Accepted Manuscript

Constrained polynomial approximation of rational Bézier curves using reparameterization

Qianqian Hu, Huixia Xu

PII: S0377-0427(13)00092-7
DOI: 10.1016/j.cam.2013.02.022
Reference: CAM 9047

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 7 June 2012
Revised date: 12 February 2013

Constrained polynomial approximation of rational Bézier curves using reparameterization

Qianqian Hu [a,*], Huixia Xu [b]

[a] *Department of Mathematics, Zhejiang Gongshang University, Hangzhou 310018, PR China*

[b] *Institute of Mathematics, Zhejiang Wanli University, Ningbo 315100, PR China*

Abstract: This paper proposes a novel method for polynomial approximation of rational Bézier curves with constraints. Different from the previous techniques, for a given rational Bézier curve $\mathbf{r}(t)$, a polynomial curve $\mathbf{q}(s)$ with a parameter transformation $s = \varphi(t)$, such that $\mathbf{q}(\varphi(t))$ is the closet point to the point $\mathbf{r}(t)$, is considered to approximate it. To minimize the distance between these two curves in the L_2 norm produces a similar effect as that of the Hausdorff distance. We use a rational function $s(t)$ of a Möbius parameter transformation to approximate the function $\varphi(t)$. The method can preserve parametric continuity or geometric continuity of any $u, v (u, v \geq 0)$ orders at two endpoints, respectively. And applying the least squares method, we deduce a matrix-based representation of the control points of the approximation curve. Finally, numerical examples show that the reparameterization-based method is feasible and effective, and has a smaller approximation error under the Hausdorff distance than the previous methods.

Keywords: rational Bézier curves, polynomial approximation, Möbius parameter transformation, reparameterization, the least squares method

1. Introduction

Rational Bézier curves have been widely used for geometric modeling in computer aided geometric design (CAGD). It realizes the uniform representation of conic sections and polynomial parametric curves. Rational Bézier curves not only enjoy all the properties that their nonrational counterparts possess, but also can be modified their shapes by changing their weights used as shape parameters [1]. However, as applications grow in size, people often meet with some inconvenience in directly processing rational Bézier curves. First of all, with the rapid expansion and development of digital technology, data exchange and transmission between different modeling systems are getting more and more frequently. Then conversion between rational curves and polynomial curves is absolutely necessary. Secondly, differential and integral operations of rational curves are very complicated, even cannot be calculated. And it is not easy to obtain geometric information of rational curves, for example, curvature, torsion, volume, and so on. Therefore, approximation of rational curves by polynomial curves is an important and fundamental task in CAD/CAM.

In 1991, Sederberg and Kakimoto [2] first proposed the famous hybrid approximation algorithm for polynomial approximation of rational curves, and its convergence condition [3], bound estimation on the moving control point [4] and the convergence condition for hybrid polynomial approximation to higher derivatives of rational curves [5,6] were studied. Then in the past twenty years, many techniques for

[*] Corresponding author; email address: qianqianhu@hotmail.com

approximating rational curves by polynomial curves have been developed [7-9,12,13]. Floater [7] constructed polynomial curves to interpolate rational curves in a Hermite sense with high order approximation. Huang et al [8] presented a very simple method for approximating a rational Bézier curve by Bézier curves with control points obtained by successive degree elevation. Lu [9] presented an iteration method for polynomial approximation of rational Bézier curves based on the weighted progressive iteration approximation (abbr. WPIA) property [10]. Recently, Lewanowicz and Woźny applied the dual constrained Bernstein basis polynomials [11] to derive a polynomial curve approximating a rational Bézier curve with endpoints constraints in the L_2 norm [12]. The same approximation curve is also obtained by Cai and Wang using the least squares method [13]. The approximation methods proposed in [12] and [13] are mathematically equivalent. However, it should be pointed out that they are based on completely different ideas, and the numerical algorithms implementing them differ substantially. From the point of view of the computational cost of the algorithm, the method in [12] is more efficient than that in [13].

In all of the abovementioned methods, the distance metric between two parametric curves is the Euclidean distance (L_2 metric), the Manhattan distance (L_1 metric), or the Chebyshev distance (L_∞ metric). As we all know, the most appropriate metric for the curves in geometrical terms would be the Hausdorff distance [14]. For example, for a parametric curve with different parametric representations, the L_p (p=1,2,∞) distance between these curves is nonzero, but the Hausdorff distance is zero. However, it is very difficult to accurately calculate the Hausdorff distance for parametric curves. Therefore, similarly to [15], we use a polynomial curve after a parameter transformation $s = \varphi(t)$ to approximate the rational curve. Here the approximation curve at the parameter $\varphi(t_0)$ is the closest point to the given curve at t_0 for $\forall t_0 \in [0,1]$. Then a new distance function is defined as the Euclidean distance between the given curve and the approximation curve after reparameterization, which produces a similar effect as the Hausdorff distance [15]. The function $\varphi(t)$ may be very complicated and it is almost impossible to calculate the minimum of the new distance function. However, it is interesting to note that the control points and the shape of a rational Bézier curve after a Möbius parameter transformation remain unchanged, while only the weights change. So we want to find a rational function of a Möbius parameter transformation to approximate the function $\varphi(t)$. The main advantage lies in that the method can preserve parametric continuity or geometric continuity of high order at two endpoints of the rational curve and the approximation curve, respectively. Using the least squares method, the optimal approximation curve with constraints under the new distance metric is obtained. The control points of the approximation curve can be very quickly computed by multiplying the column vector composed of the control points of the given curve by a matrix that is pre-calculated before processing approximation.

The rest of the paper is laid out as follows. We give a brief introduction to rational Bézier curves and define the new distance metric in Section 2. In Section 3, we describe the problem of constrained polynomial approximation of rational Bézier curves using reparameterizatrion. Then we propose an efficient algorithm for polynomial approximation with constraints recurring to the Möbius parameter transformation and the least squares method in Section 4. Numerical examples are presented in Section 5 to confirm the effectiveness of this algorithm. Finally, we conclude the paper with the description of future work in Section 6.

2. Preliminary

Definition 1. A rational Bézier curve of degree n is represented by [1]

$$\mathbf{r}(t) = \frac{\sum_{i=0}^{n} \omega_i \mathbf{r}_i B_i^n(t)}{\sum_{i=0}^{n} \omega_i B_i^n(t)}, \tag{1}$$

where $B_i^n(t) = \binom{n}{i}(1-t)^i t^{n-i}$, i=0,1,...,n are the Bernstein bases functions of degree n, $\mathbf{r}_i (i = 0,1,...,n)$ are the control points, and $\omega_i (i = 0,1,...,n)$ are the associated weights. To guarantee every control point is valid [1], we provide that all the weights are positive.

For the rational Bézier curve $\mathbf{r}(t)$ of degree n expressed as (1), do a Möbius parameter transformation

$$t(s) = \frac{\lambda s}{\lambda s + (1-s)}, \tag{2}$$

where $\lambda > 0$ is a parameter. Then $\mathbf{r}(t)$ can be rewritten with the parameter s as

$$\mathbf{r}(t(s)) = \frac{\sum_{i=0}^{n} \lambda^i \omega_i \mathbf{r}_i B_i^n(s)}{\sum_{i=0}^{n} \lambda^i \omega_i B_i^n(s)}. \tag{3}$$

Here the control points and the shape of the rational curve (1) remain unchanged, while the weights become $\tilde{\omega}_i = \lambda^i \omega_i$, $(i = 0,...,n)$.

