Improved Computation of Bounds for Positive Roots of Polynomials

Masami Takata¹, Takuto Akiyama², Sho Araki³, Kinji Kimura⁴, and Yoshimasa Nakamura⁵

¹Academic Group of Information and Computer Sciences, Nara Women’s University, Kita-Uoya-Nishi-Machi, Nara 630-8506, JAPAN
²³Graduate School of Informatics, Kyoto University, Yoshidahonmachii, Sakyo-ku, Kyoto 606-8501, JAPAN
⁴Graduate School of Informatics, Kyoto University, Yoshidahonmachii, Sakyo-ku, Kyoto 606-8501, JAPAN
⁵Graduate School of Informatics, Kyoto University, Yoshidahonmachii, Sakyo-ku, Kyoto 606-8501, JAPAN
¹takata@ics.nara-wu.ac.jp, ²akiyama@amp.i.kyoto-u.ac.jp, ³araki@amp.i.kyoto-u.ac.jp, ⁴kkimur@amp.i.kyoto-u.ac.jp, ⁵ynaka@i.kyoto-u.ac.jp

Abstract—A new lower bound for computing positive roots of polynomial equations is proposed. We discuss a two-stage algorithm for computing positive roots of polynomial equations. In the first stage, we use the continued fraction method based on Vincent’s theorem, which employs the lower bound of real roots, for isolating the positive roots into intervals. In the second stage, we apply a bisection method to each interval. In order to compute the proposed lower bound, we follow three steps. First, we compute a candidate for the lower bound generated by Newton’s method. Second, by using Laguerre’s theorem, we check whether the candidate for the lower bound is suitable. Third, we compare the local-max bound and the proposed lower bound. Then, we employ the larger bound to accelerate the continued fraction method based on Vincent’s theorem. Finally, we conduct experiments to evaluate the effectiveness of the proposed lower bound.

(This paper is submitted to PDPTA’13.)

Keywords: continued fraction method, Vincent’s theorem, local-max bound, Newton’s method, Laguerre’s theorem

1. Introduction

The real roots of univariate polynomial equations are more useful than the imaginary roots for practical applications in various engineering fields. Thus, the objective of this study is the computation of all real roots of polynomial equations. For this purpose, we develop a real-root isolation algorithm. For polynomial equations without multiple roots, each root can be isolated into a numeric interval. Then, the accuracy of the isolated real roots can be easily enhanced by using a bisection method.

The continued fraction method for isolating the positive roots of univariate polynomial equations is based on Vincent’s theorem [2], [10]. In this method, each positive root is isolated using Descartes’ rule of signs [3], which focuses on the coefficients of the polynomial equations. The execution of Descartes’ rule of signs requires origin shifts. Thus, several coefficients of a polynomial equation are transformed into nonzero coefficients, even in the case of sparse polynomial equations, which have many zero coefficients. The Krawczyk method [8], which is based on the numerical verification method, is a technique that has been developed for isolating the positive roots of polynomial equations that have many zero coefficients. In this paper, we investigate the continued fraction method, which is based on Vincent’s theorem, for isolating the positive roots of polynomial equations that have many nonzero coefficients.

To accelerate the continued fraction method based on Vincent’s theorem, the lower bound of the smallest positive root is required. In general, to obtain the lower bound of positive roots of a polynomial equation, we first substitute $1/x$ for $x$ in the polynomial equation $f(x)$. Second, we compute the upper bound of the positive roots. Third, we obtain the lower bound by computing the inverse of the upper bound. The Cauchy bound [9] and the Kioustelidis bound [6] are known as upper bounds of the positive roots of polynomial equations. Akritas et al. introduced a generalized theorem including the Cauchy bound and the Kioustelidis bound [11]. Then, by specializing this generalized theorem, they proposed a new upper bound called the local-max bound, which is different from both the Cauchy bound and the Kioustelidis bound.

In this paper, we propose a new lower bound for accelerating the continued fraction method based on Vincent’s theorem. The new lower bound is obtained using Newton’s method [5] and Laguerre’s theorem [7]. The magnitude correlation of the local-max bound and the new lower bound depends on the input polynomial. Thus, after we compare the local-max bound and the new lower bound, the larger lower bound is adopted for the continued fraction method based on Vincent’s theorem. Note that computing lower bounds incurs computation time. Therefore, the new lower bound must be sometimes larger than the local-max bound.

