

Article

# Convergence Analysis on a Second Order Algorithm for Orthogonal Projection onto Curves

Xiaowu Li <sup>1,†</sup> , Lin Wang <sup>1,†</sup>, Zhinan Wu <sup>2,†</sup>, Linke Hou <sup>3,\*,†</sup>, Juan Liang <sup>4,\*,†</sup> and Qiaoyang Li <sup>5,\*,†</sup>

<sup>1</sup> College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China; lixiaowu002@126.com (X.L.); wanglin@gzmu.edu.cn (L.W.)

<sup>2</sup> School of Mathematics and Computer Science, Yichun University, Yichun 336000, China; zhi\_nan\_7@163.com

<sup>3</sup> Center for Economic Research, Shandong University, Jinan 250100, China

<sup>4</sup> Department of Science, Taiyuan Institute of Technology, Taiyuan 030008, China

<sup>5</sup> Center for the World Ethnic Studies, Guizhou Minzu University, Guiyang 550025, China

\* Correspondence: abram75@163.com (L.H.); liangjuan76@126.com (J.L.); liqiaoy@gzmu.edu.cn (Q.L.); Tel.: +86-135-0640-1186 (L.H.); +86-137-3401-6169 (J.L.); +86-152-8558-3477 (Q.L.)

† These authors contributed equally to this work.

Received: 11 August 2017; Accepted: 27 September 2017; Published: 1 October 2017

**Abstract:** Regarding the point projection and inversion problem, a classical algorithm for orthogonal projection onto curves and surfaces has been presented by Hu and Wallner (2005). The objective of this paper is to give a convergence analysis of the projection algorithm. On the point projection problem, we give a formal proof that it is second order convergent and independent of the initial value to project a point onto a planar parameter curve. Meantime, for the point inversion problem, we then give a formal proof that it is third order convergent and independent of the initial value.

**Keywords:** point projection; planar parametric curve; second order convergence; third order convergence

## 1. Introduction

A vital problem is the projection of a point onto a surface or parametric curve in order to get the foot point and calculate the corresponding parameter values, which is widely applied in the field of Geometric Modeling [1–7], Computer Graphics and Computer Vision [5–7], Computer Aided Geometric Design [8–12], geometric design [13–15], geometry sculpt [16], scientific research and engineering applications [17] and computer animation [18–20].

To find the orthogonal projection of a point onto surfaces or parametric curves, Limaien and Trochu [1] constructed an auxiliary function and obtain all its zeros. Based on the geometric method [4], Li et al. [2] use the geometric iterative method with second order approximation properties. The common features of [2] and [4] are that they both use the curvature information, and they are independent of the initial value, respectively. If the algorithm in Hu et al. [4] is applied in the spatial parametric curve, the order of convergence will be one rather than two. In addition, the line segment between the center of circle, test point and osculating circle will not intersect in some cases.

Pottmann et al. [3] adopt the idea of the ICP algorithm and provide a new method to geometrically align a point cloud to a surface and to the related registration problems. An alternative notion has been presented in [3], which depends on both instantaneous kinematics and the geometry of the squared distance function corresponding to a surface. Evidenced by local convergence analysis, as well as by experiment results, the newly proposed ICP method is faster to converge than the classic one. Piegler et al. [5] present a method to project a point onto NURBS surfaces with three steps: decompose a NURBS surface into quadrilaterals, project the test point onto the nearest quadrilateral, and finally get the parameter from the nearest quadrilateral. Mortenson [21] converted the projection problem

into the problem of obtaining a polynomial's root, and then adopted the Newton-Raphson approach to get the root.

Chen et al. [18] used clipping circle technology and proposed an approach to find the minimum distance between a point and a NURBS curve. In the same vein, Chen et al. [19] used clipping sphere technology and provided an approach to find the minimum distance between a clamped B-Spline surface and a point. Similar to [18,19], Young-Taek Oh et al. [22] used an efficient culling technique to reduce redundant curves and surfaces and proposed an efficient method to project a point to its nearest point on a set of freeform surfaces and curves. Chen et al. [20] proposed a new quadratic clipping approach to calculate a root of a polynomial  $f(t)$  of degree  $n$  within an interval. Distinct from the classic method in  $R^1$  space, it approximates  $(t, f(t))$  and gets three quadratic curves in  $R^2$  space, which results in a higher order of approximation. Among various methods to find the roots to a polynomial equation in computer graphics and computer aided geometric design, clipping methods with the Bernstein–Bézier form demonstrate great numerical stability. To search the roots of a polynomial  $f(t)$  in an interval, Chen et al. [11] proposed a rational cubic clipping method, of which the approximation order and the corresponding convergence rate to search a single root are both seven. Based on their previous method in [11], Chen et al. [12] revise a rational cubic clipping approach to get two bounding cubic within  $O(n)$  time, which could get a faster convergence rate of five instead of four in a previous attempt. For Bézier curves, Chen et al. [23] improved the algebraic pruning method to prevent invalid roots after conversion of the point projection problem into a root finding problem, which is applicable for NURBS curves because, in NURBS curves, is not hard to be changed into Bézier form.

Song et al. [24] presented an algorithm to solve the problem of orthogonal projection of parametric curves onto B-spline surfaces. They use a tracing method and then create a polyline approximating the pre-image curve of the projection curve. Bharath Ram Sundar et al. [25] have constructed a measure of distance computation between surfaces and curves. To find point inversion for a parametric surface, Wang et al. [26] proposed a new method to give a simple procedure to get the parameters determined by a given point on a surface. They essentially formulate ordinary differential equation systems, which are determined by the intersection curve segment or by the projection curve segment. Li et al. [27] combined a first order algorithm with Newton's second order algorithm to propose the hybrid second order approach, which goes faster than the current methods and does not depend on the initial value. The adoption of ADMM in [28] contributes to refine the estimate in each iteration because it can incorporate information about the direction of estimates gotten in previous steps. In sum, those methods use lots of techniques, for instance, converting the projection problem into a root searching problem for a nonlinear equation system, subdivision approaches, and geometric approaches as well as circular clipping approaches.

