Computer Aided Geometric Design 38 (2015) 40–50

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

A rational cubic clipping method for computing real roots of a polynomial ☆

Xiao-Diao Chen [a,b,*], Weiyin Ma [b], Yangtian Ye [a]

[a] School of Computer, Hangzhou Dianzi University, Hangzhou, 310018, PR China
[b] Department of MBE, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 8 January 2015
Received in revised form 8 August 2015
Accepted 10 August 2015
Available online 10 September 2015

Keywords:
Approximation order
Rational cubic clipping method
Root-finding
Convergence rate

ABSTRACT

Many problems in computer aided geometric design and computer graphics can be turned into a root-finding problem of a polynomial equation. Among various solutions, clipping methods based on the Bernstein–Bézier form usually have good numerical stability. A traditional clipping method using polynomials of degree r can achieve a convergence rate of $r+1$ for a single root. It utilizes two polynomials of degree r to bound the given polynomial $f(t)$ of degree n, where $r=2,3$, and the roots of the bounding polynomials are used for clipping off the subintervals containing no roots of $f(t)$. This paper presents a rational cubic clipping method for finding the roots of a polynomial $f(t)$ within an interval. The bounding rational cubics can achieve an approximation order of 7 and the corresponding convergence rate for finding a single root is also 7. In addition, differently from the traditional cubic clipping method solving the two bounding polynomials in $O(n^2)$, the new method directly constructs the two rational cubics in $O(n)$ which can be used for bounding $f(t)$ in many cases. Some examples are provided to show the efficiency, the approximation effect and the convergence rate of the new method.

1. Introduction

Many problems in computer aided geometric design and computer graphics can be turned into a root-finding problem of polynomial equations, such as curve/surface intersection (Efremov et al., 2005; Liu et al., 2009; Nishita et al., 1990; Patrikalakis and Maekawa, 2002), point projection (Chen et al., 2008), collision detection (Choi et al., 2006; Lin and Gottschalk, 1998), and bisectors/medial axes computation (Elber and Kim, 2001). In principle, a system of polynomial equations of multiple variables can be turned into a univariate polynomial equation by using the resultant theory. This paper discusses the root-finding problem of a univariate polynomial equation within an interval.

Many references turn the given polynomial $f(t)$ into its power series, and a collection of related references can be found in McNamee (1993–2002), Isaacson and Keller (1966), Mourrain and Pavone (2005), Reuter et al. (2007), Rouillier and Zimmermann (2004). The Bernstein–Bézier form of $f(t)$ has a good numerical stability with respect to perturbations of the coefficients (Farouki et al., 1987; Farouki and Goodman, 1996; Jüttler, 1998). Several clipping methods based on the Bernstein–Bézier form are developed (Bartoň and Jüttler, 2007; Liu et al., 2009; Morken and Reimers, 2007; Sederberg and Nishita, 1990). Note that the number of zeros of a Bézier function is less or equal to that of its control polygon. The method

☆ This paper has been recommended for acceptance by Michael Floater.

* Corresponding author.
E-mail addresses: xiaodiao@hdu.edu.cn (X.-D. Chen), mewma@cityu.edu.hk (W. Ma).

http://dx.doi.org/10.1016/j.cagd.2015.08.002

in Morken and Reimers (2007) utilizes the corresponding control polygon to approximation $f(t)$, in which the zeros of the control polygon are used to approximate the zeros of $f(t)$ from one side. The method in Morken and Reimers (2007) achieves a convergence rate of 2 for a simple root. In principle, a B-spline (or Bézier) curve is bounded by the convex hull of its control polygon, the corresponding roots are then bounded by the roots of the convex hull. The corresponding approximation order of the approach using convex hull is 2 (Schulz, 2009). Comparing with the method in Morken and Reimers (2007), the r-clipping method in Bartoň and Jüttler (2007), Liu et al. (2009) bounds the zeros of $f(t)$ by using the zeros of two bounding polynomials of degree r, which achieves a higher approximation order $r+1$, where $r=2,3$.

In principle, one can also use rational polynomials to bound $f(t)$ for root finding. A rational quadratic polynomial has five free variables, which can achieve an approximation order 5 to $f(t)$. If two rational quadratics are utilized to bound $f(t)$, one can achieve a convergence rate of 5 for a simple root, which is much higher than that of 3 when using a quadratic clipping polynomial. However, in some cases when the curve $(t, f(t))$ is not convex within $[a,b]$, the denominators of the rational quadratic polynomials for bounding $f(t)$ may have one or more zeros within $[a,b]$, which leads to a bad approximation effect between $f(t)$ and its bounding polynomials.

A rational cubic polynomial, on the other hand, can approximate $f(t)$ in a much better way than that of a rational quadratic polynomial, even in case that $(t, f(t))$ is not convex within the given interval $[a,b]$. This paper presents a rational cubic clipping method which utilizes two rational cubic polynomials to bound $f(t)$ for root-finding. The bounding rational cubics achieve the approximation order 7 to $f(t)$ and the corresponding rational cubic clipping method can achieve a convergence rate of 7 for a simple root, which is much higher than that of 4 of previous cubic clipping methods. In addition, the method proposed in this paper directly constructs two rational cubic polynomials interpolating four positions and three derivatives of $f(t)$, which can bound $f(t)$ in many cases and it leads to a much higher computation efficiency. Some numerical examples are provided to show both higher convergence rate and higher computation efficiency of the new method.

The remainder of this paper is organized as follows. Section 2 provides an outline of clipping methods. Section 3 illustrates the rational cubic clipping method for finding two bounding rational cubics in details. Numerical examples and some further discussions are provided in Section 4, and the conclusions are drawn at the end of this paper.

2. Outline of the clipping methods

Suppose that $f(t)$, $t \in [a,b]$, is the given polynomial of degree n. The basic idea of the clipping methods is to find two bounding polynomials, and then to clip off the subintervals containing no roots of $f(t)$ by using the roots of the bounding polynomials. The clipping process continues until the lengths of the remaining subintervals are less than a given tolerance. Finally, the middle points of the remaining subintervals are recorded as the roots of $f(t)$.

The numerical convergence rate of a clipping method within an interval tends to be $\bar{m}/\bar{k}$, where $\bar{m}$ is the convergence rate of a method for a single root, and $\bar{k}$ is the sum of multiplicities of all of the roots within the interval. If $\bar{k}$ is large for some cases, the numerical convergence rate is very slow. In this paper, at the beginning, we apply the method in Morken and Reimers (2007) to divide the given interval into several sub-intervals by utilizing the zeros of the control polygon of the given Bézier function, which can improve the corresponding convergence rate.

