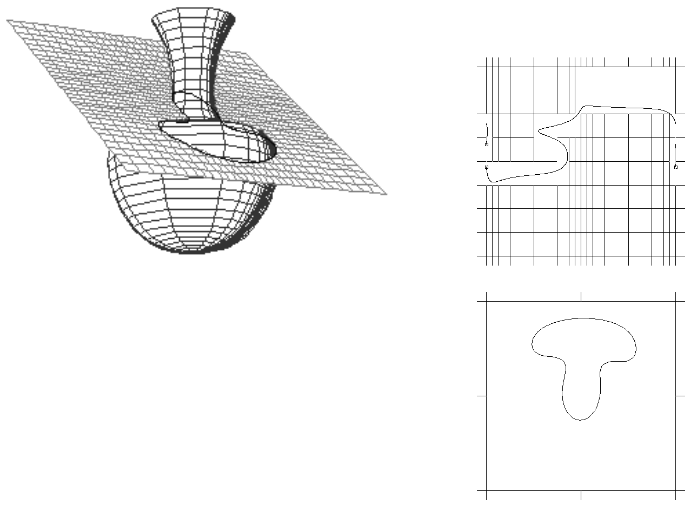

Fig. 3. Badly parametrized surfaces; Bottle domain (top), Plane domain (bottom).

where $P_i \in \mathrm{I\!R}^2$ are the control points, w_i their corresponding positive real weights, and $R_{i,m}$ the rational B-spline basis functions defined on the knot sequence $(t_i)_{i=-m+1,\cdots,m+K}$ where $t_0 = 0$ and $t_{K+1} = 1$.

We denote by r_i the multiplicity of knot t_i, $1 \leq i \leq K$ (that is, the number of times t_i occurs in the knot sequence). We assume that r_i is always less than or equal to $m - 1$.

Note that at knot t_i, $1 \leq i \leq K$, the order of continuity of the homogeneous curve is $m - r_i - 1$, since this curve is a nonrational B-spline curve in 3D space.

The arc length parametrization function $\phi(t)$ for $\mathbf{c}(t)$ is defined as

$$\phi : [0, 1] \Longrightarrow [0, L]$$

$$\tau = \phi(t) = \int_0^t ||\mathbf{c}^{'}(u)||_2 du$$

$$with \quad L = \int_0^1 ||\mathbf{c}^{'}(u)||_2 du$$

where L denotes the length of the curve $\mathbf{c}(t)$. Note that the $\phi(t)$ function is strictly increasing. In figure 6 two examples of $\phi(t)$ are shown.

Let a reparametrization function be a $\psi(t)$ satisfying the following conditions:

$$\psi : [0, 1] \Longrightarrow [0, \Lambda]$$

$$\psi^{'}(t) > 0 \quad t \in [0, 1]$$

then a reparametrized curve can be defined as

$$\hat{\mathbf{c}}(\tau) = \mathbf{c}(\psi^{-1}(\tau)) \;\; \tau \in [0, \Lambda]$$

A curve will be reparametrized by arc length when the $\psi(t)$ is the $\phi(t)$ function.

We now assume that $\hat{\mathbf{c}}(\tau)$ is a reparametrized curve by arc length.

In general $\hat{\mathbf{c}}(\tau)$ is not a NURBS curve, unless $\mathbf{c}(t)$ is a straight line.

We remind the reader that the following relationship holds:

$$||\hat{\mathbf{c}}^{'}(t)||_2 = 1 \;\; \forall \;\; t \in [0, 1]$$

This means that the scalar speed of a point moving along the curve is constant and equal to 1. We will assume this to be the property that characterizes the arc length parametrization of a curve.

4. Approximate reparametrization

Since, as mentioned above, $\phi(t)$ is not a rational function unless $\mathbf{c}(t)$ is a straight line, and because of the need to consider a rational reparametrization function in order to obtain a NURBS curve, we are led to determine a $\phi(t)$ approximation by means of a rational function $\psi(t)$. In this way we hope to obtain a good approximation of arc length parametrization.

We are searching for a $\psi(t)$ that satisfies the following conditions:

1. $\psi(t)$ *rational linear function*
2. $\psi(0) = \phi(0), \;\; \psi(1) = \phi(1)$
3. $\psi(t)$ *shape preserving of* $\phi(t)$
4. $||\psi(t) - \phi(t)||_\infty$ *small*

Condition 1. is necessary to guarantee that the reparametrized curve remains a NURBS curve in a space with the same dimension as $\mathbf{c}(t)$; condition 2., necessary for the reason we will describe in the following, together with condition 4., involve the solution of a problem similar to the best uniform approximation problem (b.u.a.) with interpolatory constraints, by means of a rational linear function. Finally, condition 3. is used to guarantee a small minmax value as a solution to the constrained b.u.a. problem.