To evaluate the new lower bound, we compare the time for computing the lower bound generated by only the local-max bound with that for computing the lower bound generated by both the local-max bound and the new lower bound. If the computation time decreases by virtue of the new lower bound, it is proved that the new lower bound is sometimes larger than the local-max bound.

The remainder of this paper is organized as follows. In Section 2, we describe univariate polynomial equations. In
section 3, we introduce the continued fraction method based on vincent’s theorem. in section 4, we propose the new lower bound, which is computed using newton’s method and checked using laguerre’s theorem. in section 5, we evaluate the effectiveness of the proposed lower bound. finally, in section 6, we state our conclusions and briefly describe the scope for further investigation.

2. Positive Roots of Polynomials

To compute the positive roots of a polynomial equation

\[ f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0, \quad x \in \mathbb{R}, a_i \in \mathbb{Z}, \]

in the interval \( x \in (0, \infty) \), we first isolate each root into a numeric interval. second, we improve the accuracy of the real roots by using a bisection method. if a real root is equal to 0, then \( a_n = 0 \). thus, we discuss only the problem in \( x \in (0, \infty) \). here, the intervals are defined by,

\[ x \in [a, b], \quad x \in (a, b) \text{ or } x \in [a, b), \quad a, b \in \mathbb{R}, a \leq b, \tag{2} \]

where \([,], (,)\) denote a closed interval, a left-open right-closed interval, and a left-closed right-open interval, respectively.

A polynomial equation \( f(x) \) should have no multiple roots so that the continued fraction method can be employed for isolating its positive roots. if a polynomial equation \( g_1(x) \) has multiple roots, then these roots are transformed into simple roots by using the equation

\[ f(x) := \frac{g_1(x)}{\text{G.C.D.}(g_1(x), g'_1(x))}, \tag{3} \]

where G.C.D. is the greatest common divisor related to two equations, and \( g'_1(x) \) denotes the first derivative of \( g_1(x) \). thus, we can assume that \( f(x) \) have no multiple roots.

A polynomial equation \( g_2(x) \) with rational coefficients \( b_i \) is given by the equation

\[ g_2(x) = b_0x^n + b_1x^{n-1} + \cdots + b_{n-1}x + b_n = 0, \tag{4} \]

\[ b_i = \frac{d_{i,1}}{d_{i,1}} \in \mathbb{Q}. \]

By using the least common multiple, or L.C.M., of all \( d_{i,1} \), the polynomial equation \( g_2(x) \) can be transformed into the equation \( f(x) \) with integer coefficients:

\[ f(x) := \text{L.C.M.}(d_{i,1})g_2(x). \tag{5} \]

A polynomial equation \( g_3(y) \) with integer coefficients and no multiple roots is set to the equation

\[ g_3(y) = a_0y^n + a_1y^{n-1} + \cdots + a_{n-1}y + a_n = 0, \quad y \in \mathbb{R}, a_i \in \mathbb{Z}. \tag{6} \]

The real roots of \( g_3(y) \) in the interval \([u,v], u, v \in \mathbb{R}\) are isolated through the following procedure. By using the replacement,

\[ y \rightarrow -\frac{1}{x + \frac{1}{-u+v}} + v, \tag{7} \]

we transform \( g_3(y) \) into the equation

\[ g_4(x) = g_3 \left( \frac{-1}{x + \frac{1}{-u+v}} + v \right), \tag{8} \]

\[ f(x) := \text{numerator} \left( g_4(x) \right), \tag{9} \]

where the function “numerator” implies computation of the numerator of \( g_4(x) \). under eq. (7), \( y \in [u,v] \) in \( g_3(y) \) corresponds to \( x \in [0, \infty) \) in \( f(x) \). however, the case that \( y = v \) is not included in the interval \([u,v]\). Thus we treat the case that \( y = v \) as a special case.

Hence, polynomial equations with rational coefficients and multiple roots in the interval \([u,v]\) can be transformed into eq. (1) by the above operations.

3. Continued Fraction Method based Vincent’s Theorem for Isolating Positive Roots

3.1 Concept

In the continued fraction method based on vincent’s theorem, real roots in \((0, \infty)\) can be isolated using the Descartes’ rule of signs.

