

ORIGINAL ARTICLE

Open Access



Curve intersection based on cubic hybrid clipping

Yaqiong Wu and Xin Li*

Abstract

This study presents a novel approach to computing all intersections between two Bézier curves using cubic hybrid clipping. Each intersection is represented by two strip intervals that contain an intersection. In each step, one curve is bounded by two fat lines, and the other is bounded by two cubic Bézier curves, clipping away the domain that does not contain the intersections. By selecting the moving control points of the cubic hybrid curves, better cubic polynomial bounds are obtained to make the proposed method more efficient. It was proved that the two strip intervals have second- and fourth-order convergence rates for transversal intersections. Experimental results show that the new algorithm is the most efficient among all existing curve/curve intersection approaches.

Keywords: Bézier curve, Curve intersection, Hybrid clipping

Introduction

Given an interval $[\alpha, \beta] \subset \mathbb{R}$, a Bézier curve $\mathbf{P}(t)$, $t \in [\alpha, \beta]$ is defined as

$$\mathbf{P}(t) = \sum_{i=0}^n \mathbf{P}_i B_{i, [\alpha, \beta]}^n(t) \quad (1)$$

where $B_{i, [\alpha, \beta]}^n(t) = \binom{n}{i} \frac{(t-\alpha)^i (\beta-t)^{n-i}}{(\beta-\alpha)^n}$, $i = 0, 1, \dots, n$ are Bernstein polynomials in $[\alpha, \beta]$, and $\{\mathbf{P}_i\}_{i=0}^n$ are the control points [1]. Given two Bézier curves $\mathbf{P}(t)$, $t \in [\alpha, \beta]$ and $\mathbf{Q}(s)$, $s \in [\xi, \eta]$, the problem considered in the present study is how to compute all (t^*, s^*) , $t^* \in [\alpha, \beta]$, $s^* \in [\xi, \eta]$ such that $\mathbf{P}(t^*) = \mathbf{Q}(s^*)$. Various methods have been developed to solve this task, such as a subdivision-based approach [2], binary subdivision approach [3], implicitization [4] and Bézier clipping [5].

The most common approach consists of clipping away the regions of the curves that are guaranteed to not intersect. Each intersection parameter pair (s^*, t^*) is replaced with an interval that is iteratively computed. The

k -th iteration interval is denoted as $[\alpha_k, \beta_k] \times [\xi_k, \eta_k]$. Let $h_k = \beta_k - \alpha_k$ and $d_k = \eta_k - \xi_k$. If a constant γ_i exists such that

$$h_{k+1} \leq C_1 h_k^{\gamma_1} + C_2 d_k^{\gamma_2}, d_{k+1} \leq C_3 h_k^{\gamma_3} + C_4 d_k^{\gamma_4} \quad (2)$$

where C_i are constants independent of k and the curves, and thus γ_i is the convergence rate of the sequence intervals $\{[\alpha_k, \beta_k]\}_k$ and $\{[\xi_k, \eta_k]\}_k$. The key problem is to find an algorithm for which γ_i is as large as possible with as few computations as possible during each iteration.

The above problem plays an important role in many engineering fields, such as computer-aided design and manufacturing (CAD/CAM), collision detection, and geometric modeling [1], and is a basic operation in solid modeling. In geometric processing, the intersections and intersection curves in a solid model are extremely important for the visualization, analysis, and manufacturing of the model [6]. With the continuous development of computer-aided geometric design and CAD/CAM, as well as the continuous progress made in science and technology, the numbers of calculations and data to be processed for intersection problems are increasing. It is therefore important to develop efficient and stable methods for dealing with intersection problems.

To solve such problems, the Bézier clipping algorithm introduced in ref. [5] is a widely used, fast, and

*Correspondence: lixustc@ustc.edu.cn

School of Mathematical Science, University of Science and Technology of China, Hefei 200026, Anhui, China

robust method. The Bézier clipping algorithm was proven to have a second-order convergence rate [7]. Subsequently, several different approaches have been proposed to improve the Bézier clipping algorithm. Bartoň and Jüttler [8] and Liu et al. [9] developed quadratic and cubic clipping techniques based on a degree reduction to compute all roots of a univariate polynomial equation. Lou and Liu [10] extended the approach in ref. [9] to curve/curve intersection problems and proved that the algorithm achieves at least a second-order convergence rate. In addition, North [11] developed a geometry interval clipping algorithm based on quadratic hybrid curves [12] for use with curve/curve intersection problems, Liu and Li [13] proved that the algorithm achieves a quadratic convergence rate. Moreover, Yuan [14] recently developed a cubic hybrid clipping (HybClip) based on hybrid curves to compute all roots of a univariate polynomial equation with a numerically verified fourth-order convergence rate.

In this study, the approach in ref. [14] is extended to handle curve/curve intersection problems. Unlike the approach in ref. [14], a better bound is chosen for cubic HybClip, and thus the algorithm requires 8% less time than a method that directly uses the cubic hybrid curve [14]. In addition, it is proved that the two sequences in the new clipping algorithm have second- and fourth-order convergence rates. Subsequently, a complete comparison is provided with all existing curve/curve intersection algorithms based on subdivisions on a random 40,000 curve/curve intersection database. The new algorithm requires 30% less time than the geometry interval clipping algorithm [11] and 60% less time than the cubic clipping algorithm [10].

The remainder of this paper is organized as follows. In **Methods** section, the cubic hybrid curves are presented with two moving control points, and the details of the curve/curve intersection algorithms are described when applied in both 2D and 3D. In **Results** section, a proof of the convergence rate of the new intersection algorithm is provided, and the six techniques are compared from various perspectives. Finally, some concluding remarks are provided in **Conclusions** section and areas of future work are discussed in **Discussion** section.

Methods

Hybrid curve

A hybrid curve refers to a curve with at least one moving control point, which is itself a parametric curve and shares one parameter with the hybrid. Sederberg and Kakimoto [12] originated the idea of using hybrid polynomial Bézier curves to approximate rational Bézier curves. Later, North [11] transformed all polynomial Bézier curves of degree $d \geq 2$ into equivalent quadratic hybrid curves with a single moving control point and fixed endpoints. As an illustration, a simple quadratic hybrid curve was constructed with a single moving control point, equivalent to a cubic Bézier curve, as shown in Fig. 1.

$$\begin{aligned}
 \mathbf{P}(t) &= \mathbf{P}_0(1-t)^3 + 3\mathbf{P}_1(1-t)^2t + 3\mathbf{P}_2(1-t)t^2 + \mathbf{P}_3t^3 \\
 &= \mathbf{P}_0B_0^2(t) + \left(\frac{3\mathbf{P}_1-\mathbf{P}_0}{2}(1-t) + \frac{3\mathbf{P}_2-\mathbf{P}_3}{2}t\right)B_1^2(t) + \mathbf{P}_3B_2^2(t) \quad (3)
 \end{aligned}$$

To evaluate a point on a hybrid curve, the locations of all moving control points are first determined at the given parameter value t . Once the moving control points are determined, the hybrid curve can be

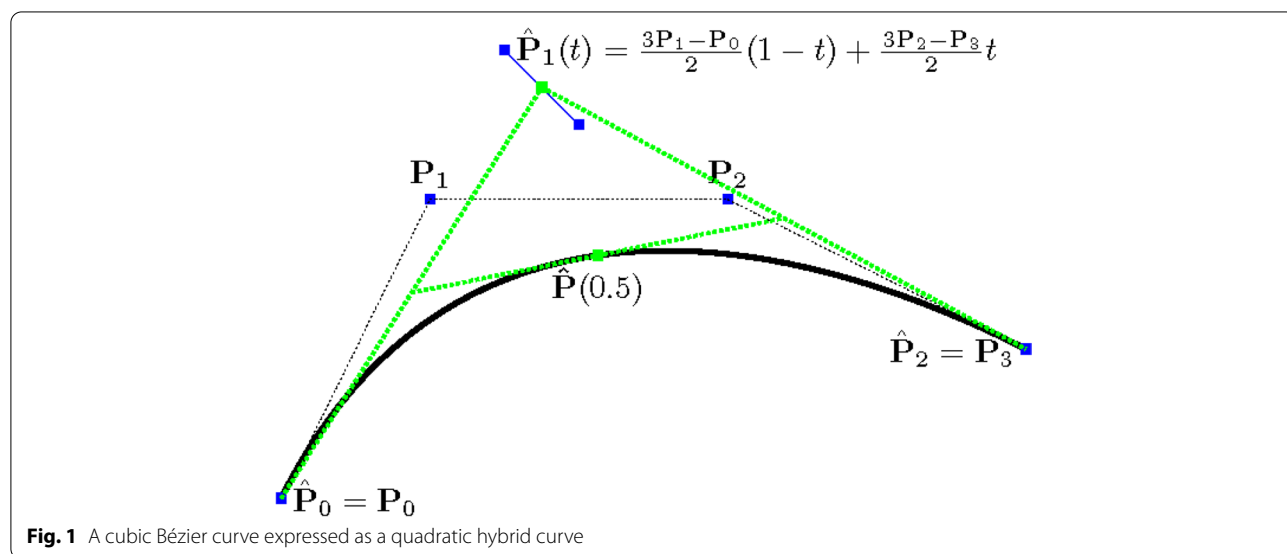


Fig. 1 A cubic Bézier curve expressed as a quadratic hybrid curve

evaluated as a common curve. For example, to evaluate $\hat{\mathbf{P}}(t)$ at $t=0.5$, $\hat{\mathbf{P}}_1(0.5)$ is first evaluated and the resulting point is then used to evaluate $\hat{\mathbf{P}}(0.5)$, as shown in Fig. 1.