Since in the literature no result relating to the constrained rational b.u.a. problem is known, but results concerned with constrained polynomial b.u.a. problem (see [7]) and with rational b.u.a. problem (see [8]) are known, we have proceeded as if the results were also valid in the constrained rational case. Hence, we searched for a $\psi(t)$ function from the set of all functions characterized by the property of having an error function, $\psi(t) - \phi(t)$, that equioscillates at least on two points ($m + n + 2$ equioscillation points, with m and n respectively the degrees of the numerator and denominator of the $\psi(t)$ function, minus two points corresponding to the interpolation constraints at the endpoints).

This way of proceeding guarantees a solution to our problem even with no theoretical support for the b.u.a. problem.

5. Reparametrization on a single interval

The objective of this section is to make a suitable choice for $\psi(t)$. To do this we will give a characterization of $\psi(t)$ and provide the numerical method used in order to achieve it.

5.1. *Approximate reparametrization function*

We consider the following reparametrization function:

$$\psi(t) = \frac{\phi(a)(b-t) + w\phi(b)(t-a)}{(b-t) + w(t-a)} \quad w > 0 \quad t \in [a, b]$$

which is a linear rational function that interpolates $\phi(t)$ at the endpoints of a generic interval $[a, b]$.

From a study of its behaviour, for any w we have $\psi'(t) > 0 \; \forall t$, that is, the $\psi(t)$ is strictly increasing throughout $[a, b]$; this implies that it is an invertible function and therefore can be used as a reparametrization function.

Moreover, $\psi''(t) \neq 0 \ \forall t$, hence the $\psi(t)$ function is either concave or convex on the interval $[a, b]$. This property, together with monotonicity, lead us to apply the $\psi(t)$ approximation function only to intervals $[a, b]$ in which $\phi(t)$ is either concave or convex and this makes us satisfy condition 3 of section 4.

5.2. *The second Remez algorithm*

The object is to find a $\psi(t)$ as an approximation of the $\phi(t)$ which is the limit of a sequence of $\psi(t)$ that have an error function that alternates on at least two points.

To do this we use the second Remez algorithm (see [8]) in which at each iteration we check that the maximum error function is still decreasing. If this does not occur, the procedure is stopped and the penultimate $\psi(t)$ found in the sequence is taken as the solution.

The following non-linear system, with unknowns d and w, satisfies the equioscillation constraints:

$$\psi(t_i) - \phi(t_i) = (-1)^i d \quad i = 0, 1$$

that is:

$$\begin{cases} \phi(a)(b - t_0) + w\phi(b)(t_0 - a) + (-d - \phi(t_0))[b - t_0 + w(t_0 - a)] = 0 \\ \phi(a)(b - t_1) + w\phi(b)(t_1 - a) + (+d - \phi(t_1))[b - t_1 + w(t_1 - a)] = 0 \end{cases} \tag{1}$$

Procedure

1. $ymax\_old = \infty$
2. Let $t_0 = a + \frac{(b-a)}{4}$ and $t_1 = b - \frac{(b-a)}{4}$
3. Compute d and w explicitly from (1)
4. Let $e(t) = \psi(t) - \phi(t)$ be the error function and $(\bar{t}_0, y_0)$, $(\bar{t}_1, y_1)$ such that

$$y_0 = \max_{\substack{t \\ sgn(e(t)) = sgn(e(t_0))}} |e(t)| \quad and \quad y_1 = \max_{\substack{t \\ sgn(e(t)) = sgn(e(t_1))}} |e(t)|$$

$$ymax = max(y_0, y_1)$$

5. If $ymax\_old < ymax$ then $w = w\_old$; go to step 9.
6. Update (t_0, t_1) with $(\bar{t}_0, \bar{t}_1)$ so that the error function has two variations in sign
7. $w\_old = w$; $ymax\_old = ymax$
8. If $ymax - |d| > \epsilon$ go to step 3.
9. end

Proposition System (1) always provides a unique function $\psi(t)$ satisfying the equioscillation property on two points.

