Communications in Information and Systems
Volume 17, Number 2, 115–131, 2017

Efficient rational quadratic clipping method for computing roots of a polynomial

Xiao-Diao Chen, Longquan Wang, Ligeng Chen, Yigang Wang, and Weiyin Ma

The root-finding problem is one of key issues for visualizing implicit curves and surfaces, and has wide applications in computer-aided design, computer graphics and geometric computing for image and video. This paper presents a rational quadratic clipping method for computing a simple root of a polynomial $f(t)$ of degree n within an interval, which preserves the optimal computation stability of the Bernstein-Bézier representation. Different from previous clipping methods based on interpolation, it optimizes the selection of the inner point, which can achieve the convergence rate 12 by using rational quadratic polynomials. Difference from previous clipping methods by computing bounding polynomials, it provides a simple method of linear complexity to directly bound the root; at the same time, it needs to compute the roots of quadratic polynomials and avoids solving cubic equations, and leads to a higher computational efficiency. In principle, it also works well for a non-polynomial case. Numerical examples show higher convergence rate and better computation efficiency of the new method.

1. Introduction

In various geometric problems, such as curve/surface intersection [3, 13, 22], point projection [6], bisectors/medial axes computation [14], collision detection [10], and geometric optimization for image and video, it is often needed to solve systems of non-linear equations [23]. The root-finding problem is also one of the key issues in computer-aided design, computer graphics and geometric computing for image and video. There are many references discussing how to solve a polynomial equation or equation system [1, 12, 20, 24] (see also the references therein).

By using the optimal computation stability of the Bernstein-Bézier representation with respect to perturbations of the coefficients [15], several clipping methods are developed to find the roots of polynomials [2, 7–9, 19]. Given a polynomial $f(t)$ of degree n, and an interval $[a, b]$, the basic idea of the above clipping methods is to clip the parts of the interval containing no root of $f(t)$ by using the roots of its bounding polynomials, and iteratively do the clipping processes for the remaining subintervals (if any), until either there remains no subinterval or the length of the subinterval is within a given tolerance. The bounding polynomials of $f(t)$ within $[a, b]$ can be computed either by using the degree reduction technique [2, 19], which needs to turn $f(t)$ into a Bézier form corresponding to the given interval with the time cost $O(n^2)$; or by using interpolation technique [7, 8], the error bounds between $f(t)$ and its interpolation polynomials can be estimated within $O(n^2)$ time. Suppose that there is a unique root of $f(t)$ within $[a, b]$, where $h = b - a$ tends to be very small, and after a clipping step, one obtains a subinterval $[a_1, b_1]$ whose length is $O(h^d)$; the convergence rate is defined as d. For a simple root case, the convergence rate is the same as the approximation order. The approximation order between $f(t)$ and its bounding polynomials can be improved by using planar interpolation technique or rational polynomial interpolation technique, e.g., 4 by using planar quadratics [7], 7 by using rational cubics [8]. In [9], a rational cubic clipping method whose bounding polynomials are constructed within $O(n)$ time is presented, while its convergence rate is 5 for a single root case, which is lower than that 7 of [8].

In principle, one only needs to bound $f(t)$ within local subintervals containing the roots instead of the whole given interval, the roots of the bounding polynomials can also be used for clipping; furthermore, one can directly bound the roots of $f(t)$, which can lead to a much higher approximation order and computational efficiency. Based on this observation, this paper presents an efficient rational quadratic clipping method of convergence rate 12, which directly bounds the roots instead of bounding $f(t)$. Note that the roots of $f(t)$ can be separated by using previous methods such as the ones in [8, 9, 19]. In this paper, we assume that there is one simple root within the given interval. It mainly has two steps. Firstly, a rational quadratic polynomial $q(t)$ interpolates three point of $f(t)$, i.e., two end points and one inner point, and two of their three derivatives. In principle, different selections of the inner point lead to the same approximation order 5 between $f(t)$ and $q(t)$ in the whole interval, but the approximation order between the roots of $f(t)$ and $q(t)$ can reach 12 by optimizing the selection of the inner point. Secondly, based on the roots of $q(t)$, it presents a simple method of linear complexity to directly bound the roots of $f(t)$ without computing the bounding polynomials. Numerical examples show that the new method can achieve much higher convergence rate 12 and much higher computational efficiency.

The remainder of this paper is organized as follows. Section 2 explains the rational quadratic clipping method and analyzes the corresponding convergence rate. Section 3 provides more numerical examples and related discussions. Conclusions are drawn at the end of this paper.

2. The rational quadratic clipping method

In principle, the roots of a given polynomial $f(t)$ within an interval $[a, b]$ can be isolated by using previous methods such as the ones in [9, 19, 21]. In this paper, we assume that there is one simple root $t^\star \in [a, b]$ of $f(t)$, and we have that $f(a) \cdot f(b) < 0$. The basic idea of the clipping method is to find a sequence of subintervals $[a_i, b_i]$, $i = 0, 1, \ldots,$ by iteratively executing the clipping processes, such that $[a_i, b_i]$ contains $t^\star$, and $b_{i+1} - a_{i+1}$ tends to be $O((b_i - a_i)^d)$, where d denotes the convergence rate.

For the sake of convenience, let $h = b - a$, and we introduce Theorem 3.5.1 in Page 67, Chapter 3.5 of [11] as follows.