Using the same principles as in ref. [11], by properly selecting the moving control points, the hybrid curve can produce any traditional Bézier curve. In this study, a hybrid curve having the following form is focused on:

Theorem 1. *Given a degree $n \geq 3$ Bézier curve $\mathbf{P}(t)$ with control points $\{\mathbf{P}_i\}_{i=0}^n$ there exists an equivalent cubic hybrid curve $\hat{\mathbf{P}}(t)$ with two fixed control points $\hat{\mathbf{P}}_0 = \mathbf{P}_0, \hat{\mathbf{P}}_3 = \mathbf{P}_n$ and two moving control points $\hat{\mathbf{P}}_1(t), \hat{\mathbf{P}}_2(t)$. The two moving control points $\hat{\mathbf{P}}_1(t), \hat{\mathbf{P}}_2(t)$ are Bézier curves of degree $n-3$ with control points $\{\hat{\mathbf{P}}_{1,i-1}\}_{i=1}^{n-2}$ and $\{\hat{\mathbf{P}}_{2,i-2}\}_{i=2}^{n-1}$, respectively, where*

$$\frac{n-i-1}{n-2} \hat{\mathbf{P}}_{1,i-1} + \frac{i-1}{n-2} \hat{\mathbf{P}}_{2,i-2} = \frac{a_i \mathbf{P}_0 + b_i \mathbf{P}_i + c_i \mathbf{P}_n}{a_i + b_i + c_i} \tag{4}$$

$a_i = -(n-i)(n-i-1)(n-i-2)$, $b_i = n(n-1)(n-2)$, $c_i = -i(i-1)(i-2)$, and $i \in \{1, \dots, n-1\}$.

Proof. The degree $n \times m$ tensor product Bézier surface patch [1] is defined as

$$\mathbf{Q}(s, t) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{Q}_{i,j} B_i^n(s) B_j^m(t) \tag{5}$$

where $B_i^n(s) B_j^m(t), 0 \leq s, t \leq 1$ is the product of the two Bernstein bases in $[0, 1]$, and $\mathbf{Q}_{i,j}, i=0, \dots, n; j=0, \dots, m$ are the control points of $\mathbf{Q}(s, t)$.

From ref. [15], a degree $m+d$ Bézier curve $\mathbf{P}(t)$ with control points \mathbf{P}_i can be described as the diagonal curve $\mathbf{P}(t) = \mathbf{Q}(t, t)$ of a degree $m \times d$ Bézier surface $\mathbf{Q}(s, t)$, i.e.,

$$\mathbf{P}_i = \frac{1}{\binom{m+d}{i}} \sum_{j+k=i} \binom{m}{j} \binom{d}{k} \mathbf{Q}_{j,k} \tag{6}$$

If $m=3$, then $j \in \{0, 1, 2, 3\}$ and $(j, k) \in \{(0, i), (1, i-1), (2, i-2), (3, i-3)\}$. Expanding the summation and rearranging the terms, the following is obtained:

$$\binom{d+3}{i} \mathbf{P}_i = \binom{3}{0} \binom{d}{i} \mathbf{Q}_{0,i} + \binom{3}{1} \binom{d}{i-1} \mathbf{Q}_{1,i-1} + \binom{3}{2} \binom{d}{i-2} \mathbf{Q}_{2,i-2} + \binom{3}{3} \binom{d}{i-3} \mathbf{Q}_{3,i-3} \tag{7}$$

If the control points \mathbf{P}_i of degree $n=d+3$ diagonal curve $\mathbf{P}(t)$ are known, $\mathbf{Q}_{0,i} = \mathbf{P}_0$ and $\mathbf{Q}_{3,i-3} = \mathbf{P}_n$ can be set. Thus,

$$\binom{n}{i} \mathbf{P}_i = \binom{n-3}{i} \mathbf{P}_0 + 3 \binom{n-3}{i-1} \mathbf{Q}_{1,i-1} + 3 \binom{n-3}{i-2} \mathbf{Q}_{2,i-2} + \binom{n-3}{i-3} \mathbf{P}_n \tag{8}$$

$$\binom{n-3}{i-1} \mathbf{Q}_{1,i-1} + \binom{n-3}{i-2} \mathbf{Q}_{2,i-2} = \frac{1}{3} \left[\binom{n}{i} \mathbf{P}_i - \binom{n-3}{i} \mathbf{P}_0 - \binom{n-3}{i-3} \mathbf{P}_n \right]$$

Simplifying the above formulas, the following is achieved:

$$\frac{n-i-1}{n-2} \mathbf{Q}_{1,i-1} + \frac{i-1}{n-2} \mathbf{Q}_{2,i-2} = \frac{(i-n)(n-i-1)(n-i-2)}{3i(n-i)(n-2)} \mathbf{P}_0 + \frac{n(n-1)(n-2)}{3i(n-i)(n-2)} \mathbf{P}_i + \frac{-i(i-1)(i-2)}{3i(n-i)(n-2)} \mathbf{P}_n \tag{9}$$

for $i \in \{1, \dots, n-1\}$. Setting $a_i = -(n-i)(n-i-1)(n-i-2)$, $b_i = n(n-1)(n-2)$, $c_i = -i(i-1)(i-2)$, and observing that $a_i + b_i + c_i = 3i(n-i)(n-2)$, the following occur:

$$\frac{n-i-1}{n-2} \mathbf{Q}_{1,i-1} + \frac{i-1}{n-2} \mathbf{Q}_{2,i-2} = \frac{a_i \mathbf{P}_0 + b_i \mathbf{P}_i + c_i \mathbf{P}_n}{a_i + b_i + c_i} \tag{10}$$

Because $\mathbf{Q}_{0,i} = \mathbf{P}_0$ and $\mathbf{Q}_{3,i-3} = \mathbf{P}_n$, the $s=t$ diagonal curve of $\mathbf{Q}(s, t)$ can be evaluated using the following formula:

$$\mathbf{Q}(t, t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \hat{\mathbf{P}}_1(t) + 3(1-t)t^2 \hat{\mathbf{P}}_2(t) + t^3 \mathbf{P}_n \tag{11}$$

where $\hat{\mathbf{P}}_1(t)$ and $\hat{\mathbf{P}}_2(t)$ are the degree $n-3$ Bézier curves comprising the control points $\hat{\mathbf{P}}_{1,i-1}$ and $\hat{\mathbf{P}}_{2,i-2}$, respectively, where

$$\hat{\mathbf{P}}_{1,i-1} = \mathbf{Q}_{1,i-1}, i=1, \dots, n-2 \quad \text{and} \quad \hat{\mathbf{P}}_{2,i-2} = \mathbf{Q}_{2,i-2}, i=2, \dots, n-1 \tag{12}$$

This is a cubic hybrid curve with two moving control points $\hat{\mathbf{P}}_1(t), \hat{\mathbf{P}}_2(t)$, and fixed control points $\mathbf{P}_0, \mathbf{P}_n$.

From Theorem 1, if $i=1$ or $i=n-1$, the first control point of $\hat{\mathbf{P}}_1(t)$ and the last control point of $\hat{\mathbf{P}}_2(t)$ are fixed as follows:

$$\hat{\mathbf{P}}_{1,0} = \frac{(3-n)\mathbf{P}_0 + n\mathbf{P}_1}{3} \quad \text{and} \quad \hat{\mathbf{P}}_{2,n-3} = \frac{(3-n)\mathbf{P}_n + n\mathbf{P}_{n-1}}{3} \tag{13}$$

Theorem 1 indicates that the two moving control points are relevant, for which three cases are discussed:

Case 1

If the first moving control point is a fixed point denoted by $\hat{\mathbf{Q}}_1$ and the second moving control point is denoted by $\hat{\mathbf{Q}}_2(t)$, then the control points $\{\hat{\mathbf{Q}}_{2,i}\}_{i=0}^{n-3}$ of $\hat{\mathbf{Q}}_2(t)$ can be calculated from Eq. (4), as indicated by Yuan [14]:

$$\hat{\mathbf{Q}}(t) = \mathbf{P}_0 B_0^3(t) + \hat{\mathbf{Q}}_1 B_1^3(t) + \hat{\mathbf{Q}}_2(t) B_2^3(t) + \mathbf{P}_n B_3^3(t) \tag{14}$$

Case 2

If the second moving control point is a fixed point denoted by $\hat{\mathbf{R}}_2$ and the first moving control point is

denoted by $\hat{\mathbf{R}}_1(t)$, the control points $\{\hat{\mathbf{R}}_{1,i}\}_{i=0}^{n-3}$ of $\hat{\mathbf{R}}_1(t)$ can be obtained from Eq. (4), i.e.,

$$\hat{\mathbf{R}}(t) = \mathbf{P}_0 B_0^3(t) + \hat{\mathbf{R}}_1(t) B_1^3(t) + \hat{\mathbf{R}}_2 B_2^3(t) + \mathbf{P}_n B_3^3(t) \tag{15}$$

Case 3

In general, $\hat{\mathbf{P}}_1(t)$ and $\hat{\mathbf{P}}_2(t)$ are moving to control points. Because they are equivalent to $\hat{\mathbf{P}}(t)$,

$$\begin{aligned} \mathbf{P}(t) = \hat{\mathbf{Q}}(t) &= \mathbf{P}_0 B_0^3(t) + \hat{\mathbf{Q}}_1 B_1^3(t) + \hat{\mathbf{Q}}_2 B_2^3(t) + \mathbf{P}_n B_3^3(t) \\ &= \hat{\mathbf{R}}(t) = \mathbf{P}_0 B_0^3(t) + \hat{\mathbf{R}}_1(t) B_1^3(t) + \hat{\mathbf{R}}_2 B_2^3(t) + \mathbf{P}_n B_3^3(t) \\ &= \hat{\mathbf{P}}(t) = \mathbf{P}_0 B_0^3(t) + \hat{\mathbf{P}}_1(t) B_1^3(t) + \hat{\mathbf{P}}_2(t) B_2^3(t) + \mathbf{P}_n B_3^3(t) \end{aligned} \tag{16}$$

Hence, $\lambda \in [0, 1]$ exists such that

$$(1 - \lambda)\hat{\mathbf{Q}}(t) + \lambda\hat{\mathbf{R}}(t) = \hat{\mathbf{P}}(t) \tag{17}$$

Through a simple approach,

$$(1 - \lambda)\hat{\mathbf{Q}}_1 + \lambda\hat{\mathbf{R}}_1(t) = \hat{\mathbf{P}}_1(t) \quad \text{and} \quad (1 - \lambda)\hat{\mathbf{Q}}_2 + \lambda\hat{\mathbf{R}}_2 = \hat{\mathbf{P}}_2(t) \tag{18}$$

and obtain

$$\hat{\mathbf{P}}_{1,i} = (1 - \lambda)\hat{\mathbf{Q}}_1 + \lambda\hat{\mathbf{R}}_{1,i} \quad \text{and} \quad \hat{\mathbf{P}}_{2,i} = (1 - \lambda)\hat{\mathbf{Q}}_{2,i} + \lambda\hat{\mathbf{R}}_2 \tag{19}$$

where $i \in \{0, \dots, n - 3\}$; in addition, $\hat{\mathbf{Q}}_1$, $\hat{\mathbf{Q}}_{2,i}$ and $\hat{\mathbf{R}}_{1,i}$, $\hat{\mathbf{R}}_2$ are known from the two cases above, and based on Eqs. (4) and (13), the two moving control points depend on the value of λ .