In most previous methods [7-9,11,13], a polynomial Bézier curve $\mathbf{q}(t)$ with the parameter t is found to approximate the rational curve $\mathbf{r}(t)$. And the distance function between the two curves in the L_2 norm is often defined by

$$d(\mathbf{r},\mathbf{q}) = \sqrt{\int_0^1 \left\|\mathbf{r}(t) - \mathbf{q}(t)\right\|^2 \mathrm{d}t}.$$

In this paper, we discuss the same new distance function in the L_2 norm as in [15]

$$d_\varphi(\mathbf{r},\mathbf{q}) = \sqrt{\int_0^1 \left\|\mathbf{r}(t) - \mathbf{q}(\varphi(t))\right\|^2 \mathrm{d}t},$$

where $\varphi(t)\colon [0,1]\to[0,1]$ is a strictly increasing continuous function such that $\mathbf{q}(\varphi(t_0))$ is the closest point to the point $\mathbf{r}(t_0)$ for $\forall t_0\in[0,1]$. To minimize $d_\varphi(\mathbf{r},\mathbf{q})$ produces a similar effect as that of the Hausdorff distance. Chen et al used a piecewise linear function to approximate the function $\varphi(t)$ [15]. Considering that the shape and control points of the rational curve $\mathbf{r}(t)$ keep unchanged and only the weights changed after the Möbius parameter transformation (2), we use the inverse function of $t(s)$ to approximate $\varphi(t)$. And the new distance function is estimated by

$$d_\lambda(\mathbf{r},\mathbf{q})=\sqrt{\int_0^1\left\|\mathbf{r}(t)-\mathbf{q}(s(t))\right\|^2\mathrm{d}t}, \tag{4}$$

where the parameter transformation

$$s(t)=\frac{t}{t+\lambda(1-t)} \tag{5}$$

is the inverse function of $t(s)$ (2). Obviously, $s(t)\colon[0,1]\to[0,1]$ is again a Möbius parameter transformation. It is a strictly increasing continuous function and satisfies $s(0)=0,\ s(1)=1$. When the parameter equals to 1, i.e., $\lambda=1$, the new distance function (4) returns to the conventional L_2-distance function.

Definition 2. We call two curves $\mathbf{r}(t)$ and $\mathbf{q}(s)$ have *geometric continuity* of $u,v\ (u,v\geq 0)$ orders at two endpoints respectively, if there exists a parameter transformation $s=f(t),\ f'(t)\big|_{t=0,1}\neq 0$, such that $\mathbf{r}(t)$ and $\mathbf{q}(f(t))$ satisfy the following equations

$$\begin{cases}\left.\dfrac{\mathrm{d}^i\mathbf{r}(t)}{\mathrm{d}t^i}\right|_{t=0}=\left.\dfrac{\mathrm{d}^i\mathbf{q}(f(t))}{\mathrm{d}t^i}\right|_{t=0}, & i=0,1,\ldots,u,\\[2ex] \left.\dfrac{\mathrm{d}^j\mathbf{r}(t)}{\mathrm{d}t^j}\right|_{t=1}=\left.\dfrac{\mathrm{d}^j\mathbf{q}(f(t))}{\mathrm{d}t^j}\right|_{t=1}, & j=0,1,\ldots,v.\end{cases}$$

And we call $\mathbf{r}(t)$ and $\mathbf{q}(s)$ preserve *parametric continuity* of $u,v\ (u,v\geq 0)$ orders at two endpoints, if $s=f(t)$ is an identical transformation, i.e., $f(t)=t$.

For the sake of brevity, we denote $G^{(u,v)}$-continuity by geometric continuity of $u,v\ (u,v\geq 0)$ orders at two endpoints, and denote $C^{(u,v)}$-continuity by parametric continuity of $u,v\ (u,v\geq 0)$ orders at two endpoints. Clearly, since $s'(t)\big|_{t=0}=\lambda^{-1}\neq 0,\ s'(t)\big|_{t=1}=\lambda\neq 0$, $\mathbf{r}(t)$ and $\mathbf{q}(s)$ satisfy $G^{(u,v)}$-continuity, if

$$\begin{cases} \left.\dfrac{\mathrm{d}^i\mathbf{r}(t)}{\mathrm{d}t^i}\right|_{t=0} = \left.\dfrac{\mathrm{d}^i\mathbf{q}(s(t))}{\mathrm{d}t^i}\right|_{t=0}, & i=0,1,\ldots,u, \\ \left.\dfrac{\mathrm{d}^j\mathbf{r}(t)}{\mathrm{d}t^j}\right|_{t=1} = \left.\dfrac{\mathrm{d}^j\mathbf{q}(s(t))}{\mathrm{d}t^j}\right|_{t=1}, & j=0,1,\ldots,v. \end{cases} \tag{6}$$

3. Problem Description

Given a rational Bézier curve $\mathbf{r}(t)$ of degree n expressed as (1), to find a Möbius parameter transformation $s = s(t)$ (5) and a Bézier curve $\mathbf{q}(s)$ of degree m expressed as

$$\mathbf{q}(s) = \sum_{i=0}^{m} \mathbf{q}_i B_i^m(s), \tag{7}$$

such that the similar Hausdorff distance function $d_\lambda(\mathbf{r},\mathbf{q})$ (4) reaches minimum. Meanwhile, $\mathbf{r}(t)$ and $\mathbf{q}(s)$ have parametric continuity or geometric continuity of u,v $(u,v \geq 0,\, u+v < m-1)$ orders at two endpoints, respectively.

Obviously, to minimize $d_\lambda(\mathbf{r},\mathbf{q})$ is very difficult, because the parameter of the approximation curve $\mathbf{q}(s(t))$ is t but not s. So the substitution rule for integrals is introduced to address this problem, and an alternative distance function

$$d_\lambda(\mathbf{r},\mathbf{q}) = \sqrt{\int_0^1 \left\|\mathbf{r}(t(s)) - \mathbf{q}(s)\right\|^2 \mathrm{d}t(s)}, \tag{8}$$

is used, where $t(s)$ shown as in (2) is the inverse function of $s(t)$. According to (2) and the definition of differential expression, we have

$$\mathrm{d}t(s) = \frac{\lambda}{(\lambda s + (1-s))^2}\,\mathrm{d}s.$$

According to integration by substitution and the abovementioned equation, (8) can be rewritten as

$$d_\lambda(\mathbf{r},\mathbf{q}) = \sqrt{\lambda \int_0^1 \left\|\frac{\mathbf{r}(t(s)) - \mathbf{q}(s)}{\lambda s + (1-s)}\right\|^2 \mathrm{d}s}. \tag{9}$$

For the sake of simplicity, we rewrite $\mathbf{q}(s)$ (7) in matrix form as

$$\mathbf{q}(s) = \mathbf{B}_m \mathbf{Q}_m, \tag{10}$$

where $\mathbf{B}_m = (B_0^m(s), B_1^m(s),..., B_m^m(s))$ and $\mathbf{Q}_m = (\mathbf{q}_0, \mathbf{q}_1,..., \mathbf{q}_m)^{\mathrm{T}}$, and also rewrite $\mathbf{r}(t(s))$ (3) in matrix form as

$$\mathbf{r}(t(s)) = \frac{\mathbf{B}_n \boldsymbol{\lambda} \mathbf{W} \mathbf{R}_n}{\mathbf{B}_n \boldsymbol{\lambda} \boldsymbol{\omega}}, \tag{11}$$

where $\mathbf{B}_n = (B_0^n(s), B_1^n(s),..., B_n^n(s))$, $\boldsymbol{\omega} = (\omega_0, \omega_1,..., \omega_n)^{\mathrm{T}}$, $\mathbf{R}_n = (\mathbf{r}_0, \mathbf{r}_1,..., \mathbf{r}_n)^{\mathrm{T}}$, $\boldsymbol{\lambda} = diag(1, \lambda,..., \lambda^n)$, and $\mathbf{W} = diag(\omega_0, \omega_1,..., \omega_n)$.