One widely used, fast and robust method is introduced by the paper [4] and consists of two steps: calculate parameter values by projecting points onto curvature circles in the first step and then compute parameter increments using Taylor's expansion of the surface or curve in the second step. The paper [4] claims to be quadratic in its convergence rate but does not prove that. In this paper, regarding the point projection problem, we formally prove that projecting a point onto a planar parameter curve is of second order convergence and independent of the initial value. On the point inversion problem, we formally prove that it is of third order convergence and independent of the initial value.

## 2. Convergence Analysis

In this part, we conduct a convergence analysis of iteration, presented in formula Equation (5) in paper [4]. We parameterize the curvature circle  $\bar{c}$  such that it shares the same Taylor's polynomial with the curve  $c$ . Then this equality holds:

$$\begin{aligned} q &= \bar{c}(t_0 + \Delta t) = c(t_0 + \Delta t) + o(\Delta t^2) \\ &= c(t_0) + \Delta t c'(t_0) + \frac{\Delta t^2}{2} c''(t_0) + o(\Delta t^2) \end{aligned} \quad (1)$$

In  $R^2$ , we deal with the Equation (1) using the method in the paper [4] and get  $\det(q - c(t_n), c''(t_n)) = \Delta t \det(c'(t_n), c''(t_n)) + o(\Delta t^2)$ , which yields the iterative formula

$$\Delta t = \frac{\det(q - c(t_n), c''(t_n))}{\det(c'(t_n), c''(t_n))} = \frac{1}{k \|c'\|^3} \det(q - c(t_n), c''(t_n)) \quad (2)$$

On the point projection problem, to verify that the algorithm defined by Equation (2) (hereafter, the iterative method (2)) is quadratically convergent, we firstly derive the expression of the footpoint  $q$ . The corresponding parameter is denoted as  $\alpha$  when we orthogonally project a test point  $p$  onto a parameter curve  $c(t)$ , where  $p = (p_1, p_2)$  and  $c(t) = (f_1(t), f_2(t))$ . By the definition of the minimum distance between a curve and a point, we will get the relation as follows

$$(p - h) \times \mathbf{n} = 0 \quad (3)$$

where  $h = (f_1(\alpha), f_2(\alpha))$  and normal vector  $\mathbf{n} = (-f_2'(\alpha), f_1'(\alpha))$ . The unit normal vector, the curvature, the relative curvature, and the radius of curvature circle of curve  $c(t)$  can be expressed as follows, respectively,

$$\beta(t) = \left( -\frac{f_2'(t)}{\sqrt{(f_1'(t))^2 + (f_2'(t))^2}}, \frac{f_1'(t)}{\sqrt{(f_1'(t))^2 + (f_2'(t))^2}} \right) \quad (4)$$

$$k(t) = \frac{|f_1'(t)f_2''(t) - f_1''(t)f_2'(t)|}{((f_1'(t))^2 + (f_2'(t))^2)^{3/2}} \quad (5)$$

$$k_r(t) = \frac{f_1'(t)f_2''(t) - f_1''(t)f_2'(t)}{((f_1'(t))^2 + (f_2'(t))^2)^{3/2}} \quad (6)$$

$$R(t) = \frac{1}{\kappa(t)} = \frac{((f_1'(t))^2 + (f_2'(t))^2)^{\frac{3}{2}}}{|f_1'(t)f_2''(t) - f_1''(t)f_2'(t)|} \quad (7)$$

In this paper, we prove the case where  $|f_1'(t)f_2''(t) - f_1''(t)f_2'(t)| = f_1'(t)f_2''(t) - f_1''(t)f_2'(t)$ . Additionally, the proof method also holds for the case  $|f_1'(t)f_2''(t) - f_1''(t)f_2'(t)| = -(f_1'(t)f_2''(t) - f_1''(t)f_2'(t))$  only with an opposite sign to the previous case. So the formula Equation (7) can be written as

$$R(t) = \frac{((f_1'(t))^2 + (f_2'(t))^2)^{\frac{3}{2}}}{f_1'(t)f_2''(t) - f_1''(t)f_2'(t)} \quad (8)$$

The center of curvature circle of the planar curve  $c(t)$  is

$$m = (m_1, m_2) = c(t) + \beta(t)/k_r(t) \quad (9)$$

Now, inserting Equations (4) and (6) into Equation (9), we get

$$m_1(t) = f_1(t) - \frac{f_2'(t)((f_1'(t))^2 + (f_2'(t))^2)}{f_1'(t)f_2''(t) - f_1''(t)f_2'(t)} \quad (10)$$

and

$$m_2(t) = f_2(t) + \frac{f_1'(t)((f_1'(t))^2 + (f_2'(t))^2)}{f_1'(t)f_2''(t) - f_1''(t)f_2'(t)} \quad (11)$$