Curve/curve intersection based on cubic HybClip

Given two Bézier curves $\mathbf{P}(t)$, $t \in [\alpha, \beta]$ and $\mathbf{Q}(s)$, $s \in [\xi, \eta]$, in this section, a cubic hybrid clipping algorithm is proposed for computing all intersections.

2D curve/curve intersection

The algorithm for two planar Bézier curves is first discussed. This algorithm is presented in Algorithm 1, and illustrated in Fig. 2.

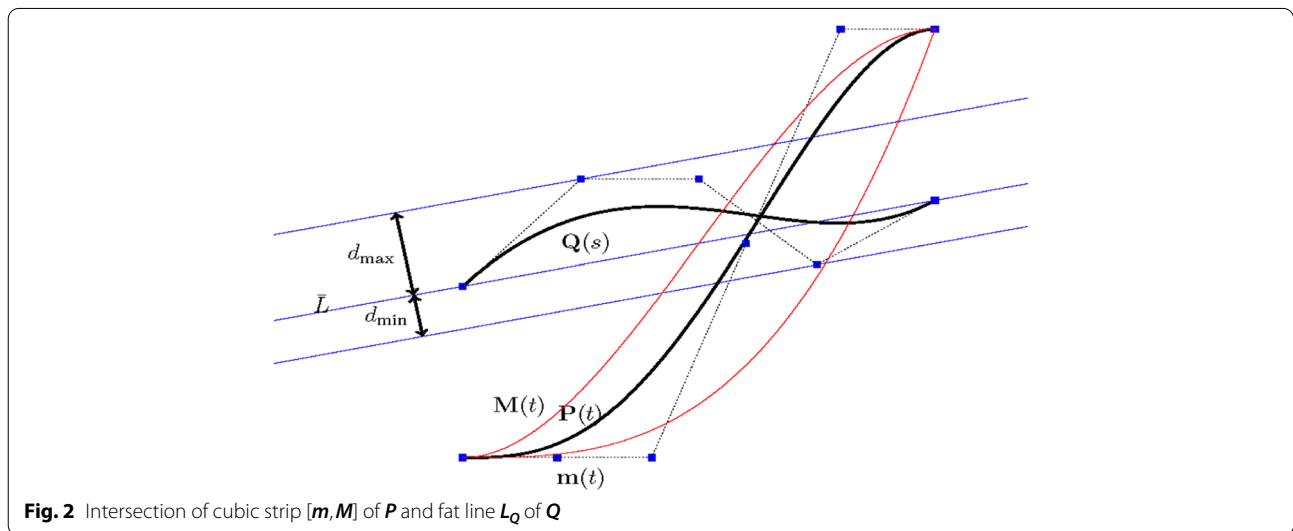
Algorithm 1 HybridClip($\mathbf{P}, \mathbf{Q}, [\alpha, \beta], [\xi, \eta], k, \varepsilon$)

```

1 if  $\beta - \alpha > \varepsilon$  and  $\eta - \xi > \varepsilon$  then
2   if  $\beta - \alpha > \eta - \xi$  then
3      $\mathbf{L}_Q \leftarrow$  the fat line of  $\mathbf{Q}$ 
4      $\hat{\mathbf{P}}(t) \leftarrow$  the cubic hybrid curve of  $\mathbf{P}$ 
5      $\hat{d}(t) \leftarrow$  the distance function of  $\hat{\mathbf{P}}(t)$  to  $\bar{L}$ 
6      $[\hat{d}](t) = [\hat{d}_{\min}(t), \hat{d}_{\max}(t)] \leftarrow$  the cubic polynomial bounds of  $\hat{d}(t)$ 
7     if  $[\hat{d}](t)$  don't intersect the fat line  $\mathbf{L}_Q$  in  $[\alpha, \beta]$  then
8       return  $(\emptyset)$ 
9     else
10      Find intervals  $[\alpha_i, \beta_i]$ ,  $i = 1, \dots, k$ ,  $k \in \{1, 2, 3\}$  by intersecting  $\mathbf{L}_Q$  with  $[\hat{d}](t)$ 
11      if  $\max_i |\alpha_i - \beta_i| \geq \frac{1}{2}|\alpha - \beta|$  then
12        return HybridClip ( $\mathbf{P}, \mathbf{Q}, [\alpha, \frac{1}{2}(\alpha + \beta)], [\xi, \eta], k + 1, \varepsilon$ )
13         $\cup$  HybridClip ( $\mathbf{P}, \mathbf{Q}, [\frac{1}{2}(\alpha + \beta), \beta], [\xi, \eta], k + 1, \varepsilon$ )
14      else
15        return  $\cup_{i=1}^k$  HybridClip ( $\mathbf{P}, \mathbf{Q}, [\alpha_i, \beta_i], [\xi, \eta], k + 1, \varepsilon$ )
16      end if
17    else
18      return HybridClip ( $\mathbf{Q}, \mathbf{P}, [\xi, \eta], [\alpha, \beta], k, \varepsilon$ )
19    end if
20  else
21    return  $([\alpha, \beta], [\xi, \eta])$ 
22  end if

```

In each step, one curve is bounded by two lines, called fat lines, which were first introduced in ref. [5]. Let \bar{L} be a line passing through \mathbf{P}_0 and \mathbf{P}_n of a degree n Bézier curve, $\mathbf{P}(t)$, and suppose \bar{L} has an implicit equation:



$$d(x, y) = ax + by + c = 0, \left(a^2 + b^2 = 1 \right) \tag{20}$$

The fat line of \mathbf{P} is defined as a region

$$\mathbf{L_P} = \{ (x, y) | d(x, y) \in [d_{\min}, d_{\max}] \} \tag{21}$$

where $[d_{\min}, d_{\max}] = [\min_{0 \leq i \leq n} d(\mathbf{P}_i), \max_{0 \leq i \leq n} d(\mathbf{P}_i)]$, and $d(\mathbf{P}_i) = ax_i + by_i + c$, $\mathbf{P}_i = (x_i, y_i)$.

The steps of Algorithm 1 are described in more detail in the following:

- (1) In line 1, if the intervals are within the specified level of accuracy, the parameter intervals corresponding to the intersection in line 21 can be directly obtained.
- (2) In line 2, the curve with a larger priority parameter interval is always clipped.
- (3) In line 5, $\hat{d}(t)$ is a cubic hybrid polynomial in the Bernstein basis,

$$\hat{d}(t) = d(\hat{\mathbf{P}}(t)) = (1-t)^3 \hat{d}_0 + 3t(1-t)^2 \hat{d}_1(t) + 3t^2(1-t) \hat{d}_2(t) + t^3 \hat{d}_3 \tag{22}$$

where

$$i = 0, 3, \hat{d}_i = d(\hat{\mathbf{P}}_i)$$

$$i = 1, 2, \hat{d}_i(t) = d(\hat{\mathbf{P}}_i(t)) = \sum_{j=0}^{n-3} d(\hat{\mathbf{P}}_{i,j}) B_j^{n-3}(t)$$

- (4) In line 6, to obtain the cubic lower and upper bounds, defining $[\hat{d}_1], [\hat{d}_2]$ as the intervals containing the coefficients $d(\hat{\mathbf{P}}_{1,j})$ and $d(\hat{\mathbf{P}}_{2,j})$ of $\hat{d}_1(t)$, $\hat{d}_2(t)$, respectively, $\hat{d}(t)$ is bound using an interval Bernstein polynomial [16]:

$$\begin{aligned} [\hat{d}](t) &= B_0^3(t)\hat{d}_0 + B_1^3(t)[\hat{d}_1] + B_2^3(t)[\hat{d}_2] + B_3^3(t)\hat{d}_3 \\ [\hat{d}_i] &= [\hat{d}_{i,\min}, \hat{d}_{i,\max}] = \left[\min_{0 \leq j \leq n-3} d(\hat{\mathbf{P}}_{i,j}), \max_{0 \leq j \leq n-3} d(\hat{\mathbf{P}}_{i,j}) \right] \end{aligned} \tag{23}$$

The lower and upper bounds of $[\hat{d}](t)$ are defined through cubic polynomials in a simple manner:

$$\begin{aligned} \hat{d}_{\min}(t) &= B_0^3(t)\hat{d}_0 + B_1^3(t)\hat{d}_{1,\min} + B_2^3(t)\hat{d}_{2,\min} + B_3^3(t)\hat{d}_3 \\ \hat{d}_{\max}(t) &= B_0^3(t)\hat{d}_0 + B_1^3(t)\hat{d}_{1,\max} + B_2^3(t)\hat{d}_{2,\max} + B_3^3(t)\hat{d}_3 \end{aligned} \tag{24}$$