4. Polynomial approximation of rational curves

In order to distinguish the constrained and unknown control points, we divide the column vector $\mathbf{R}_n$ composed by the control points of $\mathbf{r}(t)$ into three parts, i.e.,

$$\mathbf{R}_n = \begin{pmatrix} \mathbf{R}_n^1 \\ \mathbf{R}_n^2 \\ \mathbf{R}_n^3 \end{pmatrix},$$

where $\mathbf{R}_n^1 = (\mathbf{r}_0, \mathbf{r}_1,..., \mathbf{r}_u)^{\mathrm{T}}$ and $\mathbf{R}_n^3 = (\mathbf{r}_{n-v},..., \mathbf{r}_{n-1}, \mathbf{r}_n)^{\mathrm{T}}$ are the constrained control points, and $\mathbf{R}_n^2 = (\mathbf{r}_{u+1}, \mathbf{r}_{u+2},..., \mathbf{r}_{n-v-1})^{\mathrm{T}}$ are the unknown ones of the curve $\mathbf{r}(t)$. And $\mathbf{Q}_m$ is handled in a similar fashion. To obtain the constrained approximation curve $\mathbf{q}(s)$ and the Möbius parameter transformation $s(t)$, first we present a necessary and sufficient (or sufficient) condition for the two curves preserving parametric continuity (or geometric continuity) of u,v orders at two endpoints respectively, and obtain the constrained control points of the approximation curve $\mathbf{q}(s)$; next, we use the golden section search method to find the optimal value of λ such that the similar Hausdorff distance function (9) reaches minimum and calculate the other unknown control points according to the least squares method.

4.1. Constrained control points of the approximation curve

It is very hard to calculate the constrained control points of $\mathbf{q}(s)$ directly from the geometric continuous condition (6). The reason is that the reparameterized approximation curve $\mathbf{q}(s(t))$ is a composite function, and the high order derivatives of the composite function are computationally difficult. The following lemma is introduced to tackle this issue.

Lemma 1. Given two curves $\mathbf{r}(t)$ (1) and $\mathbf{q}(s)$ (7) with the parameter function $s(t)$ (5), the curves preserve geometric continuity of u,v orders at two endpoints respectively, if the following equations

$$\begin{cases} \left.\dfrac{\mathrm{d}^i \mathbf{r}(t(s))}{\mathrm{d}s^i}\right|_{s=0} = \left.\dfrac{\mathrm{d}^i \mathbf{q}(s)}{\mathrm{d}s^i}\right|_{s=0}, \quad i = 0,1,\ldots,u, \\ \left.\dfrac{\mathrm{d}^j \mathbf{r}(t(s))}{\mathrm{d}s^j}\right|_{s=1} = \left.\dfrac{\mathrm{d}^j \mathbf{q}(s)}{\mathrm{d}s^j}\right|_{s=1}, \quad j = 0,1,\ldots,v. \end{cases} \tag{12}$$

are true. Here $t(s)$ is defined by (2).

Proof. To prove this lemma, we should just prove that (12) is equivalent to the geometric continuous condition (6). That is, given the equations (6), we will prove that the equations (12) hold. Here mathematical induction is used to prove it. According to (2) and (5), it is easy to know that

$$t(0) = 0,\ t(1) = 1,\ s(0) = 0,\ s(1) = 1,\ s'(0) = \lambda^{-1},\ s'(1) = \lambda. \tag{13}$$

We only prove the statement for the first equation in (12), and the second equation is treated analogously. When $u = 0$, the first equation is rewritten as $\mathbf{r}(t(0)) = \mathbf{q}(0)$. Obviously, according to (6) and (13), it is equivalent to $\mathbf{r}(0) = \mathbf{q}(s(0))$, and the statement holds for $u = 0$.

Assume that the statement holds for some unspecified value of $u-1$, that is, given

$$\left.\frac{\mathrm{d}^k \mathbf{r}(t)}{\mathrm{d}t^k}\right|_{t=0} = \left.\frac{\mathrm{d}^k \mathbf{q}(s(t))}{\mathrm{d}t^k}\right|_{t=0},\ k=0,1,\ldots,u-1.$$

we have

$$\left.\frac{\mathrm{d}^k \mathbf{r}(t(s))}{\mathrm{d}s^k}\right|_{s=0} = \left.\frac{\mathrm{d}^k \mathbf{q}(s)}{\mathrm{d}s^k}\right|_{s=0},\ k=0,1,\ldots,u-1. \tag{14}$$

Next, we will prove that there holds $\left.\frac{\mathrm{d}^u \mathbf{r}(t(s))}{\mathrm{d}s^u}\right|_{s=0} = \left.\frac{\mathrm{d}^u \mathbf{q}(s)}{\mathrm{d}s^u}\right|_{s=0}$, if $\left.\frac{\mathrm{d}^u \mathbf{r}(t)}{\mathrm{d}t^u}\right|_{t=0} = \left.\frac{\mathrm{d}^u \mathbf{q}(s(t))}{\mathrm{d}t^u}\right|_{t=0}$. We can regard $\mathbf{r}(t)$ as a composite function $\mathbf{y} \circ s$, where $\mathbf{y}(s) = \mathbf{r}(t(s))$ and $s = s(t)$. According to the chain rule for the high derivatives of the composite of two functions [16], we have

$$\frac{\mathrm{d}^u \mathbf{r}(t)}{\mathrm{d}t^u} = \sum_{k=1}^{u} \frac{A_{u,k}(t)}{k!} \cdot \frac{\mathrm{d}^k \mathbf{r}(t(s))}{\mathrm{d}s^k}, \qquad \frac{\mathrm{d}^u \mathbf{q}(s(t))}{\mathrm{d}t^u} = \sum_{k=1}^{u} \frac{A_{u,k}(t)}{k!} \cdot \frac{\mathrm{d}^k \mathbf{q}(s)}{\mathrm{d}s^k},$$

where

$$A_{u,k}(t) = \sum_{h=1}^{k} (-1)^{k-h} \binom{k}{h} s^{k-h}(t) \frac{\mathrm{d}^u s^h(t)}{\mathrm{d}t^u}, \quad 1 \le k \le u.$$

According to the third equation in (13), i.e., $s(0) = 0$, we have

$$A_{u,k}(0)=\sum_{h=1}^{k}(-1)^{k-h}\binom{k}{h}s^{k-h}(0)\left.\frac{\mathrm{d}^u s^h(t)}{\mathrm{d}s^u}\right|_{t=0}=\left.\frac{\mathrm{d}^u s^k(t)}{\mathrm{d}s^u}\right|_{t=0},\quad 1\le k\le u,$$

and the equation $\left.\frac{\mathrm{d}^u\mathbf{r}(t)}{\mathrm{d}t^u}\right|_{t=0}=\left.\frac{\mathrm{d}^u\mathbf{q}(s(t))}{\mathrm{d}t^u}\right|_{t=0}$ is equivalent to

$$\sum_{k=1}^{u}\frac{1}{k!}\cdot\left.\frac{\mathrm{d}^u s^k(t)}{\mathrm{d}s^u}\right|_{t=0}\cdot\left.\frac{\mathrm{d}^k\mathbf{r}(t(s))}{\mathrm{d}s^k}\right|_{s=0}=\sum_{k=1}^{u}\frac{1}{k!}\cdot\left.\frac{\mathrm{d}^u s^k(t)}{\mathrm{d}s^u}\right|_{t=0}\cdot\left.\frac{\mathrm{d}^k\mathbf{q}(s)}{\mathrm{d}s^k}\right|_{s=0}.$$