2.1. The algorithm of a clipping step of the clipping method

In each clipping step, one needs to find two bounding polynomials for a given interval $[a,b]$ and then to split the interval $[a,b]$ into several subintervals by using the roots of the bounding polynomials. The subintervals containing no root of $f(t)$ are further clipped off. The computation of the two bounding polynomials is one of the key issues of a clipping step. Different clipping methods may obtain different bounding polynomials. Suppose that the two bounding polynomials are obtained such that $g_1(t) \le f(t) \le g_2(t)$. Note that the roots of $g_i(t)$ is usually easily obtained, it is trivial to check whether or not a root of $g_i(t)$ is a root of $f(t)$. Let Λ be a subinterval of $[a,b]$. We utilize the following lemma to clip the subintervals containing no roots of $f(t)$.

Lemma 1. *If $g_1(t) > 0$ or $g_2(t) < 0$ for all $t \in \Lambda$, then Λ can be removed.*

Proof. Firstly, if $g_1(t) > 0$, we have $f(t) \ge g_1(t) > 0$, for all $t \in \Lambda$. That is to say, Λ contains no root of $f(t)$ and can be removed.

Secondly, if $g_2(t) < 0$, we have $f(t) \le g_2(t) < 0$. Similarly, Λ contains no root of $f(t)$ and can also be removed. □

In this paper, we provide a rational cubic clipping method to find the roots of $f(t)$, in which two rational cubic polynomials are used for bounding $f(t)$. The roots of the two bounding rational cubic polynomials within an interval can be solved by using the Cardano formula (see more details in Liu et al., 2009). Finally, from Lemma 1, the roots of the bounding polynomials can be used for clipping the subintervals containing no roots of $f(t)$.

2.2. The analysis of convergence rate

The analysis of the convergence rate is another key issue of the clipping method, which depends on the approximation order of the bounding polynomials. We have the following theorem.

Theorem 1. *Suppose that the two bounding polynomials $g_i(t)$, $i = 1, 2$, achieve an approximation order m to $f(t)$ within interval $[a, b]$ whose length is small enough for satisfying Eq. (2), then the corresponding convergence rate for a root $t^\star$ of multiplicity k is $\frac{m}{k}$.*

Proof. From the assumption that $t^\star$ has multiplicity k, we have that

$$f^{(j)}(t^\star) = 0, \quad \text{for } j = 0, 1, \cdots, k-1, \quad \text{and} \quad f^{(k)}(t^\star) \neq 0. \tag{1}$$

Combining Taylor's expansion with Eq. (1), there exist a small enough $\eta > 0$ and $\xi_1(t)$ such that

$$\begin{aligned} |f(t) - f(t^\star)| &= |\frac{f^{(k)}(t^\star)}{k!}(t-t^\star)^k + \frac{f^{(k+1)}(\xi_1(t))}{(k+1)!}(t-t^\star)^{k+1}| \\ &> |\frac{1}{2}\frac{f^{(k)}(t^\star)}{k!}(t-t^\star)^k|, \ \forall t \in [t^\star - \eta, t^\star + \eta]. \end{aligned} \tag{2}$$

Let $t_i^\star$ be a root of $g_i(t)$. From Eq. (2), we have that

$$|f(t_i^\star) - f(t^\star)| > |\frac{1}{2}\frac{f^{(k)}(t^\star)}{k!}(t_i^\star - t^\star)^k|, \ i = 1, 2. \tag{3}$$

From the assumption that $g_i(t)$ achieves an approximation order m to $f(t)$ within $[a, b]$, there exists a constant c_i such that

$$|f(t) - g_i(t)| < c_i(b-a)^m, \ \text{for all } t \in [a, b]. \tag{4}$$

Combining Eq. (3) and Eq. (4), we have that

$$\begin{aligned} |\frac{1}{2}\frac{f^{(k)}(t^\star)}{k!}(t_i^\star - t^\star)^k| &< |f(t_i^\star) - f(t^\star)| = |f(t_i^\star)| \\ &= |f(t_i^\star) - g_i(t_i^\star)| \\ &< c_i(b-a)^m. \end{aligned} \tag{5}$$

Let $\bar{C}_i = \frac{2c_i \cdot k!}{f^{(k)}(t^\star)}$. From Eq. (5), we obtain that

$$|t_i^\star - t^\star| < \bar{C}_i^{\frac{1}{k}}(b-a)^{\frac{m}{k}}, \tag{6}$$

which means that the corresponding convergence rate is $\frac{m}{k}$. □

3. Finding two rational cubics for bounding $f(t)$

For the sake of convenience, let

$$\begin{aligned} &h = b - a, \qquad t_j = a + \frac{j}{3}h, \quad j = 0, 1, 2, 3, \\ &d_j = f(t_j) \quad \text{and} \quad v_j = f'(t_j), \quad j = 0, 1, 2, 3, \\ &g_1(t) = (t-t_0)^2(t-t_1)^2(t-t_2)^2(t-t_3), \\ &g_2(t) = (t-t_0)(t-t_1)^2(t-t_2)^2(t-t_3)^2. \end{aligned}$$

Let $\kappa = \sup_{0 \le u \le 1} |(u-0)^2(u-1/3)^2(u-2/3)^2(u-1)| \approx 0.00149921$. By substituting $t = a + uh$, we have that

$$\begin{aligned} &-\kappa h^7 \le g_1(a+uh) = (u-0)^2(u-1/3)^2(u-2/3)^2(u-1) \cdot h^7 \le 0, \\ &0 \le g_2(a+uh) = (u-0)(u-1/3)^2(u-2/3)^2(u-1)^2 \cdot h^7 \le \kappa h^7. \end{aligned} \tag{7}$$

We also introduce Theorem 3.5.1 in page 67, Chapter 3.5 of Davis (1975) as follows.

