Accurate Parametrization of Conics by NURBS

Carole Blanc and Christophe Schlick
University of Bordeaux

Zigzag reparametrization improves the parametrization of a NURBS curve or surface according to a given criterion, as illustrated here through parametrization of a circle.

One argument often given to explain the popularity of NURBS (nonuniform rational B-spline) is that it permits the definition of free-form curves and surfaces (as do most spline models). It also provides an exact representation of conic sections and thus of a large set of curves and surfaces used intensively in CAD: circular arcs, circles, cylinders, cones, spheres, surfaces of revolution, and so forth. Nevertheless, few published works discuss the mathematical properties behind the representation of conics by NURBS except for two recent monographs by Piegl and Tiller[1] and by Farin.[2] This article does not pretend to fill this theoretical lack but rather intends to deal with the following problems:

- All known NURBS representations of curves and surfaces based on conics have only a C^1 continuity. Moreover, no technique exists that would eventually allow us to find a parametrization with a higher level of continuity.
- The parametrization resulting from the NURBS representation of conics can deviate significantly from the ideal arc length (that is, uniform) parametrization. The only known solution to reduce this deviation is to increase the number of control points of the spline (by using refinement algorithms, for instance), but such a process converges only slowly to the uniform parametrization.

The solution we propose here uses an original reparametrization process that we call *zigzag reparametrization*, based on a specific family of rational polynomials. This technique improves the parametrization by minimizing a given error criterion (in our case, giving a higher order continuity as well as a more uniform parametrization). On the other hand, it raises the degree of the resulting NURBS (in our case, from two to four).

Conics, circles, spheres, and splines

We briefly review here how conic sections (and related curves and surfaces) can be modeled by splines.

Conics

In the early 1800s, Moebius stated that a conic can be considered the projection on $\mathbb{R}^2$ of a parabola defined in $\mathbb{R}^3$. Using this result, we can show that a conic can also be expressed as a *rational quadratic Bezier curve* (and alternatively as a *quadratic NURBS*).[3,4] Such a curve is defined by three points, P_0, P_1, P_2, and three weights, w_0, w_1, w_2:

$$\forall t \in [0,1] \quad C(t) = \frac{w_0\, B_0^2(t) P_0 + w_1\, B_1^2(t) P_1 + w_2\, B_2^2(t)\, P_2}{w_0\, B_0^2(t) + w_1\, B_1^2(t) + w_2\, B_2^2(t)} \tag{1}$$

or by explicating the Bernstein polynomials $B_0^2(t)$, $B_1^2(t)$, and $B_2^2(t)$:

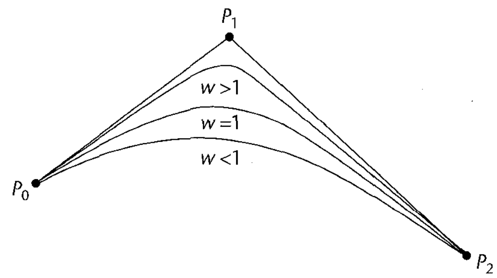
$$\forall t \in [0,1] \quad C(t) = \frac{w_0\, (1-t)^2 P_0 + 2w_1 (1-t) t\, P_1 + w_2\, t^2\, P_2}{w_0 - 2t(w_0 - w_1) + t^2 (w_0 - 2w_1 + w)} \tag{2}$$

Parameters w_0, w_1, and w_2 are linked by the *conic invariant*[5] relationship $w_0 w_1 / w_1^2$ which means that every set (w_0, w_1, w_2) that provides the same value for the conic invariant also provides the same curve.[3,4,6] Moreover, because the roles of P_0 and P_2 are symmetric, it is preferable to have $w_0 = w_2$. Finally, without loss of generality, we can assume that $w_0 = w_2 = 1$ and $w_1 = w$ (this condition is sometimes called the *normal form* of the rational curve[7]), which leads to

$$\forall t \in [0,1] \quad C(t) = \frac{(1-t)^2\, P_0 + 2w\, (1-t) t\, P_1 + t^2\, P_2}{1 - 2t(1-w) + 2t^2 (1-w)} \tag{3}$$

This expression enables a simple classification of the resulting curve (as shown in Figure 1):

- if $w < 1$, then $C(t)$ is an ellipse,
- if $w = 1$, then $C(t)$ is a parabola, and
- if $w > 1$, then $C(t)$ is a hyperbola.