Theorem 1. Let $w_0, w_1, \ldots, w_r$ be $r+1$ distinct points in $[a, b]$, and $n_0, \ldots,$ n_r be $r+1$ positive integers. Let $N = n_0 + n_1 + \cdots + n_r$. Suppose that $g(t)$ is a polynomial of degree $N - 1$ such that

$$g^{(i)}(w_j) = f^{(i)}(w_j), \quad i = 0, 1, \ldots, n_j - 1, \quad j = 0, \ldots, r.$$

Then there exists $\xi_1(t) \in [a, b]$ such that

$$|f(t) - g(t)| = \left| \frac{f^{(N)}(\xi_1(t))}{N!} \prod_{i=0}^{r} (t - w_i)^{n_i} \right| = O(\prod_{i=0}^{r} |(t - w_i)^{n_i}|).$$

2.1. One clipping process

One clipping process of the rational quadratic clipping method (RQCM) mainly has four steps: (1) Optimize the value of t_1, which is a root of $q(t)$ determined by Eq. (1); (2) Compute the value of t_2, which is a root of $r(t)$ determined by Eq. (4); (3) Compute the value of t_3, which is a root of $p(t)$ determined by Eq. (8); and (4) Bound the root $t^\star$ by using either t_2 and t_3, or t_3 and t_4, where $t_4 = 2t_3 - t_2$.

Let $h = b - a$, $l(t) = \frac{f(a)(b-t)+f(b)(t-a)}{h}$, $t_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$. It can be verified that $l(t_0) = 0$, $l(a) = f(a)$ and $l(b) = f(b)$. Combining with Theorem 1, we have that $|f(t) - l(t)| = O(|(t - a)(t - b)|)$, and $|t^\star - t_0| = O(h^2)$. Firstly, let $q(t) = q_0 + q_1 t + q_2 t^2$, where

$$q_0 = \frac{b \cdot t_0 \cdot f(a)}{(a-b)(a-t_0)} + \frac{a \cdot t_0 \cdot f(b)}{(b-a)(-t_0+b)} + \frac{a \cdot b \cdot f(t_0)}{(-t_0+b)(a-t_0)},$$
$$q_1 = \frac{-(b+t_0) \cdot f(a)}{(a-b)(a-t_0)} + \frac{(a+t_0) \cdot f(b)}{(a-b)(-t_0+b)} + \frac{-(a+b) \cdot f(t_0)}{(-t_0+b)(a-t_0)},$$
$$q_2 = \frac{f(a)}{(a-b)*(a-t_0)} + \frac{-f(b)}{(a-b)(-t_0+b)} + \frac{f(t_0)}{(-t_0+b)(a-t_0)},$$

which satisfies

$$f(a) = q(a), \quad f(t_0) = q(t_0), \quad f(b) = q(b). \tag{1}$$

From $q(a) \cdot q(b) = f(a) \cdot f(b) < 0$, there is a root $t_1 \in [a, b]$ of $q(t)$.

Theorem 2. We claim that $|t_1 - t^\star| = O(h^4)$.

Proof. Combining Eq. (1) with Theorem 1, there exists $\xi_2(t)$ such that

$$|f(t) - q(t)| = \left| \frac{f^{(3)}(\xi_2(t))}{3!}(t-a)(t-t_0)(t-b) \right| = O(h^4). \tag{2}$$

From Eq. (2),

$$|f(t^\star) - f(t_1)| = |f(t_1)| = |f(t_1) - q(t_1)| = O(h^4). \tag{3}$$

On the other hand, note that there exists α such that $|f(t^\star) - f(t_1)| = |f'(\alpha)(t^\star - t_1)|$, combining with Eq. (3), we have that $|t_1 - t^\star| = O(h^4)$. □

Secondly, we find a rational polynomial $r(t) = \frac{r_0 + r_1 t + r_2 t^2}{1 + r_3 t + r_4 t^2} \triangleq \frac{X(t)}{Y(t)}$ interpolating $f(t)$ such that

$$f(a) = r(a), \; f(t_1) = r(t_1), \; f(b) = r(b), \; f(t_0) = r(t_0), \; f'(t_1) = r'(t_1). \tag{4}$$

Let $E(t) = f(t) \cdot Y(t) - X(t)$. From Eq. (4), it can be verified that $E(a) = E(b) = E(t_0) = E(t_1) = E'(t_1) = 0$. Combining with Theorem 1, we have that

$$|E(t)| = |Y(t)\cdot f(t) - X(t)| \\ = O((t-a)(t-t_0)(t-t_1)^2(t-b)) = O(h^{12}). \tag{5}$$

Suppose that $m_1 = \min\limits_{t\in[a,b]} Y(t) > 0$. Combining with Eq. (5), we have that

$$|f(t) - r(t)| = \left|\frac{E(t)}{Y(t)}\right| \leq \left|\frac{E(t)}{m_1}\right| \\ = O(|(t-a)(t-t_0)(t-t_1)^2(t-b)|) = O(h^{12}). \tag{6}$$