Descartes’ rule of signs is derived from the following theorem.

Theorem 1 (The Descartes’ rule of signs): In a polynomial

\[ f(x) = a_0x^n + \cdots + a_{n-1}x + a_n = 0, \quad x \in \mathbb{R}, \]

with real coefficients,

\[ W := \text{the number of “changes of sign” in the list of coefficients}\{a_0, a_1, \ldots, a_n\} \text{ except } a_i = 0, \]

\[ N := \text{the number of positive roots in } (0, \infty) \]

are defined. under these definitions, we have,

\[ N = W - 2h. \]

Here, \( h \) is a non-negative integer.

By using Theorem 1, the number of positive roots of the polynomial equation \( f(x) \) is determined in the following conditional branch:

- In the case that \( W = 0, f(x), x \in (0, \infty) \) does not have any positive roots.
In the case that \( W = 1 \), \( f(x) \) has only one positive root in the interval \( x \in (0, \infty) \).

In the case that \( W \geq 2 \), the number of positive roots of \( f(x) \) cannot be determined.

In the case that \( W = 1 \), the isolated interval should be set to \((0, ub]\), where \( ub \) denotes the upper bound of positive roots for a polynomial equation \( f(x) \). Computation methods for the upper bound of positive roots of a polynomial equation \( f(x) \) are described in Section 3.2.

In the case that \( W \geq 2 \), we divide the interval \((0, \infty)\) in the two intervals. Then, Descartes’ rule of signs is applied in each interval. In the continued fraction method based on Vincent’s theorem, the interval \((0, \infty)\) is divided in \((0, 1)\) and \((1, \infty)\). The division is performed by the replacement

\[
x \to x + 1,
\]

\[
x \to \frac{1}{x + 1}.
\]

By using the replacement \( x \to x + 1 \), the interval \((0, \infty)\) of the replaced polynomial equation corresponds to the interval \((1, \infty)\) of the original polynomial equation. Similarly, by using the replacement \( x \to 1/(x + 1) \), the interval \((0, \infty)\) of the replaced polynomial equation corresponds to the interval \((0, 1)\) of the original polynomial equation. The intervals \((1, \infty)\) and \((0, 1)\) do not include the case that \( x = 1 \). To solve for this case, after either replacement, we check whether the coefficient \( a_n \), which is a constant term, vanishes. If \( a_n = 0 \) in the replaced polynomial equation, then \( 1 \) is a root of the original polynomial equation.

In the above replacements, the computation of coefficients of a replaced polynomial equation is required. For example, in the case that the coefficients of

\[ g_5(x) = a_0(x + 1)^3 + a_1(x + 1)^2 + a_3(x + 1) + a_4, \]  

are computed, the synthetic division, which is shown in Table 1, can be adopted. In Table 1, the coefficients of \( x^3, x^2, x^1 \), and \( x^0 \) become \( a_0, 3a_0 + a_1, 3a_0 + 2a_1 + a_2, \text{ and } a_0 + a_1 + 2a_2 + a_3 \), respectively. Thus, by using the synthetic division, the computation cost, which is incurred for obtaining the coefficients of the replaced polynomial equation for a replacement \( x \to x + 1 \), is \( O(n^2) \).

### 3.2 Computation for Upper Bound

To obtain the upper bound, the following theorem [1] can be applied.

---

**Table 1: Synthetic division in \( g_5(x) \).**

<table>
<thead>
<tr>
<th>( a_0 )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_0 )</td>
<td>( a_0 + 1 )</td>
<td>( a_0 + 1 + 2 )</td>
<td>( a_0 + 1 + 2 + 2 )</td>
</tr>
<tr>
<td>( 2a_0 + 1 )</td>
<td>( 3a_0 + 2a_1 + a_2 )</td>
<td>( a_0 + a_1 + 2a_2 + a_3 )</td>
<td></td>
</tr>
<tr>
<td>( 3a_0 + a_1 )</td>
<td>( a_0 )</td>
<td>( a_0 + a_1 + 2a_2 + a_3 )</td>
<td></td>
</tr>
</tbody>
</table>

---

**Theorem 2 (Akritas, 2006):** Let \( f(x) \) be a polynomial with real coefficients and we assume \( a_0 > 0 \) in this section. Let \( d(f) \) and \( t(f) \) denote its degree and number of terms, respectively.