Also substituting (14) into the abovementioned equation, it holds

$$\left.\frac{\mathrm{d}^u s^u(t)}{\mathrm{d}s^u}\right|_{t=0}\cdot\left.\frac{\mathrm{d}^u\mathbf{r}(t(s))}{\mathrm{d}s^u}\right|_{s=0}=\left.\frac{\mathrm{d}^u s^u(t)}{\mathrm{d}s^u}\right|_{t=0}\cdot\left.\frac{\mathrm{d}^u\mathbf{q}(s)}{\mathrm{d}s^u}\right|_{s=0}. \tag{15}$$

According to the definition of $s(t)$ (5), we have

$$\left.\frac{\mathrm{d}^u s^u(t)}{\mathrm{d}s^u}\right|_{t=0}=\frac{u!}{\lambda^u}\neq 0.$$

Then (15) is revised as

$$\left.\frac{\mathrm{d}^u\mathbf{r}(t(s))}{\mathrm{d}s^u}\right|_{s=0}=\left.\frac{\mathrm{d}^u\mathbf{q}(s)}{\mathrm{d}s^u}\right|_{s=0}.$$

Combining (14) and the abovementioned equation, we obtain

$$\left.\frac{\mathrm{d}^i\mathbf{r}(t(s))}{\mathrm{d}s^i}\right|_{s=0}=\left.\frac{\mathrm{d}^i\mathbf{q}(s)}{\mathrm{d}s^i}\right|_{s=0},\ i=0,1,\ldots,u.$$

Therefore, it is shown that indeed the statement holds for u. Since both the basis and the inductive step have been proved, then the statement holds for all u. And Lemma 1 is proved.

To obtain a reparameterized approximation curve $\mathbf{q}(s(t))$ with endpoints constraints, the necessary and sufficient condition for $C^{(u,v)}$-continuity and the sufficient condition for $G^{(u,v)}$-continuity in matrix form is given in the following theorem.

Theorem 1. The rational Bézier curve $\mathbf{r}(t)$ (1) and the Bézier curve $\mathbf{q}(s)$ (7) satisfy (12) if and only if the following matrix equations hold:

$$\mathbf{Q}_m^1=\mathbf{N}_1(\lambda)\cdot diag(\omega_0,\omega_1,\ldots,\omega_u)\cdot\boldsymbol{\lambda}_1\cdot\mathbf{R}_n^1, \tag{16}$$

$$\mathbf{Q}_m^3 = \mathbf{N}_3(\lambda)\cdot diag(\omega_{n-v},...,\omega_{n-1},\omega_n)\cdot \boldsymbol{\lambda}_3 \cdot \mathbf{R}_n^3, \tag{17}$$

where

$$\mathbf{N}_1(\lambda) = diag((C_m^0)^{-1},(C_m^1)^{-1},...,(C_m^u)^{-1})\cdot \mathbf{S}_1^{-1}(\lambda)\cdot \mathbf{T}_1 \cdot diag(C_n^0,C_n^1,...,C_n^u),$$

$$\mathbf{S}_1(\lambda) = (s_{ij}^1(\lambda))_{(u+1)\times(u+1)},\ s_{ij}^1(\lambda) = \begin{cases} C_n^{i-j}\lambda^{i-j}\omega_{i-j}, & 1\le j\le i\le u+1; \\ 0, & 1\le i< j\le u+1. \end{cases}$$

$$\mathbf{T}_1 = (t_{ij}^1)_{(u+1)\times(u+1)},\ t_{ij}^1 = \begin{cases} C_m^{i-j}, & 1\le j\le i\le u+1; \\ 0, & 1\le i< j\le u+1. \end{cases}$$

$$\mathbf{N}_3(\lambda) = diag((C_m^{m-v})^{-1},...,(C_m^{m-1})^{-1},(C_m^m)^{-1})\cdot \mathbf{S}_2^{-1}(\lambda)\cdot \mathbf{T}_2 \cdot diag(C_n^{n-v},...,C_n^{n-1},C_n^n),$$

$$\mathbf{S}_2(\lambda) = (s_{ij}^2)_{(v+1)\times(v+1)},\ s_{ij}^2(\lambda) = \begin{cases} C_n^{n+i-j}\lambda^{n+i-j}\omega_{n+i-j}, & 1\le i\le j\le v+1; \\ 0, & 1\le j< i\le v+1. \end{cases}$$

$$\mathbf{T}_2 = (t_{ij}^2)_{(v+1)\times(v+1)},\ t_{ij}^2 = \begin{cases} C_m^{m+i-j}, & 1\le i\le j\le v+1; \\ 0, & 1\le j< i\le v+1. \end{cases},$$

here $C_n^i = \begin{pmatrix} n \\ i \end{pmatrix}$ is the binomial coefficient, and

$$\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 & & \\ & \boldsymbol{\lambda}_2 & \\ & & \boldsymbol{\lambda}_3 \end{pmatrix}, \tag{18}$$

where $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}_3$ are the diagonal matrices of order $u+1$, $n-u-v-1$ and $v+1$, respectively.

Proof. The curve $\mathbf{r}(t(s))$ in (12) is defined by (3), and it has the same shape and control points as those of $\mathbf{r}(t)$, only has different weights $\tilde{\omega}_k = \lambda^k\omega_k$ $(k=0,1,\ldots,n)$ from $\mathbf{r}(t)$. Then according to (3) and (12), we can directly obtain (16) and (17) by Theorem 2.3 in [13]. The theorem is proved.

Remark 1. The curves $\mathbf{r}(t)$ and $\mathbf{q}(s)$ satisfy $G^{(u,v)}$-continuity if (16) and (17) hold, i.e., it is a sufficient condition. The curves $\mathbf{r}(t)$ and $\mathbf{q}(s)$ satisfy $C^{(u,v)}$-continuity if and only if (16) and (17) hold when we set $\lambda=1$. That is, the parameter transformation is $s(t)=t$. It is a necessary and sufficient condition for $C^{(u,v)}$-continuity.

4.2. Unknown control points of the approximation curve

Note that for the sake of clarity, we also divide $\mathbf{B}_m$ into three parts as $\mathbf{B}_m = (\mathbf{B}_m^1, \mathbf{B}_m^2, \mathbf{B}_m^3)$, where

$$\mathbf{B}_m^1 = (B_0^m(s), B_1^m(s),..., B_u^m(s)), \mathbf{B}_m^2 = (B_{u+1}^m(s), B_{u+2}^m(s),..., B_{m-v-1}^m(s)), \mathbf{B}_m^3 = (B_{m-v}^m(s),..., B_{m-1}^m(s), B_m^m(s)).$$

Then an alternative function of the distance function (9) is expressed in matrix form as

$$d_\lambda^2(\mathbf{r}, \mathbf{q}) = \lambda \int_0^1 \frac{\left\| \frac{1}{\mathbf{B}_n \boldsymbol{\lambda\omega}} \mathbf{B}_n \boldsymbol{\lambda} \mathbf{W} \mathbf{R}_n - \mathbf{B}_m^1 \mathbf{Q}_m^1 - \mathbf{B}_m^2 \mathbf{Q}_m^2 - \mathbf{B}_m^3 \mathbf{Q}_m^3 \right\|^2}{(\lambda s + (1-s))^2} \mathrm{d}s.$$

