

Article

An Improved Curvature Circle Algorithm for Orthogonal Projection onto a Planar Algebraic Curve

Zhinan Wu ^{1,†} and Xiaowu Li ^{2,*,†} ¹ School of Mathematics and Computer Science, Yichun University, Yichun 336000, China; zhi_nan_7@163.com² College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China

* Correspondence: lixiaowu002@126.com; Tel.: +86-187-8613-2431

† These authors contributed equally to this work.

Received: 10 August 2019; Accepted: 25 September 2019; Published: 1 October 2019



Abstract: Point orthogonal projection onto planar algebraic curve plays an important role in computer graphics, computer aided design, computer aided geometric design and other fields. For the case where the test point \mathbf{p} is very far from the planar algebraic curve, we propose an improved curvature circle algorithm to find the footpoint. Concretely, the first step is to repeatedly iterate algorithm (the Newton's steepest gradient descent method) until the iterated point could fall on the planar algebraic curve. Then seek footpoint by using the algorithm (computing footpoint \mathbf{q}) where the core technology is the curvature circle method. And the next step is to orthogonally project the footpoint \mathbf{q} onto the planar algebraic curve by using the algorithm (the hybrid tangent vertical foot algorithm). Repeatedly run the algorithm (computing footpoint \mathbf{q}) and the algorithm (the hybrid tangent vertical foot algorithm) until the distance between the current footpoint and the previous footpoint is near 0. Furthermore, we propose Second Remedial Algorithm based on Comprehensive Algorithm B. In particular, its robustness is greatly improved than that of Comprehensive Algorithm B and it achieves our expected result. Numerical examples demonstrate that Second Remedial Algorithm could converge accurately and efficiently no matter how far the test point is from the plane algebraic curve and where the initial iteration point is.

Keywords: point projection; intersection; planar algebraic curve; Newton's iterative method; the improved curvature circle algorithm

1. Introduction

Reconstructing curve/surface is an important work in the field of computer aided geometric design, especially in geometric modeling and processing where it is crucial to fit curve/surface in high accuracy and reduce the error of representation curve/surface. The representation of the four curve types are the explicit-type, implicit-type, parametric-type and subdivision-type. Because implicit representation has unique advantage in the process of computer aided geometric design, it has wide and far-reaching applications. From scattered and unorganized three-dimensional data, Bajaj et al. [1] reconstructed surface and functions on surfaces. They [2,3] have constructed the algebraic B-spline surfaces with least-squares fitting feature using tensor product technique. Schulz et al. [4] constructed an enveloping algebraic surface using gradually approximate algebraization method. Kanatani et al. [5] applied the algebraic curve to construct geometric ellipse fitting using unified strict maximum likelihood estimation method. Mullen et al. [6] reconstructed robust and accurate algebraic surface functions to sign the unsigned from scattered and unorganized three-dimensional data point sets. Upreti et al. [7] used a technique to sign algebraic level sets on NURBS surface and algebraic Boolean level sets on NURBS surfaces. Rouhani et al. [8] applied the algebraic function for polynomial

representation system. And L.G. Zagorchev et al. [9] applied the algebraic function for general algebraic surface.

Up to now, there are three main types of methods to solve the problem of point orthogonal projection onto planar algebraic curve: local method, global method and compromise method between these two methods. Here are three typical approaches.

According to the most basic geometric characteristic, orthogonal projection of test point \mathbf{p} onto the planar algebraic curve is actually the point \mathbf{x} on the curve such that cross product of vectors $\vec{\mathbf{x}\mathbf{p}}$ and $\nabla f(\mathbf{x})$ is 0.

$$\nabla f(\mathbf{x}) \times (\mathbf{p} - \mathbf{x}) = 0. \quad (1)$$

Equation (1) can be transformed into Newton's iteration formula (3). Furthermore, Sullivan et al. [10] adopted a hybrid method with Lagrange multiplier and Newton's iterative method to compute the closest point on the planar algebraic curve for each test point. Some orthogonal projection problems can be transformed into solving system of nonlinear equations. The common characteristic of methods [10,11] is that they converge locally and fast, while methods [10,11] are dependent on the initial points.

The first global method of solving system of nonlinear equations is the Homotopy continuous method [12,13]. They constructed Homotopy continuous formula.

$$H(\mathbf{x}, t) = (1 - t)P(\mathbf{x}) + tQ(\mathbf{x}), \quad t \in [0, 1] \quad (2)$$

where t is a parameter of continuous transformation from 0 to 1, $P(\mathbf{x}) = 0$ is the original system of nonlinear equations to be solved, $Q(\mathbf{x})$ is the objective solution of system of nonlinear equations $P(\mathbf{x}) = 0$. All isolated solutions of system of nonlinear equations $P(\mathbf{x}) = 0$ can be computed by the numerical continuous Homotopy methods [12,13]. So the Homotopy methods [12,13] are global convergence. The Homotopy methods' robustness is proved by [14] and their high time-consuming property is verified in [15]. Of course, the Homotopy methods [12,13] are ideal in theory, but it is difficult to find or construct the objective system of nonlinear equations $Q(\mathbf{x}) = 0$ in practical engineering applications.

The second global resultant methods convert system of nonlinear equations into the expression of the resultants and then solve the resultants [16–19]. According to classical elimination theory, system of two nonlinear equations with two variables can be turned into a resultant polynomial with one variable, which is equivalent to the two simultaneous equations. The Sylvester's resultant and Cayley's statement of Bézout's method are the most famous resultant methods [16–19]. Because the resultant methods [16–19] can solve all roots if the degree of the planar algebraic curve is less than 4, they are good global methods. However, if the degree of the planar algebraic curve is more than quintic, it becomes harder and harder with increasing degree to solve two-polynomial system with the resultant methods.

The third global method is the adoption of the Bézier clipping technique [20–22]. In the first step, solving the nonlinear system of Equation (1) is transformed into solving all roots of Bernstein-Bézier representation with convex hull property. In the second step, if the parts of the domains do not include the solution, we clip the parts of the domains by using convex hull box with Bernstein-Bézier form such that the discarded parts of the region has no solution and all the solutions are in the retained parts of the region. In the third step, the de Casteljau subdivision rule is used to segment the remaining part of the curve obtained by elimination in step 2. Repeat steps 2 and 3 until we can find all the solutions to Equation (1). The advantage of this method is that all solutions of Equation (1) can be found. But this global clipping method has one difficulty: sometimes Equation (1) is difficult or even impossible to convert into Bernstein-Bézier form. For example, specific Equation (1) $\nabla f(\mathbf{x}) \times (\mathbf{p} - \mathbf{x}) = -36p_1y^{17} + 36xy^{17} + 6p_2x^5 - 6x^5y - 4p_1x + 4p_2y + 4x^2 - 4y^2$ is impossible to convert into Bernstein-Bézier form where $f(\mathbf{x}) = x^6 + 4xy + 2y^{18} - 1 = 0$.

The compromise method is between local and global methods. Consisting of the geometric property with computing the nearest point is proposed by Hartmann [23,24] named as the first

compromise method. Repeatedly run the Newton’s steepest gradient descent method (3) until the iterative point falls on the planar algebraic curve, where the initial iterative point is unrestricted.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (f(\mathbf{x}_n) / \langle \nabla f(\mathbf{x}_n), \nabla f(\mathbf{x}_n) \rangle) \nabla f(\mathbf{x}_n). \tag{3}$$

$$\mathbf{q} = \mathbf{p} - (\langle \mathbf{p} - \mathbf{y}_n, \nabla f(\mathbf{y}_n) \rangle / \langle \nabla f(\mathbf{y}_n), \nabla f(\mathbf{y}_n) \rangle) \nabla f(\mathbf{y}_n). \tag{4}$$

Running the iterative formula (4) one time, the method [23,24] can obtain the vertical foot point \mathbf{q} where the iterative point \mathbf{y}_n of the formula (4) is the final iterative point obtained by formula (3). Continuously iterate the above two steps until the vertical foot point \mathbf{q} is on the planar algebraic curve $f(\mathbf{x})$. Unluckily, progressive geometric tangent approximation iteration method with computing vertical foot point \mathbf{q} fails for some planar algebraic curves $f(\mathbf{x})$.

The second compromise method is developed by Nicholas [25] who adopted the osculating circle technique to realize orthogonal projection onto the planar algebraic curve. Calculate the corresponding curvature at one point on the planar algebraic curve, and then the radius and center of the curvature circle. The line segment formed by the test point and the center of the curvature circle intersects the curvature circle at footpoint \mathbf{q} . Approximately take the footpoint \mathbf{q} as a point on the planar algebraic curve. For the new point on the planar algebraic curve, repeat the above procedure to get a new footpoint and corresponding new approximate point on the planar algebraic curve. Repeat the above behavior until the footpoint \mathbf{q} is the orthogonal projection point \mathbf{p}_Γ . Because the planar algebraic curve does not have parametric control like parametric curve, taking the footpoint as an approximate point on the planar algebraic curve will bring about large errors. So it makes the operation of the whole algorithm unstable.

The third compromise method is the circle shrinking technique [26]. Repeatedly run the iterative formula (3) such that the final iterative point \mathbf{p}_c falls on the planar algebraic curve as far as possible, where the selection of initial iterative point is arbitrary. The next iterative point on the planar algebraic curve is obtained through a series of combined operations of circle and the planar algebraic curve, where the center and radius of the circle are test point \mathbf{p} and $\|\mathbf{p} - \mathbf{p}_c\|$, respectively. A series of combined operations include the two most important steps: Find a point \mathbf{p}^+ on the circle by means of the mean value theorem; Seek the intersection of the line segment \mathbf{pp}^+ and the circle where we call this intersection as the current intersection point \mathbf{p}_c . Repeatedly run this series of combined operations until the distance between the current point \mathbf{p}_c and the previous point \mathbf{p}_c is 0. The circle shrinking technique [26] takes a lot of time to seek point \mathbf{p}^+ each time. The algorithm has one difficulty: if the degree of the planar algebraic curve is higher than 5, the intersection point \mathbf{p}_c of line segment \mathbf{pp}^+ and the planar algebraic curve cannot be solved directly by formula or the iterative methods to find the intersection \mathbf{p}_c will lead to instability.

The four compromise method is a circle double-and-bisect algorithm [27]. The circle doubling algorithm begins with a very small circle where the center is the test point \mathbf{p} and the radius is very small r_1 . Keep the same center of the circle, take the radius r_2 twice of r_1 to draw a new circle. If there is no intersection between the new circle and the curve, draw a new circle with radius twice of r_2 . Continuously repeat the above process until new circle can intersect with the planar algebraic curve and the former circle does not. Naturally, the former circle and the current circle are called interior circle and exterior circle, respectively. Moreover, the bisecting technology implements the rest of the process. Continue to draw a new circle with new radius $r = (r_1 + r_2)/2$. If the new current circle whose radius is r intersects with the curve, substitute r for r_2 , else for r_1 . Repeatedly run the above progress until the difference between the two radii is approximate zero ($|r_1 - r_2| < \epsilon$). But this method is very difficult to judge whether the exterior circle intersects the planar algebraic curve or not [27].

The fifth compromise method is the integrated hybrid second order algorithm [28]. It includes two sub-algorithms: the hybrid second order algorithm and the initial iterative value estimation algorithm. They mainly exploit three ideas: (1) the tangent orthogonal vertical foot method coupled with calibration method; (2) Newton’s steepest gradient descent iterative method to impel the iteration point

to be on the planar implicit curve; (3) Newton’s iterative method to speed up the whole iteration process. Before running the hybrid second order algorithm, the initial iterative value estimation algorithm is used to force the initial iterative value of the formula (17) of the hybrid second order algorithm and the orthogonal projection point \mathbf{p}_Γ as close as possible. After a lot of tests, if the distance between the test point \mathbf{p} and the curve is not very far, the advantages of this algorithm are obvious in term of robustness and efficiency. But when the test point is very far from the curve, the integrated hybrid second order algorithm is invalid.