The line connecting the test point  $p$  and the center of curvature circle  $m$  can be parameterized as

$$\begin{cases} x = p_1 + (m_1 - p_1)u \\ y = p_2 + (m_2 - p_2)u \end{cases} \quad (12)$$

where  $u$  is a parameter. Also, the equation of the curvature circle is

$$(x - m_1)^2 + (y - m_2)^2 = R^2 \quad (13)$$

As the footpoint  $q = (q_1, q_2)$  is the intersection of the line Equation (12) and the curvature circle Equation (13), substituting Equation (12) into Equation (13), we get

$$(p_1 + (m_1 - p_1)u - m_1)^2 + (p_2 + (m_2 - p_2)u - m_2)^2 = R^2 \quad (14)$$

Solving Equation (14), we have

$$u = 1 \pm \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}} \quad (15)$$

Since the footpoint  $q$  lies between the center of curvature circle  $m$  and the test point  $p$ , the parameter  $u$  must belong to  $(0, 1)$ . By Equation (15), the parameter  $u$  will be

$$u = 1 - \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}} \quad (16)$$

From Equations (3), (10), (12) and (16), we get horizontal coordinate  $q_1$  of the intersection  $q$

$$\begin{aligned} q_1 &= p_1 + (m_1 - p_1)u \\ &= p_1 + (m_1 - p_1)\left(1 - \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}}\right) \end{aligned} \quad (17)$$

From Equations (11), (12) and (16), we obtain vertical coordinate  $q_2$  of the intersection  $q$

$$\begin{aligned} q_2 &= p_2 + (m_2 - p_2)u \\ &= p_2 + (m_2 - p_2)\left(1 - \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}}\right) \end{aligned} \quad (18)$$

From Equation (17), we have

$$\begin{aligned} q_1 - f_1(t) &= -f_1(t) + p_1 + (m_1 - p_1)u \\ &= -f_1(t) + p_1 + (m_1 - p_1)\left(1 - \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}}\right) \end{aligned} \quad (19)$$

From Equation (18), we get

$$\begin{aligned}
 q_2 - f_2(t) &= -f_2(t) + p_2 + (m_2 - p_2)u \\
 &= -f_2(t) + p_2 + (m_2 - p_2)\left(1 - \frac{R}{\sqrt{(m_1 - p_1)^2 + (m_2 - p_2)^2}}\right)
 \end{aligned}
 \tag{20}$$

Till now, we provide the detailed illustration of the iterative method (2). Next, we will use a method similar to those in the literature [29–31] to prove the theorem. The standard convergence analysis method in numerical analysis in [29–31] are used, namely,  $e_{n+1} = C_0 e_n^k + o(e_n^{k+1})$ , where  $e_{n+1} = t_{n+1} - \alpha, e_n = t_n - \alpha, k$  is a positive integer,  $C_0$  is constant.

**Theorem 1.** Let  $q(t) - c(t) = (q_1(t) - f_1(t), q_2(t) - f_2(t))$  be a real parametric curve function in two dimensions, suppose  $q(t) - c(t)$  has first and second derivatives within an interval  $I$ . When  $q(t) - c(t)$  has only a single root  $\alpha \in I$ , and  $t_n$  stays close sufficiently to  $\alpha$ , then the iterative method (2) will be of second order convergence for the point projection problem.

**Proof.** Let  $e_n = t_n - \alpha, a_i = (1/i!) (f_1^{(i)}(\alpha)), b_i = (1/i!) (f_2^{(i)}(\alpha)), i = 0, 1, 2, 3, \dots$ . Assume that  $\alpha$  is a simple root of  $q - c(t) = (q_1 - f_1(t), q_2 - f_2(t)) = 0$ , i.e.,  $q(\alpha) - c(\alpha) = (q_1(\alpha) - f_1(\alpha), q_2(\alpha) - f_2(\alpha)) = 0, q'(\alpha) - c'(\alpha) = (q_1'(\alpha) - f_1'(\alpha), q_2'(\alpha) - f_2'(\alpha)) \neq 0$ . For  $f_1(t), f_2(t)$ , we take the Taylor’s expansion around  $\alpha$ ,

$$f_1(t_n) = a_0 + a_1 e_n + a_2 e_n^2 + o(e_n^3) \tag{21}$$

$$f_2(t_n) = b_0 + b_1 e_n + b_2 e_n^2 + o(e_n^3) \tag{22}$$

Furthermore, we have

$$f_1'(t_n) = a_1 + 2a_2 e_n + o(e_n^2) \tag{23}$$

$$f_2'(t_n) = b_1 + 2b_2 e_n + o(e_n^2) \tag{24}$$

and

$$f_1''(t_n) = 2a_2 + o(e_n) \tag{25}$$

$$f_2''(t_n) = 2b_2 + o(e_n) \tag{26}$$

For the sake of simplicity, we use the Maple 18 package to calculate the Taylor’s expansion in the following way. Combine the derivation of Equations (4)–(20) with the orthogonal projection condition for Equation (3) and Equation (21)–(26), and then simplify these equations, we acquire parameterized increment  $\Delta t$  as follows:

$$\begin{aligned}
 \Delta t &= -e_n - \frac{6}{\delta(a_1^2 + b_1^2)} (a_1 a_2^2 p_2 b_1 - a_1^2 b_2 p_2 a_2 + a_1^2 b_2 b_0 a_2 - a_1 a_2^2 b_0 b_1 \\
 &\quad - b_1 b_2^2 p_2 a_1 + b_1^2 b_2 p_2 a_2 + b_1 b_2^2 b_0 a_1 - b_1^2 b_2 b_0 a_2) e_n^2 + o(e_n^3)
 \end{aligned}
 \tag{27}$$

where  $\delta = a_1^3 + a_1 b_1^2 - 2p_2 b_2 a_1 + 2p_2 a_2 b_1 + 2b_0 a_1 b_2 - 2b_0 a_2 b_1, p_2 = \frac{a_0 a_1 - a_1 p_1 + b_0 b_1}{b_1}$  and  $p_2$  also satisfies the relationship  $(p_1 - f_1(\alpha))f_1'(\alpha) + (p_2 - f_2(\alpha))f_2'(\alpha) = 0$ , which is transformed from the Equation (3).