Note that $r(a)\cdot r(b) = f(a)\cdot f(b) < 0$, suppose that $1 + r_3t + r_4t^2 \neq 0, \forall t \in [a,b]$, we can compute the root $t_2 \in [a,b]$ of $r(t)$. Similarly, combining Eq. (6), we have that

$$|t_2 - t^\star| = O(|f(t_2) - f(t^\star)|) = O(|f(t_2) - r(t_2)|) = O(h^{12}). \tag{7}$$

Thirdly, let $p(t) = p_0 + p_1t + p_2t^2$, where

$$p_0 = \frac{b\cdot t_2\cdot f(a)}{(a-b)(a-t_2)} + \frac{a\cdot t_2\cdot f(b)}{(b-a)(-t_2+b)} + \frac{a\cdot b\cdot f(t_2)}{(-t_2+b)(a-t_2)},$$
$$p_1 = \frac{-(b+t_2)\cdot f(a)}{(a-b)(a-t_2)} + \frac{(a+t_2)\cdot f(b)}{(a-b)(-t_2+b)} + \frac{-(a+b)\cdot f(t_2)}{(-t_2+b)(a-t_2)},$$
$$p_2 = \frac{f(a)}{(a-b)*(a-t_2)} + \frac{-f(b)}{(a-b)(-t_2+b)} + \frac{f(t_2)}{(-t_2+b)(a-t_2)},$$

which satisfies

$$f(a) = p(a), \quad f(t_2) = p(t_2), \quad f(b) = p(b). \tag{8}$$

Similarly, there exists a root $t_3 \in [a,b]$ of $p(t)$ such that $|t_3 - t^\star| = O(|(t-a)(t-b)(t-t_2)|) = O(h^{14})$.

Finally, in the fourth step, note that $|t_2 - t^\star| = O(h^{12})$ and $|t_3 - t^\star| = O(h^{14})$, we assume that $|t_2 - t^\star| > 2|t_3 - t^\star|$ and directly bound $t^\star$ based on t_2 and t_3 as follows. Without loss of generality, suppose that $t_2 < t^\star$, and $t^\star - t_2 = \beta|t_3 - t^\star|$, where $\beta > 2$. Let $t_4 = 2t_3 - t_2$. Note that $t_4 - t^\star = 2t_3 - t_2 - t^\star = 2(t_3 - t^\star) - (t_2 - t^\star) = \beta|t_3 - t^\star| + 2(t_3 - t^\star) \geq (\beta - 2)|t_3 - t^\star| \geq 0$. So we have that t_2 and t_4 bound $t^\star$, and $|t_4 - t_2| = |2(t_3 - t^\star) - (t_2 - t^\star)| = O(h^{12})$.

Remark 1. If $1 + r_3t + r_4t^2$ has one or more roots within $[a,b]$, one computes $R(t) = r_0 + r_1t + r_2t^2$ such that $R(t_1) = f(t_1)$, $R'(t_1) = f(t_1)$ and

$R(\tau) = f(\tau)$, where $\tau \in \{a, b\}$ such that $f(\tau) \cdot f(t_1) < 0$; and we have that $|t_2 - t^\star| = O(h^9)$ instead.

Remark 2. Note that four of the five equations in Eq. (4) are linear in the unknown r_i, while one is quadratic in r_i. There are two solutions of $r(t)$ from Eq. (4) which can be explicitly expressed [4], and the one where $r_4 = \frac{-((t_0-b)f(a)+f(t_0)(b-a)+(a-t_0)f(b))}{t_1((a-b)t_0 f(t_0)+b(t_0-a)f(b)+(b-t_0)af(a))}$ is removed.

Remark 3. For a multiple root case, i.e., $t^\star$ is a multiple root of $f(t)$, if $f'(t)$ has a unique root $t^\star$ within the given interval $[a, b]$, one can compute $F(t) = f(t)/f'(t)$ instead, which has a simple root $t^\star$.

2.2. Illustrations of one clipping process of the RQCM method

We show two examples to illustrate one clipping process of the RQCM method.

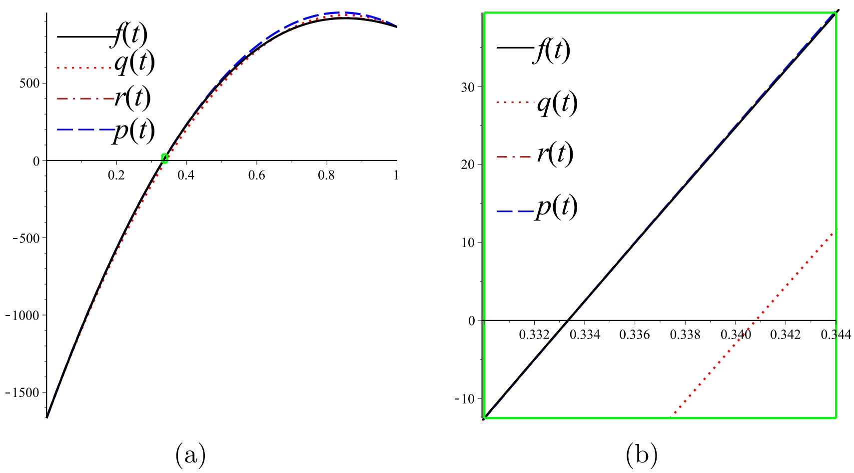
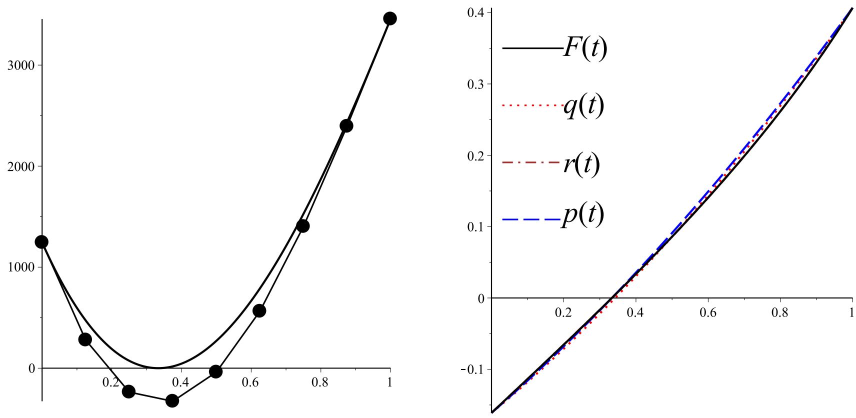