2. The Improved Curvature Circle Algorithm

In Reference [28], when the test point \mathbf{p} is not particularly far away, the integrated hybrid algorithm can have ideal result. But if the test point \mathbf{p} is very far from the curve, the algorithm is invalid where the test point \mathbf{p} can not be robustly and effectively orthogonally projected onto the planar algebraic curve. In order to overcome this difficulty, we propose an improved curvature circle algorithm to ensure robustness and effective convergence with the test point \mathbf{p} being arbitrarily far away. No matter how far the test point \mathbf{p} is from the planar algebraic curve, if the initial iteration point \mathbf{x}_0 is very close to the orthogonal projection point of the test point \mathbf{p} , the preconceived algorithm can converge well. So we attempt to construct an algorithm to find an initial iterative point very close to the orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} . The general idea is the following. Repeatedly iterate the formula (3) by utilizing the Newton’s steepest gradient descent method until the iteration point fall on the planar algebraic curve as far as possible, written as \mathbf{p}_c . This time, the distance between the iteration point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ is much smaller than that between the original iteration point \mathbf{x}_0 and the orthogonal projection point \mathbf{p}_Γ . The iteration point \mathbf{p}_c is closer to the orthogonal projection point \mathbf{p}_Γ . In order to further promote the iteration point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ to be closer, we introduce a key step with curvature circle algorithm. Draw a curvature circle through point \mathbf{p}_c on the planar algebraic curve with the radius R determined by the curvature k and the center \mathbf{m} being a normal direction point of point \mathbf{p}_c on the planar algebraic curve. Line segment $\overline{\mathbf{m}\mathbf{p}}$ determined by the test point \mathbf{p} and the center \mathbf{m} intersects curvature circle at point \mathbf{q} . We take the intersection point \mathbf{q} as the next iteration point for the iteration point \mathbf{p}_c . Of course, the distance between the intersection point \mathbf{q} and the orthogonal projection point \mathbf{p}_Γ is much smaller than the previous one. We use the intersection point \mathbf{q} as the new test point, and run the hybrid algorithm again where the initial iterative point at this moment can be set as $\mathbf{q} - (0.1, 0.1)$. Repeatedly iterate until the iteration point falls on the planar algebraic curve $f(\mathbf{x})$, written as \mathbf{p}_c . We repeat the last two key steps in this procedure until the iteration point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ overlap (See Figure 1).

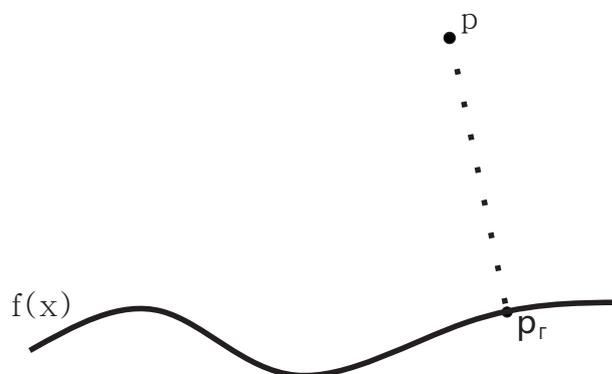


Figure 1. Test point \mathbf{p} orthogonal projection onto planar algebraic curve $f(\mathbf{x})$.

Let’s elaborate on the general idea. Let \mathbf{p} be a test point on the plane. There is an planar algebraic curve Γ on the plane.

$$f(x, y) = 0. \tag{5}$$

The plane algebraic curve (5) can be simply written as

$$f(\mathbf{x}) = 0, \tag{6}$$

where $\mathbf{x} = (x, y)$. The goal of this paper is to find a point \mathbf{p}_Γ on the planar algebraic curve $f(\mathbf{x})$ to satisfy the basic relationship

$$\|\mathbf{p} - \mathbf{p}_\Gamma\| = \min_{\mathbf{x} \in \Gamma} \|\mathbf{p} - \mathbf{x}\|. \tag{7}$$

The above problem can be written as

$$\begin{cases} f(\mathbf{p}_\Gamma) = 0, \\ \nabla f(\mathbf{p}_\Gamma) \times (\mathbf{p} - \mathbf{p}_\Gamma) = 0, \\ \|\mathbf{p} - \mathbf{p}_\Gamma\| = \min_{\mathbf{x} \in \Gamma} \|\mathbf{p} - \mathbf{x}\|, \end{cases} \tag{8}$$

where $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$ is Hamiltonian operator and symbol \times is cross product. We take s as the arc length parameter of the planar algebraic curve $f(\mathbf{x})$ and $\mathbf{t} = \left[\frac{dx}{ds}, \frac{dy}{ds} \right]$ is unit tangent vector along the planar algebraic curve $f(\mathbf{x})$. Take derivative of Equation (6) with respect to arc length parameter s and combine with unit tangent vector condition $\|\mathbf{t}\| = 1$, we obtain the following simultaneous system of nonlinear equations,

$$\begin{cases} \langle \mathbf{t}, \nabla f \rangle = 0, \\ \|\mathbf{t}\| = 1. \end{cases} \tag{9}$$

It is easy to get the solution of Equation (9).

$$\mathbf{t} = \left[-\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x} \right] / \|\nabla f\|. \tag{10}$$

Repeatedly iterate Equation (3) called as the Newton’s steepest gradient descent method until until the iterative termination criteria $|f(\mathbf{x}_{n+1})| < \varepsilon$, where the initial iterative point is $\mathbf{x}_0 = \mathbf{p} - (0.1, 0.1)$ and refer to the iterative point \mathbf{x}_{n+1} as \mathbf{p}_c . The first advantage of the Newton’s steepest gradient descent method (3) is to make the iteration point fall on the planar algebraic curve $f(\mathbf{x})$ as far as possible. Its second advantage of the Newton’s steepest gradient descent method (3) is that the iteration point fallen on the planar algebraic curve is relatively close to the orthogonal projection point \mathbf{p}_Γ , and it brings great convenience to implementation of the subsequent sub-algorithms. The Newton’s steepest gradient descent method (Algorithm 1) can be specifically described as (See Figure 2).

Algorithm 1: The Newton’s steepest gradient descent method.

Input: The test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x}) = 0$

Output: The iterative point \mathbf{p}_c fallen on planar algebraic curve $f(\mathbf{x}) = 0$

Description:

Step 1:

$$\mathbf{x}_{n+1} = \mathbf{p} - (0.1, 0.1);$$

Do {

$$\mathbf{x}_n = \mathbf{x}_{n+1};$$

Update \mathbf{x}_{n+1} according to the iterative Equation (3);

}while ($|f(\mathbf{x}_{n+1})| > \varepsilon \&\& \|\mathbf{x}_{n+1} - \mathbf{x}_n\| > \varepsilon$);

Step 2:

$$\mathbf{p}_c = \mathbf{x}_{n+1};$$

Return \mathbf{p}_c ;

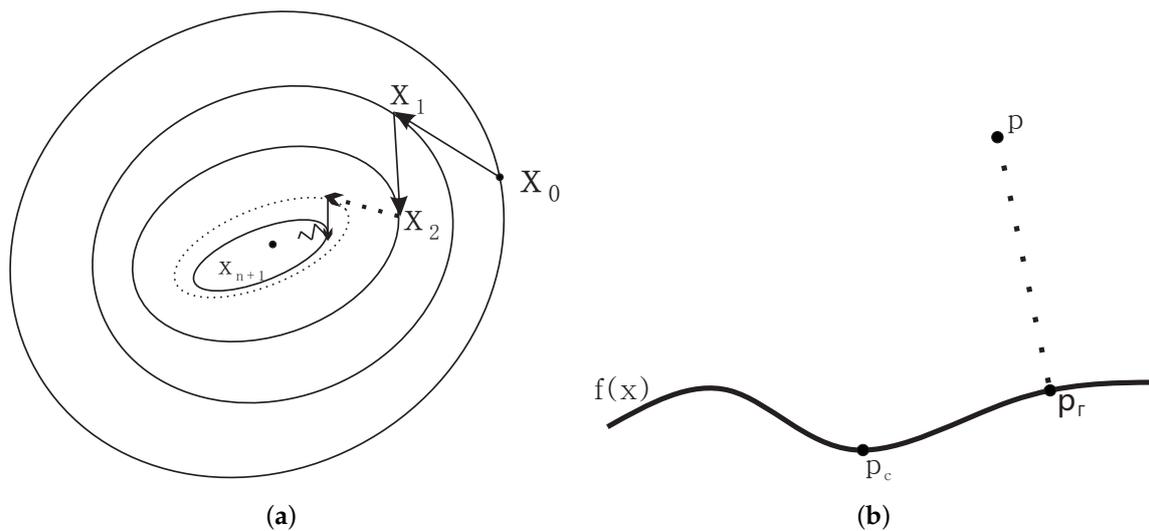


Figure 2. The entire graphic demonstration of Algorithm 1. (a) The whole iterative process of the Newton’s steepest gradient descent method; (b) The last step of the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$ through the Newton’s steepest gradient descent method.

In this case, if the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$ is taken as the initial iterative point of the hybrid algorithm, convergence or divergence may occur where divergence can not improve the algorithm. As for divergence, it can not achieve the purpose of improving the algorithm. From another point of view, the distance between iteration point \mathbf{p}_c and orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} should be closer. It lays a good foundation for the implementation of subsequent sub-algorithms. In order to get the iteration point and the orthogonal projection point \mathbf{p}_Γ being closer, we adopt curvature circle way to promote the iteration point and the orthogonal projection point \mathbf{p}_Γ being closer. Because the iterative point is on the planar algebraic curve, the curvature k at the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$ is defined as [29],

$$k = k(x, y) = \frac{[-f_y, f_x] \mathbf{G} [-f_y, f_x]^T}{\|\nabla f\|^3}, \tag{11}$$

where $\mathbf{G} = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$. The radius R and the center \mathbf{m} of the curvature circle \bigcirc directed by the curvature k are

$$R = |1/k|, \tag{12}$$

and

$$\mathbf{m} = \mathbf{p}_c + \frac{\vec{\mathbf{n}}}{k}, \tag{13}$$

where the unit normal vector $\vec{\mathbf{n}}$ is $\vec{\mathbf{n}} = \frac{\nabla f}{\|\nabla f\|}$. The line segment $\overline{\mathbf{m}\mathbf{p}}$ determined by the test point \mathbf{p} and the center \mathbf{m} of the curvature circle \bigcirc intersects the curvature circle \bigcirc at point \mathbf{q} which is named as footpoint \mathbf{q} . From elementary geometric knowledge, the parametric equation of the line segment $\overline{\mathbf{m}\mathbf{p}}$ can be expressed as

$$\mathbf{x} = \mathbf{p} + (\mathbf{m} - \mathbf{p})w, \tag{14}$$

where parametric $0 \leq w \leq 1$ is undetermined. In addition, the equation of the curvature circle \bigcirc can be written as

$$\|\mathbf{m} - \mathbf{x}\| = R. \tag{15}$$

By solving Equation (14) and Equation (15) together, the analytic expression of the intersection \mathbf{q} is obtained

$$\mathbf{q} = \mathbf{p} + (\mathbf{m} - \mathbf{p})w, \tag{16}$$

where the undetermined parameter w is accurately identified as $w = 1 - \frac{R}{\|\mathbf{m} - \mathbf{p}\|}$. The computation of the footpoint \mathbf{q} can be realized through Algorithm 2 (See Figure 3).

Algorithm 2: Computing footpoint \mathbf{q} via the curvature circle \bigcirc and the line segment $\overline{\mathbf{m}\mathbf{p}}$.

Input: The test point \mathbf{p} , the planar algebraic curve $f(\mathbf{x}) = 0$ and the iterative point \mathbf{p}_c on the planar algebraic curve $f(\mathbf{x}) = 0$.

Output: The footpoint \mathbf{q} .

Description:

Step 1:

Compute the curvature k of the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x}) = 0$ by the curvature calculation formula (11).

Step 2:

Calculate the radius R and the center \mathbf{m} of the curvature circle \bigcirc through the formulas (12) and (13), respectively.