Because

$$\Delta t = e_{n+1} - e_n \tag{28}$$

Substituting Equations (28) into (27), we can get the following error equation similar to that in [32],

$$e_{n+1} = -\frac{6}{\delta(a_1^2 + b_1^2)}(a_1 a_2^2 p_2 b_1 - a_1^2 b_2 p_2 a_2 + a_1^2 b_2 b_0 a_2 - a_1 a_2^2 b_0 b_1 - b_1 b_2^2 p_2 a_1 + b_1^2 b_2 p_2 a_2 + b_1 b_2^2 b_0 a_1 - b_1^2 b_2 b_0 a_2)e_n^2 + o(e_n^3) \quad (29)$$

This implies that the algorithm defined by the iterative method (2) is quadratically convergent for the point projection problem. The proof is completed.  $\square$

Then, we will illustrate that the algorithm expressed by the iterative method (2) is third order convergent for the point inversion problem.

**Theorem 2.** Let  $q(t) - c(t) = (q_1(t) - f_1(t), q_2(t) - f_2(t))$  be a real parametric curve function in two dimensions, suppose  $q(t) - c(t)$  has first and second derivatives within an interval. When  $q(t) - c(t)$  has only one single root  $\alpha \in I$ , and  $t_n$  stays close sufficiently to  $\alpha$ , so the iterative method (2) will be of cubic convergence for point the inversion problem.

**Proof.** Given the point inversion problem is a special case of the point projection problem, the proof of this theorem will use the results in the proof of Theorem 1. For the same part, we will not repeat. Since the test point  $p$  is on the parametric curve  $c(t)$ , from Equation (3), we can know that  $(p_1, p_2) = (a_0, b_0)$ . From Equation (29) and  $(p_1, p_2) = (a_0, b_0)$ , we acquire the following error equation in [32],

$$e_{n+1} = \frac{2(a_1 a_2 + b_1 b_2)^2}{(a_1^2 + b_1^2)^2} e_n^3 + o(e_n^4) \quad (30)$$

This implies that the algorithm expressed by the iterative method (2) is third order convergent for the point inversion problem. The proof is completed.  $\square$

**Theorem 3.** The iterative method (2) is independent of the initial value for point projection and inversion problem.

**Proof.** Firstly, we presents the interpretation for Figure 1. Assume there exists a horizontal axis  $t$  and planar parametric curve  $c(t)$ . There are two points on the planar parametric curve  $c(t)$ : the first point  $c(t_n)$  with  $t_n$  on the horizontal axis, and the second point  $c(\alpha)$  is the corresponding orthogonal projection point of the test point  $p$  onto the planar parametric curve  $c(t)$ , where  $\alpha$  is the corresponding parameter value on the horizontal axis. While, according to the iterative method (2), the line segment determined by point  $p$  and point  $c(\alpha)$  and tangent line of planar parametric curve  $c(t)$  at  $t = \alpha$  are mutually perpendicular. The tangent line of the planar parametric curve  $c(t)$  through point  $c(t_n)$  will determine a footpoint  $q$ , which is obtained by orthogonally projecting test point  $p$  onto the tangent line of the curve  $c(t)$  at  $t = t_n$ . The corresponding parametric value  $t_{n+1}$  of footpoint  $q$  on the horizontal axis is obviously the next iterative value used by the iterative method (2). The corresponding parametric value of the middle point of the line segment determined by the point  $c(t_n)$  and the footpoint  $q$  is  $M$ .

Secondly, we present the proof of Theorem 3. When the iterative method (2) only has one solution, it is independent of the initial value for point projection and inversion problem.

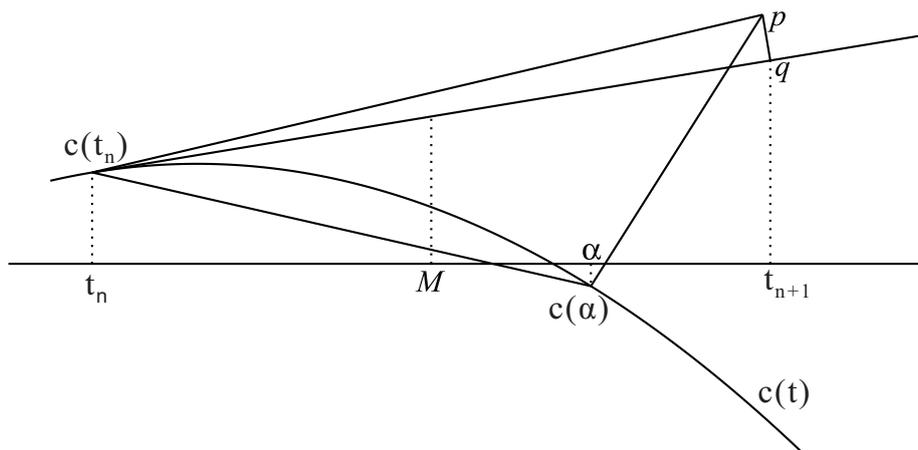


Figure 1. Graphic demonstration for convergence analysis.