In addition, assume that \( f(x) \) can be reshaped as follows:

\[
f(x) = q_1(x) - q_2(x) + \cdots - q_{2m}(x) + g_0(x),
\]

where all the polynomials \( q_i(x), i = 1, 2, \ldots, 2m \) and \( g_0(x) \) have only positive coefficients. Moreover, assume that for \( i = 1, 2, \ldots, m \) we obtain,

\[
q_{2i-1}(x) = c_{2i-1,1}x^{e_{2i-1,1}} + \cdots + c_{2i-1,l(q_{2i-1})}x^{e_{2i-1,l(q_{2i-1})}}
\]

and

\[
q_{2i}(x) = b_{2i,1}x^{e_{2i,1}} + \cdots + b_{2i,l(q_{2i})}x^{e_{2i,l(q_{2i})}}
\]

where \( e_{2i-1,1} = d(q_{2i-1}) \) and \( e_{2i,1} = d(q_{2i}) \), and the exponent of each term in \( q_{2i-1}(x) \) is greater than the exponent of each term in \( q_{2i}(x) \). If for all indices \( i = 1, 2, \ldots, m \), we obtain

\[
t(q_{2i-1}) \geq t(q_{2i}),
\]

then the upper bound of the positive roots of \( f(x) \) is defined by

\[
ub = \max_{i=1,2,\ldots,m} \left\{ \left( \frac{b_{2i,1}}{c_{2i-1,1}} \right)^{\frac{1}{e_{2i-1,l(q_{2i-1})}-e_{2i,1}}}, \ldots, \left( \frac{b_{2i,l(q_{2i})}}{c_{2i-1,l(q_{2i})}} \right)^{\frac{1}{e_{2i-1,l(q_{2i-1})}-e_{2i,l(q_{2i})}}} \right\},
\]

(15)

for any permutation of the positive coefficients \( c_{2i-1,j}, j = 1, 2, \ldots, t(q_{2i-1}) \). Otherwise, for each of the indices \( i \) for which we obtain

\[
t(q_{2i-1}) < t(q_{2i}),
\]

we break up one of the coefficients of \( q_{2i-1}(x) \) into \( t(q_{2i}) - t(q_{2i-1}) + 1 \) parts, so that now \( t(q_{2i}) = t(q_{2i-1}) \) and we apply the same formula defined in eq. (15).

In general, we can get better bounds if we pair coefficients from non-adjacent polynomials \( q_{2i-1}(x) \) and \( q_{2i}(x) \), for \( 1 \leq l < i \). A well-known implementation of this type is the Cauchy rule. In the Cauchy rule, if \( f(x) \) is given by eq. (1), of degree \( n \geq 0 \), with \( a_k < 0 \) for at least one \( k, 1 \leq k \leq n \), and if \( \lambda \) is the number of negative coefficients, then an upper bound of the positive roots of \( f(x) \) is defined by

\[
ub_1 = \max_{1 \leq k \leq n; a_k < 0} \sqrt[\lambda]{\frac{1}{\lambda} a_k}.
\]

(17)
Algorithm 1 Implementation of the “local-max” bound.

\[
cl \leftarrow \{a_n, a_{n-1}, \ldots, a_1, a_0\}
\]

if \( n + 1 \leq 1 \) then

return \( ub_3 = 0 \)

end if

\( j = n + 1 \)

\( t = 1 \)

for \( i = n \) to 1 step -1 do

if \( cl(i) < 0 \) then

\( tempub = (2^i(-cl(i)/cl(j)))^{1/j-i} \)

if \( tempub > ub \) then

\( ub = tempub \)

end if

\( t = t + 1 \)

else if \( cl(i) > cl(j) \) then

\( j = i \)

\( t = 1 \)

end if

end for

\( ub_3 = ub \)

We have another bound called Kioustelidis’ bound [6] as follows:

\[
ub_2 = 2 \max_{1 \leq k \leq n; a_k < 0} \sqrt[k]{-\frac{a_k}{a_0}} \quad (18)
\]

The Kioustelidis bound is closely related to the Cauchy rule. However, the Cauchy bound and the Kioustelidis bound give an overestimation of the upper bound for some cases. Thus, Akritas et al. introduced the local-max pairing strategy (defined in Definition 1) in order to generate a suitable bound.