Proof. The second equation of (1) can be solved for d so that after substitution in the first equation we can reduce the problem to the solution of the following second degree equation:

$$\alpha w^2 + \beta w + \gamma = 0$$

with

$$\alpha = [2\phi(b) - \phi(t_1) - \phi(t_0)](t_1 - a)(t_0 - a)$$
$$\beta = [\phi(a) + \phi(b) - \phi(t_1) - \phi(t_0)][(b - t_0)(t_1 - a) + (b - t_1)(t_0 - a)]$$
$$\gamma = [2\phi(a) - \phi(t_1) - \phi(t_0)](b - t_1)(b - t_0)$$

Note that the discriminant Δ is always positive and furthermore $\Delta > |b|$; this implies that we obtain two real solutions that are different in sign. Hence, the unique alternating function derives from considering the positive w solution $\square$.

6. Evaluation parameters

In order to test the good or bad quality of approximate reparametrization, we chose the two parameters p and q and defined them as follows.

Let

$$p := \|\| \mathbf{c}'(t) \|_2 - 1 \|_\infty$$

be the displacement from the unit speed; in the case of arc length reparametrization, p is equal to 0.

We consider a uniform subdivision of the parametric domain $(\tau_i)_{i=1,\cdots,N}$ and we define by

$$q := \frac{\max_{i=2,\cdots,N} \int_{\tau_{i-1}}^{\tau_i} \| \mathbf{c}'(t) \|_2 \, dt}{\min_{i=2,\cdots,N} \int_{\tau_{i-1}}^{\tau_i} \| \mathbf{c}'(t) \|_2 \, dt}$$

the displacement from the uniform distribution of the points on the curve; in fact q is the ratio between the segment of maximum and minimum length of the curve, where a segment is defined as a span of curve in between two consecutive points τ_i. In the case of an arc length parametrization q is equal to 1, as points on the curve are uniformly spaced.

Parameter q provides more information than parameter p; in fact, if the approximation strip of the unit speed is defined as that given by $[\min \| \mathbf{c}'(t) \|_2, \max \| \mathbf{c}'(t) \|_2]$, parameter p provides only partial information about the strip while q gives more complete information of its size; in fact

$$q \approx \frac{\max \| \mathbf{c}'(t) \|_2}{\min \| \mathbf{c}'(t) \|_2}$$

holds.

7. Numerical results: part 1

In this initial phase seven test curves have been considered, all having convex $\phi(t)$ over the whole parametric domain in order to apply a single approximation $\psi(t)$ as indicated in section 5.

The curves considered here are chosen of different degrees 1, 2, 3 and 4, as pointed out from their names, and are badly parametrized with respect to arc length; the ***conv*1** curves all have uniformly spaced control points and increasing weights, while the ***conv*2** curves have control points placed at increasing distances and equal weights.

Table 1 reports the initial value of the evaluation parameters $\boldsymbol{p}$ and $\boldsymbol{q}$, denoted by $\boldsymbol{p}_0$ and $\boldsymbol{q}_0$, and the values after applying a single rational reparametrization to the test curves. The last column gives the maximum error function $\boldsymbol{e}(\boldsymbol{t})$ indicated with $\boldsymbol{MinMax}$ since, in all the cases in our experimentation, Remez algorithm found the b.u.a.; in fact the divergence test of the maximum error function was never positive and convergence was always reached in three or four iterations.

Further proof that the approximation found was the constrained b.u.a. is given by the fact that reparametrizing the curve a second time the same parametrization was obtained.

In fact, it must be remembered that the double reparametrization obtained through the two functions ψ_1 and ψ_2 is equivalent to a single reparametrization with the composite $\psi_2(\psi_1(t))$, which is still a linear rational function; therefore, the fact that the reparametrization does not vary means that the $\psi_2(\psi_1(t)) \equiv \psi_1(t)$, and then ψ_1 is already the constrained b.u.a. of the $\phi(t)$ with $\boldsymbol{MinMax}$ given by $\| \phi(t) - \psi_1(t) \|_\infty$.

Finally, we can observe that the $\boldsymbol{conv}1 - \boldsymbol{d}1$ curve is a straight line and that our method manages to reparametrize it exactly at arc length.

Table 1. Reparametrization on a single interval.