Figure 1: Example 1: (a) Plots of $f_1(t)$ and its corresponding $q(t)$, $r(t)$ and $p(t)$; and (b) Magnified plots of framed area.

Example 1. Let $f_1(t) = (t - 1/3)(2 - t)^3(t + 5)^4$, which has a simple root $t^\star = 1/3$ within $[0, 1]$, as shown in Fig. 1(a). Firstly, from Eq. (1), we obtain $q(t) = -1666.6667 + 6110.3926t - 3579.7259t^2$ and its root $t_1 = t^\star + 0.0074$ as well. Secondly, from Eq. (4), we compute $r(t) = (-1666.6667 + 6097.6523t - 3292.9691t^2)/(1 + 0.02909t + 0.2881t^2)$ and its root $t_2 = t^\star - 3.4 \cdot 10^{-7}$ as well. Thirdly, from Eq. (8), we have $p(t) = -1666.6667 +$

$6234.6666t - 3703.9999t^2$ and its root $t_3 = t^\star - 1.7 \cdot 10^{-9}$. Finally, we compute $t_4 = 2t_3 - t_2 = t^\star + 1.9 \cdot 10^{-8}$, and the resulting subinterval is $[t_3, t_4]$ containing $t^\star = 1/3$, whose length is $6.7 \cdot 10^{-7}$.



(a) (b)

Figure 2: Example 2: (a) Plots of $f_2(t)$ and its control polygon; and (b) Plots of $F(t)$, $q(t)$, $r(t)$ and $p(t)$.

Example 2. Let $f_2(t) = (t - 1/3)^2(4 - t)^3(t + 7)(t + 5)^2$, which has a double root $t^\star = 1/3$ within $[0, 1]$, as shown in Fig. 2(a). We compute the root of $F(t) = f(t)/f'(t)$ instead, which has the simple root $t^\star$ within $[0, 1]$. Similarly, from Eq. (1), Eq. (4) and Eq. (8), we obtain $q(t) = -0.1611 + 0.4466t + 0.1212t^2$, $r(t) = (-0.1611 + 0.5073t - 0.0721t^2)/(1 - 0.2035t - 0.1226t^2)$ and $p(t) = -0.1611 + 0.4410t + 0.1268t^2$, and their roots $t_1 = t^\star - 0.0023$, $t_2 = t^\star - 3.9 \cdot 10^{-10}$ and $t_3 = t^\star - 1.9 \cdot 10^{-11}$, respectively. Finally, we compute $t_4 = 2t_3 - t_2 = t^\star + 7.8 \cdot 10^{-10}$. And the resulting subinterval is $[t_2, t_4]$, which contains $t^\star = 1/3$, and the corresponding length is $7.4 \cdot 10^{-10}$.

Remark 4. In principle, the new method can work for solving a simple root of $f(t)$ within an interval $[a, b]$, and it doesn't matter whether or not $f(t)$ is a polynomial. However, $F(t) = f(t)/f'(t)$ may introduce trouble in the cases that $f'(t)$ has one or more roots within $[a, b]$ which are not roots of $f(t)$, and the method may fail. For computing a multiple root of a polynomial function $f(t)$, other methods such as the one which computes the greatest common divisor of $f(t)$ and $f'(t)$ may be used instead.

3. Numerical examples and discussions

For the sake of convenience, let M_0, M_1, M_2, M_3, M_4, and M_5 be the classical Newton's method, the quadratic clipping method in [2], the cubic clipping method in [19], the rational cubic clipping method in [8], the rational cubic clipping method in [9], and the RQCM method in this paper, respectively.

The classical Newton's method M_0 is known to be very efficient for intervals which are known to have a single root of the function. So we firstly compare M_5 with M_0, by measuring the efficiency index (EI), which is defined as $d^{1/\gamma}$, where d is the convergence rate and γ is the number of functional evaluations (FE). There are six FEs in each clipping step of M_5 which achieves convergence rate 12, while there are two FEs in that of M_0 which achieves convergence rate 2, the efficiency indexes of M_5 is $12^{1/6} \approx 1.51$, which is better than that $\sqrt{2} \approx 1.41$ of M_0. In this paper, we compare the results between one clipping step of M_5 and three clipping steps of Newton's method M_0, both of which cost six FEs, and the corresponding convergence rates of M_5 and M_0 are 12 and 8, respectively. More details of the comparison results are shown in Table 1.