Definition 1 (“local-max”): For a polynomial equation \( f(x) \) given by eq. (1), the coefficient \(-a_k\) of the term \(-a_kx^{n-k}\) in \( f(x) \) is paired with the coefficient \( a_m/2^t \) of the term \( a_mx^{n-m} \), where \( a_m \) is the largest positive coefficient with \( 0 \leq m < k \) and \( t \) denotes the number of times the coefficient \( a_m \) has been used.

The implementation of the local-max bound is described in Algorithm 1, and the output is \( ub_3 \).

3.3 Acceleration using Lower Bound

The continued fraction method based on Vincent’s theorem requires many replacement operations \( x \to x + 1 \) and \( x \to 1/(x + 1) \). In other words, the origin shift is realized by \( x \to x + 1 \). Thus, if the positive roots are much larger than 1, then the computation time increases, as we must repeat the replacement operation \( x \to x + 1 \). To decrease the computation time, the lower bound of the smallest positive root of a polynomial equation should be used as a shift.

In general, to obtain the lower bound \( lb \) of an original polynomial equation, we first substitute \( 1/x \) for \( x \) in the original polynomial equation. Second, we compute the upper bound \( ub_3 \) of the positive roots. Third, we obtain the lower bound \( lb \) by computing the inverse of the upper bound as follows:

\[
lb = \frac{1}{ub_3}. \quad (19)
\]

If \( lb > 1 \), then the replacement \( x \to x + lb \) is adopted, as the computation time for isolating the positive roots decreases. If \( lb \leq 1 \), then we do not adopt the lower bound \( lb \), as the lower bound \( lb \) is not sufficiently large to reduce the computation time.

Algorithm 2 shows a continued fraction method based on Vincent’s theorem with origin shift using the local-max bound. The computation time for the Algorithm 2 is less than that for the continued fraction method based on Vincent’s theorem without the origin shift. The replacements \( x \to x + 1 \) and \( x \to 1/(x + 1) \) are called Möbius transformations. After the intervals for isolating the positive roots of a polynomial equation are determined, each interval should be replaced by the interval processed by all inverse transformations of Möbius transformations.

4. New Lower Bound

The acceleration of the continued fraction method based on Vincent’s theorem employs the origin shift using the local-max bound. Thus, if the lower bound tends to the smallest positive root, then the computation time of the continued fraction method decreases.

In this paper, we propose a new lower bound generated by Newton’s method. Note that in some polynomial equations, a bound generated by Newton’s method is not suitable as the lower bound. Hence, by using Laguerre’s theorem, it must be checked whether a bound generated by Newton’s method is a suitable lower bound.

Newton’s method is defined by the following recurrence formula:

\[
x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}. \quad (20)
\]

Here, \( f'(x) \) denotes the first derivative of \( f(x) \). If Newton’s method is adopted at the origin, then a candidate for the lower bound \( r \) is computed as follows:

\[
r = 0 - \frac{f(0)}{f'(0)} = -\frac{a_n}{a_{n-1}}. \quad (21)
\]

The cost for computing \( r \) is \( O(1) \).

We can check whether a candidate for the lower bound \( r \) is suitable by using the following theorem.
Algorithm 2 Continued fraction method based on Vincent’s theorem with the local-max shift strategy.

\[ R \leftarrow \phi \]
\[ S \leftarrow \{poly\} \]
if 0 is solution of poly then
\[ R \leftarrow R \cup [0, 0] \]
\[ poly \leftarrow poly/x \]
end if
while \( S \neq \phi \) do
\[ poly \leftarrow \text{dequeue}(S) \]
\[ W \leftarrow \text{Descartes}(poly) \]
if \( W = 1 \) then
\[ ub_3 \leftarrow \text{Algorithm1 with poly} \]
\[ R \leftarrow R \cup \text{Inverse Möbius trans}\((0, ub_3)\) \]
else if \( W \geq 2 \) then
\[ poly_2 \leftarrow \text{Trans}(poly, x \rightarrow 1/x) \]
\[ ub_3 \leftarrow \text{Algorithm1 with poly2} \]
\[ lb \leftarrow 1/ub_3 \]
if \( lb > 1 \) then
\[ poly \leftarrow \text{Trans}(poly, x \rightarrow x + lb) \]
if 0 is solution of poly then
\[ R \leftarrow R \cup \text{Inverse Möbius trans}\([lb, lb]\) \]
\[ poly \leftarrow poly/x \]
end if
end if
\[ poly_3 \leftarrow \text{Trans}(poly, x \rightarrow x + 1) \]
if 0 is solution of poly3 then
\[ R \leftarrow R \cup \text{Inverse Möbius trans}\([1, 1]\) \]
\[ poly_3 \leftarrow poly_3/x \]
end if
\[ poly_4 \leftarrow \text{Trans}(poly, x \rightarrow 1/x + 1) \]
\[ S \leftarrow S \cup \{poly_3, poly_4\} \]
end if
end while