Theorem 2. *Let $w_0, w_1, \cdots, w_r$ be $r+1$ distinct points in $[a,b]$, and $n_0, \cdots, n_r$ be $r+1$ integers ≥ 0. Let $N = n_0 + \cdots + n_r + r$. Suppose that $g(t)$ is a polynomial of degree N such that*

$$g^{(i)}(w_j) = f^{(i)}(w_j), \quad i = 0, \cdots, n_j, \; j = 0, \cdots, r.$$

Then there exists $\xi(t) \in [a,b]$ such that

$$f(t) - g(t) = \frac{f^{(N+1)}(\xi(t))}{(N+1)!} \prod_{i=0}^{r} (t - w_i)^{n_i+1}. \qquad \square$$

3.1. Constructing two rational cubics as reference polynomials

Let $B_{3,j}(t) = C_j^3 (b-t)^{3-j}(t-a)^j / h^3$ be a Bernstein basis function of degree 3 mapping to $[a,b]$, $j = 0,1,2,3$. We introduce two rational cubic polynomials

$$R_i(t) = \frac{\sum_{j=0}^{3} r_{i,j} B_{3,j}(t)}{B_{3,0}(t) + \sum_{j=1}^{3} r_{i,j+3} B_{3,j}(t)} = \frac{Y_i(t)}{\omega_i(t)}, \tag{8}$$

where $r_{i,j}$ are the unknowns, $i = 1,2$ and $j = 0,1,\cdots,6$. By adding the following constraints

$$R_1(t_j) = d_j \quad \text{and} \quad R_1'(t_l) = v_l, \quad j = 0,1,2,3, \; l = 0,1,2, \tag{9}$$

we obtain the values of the seven unknowns $r_{1,j}$ in $R_1(t)$, $j = 0,1,\cdots,6$. Similarly, we compute the values of $\{r_{2,j}\}_{j=0}^{6}$ in $R_2(t)$ from the constraints

$$R_2(t_j) = d_j \quad \text{and} \quad R_2'(t_l) = v_l, \quad j = 0,1,2,3, \; l = 1,2,3. \tag{10}$$

3.2. Computing the two rational cubic polynomials for bounding $f(t)$

For a general case, $R_1(t)$ can be utilized as the reference polynomial, and the error bounds between $f(t)$ and $R_1(t)$ can be estimated, i.e., there exist ε_1 and ε_2 such that $\varepsilon_1 \leq f(t) - R_1(t) \leq \varepsilon_2$, $\forall t \in [a,b]$. Thus, the two bounding polynomials can be set as

$$R_1(t) + \varepsilon_1 \leq f(t) \leq R_1(t) + \varepsilon_2, \; \forall t \in [a,b]. \tag{11}$$

The details for estimating ε_1 and ε_2 are as follows. Let $H_1(t) = \omega_1(t) f(t) - Y_1(t)$ be a polynomial of degree $n+3$. From Eq. (9), we have that

$$H_1(t_j) = 0, \qquad H_1'(t_l) = 0, \quad j = 0,1,2,3, \; l = 0,1,2. \tag{12}$$

Combing Eq. (12) with Theorem 2, there exists $\xi_1(t) \in [a,b]$ such that

$$H_1(t) = \frac{H_1^{(7)}(\xi_1(t))}{7!} g_1(t). \tag{13}$$

Following Eq. (13), $g_1(t)$ is a factor of $H_1(t)$, i.e., $F_1(t) = \frac{H_1(t)}{g_1(t)}$ is a polynomial of degree $n-4$. Both $F_1(t)$ and $H_1^{(7)}(t)$ are thus of degree $n-4$, which can be rewritten in Bernstein form such that

$$F_1(t) = \sum_{j=0}^{n-4} p_{1,j} B_{n-4,j}(t; a,b) \quad \text{and} \quad H_1^{(7)}(t) = \sum_{j=0}^{n-4} p_{2,j} B_{n-4,j}(t; a,b), \tag{14}$$

where $B_{n-4,j}(t; a,b)$ is a Bernstein basis function within $[a,b]$ of degree $n-4$. Suppose that

$$\lambda_{i,1} = \sup_{0 \leq j \leq n-4} \{p_{i,j}\}, \quad \lambda_{i,2} = \inf_{0 \leq j \leq n-4} \{p_{i,j}\} \quad \text{and} \quad \mu_1 = \inf_{a \leq t \leq b} \omega_1(t). \tag{15}$$

If $\mu_1 > 0$, we have that

$$\lambda_{1,1} \leq F_1(t) \leq \lambda_{1,2}, \quad \lambda_{2,1} \leq H_1^{(7)}(t) \leq \lambda_{2,2}, \quad \text{for all } t \in [a,b].$$

From Eq. (13), we obtain that

$$\mu_3 = \min\{-\frac{\kappa \lambda_{i,2}}{7!}, 0\} \leq H_1(t) \leq \max\{-\frac{\kappa \lambda_{i,1}}{7!}, 0\} = \mu_4, \quad \forall t \in [a,b], \; i = 1,2. \tag{16}$$

Table 1
Comparisons on the computational complexity and the convergence rate.

Methods	M_3	M_r	M_I
Time	$O(n^2)$	$O(n^2)$	$O(n)$
AO	4	7	7
CR	$4/k$	$7/k$	$7/k$

AO: Approximation order; **CR**: Convergence rate. k: The multiplicity of a root; **AO** $=$ **CR** $\times k$.

Thus, we have that

$$\varepsilon_1 = \frac{\mu_3 \cdot h^7}{\mu_1} \le \frac{H_1(t)}{\omega_1(t)} = f(t) - R_1(t) \le \frac{\mu_4 \cdot h^7}{\mu_1} = \varepsilon_2, \quad \forall t \in [a,b]. \tag{17}$$

Finally, we obtain two bounding rational cubic polynomials as

$$R_l(t) = R_1(t) + \varepsilon_1 \le f(t) \le R_1(t) + \varepsilon_2 = R_r(t). \tag{18}$$

Let $M = \max\{|\frac{2\mu_3}{\mu_1}|, |\frac{2\mu_4}{\mu_1}|\}$. From Eq. (18), we have that

$$|f(t) - R_i(t)| < Mh^7, \quad i = l, r, 1, \tag{19}$$

which means that the two bounding polynomials achieve an approximation order 7 to $f(t)$. From Theorem 1, it achieves a convergence rate of $7/k$ for a root of multiplicity k. Thus, we have Theorem 3 as follows.

Theorem 3. *The rational cubic clipping method by using rational cubic polynomials can achieve a convergence rate of $7/k$ for a root of multiplicity k.* □

Remark 1. Let $H_2(t) = \omega_2(t) f(t) - Y_2(t)$. If $f^{(4)}(t)$ doesn't change its sign for all $t \in [a,b]$, i.e., either $f^{(4)}(t) \le 0$ or $f^{(4)}(t) \ge 0$ for all $t \in [a,b]$, we have that $H_1^{(7)}(t) \cdot H_2^{(7)}(t) \ge 0, \forall t \in [a,b]$, which means that $R_1(t)$ and $R_2(t)$ directly bound $f(t)$ within $[a,b]$. The corresponding method is called the improved one in this paper.