The proof method is similar to the ones in references [33–35]. It is not difficult to show that the corresponding parameter of the first dimensional for the planar parametric curve on the two-dimensional horizontal plane is  $t$ . When the iterative method (2) begins to iterate, as the iterative parameter value satisfies the inequality relationship  $t_n < \alpha$ , by the graphical demonstration, the parameter corresponding to the footpoint  $q$  will be  $t_{n+1}$ . The middle point of point  $(t_{n+1}, 0)$  and point  $(t_n, 0)$  is  $(M, 0)$ , which implies the relationship  $M = \frac{t_n + t_{n+1}}{2}$ . Additionally, because of  $0 < \Delta t = t_{n+1} - t_n$ , there is an inequality relationship  $t_n < M < \alpha$ . This result sufficiently yields the inequality  $t_n - \alpha < t_{n+1} - \alpha < \alpha - t_n = -(t_n - \alpha)$ , namely, there is an iterative error expression  $|e_{n+1}| < |e_n|$ , where  $e_n = t_n - \alpha$ . If  $t_n > \alpha$ , then the proving method is similar to the case with  $t_n < \alpha$ . Thus, for the planar parametric curve  $c(t)$ , there is an iterative error expression  $|e_{n+1}| < |e_n|$  in the two-dimensional horizontal plane, namely, the iterative method (2) is independent of the initial iterative parametric value.

Alternatively, we will present a descriptive proof that the iterative method (2) is independent of the initial iterative parametric value. This proof method is also similar to the one in the literature [33–35]. Suppose that the starting point lies on the left of  $\alpha$ , then footpoint  $q$  will lie on the right to the starting point, and hence  $\Delta t$  will be positive. By the iterative method (2), the iterative sequence is expressed as follows,

$$t_n = t_{n-1} + \Delta t_{n-1} \tag{31}$$

where

$$\Delta t_{n-1} = \frac{\det(q - c(t_{n-1}), c''(t_{n-1}))}{\det(c'(t_{n-1}), c''(t_{n-1}))} = \frac{1}{k \|c'\|^3} \det(q - c(t_{n-1}), c''(t_{n-1}))$$

If  $t_n < \alpha$ , the sequence  $t_n$  is strictly monotonically increasing. If  $t_n > \alpha$ , by at most three iterations, the sequence  $t_n$  converge to  $\alpha$ . Notice that this iterative sequence performs like an attenuated pendulum and converges to  $\alpha$ . Furthermore, when the starting point lies to the right of  $\alpha$ , the convergence holds in the same way. To summarize, the footpoint  $q$  depends on the test point  $p$  all the time, rather than on the starting iteration point. So the iterative method (2) is independent of the initial value. The proof is completed.  $\square$

**Remark 1.** The iterative method (2) is independent of the initial value, but it only requires that  $c(t)$  is  $C^2$ , while the methods in the literature [33–35] need to satisfy the sufficient conditions for global convergence. In addition, the iterative method (2) fits in with robustness and stabilization in Professor Les A. Piegl's view [36].

**Remark 2.** If  $k(t) = 0$  of the Equation (5), then  $t = t_0$ . Hence, we calculate the distance between the point  $c(t_0)$  on the curve  $c(t)$  and the test point  $p$ , i.e.,  $d_1 = \|p - c(t_0)\|$ . At the same time, in order to deal with

the possible failure of the projection or inversion, we use the parameter perturbation method and increment the parameter  $t_0$  by a small positive constant  $\epsilon$ ,  $t_0 = t_0 + \epsilon$ , so that the iteration can continue. During the iteration process, when the derivative of the curve  $c(t)$  at  $t = t_0$  becomes zero, the same perturbation approach is applied. Eventually,  $d = \min\{d_1, d_2, d_3, \dots\}$  becomes the minimum distance between the curve  $c(t)$  and the test point  $p$ .

### 3. Numerical Experiments

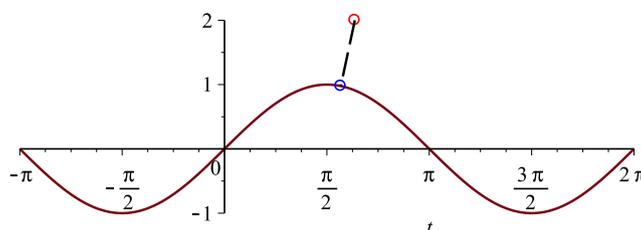
This section provides numerical evidence to compare the behavior of the Newton iterative algorithm and the iterative method (2) discussed above.

**Example 1.** We consider the curve  $c(t) = (t, \sin(t))$  with test point  $p = (2, 2)$  and the corresponding parameter of projection point  $\alpha = 1.7838126561069$ . Table 1 shows the results of Newton iteration and the iterative method (2). The results verify that the iterative method (2) has better convergence and robustness, while the Newton algorithm is dependent on the initial value but is unstable (See Figure 2).

**Table 1.** Step sizes  $\Delta t_1, \Delta t_2$  in Example 1 for the Newton method and the iterative method (2).

$p = (2, 2), t_0 = -0.32$							
iterative methods	Step	1	2	3	4	5	6
Newton method	$\Delta t_1$	NC	NC	NC	NC	NC	NC
The iterative method (2)	$\Delta t_2$	1.82	$2.8 \times 10^{-1}$	$2.1 \times 10^{-3}$	$8.3 \times 10^{-7}$	$1.4 \times 10^{-13}$	0
$p = (2, 2), t_0 = 4.2$							
iterative methods	Step	2	3	4	5	6	7
Newton method	$\Delta t_1$	NC	NC	NC	NC	NC	NC
The iterative method (2)	$\Delta t_2$	$-1.65 \times 10^{-1}$	$-4.1 \times 10^{-2}$	$3.5 \times 10^{-4}$	$2.5 \times 10^{-8}$	$2.2 \times 10^{-16}$	0

Note: we use NC in Table 1 to represent that the Newton method does not converge, and the same notation applies in Table 2 to Table 6.