1 **Three types of conics.**

Circular arcs

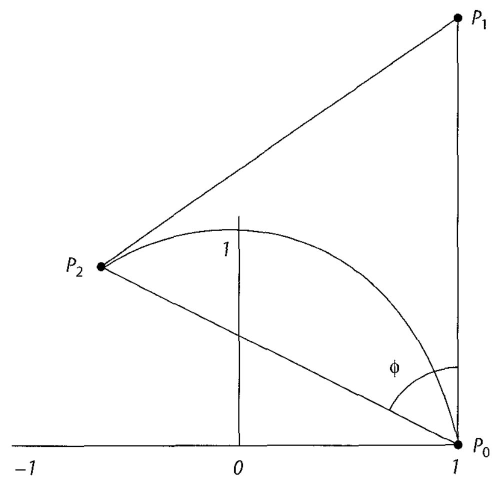
For geometric modeling, the circular arc is undoubtedly the most useful particular case of conic sections. Such an arc results when P_0, P_1, P_2 form an isosceles triangle and when $w = \cos\phi$ (where ϕ is the angle between P_0P_1 and P_0P_2).[3,4,6] For instance, a circular arc of length 2ϕ starting from the trigonometric origin (see Figure 2) is obtained with the following control points:

$$\begin{aligned} &P_0(1,0) \\ &P_1(1,\tan\phi) \\ &P_2(\cos 2\phi, \sin 2\phi) \end{aligned} \tag{4}$$

which can be computed more efficiently by

$$\begin{aligned} &P_0(1,0) \\ &P_1(1,T) \\ &P_2\left(\frac{1-T^2}{1+T^2}, \frac{2T}{1+T^2}\right) \end{aligned} \tag{5}$$

where $T = \tan\phi$.



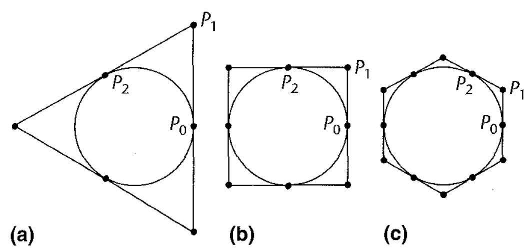
2 **Circular arc.**

Circles and spheres

Equation 5 gives us a control triangle $P_0P_1P_2$ that only permits the definition of a circular arc sweeping less than 180 degrees (T becomes infinite for $2\phi = \pi$). Therefore, at least three circular arcs are needed to obtain a full circle. These arcs are pieced together (the last point of each arc becomes the first point of the next one), giving a circle defined by n arcs and $2n+1$ control points. The resulting curve can be considered either a piecewise rational Bezier or a NURBS. Arcs of different lengths may be pieced together, but concatenating similar arcs provides a much better parametrization.

Several authors have proposed representations for circular arcs sweeping over 180 degrees by using infinite control points or negative weights.[6,8] Such constructions are less relevant to CAD, because they lose the convex hull property and give bad parametrizations. The reparametrization process we propose, on the other hand, may eventually be employed to improve the quality of this parametrization.

Among all the possible representations, the following ones appear particularly interesting.



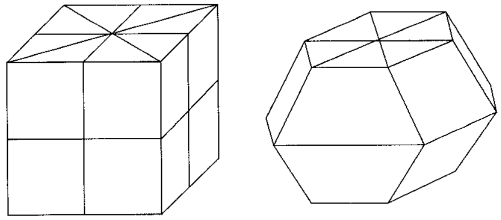
3 **Three different control lattices defining a circle.**

Triangle-based circle. Versprille[4] proposed the first exact representation of a circle by a NURBS. In that model, the circle is composed of three arcs (each of length $2\pi/3$) and defined by six control points (one point being repeated) regularly placed on an equilateral triangle (see Figure 3a):

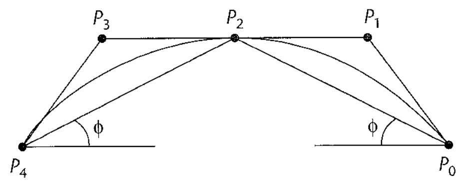
$$\begin{aligned} &P_0(1,0) \\ &P_1(1,\sqrt{3}) \\ &P_2\left(\frac{-1}{2}, \frac{\sqrt{3}}{2}\right) \\ &P_3(-2,0) \end{aligned} \qquad \begin{aligned} &P_4\left(\frac{-1}{2}, \frac{-\sqrt{3}}{2}\right) \\ &P_5(1,-\sqrt{3}) \\ &P_6(1,0) \end{aligned}$$

where even-indexed (or, respectively, odd-indexed) points have their weights set to 1 (respectively, 1/2).

Square-based circle. The main drawback of the triangle-based circle is that the size of the control polygon is relatively large compared to the circle. An improvement proposed by Tiller[9] defines the circle by four arcs (each of length $\pi/2$) using eight control points (one repeated) placed on a square (see Figure 3b):



4 Two different control lattices defining a sphere.



5 Junction of two circular arcs.

$$
\begin{array}{ll}
P_0(1,0) & P_5(-1,-1) \\
P_1(1,1) & P_6(0,-1) \\
P_2(0,1) & P_7(1,-1) \\
P_3(-1,1) & P_8(1,0) \\
P_4(-1,0) &
\end{array}
$$

where even-indexed (respectively, odd-indexed) points have their weights set to 1 (respectively, $\sqrt{2}/2$).