Remark 2. If the denominator $\omega_1(t)$ of $R_1(t)$ has one or more zeros within $[a,b]$ such that $\mu_1 \le 0$ and it leads to a bad approximation effect. In such a case, the interval $[a,b]$ is divided into two halves for further clipping steps, which is similar to that used in the cubic clipping method in Liu et al. (2009).

4. Discussions and numerical examples

For the sake of convenience, let M_3, M_r and M_I be the cubic clipping method in Liu et al. (2009), the rational cubic clipping method in this paper and the improved method in Remark 1, respectively. If there are many roots of $f(t)$ within $[a,b]$, both of M_3 and M_r may converge very slowly. In this paper, at the beginning, if there are four or more zeros of the control polygon of $f(t)$ (in Bézier form), or if the first clipping step fails to clip the given interval, we utilize these zeros to divide $[a,b]$ into several subintervals. All of the examples are implemented by using the Maple software on PC with 4 G memory and 2.5 G CPU, the average computation time of a clipping step is tested by setting the number of digits after the decimal point as 16, the corresponding unit is millisecond.

4.1. Analyzing the computational complexity and the convergence rate

Firstly, we compare the computational complexity and the convergence rate of M_3 and M_r. If the two bounding polynomials are obtained, the remaining computation of a clipping step can be done within $O(n)$, so the computation of the two bounding polynomials would dominant the computational complexity of the methods under discussion. In M_3, it needs to estimate the bounds of a polynomial of degree n, whose computational complexity is $O(n^2)$; while in M_r, for a general case, it needs to estimate the bounds of a polynomial of degree $n-4$ in Eq. (14), whose computational complexity is $O((n-4)^2)$. So for a general case, both M_3 and M_r have a comparable computational complexity. In some cases that the conditions in Remark 1 are satisfied, M_I is applied to construct $R_1(t)$ and $R_2(t)$ which directly bound $f(t)$, the corresponding computational complexity is $O(n)$, which is much higher than that of M_3. The approximation order of the bounding polynomials to $f(t)$ in M_3, M_r and M_I are 4, 7 and 7, respectively. As shown in Theorem 3 and also Table 1, the convergence rates of both M_r and M_I are $7/k$ for a root of multiplicity k, which is much higher than that of $4/k$ of M_3. (See Table 1.)

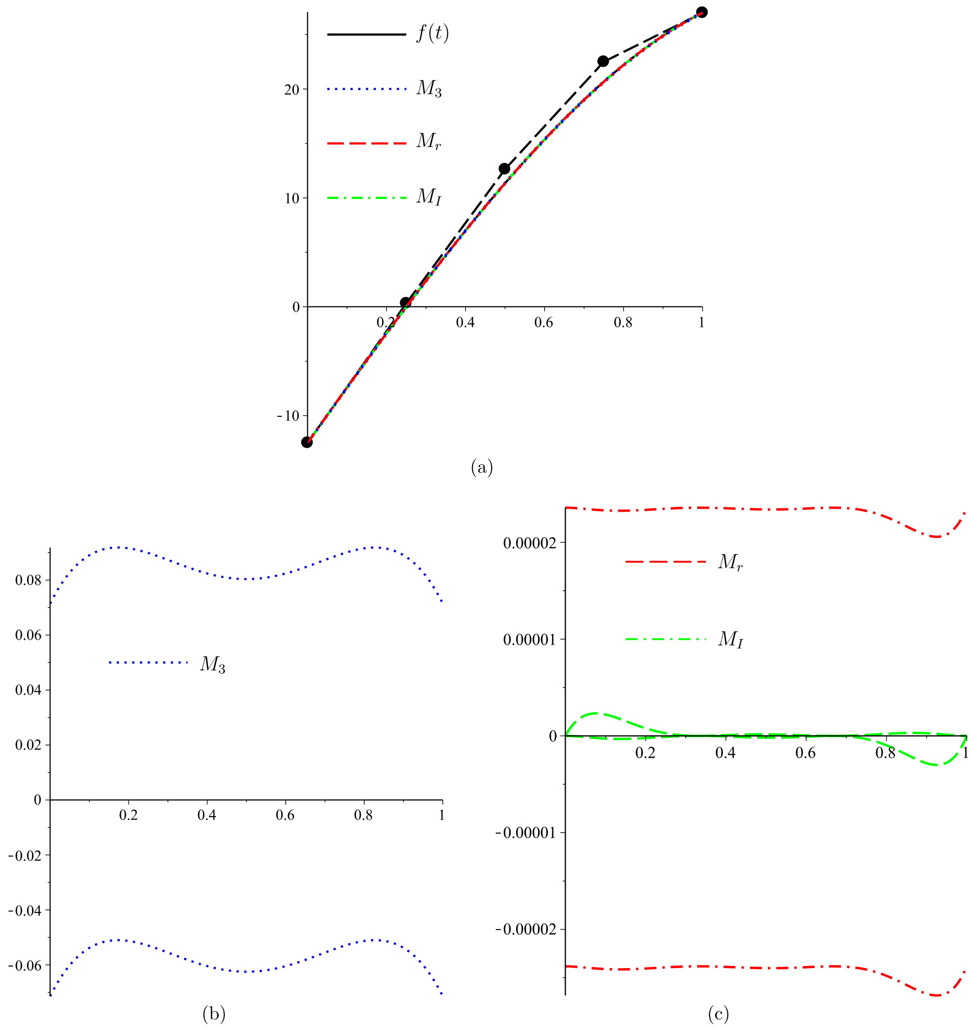

Fig. 1. Example 1: (a) plots of $f_1(t)$ and its control polygon in solid black line, and the bounding curves from M_3, M_r and M_I; (b–c) error plots between $f_2(t)$ and its bounding polynomials from M_3, M_r and M_I, respectively.

4.2. Comparing the approximation effect and the computational efficiency

By using the zeros of the control polygon of $f(t)$ to divide $[a, b]$ at the beginning, the improved method M_I mentioned in Remark 1 can be applied for computing the two bounding polynomials in most of cases. Firstly, we show two examples for illustrating the new method.

Example 1. Given $f_1(t) = (t - 1/4)(2 - t)(t + 5)^2$ having a single root in $t \in [0, 1]$ (see also Fig. 1), the results of the first clipping process are as follows. In this case, M_3 obtains the resulting subinterval of length 3.9e−2, and the maximum error between $f_1(t)$ and its bounding polynomials is 9.1e−2. From M_r, the resulting subinterval is of length 9.6e−7, while the corresponding maximum error is 2.6e−5. From M_I, the resulting subinterval is of length 2.0e−8, while the corresponding maximum error is 2.9e−6. See Fig. 1(b–c) for the corresponding error plots.