To minimize the object function $d_\lambda^2(\mathbf{r}, \mathbf{q})$, the derivatives of $d_\lambda^2(\mathbf{r}, \mathbf{q})$ with respect to the elements of $\mathbf{Q}_m^2$ are zero. That is

$$\frac{\partial(d_\lambda^2(\mathbf{r}, \mathbf{q}))}{\partial(\mathbf{Q}_m^2)} = 2\lambda \int_0^1 \frac{\left(\mathbf{B}_m^2\right)^{\mathrm{T}} \mathbf{B}_n}{(\lambda s + (1-s))^2 \mathbf{B}_n \boldsymbol{\lambda\omega}} \mathrm{d}s \boldsymbol{\lambda} \mathbf{W} \mathbf{R}_n - 2\lambda \int_0^1 \frac{\left(\mathbf{B}_m^2\right)^{\mathrm{T}} \mathbf{B}_m^1}{(\lambda s + (1-s))^2} \mathrm{d}s \mathbf{Q}_m^1 -$$

$$2\lambda \int_0^1 \frac{\left(\mathbf{B}_m^2\right)^{\mathrm{T}} \mathbf{B}_m^2}{(\lambda s + (1-s))^2} \mathrm{d}s \mathbf{Q}_m^2 - 2\lambda \int_0^1 \frac{\left(\mathbf{B}_m^2\right)^{\mathrm{T}} \mathbf{B}_m^3}{(\lambda s + (1-s))^2} \mathrm{d}s \mathbf{Q}_m^3 = \mathbf{0}.$$

The abovementioned equation can be rewritten in matrix form as

$$\mathbf{N}_2 \boldsymbol{\lambda} \mathbf{W} \mathbf{R}_n - \mathbf{M}_1 \mathbf{Q}_m^1 - \mathbf{M}_2 \mathbf{Q}_m^2 - \mathbf{M}_3 \mathbf{Q}_m^3 = \mathbf{0}, \tag{19}$$

where

$$\mathbf{N}_2 := \mathbf{N}(u+2, u+3, ..., m-v; 1, 2, ..., n+1),$$

$$\mathbf{N} = (n_{ij})_{(m+1)\times(n+1)},\ n_{ij} = \int_0^1 \frac{B_i^m(s) B_j^n(s)}{(\lambda s + (1-s))^2 \mathbf{B}_n \boldsymbol{\lambda\omega}} \mathrm{d}s, \tag{20}$$

$$\mathbf{M}_1 := \mathbf{M}(u+2, u+3, ..., m-v; 1, 2, ..., u+1),$$

$$\mathbf{M}_2 := \mathbf{M}(u+2, u+3, ..., m-v;\ u+2, u+3, ..., m-v),$$

$$\mathbf{M}_3 := \mathbf{M}(u+2, u+3, ..., m-v;\ m-v+1, ..., m, m+1),$$

$$\mathbf{M} = (m_{ij})_{(m+1)\times(m+1)},\ m_{ij} = \int_0^1 \frac{B_i^m(s) B_j^m(s)}{(\lambda s + (1-s))^2} \mathrm{d}s,\ 0 \le i, j \le m \tag{21}$$

The notation $\mathbf{M}(\cdots;\cdots)$ denotes the submatrix of the matrix $\mathbf{M}$ obtained by extracting the specific rows and columns, and $\mathbf{N}(\cdots;\ \cdots)$ is treated analogously. Obviously, for any nonzero $(m-u-v-1)$ -dimensional column vector $\boldsymbol{\alpha}$, we have

$$\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{M}_2\boldsymbol{\alpha} = \int_0^1 \left(\frac{\mathbf{B}_m^2 \boldsymbol{\alpha}}{\lambda s + (1-s)} \right)^2 \mathrm{d}s > 0.$$

It is clear from the definition that the matrix $\mathbf{M}_2$ is a positive definite matrix, see [17]. So it is invertible. The unknown control points of the approximation curve are obtained by

$$\mathbf{Q}_m^2 = \mathbf{M}_2^{-1}\mathbf{N}_2\boldsymbol{\lambda}\mathbf{W}\mathbf{R}_n - \mathbf{M}_2^{-1}\mathbf{M}_1\mathbf{Q}_m^1 - \mathbf{M}_2^{-1}\mathbf{M}_3\mathbf{Q}_m^3. \tag{22}$$

Therefore, the control points $\mathbf{Q}_m$ of the approximation curve with endpoints $G^{(u,v)}$-continuity can be calculated by (16), (17) and (22).

Remark 2. For a polynomial approximation with endpoints $C^{(u,v)}$-continuity, the control points $\mathbf{Q}_m$ also can be calculated by (16), (17) and (22). Here, the parameter λ of the matrices $\mathbf{N}_1(\lambda)$ and $\mathbf{N}_3(\lambda)$ in (16) and (17) is equal to 1.

Remark 3. For a polynomial approximation without endpoints constraints, the matrixes $\mathbf{N}_1$ and $\mathbf{N}_3$ have no meanings. And its control points $\mathbf{Q}_m$ are represented by

$$\mathbf{Q}_m = \mathbf{M}^{-1}\mathbf{N}\mathbf{W}\boldsymbol{\lambda}\mathbf{R}_n,$$

where the matrixes $\mathbf{N}$ and $\mathbf{M}$ are defined by (20) and (21), respectively.

4.3. Minimization of the new distance function

After expressing all the control points of the approximation curve by the variable λ, the new distance function (9) can be revised as

$$d_\lambda(\mathbf{r},\mathbf{q}) = \sqrt{\lambda \int_0^1 \frac{1}{(\lambda s + (1-s))^2} \left\| \frac{\mathbf{B}_n\boldsymbol{\lambda}\mathbf{W}\mathbf{R}_n}{\mathbf{B}_n\boldsymbol{\lambda}\boldsymbol{\omega}} - \mathbf{B}_m\mathbf{Q}_m \right\|^2 \mathrm{d}s}$$

$$= \sqrt{\lambda \sum_{i=1}^{\delta} \left(\mathbf{R}_{n,i}^{\mathrm{T}}\mathbf{W}\boldsymbol{\lambda}\mathbf{V}\boldsymbol{\lambda}\mathbf{W}\mathbf{R}_{n,i} - 2\mathbf{Q}_{m,i}^{\mathrm{T}}\mathbf{N}\boldsymbol{\lambda}\mathbf{W}\mathbf{R}_{n,i} + \mathbf{Q}_{m,i}^{\mathrm{T}}\mathbf{M}\mathbf{Q}_{m,i} \right)}, \tag{23}$$

where

$$\mathbf{V} = (v_{ij})_{(n+1)\times(n+1)},\ v_{ij} = \int_0^1 \frac{B_i^n(s)B_j^n(s)}{\left((\lambda s + (1-s))\mathbf{B}_n\boldsymbol{\lambda}\boldsymbol{\omega}\right)^2}\mathrm{d}s, \tag{24}$$

the matrixes $\mathbf{W}$ and $\boldsymbol{\lambda}$ are defined as in (11), and the matrixes $\mathbf{N}$ and $\mathbf{M}$ are shown as in (20) and (21), respectively, δ is the dimension of the control points $\mathbf{r}_i\,(i = 0,1,\ldots,n)$ and $\mathbf{q}_j\,(j = 0,1,\ldots,m)$, $\mathbf{R}_{n,i}$

(i=1,2,…, δ) is the column vector composed by the i-th coordinate of the control points $\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_n$ in space $\Re^{\delta}$, and similar for $\mathbf{Q}_{m,i}$.