Hexagon-based circle. The square-based circle is by far the most commonly used in modeling software, mainly because of its simplicity. Nevertheless, another simple representation provides a tighter convex hull as well as a better parametrization. In this model, the circle is composed of six arcs (each of length $\pi/3$) and defined by 12 control points (one repeated) on a hexagon (see Figure 3c):

$$
\begin{array}{ll}
P_0(1,0) & P_4\left(\frac{-1}{2},\frac{\sqrt{3}}{2}\right) \\
P_1\left(1,\frac{\sqrt{3}}{3}\right) & P_5\left(-1,\frac{\sqrt{3}}{3}\right) \\
P_2\left(\frac{1}{2},\frac{\sqrt{3}}{2}\right) & \ldots \\
P_3\left(0,\frac{2\sqrt{3}}{2}\right) & P_{12}(1,0)
\end{array}
$$

where even-indexed (respectively, odd-indexed) points have their weights set to 1 (respectively, $\sqrt{3}/2$).

Any regular polygon can be used to define the circle, but the square and the hexagon provide the best trade-off among cost, accuracy, and ease of use.

Sphere. Starting from a NURBS-based representation of a circle, we can easily obtain a NURBS-based representation of a sphere by computing a tensor product of a set of circles (providing the parallels) by a set of half-circles (providing the meridians). The most popular representation, proposed by Tiller,[9] is based on the circumscribing cube (defined with 26 control points, five of them repeated). We can also imagine a tighter control lattice having a kind of hexahedrical shape (defined with 62 control points, seven of them repeated), as shown in Figure 4.

Study of the continuity

Since the circle results from concatenation of several circular arcs, the question of continuity at the junction points (*knots* in B-spline terminology) arises naturally. To study this problem,[10] let us take the case where two arcs, defined by the control triangles $P_0P_1P_2$ and $P_2P_3P_4$, respectively (see Figure 5), are connected. When the arcs have the same length, we necessarily have $w_0 = w_2 = w_4 = 1$ and $w_1 = w_3 = w$. Moreover, let us suppose that the parameter range is [0,1] for the first segment and [1,2] for the second one. If we use the 1^- and 1^+ notation to express the left and right limits for the curve at the parameter value $t = 1$, Equation 3 gives

$$
\begin{aligned}
C(1^-) &= P_2 \\
C'(1^-) &= 2w\,(P_2 - P_1) \\
C''(1^-) &= 2\,(P_0 - P_2) + 4w\,(1-w)\,(P_1 - P_2) \\
C(1^+) &= P_2 \\
C'(1^+) &= 2w\,(P_3 - P_2) \\
C''(1^+) &= 2\,(P_4 - P_2) + 4w\,(1-w)\,(P_3 - P_2)
\end{aligned}
\tag{6}
$$

Notice that even if the curve $C(t)$ and its derivatives are rational in the general case, they become polynomial at the parameter value $t = 1$ because the denominator vanishes.

Equation 6 shows that at least a C^0 continuity exists at point P_2. Moreover, since the two arcs have the same length, triangles $P_0P_1P_2$ and $P_2P_3P_4$ are similar, which means, in particular, that

$$P_2 - P_1 = P_3 - P_2 \tag{7}$$

The junction therefore also has a C^1 continuity. The similarity of triangles $P_0P_1P_2$ and $P_2P_3P_4$, as shown in Figure 5, also provides

$$
\begin{aligned}
P_4 - P_2 &= P_0 - P_2 + 2(1+\cos 2\phi)\,(P_2 - P_1) \\
&= P_0 - P_2 + 4w^2\,(P_2 - P_1)
\end{aligned}
\tag{8}
$$

The substitution of Equation 8 in the expression of $C''(1^+)$ shows that we can only get $C''(1^-) = C''(1^+)$ when $w = 0$ (a 180-degree arc, which is forbidden), or $w = 1$ (a 0-degree arc, which is not of interest). In other words, we cannot define a circle with a C^2 continuity using a quadratic NURBS (or a quadratic piecewise rational Bezier).[6,7] Note that $w = 0$ is allowed when considering infinite points, but in that case Equation 6 is not valid and there is still a C^1 continuity only (see Piegl and Tiller[6] for further details).

Note also that the C^1 continuity is only obtained in the Euclidian space (that is, after projection). In the associated homogeneous space (before projection), the continuity is not better than C^0 (see Piegl and Tiller[6]).

Study of the parametrization

Let us take again the circular arc of length 2ϕ starting from the trigonometric origin, illustrated in Figure 2. For many applications, the best parametrization of a curve is the *arc length parametrization* (also called uniform parametrization or constant-speed parametrization) because the variation of this parameter represents the exact distance on the curve from one point to another. The most obvious need for arc length parametrization is to provide constant-speed movements in animation sequences. Such a parametrization is also useful for surface design or surface rendering; for instance, arc length parametrization minimizes distortion when doing texture mapping.