Example 2. Given $f_2(t) = (t - 0.2)(t - 0.25)(t - 0.75)(t + 5)^7(t - 6)^2$ having three roots in $t \in [0, 1]$ (see also Fig. 2(a)), the results of the first clipping process are as follows. In this case, M_3 obtains two resulting subintervals $[0.05, 0.41]$ and $[0.68, 0.75]$ with lengths 0.26 and 0.07, respectively, and the corresponding maximum error between $f_2(t)$ and its bounding polynomials is 1.02e+5. From M_r, it obtains three subintervals $[0.17, 0.21]$, $[0.23, 0.27]$ and $[0.7477, 0.7506]$ with lengths 0.04, 0.04 and 0.0029, for bounding the three single roots, respectively. The corresponding maximum error is 3.73e+3. From M_I, it obtains three subintervals $[0.1986, 0.2004]$, $[0.2497, 0.2504]$ and $[0.74997, 0.75008]$ with lengths 0.0018, 7.5e−4 and 1.1e−4 for bounding the three single roots, respectively. The corresponding maximum error is 1.5e+3. See Fig. 2(b–c) for the corresponding error plots.

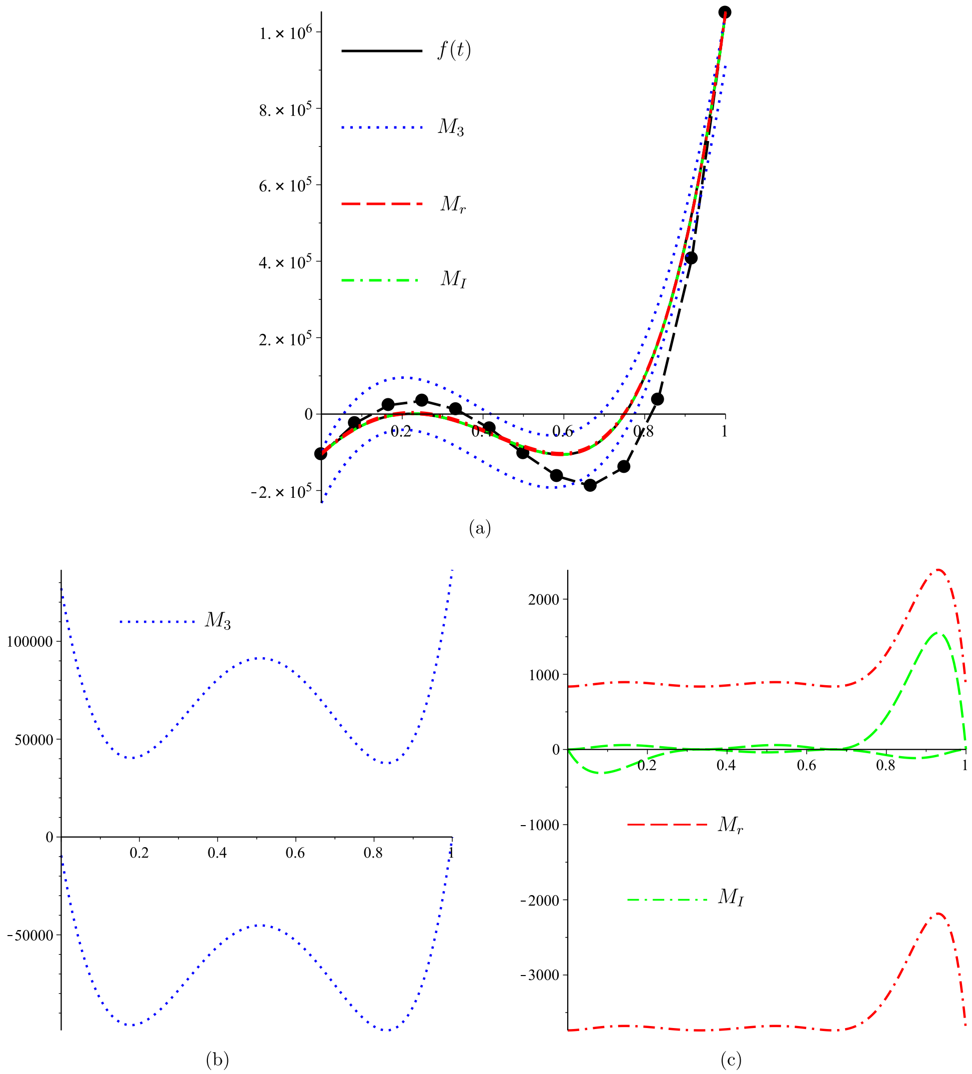

Fig. 2. Example 2: (a) plots of $f_2(t)$ and its control polygon in solid black line, and the bounding curves from M_3, M_r and M_I; and (b–c) error plots between $f_2(t)$ and its bounding polynomials from M_3, M_r and M_I, respectively.

Table 2
Comparisons on errors and convergence rates (Examples 1–2).

Exam	**M**	1	2	3	4	**CR**	**Error**
Fig. 1	M_3	3.9e−2	2.1e−13	6.5e−54	5.4e−216	4	9.1e−2
	M_r	9.6e−7	8.1e−49	1.0e−343	/	7	2.6e−5
	M_I	2.0e−8	2.9e−62	4.0e−439	/	7	2.9e−6
Fig. 2	M_3	7.0e−2	1.9e−6	1.1e−24	1.4e−97	4	1.0e+5
	M_r	2.9e−3	1.6e−21	2.9e−149	/	7	3.7e+3
	M_I	1.1e−4	3.0e−32	4.1e−225	/	7	1.5e+3

For the above two examples, more details are listed in Table 2, where "**M**" means method and "**Error**" means the maximal approximation error between $f_1(t)$ and its bounding polynomials at the first clipping step. As shown in both Figs. 1–2 and Table 2, both M_r and M_I can achieve a much better approximation effect than that of M_3, and the corresponding subintervals are of much smaller lengths than that of M_3. The results of more clipping steps are shown in Table 2. It shows that M_3 and M_r achieve convergence rates of 4 and 7, respectively.