Method	M_0 (Newton's method)	M_5(New)
CR	8	12
EI	1.41	1.51

Table 1: Comparisons on convergence rate and EI between M_0 and M_5.

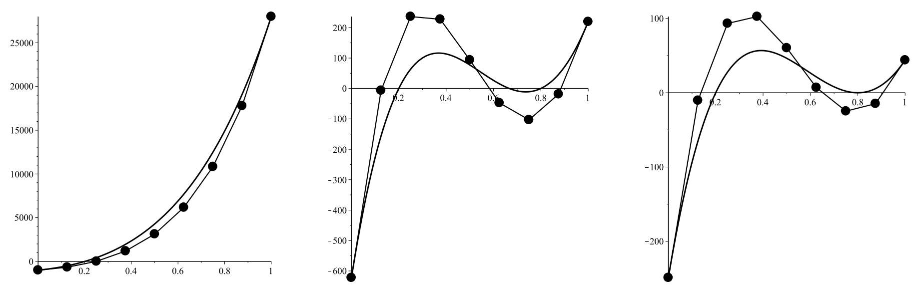
Secondly, we compare M_5 with M_i, $i = 1, 2, 3, 4$. At the beginning, we use the zeroes of the control polygon of $f(t)$ to divide the given interval into several subintervals, and consider the subintervals containing one root instead, see also the assumption. Table 2 shows the comparison results between different methods, where CR and $Time$ denote convergence rate and the computational complexity. As shown in Table 2, M_5 achieves the best convergence rate 12 for both simple and multiple root cases, while M_3 achieves convergence rate $7/k$ which is much higher than those of other remaining methods, where k is the corresponding multiplicity. Comparing with M_2, M_3 and M_4, both M_4 and M_5 are of $O(n)$ complexity, while the method M_5 needs to solve quadratic equations, which can be efficiently done than that of cubic equations in M_2, M_3 and M_4.

We have tested several examples for comparing the different methods, on a PC with CPU 2.2GHz and Memory 16GB. The average computation time of a clipping step is tested and obtained by setting the number of digits after

Method	M_1[2]	M_2[19]	M_3[8]	M_4[9]	M_5(New)
Time	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$
CR	$3/k$	$4/k$	7/k	$5/k$	12

Table 2: Comparisons on computation time and convergence rate.

decimal point as 16. The corresponding unit is millisecond, see also Tables 3 and 4. Note that, when setting the number of digits after decimal point, the larger the number, the higher the accuracy of the resulting root, and also the closer between the theoretical convergence rate and the resulting numerical convergence rate. In this paper, the number of digits after decimal point is set up to 5000 in the computation, and "/" in the following tables denotes that the corresponding number is beyond 5000 and is unavailable from the computation.



(a) (b) (c)

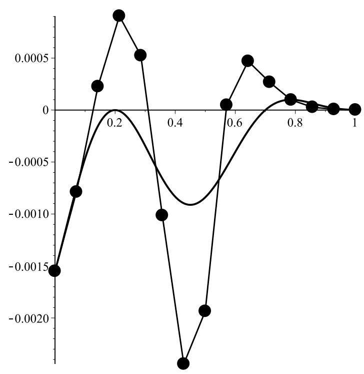
Figure 3: Example 3: Plots of (a) $f_3(t)$; (b) $f_4(t)$ and (c) $f_5(t)$, $t \in [0, 1]$.

Example 3. We have tested $f_3(t) = (t - 1/5)(t + 2)^3(t + 5)^4, t \in [0, 1]$, $f_4 = (t - 1/5)(t + 6)^3(t - 2/3)(t - 0.8)(t - 3)^2, t \in [0.128, 0.584]$ and $f_5(t) = (t - 1/5)(t + 6)^3(t - 4/5)^2(t - 3)^2, t \in [0.137, 0.653]$, which have a simple root within the subintervals obtained by using the zeros of their control polygons mapping to $[0, 1]$, see also Fig. 3 for their plots within $[0, 1]$. The comparison results are given in Table 3, including the error results of the first five clipping steps. The convergence rates of M_1, M_2, M_3, M_4 and M_5 tend to be 2, 3, 4, 7, 5 and 12, respectively. In Table 3, *Time* denotes the computational time with the unit millisecond. It shows that M_5 achieves the best computational efficiency of per clipping step, among these five methods.

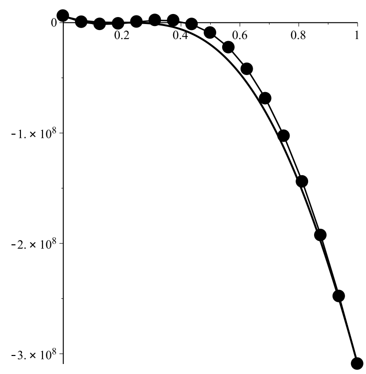
Example 4. We have tested $f_6(t) = (t - 0.20001)^2(t + 1/2)^5(t - 0.7)(t - 1.1)^6, t \in [0.126, 0.310]$ and $f_7 = (t - 1/5)^3(t - 5)^7(2 + t)^2(t + 7)^4, t \in [0, 1]$,