In our example, this arc length parametrization is given by the trigonometric parametrization:

$$\forall \theta \in [0, 2\phi] \qquad C(\theta) = (\cos\theta, \sin\theta)$$

Combining Equation 3 and Equation 5 gives us the rational Bezier parametrization of our arc:

$$\forall t \in [0,1] \qquad C(t) = \left(\frac{1-2(1-w)t - 2(1-w)t^2}{1-2(1-w)t + 2(1-w)t^2}, \frac{2Twt - 2Tw(1-w)t^2}{1-2(1-w)t + 2(1-w)t^2} \right) \tag{9}$$

We can evaluate the quality of this parametrization by comparing it with the arc length parametrization. In the ideal case, the two parametrizations are related by a linear function:

$$\forall t \in [0,1] \qquad \theta(t) = 2\phi t$$

Obviously, this is not the case here because it would mean that the cosine function could be expressed as a rational polynomial. When this linear relationship is not fulfilled, we obtain a quantitative evaluation of the parametrization by computing the distance between the actual function $\theta(t)$ and the ideal one $2\phi t$. Using a Euclidian norm, this distance $\Delta(t)$ (sometimes called *chordal deviation*) is given by

$$\Delta(t) = \sqrt{\frac{\left(\theta(t) - 2\phi t\right)^2}{1 + 4\phi^2}} \tag{10}$$

Consequently, to evaluate our parametrization, we have to find the actual expression of $\theta(t)$. For that, we involve another classical parametrization of the circular arc, the *half-tangent parametrization*:

$$\forall s \in [0,1] \qquad C(s) = \left(\frac{1 - T^2 s^2}{1 + T^2 s^2}, \frac{2Ts}{1 + T^2 s^2} \right) \quad \text{where } T = \tan\phi \tag{11}$$

The main advantage of this parametrization is that it provides a simple relationship between the arc length θ and the parameter s:

$$\forall s \in [0,1] \qquad \theta(s) = 2 \arctan Ts$$

Therefore, if we can find $s(t)$, we will get immediately $\theta(t)$. Fortunately, a fundamental result of projective geometry, developed by Moebius, states that two parametrizations of the same curve by a rational polynomial of the same degree (Equation 9 and Equation 11) are related by a rational linear function:

$$\exists (a,b,c,d) \in \mathrm{IR}^4 \ / \ s(t) = \frac{a + bt}{c + dt} \tag{12}$$

Lee and Lucian[11] recently extended this result to piecewise rational polynomial curves such as NURBS. In our case, parameters t and s are equal at the boundaries of the range $[0,1]$. This induces a more precise expression:

$$\exists p \in \mathrm{IR} \ / \ s(t) = \frac{(1+p)t}{1 + pt} \tag{13}$$

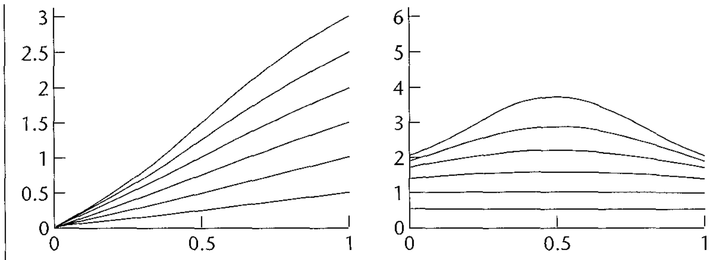
The substitution of Equation 13 in Equation 11 gives $p = w - 1$. Therefore,