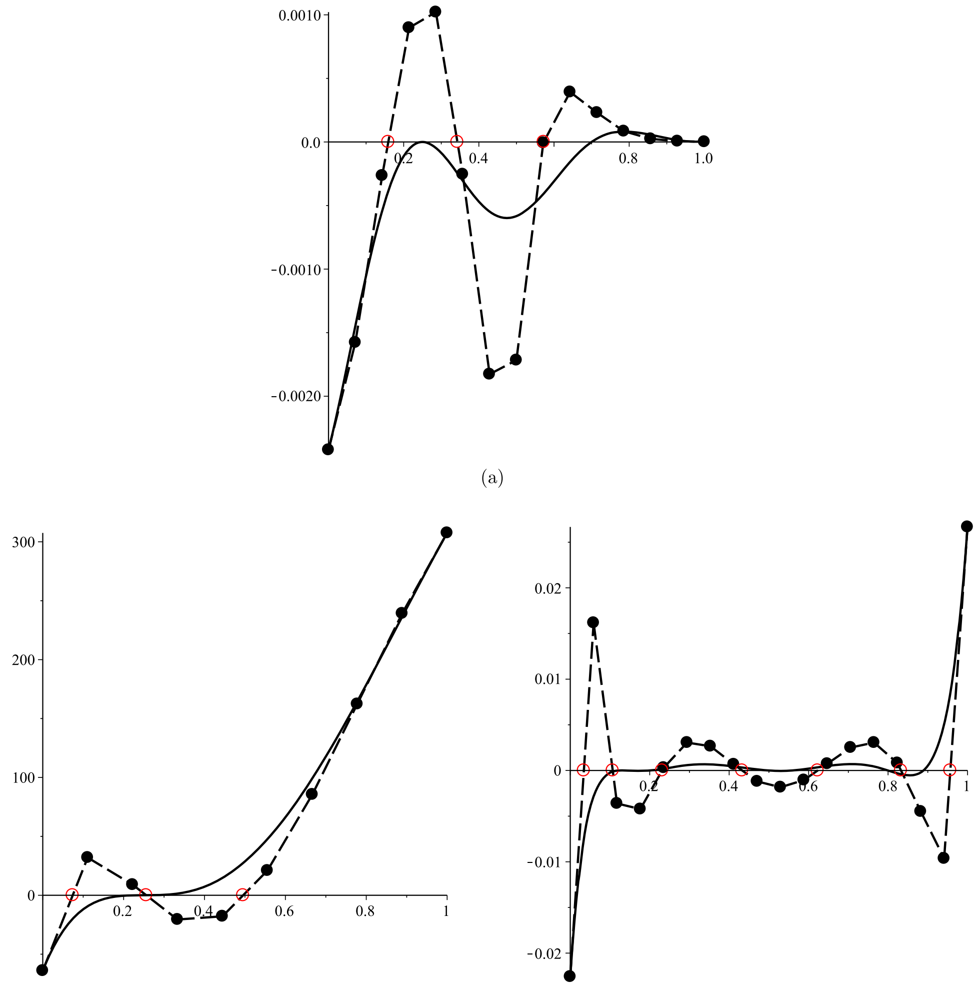

Fig. 3. Examples 3–5: Plots of (a) $f_3(t)$ of degree 14 with a double root and a single root; (b) $f_4(t)$ of degree 9 with a triple root; and (c) $f_5(t)$ of degree 17 with seven single roots. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Examples 3–5. We have tested M_r by using the following three examples

$$f_3(t) = (t - 0.250001)^2(t + 0.5)^5(t - 0.7)(t - 1.1)^6,$$
$$f_4(t) = (t - 0.25)^3(4 - t)^6,$$
$$f_5(t) = (t - \frac{1}{8})(t - \frac{1}{7})(t - \frac{1}{5})(t - \frac{1}{2})(t - \frac{5}{9})(t - \frac{4}{5})(t - \frac{8}{9})(t^2 + 2)^2(t^2 - 2t + 2)^3,$$

where $t \in [0, 1]$, which is also shown in Fig. 3. In Fig. 3, the lines in dashed black and the points in solid black are the control polygon and its control points, while the points in red circle are the zeros of its control polygon. The comparison results are shown in Table 3. In these cases, the number of the digits after decimal point is set as 1000 or more for testing purpose for achieving a high precision shown in Table 3. In Table 3, k means the multiplicity of a root and **M** means "method". Table 3 shows the results of a double root of $f_3(t)$, a triple root of $f_4(t)$ and a single root of $f_5(t)$. Similarly, note that there are seven zeros of the control polygon of $f_5(t)$, we directly use the seven zeros to divide $[0, 1]$ into eight subintervals before applying both M_r and M_3. It shows that both M_r and M_I have a much higher convergence rate $7/k$ than that of $4/k$ of M_3, where k denotes the multiplicity of the root.

The average computation time of a clipping step among M_3, M_r and M_I, i.e., T_3, T_r and T_I, are listed in Table 4, where the unit is millisecond. It shows that T_3 and T_r are comparable, while T_I is much less than that of T_3, or in other words, M_I is much faster and is about 7–10 times faster than that M_3. The convergence rates of M_r and M_I are 7/4 times higher than that of M_3. Both the computational efficiency and convergence rate of a clipping step in M_I are thus much better than that of M_3.

Table 3
Comparisons on errors and convergence rates (Examples 3–5).

Exam	k	**M**	1	2	3	4	5	**CR**
Fig. 3(a)	2	M_3	1.8e−1	1.9e−2	2.0e−4	2.1e−8	2.5e−16	2
		M_r	1.8e−1	7.1e−3	7.5e−8	2.7e−25	2.5e−86	7/2
		M_I	1.8e−1	5.8e−4	1.1e−12	3.6e−43	6.7e−150	7/2
Fig. 3(b)	3	M_3	3.1e−1	7.5e−2	1.2e−2	1.4e−3	8.0e−5	4/3
		M_r	4.5e−3	1.2e−6	6.4e−15	2.8e−34	2.0e−79	7/3
		M_I	1.9e−3	3.1e−8	1.9e−19	1.7e−45	3.1e−106	7/3
Fig. 3(c)	1	M_3	1.2e−1	1.7e−2	1.9e−6	3.0e−22	2.0e−85	4
		M_r	1.2e−1	7.4e−3	2.7e−11	2.5e−70	1.9e−483	7
		M_I	1.2e−1	3.0e−5	1.5e−31	1.4e−215	/	7

Table 4
Comparisons on average computation time (ms) of a clipping step.