Step 3:

Compute the footpoint \mathbf{q} by the formula (16).

Return \mathbf{q} ;

Remark 1. The important formula for computing the curvature k is the formula (11). If the denominator of the curvature k with the formula (11) is 0, the whole iteration process will degenerate. In order to solve this special degeneration, we adopt a small perturbation of the curvature k of the formula (11) in programming implementation of Algorithm 2. Namely, the denominator of the curvature k with the formula (11) could be incremented by a small positive constant ε , the denominator of the curvature k is the denominator of the curvature $k + \varepsilon$, and Algorithm 2 continues to calculate the center and the radius of the curvature circle corresponding to the curvature after disturbance. Of course, in all subsequent formulas or iterative formulas, we also do the same denominators perturbation treatment for the case of the zero denominators of the formulas or the iterative formulas.

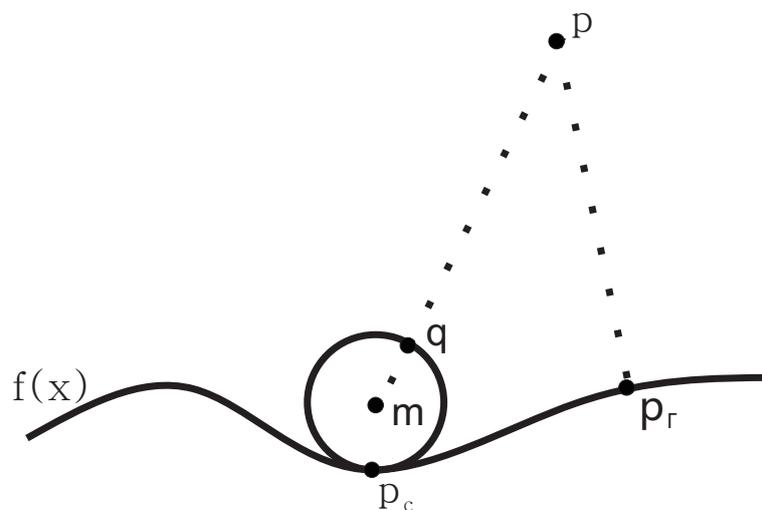


Figure 3. Graphic demonstration for Algorithm 2.

Under this circumstance, if the footpoint point \mathbf{q} at this moment is taken as the initial iteration point of the hybrid algorithm, the convergence probability of the hybrid algorithm is much greater than

that of using the point \mathbf{p}_c in Algorithm 1 as the initial iterative point of the hybrid algorithm. The reason is that the distance $\|\mathbf{q} - \mathbf{p}_\Gamma\|$ is smaller than the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$. But divergence may happen in this case. In order to further guarantee the robustness, we orthogonally project the footpoint \mathbf{q} onto the planar algebraic curve $f(\mathbf{x})$ by using the hybrid algorithm, instead of directly using the footpoint \mathbf{q} as the initial iterative point. At this time we still call the orthogonal projection point of the footpoint \mathbf{q} as the point \mathbf{p}_c which is just fallen on the planar algebraic curve $f(\mathbf{x})$. Because at this time the footpoint \mathbf{q} is close to the planar algebraic curve $f(\mathbf{x})$, the algorithm can ensure complete convergence. The distance between the iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} becomes smaller again. The core iterative formula (17) of the hybrid algorithm is as follows (See [28]).

$$\begin{cases} \mathbf{y}_n = \mathbf{x}_n - (f(\mathbf{x}_n) / \langle \nabla f(\mathbf{x}_n), \nabla f(\mathbf{x}_n) \rangle) \nabla f(\mathbf{x}_n), \\ \mathbf{z}_n = \mathbf{y}_n - (\mathbf{F}(\mathbf{y}_n) / \langle \nabla \mathbf{F}(\mathbf{y}_n), \nabla \mathbf{F}(\mathbf{y}_n) \rangle) \nabla \mathbf{F}(\mathbf{y}_n), \\ \mathbf{Q} = \mathbf{q} - (\langle (\mathbf{q} - \mathbf{z}_n), \nabla f(\mathbf{z}_n) \rangle / \langle \nabla f(\mathbf{z}_n), \nabla f(\mathbf{z}_n) \rangle) \nabla f(\mathbf{z}_n), \\ \mathbf{u}_n = \mathbf{z}_n + \text{sign}(\langle \mathbf{q} - \mathbf{z}_n, \mathbf{t}_0 \rangle) \mathbf{t}_0 \Delta s, \\ \mathbf{v}_n = \mathbf{u}_n - (\mathbf{F}(\mathbf{u}_n) / \langle \nabla \mathbf{F}(\mathbf{u}_n), \nabla \mathbf{F}(\mathbf{u}_n) \rangle) \nabla \mathbf{F}(\mathbf{u}_n), \\ \mathbf{x}_{n+1} = \mathbf{v}_n + [-\Delta e, 0] \left[\nabla f^T, (\Delta \mathbf{v}_n)^T \right]^{-1} \quad (\text{if } |[\nabla f^T, (\Delta \mathbf{v}_n)^T]| = 0, \mathbf{x}_{n+1} = \mathbf{v}_n), \end{cases} \tag{17}$$

where $\mathbf{F}_0(\mathbf{x}) = [(\mathbf{q} - \mathbf{x}) \times \nabla f(\mathbf{x})] = 0$, $\mathbf{F}(\mathbf{x}) = \frac{\mathbf{F}_0(\mathbf{x})}{\sqrt{\langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle}}$, $\mathbf{t}_0 = \frac{[-f_y, f_x]}{\|\Delta f\|}$, $\Delta s = \|\mathbf{Q} - \mathbf{z}_n\|$, $f(\mathbf{v}_n) = \Delta e$, $\Delta \mathbf{v}_n = -(\mathbf{F}(\mathbf{u}_n) / \langle \nabla \mathbf{F}(\mathbf{u}_n), \nabla \mathbf{F}(\mathbf{u}_n) \rangle) \nabla \mathbf{F}(\mathbf{u}_n)$.

The iterative formula (17) mainly contains four techniques. The core technology is the tangent foot vertical method with the third step and the fourth step of the iterative formula (17). Draw a tangent line \mathbf{L} from a point on a plane algebraic curve $f(\mathbf{x})$. Through the footpoint \mathbf{q} (The footpoint \mathbf{q} at this time is as the test point of iterative formula (17)), make a vertical line of the tangent \mathbf{L} and get its corresponding vertical foot point \mathbf{Q} , which is equivalent to the third step in the formula (17). From the fourth step of the iterative formula (17), we get the next iteration point of particular importance for the initial iteration point. When the next iteration point is not very close to the planar algebraic curve $f(\mathbf{x})$, we adopt the second important technique with the iteration point correction method, equivalent to the sixth step of the iterative formula (17). The iteration point is to move to the plane algebra curve as close as possible such that the distance between the correction point of the iteration point and the plane algebra curve $f(\mathbf{x})$ is as close as possible. These two techniques are pure geometric techniques. When the distance between the test point and the planar algebraic curve is very close, the effect of convergence is obvious. Of course, when the distance between the test point and the planar algebraic curve is relatively long, sometimes there will be non-convergence. In order to improve the robustness of convergence, we add the Newton's steepest gradient descent method before the first technique with the third step and the fourth step of the iterative formula (17). Its first aim is to bring the initial iteration point closer to the planar algebraic curve $f(\mathbf{x})$. Its second aim is to promote the accuracy of subsequent iterations. In order to accelerate the whole iteration process of the iterative formula (17), we once again incorporate the fourth technology of Newton's iterative method which is closely related to the footpoint \mathbf{q} . This technique not only accelerates the convergence rate of the whole iteration process but also improves the iteration robustness. Furthermore, the accuracy of the whole iteration process can be improved by the fourth technique. So we add Newton's iterative method after the first step with the second technique, and then add it again before the last step with the third technique. Based on the above explanation and illustration, we get the following the hybrid tangent vertical foot algorithm (Algorithm 3).

Algorithm 3: The hybrid tangent vertical foot algorithm (See Figure 4).

Input: The footpoint \mathbf{q} and the planar algebraic curve $f(\mathbf{x}) = 0$.

Output: The point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x}) = 0$.

Description:

Step 1:

$$\mathbf{x}_{n+1} = \mathbf{q} - (0.1, 0.1);$$

Do {

$$\mathbf{x}_n = \mathbf{x}_{n+1};$$

Execute \mathbf{x}_{n+1} according to the iterative Equation (17);

}while ($\|\mathbf{x}_{n+1} - \mathbf{x}_n\|^2 > \epsilon \&\& |f(\mathbf{x}_{n+1})| > \epsilon$)

Step 2:

$$\mathbf{p}_c = \mathbf{x}_{n+1};$$

Return \mathbf{p}_c ;

With the description of the above three algorithms, we propose a comprehensive and complete algorithm (Algorithm 4) closely related to Algorithm 2 (See Figure 4).

Algorithm 4: The first improved curvature circle algorithm (Comprehensive Algorithm A).

Input: Test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$.

Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} which orthogonally projects the test point \mathbf{p} onto a planar algebraic curve $f(\mathbf{x})$.

Description:

Step 1: Starting from the neighbor point of the test point \mathbf{p} , calculate the point \mathbf{p}_c fallen on the $f(\mathbf{x})$ via Algorithm 1.

Do{

Step 2: Compute the footpoint \mathbf{q} via Algorithm 2.

Step 3: Project footpoint \mathbf{q} onto the planar algebraic curve $f(\mathbf{x})$ via Algorithm 3, then get the new iterative point \mathbf{p}_c fallen on the $f(\mathbf{x})$.

}while (distance (the current \mathbf{p}_c , the previous \mathbf{p}_c) $> \epsilon$).

$$\mathbf{p}_\Gamma = \mathbf{p}_c;$$

Return \mathbf{p}_Γ ;

Through a series of rigorous deductions, Comprehensive Algorithm A is the important algorithm of our paper. No matter how far the test point \mathbf{p} is from the planar algebraic curve $f(\mathbf{x})$, test point \mathbf{p} could very robustly orthogonally projects onto the planar algebraic curve $f(\mathbf{x})$. This has achieved our desired result. After a lot of testing and observation, when the point on the curve is close to the orthogonal projection point, we find that Comprehensive Algorithm A presents two characteristics: (1) difference between the first distance and the second distance decreases slower and slower, where the first distance and the second distance are the one between the previous iterative point \mathbf{p}_c on the planar algebraic curve and the orthogonal projection point \mathbf{p}_Γ , and the one between the current iterative point \mathbf{p}_c on the planar algebraic curve and the orthogonal projection point \mathbf{p}_Γ , respectively; (2) the rate goes even slower at which the absolute value of the inner product gradually approaches zero. These two characteristics are what we don't want to obtain because they are contrary to the efficiency of computer systems. On the premise of ensuring robustness, we try our best to improve and excavate a certain degree of efficiency for the problem of point orthogonal projection onto planar algebraic curve. We have an ingenious discovery. After each running of Algorithm 3, we run the Newton's iterative method associated with the original test point \mathbf{p} , which can improve the convergence and ensure the orthogonality. Namely, that is to add this step after the last step of the formula (17). Thus the iterative formula (18) is obtained.