$$s(t) = \frac{wt}{1 - t + wt} \tag{14}$$

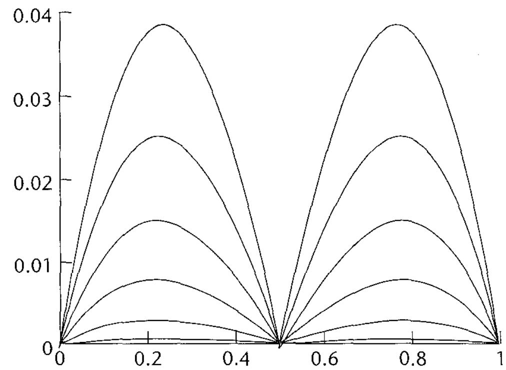
and, finally,

$$\theta(t) = 2 \arctan \frac{t\sqrt{1 - w^2}}{1 - t + wt} \tag{15}$$

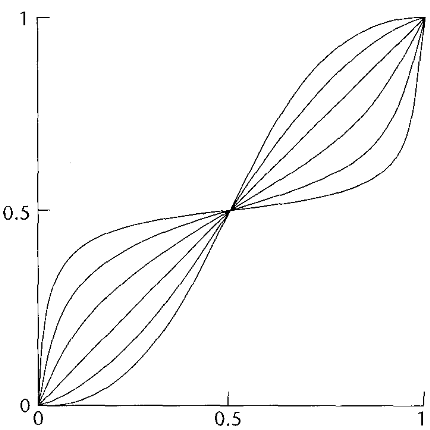
The plot of this function, for several values of ϕ, appears on the left side of Figure 6 (next page). Note that the function becomes less straight (in fact, it oscillates around the ideal straight line) when ϕ increases. This phenomenon is more apparent on the right side of the figure, which shows the deviations of $\theta(t)$ from the ideal horizontal lines. But the best quantitative information is given by the chordal deviation $\Delta(t)$ plotted in Figure 7 (next page). This means that except for very small angles, the representation of a circular arc by a quadratic NURBS (or a quadratic piece-wise rational Bezier) involves a parametrization that deviates significantly from the arc length.[12]



6 θ(t) and θ′(t) for ϕ = 0.25, 0.5, 0.75, and 1 (in radians).



7 Δ(t) for ϕ = 0.25, 0.5, 0.75, and 1 (in radians).



8 Zigzag reparametrization function S(t) for p = 0, 0.5, 1, 2, 6, and 16.

Zigzag reparametrization

Our discussion has shown that the usual representation of circles and spheres by NURBS curves or surfaces suffers from two main weaknesses. First, the continuity of the parametrization is C^1 at best. Second, the parametrization can deviate significantly from the arc length.

In fact, these two weak points have the same origin: Because the degree of the curve is only quadratic, there are not enough degrees of freedom to correct one (or both) faults. We might therefore raise the degree to get more parameters to manipulate. Unfortunately, the classical process of raising the degree of a Bezier or a B-spline without modifying its shape[7,8,13] does not work for this particular goal because it does not change the parametrization. For that reason, we propose here a technique (restricted to rational or piece-wise rational curves) that enables degree raising while controlling precisely the resulting parametrization.

Principle

We borrowed our technique's general principle from the reparametrization scheme defined by Moebius. Equation 12 showed that the reparametrization of a rational curve by a linear rational polynomial modifies neither its shape nor its degree. A corollary of this result is that a reparametrization by a quadratic rational polynomial doubles the degree of the initial curve, a cubic rational polynomial triples the degree, and so on. Therefore, the conic defined by Equation 3, for instance, can be represented by a quartic rational curve when we replace t by $S(t)$ where S is a quadratic rational polynomial.

In the most general formulation for $S(t)$, there are six degrees of freedom:

$$S(t) = \frac{a + bt + ct^2}{d + et + ft^2}$$

but several characteristics appear desirable for this reparametrization. For instance, it is important for the two parametrizations to have the same domain of variation ($S(0) = 0$ and $S(1) = 1$). Moreover, due to the isoscelism of the control triangle, the symmetry of the parametrization should be conserved ($S(1-t) = 1 - S(t)$ or in other words $S'(1-t) = S'(t)$). All these constraints imply that there remains only one degree of freedom. If we call this parameter p and define it as the value of the derivative at the boundaries ($S'(0) = S'(1) = p$), we obtain

$$S(t) = \frac{pt + (1-p)t^2}{1 - 2(1-p)t + 2(1-p)t^2} \tag{16}$$

The shape of this function (see Figure 8) inspired the name of our reparametrization technique: *zigzag reparametrization*. Notice that p has to be positive (or eventually null) in order to produce an increasing function in the range [0,1].

Zigzag reparametrization of a conic

If we apply our zigzag reparametrization on the Bezier curve defined by Equation 3, we obtain

$$C(t) = \frac{a^2 P_0 + 2wab\, P_1 + b^2 P_2}{a^2 + 2wab + b^2} \tag{17}$$

with $a = 1-(2-p)t + (1-p)t^2$ and $b = pt + (1-p)t^2$. The result is a quartic rational curve, but it is not a rational Bezier since it is no longer written in terms of Bernstein polynomials. But because the Bernstein polynomials form a basis for the polynomial space, we are assured that there exists a Bezier curve equivalent to Equation 17. To find this quartic rational Bezier curve, we have to find five points (Q_0, Q_1, Q_2, Q_3, Q_4) and five weights (w_0, w_1, w_2, w_3, w_4) that obey Equation 18 (Figure 9).

$$C(t)=\frac{w_0(1-t)^4 Q_0+4w_1 t(1-t)^3 Q_1+6w_2 t^2(1-t)^2 Q_2+4w_3 t^3(1-t) Q_3+w_4 t^4 Q_4}{w_0(1-t)^4+4w_1 t(1-t)^3+6w_2 t^2(1-t)^2+4w_3 t^3(1-t)+w_4 t^4}$$

9 Equation 18.

Obviously, $Q_0 = P_0$ and $Q_4 = P_2$ because of the interpolation of the boundaries. We also know that we can let $w_0 = 1$ and $w_4 = 1$ for symmetry reasons, without loss of generality. We obtain the remaining weights w_1, w_2, and w_3 by comparing the denominators of Equation 18 (see Figure 9) and Equation 17 term by term. Similarly, we obtain the remaining points Q_1, Q_2, and Q_3 by comparing the numerators.