Examples	Fig. 1	Fig. 2	Fig. 3(a)	Fig. 3(b)	Fig. 3(c)
T_3	34.18	46.80	43.58	38.06	63.46
T_r	35.01	44.80	42.14	37.34	61.12
T_I	4.72	6.630	5.22	4.61	6.53
T_I/T_3	13.8%	14.2%	11.9%	12.1%	10.2%

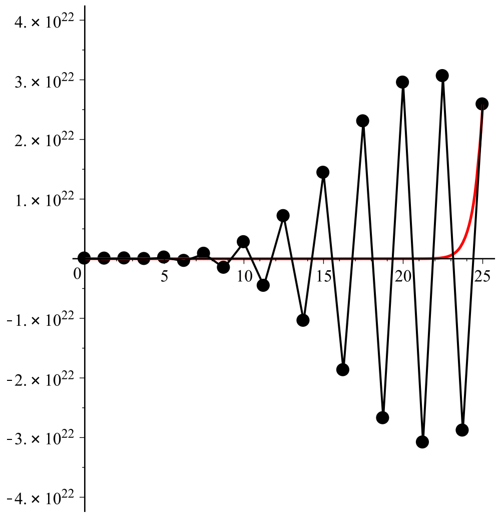

Fig. 4. Plot of the Wilkinson polynomial.

4.3. Numerical robustness

The proposed M_r method also uses the Cardano formula to solve cubic polynomial equations, which is the same to M_3. The stability of Bernstein–Bézier representation is also applicable to M_r, which is the same as that of the M_3 method.

Example 6. We have tested both M_3 and M_r to compute the roots of the Wilkinson polynomial

$$W(x) = \prod_{i=1}^{20}(x-i),$$

within [0, 25], which has twenty zeros i, $i = 1, 2, \cdots 20$, see also Fig. 4. At the beginning, we compute the zeros of the corresponding control polygon, i.e., {0.27, 1.55, 2.83, 4.11, 5.38, 6.65, 7.92, 9.19, 10.46, 11.73, 13.007, 14.27, 15.54, 16.81, 18.07, 19.34, 20.61, 21.87, 23.14, 24.40}. Thus, the given interval [0, 25] is divided into twenty-two sub-intervals by using the above twenty-one zeros. There are sixteen sub-intervals containing one or two roots of $f(t)$. We select the three of them [0.27, 1.55], [2.83, 4.11] and [16.81, 18.07] to illustrate more details, which contain one, two and two roots of $f(t)$, respectively. The bounding polynomials at the first clipping step from M_3, M_r and M_I are shown in Fig. 5. As shown in Fig. 5, both M_r and M_I work for all sub-intervals, which achieve much better approximation effect than those of M_3. The other thirteen sub-intervals are similar. More details of the results are list in Table 5. It shows that the corresponding convergence rates of M_3, M_r and M_I for a single root is 4, 7 and 7, respectively.

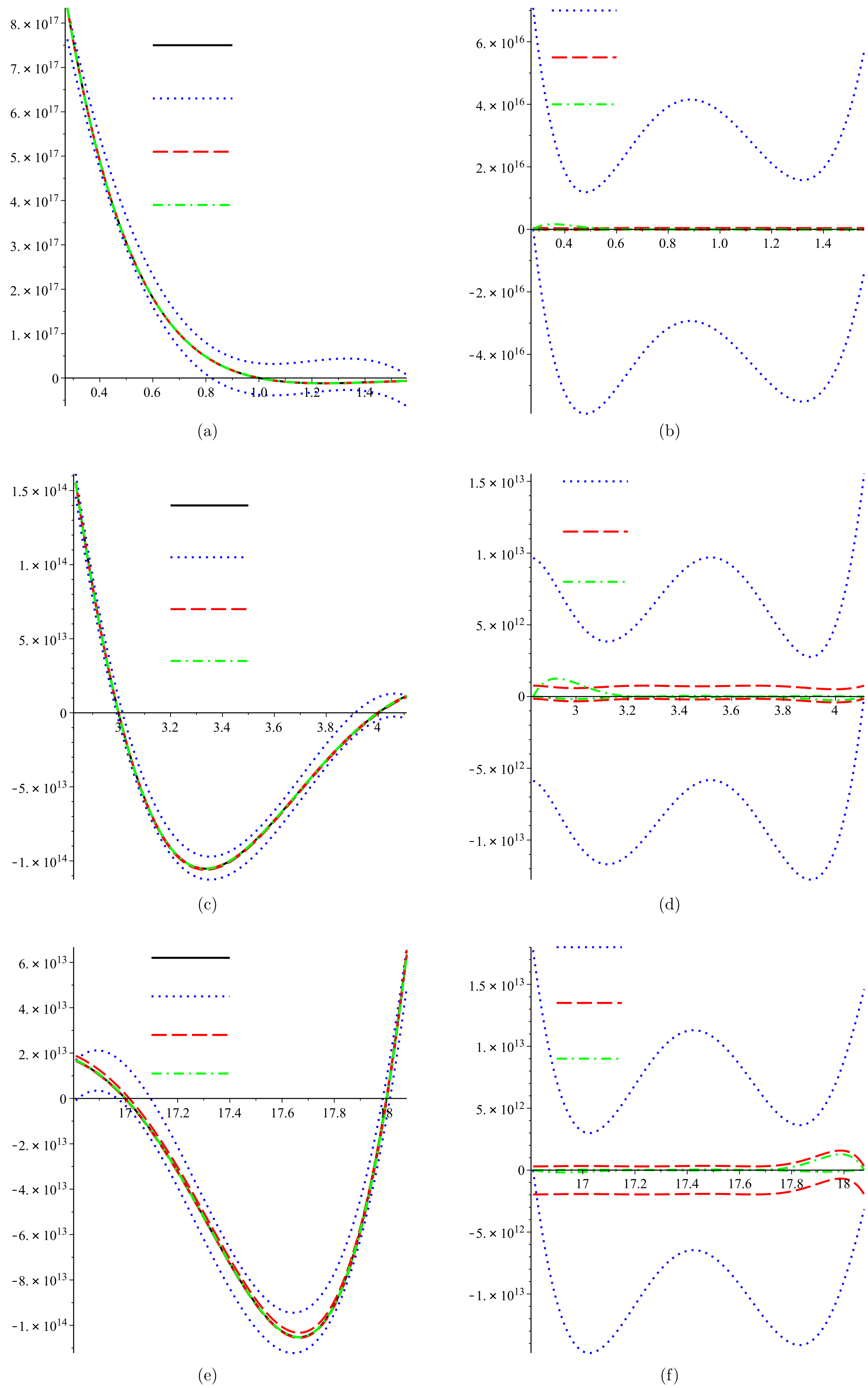

Fig. 5. Example 6: Plots of the given Wilkinson polynomial with respective bounding polynomials from M_3, M_r and M_I (left column); and corresponding error plots between the given Wilkinson polynomial and its bounding polynomials (right column), for sub-intervals (a–b) [0.27, 1.55]; (c–d) [2.83, 4.11]; and (e−f) [16.81, 18.07].