**Figure 2.** Graphic demonstration for Example 1.

**Example 2.** We use the curve  $c(t) = (t, \cos(t))$  with test point  $p = (2, 5)$  and the corresponding parameter of projection point  $\alpha = 0.402360707683495$ . Table 2 shows the results of Newton iteration and the iterative method (2). The results also verify that the iterative method (2) has better convergence and robustness, while Newton algorithm is dependent on the initial value but is unstable (See Figure 3).

**Table 2.** Step sizes  $\Delta t_1, \Delta t_2$  in Example 2 for the Newton method and the iterative method (2).

$p = (2, 5), t_0 = 1.7$								
iterative methods	Step	4	5	6	7	8	9	10
Newton method	$\Delta t_1$	NC	NC	NC	NC	NC	NC	NC
The iterative method (2)	$\Delta t_2$	-1.04	$3.29 \times 10^{-1}$	$2.94 \times 10^{-2}$	$4.48 \times 10^{-4}$	$1.05 \times 10^{-7}$	$5.77 \times 10^{-15}$	0.0
$p = (2, 5), t_0 = -2.4$								
iterative methods	Step	5	6	7	8	9	10	11
Newton method	$\Delta t_1$	NC	NC	NC	NC	NC	NC	NC
The iterative method (2)	$\Delta t_2$	-1.34	-0.461	$9.08 \times 10^{-2}$	$4.08 \times 10^{-3}$	$8.69 \times 10^{-6}$	$3.94 \times 10^{-11}$	0



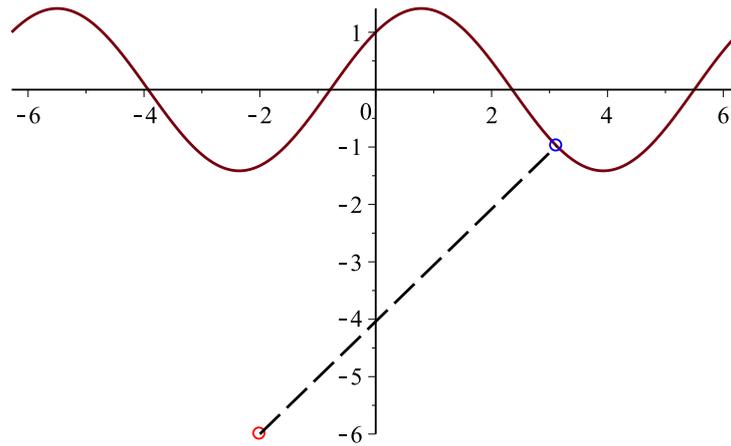


Figure 5. Graphic demonstration for Example 4.

**Example 5.** We consider the curve  $c(t) = (t, \sin(0.5t))$ ,  $t \in [-6, 2]$ . Test point is  $p = (3, -7)$ , and the corresponding parameter of projection point is  $-0.35701470284643136$ . Table 5 shows number of iterations for Newton iteration and the iterative method (2). Results show the the iterative method (2) has better convergence and robustness, at the same time, Newton’s method is sometimes dependent on the initial value but is unstable (See Figure 6).

**Table 5.** Comparison of the robustness and effectiveness for the Newton’s method and the iterative method (2).

$t_0$	-5.6	-5.5	-5.3	-5.2	-5.0	-4.9	-4.8	-4.7	-4.6	-4.5
The iterative method (2)	11	11	11	12	14	11	13	14	11	14
Newton method	NC									
$t_0$	-4.3	-4.0	-3.8	-3.4	-3.0	-2.3	-2.0	-1.0	0.4	1.0
The iterative method (2)	13	11	16	7	10	6	11	11	12	14
Newton method	NC									

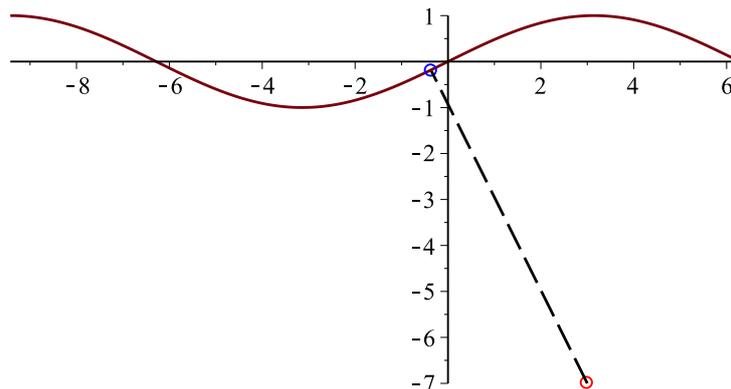
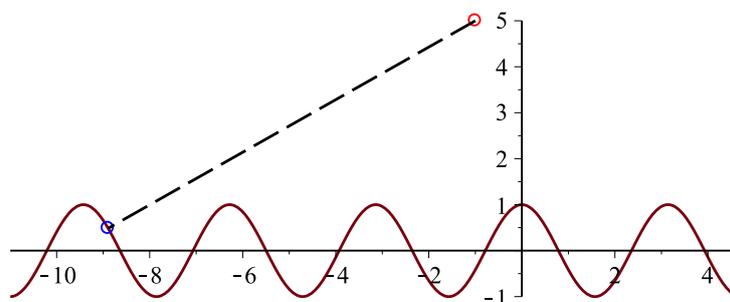


Figure 6. Graphic demonstration for Example 5.