Hence, $\hat{d}_{\min}(t) \leq \hat{d}(t) \leq \hat{d}_{\max}(t)$. To obtain a tighter bound $[\hat{d}](t)$, the following optimization function is used:

$$\begin{aligned} &\min \{ \hat{d}_{\max}(t) - \hat{d}_{\min}(t) \} \\ &= \min \{ \hat{d}_{1,\max} - \hat{d}_{1,\min} + \hat{d}_{2,\max} - \hat{d}_{2,\min} \} \\ &= \min \left\{ \max_j (\hat{d}_{1,j}) - \min_j (\hat{d}_{1,j}) + \max_j (\hat{d}_{2,j}) - \min_j (\hat{d}_{2,j}) \right\} \end{aligned} \tag{25}$$

where $\hat{d}_{i,j} = d(\hat{\mathbf{P}}_{i,j})$, $i = 1, 2$, and $j \in \{0, 1, \dots, n-3\}$. In Eq. (22), if the second moving point of $\hat{d}(t)$ is a fixed point

denoted by \hat{q}_2 , and the first moving point is denoted by $\hat{q}_1(t)$, the following is obtained:

$$\hat{q}(t) = B_0^3(t)\hat{d}_0 + B_1^3(t)\hat{q}_1(t) + B_2^3(t)\hat{q}_2 + B_3^3(t)\hat{d}_3 \tag{26}$$

If the first moving point of $\hat{d}(t)$ is a fixed point denoted by \hat{r}_1 , and the second moving point is denoted by $\hat{r}_2(t)$, the following is obtained:

$$\hat{r}(t) = B_0^3(t)\hat{d}_0 + B_1^3(t)\hat{r}_1 + B_2^3(t)\hat{r}_2(t) + B_3^3(t)\hat{d}_3 \tag{27}$$

There exists $\lambda \in [0, 1]$ such that

$$(1 - \lambda)\hat{q}(t) + \lambda\hat{r}(t) = \hat{d}(t) \tag{28}$$

From Eqs. (19) and (25), the problem becomes linear, i.e.,

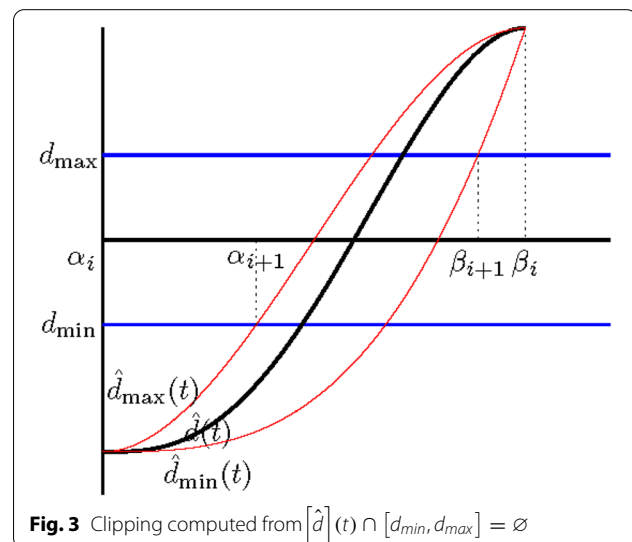
$$\begin{aligned} &\min \{ \hat{d}_{\max}(t) - \hat{d}_{\min}(t) \} \\ &= (1 - \lambda) \left(\max_j (\hat{q}_{1,j}) - \min_j (\hat{q}_{1,j}) \right) + \lambda \left(\max_j (\hat{r}_{2,j}) - \min_j (\hat{r}_{2,j}) \right) \end{aligned} \tag{29}$$

where $j \in \{0, 1, \dots, n-3\}$. Let $a = \max_j (\hat{q}_{1,j}) - \min_j (\hat{q}_{1,j})$ and $b = \max_j (\hat{r}_{2,j}) - \min_j (\hat{r}_{2,j})$. If $a \geq b$, $\lambda = 1$ is set in Eq. (28). Otherwise, $\lambda = 0$. The tighter cubic bounds of $\hat{d}(t)$ can then be obtained using Eqs. (23) and (24).

- (5) In lines 7–10, the intervals of t for which $\mathbf{P}(t)$ lies outside of $\mathbf{L_Q}$ correspond to regions where $[\hat{d}](t) \cap [d_{\min}, d_{\max}] = \emptyset$, as shown in Fig. 3. The values $t \in [0, 1]$, for which $\hat{d}_{\min}(t)$ and $\hat{d}_{\max}(t)$ cross d_{\min} and d_{\max} , correspond to the roots of

$$\begin{aligned} \hat{d}_{\min}(t) &= d_{\min}, \hat{d}_{\max}(t) = d_{\max} \\ \hat{d}_{\max}(t) &= d_{\min}, \hat{d}_{\min}(t) = d_{\max} \end{aligned} \tag{30}$$

Because $\hat{d}_{\min}(t)$ and $\hat{d}_{\max}(t)$ are cubic polynomials, these roots can be solved directly using the cubic formula.



(6) In line 7, if the intersection is empty, no intersection exists between the two curves. In line 11, if the lengths of these intervals are sufficiently small compared to the previous intervals $[\alpha, \beta]$, HybridClip is applied to line 14. Otherwise, the curve is subdivided into two subsegments and applies HybridClip to the two halves (line 12).

3D curve/curve intersection

The above algorithm can be naturally generalized to handle 3D Bézier curve/curve intersection problems. In the 3D case, “fat lines” in 2D are replaced with several bounding planes, which are called “fat planes.”

Plane \bar{L} passes through the end control points \mathbf{P}_0 and \mathbf{P}_n of a degree n Bézier curve, $\mathbf{P}(t)$. Because a plane consists of three points that are not collinear, an arbitrary control point is simply chosen that is not on the endpoint line. Here, \bar{L} is represented using the implicit equation

$$d(x, y, z) = ax + by + cz + e = 0, (a^2 + b^2 + c^2 = 1) \tag{31}$$

The fat plane containing curve $\mathbf{P}(t)$ and its control points are defined as

$$\mathbf{LP} = \{ (x, y, z) | d(x, y, z) \in [d_{\min}, d_{\max}] \} \tag{32}$$

where $[d_{\min}, d_{\max}] = [\min_{0 \leq i \leq n} d(\mathbf{P}_i), \max_{0 \leq i \leq n} d(\mathbf{P}_i)]$, and $d(\mathbf{P}_i) = ax_i + by_i + cz_i + e$, $\mathbf{P}_i = (x_i, y_i, z_i)$. The distance from one curve in a cubic hybrid form is then bound to the fat plane using two cubic polynomials and a strip domain containing the intersections is computed, which is similar to Algorithm 1 described in 2D curve/curve intersection section.

Results

Proof for the convergence rate

Although Yuan’s method [14] is based on cubic HybridClip, it is mainly used to solve univariate polynomial root problems. However, a theoretical convergence rate or proof is not provided.

In this section, the theoretical results are provided on the convergence rate of the new curve/curve intersection algorithm. This begins with two technical lemmas:

Lemma 1. For any given polynomial P , there exists two constants C_p and D_p depending solely on P , such that for all intervals $[\alpha, \beta] \subseteq [0, 1]$ the lower bound m and the upper bound M generated in line 6 of Algorithm 1 satisfy

$$\delta_{\min} = \|P - m\|_{\infty}^{[\alpha, \beta]} \leq C_p h^4 \quad \text{and} \quad \delta_{\max} = \|P - M\|_{\infty}^{[\alpha, \beta]} \leq D_p h^4 \tag{33}$$

where .

Proof. According to Eqs. (22) and (24), $P(\alpha) = m(\alpha)$, $P(\beta) = m(\beta)$, and $P(t) \geq m(t), \forall t \in [\alpha, \beta]$, and thus

$$P(t) - m(t) = (t - \alpha)(\beta - t)(P_1(t) - m_1(t)) \geq 0 \tag{34}$$

where $P_1(t)$ is a continuous function of degree $n - 2$, and $m_1(t)$ is a linear function. Let $g(t) = b_0(\beta - t) + b_1(t - \alpha)$ be a line passing through the lowest control point and parallel to the line connecting the end points of $P_1(t)$, such that $P_1(t) - g(t) \geq 0, \forall t \in [\alpha, \beta]$, and thus

$$P_1(t) - m_1(t) \leq C(P_1(t) - g(t)) \tag{35}$$

where the constant C depends solely on P .

$$\begin{aligned} P_1(t) - g(t) &= \sum_{i=0}^{n-2} a_i B_{i, [\alpha, \beta]}^{n-2}(t) - \sum_{i=0}^1 b_i B_{i, [\alpha, \beta]}^1(t) \\ &= \sum_{i=0}^{n-2} (a_i - c_i) B_{i, [\alpha, \beta]}^{n-2}(t), a_i \geq c_i, \forall i \\ &= (\beta - t)P_2(t) + (t - \alpha)P_3(t) \end{aligned} \tag{36}$$

where $\{c_i\}_{i=0}^{n-2}$ are the control points of g after the degree elevation [1],

$$P_2(t) = \sum_{i=0}^{n-3} (a_i - c_i) \binom{n-2}{i} (\beta - t)^{i-1} (t - \alpha)^{n-2-i} \geq 0$$

$$\text{and } P_3(t) = (a_{n-2} - c_{n-2})(t - \alpha)^{n-3} \geq 0.$$

Let t_1, t_2 be the minimum values of $P_2(t), P_3(t)$ in $[\alpha, \beta]$, respectively, i.e.,

$$\begin{aligned} \forall t \in [\alpha, \beta] : P_2(t) &\leq C_1(P_2(t) - P_2(t_1)) = C_1 P_2'(s_1)(t - t_1) \leq C_3(\beta - \alpha) \\ \text{and } P_3(t) &\leq C_2(P_3(t) - P_3(t_2)) = C_2 P_3'(s_2)(t - t_2) \leq C_4(\beta - \alpha) \end{aligned} \tag{37}$$

where $s_1, s_2 \in [\alpha, \beta]$. Hence,

$$P_1(t) - g(t) \leq C_3(\beta - t)(\beta - \alpha) + C_4(t - \alpha)(\beta - \alpha) \leq C_5(\beta - \alpha)^2 \tag{38}$$

From Eqs. (34), (35), and (38),

$$\begin{aligned} |P(t) - m(t)| &\leq C(t - \alpha)(\beta - t)(P_1(t) - g(t)) \\ &\leq C(t - \alpha)(\beta - t)C_5(\beta - \alpha)^2 \leq C_p(\beta - \alpha)^4 = C_p h^4 \end{aligned} \tag{39}$$

Similarly, $|M(t) - P(t)| < D_p h^4$.