$$\begin{cases} \mathbf{y}_n = \mathbf{x}_n - (f(\mathbf{x}_n) / \langle \nabla f(\mathbf{x}_n), \nabla f(\mathbf{x}_n) \rangle) \nabla f(\mathbf{x}_n), \\ \mathbf{z}_n = \mathbf{y}_n - (\mathbf{F}(\mathbf{y}_n) / \langle \nabla \mathbf{F}(\mathbf{y}_n), \nabla \mathbf{F}(\mathbf{y}_n) \rangle) \nabla \mathbf{F}(\mathbf{y}_n), \\ \mathbf{Q} = \mathbf{q} - (\langle (\mathbf{q} - \mathbf{z}_n), \nabla f(\mathbf{z}_n) \rangle / \langle \nabla f(\mathbf{z}_n), \nabla f(\mathbf{z}_n) \rangle) \nabla f(\mathbf{z}_n), \\ \mathbf{u}_n = \mathbf{z}_n + \text{sign}(\langle \mathbf{q} - \mathbf{z}_n, \mathbf{t}_0 \rangle) \mathbf{t}_0 \Delta s, \\ \mathbf{v}_n = \mathbf{u}_n - (\mathbf{F}(\mathbf{u}_n) / \langle \nabla \mathbf{F}(\mathbf{u}_n), \nabla \mathbf{F}(\mathbf{u}_n) \rangle) \nabla \mathbf{F}(\mathbf{u}_n), \\ \mathbf{w}_n = \mathbf{v}_n + [-\Delta e, 0] \left[\nabla f^T, (\Delta \mathbf{v}_n)^T \right]^{-1} \text{ (if } |[\nabla f^T, (\Delta \mathbf{v}_n)^T]| = 0, \text{ then } \mathbf{w}_n = \mathbf{v}_n), \\ \mathbf{x}_{n+1} = \mathbf{w}_n - (\mathbf{G}(\mathbf{w}_n) / \langle \nabla \mathbf{G}(\mathbf{w}_n), \nabla \mathbf{G}(\mathbf{w}_n) \rangle) \nabla \mathbf{G}(\mathbf{w}_n), \end{cases} \tag{18}$$

where $\mathbf{F}_0(\mathbf{x}) = [(\mathbf{q} - \mathbf{x}) \times \nabla f(\mathbf{x})] = 0$, $\mathbf{F}(\mathbf{x}) = \frac{\mathbf{F}_0(\mathbf{x})}{\sqrt{\langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle}}$, $\mathbf{t}_0 = \frac{[-f_y, f_x]}{\|\Delta f\|}$, $\Delta s = \|\mathbf{Q} - \mathbf{z}_n\|$, $f(\mathbf{v}_n) = \Delta e$, $\Delta \mathbf{v}_n = -(\mathbf{F}(\mathbf{u}_n) / \langle \nabla \mathbf{F}(\mathbf{u}_n), \nabla \mathbf{F}(\mathbf{u}_n) \rangle) \nabla \mathbf{F}(\mathbf{u}_n)$, $\mathbf{G}_0(\mathbf{x}) = [(\mathbf{p} - \mathbf{x}) \times \nabla f(\mathbf{x})] = 0$, $\mathbf{G}(\mathbf{x}) = \frac{\mathbf{G}_0(\mathbf{x})}{\sqrt{\langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle}}$. Because the iterative formula (17) of Algorithm 3 naturally becomes the iterative formula (18), so Algorithm 3 naturally becomes the following Algorithm 5.

Algorithm 5: The hybrid tangent vertical foot algorithm.

Input: The footpoint \mathbf{q} and the planar algebraic curve $f(\mathbf{x}) = 0$.

Output: The point \mathbf{p}_c fallen on planar algebraic curve $f(\mathbf{x}) = 0$.

Description:

Step 1:

$\mathbf{x}_{n+1} = \mathbf{q} - (0.1, 0.1)$;

Do {

$\mathbf{x}_n = \mathbf{x}_{n+1}$;

Execute \mathbf{x}_{n+1} according to the iterative Equation (18);

}while ($\|\mathbf{x}_{n+1} - \mathbf{x}_n\|^2 > \varepsilon \&\& |f(\mathbf{x}_{n+1})| > \varepsilon$)

Step 2:

$\mathbf{p}_c = \mathbf{x}_{n+1}$;

Return \mathbf{p}_c ;

Now let's replace Algorithm 3 of Comprehensive Algorithms A with Algorithm 5. We get the following Comprehensive Algorithm B (Algorithm 6).

Algorithm 6: The second improved curvature circle algorithm (Comprehensive Algorithm B).

Input: Test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$.

Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} which orthogonally projects the test point \mathbf{p} onto the planar algebraic curve $f(\mathbf{x})$.

Description:

Step 1: Starting from the neighbor point of the test point \mathbf{p} , calculate the point \mathbf{p}_c fallen on the $f(\mathbf{x})$ via Algorithm 1.

Do{

Step 2: Compute the footpoint \mathbf{q} via Algorithm 2.

Step 3: Project footpoint \mathbf{q} onto the planar algebraic curve $f(\mathbf{x})$ via Algorithm 5, then get new point \mathbf{p}_c fallen on the $f(\mathbf{x})$.

}while(distance(the current \mathbf{p}_c , the previous \mathbf{p}_c) > ε).

$\mathbf{p}_\Gamma = \mathbf{p}_c$;

Return \mathbf{p}_Γ ;

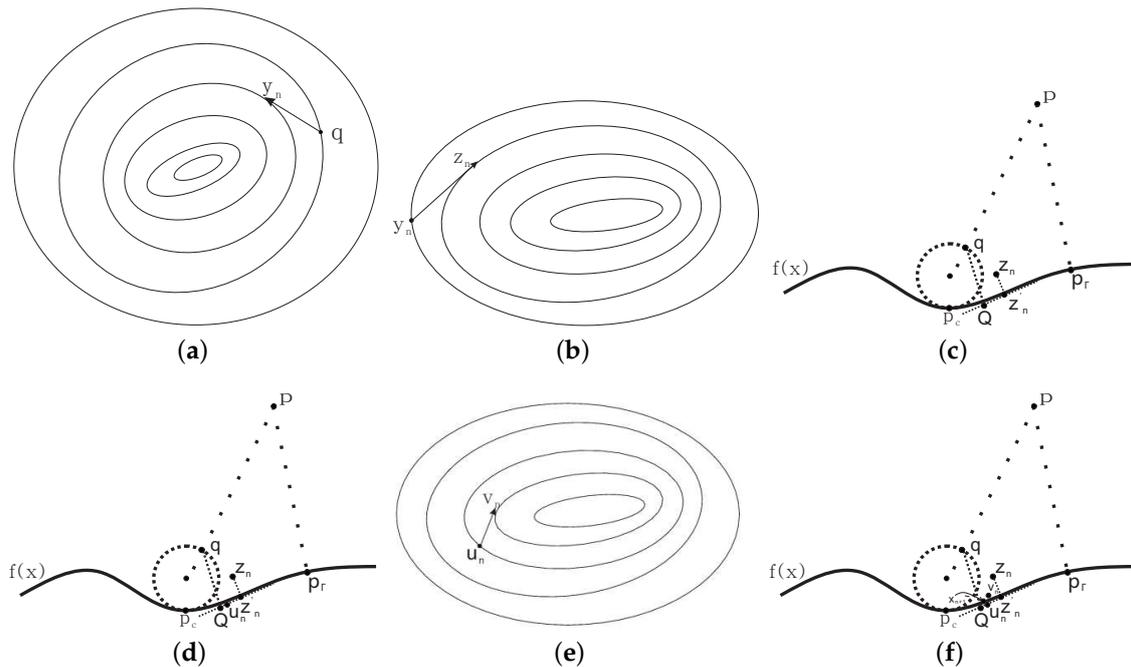


Figure 4. Graphic interpretation of the whole iteration process in Algorithm 3. (a) Newton’s steepest gradient descent method in the first step; (b) The Newton’s iteration method related to the test point in the second step; (c) The vertical foot point Q being the footpoint q orthogonal projection onto tangent line induced by the iterative point z_n on the planar algebraic curve in the third step; (d) Calculating line incremental iterative value in the fourth step; (e) Once again running the Newton’s iteration method related to the test point in the fifth step; (f) Correcting the previous iteration value to improve the robustness of iteration in the last step.

Comprehensive Algorithm A and Comprehensive Algorithm B share common advantage: the robustness of the two algorithms is substantially improved than that of the existing algorithms because our algorithms are not subject to any restrictions on test points and initial iteration points. By comparison, Comprehensive Algorithm B has four advantages over Comprehensive Algorithm A. (1) The last step of the iterative formula (18) in Comprehensive Algorithm B can make corrections continuously; (2) The last step of the iterative formula (18) in Comprehensive Algorithm B accelerates the whole Comprehensive Algorithm B; (3) The last step of the iterative formula (18) in Comprehensive Algorithm B accelerates the inner product of two vectors to 0, where the first vector refers to the vector connecting the test point p and the iteration point z_{n+1} of Comprehensive Algorithm B and the second vector $\left[-\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}\right]_{x=x_{n+1}}$ is the tangent vector derived from the iteration point x_{n+1} on the planar algebraic curve, respectively; (4) Comprehensive Algorithm B overcomes two shortcomings of Comprehensive Algorithm A.

Of course, when the test point is not too far from the plane algebra curve, Comprehensive Algorithm is also convergent for any initial iterative point. However, Comprehensive Algorithm A takes more time than directly using the hybrid second order algorithm. In practical applications such as computer graphics, it’s hard to know if the test point p is close to or far from a planar algebraic curve. Because the main reason is that the degree and the type of the planar algebraic curve restrict the relative distance between the test point p and the planar algebraic curve. In order to optimize time efficiency, we take advantage of Comprehensive Algorithm A and the hybrid second order algorithm such that no matter where the test point p is located, it can be orthogonally projected onto the planar algebraic curve efficiently and robustly. First, the hybrid second order algorithm is iterated. If it does not converge after 100 iterations, it will be changed to Comprehensive Algorithm A to iterate until the

iteration point reaches the orthogonal projection point \mathbf{p}_Γ . Specific algorithm implementation is the following Comprehensive Integrated Algorithm A (Algorithm 7).

Algorithm 7: The first comprehensive integrated improved curvature circle algorithm (Comprehensive Integrated Algorithm A).

Input: Test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$.

Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} .

Description:

Step 1:

```

 $\mathbf{x}_{n+1} = \mathbf{p} - (0.1, 0.1);$ 
for( $i = 0; i < N; i++$ ) {
     $\mathbf{x}_n = \mathbf{x}_{n+1};$ 
     $\mathbf{x}_{n+1} = \text{Hybrid second order algorithm}(f, \mathbf{p}, \mathbf{x}_n);$ 
    if( $\|\mathbf{x}_{n+1} - \mathbf{x}_n\| < \varepsilon$ ) break;
}

```

Step 2:

```

if( $i \geq N \&\& d \geq 1e - 15$ ) {
     $\mathbf{x}_n = \mathbf{x}_{n+1};$ 
     $\mathbf{x}_{n+1} = \text{Comprehensive Algorithm A}(f, \mathbf{p}, \mathbf{x}_n);$ 
}
 $\mathbf{p}_\Gamma = \mathbf{x}_{n+1};$ 
Return  $\mathbf{p}_\Gamma;$ 

```

Number N is an empirical value of the iterative times where the value N is specified as 5 or 6.

Similar to Comprehensive Algorithm A, by replacing Algorithm 3 with Algorithm 5, the following Comprehensive Integrated Algorithm B (Algorithm 8) can be obtained naturally.

Algorithm 8: The second comprehensive integrated improved curvature circle algorithm (Comprehensive Integrated Algorithm B).

Input: The test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$.

Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} .

Description:

Step 1:

```

 $\mathbf{x}_{n+1} = \mathbf{p} - (0.1, 0.1);$ 
for( $i = 0; i < N; i++$ ) {
     $\mathbf{x}_n = \mathbf{x}_{n+1};$ 
     $\mathbf{x}_{n+1} = \text{Hybrid second order algorithm}(f, \mathbf{p}, \mathbf{x}_n);$ 
    if( $\|\mathbf{x}_{n+1} - \mathbf{x}_n\| < \varepsilon$ ) break;
}

```

Step 2:

```

if( $i \geq N \&\& d \geq 1e - 15$ ) {
     $\mathbf{x}_n = \mathbf{x}_{n+1};$ 
     $\mathbf{x}_{n+1} = \text{Comprehensive Algorithm B}(f, \mathbf{p}, \mathbf{x}_n);$ 
}
 $\mathbf{p}_\Gamma = \mathbf{x}_{n+1};$ 
Return  $\mathbf{p}_\Gamma;$ 

```

Number N is an empirical value of the iterative times where the value N is specified as 5 or 6.