**Example 6.** We use the curve  $c(t) = (t, \cos(2t))$ ,  $t \in [-12, -8]$ . The test point is  $p = (-1, 5)$ , and the corresponding parameter of projection point is  $-8.8928968570625730$ . Table 6 shows the number of iterations for Newton iteration and the iterative method (2). Results show the iterative method (2) has better convergence and robustness, at the same time, Newton’s method is sometimes dependent on the initial value but is unstable (See Figure 7).

**Table 6.** Comparison of the robustness and effectiveness for the Newton's method and the iterative method (2).

$t_0$	-12.8	-12.4	-12.1	-11.9	-11.7	-11.0	-10.8	-10.7	-10.4	-10.0
The iterative method (2)	59	55	59	46	46	53	41	42	39	32
Newton method	NC									

**Figure 7.** Graphic demonstration for Example 6.

**Remark 3.** The iterative method (2) orthogonally projects a test point onto a parametric curve  $c(t)$ . For the case with multiple orthogonal points, our approach is changed in the following way:

- (1) Divide a parameter interval  $[a, b]$  of parametric curve  $c(t)$  into  $M$  subintervals with equal length.
- (2) Randomly select an initial iterative parametric value in each subinterval.
- (3) Using the iterative method (2) and using each initial iterative parametric value, iterate, respectively. Suppose that the iterative parametric values are  $\alpha_1, \alpha_2, \dots, \alpha_M$ , respectively.
- (4) Calculate the local minimum distances  $d_1, d_2, \dots, d_M$ , where  $d_i = \|p - c(\alpha_i)\|$ .
- (5) Calculate the global minimum distance  $d = \|p - c(\alpha)\| = \{d_1, d_2, \dots, d_M\}$ .

If we attempt to search all the solutions as fast as possible, divide a parameter interval  $[a, b]$  into  $M$  subintervals with equal length such that  $M$  is sufficiently large.

According to Figures 5 and 7, either blue point is the only orthogonal projection point of the test point, but not the point with minimum distance. For  $t \in [-6, 6]$  in Figure 5, four parameter values are  $-5.8827783434357889$ ,  $-2.3086073340017088$ ,  $1.0869042777921642$ ,  $3.1213051310788399$ , respectively. According to Remark 3, it is easy to find that the orthogonal projection point determined by the parameter value of  $-2.3086073340017088$  is the one with minimum distance, but the orthogonal projection points determined by the other parameter values are not. For  $t \in [-10, 3]$  in Figure 7, eight parameter values are  $-8.1893433622682510$ ,  $-5.9663756035363462$ ,  $-4.8785206470437198$ ,  $-3.0152895587637027$ ,  $-1.5956295339830861$ ,  $-5.8855753296468201 \times 10^{-2}$ ,  $1.6840779491863052$ ,  $2.8953236261013390$ , respectively. In the same way, only the orthogonal projection point determined by the parameter value of  $-5.8855753296468201 \times 10^{-2}$  is the one with minimum distance.

#### 4. Conclusions

In this paper, regarding the point projection problem, we formally prove that projection of a point onto a planar parameter curve is quadratically convergent and independent of the initial value. On the point inversion problem, we formally prove that the inversion problem is cubically convergent and independent of the initial value. In the future, we would like to present a formal proof of convergence on projecting a point onto a spatial parametric curve and surface in literature [4].

**Acknowledgments:** We would like to thank the editor and two anonymous reviewers for their efforts to improve greatly the manuscript. We acknowledge the support by the National Natural Science Foundation of China (Grant No. 61263034, Grant No. 71772106), Scientific and Technology Foundation Funded Project of Guizhou

Province (Grant No. [2014]2092, Grant No. [2014]2093), Feature Key Laboratory for Regular Institutions of Higher Education of Guizhou Province (Grant No. [2016]003), National Bureau of Statistics Foundation Funded Project (Grant No. 2014LY011), Key Laboratory of Pattern Recognition and Intelligent System of Construction Project of Guizhou Province (Grant No. [2009]4002), Information Processing and Pattern Recognition for Graduate Education Innovation Base of Guizhou Province. Linke Hou is supported by Shandong Provincial Natural Science Foundation of China (Grant No. ZR2016GM24). Juan Liang is supported by Scientific and Technology Key Foundation of Taiyuan Institute of Technology (Grant No. 2016LZ02). Qiaoyang Li is supported by Fund of National Social Science (No. 14XMZ001) and Fund of Chinese Ministry of Education (No. 15JZD034).