TEST CURVES	p_0	p	q_0	q	*MinMax*
$conv1 - d1$	3.00	0	15.88	1	0
$conv1 - d2$	0.80	0.10	3.99	1.17	0.0064
$conv2 - d2$	0.69	0.074	3.49	1.20	0.012
$conv1 - d3$	2.20	0.25	7.94	1.48	0.017
$conv2 - d3$	1.33	0.14	4.93	1.24	0.0078
$conv1 - d4$	2.24	0.21	12.38	1.39	0.025
$conv2 - d4$	0.93	0.22	5.64	1.45	0.036

8. Piecewise adaptive reparametrization

In practice, a $\mathbf{c}(t)$ curve will have a strictly increasing $\phi(t)$ which will not be simply all concave or convex and is therefore difficult to approximate it with a single linear rational function, that is to satisfy condition 4 of section 4.

As already mentioned, the idea is to approximate the $\phi(t)$ in spans which are only concave or convex by means of a $\psi(t)$ function.

In other words once a tolerance has been assigned, we propose an adaptive method that consists in carrying out an adaptive approximation on concave/convex spans (pieces of $\phi(t)$ function only concave or convex) until the requested tolerance of approximation has been reached on every span. The first step is to compute an approximate $\psi(t)$ over the whole parametric domain; if tolerance is not reached, the parametric interval is divided in two intervals to respect the concave/convex spans or divided in half if the correspondent span is already concave or convex.

The approximation is then repeated on each span, dividing it again if necessary in order to reach tolerance value.

Once the piecewise linear rational function $\psi(t) \equiv \psi_i(t)$, with $t \in [s_i, s_{i+1}]$ $\forall i$, has been obtained, the reparametrization process consists in subdividing the curve in correspondence of each s_i by inserting knots (each with multiplicity m), and applying the respective reparametrization $\psi_i(t)$ to each curve span $\mathbf{c}([s_i, s_{i+1}])$ (see [1] and [2]). As $\psi(t)$ function will be C^0, see condition 2 of section 4, the reparametrized curve will be C^0. Note that $\mathbf{c}(t)$ consists of many pieces of curve (interior knots with multiplicity m).

To make it as a NURBS curve once again a procedure of knot-removal is applied to all knots (see [9]). If this phase is preceded by a suitable scaling of the weights of each piece of curve (this do not modify the parametrization), the knot-removal is bound to be successful at least once on each knot s_i by reducing their multiplicity to at least $m - 1$.

This procedure implies the representation of the reparametrized curve in a space of larger dimension than that of the original curve because of the above-mentioned knot insertion.

To make the assigned tolerance curve length independent ($\phi : [0, 1] \Longrightarrow [0, L]$), we have established to assign a tolerance value as if the curve was of unit length and then scale it according to the real length of the curve. This involves the invariance of the reparametrization with respect to a scaling of the curve.

Table 2. Piecewise adaptive reparametrization; $tol = 0.005$.

TEST CURVES	p_0	p	q_0	q	*NoI*
$conv1 - d2$	0.80	0.046	3.99	1.07	2
$conv2 - d2$	0.69	0.039	3.49	1.07	2
$conv1 - d3$	2.20	0.072	7.94	1.10	3
$conv2 - d3$	1.33	0.056	4.93	1.11	2
$conv1 - d4$	2.24	0.098	12.38	1.18	3
$conv2 - d4$	0.93	0.064	5.64	1.09	2

9. Numerical results: part 2

In this second phase the accuracy of the piecewise adaptive approximation technique is assessed in order to use it as a semi-automatic method to obtain a well parametrized curve with respect to arc length.

A threshold value of good parametrization has been chosen in the value $q = 1.25$, below which parametrization could be considered accurate with regard to arc length; this value considers the segment of maximum length as 1/4 greater than the segment of minimum length of the curve.

Table 1 shows that it is not always possible to arrive at this threshold value using a single approximation.

In Table 2 we find the results of the adaptive piecewise approximation method applied to the same test curves as Table 1, apart from the $conv1 - d1$, with a tolerance of $tol = 0.005$.

Table 3. Piecewise adaptive reparametrization by approximation function.

TEST CURVES	*tol*	p_0	p	q_0	q	*NoI*
coil	0.01	4.71	0.59	14.17	1.73	6
	0.005		0.59		1.68	7
	0.0005		0.08		1.09	13
sole	0.01	0.82	0.22	2.88	1.49	4
	0.005		0.16		1.38	5
	0.0005		0.06		1.08	14
flower	0.01	1.17	0.38	15.71	2.08	6
	0.005		0.26		1.48	11
	0.0005		0.15		1.19	21
star	0.01	0.72	0.83	5.06	4.98	8
	0.005		0.21		1.46	11
	0.0005		0.061		1.10	23
heart	0.01	1.86	0.69	32.22	4.09	4
	0.005		0.62		3.24	5
	0.0005		0.10		1.19	17
guitar	0.01	1.27	0.84	12.56	3.93	12
	0.005		0.84		3.93	12
	0.0005		0.14		1.25	32