Table 5
Comparisons on errors and convergence rates (Example 6, $W(x)$).

Fig. 5	k	**M**	1	2	3	4	5	**CR**
(a, b)	1	M_3	6.4e−1	1.1e−2	2.3e−9	3.4e−36	1.7e−143	4
		M_r	4.4e−3	1.9e−20	4.7e−142	/	/	7
		M_I	1.6e−4	1.1e−31	6.9e−222	/	/	7
(c, d)	1	M_3	2.2e−2	1.0e−8	3.7e−34	7.5e−136	1.1e−542	4
		M_r	7.3e−3	3.9e−18	4.8e−125	1.9e−873	/	7
		M_I	1.6e−3	8.6e−22	1.4e−149	/	/	7
(e, f)	1	M_3	2.8e−1	2.2e−4	9.5e−17	3.2e−66	4.4e−264	4
		M_r	1.8e−2	1.8e−15	2.4e−106	1.6e−742	/	7
		M_I	1.4e−3	1.0e−23	1.1e−164	/	/	7

5. Conclusions

This paper presents a rational cubic clipping method (denoted as M_r) for finding the roots of a polynomial $f(t)$ within an interval, which can achieve a convergence rate of 7 for a single root by using rational cubic polynomials. Different from previous clipping methods in Bartoň and Jüttler (2007), Liu et al. (2009) for computing two bounding polynomials in $O(n^2)$ time, M_r directly constructs two rational cubic polynomials, which can be used to bound $f(t)$ in many cases and leads to a computational complexity of $O(n)$. Numerical examples also show that M_r can achieve a much higher convergence rate, much better approximation effect and much higher computation efficiency than previous clipping methods in Bartoň and Jüttler (2007), Liu et al. (2009).

As for future work, it should also be feasible to achieve a higher computation efficiency by using two bounding B-spline curves, or to achieve a higher convergence rate by combining with reparameterization techniques. Another possible future work is to extend M_r from the curve case to surface cases, which can be used for the root-finding problem of an equation system consisting of two bivariate polynomials.

Acknowledgements

This research was partially supported by the Research Grants Council of Hong Kong Special Administrative Region, China (SRG #7004245), Defense Industrial Technology Development Program and the National Science Foundation of China (61370218, 61379072).

References

Bartoň, M., Jüttler, B., 2007. Computing roots of polynomials by quadratic clipping. Comput. Aided Geom. Des. 24, 125–141.
Chen, X.D., Yong, J.H., Wang, G.Z., Paul, J.C., Xu, G., 2008. Computing the minimum distance between a point and a NURBS curve. Comput. Aided Des. 40 (10–11), 1051–1054.
Choi, Y.K., Wang, W.P., Liu, Y., Kim, M.S., 2006. Continuous collision detection for two moving elliptic disks. IEEE Trans. Robot. 22 (2), 213–224.
Davis, P.J., 1975. Interpolation and Approximation. Dover Publications, New York.
Efremov, A., Havran, V., Seidel, H.-P., 2005. Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces. In: The 21st Spring Conference on Computer Graphics. ACM Press, New York, pp. 127–135.
Elber, G., Kim, M.-S., 2001. Geometric constraint solver using multivariate rational spline functions. In: The Sixth ACM/IEEE Symposium on Solid Modeling and Applications. Ann Arbor, Michigan, pp. 1–10.
Farouki, R.T., Rajan, V.T., Rajan, V.T., 1987. On the numerical condition of polynomials in Bernstein form. Comput. Aided Geom. Des. 4 (87), 191–216.
Farouki, R.T., Goodman, T.N.T., 1996. On the optimal stability of the Bernstein basis. Math. Comput. 126 (65), 1553–1566.
Isaacson, E., Keller, H.B., 1966. Analysis of Numerical Methods. Wiley, New York.
Jüttler, B., 1998. The dual basis functions of the Bernstein polynomials. Adv. Comput. Math. 8, 345–352.
Lin, M., Gottschalk, S., 1998. Collision detection between geometric models: a survey. In: Proceedings of IMA Conference on Mathematics of Surfaces. Birmingham, UK, pp. 37–56.
Liu, L.G., Zhang, L., Lin, B.B., Wang, G.J., 2009. Fast approach for computing roots of polynomials using cubic clipping. Comput. Aided Geom. Des. 26, 547–559.
McNamee, J.M., 1993–2002. Bibliographies on roots of polynomials. J. Comput. Appl. Math. 47, 391–394; J. Comput. Appl. Math. 110, 305–306; J. Comput. Appl. Math. 142, 433–434, available at http://www1.elsevier.com/homepage/sac/cam/mcnamee/.
Morken, K., Reimers, M., 2007. An unconditionally convergent method for computing zeros of splines and polynomials. Math. Comput. 76 (258), 845–865.
Mourrain, B., Pavone, J.-P., 2005. Subdivision methods for solving polynomial equations. Technical Report, no. 5658. INRIA Sophia Antipolis. http://www.inria.fr/rrrt/rr-5658.html.
Nishita, T., Sederberg, T.W., Kakimoto, M., 1990. Ray tracing trimmed rational surface patches. In: Proceedings of Siggraph. ACM, pp. 337–345.
Patrikalakis, N.M., Maekawa, T., 2002. Intersection problems. In: Farin, G., Hoschek, J., Kim, M.-S. (Eds.), Handbook of Computer Aided Geometric Design. Elsevier, pp. 623–649.
Reuter, M., Mikkelsen, T., Sherbrooke, E., Maekawa, T., Patrikalakis, N., 2007. Solving nonlinear polynomial systems in the barycentric Bernstein basis. Vis. Comput. 24 (3), 187–200.
Rouillier, F., Zimmermann, P., 2004. Efficient isolation of polynomial's real roots. J. Comput. Appl. Math. 162 (1), 33–50.
Sederberg, T.W., Nishita, T., 1990. Curve intersection using Bézier clipping. Comput. Aided Des. 22 (9), 538–549.
Schulz, C., 2009. Bézier clipping is quadratically convergent. Comput. Aided Geom. Des. 26 (1), 61–74.