To sum up, we have presented four synthesis algorithms altogether. After analysis and judgment, Comprehensive Algorithm B and Comprehensive Integrated Algorithm B are the most robust and efficient. On the problem of orthogonal projection of point onto planar algebraic curve, if the distance between the test point and the planar algebraic curve is close, we recommend the hybrid second order algorithm, if the distance between the test point and the planar algebraic curve is not close, we recommend Comprehensive Algorithm B. Of course, if the distance between the test point and the planar algebraic curve cannot be known to be very far or close, Comprehensive Integrated Algorithm B is the best choice.

Remark 2. In sum, Comprehensive Algorithm B has strong superiority over existing algorithms [10–28]. If the distance between the test point and the planar algebraic curve is very far away, the test point can be ideally orthogonally projected onto the planar algebraic curve. But when there are singular points $\frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot \frac{\partial f}{\partial y} = 0$ in the planar algebraic curve, this case will seriously hinder the correct execution and implementation of Comprehensive Algorithm B. In order to solve the problem in the case of singularities in the planar algebraic curves, we propose a remedy to Comprehensive Algorithm B (Algorithm 9). The specific description is as follows (See Figure 5).

Algorithm 9: The first remedial algorithm of Comprehensive Algorithm B.

Input: Test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$.

Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} .

Description:

Step 1.

Starting from the neighbor point of the test point \mathbf{p} , calculate the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$ via Algorithm 1.

Step 2.

Judge whether to use curvature circle method or tangent method in the next step.

Step 3.

Find the left endpoint \mathbf{L}_0 on the other side of $f(\mathbf{x})$ relative to the test point \mathbf{p} . According to the result of **step 2**, if use curvature circle method, then the left endpoint \mathbf{L}_0 is equal to the intersection point \mathbf{q} which is computed by the curvature circle method with the formula (16). If not, then the left endpoint \mathbf{L}_0 is equal to the vertical foot \mathbf{Q} which is computed by the tangent method with the third step of the formula (17).

Step 4.

Calculate the intersection point \mathbf{p}_c of the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ connecting the current left endpoint \mathbf{L}_0 and the test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$ by the hybrid method of combining Newton’s iterative method and binary search method. The intersection point \mathbf{p}_c is called as the current iterative point \mathbf{p}_c ;

Step 5.

Repeat **Step 2**, **Step 3** and **Step 4** until the distance between the current iterative point \mathbf{p}_c and the previous iterative point \mathbf{p}_c is near zero;

Step 6.

$\mathbf{p}_\Gamma = \mathbf{p}_c$;
Return \mathbf{p}_Γ ;

Now let’s describe the hybrid method of combining Newton’s iterative method and binary search method in detail. The parameter equation of the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ can be expressed as

$$\begin{cases} x = L_1 + (p_1 - L_1)w, \\ y = L_2 + (p_2 - L_2)w, \end{cases} \tag{19}$$

where $\mathbf{L}_0 = (L_1, L_2)$, $\mathbf{p} = (p_1, p_2)$, and $0 \leq w \leq 1$ is a parameter of Equation (19). Substitute Equation (19) into Equation (6) of the planar algebraic curve to get an equation on the parameter w ,

$$K(w) = f(x, y) = 0, \quad (20)$$

where the x and y of Equation (20) are completely determined by the x and y of Equation (19). So the most basic Newton's iterative formula corresponding to Equation (20) is not difficult to write as,

$$w_{n+1} = w_n - \frac{K(w_n)}{DK(w_n)}, \quad (21)$$

where $DK(w)$ is the first derivative of $K(w)$ about the parameter w . Now we start to iterate the Newton's iterative formula (21) with the initial iterative value $w_0 = 0.0$. Based on the actual situation, the intersection of the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ and the planar algebraic curve is much closer to the left endpoint \mathbf{L}_0 and much farther from the original test point \mathbf{p} , therefore, the initial interval of the binary search method can be specified as $[a, b] = [0.0, 0.5]$. The detailed description of the hybrid method of combining Newton's iterative method and binary search method is as following Algorithm 10.

Algorithm 10: The hybrid method of combining Newton's iterative method and binary search method.

Input: The planar algebraic curve $f(\mathbf{x})$, the original test point $\mathbf{p} = (p_1, p_2)$, the iterative point \mathbf{p}_c via Algorithm 1.

Output: The intersection \mathbf{p}_c between the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ and the planar algebraic curve $f(\mathbf{x})$.

Description:

Step 1:

The initial interval of the binary search method $[a, b] = [0.0, 0.5]$, the initial iterative value $w = 0.0$;

Step 2:

```

 $w = w - K(w) / DK(w);$ 
 $kmin = \min(K(a), K(b));$ 
 $kmax = \max(K(a), K(b));$ 
if ( $K(w) < kmin$  or  $K(w) > kmax$ )
     $w = (a + b) / 2;$ 
 $sa = \text{sign}(K(a));$ 
 $sw = \text{sign}(K(w));$ 
if ( $sa == sw$ )
     $a = w;$ 
else
     $b = w;$ 

```

Step 3:

Repeatedly iterate **Step 2** until $|a - b| < \varepsilon$;

Step 4:

```

 $\mathbf{p}_c = \mathbf{L}_0 + (\mathbf{p} - \mathbf{L}_0)w;$ 
Return  $\mathbf{p}_c;$ 

```

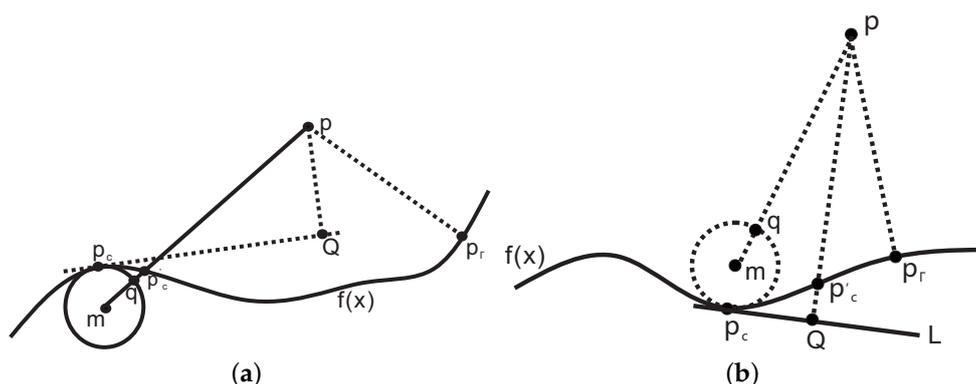


Figure 5. Graphical interpretation for the first remedial algorithm of Comprehensive Algorithm B. (a) The intersection point q between the line segment \overline{mp} and the curvature circle and the test point p on the opposite side of the planar algebraic curve $f(x)$; (b) The vertical foot Q of the tangential line L and the test point p on the opposite side of the planar algebraic curve $f(x)$.

The robustness of the first remedial algorithm of Comprehensive Algorithm B is much better than that of Comprehensive Algorithm B while the first remedial algorithm of Comprehensive Algorithm B takes much more time than Comprehensive Algorithm B. The hybrid method of combining Newton’s iterative method and binary search method is a hybrid method which binary search method ensures global convergence and the Newton’s iterative method plays an accelerating role. In order to ensure robustness and improve efficiency, we have fully excavated Comprehensive Algorithm B. We have developed Second Remedial Algorithm (Algorithm 11). The specific description is as follows (See Figure 6).

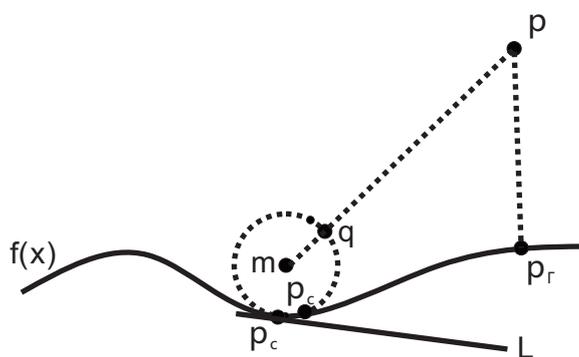


Figure 6. Graphic demonstration for Second Remedial Algorithm.

Algorithm 11: Second Remedial Algorithm.

Input: Test point p and the planar algebraic curve $f(x)$.

Output: Orthogonal projection point p_r of the test point p which orthogonally projects the test point p onto the planar algebraic curve $f(x)$

Description:

Step 1: Starting from a certain percentage of the test point p , calculate the point p_c fallen on the $f(x)$ via Algorithm 1.

Do{

Step 2: Compute the footpoint q via Algorithm 2.

Step 3: Starting from the footpoint q , compute the iterative point p_c fallen on the $f(x)$ via Algorithm 1.

}while(distance(the current p_c , the previous p_c) > ϵ).

Step 4: Compute the orthogonal projection point p_r of the test point p via Algorithm 12.

Return p_r ;

Algorithm 12: The hybrid Newton-type iterative algorithm.

Input: The current iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$ and the planar algebraic curve $f(\mathbf{x})$.
Output: Orthogonal projection point \mathbf{p}_Γ of the test point \mathbf{p} which orthogonally projects the test point \mathbf{p} onto the planar algebraic curve $f(\mathbf{x})$.
Description:
Step 1:
 $\mathbf{x}_{n+1} = \mathbf{p}_c$;
 Do {
 $\mathbf{x}_n = \mathbf{x}_{n+1}$;
 Compute \mathbf{x}_{n+1} according to the iterative formula (22);
 }while ($\|\mathbf{x}_{n+1} - \mathbf{x}_n\|^2 > \varepsilon$ && $|f(\mathbf{x}_{n+1})| > \varepsilon$)
Step 2:
 $\mathbf{p}_\Gamma = \mathbf{x}_{n+1}$;
 Return \mathbf{p}_Γ ;

The expression of the iterative formula (22) is as follow,

$$\begin{cases} \mathbf{y}_n = \mathbf{x}_n - (f(\mathbf{x}_n) / \langle \nabla f(\mathbf{x}_n), \nabla f(\mathbf{x}_n) \rangle) \nabla f(\mathbf{x}_n), \\ \mathbf{z}_n = \mathbf{y}_n - (\mathbf{F}(\mathbf{y}_n) / \langle \nabla \mathbf{F}(\mathbf{y}_n), \nabla \mathbf{F}(\mathbf{y}_n) \rangle) \nabla \mathbf{F}(\mathbf{y}_n), \\ \mathbf{x}_{n+1} = \mathbf{z}_n + [-\Delta e, 0] [\nabla f^T, (\Delta \mathbf{z}_n)^T]^{-1} \text{ (if } |\langle \nabla f^T, (\Delta \mathbf{z}_n)^T \rangle| = 0, \text{ then } \mathbf{x}_{n+1} = \mathbf{z}_n), \end{cases} \quad (22)$$

where $\mathbf{F}_0(\mathbf{x}) = [(\mathbf{p} - \mathbf{x}) \times \nabla f(\mathbf{x})] = 0$, $\mathbf{F}(\mathbf{x}) = \frac{\mathbf{F}_0(\mathbf{x})}{\sqrt{\langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle}}$, $f(\mathbf{z}_n) = \Delta e$, $\Delta \mathbf{z}_n = -(\mathbf{F}(\mathbf{y}_n) / \langle \nabla \mathbf{F}(\mathbf{y}_n), \nabla \mathbf{F}(\mathbf{y}_n) \rangle) \nabla \mathbf{F}(\mathbf{y}_n)$.