Lemma 2. For any given polynomial P , there exist constants C_i^P, D_i^P , with $i = 0, 1, 2, 3$ depending solely on P , such that for all intervals $[\alpha, \beta] \subseteq [0, 1]$ the lower bound m and upper bound M generated in line 6 of Algorithm 1 for $\forall i \in \{0, 1, 2, 3\}$ satisfy

$$\|P^{(i)} - m^{(i)}\|_{\infty}^{[\alpha, \beta]} \leq C_i^P h^{4-i} \quad \text{and} \quad \|P^{(i)} - M^{(i)}\|_{\infty}^{[\alpha, \beta]} \leq D_i^P h^{4-i} \tag{40}$$

where $h = \beta - \alpha, \|r\|_{\infty}^{[\alpha, \beta]} = \max_{t \in [\alpha, \beta]} |r(t)|$.

Proof A new norm in $[\alpha, \beta]$ is introduced as

$$\|r\|_*^{[\alpha, \beta]} = \|r\|_{\infty}^{[\alpha, \beta]} + h \|r'\|_{\infty}^{[\alpha, \beta]} + h^2 \|r''\|_{\infty}^{[\alpha, \beta]} + h^3 \|r^{(3)}\|_{\infty}^{[\alpha, \beta]} \tag{41}$$

According to the equivalence of norms in a finite-dimensional real linear space, there exists a constant C such that

$$\|r\|_*^{[\alpha, \beta]} \leq C \|r\|_{\infty}^{[\alpha, \beta]} \tag{42}$$

where the constant C does not depend on the intervals $[\alpha, \beta]$, again owing to the affine invariance. Using arguments similar to those in the previous proof, let $r = P - m$,

$$\begin{aligned} & \|P - m\|_{\infty}^{[\alpha, \beta]} \\ &= \|P - m\|_{\infty}^{[\alpha, \beta]} + h \|P' - m'\|_{\infty}^{[\alpha, \beta]} + h^2 \|P'' - m''\|_{\infty}^{[\alpha, \beta]} + h^3 \|P^{(3)} - m^{(3)}\|_{\infty}^{[\alpha, \beta]} \\ &\leq C \|P - m\|_{\infty}^{[\alpha, \beta]} \leq C_P h^4 \end{aligned} \tag{43}$$

Similarly, $\|P - M\|_{\infty}^{[\alpha, \beta]} \leq C \|P - M\|_{\infty}^{[\alpha, \beta]} \leq D_P h^4$.

From the above lemmas, the convergence rate can be analyzed using the HybClip algorithm. In Algorithm 1, if $\mathbf{Q} = \mathbf{0}$, the curve/curve intersection problem $\mathbf{P}(t) = \mathbf{Q}(s)$ becomes a root-finding problem $\mathbf{P}(t) = \mathbf{0}$; that is, the cubic HybClip technique may be used to compute the roots of the polynomials and intersections of the two curves. These two cases are discussed separately.

Theorem 2. (Single root) *If polynomial P has a root t^* in $[\alpha, \beta]$, and provided that this root has multiplicity 1, the sequence of the lengths of the intervals generated through cubic HybClip containing that root has the convergence rate $d = 4$.*

Proof. Suppose that $([\alpha_i, \beta_i])_{i=0, 1, 2, \dots}$, which converges to t^* , is a sequence of intervals generated by Algorithm 1, with lengths $h_i = \beta_i - \alpha_i$. It is assumed that the first derivative satisfies $P'(t^*) > 0$ (otherwise, the polynomial $-P$ can be considered instead of P).

Two cubic Bernstein polynomials m and M can be obtained as the lower and upper bounds of P in $[\alpha_i, \beta_i]$ based on line 6 of Algorithm 1. Because P' is continuous, and owing to Lemma 2, the following inequalities

$$\begin{aligned} & \|P' - P'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{4} P'(t^*) \quad \text{and} \quad \|m' - P'(t)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{4} P'(t^*) \\ & \|M' - P'(t)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{4} P'(t^*) \end{aligned} \tag{44}$$

hold for all but a finite number of values of i . These three inequalities above imply that

$$\begin{aligned} & \|m' - P'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \|P' - P'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} + \|m' - P'\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{2} P'(t^*) \\ & \|M' - P'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \|P' - P'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} + \|M' - P'\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{2} P'(t^*) \end{aligned} \tag{45}$$

and hence

$$\forall t \in [\alpha_i, \beta_i] : m'(t) \geq \frac{1}{2} P'(t^*), M'(t) \geq \frac{1}{2} P'(t^*) \tag{46}$$

From Lemma 1, the vertical height $\delta = \delta_{\min} + \delta_{\max}$ of m and M is bounded by $C_P h_i^4$. Thus, the length h_i of the intervals satisfies

$$h_{i+1} \leq \frac{2\delta}{P'(t^*)} \leq \frac{2C_P}{P'(t^*)} h_i^4 \tag{47}$$

for all but a finite number of values of i (Fig. 4).

For other clipping techniques [8, 9], multiple roots reduce the convergence rate. The convergence rate of cubic HybClip is now discussed in the case of double roots, as illustrated in Fig. 5.

Theorem 3. (Double root) *If the polynomial P has a root t^* in $[\alpha, \beta]$, and provided that this root has multiplicity 2,*

the sequence of the lengths of the intervals generated by cubic HybClip containing that root has the convergence rate $d = 2$.

Proof Similar to the proof of the previous theorem, the sequence of intervals $([\alpha_i, \beta_i])_{i=0, 1, 2, \dots}$ is analyzed with lengths $h_i = \beta_i - \alpha_i$ generated by Algorithm 1, which contains the double root. It is assumed that the second derivative satisfies $P'' > 0$. Otherwise, polynomial $-P$ can be considered instead of P .

Again, two cubic Bernstein polynomials m and M can be obtained as the lower and upper bounds of P in $[\alpha_i, \beta_i]$. Because P'' is continuous, and based on Lemma 2, the inequalities

$$\|P'' - P''(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{4} P''(t^*) \quad \text{and} \quad \|m'' - P''(t)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{4} P''(t^*) \tag{48}$$

hold for all but a finite number of values of i . These two inequalities imply that

$$\|m'' - P''(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \|P'' - P''(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} + \|m'' - P''\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{1}{2} P''(t^*) \tag{49}$$

and thus $m''(t) \geq \frac{1}{2} P''(t^*)$, $\forall t \in [\alpha_i, \beta_i]$. Letting $\tau = t - t^*$, and based on

$$\tilde{m}(\tau) = m(t) = b_3 \tau^3 + b_2 \tau^2 + b_1 \tau + b_0, b_i = \frac{1}{i!} m^{(i)}(t^*) \tag{50}$$

$|b_2| = \frac{1}{2} m''(t^*) \geq \frac{1}{4} P''(t^*) > 0$. From Lemmas 1 and 2,

$$\begin{aligned} |b_0| &= |m(t^*)| = |m(t^*) - P(t^*)| \leq C_{0P} h_i^4 \\ |b_1| &= |m'(t^*)| = |m'(t^*) - P'(t^*)| \leq C_{1P} h_i^3 \\ |b_3| &= \left| \frac{1}{6} m^{(3)}(t^*) \right| \leq \frac{1}{6} |P^{(3)}(t^*)| + \frac{1}{6} |m^{(3)}(t^*) - P^{(3)}(t^*)| \\ &\leq \frac{1}{6} |P^{(3)}(t^*)| + \frac{1}{6} C_{3P} h_i := D_{3P} \end{aligned} \tag{51}$$

Letting t_1, t_2 be the roots of m , $t^* \in [t_1, t_2]$, and $\tau_2 = t_2 - t^* > 0, \tau_1 = t_1 - t^* < 0$, the following is obtained:

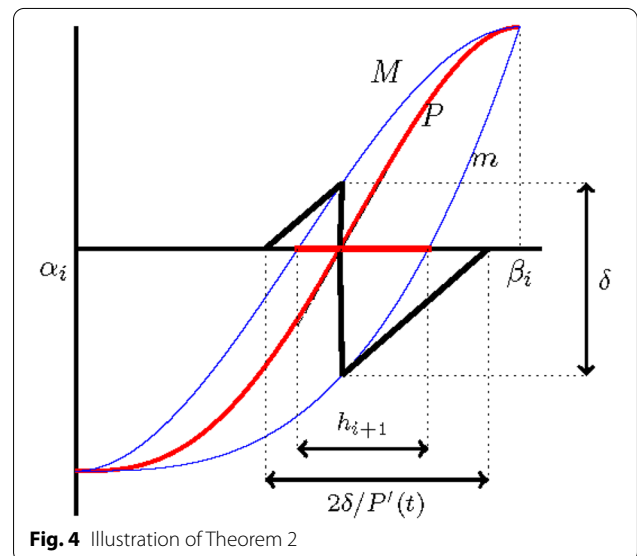


Fig. 4 Illustration of Theorem 2

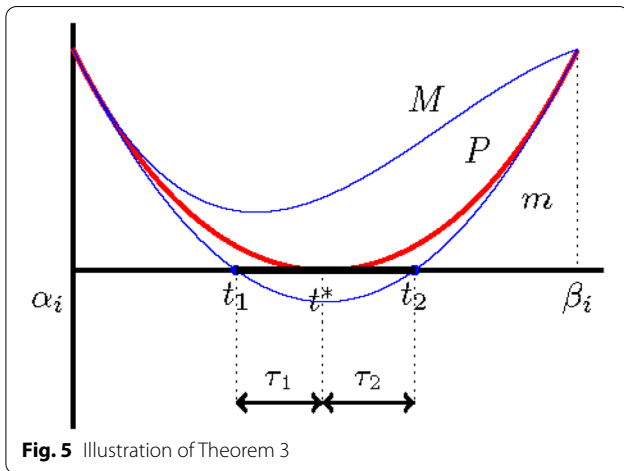


Fig. 5 Illustration of Theorem 3

$$|b_2 \tau_1^2| \leq |b_3 \tau_1^3| + |b_1 \tau_1| + |b_0| \leq \tau_1^2 \cdot D_{3P} |\tau_1| + C_{1P} h_i^4 + C_{0P} h_i^4 := \tau_1^2 \cdot D_{3P} |\tau_1| + D_{2P} h_i^4 \quad (52)$$