Exam	Method	1	2	3	4	5	*CR*	*Time*
$f_3(t)$	$M_1[2]$	5.0e-1	2.5e-2	3.0e-6	5.0e-18	2.3e-53	3	10.2
	$M_2[19]$	5.9e-2	4.5e-7	1.4e-27	1.6e-109	2.1e-437	4	18.5
	$M_3[8]$	8.7e-5	8.5e-34	6.9e-237	1.7e-1658	/	7	15.6
	$M_4[9]$	1.1e-2	7.8e-14	7.9e-71	5.9e-357	1.0e-1788	5	4.8
	M_5(New)	3.4e-7	2.1e-82	9.6e-982	/	/	12	1.9
$f_4(t)$	$M_1[2]$	4.7e-2	4.5e-5	3.8e-14	2.4e-41	5.7e-123	3	9.5
	$M_2[19]$	1.3e-3	3.1e-14	1.0e-56	1.1e-226	1.7e-906	4	19.1
	$M_3[8]$	6.9e-7	8.3e-48	3.2e-334	3.5e-2339	/	7	18.2
	$M_4[9]$	2.0e-4	3.1e-21	2.4e-105	6.3e-526	8.4e-2629	5	6.1
	M_5(New)	3.9e-5	7.5e-45	9.4e-526	/	/	12	2.1
$f_5(t)$	$M_1[2]$	5.6e-2	6.1e-5	7.6e-14	1.5e-40	1.2e-120	3	9.7
	$M_2[19]$	1.7e-3	5.9e-14	8.1e-56	2.8e-223	3.8e-893	4	18.8
	$M_3[8]$	1.3e-6	5.7e-46	1.6e-321	2.6e-2250	/	7	18.5
	$M_4[9]$	3.0e-4	1.5e-20	5.0e-102	2.0e-509	2.0e-2546	5	5.5
	M_5(New)	2.3e-5	2.1e-47	4.5e-557	/	/	12	2.0

Table 3: Comparison results of Example 4 for simple root cases.



(a)



(b)

Figure 4: Example 4: Plots of (a) $f_6(t)$; and (b) $f_7(t)$, $t \in [0, 1]$.

which have a double root and a triple root within the subintervals, see also Fig. 4 for their plots within the original interval [0, 1]. The comparison results are given in Table 4, including the error results of the first five clipping steps. The convergence rates of M_1, M_2, M_3, M_4 and M_5 tend to be $3/k$, $4/k$, $7/k$, $5/k$ and 12, respectively, where k is the corresponding multiplicity of the multiple root. The result of M_1 for the triple root is omitted, where the convergence rate of M_1 tends to be 1. Table 4 shows that M_5 achieves the best computational efficiency of per clipping step, among these five methods.

Exam	Method	1	2	3	4	5	CR	$Time$
$f_6(t)$ (k=2)	M_1[2]	5.1e-2	6.1e-3	2.5e-4	2.1e-6	1.7e-9	1.5	14.2
	M_2[19]	2.1e-2	2.7e-4	4.3e-8	1.1e-15	6.9e-31	2	24.3
	M_3[8]	3.4e-3	2.9e-9	1.6e-30	6.3e-105	2.5e-365	3.5	21.6
	M_4[9]	6.4e-3	1.6e-6	1.2e-17	9.1e-51	3.1e-150	3	6.3
	M_5(New)	2.3e-10	2.4e-118	2.5e-1414	/	/	12	3.1
$f_7(t)$ (k=3)	M_2[19]	5.0e-1	8.9e-2	4.6e-3	6.6e-5	2.3e-7	1.6	26.4
	M_3[8]	9.7e-2	3.3e-4	5.4e-10	1.8e-23	5.3e-55	2.4	22.6
	M_4[9]	2.9e-1	2.9e-2	5.3e-4	4.2e-7	1.4e-12	1.7	6.1
	M_5(New)	4.7e-12	6.1e-152	3.0e-1841	/	/	12	3.1

Table 4: Comparison results of Example 4 for multiple root cases

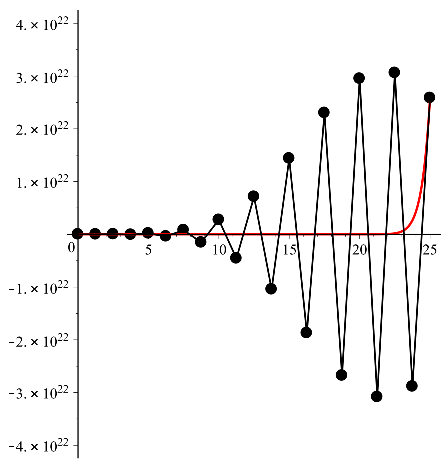


Figure 5: Plot of the Wilkinson polynomial [8].

Example 5. (also Example 6 in [8]) We have tested the Wilkinson polynomial (which is frequently used for testing robustness of clipping methods)

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

within $[0, 25]$, which has twenty zeros i, $i = 1, 2, \ldots, 20$, see also Fig. 5. 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 $W(t)$. Similarly as that of [8], we do the comparisons within the two of them $[2.83, 4.11](\triangleq \Lambda_1)$ and $[16.81, 18.07](\triangleq \Lambda_2)$, which contain two and two roots of $W(t)$, respectively. In this example, at the beginning, note that there is no root of the line determined by the two end points, we simply set t_0 as the mid point of the corresponding interval; after one clipping step, the two roots are separated and M_5 is strictly executed. The comparison results are shown in Table 5. Again, M_5 achieves much better performance than those of other four methods.