**Theorem 3 (the Laguerre theorem):** For a polynomial equation

\[ f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n = 0 \]

with real coefficients, let \( N \) be the number of positive roots that are larger than a positive value \( \alpha \). The number \( N \) is less than or equal to the number of sign changes in the following polynomials \( f_k(\alpha) \):

\[ f_k(\alpha) = a_0\alpha^k + a_1\alpha^{k-1} + \cdots + a_k, k = 0, 1, \ldots, n. \]

Here, \( f(\alpha) \neq 0 \) is assumed.

If \( x \) in Theorem 3 is replaced by \( 1/x \), Theorem 3 can be transformed into the following theorem.

**Theorem 4:** The number of positive roots of a polynomial equation

\[ f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0, \]

in the interval \( 0 < x < r \) is less than or equal to the number of sign changes in the following polynomials \( p_k(r) \):

\[ p_0(r) = a_0, \]
\[ p_1(r) = a_0 + a_1r, \]
\[ p_2(r) = a_0 + a_1r + a_2r^2, \ldots, \]
\[ p_n(r) = a_0 + a_1r + \cdots + a_nr^n. \]

Here, \( r > 0 \) and \( p_n(r) \neq 0 \).

Thus, if the number of sign changes in \( p_k(r), k = 0, \ldots, n \) is 0, then no positive roots in the interval \( 0 < x < r \) exist. In such a case, the computation cost is \( O(n) \).

To get a candidate for the lower bound \( r \), it is necessary that the signs of \( a_n \) and \( a_{n-1} \) must be opposite. Moreover, it is needed to check \( p_n(r) \neq 0 \).

If a candidate for the lower bound \( r \) generated by Newton’s method is the lower bound of the smallest positive root, then we adopt the lower bound \( lb \) as an origin shift defined in the following equation:

\[ lb = \max\left(\frac{1}{ub_3}, r\right). \]  (22)

If \( lb > 1 \), then the origin shift \( x \rightarrow x + lb \) is performed.

The improved algorithm of the continued fraction method based on Vincent’s theorem with the shift strategy, including both the local-max bound and the new lower bound generated by Newton’s method, is shown in Algorithm 3.

**5. Experiment**

In this section, we conduct experiments to evaluate the effectiveness of the proposed lower bound.

Here, Algorithm 2 and Algorithm 3 are compared.

As test polynomial equations, we use \( f(x) \) with integer coefficients:

\[ f(x) = \prod_{i=0}^{r} (x - x_i) \times \prod_{j=0}^{s} (x - \alpha_j + i\beta_j)(x - \alpha_j - i\beta_j), \]  (23)
\[ x_i, \alpha_j, \beta_j \in \mathbb{R}. \]

Here, parameters \( x_i, \alpha_j, \) and \( \beta_j \) are randomly set as follows:

\[-10000 \leq x_i, \alpha_j, \beta_j \leq 10000, \]  (24)

Parameters \( s \) and \( r \) are set to \( s = 490, r = 20 \). Then, we generate 100 test polynomial equations.

Table 2 shows the experimental environment. In the continued fraction method based on Vincent’s theorem, the multiple-precision arithmetic library GMP [4] is needed to compute all coefficients in replaced polynomial equations.