Remark 4. For a polynomial approximation without endpoints constraints, the distance error is

$$d_{\lambda}(\mathbf{r}, \mathbf{q}) = \sqrt{\lambda \sum_{i=1}^{\delta} \mathbf{R}_{n,i}^{\mathrm{T}} \mathbf{W} \boldsymbol{\lambda} \left(\mathbf{V} - \mathbf{N}^{\mathrm{T}} \mathbf{M}^{-1} \mathbf{N} \right) \boldsymbol{\lambda} \mathbf{W} \mathbf{R}_{n,i}},$$

where the matrixes $\mathbf{N}$ and $\mathbf{M}$ are defined by (20) and (21), respectively, $\mathbf{R}_{n,i}$ and δ are defined as shown in (23).

Obviously, $d_{\lambda}(\mathbf{r}, \mathbf{q})$ is a function of one-variable λ. So we need to find an optimal value of λ such that the distance function (23) which is also be denoted by $d(\lambda)$ reaches its minimum value. Due to the complicated representation of the object function $d(\lambda)$, it is impossible to obtain an explicit solution for the approximation. This function minimization can be accomplished by many numerical methods, e.g., one-dimensional search with first derivatives, Brent's method in one dimension, see Chapter 10 in [18]. Since the derivative of the function (23) is hardly calculated, we use the *golden section search in one dimension* to solve it [18]. We should first choose the initial bracketing triplet of abscissas *ax*, *bx*, *cx* (such that *bx* is between *ax* and *cx*, and the distance error function $d(bx)$ is less than both $d(ax)$ and $d(cx)$). Readers can refer to [18] for more details of how to choose the initial bracketing triplet.

In order to calculate the elements of the matrixes $\mathbf{N}$, $\mathbf{M}$ and $\mathbf{V}$, we need to calculate the integral of rational polynomials in (20), (21) and (24). As we know, it is very hard to integrate explicitly. So we consider using the numerical integration methods for good results. Here we apply the composite Simpson's rule to handle (20), (21) and (24). Suppose that the interval [0,1] is split up in H subintervals, with H an even number. Then the composite Simpson's rule is given by

$$\int_0^1 f(\lambda, t)\mathrm{d}t \approx \frac{1}{3H}\left[f(\lambda, 0) + 2\sum_{i=1}^{H/2-1} f(\lambda, t_{2i}) + 4\sum_{i=1}^{H/2} f(\lambda, t_{2i-1}) + f(\lambda, 1) \right],$$

where $t_i = ih$ for $i = 0,1,...,H-1$, $f(\lambda, t)$ is an integrand. To balance the speed and the accuracy of the numerical integration, we found that $50 \le H \le 100$ works well.

5. Numerical examples

We compare our method with some well-known methods for polynomial approximation of rational Bézier curves. For convenience, we call the method in [8] as the Degree Elevation method, abbreviated as the DE method; the method in [9] as the Weighted Progressive Iteration method, abbreviated as the WPI method; the method in [13] as the Least Squares method, abbreviated as the LS method.

At first, we compare these methods on scalar functions. We can see that our method and the LS method have an approximation curve of arbitrary degree and preserve $C^{(u,v)}$–continuity. And our method also preserves $G^{(u,v)}$–continuity. Whereas the DE method cannot obtain an approximation curve of low degree ($m<n$), and the DE and WPI methods only preserve $C^{(0,0)}$–continuity. Next, we present some numerical examples to compare the approximation errors under the Hausdorff distance as follows.

In the reparameterization method, we use the golden section search method to find the optimal value of the parameter λ to minimize the distance error function. As we all know, after doing a Möbius parameter transformation (2) to the rational Bézier curve (1), the weights become $\tilde{\omega}_i = \lambda^i \omega_i, (i = 0,...,n)$. If the values of the weights ω_i have almost the same magnitude order, the parameter λ should not be too big, such as $\lambda \le 10$. For example, if $\lambda = 10$ and the degree of the rational curve is 6, then the first and last new weights are $\tilde{\omega}_0 = \omega_0$, $\tilde{\omega}_6 = 10^6 \omega_6$. The difference in the magnitude order of the value of the new weights is too big. Obviously, the Möbius parameter transformation is not good. Experimentally, we found that [*ax*, *bx*, *cx*] = [0.2, 1, 5] is a suitable initial bracketing triplet for the golden section search method, if the values of the weights ω_i have almost the same magnitude order.

In addition, to achieve high accuracy of approximation error, it is desirable that the given tolerance ε in golden section search be as small as possible. However, since the computation time increases as the given tolerance decreases, we need to seek a balance between the speed and the accuracy. After our testing, we found that $\varepsilon=10^{-3}$ works well.

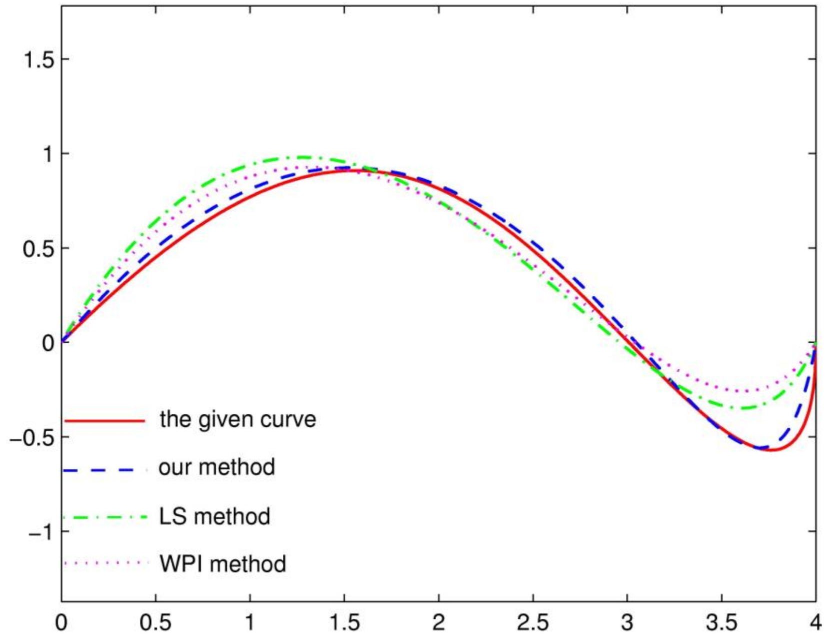
Example 1. The given curve is a rational Bézier curve of degree 4 with control points (0, 0), (2, 2), (3, 0), (4, -2), (4, 0) and the associated weights 5, 4, 2, 1, 1. We produce a 3-degree Bézier curve satisfying $C^{(0,0)}$–continuity with the given curve. For the approximation curve obtained by our method, the parameter λ is 1.480160, the iteration number for golden section search is 16, and the control points are (0, 0), (2.4696, 2.9089), (3.6159, -2.1736), (4, 0). The resulting curves are shown in the left-hand side of Fig. 1 and the corresponding error distance curves are illustrated in the right-hand side of Fig. 1. Also see Table 1 for comparisons of approximation errors under Hausdorff distance for different degrees. As shown in Table 1, the resulting approximation effect using our method is better than those of the other methods.

Table 1. Hausdorff distance comparisons of different approximation methods with different degrees.