Because $\tau_1 \leq h_i$ and $h_i \rightarrow 0$, $D_{3P} |\tau_1| \rightarrow 0$,

$$|b_2 \tau_1^2| \leq \frac{1}{2} |b_2| |\tau_1^2| + D_{2P} h_i^4 \quad (53)$$

for a sufficiently large i . Therefore, $D_{2P} h_i^4 \geq \frac{1}{2} |b_2| |\tau_1^2| \geq \frac{1}{8} P''(t^*) |\tau_1^2|$, and hence

$$\tau_1 \leq \left(\frac{8D_{2P}}{P''(t^*)} \right)^{\frac{1}{2}} h_i^2 \quad (54)$$

Similarly, the following bound for t_2 is obtained:

$$\tau_2 \leq \left(\frac{8D'_{2P}}{P''(t^*)} \right)^{\frac{1}{2}} h_i^2 \quad (55)$$

Because $\tau_1 < 0$, $\tau_2 > 0$,

$$h_{i+1} = |t_2 - t_1| = \tau_2 - \tau_1 \leq \left(\left(\frac{8D_{2P}}{P''(t^*)} \right)^{\frac{1}{2}} - \left(\frac{8D'_{2P}}{P''(t^*)} \right)^{\frac{1}{2}} \right) h_i^2 \quad (56)$$

Hence, the sequence $(h_i)_{i=0, 1, 2, \dots}$ has a convergence rate of 2.

From Theorems 2 and 3, it can be seen that the new algorithm has a higher convergence rate compared with geometry interval clipping [11] and quadratic clipping [8] when computing all roots of a univariate polynomial equation. The following theorem provides the convergence rate for the curve/curve intersection problems.

Theorem 4. Suppose $\mathbf{f}(t)$, $\mathbf{g}(s)$ have a transversal intersection ($\mathbf{f}(t^*) \times \mathbf{g}(s^*) \neq 0$) at $\mathbf{p}^* = \mathbf{f}(t^*) = \mathbf{g}(s^*)$. Furthermore, supposing that $[\alpha_p, \beta_i]_{i=0, 1, 2, \dots}$ is the sequence of generated intervals that contain t^* , and $[\xi_p, \eta_i]_{i=0, 1, 2, \dots}$ is the corresponding sequence of generated intervals

that contain s^* , there then exist constants C_1, C_2, C_3, C_4 depending solely on \mathbf{f} and \mathbf{g} , such that

$$h_{i+1, \mathbf{f}} \leq C_1 h_{i, \mathbf{f}}^3 + C_2 h_{i, \mathbf{g}}^2 \quad \text{and} \quad h_{i+1, \mathbf{g}} \leq C_3 h_{i, \mathbf{g}}^3 + C_4 h_{i, \mathbf{f}}^4 \quad (57)$$

Proof From line 11 of Algorithm 1, it can be seen that the length of intervals $[\xi_p, \eta_i]$ tends toward zero as i tends toward infinity, that is, the interval $[\xi_p, \eta_i]$ tends toward s^* .

Let \bar{L}_g be the line or plane that passes through the endpoints $\mathbf{b}_0, \mathbf{b}_m$ of \mathbf{g} in $[\xi_p, \eta_i]$. Denote \mathbf{n} as the unit normal vector of \bar{L}_g . Then, the distance function from $\mathbf{f}(t)$ to \bar{L}_g is defined as

$$d(t) = \mathbf{n} \cdot (\mathbf{f}(t) - \mathbf{b}_0) \quad (58)$$

Denote \mathbf{T}_f^* as the tangent line of \mathbf{f} at t^* . Let $\phi \in [0, \frac{\pi}{2}]$ be the angle between \mathbf{T}_f^* and \bar{L}_g , and $\theta \in [0, \frac{\pi}{2}]$ be the angles between \mathbf{T}_f^* and $\mathbf{b}_0 \mathbf{b}_m$. As $h_{i, \mathbf{g}} = [\xi_p, \eta_i]$ tends toward 0, the line or plane \bar{L}_g converges at $\mathbf{b}_0 \mathbf{b}_m$, and angle ϕ converges at θ . Thus, for a sufficiently small $h_{i, \mathbf{g}}$, $\phi > \frac{\theta}{2} > 0$, and thus $0 < \sin(\frac{\theta}{2}) < \sin(\phi) \leq 1$.

The angle ρ between $\mathbf{f}(t^*)$ and \mathbf{n} is either $\rho = \frac{\pi}{2} + \phi$ or $\rho = \frac{\pi}{2} - \phi$. Using this, the derivative of the distance function can be bound at the intersection as

$$|d'(t^*)| = |\mathbf{n} \cdot \mathbf{f}'(t^*)| = \|\mathbf{f}'(t^*)\| \left| \cos\left(\frac{\pi}{2} \pm \phi\right) \right| = \|\mathbf{f}'(t^*)\| \sin(\phi) > 0 \quad (59)$$

Because $d'(t^*) \neq 0$, and for convenience, $w = d'(t^*) > 0$ is denoted (otherwise, the vector $-\mathbf{n}$ can be considered instead of \mathbf{n}).

Because $d'(t)$ is continuous, the inequality

$$\|d' - d'(t^*)\|_{\infty}^{[\alpha_i, \beta_i]} < \frac{w}{2} \quad (60)$$

holds for all but a finite number of values of i . Hence,

$$\forall t \in [\alpha_i, \beta_i], d'(t) > \frac{w}{2} \quad (61)$$

From line 6 of Algorithm 1, the cubic polynomial bound $[d_m(t), d_M(t)]$ of the distance function $d(t)$ can be obtained. Based on Lemma 2,

$$\|d' - d_m\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{w}{4} \quad \text{and} \quad \|d^{(i)} - d_M(t)\|_{\infty}^{[\alpha_i, \beta_i]} \leq \frac{w}{4} \quad (62)$$

and by Eq. (61), the following is obtained:

$$d'_m(t) \geq \frac{w}{4} \quad \text{and} \quad d'_M(t) \geq \frac{w}{4} \quad (63)$$

From Fig. 6, the bound for $h_{i+1, \mathbf{f}}$ is obtained as

$$h_{i+1, \mathbf{f}} = \beta_{i+1} - \alpha_{i+1} \leq l_1 + l_2 + l_3 \quad (64)$$

$$l_1 + l_3 = \frac{d_{\max} - d_{\min}}{w/4}$$

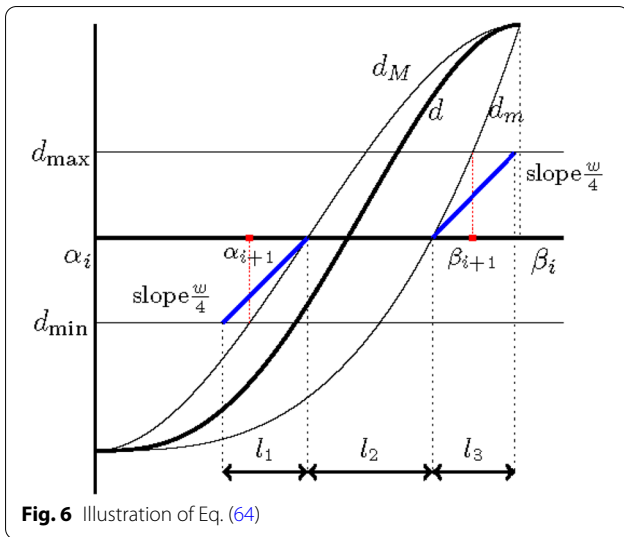


Fig. 6 Illustration of Eq. (64)

Based on Lemma 1, the vertical heights δ_i of d_m and d_M are bounded as follows:

$$\delta_i = \|d_M - d_m\|_\infty^{[\alpha_i, \beta_i]} \leq \|d_M - d\|_\infty^{[\alpha_i, \beta_i]} + \|d - d_m\|_\infty^{[\alpha_i, \beta_i]} \leq C_f h_i^4 \quad (65)$$

Let t_1 and t_2 be the roots of d_m and d_M respectively. From Eq. (63),

$$l_2 = |t_1 - t_2| < \frac{\delta_i}{w/4} = \frac{C_f h_i^4}{w/4} \quad (66)$$

From Eq. (64), the above inequality implies that

$$h_{i+1, f} < \frac{d_{\max} - d_{\min}}{w/4} + \frac{C_f h_i^4}{w/4} \quad (67)$$

This thus implies the first inequality in Eq. (57) from $d_{\max} - d_{\min} < h_{i, g}^2 C_g$ (see the proof of Theorem 6 in ref. [7]). Similarly, in the next iteration step, the following is obtained:

$$h_{i+1, g} \leq C'_g h_{i, g}^4 + C'_f h_{i+1, f}^2 \quad (68)$$

where C'_f is solely dependent on f , and C'_g is solely dependent on g . Based on the first inequality of Eq. (57), the following is obtained:

$$\begin{aligned} h_{i+1, g} &\leq C'_g h_{i, g}^4 + C'_f (C_f h_{i, f}^4 + C_g h_{i, g}^2)^2 \\ &\leq C'_g h_{i, g}^4 + C'_f (C_f^2 h_{i, f}^8 + C_g^2 h_{i, g}^4 + 2C_f C_g h_{i, f}^4 h_{i, g}^2) \\ &\leq (C'_g + C'_f C_g^2) h_{i, g}^4 + (C'_f C_f^2 h_{i, f}^4 + 2C'_f C_f C_g h_{i, f}^2) h_{i, f}^4 \end{aligned} \quad (69)$$

which implies the second inequality.

