Computing 78, 355–367 (2006)
Digital Object Identifier (DOI) 10.1007/s00607-006-0186-y

Computing
Printed in Austria

Implementations of a New Theorem for Computing Bounds for Positive Roots of Polynomials

A. Akritas, Volos, **A. Strzeboński**, Champaign, and **P. Vigklas**, Volos

Received August 24, 2006; revised October 16, 2006

Abstract

Finding an upper bound for the positive roots of univariate polynomials is an important step of the continued fractions real root isolation algorithm. The revived interest in this algorithm has highlighted the need for better estimations of upper bounds of positive roots. In this paper we present a new theorem, based on a generalization of a theorem by D. Stefanescu, and describe several implementations of it – including Cauchy's and Kioustelidis' rules as well as two new rules recently developed by us. From the empirical results presented here we see that applying various implementations of our theorem – and taking the minimum of the computed values – greatly improves the estimation of the upper bound and hopefully that will affect the performance of the continued fractions real root isolation method.

AMS Subject Classifications: 65H05, 68W30, 26C10.

Keywords: Upper bounds, positive roots, polynomial real root isolation.

1. Introduction

We begin by first reviewing some basic facts about the continued fractions method for isolating the positive roots of polynomial equations. This method is based on Vincent's theorem of 1836, [11], which states:

Theorem 1 (Vincent, 1836): *If in a polynomial equation, $p(x)$, with rational coefficients and without multiple roots we perform sequentially replacements of the form*

$$x \leftarrow \alpha_1 + \frac{1}{x}, \quad x \leftarrow \alpha_2 + \frac{1}{x}, \quad x \leftarrow \alpha_3 + \frac{1}{x}, \dots$$

where $\alpha_1 \geq 0$ is a random non negative integer and $\alpha_2, \alpha_3, \dots$ are random positive integers, $\alpha_i > 0$, $i > 1$, then the resulting polynomial either has no sign variations or it has one sign variation. In the last case the equation has exactly one positive root, which is represented by the continued fraction

$$\alpha_1 + \cfrac{1}{\alpha_2 + \cfrac{1}{\alpha_3 + \cfrac{1}{\ddots}}}$$

whereas in the first case there are no positive roots.

Note that if we represent by $\frac{ax+b}{cx+d}$ the continued fraction that leads to a polynomial $f(x) = p(\frac{ax+b}{cx+d})$, with one sign variation, then the single positive root of $f(x)$ – in the interval $(0, \infty)$ – corresponds to that positive root of $p(x)$ which is located in the open interval with endpoints $\frac{b}{d}$ and $\frac{a}{c}$. These endpoints are *not* ordered and are obtained from $\frac{ax+b}{cx+d}$ by replacing x with 0 and ∞, respectively.

Therefore, with Vincent's theorem we can isolate the (positive) roots of a given polynomial $p(x)$. The negative roots are isolated – as suggested by Sturm – after we transform them to positive with the replacement $x \leftarrow -x$ performed on $p(x)$. The requirement that $p(x)$ have no multiple roots does not restrict the generality of the theorem because in the opposite case we first apply square-free factorization and then isolate the roots of each one of the square-free factors.

In 1978, see [1]–[3], it was realized that each partial quotient α_i is the integer part of a real number – i.e., $\alpha_i = \lfloor \alpha_s \rfloor$, where α_s is the smallest positive root of some polynomial $f(x)$ – and, hence, that it can be computed as the lower bound, ℓb, on the values of the positive roots of a polynomial. So assuming that $\ell b = \lfloor \alpha_s \rfloor$ we now set $\alpha_i \leftarrow \ell b, \ell b \geq 1$, and perform the replacement $x \leftarrow x + \ell b, \ell b \geq 1$ – which takes about the same time as the replacement $x \leftarrow x + 1$.

A *lower* bound, ℓb, on the values of the positive roots of a polynomial $f(x)$, of degree n, is found by first computing an *upper* bound, *ub*, on the values of the positive roots of $x^n f(\frac{1}{x})$ and then setting $\ell b = \frac{1}{ub}$. So what is needed is an efficient method for computing upper bounds on the values of (just) the positive roots of polynomial equations[1].

It should be emphasized that bounds on the values of just the *positive* roots of polynomials are scarce in the literature. Since 1978, Cauchy's rule [9] has been used in the continued fractions real root isolation method. Subsequently, Kioustelidis' rule appeared in 1986, [7], but went rather unnoticed [6].