Finally, the zigzag reparametrization of Equation 17 provides a quartic rational Bezier (and therefore quartic NURBS) defined by

$$\begin{aligned}
Q_0 &= P_0\\
Q_1 &= \frac{P_0+wP_1}{1+w}\\
Q_2 &= \frac{p^2P_0+2w\left(1+p^2\right)P_1+p^2P_2}{2\left(p^2+p^2w+w\right)}\\
Q_3 &= \frac{P_2+wP_1}{1+w}\\
Q_4 &= P_2\\
w_0 &= 1\\
w_1 &= \frac{1+w}{2}p\\
w_2 &= \frac{p^2+p^2w+w}{3}\\
w_3 &= \frac{1+w}{2}p\\
w_4 &= 1
\end{aligned} \tag{19}$$

Four important observations emerge from this formulation:

- Only the position of point Q_2 depends on the value of parameter p. A nice consequence of this is that the zigzag reparametrization has a very simple geometric interpretation: changing the value of p consists in moving Q_2 along a straight line (more precisely, a half line because we force $p \geq 0$). The simultaneous modification of w_1, w_2, and w_3 keeps the curve unchanged (except for its parametrization) despite the displacement of Q_2.
- When $p = 1$, we obtain exactly the points and weights provided by the classical degree raising technique mentioned above. In other words, it means that zigzag reparametrization is a kind of generalization of this process, which provides an additional degree of freedom.
- The principle of the zigzag reparametrization can be extended to nonsymmetric reparametrization by permitting two degrees of freedom ($p = S'(0)$ and $q = S'(1)$) for Equation 16. We can also use such an extension for the reparametrization of nonsymmetric curves such as ellipses or parabolas.
- Zigzag reparametrization can also be extended above degree 2. We do this by defining a family of rational polynomials where each member $S_k(t)$ of degree k fulfills

$$\begin{aligned}
&S_k(0)=0\\
&S_k(1)=1\\
&\forall i=0..k\\
&\exists p_i \in \mathrm{IR}^+ / \frac{d^iS}{dt^i}(0)=p_i\\
&\frac{d^iS}{dt^i}(1)=(-1)^{i-1}p_i
\end{aligned}$$

Therefore, each successive member of this *zigzag polynomial family* includes one additional parameter that enables more precise control of the parametrization.

Next, we show how to use zigzag reparametrization to improve the parametrization of a circle represented by a NURBS. We provide these examples only as illustration. The new technique may in fact be employed in many applications that involve reparametrization according to a given criterion.

A C^2 parametrization of the circle

We first apply zigzag parametrization to obtain a circle with a C^2 parametrization. Starting again from the configuration of control points illustrated in Figure 5, we can study the continuity at point P^2:

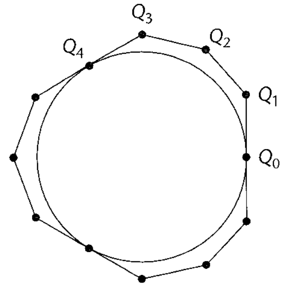
$$\begin{aligned}
C'(1^-) &= 2wp(P_2-P_1)\\
C''(1^-) &= 2p^2(P_0-P_2)\\
&\quad +4w\left(1+p-p^2-2wp^2\right)(P_1-P_2)\\
C'(1^+) &= 2wp(P_3-P_2)\\
C''(1^+) &= 2p^2(P_4-P_2)\\
&\quad +4w\left(1+p-p^2-2wp^2\right)(P_3-P_2)
\end{aligned} \tag{20}$$

As previously, Equation 7 yields C^1 continuity. To check the C^2 continuity, we have to substitute Equation 7 and Equation 8 in the expression of $C''(1^+)$:

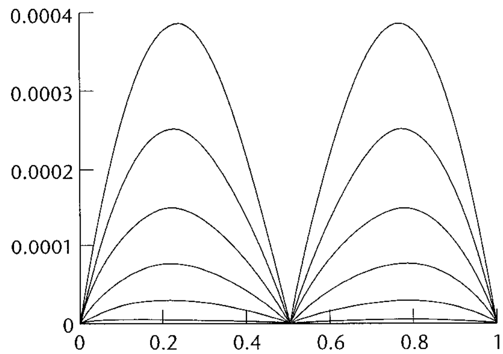
$$C''(1^+)=2p^2(P_0-P_2)+4w\left(p^2-p-1\right)(P_1-P_2)$$