**Author Contributions:** The contributions of all of the authors are the same. All of them have worked together to develop the present manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Limaïem, A.; Trochu, F. Geometric algorithms for the intersection of curves and surfaces. *Comput. Graph.* **1995**, *19*, 391–403.
2. Li, X.; Wu, Z.; Hou, L.; Wang, L.; Yue, C.; Xin, Q. A geometric orthogonal projection strategy for computing the minimum distance between a point and a spatial parametric curve. *Algorithms* **2016**, *9*, 15.
3. Pottmann, H.; Leopoldseder, S.; Hofer, M. Registration without ICP. *Comput. Vis. Image Underst.* **2004**, *95*, 54–71.
4. Hu, S.M.; Wallner, J. A second order algorithm for orthogonal projection onto curves and surfaces. *Comput. Aided Geom. Des.* **2005**, *22*, 251–260.
5. Piegl, L.A.; Tiller, W. Parameterization for surface fitting in reverse engineering. *Comput. Aided Des.* **2001**, *33*, 593–603.
6. Ma, Y.L.; Hewitt, W.T. Point inversion and projection for NURBS curve and surface: Control polygon approach. *Comput. Aided Geom. Des.* **2000**, *20*, 79–99.
7. Oh, Y.T.; Kim, Y.J.; Lee, J.; Kim, M.S.; Elber, G. Continuous point projection to planar freeform curves using spiral curves. *Vis. Comput.* **2011**, *28*, 111–123.
8. Srijuntongsiri, G. An iterative/subdivision hybrid algorithm for curve/circle intersection. *Vis. Comput.* **2011**, *27*, 365–371.
9. Chen, X.D.; Ma, W. Geometric point interpolation method in  $R^3$  space with tangent directional constraint. *Comput. Aided Des.* **2012**, *44*, 1217–1228.
10. Chen, X.D.; Yang, C.; Ma, W. Coincidence condition of two Bézier curves of an arbitrary degree. *Comput. Graph.* **2016**, *54*, 121–126.
11. Chen, X.D.; Ma, W.; Ye, Y. A rational cubic clipping method for computing real roots of a polynomial. *Comput. Aided Geom. Des.* **2015**, *38*, 40–50.
12. Chen, X.D.; Ma, W. Rational cubic clipping with linear complexity for computing roots of polynomials. *Appl. Math. Comput.* **2016**, *273*, 1051–1058.
13. Lin, H.W. Local progressive-iterative approximation format for blending curves and patches. *Comput. Aided Geom. Des.* **2010**, *27*, 322–339.
14. Lin, H.W. The convergence of the geometric interpolation algorithm. *Comput. Aided Des.* **2010**, *42*, 505–508.
15. Lin, H.; Qin, Y.; Liao, H.; Xiong, Y. Affine arithmetic-based B-Spline surface intersection with gpu acceleration. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 172–181.
16. Sederberg, T.W.; Lin, H.; Li, X. Curvature of singular Bézier curves and surfaces. *Comput. Aided Geom. Des.* **2011**, *28*, 233–244.
17. Lin, H.; Zhang, Z. An efficient method for fitting large data sets using T-Splines. *SIAM J. Sci. Comput.* **2013**, *35*, A3052–A3068.
18. Chen, X.D.; Yong, J.H.; Wang, G.; Paul, J.C.; Xu, G. Computing the minimum distance between a point and a NURBS curve. *Comput. Aided Des.* **2008**, *40*, 1051–1054.
19. Chen, X.D.; Xu, G.; Yong, J.H.; Wang, G.; Paul, J.C. Computing the minimum distance between a point and a clamped B-spline surface. *Graph. Model.* **2009**, *71*, 107–112.
20. Chen, X.D.; Ma, W. A planar quadratic clipping method for computing a root of a polynomial in an interval. *Comput. Graph.* **2015**, *46*, 89–98.
21. Mortenson, M.E. *Geometric Modeling*; Wiley: New York, NY, USA, 1985; pp. 305–317.

22. Oh, Y.T.; Kim, Y.J.; Lee, J.; Kim, M.S.; Elber, G. Efficient point-projection to freeform curves and surfaces. *Comput. Aided Geom. Des.* **2012**, *29*, 242–254.
23. Chen, X.; Zhou, Y.; Shu, Z.; Su, H. Improved Algebraic Algorithm On Point Projection For Bézier Curves. In Proceedings of the Second International Multi-Symposium of Computer and Computational Sciences (IMSCCS 2007), the University of Iowa, Iowa City, IA, USA, 13–15 August 2007.
24. Song, H.C.; Yong, J.H.; Yang, Y.J.; Liu, X.M. Algorithm for orthogonal projection of parametric curves onto B-spline surfaces. *Comput. Aided Des.* **2011**, *43*, 381–393.
25. Sundar, B.R.; Chunduru, A.; Tiwari, R.; Gupta, A.; Muthuganapathy, R. Footpoint distance as a measure of distance computation between curves and surfaces. *Comput. Graph.* **2014**, *38*, 300–309.
26. Wang, X.; Zhang, W.; Huang, X. Computation of point inversion and ray-surface intersection through tracing along the base surface. *Vis. Comput.* **2015**, *31*, 1487–1500.
27. Li, X.; Wang, L.; Wu, Z.; Hou, L.; Liang, J.; Li, Q. Hybrid second-order iterative algorithm for orthogonal projection onto a parametric surface. *Symmetry* **2017**, *9*, 146.
28. Li, K.; Sundin, M.; Rojas, C.R.; Chatterjee, S.; Jansson, M. Alternating strategies with internal ADMM for low-rank matrix reconstruction. *Signal Process.* **2016**, *121*, 153–159.
29. Li, X.; Mu, C.; Ma, J.; Wang, C. Sixteenth-order method for nonlinear equations. *Appl. Math. Comput.* **2010**, *215*, 3754–3758.
30. Li, X.; Mu, C.; Ma, J.; Hou, L. Fifth-order iterative method for finding multiple roots of nonlinear equations. *Numer. Algorith.* **2011**, *57*, 389–398.
31. Li, X.; Xin, Q.; Wu, Z.; Zhang, M.; Zhang, Q. A geometric strategy for computing intersections of two spatial parametric curves. *Vis. Comput.* **2013**, *29*, 1151–1158.
32. Traub, J.F. *Iterative Methods for Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
33. Melman, A. Geometry and Convergence of Euler's and Halley's Methods. *SIAM Rev.* **1997**, *39*, 728–735.
34. Melman, A. Analysis of third order Methods for Secular Equations. *Math. Comput.* **1998**, *67*, 271–286.
35. Traub, J.F. A Class of Globally Convergent Iteration Functions for the Solution of Polynomial Equations. *Math. Comput.* **1966**, *20*, 113–138.
36. Piegl, L.A. Ten challenges in computer-aided design. *Comput. Aided Des.* **2005**, *37*, 461–470.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).