Note that the property of w being nonzero is key to binding l_1 and l_3 . Therefore, a transversal intersection is required in the proof. From Theorem 4, the two sequences $\{\{\alpha_i, \beta_i\}\}_i$ and $\{\{\xi_i, \eta_i\}\}_i$ of the new intersection algorithm have second- and fourth-order convergence rates, respectively, and the 3D curve intersection problem yields the same results.

Experimental results

In this section, all six algorithms are compared based on three criteria: the amount of time per iteration step, the number of iterations, and the computing time required to achieve a certain accuracy. All algorithms were implemented in C++ on a PC with an 2.60-GHz Intel^(R) Core^(TM) i7-9750H CPU and 16.0 GB of RAM. In all experiments, both curves $P(t)$ and $Q(s)$ have a parameter domain $[0, 1]$.

For convenience, denote Bézier clipping as BezClip [5]; quadratic clipping and cubic clipping based on a degree reduction as 2-DegClip [8] and 3-DegClip [10], respectively; geometry interval clipping as 2-HybClip [11]; and cubic HybClip based on hybrid curves in ref. [14] as 3-HybClip*. In addition, the proposed cubic HybClip algorithm is denoted as 3-HybClip.

To analyze the relationship between the computational effort and the desired accuracy, two examples representing polynomials with transversal and tangent intersections are discussed. The five algorithms are first applied

Table 1 Transversal intersections

(n, m)	ϵ	10^{-6}					10^{-10}				
		BezClip	2-HybClip	3-HybClip	2-DegClip	3-DegClip	BezClip	2-HybClip	3-HybClip	2-DegClip	3-DegClip
(4,4)	$[N_P, N_Q]$	[4,3]	[3,3]	[3,2]	[3,3]	[3,2]	[16,15]	[4,3]	[3,3]	[4,3]	[3,3]
	$t/\mu s$	125	120	120	150	150	90	90	100	125	130
(8,4)	$[N_P, N_Q]$	[4,3]	[3,3]	[3,2]	[3,3]	[3,2]	[15,14]	[4,3]	[3,2]	[4,3]	[3,3]
	$t/\mu s$	160	150	150	190	190	100	120	100	140	130
(8,8)	$[N_P, N_Q]$	[4,4]	[4,4]	[3,3]	[3,3]	[3,3]	[16,16]	[5,4]	[3,3]	[4,4]	[3,3]
	$t/\mu s$	190	180	180	250	250	190	130	140	180	170

Number of iterations $[N_P, N_Q]$ and computing time t (in microseconds) of accuracy ϵ . In addition, (n, m) are degrees of P, Q , respectively

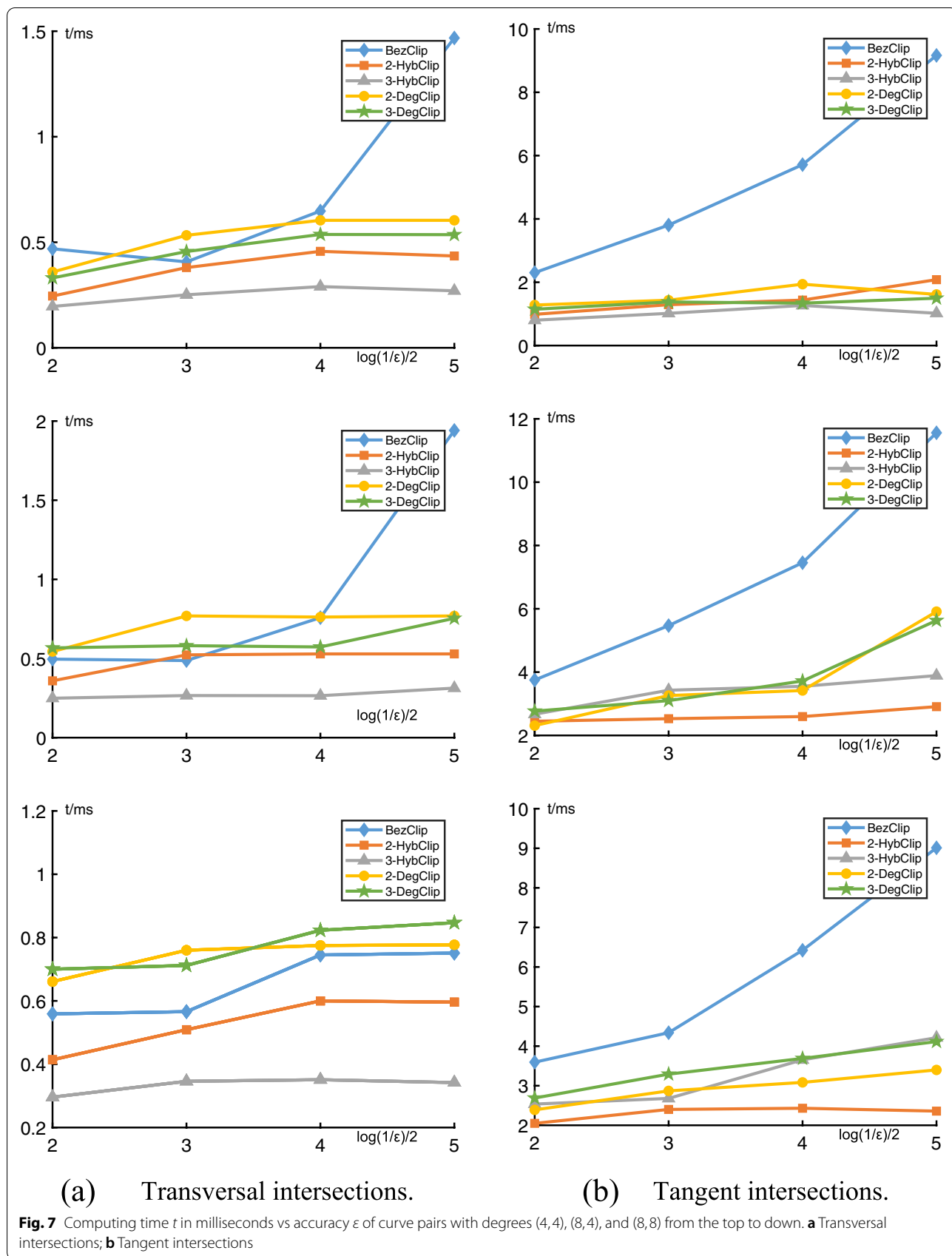


Table 2 Tangent intersections

(n, m)	ϵ	10^{-6}					10^{-10}				
		BezClip	2-HybClip	3-HybClip	2-DegClip	3-DegClip	BezClip	2-HybClip	3-HybClip	2-DegClip	3-DegClip
(4,4)	$[N_P, N_Q]$	[12,10]	[6,5]	[6,5]	[6,6]	[5,5]	[20,20]	[6,6]	[6,6]	[7,6]	[6,5]
	$t/\mu s$	138	117	93	119	138	164	129	85	124	136
(8,4)	$[N_P, N_Q]$	[20,20]	[8,8]	[10,10]	[6,6]	[5,6]	[34,34]	[9,8]	[11,10]	[7,6]	[5,7]
	$t/\mu s$	160	175	170	190	200	160	175	185	232	246
(8,8)	$[N_P, N_Q]$	[18,17]	[7,7]	[10,9]	[7,7]	[4,6]	[32,30]	[8,7]	[10,10]	[8,7]	[5,6]
	$t/\mu s$	170	185	180	226	263	177	157	133	272	282

Number of iterations $[N_P, N_Q]$ and computing time t (in microseconds) of accuracy ϵ . In addition, (n, m) are degrees of P, Q , respectively

to three pairs of Bézier curves with a transversal intersection. Table 1 reports the number of pairs of iterations and the computing time in microseconds of the desired accuracy for computing the transversal intersections between the three curve pairs with various degrees. Figure 7a shows the relationship between the computing time and desired accuracy, and indicates that 3-HybClip based on cubic hybrid curves is significantly improved in comparison with BezClip, 2HybClip, 2-DegClip, and 3-DegClip.

$$\begin{cases} P_4(t) = ((t - 1/2)(t - 3)(t + 1)^2, (t - 1/2)(t - 2)(t + 1)^2) \\ Q_4(s) = ((s - 1/2)(s - 2)(s + 2)^2, (s - 1/2)(s - 2)^2(s + 1)) \end{cases} \quad (70)$$

$$\begin{cases} P_8(t) = ((t - 1/2)(t - 2)^4(t + 1/2)^3, (t - 1/2)(t - 2)^4(t + 1)^3) \\ Q_4(s) = ((s - 1/2)(s - 2)(s + 2)^2, (s - 1/2)(s - 2)^2(s + 1)) \end{cases} \quad (71)$$

$$\begin{cases} P_8(t) = ((t - 1/2)(t - 2)^4(t + 1/2)^3, (t - 1/2)(t - 2)^4(t + 1)^3) \\ Q_8(s) = ((s - 1/2)(s - 1)^3(s + 1)^4, (s - 1/2)(s - 2)^4(s + 1)^3) \end{cases} \quad (72)$$