Finally, comparing $C''(1^-)$ and $C''(1^+)$ provides

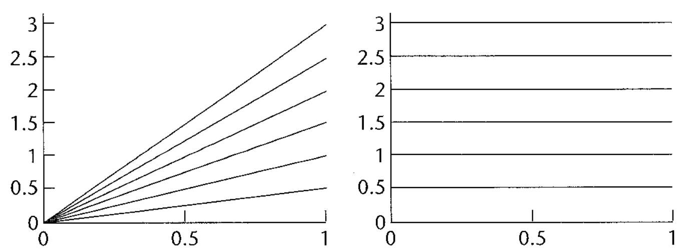
$$p^2(1+w)-p-1=0 \tag{21}$$



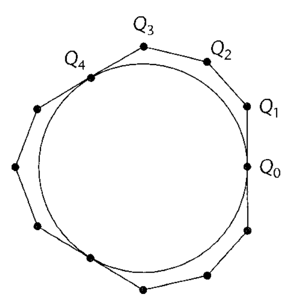
10 C^2 reparametrization of the triangle-based circle.



11 $\Delta(t)$ for ϕ = 0.25, 0.5, 0.75, and 1 (in radians).



12 $\theta(t)$ and $\theta'(t)$ for ϕ = 0.25, 0.5, 0.75, 1, 1.25, and 1.5 (in radians).



13 Quasi-uniform reparametrization of the triangle-based circle.

This quadratic equation has always two solutions ($\Delta = 4w+5 > 0$), but only the positive solution is useful for our reparametrization:

$$p = \frac{1+\sqrt{5+4w}}{2(1+w)} \tag{22}$$

Therefore, Equation 22 combined with Equation 19 enables a C^2 continuous junction for circular arcs of the same length, with the resulting curve being represented either by quartic rational Bezier or a quartic NURBS. To illustrate this, Figure 10 shows a circle with a C^2 parametrization obtained by zigzag reparametrization of the triangle-based circle, using for p the value provided by Equation 22.

Quasi-uniform circle parametrization

A second application of the zigzag reparametrization is to obtain a circle with a *quasi-uniform* parametrization, as close as possible to the trigonometric parametrization. As we have seen, the parametrization error can be quantified by computing the chordal deviation $\Delta(t)$ between the arc length parametrization and the actual one. Substituting Equation 16 in Equation 10 expresses the chordal deviation for a circular arc on which we have applied the zigzag reparametrization. The formulation of this new chordal deviation is a function of parameter p:

$$\Delta(t) = \frac{2\arctan\dfrac{\sqrt{1-w^2}\left(pt+(1-p)t^2\right)}{1-2t+(1+w)\left(pt+(1-p)t^2\right)} - 2\phi t}{\sqrt{1+4\phi^2}} \tag{23}$$

What we have here is a classical optimization problem: Find the optimal value of a parameter (in our case, the reparametrization factor p) that minimizes a given condition (in our case, the chordal deviation $\Delta(t)$). One possibility is to minimize the L^2 norm of $\Delta(t)$ by employing a least-squares minimization method. The drawback of this technique (and all other techniques of the same family) is that it provides only a numerical (and not an analytical) solution. Thus, for each new angle ϕ needed, the user has to restart the whole least-squares process to find the new parameter p.

Another possibility is to minimize the L^∞ norm of $\Delta(t)$, which only requires computing the maximal chordal deviation. This maximum has an analytical expression that can be computed by symbolic calculation software such as Maple or Mathematica, but this expression is so complex as to be useless in practice.

For these reasons, we propose here a third solution that is only heuristic but, in fact, very close to the optimum. If we study the variation of the maximal chordal deviation for different values of ϕ, we can see that this maximum always results for a value t that belongs in [0.19, 0.24]. If we take a value in that range ($t = 1/5$ seems to work well), we are assured that the chordal deviation at point $\Delta(1/5)$ will not be too far from the maximal deviation. Therefore, we simply have to force $\Delta(1/5) = 0$, that is,

$$\frac{(1+4p)\sin\phi}{18+12p-(1+4p)\cos\phi} - \tan(\phi/5) = 0 \tag{24}$$

to get a quasi-optimal value for p:

$$p = \frac{4 - 2\cos^3(\phi/5) + \cos(\phi/5)}{-1 + 8\cos^3(\phi/5) - 4\cos(\phi/5)} \qquad \textbf{(25)}$$

Figure 11 shows the plot of the chordal deviation (t) obtained with the quasi-optimal value p. Notice that there is a factor of 100 in the scale between Figure 7 and Figure 11, meaning that the parametrization error has been reduced more than 100 times. Figure 12 confirms this result, showing the variations of $\theta(t)$ and $\theta'(t)$ for different values of ϕ when using the quasi-optimal parameter: The resulting curves are almost straight, and no more oscillations are visible.