Remark 3. In this remark, we present the geometric interpretation of Second Remedial Algorithm. The purpose of the first step is to make the iterative point \mathbf{p}_c fall on the planar algebraic curve as much as possible through Newton’s steepest gradient descent method of Algorithm 1, where the coordinates of the initial iterative point take proportional value of that of the test point \mathbf{p} to ensure that Algorithm 1 converges successfully. Otherwise, the distance between the initial iterative point and the planar algebraic curve is very large, which easily leads to the divergence of Algorithm 1. The purpose of Do ... While cycle body in Second Remedial Algorithm is to continuously and gradually move the iterative point \mathbf{p}_c to fall on the planar algebraic curve to the orthogonal projection point \mathbf{p}_Γ . The second step in Do ... While cycle body in Second Remedial Algorithm has two characteristics. Since the footpoint \mathbf{q} is the unique intersection point of the curvature circle and the straight line segment $\overline{m\mathbf{p}}$ connecting the centre \mathbf{m} of the curvature circle and the test point \mathbf{q} , the footpoint \mathbf{q} is always closely related to the iterative point \mathbf{p}_c fallen on the planar algebraic curve and the test point \mathbf{p} . The first characteristic is that the footpoint \mathbf{q} can guarantee the global convergence of the whole algorithm (Second Remedial Algorithm). The second characteristic is that the distance between the footpoint \mathbf{q} and the planar algebraic curve is much smaller than the distance between the test point \mathbf{p} and the planar algebraic curve. So the third step with Algorithm 1 in Do ... While cycle body can very robustly iterate the footpoint \mathbf{q} onto the planar algebraic curve. The core thought of Do ... While cycle body in Second Remedial Algorithm is to keep the iterative point \mathbf{p}_c to fall on the planar algebraic curve and to move towards the orthogonal projection point \mathbf{p}_Γ such that the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ becomes smaller and smaller. As the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ gets smaller and smaller, we have found that there is a defect in Do ... While cycle body in Second Remedial Algorithm. The decreasing speed of the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ is getting slower and slower. Especially the second formula of the formula (8) is very difficult to be satisfied. Namely, it is difficult to orthogonalize the vector $\overrightarrow{p\mathbf{p}_c}$ and the vector $\nabla f(\mathbf{p}_c)$. In order to overcome the difficulty, we add Algorithm 12 behind Do ... While cycle body in Second Remedial Algorithm.

Algorithm 12 includes three components: (1) The Newton's steepest gradient descent method in the first step; (2) The Newton's iterative method closely associated with the test point \mathbf{p} in the second step; (3) Correcting method in the third step. Algorithm 12 has four advantages and important roles: (1) Algorithm 12 plays a role for accelerating the whole algorithm (Second Remedial Algorithm); (2) The first step plays a role for making the iteration point fall on the planar algebraic curve as far as possible; (3) The second step plays a role for accelerating orthogonalization between the vector $\overrightarrow{\mathbf{p}\mathbf{p}_c}$ and the vector $\nabla f(\mathbf{p}_c)$; (4) The third step plays a dual role for the accelerating orthogonalization and the promoting the iterative point to fall on the planar algebraic curve. The numerical tests of Second Remedial Algorithm achieve our expected results. No matter how far the test point \mathbf{p} is from the planar algebraic curve $f(\mathbf{x})$, Second Remedial Algorithm can converge very robustly and efficiently. Second Remedial Algorithm is the best one in our paper. Of course, the robustness and the efficiency of Second Remedial Algorithm are better than that of the existing algorithms. We are very happy about that.

Remark 4. In order to further improve the efficiency of the test point \mathbf{p} orthogonal projecting onto plane algebraic curve $f(\mathbf{x})$, we construct a Comprehensive Integrated Algorithm C which includes two parts: the hybrid second order algorithm in [28] and Second Remedial Algorithm. Firstly run the hybrid second order algorithm in [28]. If the hybrid second order algorithm converges, then it means that Comprehensive Integrated Algorithm C is finished. Otherwise, Second Remedial Algorithm runs until it converges successfully. That is the end of Comprehensive Integrated Algorithm C. The specific description of Comprehensive Integrated Algorithm C is similar to that of Comprehensive Integrated Algorithm B. Here, we are not giving a detailed description of Comprehensive Integrated Algorithm C. When the distance between the test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$ is not far, the hybrid second order algorithm in [28] has very high robustness and efficiency. When the distance between the test point \mathbf{p} and the planar algebraic curve $f(\mathbf{x})$ is particularly far, the hybrid second order algorithm does not converge and fails, while Second Remedial Algorithm converges particularly robustly and successfully. To sum up, Comprehensive Integrated Algorithm C absorbs the advantages of two sub-algorithms and overcomes their respective shortcomings such that the robustness and the efficiency of Comprehensive Integrated Algorithm C are maximized.

3. Convergence Analysis

This section proves that several Comprehensive Algorithms do not depend on the initial iteration points.

Theorem 1. Comprehensive Algorithm A is independent of the initial iterative point.

Proof. Firstly, we state the whole operation process of Comprehensive Algorithm A. Comprehensive Algorithm A contains three sub-algorithms (Algorithms 1–3). The role of Algorithm 1 is to repeatedly iterate the iterative formula (3) through Newton's steepest gradient descent method such that the final iteration point \mathbf{x}_{n+1} could fall on the planar algebraic curve where the final iteration point \mathbf{x}_{n+1} is denoted as \mathbf{p}_c . The function of Algorithm 2 is to seek the footpoint \mathbf{q} . The curvature circle \bigcirc determined by the point \mathbf{p}_c is obtained from the iterative point \mathbf{p}_c on the planar algebraic curve $f(\mathbf{x})$ of Algorithm 1, where the curvature k , the radius R and the center \mathbf{m} are determined by formulas (11)–(13), respectively. The intersection of the line segment $\overline{\mathbf{m}\mathbf{p}}$ connecting the center \mathbf{m} and the test point \mathbf{p} and the curvature circle \bigcirc is foot point \mathbf{q} . The footpoint \mathbf{q} could be orthogonally projected onto the planar algebraic curve $f(\mathbf{x})$ by repeated iteration of Algorithm 3 where at this moment the test point is not the original test point \mathbf{p} but the footpoint point \mathbf{q} solved by Algorithm 2. Repeatedly run Algorithm 2 and Algorithm 3 bound together until the distance between the current footpoint \mathbf{q} and the previous footpoint \mathbf{q} is near zero.

Secondly, the Comprehensive Algorithm A is independent of the initial iterative point. No matter how far the original test point \mathbf{p} is from the planar algebraic curve $f(\mathbf{x})$, no matter where the initial iterative point \mathbf{x}_0 is located, Algorithms 1 can ensure that the final iterative point \mathbf{x}_{n+1} or \mathbf{p}_c of the initial iterative point can fall on the planar algebraic curve $f(\mathbf{x})$. It is obvious that the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$

between the iteration point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ is much smaller than the distance $\|\mathbf{p} - \mathbf{p}_\Gamma\|$ between the orthogonal projection point \mathbf{p}_Γ and the original test point \mathbf{p} . From the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$, we can calculate the corresponding curvature k and its center \mathbf{m} and radius R . The intersection point \mathbf{q} of the curvature circle \bigcirc and the line segment $\overline{\mathbf{m}\mathbf{p}}$ connecting the original test point \mathbf{p} and the center \mathbf{m} of the curvature circle \bigcirc is just the footpoint \mathbf{q} . That is to say, the footpoint \mathbf{q} is directly generated by the curvature circle \bigcirc and the line segment $\overline{\mathbf{m}\mathbf{p}}$, and the curvature circle \bigcirc is controlled by the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$. So the footpoint \mathbf{q} is directly controlled by the original test point \mathbf{p} and the iterative point \mathbf{p}_c , while the current footpoint \mathbf{q} is between the orthogonal projection point \mathbf{p}_Γ and the current iterative point \mathbf{p}_c . It also shows that Algorithm 2 plays a decisive role in the convergence robustness of Comprehensive Algorithm. In addition, we can also know that the distance between the footpoint point \mathbf{q} and the planar algebraic curve $f(\mathbf{x})$ is much smaller than the distance between the original test point \mathbf{p} and the planar algebra curve $f(\mathbf{x})$. At this point, we keep running Algorithm 3 with the footpoint point \mathbf{q} as the current test point until the current test point can be orthogonally projected onto the plane algebraic curve $f(\mathbf{x})$ with guaranteed convergence of Algorithm 3. And now we can call the orthogonal projection point of the footpoint point \mathbf{q} as also the iterative point \mathbf{p}_c fallen on the planar algebraic curve $f(\mathbf{x})$. The first reason is the distance between the current iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} is smaller than the one between the previous iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} . The second reason is that it establishes a solid foundation for the convergence robustness of the subsequent sub-algorithms implementation. Then according to the requirements of Comprehensive Algorithm A, the second step and third step of Comprehensive Algorithm A are executed once per cycle, the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the current iterative point \mathbf{p}_c on the planar algebraic curve and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} of the execution result is smaller than that between the previous iterative point \mathbf{p}_c on the planar algebraic curve $f(\mathbf{x})$ and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} . The distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the current iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} is becoming smaller. So repeatedly iterate the second step and the third step of Comprehensive Algorithm A until the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the current iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} is becoming smaller and smaller. Ultimately, the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the current iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} is becoming zero. It also demonstrates that Comprehensive Algorithm A is completely convergent. This further proves that Comprehensive Algorithm A can completely converge no matter how far away the original test point \mathbf{p} is from the planar algebraic curve and no matter where the initial iterative point \mathbf{x}_0 of Comprehensive Algorithm A is on the plane. This means Comprehensive Algorithm A is independent of the initial iterative point. \square

Theorem 2. *Comprehensive Algorithm B is independent of the initial iterative point.*

Proof. In the last step of the iterative formula (18) in Algorithm 5, Newton's iteration method, which is closely related to the original test point \mathbf{p} , is added. In this way, the iterative formula (17) is transformed into the iterative formula (18) in Algorithm 5. Algorithm 5 has several advantages over Algorithm 3. It can speed up the iteration, improve its accuracy and promote the orthogonalization of the tangent vector derived from the iteration point on the planar algebraic curve and the tangent vector connecting the test point and the iterative point. Replace Algorithm 3 of Comprehensive Algorithm A with Algorithm 5 to get Comprehensive Algorithm B. Since Comprehensive Algorithm A is independent of the initial iterative point, so Comprehensive Algorithm B is naturally independent of the initial iterative point. \square

Theorem 3. *Comprehensive Integrated Algorithm A is independent of the initial iterative point.*

Proof. Comprehensive Integrated Algorithm A consists of two parts: the hybrid second order algorithm and Comprehensive Algorithm A. Whether the test point is very far or very close to the planar algebraic curve, the hybrid second order algorithm is executed several times. If this algorithm converges, then it represents that the execution of Comprehensive integrated Algorithm A is over. So Comprehensive Integrated Algorithm A is independent of the initial iterative point. If the hybrid second order algorithm does not converge, then run Comprehensive Algorithm A of the second step of Comprehensive Integrated Algorithm A. Because whether the test point is very far from or very close to the planar algebraic curve, we know from Theorem 1 that Comprehensive Algorithm A is independent of the initial iterative point. To sum up, Comprehensive Integrated Algorithm A is independent of the initial iterative point. \square

In a similar way to the proof of Theorem 3, we can state the following result.