The five algorithms are applied to three pairs of Bézier curves with tangent intersections. Table 2 and Fig. 7b report the number of pairs of iterations and the computing time in milliseconds of various desired accuracies ϵ for computing the tangent intersections between the three curve pairs with various degrees. Experimental results show that the quadratic and cubic clipping techniques are better than Bézier clipping; however, compared with quadratic clipping based on hybrid curves or a degree reduction, the cubic clipping techniques show no substantial improvements. This is due to the fact that all clipping algorithms require a large number of binary subdivisions for tangent intersections.

$$\begin{cases} P_4(t) = (2t - 1, -4t^4 + 8t^3 - 4t + 1.5) \\ Q_4(s) = (2s - 1, 4s^4 - 8s^3 + 4s - 1) \end{cases} \quad (73)$$

$$\begin{cases} P_8(t) = (2t - 1, -20t^8 + 80t^7 - 112t^6 + 56t^5 - 4t + 1.7031) \\ Q_4(s) = (2s - 1, 4s^4 - 8s^3 + 4s - 1) \end{cases} \quad (74)$$

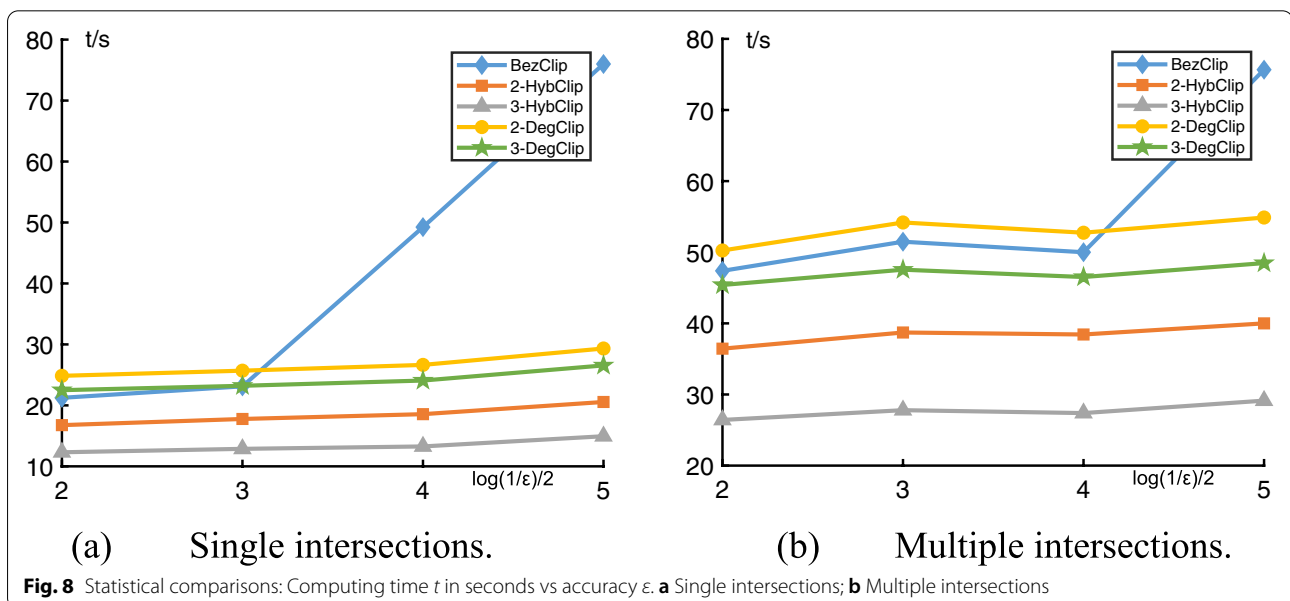


Table 3 Relative computing iterations N and computing time t

ϵ		BezClip	2-HybClip	3-HybClip*	3-HybClip	2-DegClip	3-DegClip
10^{-6}	N	1.36	1.14	1.02	1	1.18	1.02
	t	1.85	1.39	1.08	1	1.95	1.71
10^{-10}	N	1.79	1.16	1.03	1	1.18	1.03
	t	2.60	1.37	1.09	1	1.88	1.66

$$\begin{cases} \mathbf{P}_8(t) = (2t - 1, -20t^8 + 80t^7 - 112t^6 + 56t^5 - 4t + 1.7031) \\ \mathbf{Q}_8(s) = (2s - 1, 20s^8 - 80s^7 + 112s^6 - 56s^5 + 4s - 1.2031) \end{cases} \quad (75)$$

To compare these six algorithms numerically, statistics are generated on 40,000 pairs of randomly generated polynomial curves of degree 4–10 for single and multiple intersections. Figure 8 shows computing time needed to achieve the given accuracy of the five algorithms. The relative computing iterations and computing time for these tests are listed in Table 3. As shown in Table 3, 3-HybClip requires 2% fewer iterations and 8% less time than 3-HybClip* [14]. In addition, 3-HybClip has 2% fewer computing iterations than 3-DegClip, and at least 10% fewer iterations than 2-HybClip and 2-DegClip. With respect to the computing time, 3-HybClip is at least 60% faster than 3-DegClip and 2-DegClip, and at least 30% faster than 2-HybClip.

Conclusions

In this study, an algorithm called 3-HybClip was derived for computing all intersections between two Bézier curves within a given domain. By selecting the moving control points, better bounds were obtained than those in ref. [14]. It was proved that the two sequences of bounded intervals for intersections have second- and fourth-order convergence rates for transversal intersections. The experimental results show that the newly proposed 3-HybClip algorithm requires 2% fewer iterations and 8% less time than 3-HybClip* from ref. [14], 10% fewer iterations than 2-HybClip and 2-DegClip, and at least 30% less time than other techniques such as BezClip, 2-HybClip, 2-DegClip, and 3-DegClip.

Discussion

As discussed in 3D curve/curve intersection section, for 3D curve/curve intersection problems, the “fat planes” are computed to bound a 3D Bézier curve. The distance from one curve in a cubic hybrid form to the fat plane is bound by two cubic polynomials, and a strip domain containing the intersections is then computed. Similarly, in curve/surface intersection problems, “fat planes” can also be used to bound a Bézier surface, and then the distance from the curve to the fat plane is bound by two cubic

polynomials. Then the intersection of “fat planes” and the two cubic polynomials is the strip domain containing the intersections. The details of the algorithm and comparisons with previous approaches are left for future work.

Abbreviations

CAD/CAM: Computer-aided design and manufacturing; HybClip: Hybrid clipping.

Acknowledgements

Not applicable.

Authors' contributions

XL provided conceptualization, supervision, formal analysis, methodology, and investigation; YQW provided visualization and validation; and XL and YQW wrote the paper. The authors have read and approved the final manuscript.

Funding

This work was supported by the National Key R&D Program of China, No. 2020YFB1708900; and Natural Science Foundation of China, No. 61872328.

Availability of data and materials

Code sets and date sets: <https://gitee.com/yaqiong-wu/CurveCurveIntersect.git>.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 18 February 2022 Accepted: 5 June 2022

Published online: 22 June 2022

References

- Farin G (2001) Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann Publishers Inc., San Francisco.
- Koparkar PA, Mudur SP (1983) A new class of algorithms for the processing of parametric curves. *Comput Aided Des* 15(1):41–45. [https://doi.org/10.1016/S0010-4485\(83\)80050-5](https://doi.org/10.1016/S0010-4485(83)80050-5)
- Lane JM, Riesenfeld RF (1980) A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans Pattern Anal Mach Intell* PAMI-2(1):35–46. <https://doi.org/10.1109/TPAMI.1980.4766968>
- Sederberg TW, Parry SR (1986) Comparison of three curve intersection algorithms. *Comput Aided Des* 18(1):58–63. [https://doi.org/10.1016/S0010-4485\(86\)80013-6](https://doi.org/10.1016/S0010-4485(86)80013-6)
- Sederberg TW, Nishita T (1990) Curve intersection using Bézier clipping. *Comput Aided Des* 22(9):538–549. [https://doi.org/10.1016/0010-4485\(90\)90039-F](https://doi.org/10.1016/0010-4485(90)90039-F)
- Patrikalakis NM, Maekawa T (2002) Shape interrogation for computer aided design and manufacturing. Springer, Berlin Heidelberg.
- Schulz C (2009) Bézier clipping is quadratically convergent. *Comput Aided Geom Des* 26(1):61–74. <https://doi.org/10.1016/j.cagd.2007.12.006>

8. Bartoň M, Jüttler B (2007) Computing roots of polynomials by quadratic clipping. *Comput Aided Geom Des* 24(3):125-141. <https://doi.org/10.1016/j.cagd.2007.01.003>
9. Liu LG, Zhang L, Lin BB, Wang GJ (2009) Fast approach for computing roots of polynomials using cubic clipping. *Comput Aided Geom Des* 26(5):547-559. <https://doi.org/10.1016/j.cagd.2009.02.003>
10. Lou Q, Liu LG (2012) Curve intersection using hybrid clipping. *Comput Graph* 36(5):309-320. <https://doi.org/10.1016/j.cag.2012.03.021>
11. North NS (2007) Intersection algorithms based on geometric intervals. Dissertation, Brigham Young University, Utah.
12. Sederberg TW, Kakimoto M (1991) Approximating rational curves using polynomial curves. Farin G (ed) *NURBS for curve and surface design*. SIAM, Philadelphia, 144-158.
13. Liu HX, Li X (2010) Convergence analysis and comparison for geometric interval clipping. *J Comput Aided Des Comput Graph* 22(12):2250-2258.
14. Yuan Q (2012) Study on hybrid clipping method for solving polynomial roots. Dissertation, Zhejiang University.
15. Holliday DJ, Farin GE (1999) A geometric interpretation of the diagonal of a tensor-product Bézier volume. *Comput Aided Geom Des* 16(8):837-840. [https://doi.org/10.1016/S0167-8396\(99\)00004-7](https://doi.org/10.1016/S0167-8396(99)00004-7)
16. Sederberg TW, Farouki RT (1992) Approximation by interval Bézier curves. *IEEE Comput Graph Appl* 12(5):87-95. <https://doi.org/10.1109/38.156018>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