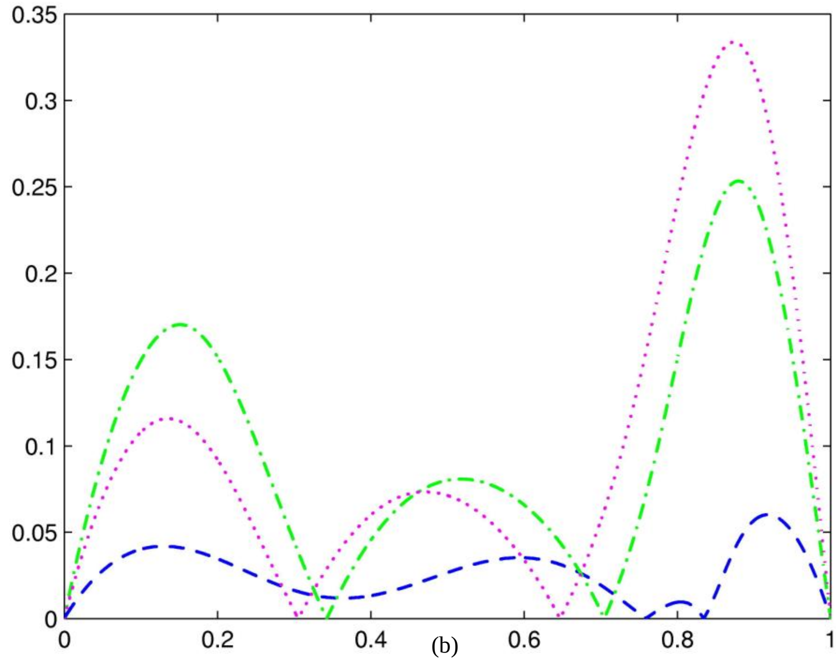
	Our method			the LS method	the WPI method (the reduction factor is 0.95)		the DE method
m	λ	Iter	error	error	Iter	error	error
3	1.480160	16	6.037148e-002	2.532691e-001	6	3.337644e-001	N/A
4	1.305553	17	1.689231e-002	5.082158e-002	10	7.247904e-002	1.994508e-001
5	0.893806	15	1.175240e-002	1.377046e-002	10	1.917103e-002	1.205578e-001

Example 2. (Also Example 3 in [12]) The given curve is a rational Bézier curve of degree 9 with the control points (17, 12), (32, 34), (-23, 24), (33, 62), (-23, 15), (25, 3), (30, -2), (-5, -8), (-5, 15), (11, 8) and the associated weights 1, 2, 3, 6, 4, 5, 3, 4, 2, 1. We find a 10-degree Bézier curve satisfying $C^{(0,0)}$–continuity with the given curve. The parameter λ in our method is 0.868737, and the iteration number is 15. The iteration number in WPI method is 6. The approximation errors under the Hausdorff distance from our method, LS method, WPI method and DE method are 0.246726, 0.317210, 3.253819, and

3.652520, respectively. The resulting curves are shown in the left-hand side of Fig. 2 and the corresponding error distance curves are illustrated in the right-hand side of Fig. 2. Clearly, our method also has a better approximation.



(a)



(b)

Fig. 1. (a) The rational Bézier curve of degree 4 and the resulting $C^{(0,0)}$ approximation curves of degree 3 obtained by different methods. (b) The corresponding error distance curves.

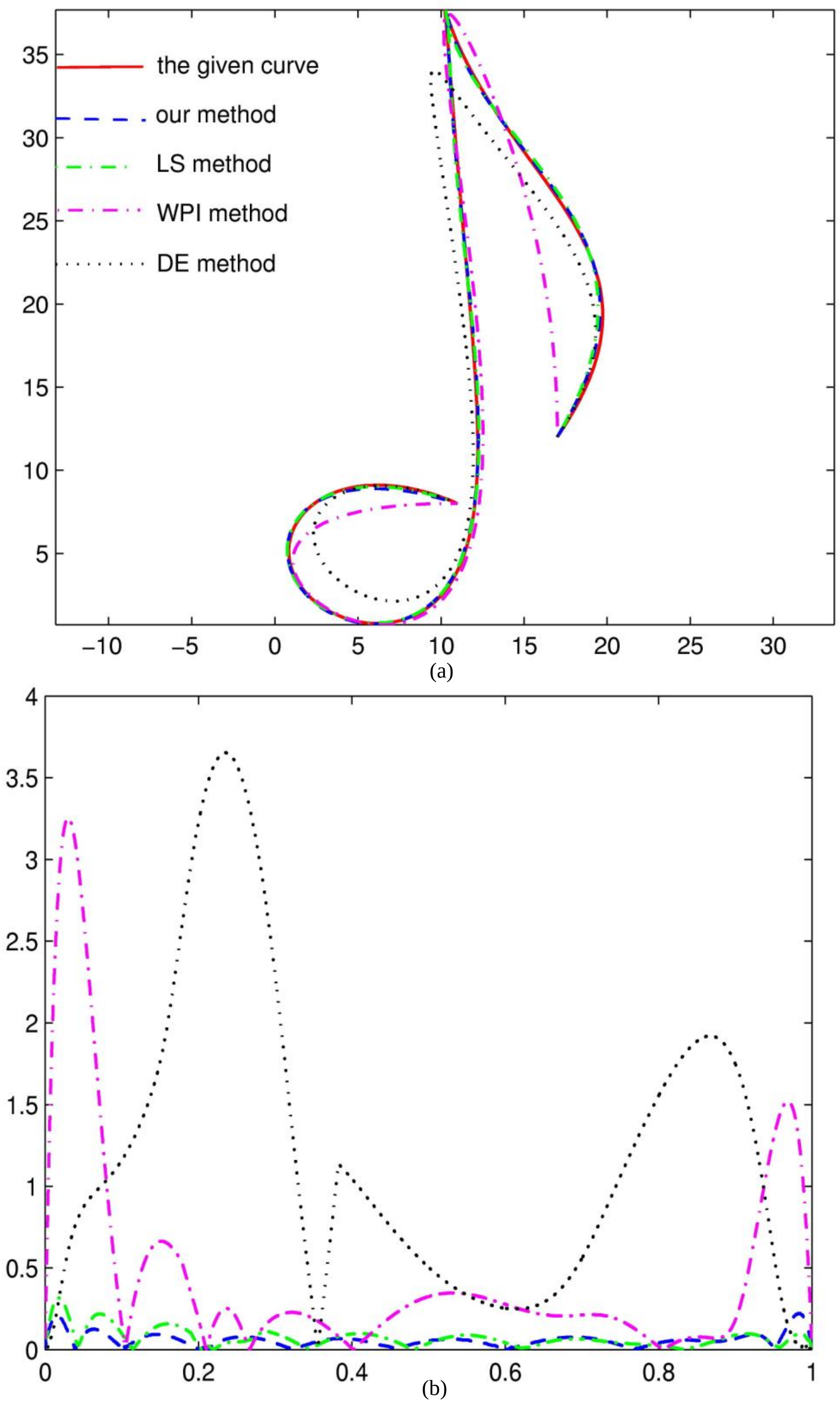


Fig. 2. (a) The rational Bézier curve of degree 9 and the resulting $C^{(0,0)}$ approximation curves of degree 10 obtained by different methods. (b) The corresponding error distance curves.

A 10-degree Bézier approximation curve satisfying $G^{(1,1)}$–continuity or $C^{(1,1)}$–continuity with the given curve is also discussed. For the case of $G^{(1,1)}$–continuity, the parameter λ in our method is 0.884231, the iteration number is 15, and the approximation errors under the Hausdorff distance is 0.402770. For the case of $C^{(1,1)}$–continuity, the parameter λ in our method is 0.980849, the iteration number is 15, and the approximation error under the Hausdorff distance is 0.691012.