Theorem 4. *Comprehensive Integrated Algorithm B is independent of the initial iterative point.*

Theorem 5. *The first remedial algorithm of Comprehensive Algorithm B is independent of the initial iterative point.*

Proof. From the Figure 5, for any initial iterative point, the final iterative point \mathbf{p}_c of Algorithm 1 in the first step of the first remedial algorithm of Comprehensive Algorithm B can ensure that it falls on the planar algebraic curve $f(\mathbf{x})$. The left endpoint \mathbf{L}_0 is the only one that can be determined through third step of the first remedial algorithm of Comprehensive Algorithm B. Graphic display shows that the left endpoint \mathbf{L}_0 and the original test point \mathbf{p} are on both sides of the planar algebraic curve. Namely, there is only one intersection point (also written as \mathbf{p}_c) between the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ and the planar algebraic curve $f(\mathbf{x})$. Because the hybrid method of combining Newton's iterative method and binary search method is global convergence method, the intersection \mathbf{p}_c of the line segment $\overline{\mathbf{L}_0\mathbf{p}}$ and the planar algebraic curve $f(\mathbf{x})$ can be accurately and uniquely solved by this method. Then repeatedly iterate and run Step 2, Step 3 and Step 4, the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the current intersection point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ of the original test point \mathbf{p} continues to shrink to zero. So we have this conclusion that the first remedial algorithm of Comprehensive Algorithm B is independent of the initial iterative point. \square

Theorem 6. *Second Remedial Algorithm is independent of the initial iterative point.*

Proof. In Remark 3, we give a detailed description of the geometric interpretation of Second Remedial Algorithm. In this proof, we only elaborate on the most important geometric significance of Second Remedial Algorithm. The first step of Second Remedial Algorithm is to let the initial iteration point fall on the planar algebraic curve as much as possible through Newton's steepest gradient descent method of Algorithm 1. Moreover, there is few restriction on the selection of the initial iterative point. The purpose of **Do ... While** cycle body in Second Remedial Algorithm is to continuously and gradually move the iterative point \mathbf{p}_c to fall on the planar algebraic curve to the orthogonal projection point \mathbf{p}_Γ . The second step in **Do ... While** cycle body in Second Remedial Algorithm has two characteristics. Since the footpoint \mathbf{q} is the unique intersection point of the curvature circle and the straight line segment $\overline{\mathbf{m}\mathbf{p}}$ connecting the centre \mathbf{m} of the curvature circle and the test point \mathbf{p} , the footpoint \mathbf{q} is always closely related to the iterative point \mathbf{p}_c fallen on the planar algebraic curve and the test point \mathbf{p} . The first characteristic is that the footpoint \mathbf{q} can guarantee the global convergence of the whole algorithm (Second Remedial Algorithm). The second characteristic is that the distance between the footpoint \mathbf{q} and the planar algebraic curve is much smaller than the distance between the test point \mathbf{p} and the planar algebraic curve. So the third step with Algorithm 1 in **Do ... While** cycle body can very robustly iterate the footpoint \mathbf{q} onto the planar algebraic curve. The core thought of **Do ... While** cycle body in Second Remedial Algorithm is to keep the iterative point \mathbf{p}_c fallen on the planar

algebraic curve moving towards the orthogonal projection point \mathbf{p}_Γ such that the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ between the iterative point \mathbf{p}_c and the orthogonal projection point \mathbf{p}_Γ becomes smaller and smaller. If the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ gets smaller and smaller, we have found that the decreasing speed of the distance $\|\mathbf{p}_c - \mathbf{p}_\Gamma\|$ is getting slower and slower. Especially the second formula of the formula (8) is very difficult to be satisfied. Algorithm 12 behind the loop body has four advantages and important roles: (1) Algorithm 12 plays a role for accelerating the whole algorithm (Second Remedial Algorithm); (2) The first step plays a role for making the iteration point fall on the planar algebraic curve as far as possible; (3) The second step plays a role for accelerating orthogonalization between the vector $\overrightarrow{\mathbf{p}\mathbf{p}_c}$ and the vector $\nabla f(\mathbf{p}_c)$; (4) The third step plays a dual role for the accelerating orthogonalization and the promoting the iterative point to fall on the planar algebraic curve. No matter how far the test point is from the planar algebraic curve, Second Remedial Algorithm converges very robustly and efficiently. By adding this step, the efficiency and the robustness for Algorithm 12 of Second Remedial Algorithm is further improved. Then the robustness and the efficiency of Second Remedial Algorithm is also further improved. So Second Remedial Algorithm is independent of the initial iterative point. In addition, in a similar way to the proof of Theorem 3, it is not difficult to know that Comprehensive Integrated Algorithm C is also independent of the initial iterative point. \square

4. Numerical Comparison Results

We now present some examples to illustrate the efficiency and the comparison of the newly developed method of Comprehensive Algorithm B and Second Remedial Algorithm to show the two algorithms' high robustness and efficiency for very remote test points. We have three examples to represent closed planar algebraic curve, two sub-closed planar algebraic curves, two branches but not closed planar algebra curves and a single branch not closed the planar algebra curve, respectively. All computations were done using VC++6.0. We used $\epsilon = 10^{-16}$. The following stopping criteria is used for Comprehensive Algorithm B and Second Remedial Algorithm. In Tables 1–3, the four symbols \mathbf{p} , \mathbf{p}_Γ , $|f(\mathbf{p}_\Gamma)|$ and $|\langle \mathbf{V}_1, \mathbf{V}_2 \rangle|$ are the original test point, the orthogonal projection point of the original test point, the deviation degree of the orthogonal projection point on the planar algebraic curve and the absolute value of the inner product of two vectors \mathbf{V}_1 and \mathbf{V}_2 , respectively, where \mathbf{V}_1 is $\overrightarrow{\mathbf{p}\mathbf{p}_\Gamma}$ and \mathbf{V}_2 is the tangent vector $\left[-\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}\right]$ of the orthogonal projection point \mathbf{p}_Γ on the planar algebraic curve $f(\mathbf{x})$. Thanks to the suggestions by the reviewers, the fourth quadrant result values of the three tables are implemented by Second Remedial Algorithm in Maple 18 environment.

Example 1 (Reference to [28]). Suppose a planar algebraic curve $f(x, y) = x^6 + 2x^5y - 2x^3y^2 + x^4 - y^3 + 2y^8 - 4 = 0$ (See Figure 7). In each of the four quadrants, randomly select four distant test points. We calculate the corresponding orthogonal projection point for each test point via computation by Comprehensive Algorithm B and Second Remedial Algorithm. The specific results are shown in Table 1).

Example 2. Suppose a planar algebraic curve $f(x, y) = x^{10} + 6xy + 2y^{18} - 2 = 0$ (See Figure 8). In each of the four quadrants, randomly select four distant test points. We calculate the corresponding orthogonal projection point for each test point via computation by Comprehensive Algorithm B and Second Remedial Algorithm. The specific results are shown in Table 2.

Example 3. Suppose a planar algebraic curve $f(x, y) = x^{10} + 6xy + 2y^{16} + 2 = 0$ (See Figure 9). In each of the four quadrants, randomly select four distant test points. We calculate the corresponding orthogonal projection point for each test point via computation by Comprehensive Algorithm B and Second Remedial Algorithm. The specific results are shown in Table 3.

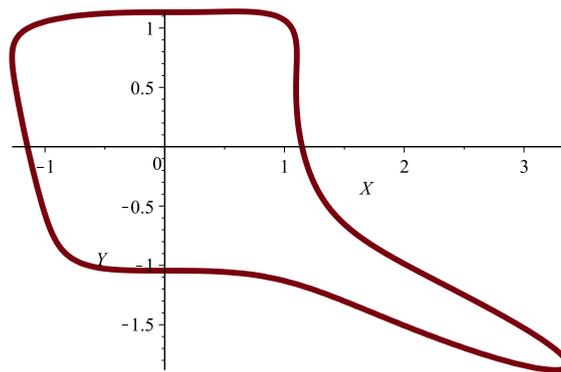


Figure 7. Graphical representation of the planar algebraic curve for Example 1.

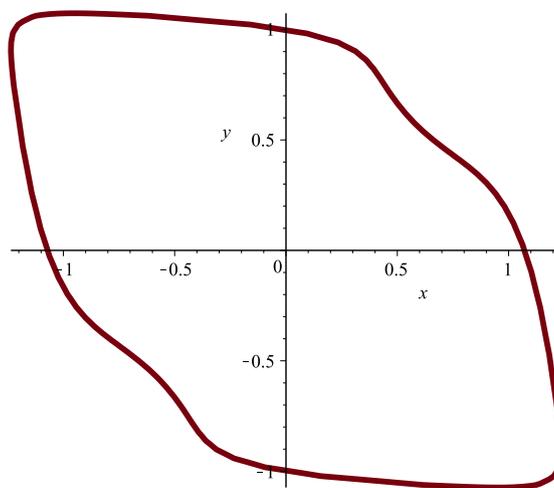


Figure 8. Graphical representation of the planar algebraic curve for Example 2.

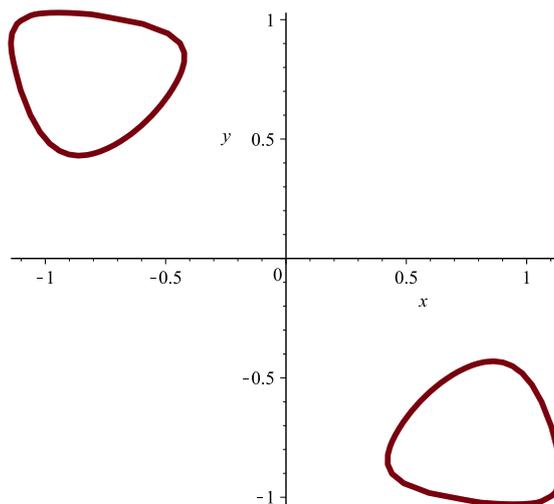


Figure 9. Graphical representation of the planar algebraic curve for Example 3.

Table 1. Test results of Comprehensive Algorithm B for Example 1.

p	(1325, 7447)	(779, 325)	(990, 1375)	(0.59, 1377)	(−0.5, 8623)	(−16, 598)	(−3, 231)	(−21, 247)
p_Γ	(3.1087, −1.8666)	(3.3647, −1.8184)	(3.3695, −1.8380)	(0.60, 1.140)	(−0.01582, 1.13368)	(−0.3353, 1.13074)	(−0.2268, 1.1328)	(−0.6071, 1.1166)
$ f(\mathbf{p}_\Gamma) $	$3.9 \cdot 10^{-10}$	$7.9 \cdot 10^{-11}$	$1.4 \cdot 10^{-11}$	$2.7 \cdot 10^{-12}$	$2.3 \cdot 10^{-12}$	$4.0 \cdot 10^{-12}$	$1.9 \cdot 10^{-12}$	$2.3 \cdot 10^{-12}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	$2.3 \cdot 10^{-10}$	$4.4 \cdot 10^{-11}$	$2.9 \cdot 10^{-09}$	$4.8 \cdot 10^{-12}$	$2.1 \cdot 10^{-13}$	$1.5 \cdot 10^{-12}$	$1.4 \cdot 10^{-11}$	$9.6 \cdot 10^{-12}$
p	(−42, −127)	(−5, −38)	(−9, −579)	(−537, −11)	(78, −123)	(168, −12)	(537, −31)	(91, −221)
p_Γ	(0.6633, −0.9941)	(0.4951, −1.0292)	(−0.2183, −1.0445)	(−1.2752, 0.7396)	(3.3519, −1.8685)	(3.3694, −1.8417)	(3.3694, −1.8414)	(3.3415, −1.8736)
$ f(\mathbf{p}_\Gamma) $	$7.4 \cdot 10^{-12}$	$1.4 \cdot 10^{-12}$	$1.7 \cdot 10^{-12}$	$6.1 \cdot 10^{-12}$	$4.9 \cdot 10^{-12}$	$1.8 \cdot 10^{-13}$	$5.4 \cdot 10^{-13}$	$8.6 \cdot 10^{-12}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	$1.7 \cdot 10^{-13}$	$1.2 \cdot 10^{-12}$	$1.1 \cdot 10^{-12}$	$1.7 \cdot 10^{-12}$	$2.3 \cdot 10^{-13}$	$5.7 \cdot 10^{-14}$	$2.8 \cdot 10^{-13}$	$4.5 \cdot 10^{-12}$

Table 2. Test results of Comprehensive Algorithm B for Example 2.