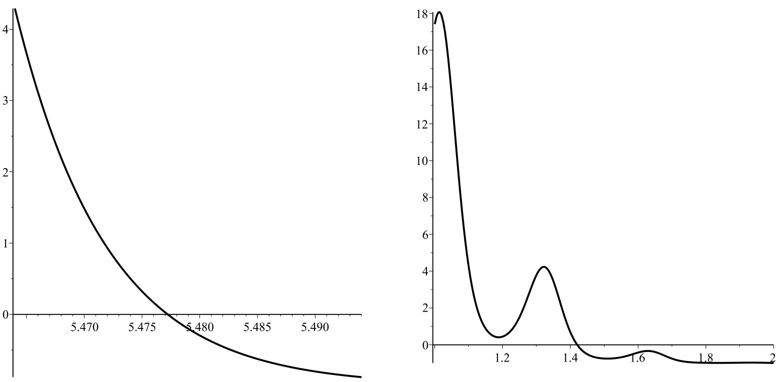
Case	Method	1	2	3	4	5	**CR**	**Time**
Λ_1	M_1[2]	3.3e-1	2.6e-3	7.9e-10	2.3e-29	5.7e-88	3	16.8
	M_2[19]	2.3e-2	1.0e-8	4.1e-34	1.0e-135	4.2e-542	4	28.2
	M_3[8]	5.6e-3	5.7e-18	1.1e-122	1.5e-855	/	7	26.5
	M_4[9]	3.1e-2	3.9e-9	8.7e-44	5.0e-217	3.1e-1083	5	9.3
	M_5(New)	5.6e-4	2.0e-36	9.2e-342	1.5e-3219	/	9	5.3
Λ_2	M_1[2]	3.8e-1	3.6e-3	1.3e-9	6.3e-29	6.8e-87	3	15.3
	M_2[19]	2.8e-1	2.2e-4	1.0e-16	4.1e-66	1.1e-263	4	27.5
	M_3[8]	2.8e-2	3.0e-14	4.3e-98	5.2e-685	/	7	23.6
	M_4[9]	5.9e-2	8.2e-8	4.0e-37	9.4e-184	4.4e-917	5	8.4
	M_5(New)	4.3e-2	5.8e-18	1.3e-167	7.5e-1578	/	9	5.6

Table 5: Comparison results of Example 5 between different methods

Example 6. In principle, the RQCM method can also work for non-polynomial function cases. We have compared M_5 with the Newton's methods M_6 in [18] and M_7 in [28], by testing $f_8(t) = 10^{(150-5t^2)} - 1, t \in [5.464, 5.494]$ and $f_9(t) = e^{\sin(20t)-t^3+3} - 1, t \in [1, 2]$, which have simple roots 5.4772 and 1.4204, respectively, as shown in Fig. 6. The comparison results are shown in Table 6. It shows that M_5 converges to correct results while the other two methods diverge (denoted by —).

4. Conclusions

This paper presents a rational quadratic clipping method (denoted by M_5) for finding a simple root of a polynomial within an interval. By optimizing the position of an inner interpolation point, it achieves convergence rate 12 to the root $t^\star$. By directly bounding $t^\star$, it is of linear complexity for per clipping process. In principle, it can also work for a non-polynomial function case. Numerical examples show that M_5 can achieve a better performance



(a) (b)

Figure 6: Example 4: Plots of (a) $f_8(t), t \in [5.464, 5.494]$; and (b) $f_9(t), t \in [1, 2]$.

Exam	Method	1	2	3	4	CR
$f_8(t)$	M_6[18]	1.6e+147	—			
	M_7[28]	41.5	—			
	M_5(New)	5.0e-6	2.2e-52	6.3e-605	/	12
$f_9(t)$	M_6[18]	0.5515	—			
	M_7[28]	1.0178	1.5120	5.4108	—	
	M_5(New)	9.0e-2	6.5e-8	2.6e-81	2.4e-957	12

Table 6: Comparison results of Example 6 for non-polynomial cases

than those of other four clipping methods, and can converge to correct result in some cases where Newton's methods fail.

At the moment, we assume that the roots of $f(t)$ are isolated by using previous methods and there is a simple root within the given interval. As for future work, one may discuss how to isolate the roots of a non-polynomial function within an interval, and how to deal with a multiple root case for a non-polynomial function of complicated shape. Moreover, methods similar to the one presented here can be applied to other system of B-spline functions, where the given curve is approximated by a rational B-spline curve of degree two or three. Another topic for future research is to extend our method to the cases for computing the zero sets of tensor product spline surfaces.

Acknowledgement

This research was partially supported by the National Science Foundation of China (61672009,61502130) and City University of Hong Kong (SRG 7004452).

References

[1] M. Aizenshtein, B. Michael, and E. Gershon, *Global solutions of well-constrained transcendental systems using expression trees and a single solution test*, Computer Aided Geometric Design **29** (2012), no. 5, 265–279.

[2] M. Bartŏn and B. Jüttler, *Computing roots of polynomials by quadratic clipping*, Computer Aided Geometric Design **24** (2007), no. 3, 125–141.