Figure 1 shows the plots of the computation time in all test polynomial equations. In Figure 1, the computation time for Algorithm 3 is less than that for Algorithm 2, and the
Algorithm 3 Improvement of the continued fraction method based on Vincent’s theorem with the shift strategy including both the local-max bound and the new lower bound generated by Newton’s method.

\[
\begin{align*}
R & \leftarrow \phi \\
S & \leftarrow \{poly\} \\
\text{if} \ 0 \ \text{is a solution of } poly \ & \text{then} \\
R & \leftarrow R \cup [0, 0] \\
poly & \leftarrow poly/x \\
\text{end if} \\
\text{while } S \neq \phi \text{ do} \\
poly & \leftarrow \text{dequeue}(S) \\
W & \leftarrow \text{Descartes}(poly) \\
\text{if} \ W = 1 \ & \text{then} \\
u_b & \leftarrow \text{Algorithm 1 with } poly \\
R & \leftarrow R \cup \text{Inverse Möbius trans } \((0, u_b)\) \\
\text{else if } W \geq 2 \ & \text{then} \\
poly_2 & \leftarrow \text{Trans}(poly, x \rightarrow 1/x) \\
u_b_3 & \leftarrow \text{Algorithm 1 with } poly_2 \\
r & \leftarrow \text{NewtonLowerbound}(poly) \text{ which is checked by using the Laguerre theorem} \\
lb & \leftarrow \max(1/u_b_3, r) \\
\text{if} \ lb > 1 \ & \text{then} \\
poly & \leftarrow \text{Trans}(poly, x \rightarrow x + lb) \\
\text{if} \ 0 \ \text{is a solution of } poly \ & \text{then} \\
R & \leftarrow R \cup \text{Inverse Möbius trans } \([lb, lb]\) \\
poly & \leftarrow poly/x \\
\text{end if} \\
\text{end if} \\
poly_3 & \leftarrow \text{Trans}(poly, x \rightarrow x + 1) \\
\text{if} \ 0 \ \text{is a solution of } poly_3 \ & \text{then} \\
R & \leftarrow R \cup \text{Inverse Möbius trans } \([1, 1]\) \\
poly_3 & \leftarrow poly_3/x \\
\text{end if} \\
poly_4 & \leftarrow \text{Trans}(poly, x \rightarrow 1/x + 1) \\
S & \leftarrow S \cup \{poly_3, poly_4\} \\
\text{end if} \\
\text{end while}
\end{align*}
\]

The difference among the computation time in Algorithm 3 is small.

Table 3 shows the computation time for the 100 random polynomial equations. The maximum computation time for Algorithm 3 is 1.48 times faster than that for Algorithm 2. The average computation time for Algorithm 3 is 1.09 times faster than that for Algorithm 2. The standard deviations in Algorithm 2 and Algorithm 3 are not considerably large.

The computation cost of the local-max bound, which is used the continued fraction method based on Vincent’s theorem described in Section 3.3, is \(O(n)\). The computation cost of Newton’s method, which can compute a candidate for the lower bound \(r\), is \(O(1)\). The computation cost of Laguerre’s theorem, which can check whether a candidate for the lower bound \(r\) is suitable, is \(O(n)\). Thus, the computation cost of Algorithm 3 is equal to that of Algorithm 2. However, Figure 1 and Table 3, the computation time for Algorithm 3 is less than that for Algorithm 2. This is because some lower bounds generated from Newton’s method are more suitable than the local-max bound. Consequently, the continued fraction method based on Vincent’s theorem is improved by using the proposed lower bound.

Hence, the improved continued fraction method with the local-max bound and the proposed lower bound generated by Newton’s method is efficient.

### 6. Conclusions

In this paper, we proposed a new lower bound for accelerating the continued fraction method based on Vincent’s theorem. To accelerate this method, a suitable lower bound should be computed. In the original continued fraction method based on Vincent’s theorem, the local-max bound is adopted. In contrast, we used Newton’s method to obtain another lower bound. This method can sometimes generate a lower bound larger than the local-max bound. To compute the proposed lower bound, we followed three steps. First, we computed a candidate for the lower bound generated by Newton’s method. Second, we used Laguerre’s theorem to check whether the candidate is suitable. Third, we compared the local-max bound and the new lower bound. Then, we employed the larger bound to accelerate the continued fraction method based on Vincent’s theorem. Experiments were conducted to evaluate the proposed lower bound. The results showed that the average computation time of the
continued fraction method with both the local-max bound and the proposed lower bound is 1.09 times faster than that with only the local-max bound. Hence, the proposed lower bound is effective.

In the future, the proposed lower bound should be evaluated using different types of test polynomials from (24).

References