p	(565, 945)	(979, 325)	(375, 405)	(1959, 1377)	(−9356, 8623)	(−816, 798)	(−3987, 1231)	(−4821, 647)
p_Γ	(0.2978, −0.9108)	(1.0773, −0.0164)	(0.6094, 0.5450)	(0.4780, 0.6961)	(−1.2055, 1.0209)	(−1.2035, 1.0230)	(−1.2288, 0.9794)	(−1.2341, 0.9554)
$ f(\mathbf{p}_\Gamma) $	$1.1 \cdot 10^{-14}$	$6.0 \cdot 10^{-13}$	$8.0 \cdot 10^{-18}$	$9.2 \cdot 10^{-15}$	$5.8 \cdot 10^{-15}$	$1.1 \cdot 10^{-14}$	$2.5 \cdot 10^{-15}$	$1.1 \cdot 10^{-14}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	0	$1.8 \cdot 10^{-12}$	$2.3 \cdot 10^{-13}$	$4.5 \cdot 10^{-12}$	$6.9 \cdot 10^{-10}$	$2.1 \cdot 10^{-11}$	$1.7 \cdot 10^{-10}$	$1.1 \cdot 10^{-10}$
p	(−7942, −275)	(−598, −98)	(−3709, −1979)	(−2937, −1391)	(9708, −323)	(2608, −1912)	(7347, −931)	(5091, −1921)
p_Γ	(−1.2367, 0.8930)	(−1.1847, 0.4851)	(−1.0035, −0.1601)	(−1.0225, 0.1224)	(1.1427, −0.9100)	(1.1224, −0.9787)	(1.1414, −0.9273)	(1.1344, −0.9563)
$ f(\mathbf{p}_\Gamma) $	$2.3 \cdot 10^{-15}$	$1.3 \cdot 10^{-11}$	$9.5 \cdot 10^{-15}$	$1.4 \cdot 10^{-13}$	$4.7 \cdot 10^{-15}$	$3.2 \cdot 10^{-15}$	$4.4 \cdot 10^{-15}$	$1.7 \cdot 10^{-17}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	$4.0 \cdot 10^{-11}$	$3.6 \cdot 10^{-12}$	$8.7 \cdot 10^{-11}$	$7.2 \cdot 10^{-12}$	$5.8 \cdot 10^{-11}$	$8.7 \cdot 10^{-11}$	$7.2 \cdot 10^{-12}$	$2.1 \cdot 10^{-10}$

Table 3. Test results of Comprehensive Algorithm B for Example 3.

p	(1387, 645)	(1879, 395)	(3075, 205)	(1956, 777)	(−9256, 4603)	(−836, 1798)	(−5987, 1031)	(−4181, 1247)
p_Γ	(−1.0629, 0.6023)	(0.4232, −0.8650)	(0.4214, −0.8515)	(0.4276, −0.8795)	(−1.1305, 0.9655)	(−1.0823, 1.0103)	(−0.4232, 0.8220)	(−1.1369, 0.9490)
$ f(\mathbf{p}_\Gamma) $	$1.7 \cdot 10^{-13}$	$1.3 \cdot 10^{-16}$	$8.1 \cdot 10^{-17}$	$1.0 \cdot 10^{-16}$	$5.3 \cdot 10^{-17}$	$1.8 \cdot 10^{-15}$	$6.1 \cdot 10^{-17}$	$1.0 \cdot 10^{-17}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	$3.6 \cdot 10^{-12}$	$3.2 \cdot 10^{-12}$	$6.3 \cdot 10^{-13}$	$6.3 \cdot 10^{-12}$	$1.7 \cdot 10^{-10}$	$4.3 \cdot 10^{-11}$	$1.8 \cdot 10^{-12}$	$6.5 \cdot 10^{-11}$
p	(−7342, −1275)	(−5098, −918)	(−3217, −2079)	(−2337, −1251)	(9508, −375)	(6608, −712)	(2347, −931)	(1491, −1321)
p_Γ	(−0.4226, 0.8617)	(−0.4227, 0.8623)	(−1.0274, 0.5370)	(−1.0471, 0.5706)	(1.2367, −0.9267)	(1.2353, −0.9460)	(1.2255, −0.9888)	(1.2068, −1.0194)
$ f(\mathbf{p}_\Gamma) $	$5.7 \cdot 10^{-17}$	$1.3 \cdot 10^{-16}$	$4.4 \cdot 10^{-14}$	$1.1 \cdot 10^{-13}$	$2.2 \cdot 10^{-15}$	$2.8 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$1.1 \cdot 10^{-15}$
$ \langle \mathbf{V}_1, \mathbf{V}_2 \rangle $	$1.0 \cdot 10^{-11}$	$5.5 \cdot 10^{-12}$	$7.3 \cdot 10^{-12}$	$9.1 \cdot 10^{-12}$	$2.2 \cdot 10^{-11}$	$3.6 \cdot 10^{-12}$	$5.4 \cdot 10^{-11}$	$2.2 \cdot 10^{-11}$

Remark 5. Besides all test points of the three examples mentioned above are tested by Comprehensive Algorithm B, we have tested them again with the Second Remedial Algorithm. All the test results are consistent with those of Comprehensive Algorithm B and convergent. In addition, in the region $[-3000, 3000] \times [-3000, 3000]$ of each example, we randomly select a large number of test points, the probability of non-convergence is particularly low by Second Remedial Algorithm. Further, we use Second Remedial Algorithm other examples with test points in a very large area, and the probability of non-convergence is also very low. Second Remedial Algorithm is verified to be the best one again in our paper. Of course, the robustness and the efficiency of Second Remedial Algorithm is better than that of the existing algorithms.

5. Conclusions and Future Work

In this paper, we have constructed a Comprehensive Algorithm which is an improved curvature circle algorithm for orthogonal projecting onto planar algebraic curve. Based on an integrated hybrid second-order algorithm [28], the Comprehensive Algorithm (the improved curvature circle algorithm) has also incorporated the curvature circle technique and Newton's gradient steepest descent method such that it can converge robustly and efficiently no matter how far the test point is from the planar algebraic curve and no matter where the initial iterative point is located. Furthermore, we propose Second Remedial Algorithm based on Comprehensive Algorithm B. In particular, its robustness and efficiency is greatly improved than that of Comprehensive Algorithm B and it achieves our expected result. The numerical examples show that the improved curvature circle algorithm is superior to the existing ones. In future work, we try to refine the idea of Comprehensive Algorithm and Second Remedial Algorithm. And the idea is applied to point orthogonal projecting onto spatial algebraic curve and algebraic surface.

Author Contributions: The contribution of all the authors is the same. All of the authors team up to develop the current draft. Z.W. is responsible for investigating, providing resources and methodology, the original draft, writing, reviewing, validation, editing and supervision of this work. X.L. is responsible for software, algorithm, program implementation, formal analysis, visualization, writing, reviewing, editing and supervision of this work.

Funding: This research was funded by the National Natural Science Foundation of China Grant No. 61263034, the Feature Key Laboratory for Regular Institutions of Higher Education of Guizhou Province Grant No. 2016003, the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University Grant No. 2018479, the National Bureau of Statistics Foundation Grant No. 2014LY011, the Shandong Provincial Natural Science Foundation of China Grant No.ZR2016GM24.

Acknowledgments: We take the opportunity to thank the anonymous reviewers for their thoughtful and meaningful comments.

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

References

1. Bajaj, C.L.; Bernardini, F.; Xu, G. Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data. *Algorithmica* **1997**, *19*, 243–261. [[CrossRef](#)]
2. Jüttler, B.; Felis, A. Least-squares fitting of algebraic spline surfaces. *Adv. Comput. Math.* **2002**, *17*, 135–152.
3. Song, X.; Jüttler, B. Modeling and 3D object reconstruction by implicitly defined surfaces with sharp features. *Comput. Graph.* **2009**, *33*, 321–330. [[CrossRef](#)]
4. Schulz, T.; Jüttler, B. Envelope computation in the plane by approximate implicitization. *Appl. Algebra Eng. Commun. Comput.* **2011**, *22*, 265–288. [[CrossRef](#)]
5. Kanatani, K.; Sugaya, Y. Unified computation of strict maximum likelihood for geometric fitting. *J. Math. Imaging Vis.* **2010**, *38*, 1–13. [[CrossRef](#)]
6. Mullen, P.; De Goes, F.; Desbrun, M.; Cohen-Steiner, D.; Alliez, P. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Comput. Graph. Forum* **2010**, *29*, 1733–1741. [[CrossRef](#)]
7. Upreti, K.; Subbarayan, G. Signed algebraic level sets on NURBS surfaces and implicit Boolean signed algebraic level sets on NURBS surfaces and implicit Boolean. *Comput. Aided Des.* **2017**, *82*, 112–126. [[CrossRef](#)]
8. Rouhani, M.; Sappa, A.D. Implicit polynomial representation through a fast fitting error estimation. *IEEE Trans. Image Process.* **2012**, *21*, 2089–2098. [[CrossRef](#)]

9. Zagorchev, L.G.; Goshtasby, A.A. A curvature-adaptive implicit surface reconstruction for irregularly spaced points. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1460–1473. [[CrossRef](#)]
10. William, H.P.; Brian, P.F.; Teukolsky, S.A.; William, T.V. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1992.
11. Steve, S.; Sandford, L.; Ponce, J. Using geometric distance fits for 3-D object modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 1183–1196.
12. Morgan, A.P. Polynomial continuation and its relationship to the symbolic reduction of polynomial systems. In *Symbolic and Numerical Computation for Artificial Intelligence*; Academic Press: Cambridge, MA, USA, 1992; pp. 23–45.
13. Layne, T.W.; Billups, S.C.; Morgan, A.P. Algorithm 652: HOMPACT: A suite of codes for globally convergent homotopy algorithms. *ACM Trans. Math. Softw.* **1987**, *13*, 281–310.
14. Berthold, K.P.H. Relative orientation revisited. *J. Opt. Soc. Am. A* **1991**, *8*, 1630–1638.
15. Dinesh, M.; Krishnan, S. Solving algebraic systems using matrix computations. *ACM Sigsam Bull.* **1996**, *30*, 4–21.
16. Chionh, E.-W. Base Points, Resultants, and the Implicit Representation of Rational Surfaces. Ph.D. Thesis, University of Waterloo, Waterloo, ON, Canada, 1990.
17. De Montaudouin, Y.; Tiller, W. The Cayley method in computer aided geometric design. *Comput. Aided Geom. Des.* **1984**, *1*, 309–326. [[CrossRef](#)]
18. Albert, A.A. *Modern Higher Algebra*; D.C. Heath and Company: New York, NY, USA, 1933.
19. Thomas, W.; David, S.; Anderson, C.; Goldman, R.N. Implicit representation of parametric curves and surfaces. *Comput. Vis. Graph. Image Proc.* **1984**, *28*, 72–84.
20. Nishita, T.; Sederberg, T.W.; Kakimoto, M. Ray tracing trimmed rational surface patches. *ACM SIGGRAPH Comput. Graph.* **1990**, *24*, 337–345. [[CrossRef](#)]
21. Elber, G.; Kim, M.-S. Geometric Constraint Solver Using Multivariate Rational Spline Functions. In Proceedings of the 6th ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, USA, 4–8 June 2001; pp. 1–10.
22. Sherbrooke, E.C.; Patrikalakis, N.M. Computation of the solutions of nonlinear polynomial systems. *Comput. Aided Geom. Des.* **1993**, *10*, 379–405. [[CrossRef](#)]
23. Hartmann, E. The normal form of a planar curve and its application to curve design. In *Mathematical Methods for Curves and Surfaces II*; Vanderbilt University Press: Nashville, TN, USA, 1997; pp. 237–244.
24. Hartmann, E. On the curvature of curves and surfaces defined by normal forms. *Comput. Aided Geom. Des.* **1999**, *16*, 355–376. [[CrossRef](#)]
25. Nicholas, J.R. Implicit polynomials, orthogonal distance regression, and the closest point on a curve. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 191–199.
26. Martin, A.; Bert, J. Robust computation of foot points on implicitly defined curves. In *Mathematical Methods for Curves and Surfaces: Tromsø*; Nashboro Press: Brentwood, TN, USA, 2004; pp. 1–10.
27. Hu, M.; Zhou, Y.; Li, X. Robust and accurate computation of geometric distance for Lipschitz continuous implicit curves. *Vis. Comput.* **2017**, *33*, 937–947. [[CrossRef](#)]
28. Li, X.; Pan, F.; Cheng, T.; Wu, Z.; Liang, J.; Hou, L. Integrated hybrid second order algorithm for orthogonal projection onto a planar implicit curve. *Symmetry* **2018**, *10*, 164. [[CrossRef](#)]
29. Goldman R. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.* **2005**, *22*, 632–658. [[CrossRef](#)]