Figure 13 illustrates the circle with a quasi-uniform parametrization obtained by zigzag reparametrization of the triangle-based circle, using the parameter p provided by Equation 25. Comparing Figure 10 and Figure 13 shows that the C^2 continuous circle and the quasi-uniform circle are almost identical. In fact, the difference between the reparametrizations is so small ($p \simeq 1.21525$ for the first and $p \simeq 1.20669$ for the second) that the variation of the position of the control points is hardly visible (remember that only Q_2, Q_6, and Q_{10} have actually moved).

Table 1 presents all the numerical results relative to zigzag reparametrization of the triangle-based, square-based, and hexagon-based circles. Notice that the chordal deviation can be reduced more than 1,000 times by the reparametrization. Notice also that the analytic L^∞ minimization offers only an average of 5 percent additional precision compared to our heuristic minimization, for a much more expensive computation cost. This confirms the nice behavior of the "$\Delta(1/5) = 0$" heuristic.

To conclude, we would like to note that we have chosen to use the circle here only as an illustrating example. The idea of zigzag reparametrization is much more general and we feel that it could benefit many situations where well-behaving parametrization is important. ■

Table 1. Summary of numerical results.

Type of Representation	Triangle-based	Square-based	Hexagon-based
Original parametrization			
Chordal deviation	$1.68E^{-2}$	$8.61E^{-3}$	$3.23E^{-3}$
C^2 reparametrization			
Continuity factor p	1.21525	1.11239	1.04750
Analytic reparametrization			
Optimal factor $\bar{p}$	1.20675	1.11000	1.047063
Chordal deviation $\bar{\Delta}$	$6.11E^{-5}$	$1.82E^{-5}$	$3.04E^{-6}$
Error reduction factor $\Delta/\bar{\Delta}$	275	473	1062
Heuristic reparametrization			
Quasi-optimal factor $\hat{p}$	1.20669	1.10998	1.047059
Chordal deviation $\hat{\Delta}$	$6.38E^{-5}$	$1.89E^{-5}$	$3.20E^{-6}$
Error reduction factor $\Delta/\hat{\Delta}$	263	456	1009

References

1. L. Piegl and W. Tiller, *The Book of NURBS*, Springer Verlag, New York, 1995.
2. G. Farin, *NURB Curves and Surfaces*, A.K. Peters, Boston, 1995.
3. A. Forrest, *Curves and Surfaces for Computer-Aided Design*, doctoral dissertation, Cambridge University, Cambridge, U.K., 1968.
4. K. Versprille, *Computer Aided Design Applications of the Rational B-Spline Approximation Form*, doctoral dissertation, University of Syracuse, New York, 1975.
5. G. Farin, "From Conics to NURBS: A Tutorial and Survey," *IEEE Computer Graphics & Applications*, Vol. 12, No. 5, 1989, pp. 78-86.
6. L. Piegl and W. Tiller, "A Menagerie of Rational B-Spline Circles," *IEEE CG&A*, Vol. 9, No. 5, Sept. 1989, pp. 48-56.
7. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1990.
8. L. Piegl, "Infinite Control Points: A Method for Representing Surfaces of Revolution Using Boundary Data," *IEEE CG&A*, Vol. 7, No. 3, May 1989, pp. 45-55.
9. W. Tiller, "Rational B-Splines for Curve and Surface Representation," *IEEE CG&A*, Vol. 3, No. 6, Sept. 1983, pp. 61-69.
10. C. Blanc, *Modélisation et Déformation des Surfaces pour la Synthèse d'Images*, doctoral dissertation (in French), Université de Bordeaux, Bordeaux, France, 1994.
11. E. Lee and M. Lucian, "Moebius Reparametrization of Rational B-Splines," *Computer Aided Geometric Design*, Vol. 8, 1991, pp. 213-238.
12. R. Farouki and T. Sakkalis, "Real Rational Curves are not Unit Speed," *Computer Aided Geometric Design*, Vol. 8, No. 2, 1991, pp. 151-158.
13. R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Computer Graphics and Geometric Modeling*, Morgan Kaufmann, San Francisco, 1987.

***Carole Blanc** is a lecturer at the University of Bordeaux, France, where she received a PhD in Computer Science in 1994. Her research focuses on geometric modeling, including parametric surfaces, implicit surfaces, and deformation techniques.*

***Christophe Schlick** is an assistant professor at the University of Bordeaux, where he obtained a PhD in Computer Science in 1992. His research interests include illumination models, Monte Carlo techniques, procedural textures, and spline curves and surfaces.*

Contact the authors at the Laboratoire Bordelais de Recherce en Informatique, University of Bordeaux, 351 Cours de la Liberation, 33405 Talence, France, e-mail {blanc, schlick}@labri.u-bordeaux.fr.