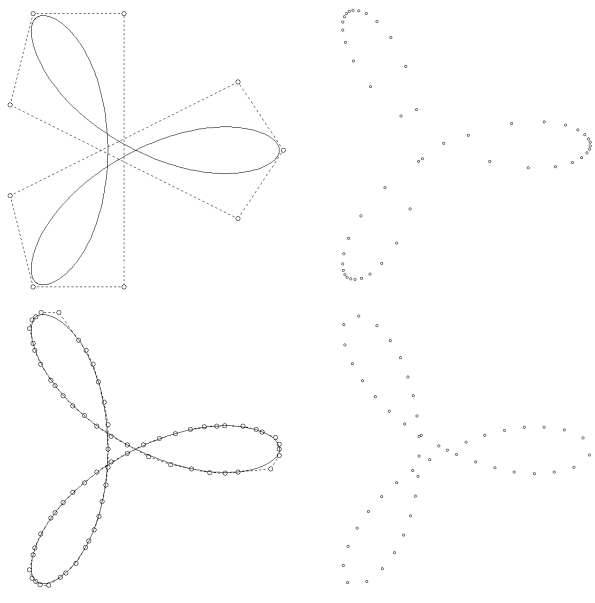

Fig. 4. Flower.

The results show that the threshold value has been reached for all the curves. The last column, marked NoI, reports the number of intervals in which the method has subdivided the initial parametric domain in order to reach the approximate precision requested.

Table 3 reports the results relative to six more complex test curves (the $\phi(t)$ is not always convex), varying in degree, shape and initial parameter values p_0 and q_0.

Figures 4 and 5 show two of the test curves considered, with their relative control points; the curve and its drawing in dots relating to the initial parametrization are given above, and below is the final result after reparametrization with $\boldsymbol{tol} = 0.0005$.

Figure 6 illustrates the graphs of the initial $\phi(t)$ functions relative to the test curves of Figures 4 and 5.

An asymptotic reduction of the parameter values p and q, due to a decreasing tolerance, can be observed in the tests summarized in Table 3, even though this is obtained using numerous subdivisions.

It is observed that all the curves undergoing a $\boldsymbol{tol} = 0.0005$ go below the threshold value, although the initial values are extremely varied.

However, with bigger tolerances the NoI is fewer, and parametrization is still accurate. Note that in a geometric design system we can accept a non-optimal parametrization in order to avoid to split the curve too much.

10. C^1 reparametrization function

If the initial curve is at least C^1 it could be useful to determine a reparametrized curve that is at least C^1. This is made possible by approximating $\phi(t)$ with a $\psi(t)$ that is C^1.

In our piecewise adaptive scheme this was obtained by using the piecewise linear rational interpolation function $\psi(t) \equiv \psi_i(t)$, with $t \in [s_i, s_{i+1}]$ $\forall i$, defined as in [10] that is on every two consecutive intervals $[s_i, s_{i+1}], [s_{i+1}, s_{i+2}]$ such that

$$\begin{aligned}
\psi_i(s_i) &= \phi(s_i), \\
\psi_i^{'}(s_i) &= \phi^{'}(s_i), \\
\psi_{i+1}(s_{i+2}) &= \phi(s_{i+2}), \\
\psi_{i+1}^{'}(s_{i+2}) &= \phi^{'}(s_{i+2}), \\
\psi_i(s_{i+1}) &= \psi_{i+1}(s_{i+1}), \\
\psi_i^{'}(s_{i+1}) &= \psi_{i+1}^{'}(s_{i+1}).
\end{aligned}$$

Table 4, that must be compared with Table 3, reports the results of the method just described on the six test curves already presented.

We can observe that, in order to avoid excessive subdivisions into spans, a higher tolerance values must be given. In fact, in order to satisfy the C^1 condition, the requested tolerance value is reached only by making a lot of subdivisions.

However, all these subdivisions lead to a good reparametrization.

As in case C^0, also in C^1 reparametrization, an asymptotic reduction of the parameter values p and q can be experimentally observed.