[3] M. Bartoň, G. Elber, and I. Hanniel, *Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests*, Computer-Aided Design **43** (2011), no. 8, 1035–1044.

[4] G. Baker and P. Graves-Morris, *Padé approximants*, New York: Cambridge University Press, 1996.

[5] P. Borwein and T. Erdlyi, *Polynomials and polynomial inequalities*, Springer-Verlag, New York, 1995.

[6] X. D. Chen, J. H. Yong, G. Z. Wang, J. C. Paul, and G. Xu, *Computing the minimum distance between a point and a NURBS curve*, Computer-Aided Design **40** (2008), no. 10-11, 1051–1054.

[7] X. D. Chen and W. Ma, *A planar quadratic clipping method for computing a root of a polynomial in an interval*, Computers & Graphics **46** (2015), no. 1, 89–98.

[8] X. D. Chen, W. Ma, and Y. T. Ye, *A rational cubic clipping method for computing real roots of a polynomial*, Computer Aided Geometric Design **38** (2015), 40–50.

[9] X. Chen and W. Ma, *Rational cubic clipping with linear complexity for computing roots of polynomials*, Applied Mathematics and Computation **273** (2016), 1051–1058.

[10] Y. K. Choi, W. P. Wang, Y. Liu, and M. S. Kim, *Continuous collision detection for two moving elliptic disks*, IEEE Transactions on Robotics **22** (2006), no. 2, 213–224.

[11] P. J. Davis, *Interpolation and approximation*, Dover Publications, New York, 1975.

[12] J. Džunić and M. S. Petković, *On generalized biparametric multipoint root finding methods with memory*, Journal of Computational and Applied Mathematics **255** (2014), 362–375.

[13] A. Efremov, V. Havran, and H.-P. Seidel, *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, 2005.

[14] G. Elber and M.-S. Kim, *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, 2001.

[15] R. T. Farouki and T. N. T. Goodman, *On the optimal stability of the Bernstein basis*, Mathematics of Computation **126** (1996), no. 65, 1553–1566.

[16] M. Floater, *High order approximation of rational curves by polynomial curves*, Computer Aided Geometric Design **23** (2006), no. 8, 621–628.

[17] B. Jüttler, *The dual basis functions of the Bernstein polynomials*, Advanced in Computational Mathematics **8** (1998), 345–352.

[18] X. Li, C. Mu, J. Ma, and C. Wang, *Sixteenth-order method for nonlinear equations*, Applied Mathematics & Computation **215** (2010), no. 10, 3754–3758.

[19] L. G. Liu, L. Zhang, B. B. Lin, and G. J. Wang, *Fast approach for computing roots of polynomials using cubic clipping*, Computer Aided Geometric Design **26** (2009), 547–559.

[20] J. Machchhar and G. Elber, *Revisiting the problem of zeros of univariate scalar Béziers*, Elsevier Science Publishers B. V. (2016), no. 43, 16–26.

[21] K. Mørken and M. Reimers, *An unconditionally convergent method for computing zeros of splines and polynomials*, Mathematics of Computation **76** (2007), no. 258, 845–865.

[22] T. Nishita, T. W. Sederberg, and M. Kakimoto, *Ray tracing trimmed rational surface patches*, in: Proceedings of Siggraph. ACM, pp. 337–345, 1990.

[23] M. Reuter, T. Mikkelsen, E. Sherbrooke, T. Maekawa, and N. Patrikalakis, *Solving nonlinear polynomial systems in the barycentric Bernstein basis*, The Visual Computer **24** (2007), no. 3, 187–200.

[24] F. Rouillier and P. Zimmermann, *Efficient isolation of polynomial's real roots*, Journal of Computational and Applied Mathematics **162** (2004), no. 1, 33–50.

[25] M. Sagraloff and K. Mehlhorn, *Computing real roots of real polynomials*, Journal of Symbolic Computation (2016), no. 73, 46–86.

[26] T. W. Sederberg and T. Nishita, *Curve intersection using Bézier clipping*, Computer-Aided Design **22** (1990), no. 9, 538–549.

[27] C. Schulz, *Bézier clipping is quadratically convergent*, Computer Aided Geometric Design **26** (2009), no. 1, 61–74.

[28] J. Sharma and H. Arora, *A new family of optimal eighth order methods with dynamics for nonlinear equations*, Applied Mathematics & Computation **273** (2016), no. 1, 924–933.

[29] R. H. Wang and G. Q. Zhu, *Rational function approximation and its application*, Beijing: Science, pp. 54–183, 2004.

Key Laboratory of Complex Systems Modeling and Simulation
Hangzhou Dianzi University, Hangzhou, China
and Department of MBE, City University of Hong Kong
Hong Kong, P. R. China
E-mail address: xiaodiao@hdu.edu.cn

Key Laboratory of Complex Systems Modeling and Simulation
Hangzhou Dianzi University, Hangzhou, China

Key Laboratory of Complex Systems Modeling and Simulation
Hangzhou Dianzi University, Hangzhou, China

Key Laboratory of Complex Systems Modeling and Simulation
Hangzhou Dianzi University, Hangzhou, China

Department of MBE, City University of Hong Kong
Hong Kong, P. R. China
E-mail address: mewma@cityu.edu.hk