Example 3. The given curve is a rational Bézier curve of degree 8 with the control points (0, 0), (0, 2), (2, 10), (4, 6), (6, 6), (11, 16), (8, 1), (9, 1), (10, 0) and the associated weights 1, 2, 3, 9, 12, 20, 30, 4, 1. We find a 5-degree Bézier curve with different continuity to approximate the given curve, see Fig. 3 for illustration. The approximation curves preserve $C^{(-1,-1)}$-continuity, $C^{(0,0)}$-continuity, and $C^{(1,1)}$-continuity at the endpoints, respectively. The parameter λ are 0.905420, 1.046971, and 0.713693, respectively. The interation numbers are 15, 16 and 15, and the corresponding Hausdorff distance errors are 0.583830, 0.245371, and 0.560612, respectively.

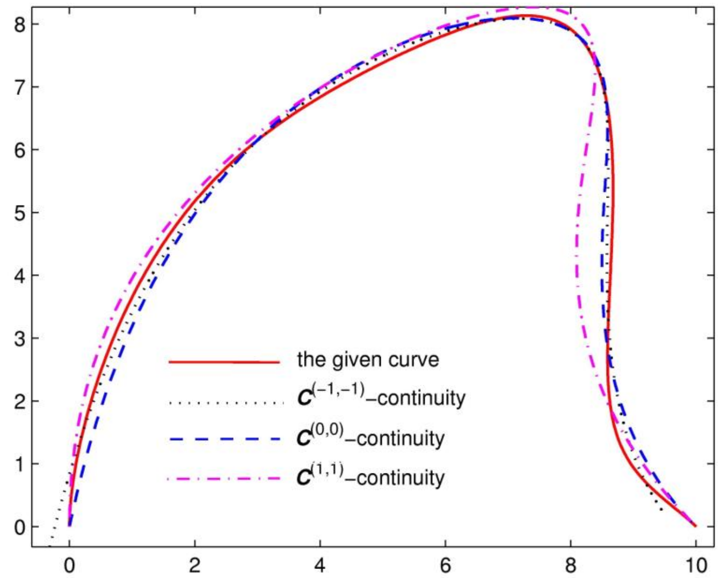


Fig. 3. The rational Bézier curve of degree 8, while the black dotted, blue dash, magenta dash-dotted curves are the approximation curves of degree 5 with $C^{(-1,-1)}$-continuity, $C^{(0,0)}$-continuity, and $C^{(1,1)}$-continuity, respectively.

Example 4. The given curve is a rational Bézier curve of degree 7 with control points (0, 0), (0.5, 2), (1.5, 2), (2.5, -2), (3.5, -2), (4.5, 2), (5.5, 2), (6, 0) and the associated weights 4, 10, 18, 8, 9, 40, 12, 20. We produce a 5-degree Bézier curve satisfying $C^{(0,0)}$–continuity with the given curve. The parameter λ is 0.681401, and the iteration number is 15 in the reparameterization method. The iteration number in WPI method is 8. The Hausdorff distance errors provided by our method, LS method, and WPI method are 0.074820, 0.101251, and 0.094650, respectively.

6. Conclusions

In this paper, we have proposed a reparameterization-based method for polynomial approximating rational Bézier curves with constraints. We use a new distance function

$$d_{\lambda}(\mathbf{r},\mathbf{q})=\sqrt{\int_0^1 \left\|\mathbf{r}(t)-\mathbf{q}(s(t))\right\|^2 \mathrm{d}t},$$

to obtain a better approximation effect under the Hausdorff distance. Using the least squares method and the golden section search method in one dimension, we get the best approximation curve under the new distance. Also the approximation curve and the given curve can satisfy parametric continuity or geometric continuity of any $u,v(u,v\geq 0)$ orders at two endpoints, respectively. Numerical examples show that our method has a better approximation effect than the previous methods under the Hausdorff distance.

To further improve the approximation results, we will consider doing a piecewise Möbius parameter transformation to the rational Bézier curve in future. However, it is not easy to determine how many pieces we need within given tolerance and the values of the subdivision points and the Möbius parameters. In addition, this paper only considers the case of rational curves. As for future work, the reparameterization method can also be applied to the cases of rational tensor-product Bézier surfaces and rational triangular Bézier surfaces.

Acknowledgements

The authors thank the anonymous referees for their valuable suggestions and comments. This work is supported by the National Natural Science Foundation of China (Grants Nos. 61202201, 61272307, 11201423, 11201430), the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1305), Zhejiang University, and the Ningbo Natural Science Foundation (Grant No. 2012A610036).

References

[1] Farin G. (Ed.), Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, fifth ed., Academic Press, San Diego, CA, 2002.

[2] Sederberg TW, Kakimoto M. Approximating rational curves using polynomial curves, in: G. Farin (Ed.), NURBS for Curve and Surface Design, SIAM, Philadelphia, 1991, pp. 149-158.

[3] Wang GJ, Sederberg TW, Chen FL. On the convergence of polynomial approximation of rational functions. Journal of Approximation Theory, 1997, 89: 267-288.

[4] Wang GZ, Zheng JM. Bounds on the moving control points of hybrid curves, CVGIP: Graphical Models and Image Processing, 1997, 59(1): 19-25.

[5] Chen J, Wang GJ. Hybrid polynomial approximation to higher derivatives of rational curves. Journal of Computational and Applied Mathematics, 2011, 235(17): 4925-2936.

[6] Wang GJ, Tai CL. On the convergence of hybrid polynomial approximation to higher derivatives of rational curves. Journal of Computational and Applied Mathematics, 2008, 214: 163-174.

[7] Floater MS. High order approximation of rational curves by polynomial curves. Computer Aided Design, 2006, 23: 621-628.

[8] Huang YD, Su HM, Lin HW. A simple method for approximating rational Bézier curves using Bézier curves. Computer Aided Geometric Design, 2008, 25: 697-699.

[9] Lu LZ. Sample-based polynomial approximation of rational Bézier curves. Journal of Computational and Applied Mathematics, 2011, 235: 1557-1563.

[10] Lu LZ. Weighted progressive iteration approximation and convergence analysis. Computer Aided Geometric Design. 2010, 27: 129-137.

[11] Woźny P, Lewanowicz S. Multi-degree reduction of Bézier curves with constraints, using dual Bernstein basis polynomials. Computer Aided Geometric Design, 2009, 26, 566-579.

[12] Lewanowicz S, P. Woźny P. Keller, Polynomial approximation of rational Bézier curves with constraints, Numerical Algorithms, 2012, 59(4): 607-622.

[13] Cai HJ, Wang GJ. Constrained approximation of rational Bézier curves based on a matrix expression of its end points continuity condition, Computer-Aided Design, 2010, 42: 495-504.

[14] Bert J. Bounding the Hausdorff distance between implicitly defined and/or parametric curves, in: T. Lyche, L.L. Schumaker (eds.), Mathematical Methods in CAGD, Vanderbilt University Press, Nashville 2001, 223-232.

[15] Chen XD, Ma WY, Paul JC. Multi-degree reduction of Bézier curves using reparameterization, Computer-Aided Design, 2011, 43: 161-169.

[16] Todorov PG. New explicit formulas for the *n*-th derivatives of composite functions. Pacific Journal of Mathematics, 1981, 92(1): 217-236.

[17] Demmel JW. Applied numerical linear algebra, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[18] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical Recipes in C++: The Art of Scientific Computing, second ed., Cambridge Univeristy Press, Cambridge, 2002.