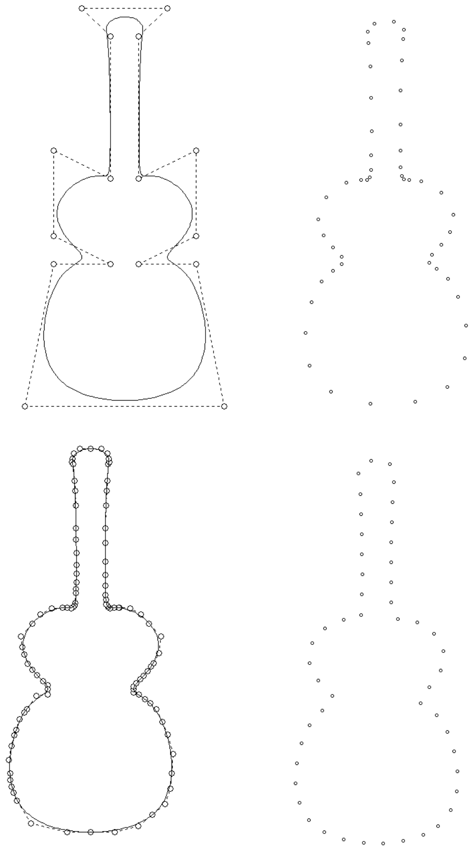

Fig. 5. Guitar.

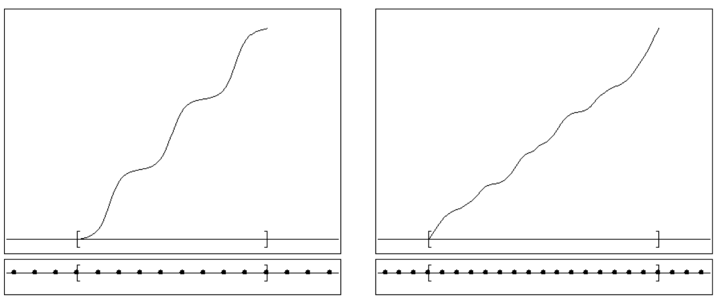

Fig. 6. $\phi(t)$ functions of flower (left) and guitar (right).

Table 4. Piecewise adaptive reparametrization by interpolation function; $tol = 0.005$.

TEST CURVES	p_0	p	q_0	q	*NoI*
coil	4.71	0.062	14.17	1.12	17
sole	0.82	0.073	2.88	1.13	15
flower	1.17	0.13	15.71	1.25	29
star	0.72	0.076	5.06	1.12	23
heart	1.86	0.11	32.22	1.19	21
guitar	1.27	0.19	12.56	1.45	42

When the number of subdivisions is the same, the C^0 reparametrization is shown to be better than the C^1 reparametrization.

11. Conclusions

In this paper we considered the problem of parametric NURBS curves reparametrization. We proposed a concave/convex piecewise adaptive technique in order to approximate the arc length parametrization function. A linear rational approximation function with constraints was applied to each span. This experimentally proved to be the best uniform approximation. The proposed method can be used both globally and locally as a semi-automatic technique for improving the parametrization of curves in a geometric design system in order to reduce the computational cost and increase the stability of the numerical methods within the system.

References

1. E.T.Y. Lee and M.L. Lucien , Möbius reparametrization of rational B-splines, *Computer Aided Geometric Design* **8** (1991) 213-215.
2. L. Alt , Rational linear reparametrization of NURBS and the blossoming principle, *Computer Aided Geometric Design* **10** (1993) 465-467.
3. R.T. Farouki and T. Sakkalis , Real rational curves are not 'unit speed', *Computer Aided Geometric Design* **8** (1991) 151-157.
4. B. Guenter and R. Parent , Computing the Arc Length of Parametric Curves, *IEEE Computer Graphics & Applications* (May 1990) 72-78.
5. R.J. Sharpe and R.W. Thorne , Numerical method for extracting an arc length parameterization from parametric curves, *Computer-Aided Design* **14** (1982) 79-81.
6. G. Casciola and S. Morigi , Il problema SSI nella modellazione solida con superfici NURBS, *Accademia delle Scienze dell' istituto di Bologna* serie V, n.6 (1995) 107-127.
7. F. Deutsch , On uniform approximation with interpolatory constraints, *Journal of Mathematical Analysis and Applications* **24** (1968) 62-79.
8. E.W. Cheney , *Introduction to Approximation Theory* (McGraw-Hill, 1966).
9. W. Tiller , Knot-removal algorithms for NURBS curves and surfaces, *Computer-Aided Design* **24** (1992) 445-453.
10. R.D. Fuhr and M. Kallay , Monotone linear rational spline interpolation, *Computer Aided Geometric Design* **9** (1992) 313-319.