2. The New Theorem

In the literature there are bounds on the absolute values of the roots and bounds on the positive roots of polynomials, [12], [5], [7], [9], [8]. Although of limited use, the most recent addition to the latter type of bounds has been by Stefanescu. Namely, in [10], the following theorem is proved:

Theorem 2 (Stefanescu, 2005): *Let* $p(x) \in R[x]$ *be such that the number of variations of signs of its coefficients is* ***even****. If*

---

[1] With suitable transformations $p(x) \equiv p(-x) = 0$ and $p(x) \equiv x^n p(-\frac{1}{x}) = 0$ one can find the lower $-ub$ and upper $-\frac{1}{ub}$ bounds of the negative roots x_- of $p(x)$, respectively, $-ub \leq x_- \leq -\frac{1}{ub}$.

$$p(x) = c_1x^{d_1} - b_1x^{m_1} + c_2x^{d_2} - b_2x^{m_2} + \ldots + c_kx^{d_k} - b_kx^{m_k} + g(x), \qquad (1)$$

with $g(x) \in R_+[x]$, $c_i > 0$, $b_i > 0$, $d_i > m_i > d_{i+1}$ *for all* i, *the number*

$$B_3(p) = \max\left\{\left(\frac{b_1}{c_1}\right)^{1/(d_1-m_1)}, \ldots, \left(\frac{b_k}{c_k}\right)^{1/(d_k-m_k)}\right\} \qquad (2)$$

is an upper bound for the positive roots of the polynomial p for any ***choice*** *of* $c_1, \ldots, c_k$.

We point out that Stefanescu's theorem introduces the concept of *matching* or *pairing* a positive coefficient with the negative coefficient of a lower-order term.

The following theorem of ours generalizes Theorem 2, in the sense that it applies to polynomials with any number of sign variations. To accomplish this, we introduced the concept of *breaking up* a positive coefficient into several parts to be paired with negative coefficients (of lower-order terms).

Theorem 3: *Let* $p(x)$

$$p(x) = \alpha_n x^n + \alpha_{n-1}x^{n-1} + \ldots + \alpha_0, \quad (\alpha_n > 0) \qquad (3)$$

be a polynomial with real coefficients and let $d(p)$ *and* $t(p)$ *denote the degree and the number of its terms, respectively.*

Moreover, assume that $p(x)$ *can be written as*

$$p(x) = q_1(x) - q_2(x) + q_3(x) - q_4(x) + \ldots + q_{2m-1}(x) - q_{2m}(x) + g(x), \qquad (4)$$

where all the polynomials $q_i(x)$, $i = 1, 2, \ldots, 2m$ *and* $g(x)$ *have only positive coefficients. In addition, assume that for* $i = 1, 2, \ldots, m$ *we have*

$$q_{2i-1}(x) = c_{2i-1,1}x^{e_{2i-1,1}} + \ldots + c_{2i-1,t(q_{2i-1})}x^{e_{2i-1,t(q_{2i-1})}} \qquad (5)$$

and

$$q_{2i}(x) = b_{2i,1}x^{e_{2i,1}} + \ldots + b_{2i,t(q_{2i})}x^{e_{2i,t(q_{2i})}}, \qquad (6)$$

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 have*

$$t(q_{2i-1}) \geq t(q_{2i}), \qquad (7)$$

then an upper bound of the values of the positive roots of $p(x)$ *is given by*

$$ub = \max_{\{i=1,2,\ldots,m\}} \left\{ \left(\frac{b_{2i,1}}{c_{2i-1,1}}\right)^{\frac{1}{e_{2i-1,1}-e_{2i,1}}}, \ldots, \left(\frac{b_{2i,t(q_{2i})}}{c_{2i-1,t(q_{2i})}}\right)^{\frac{1}{e_{2i-1,t(q_{2i})}-e_{2i,t(q_{2i})}}} \right\}, \quad (8)$$

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 have*

$$t(q_{2i-1}) < t(q_{2i}), \quad (9)$$

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 apply the same formula* (8) *given above.*

Proof: Suppose $x > 0$. We have

$$\begin{aligned}
|p(x)| \geq\; & c_{1,1}x^{e_{1,1}} + \ldots + c_{1,t(q_1)}x^{e_{1,t(q_1)}} - b_{2,1}x^{e_{2,1}} - \ldots - b_{2,t(q_2)}x^{e_{2,t(q_2)}} \\
& + \\
& \vdots \\
& + c_{2m-1,1}x^{e_{2m-1,1}} + \ldots + c_{2m-1,t(q_{2m-1})}x^{e_{2m-1,t(q_{2m-1})}} \\
& - b_{2m,1}x^{e_{2m,1}} - \ldots - b_{2m,t(q_{2m})}x^{e_{2m,t(q_{2m})}} + g(x) \\
=\; & x^{e_{2,1}}(c_{1,1}x^{e_{1,1}-e_{2,1}} - b_{2,1}) + \ldots \\
& + x^{e_{2m,t(q_{2m})}}\left(c_{2m-1,t(q_{2m})}x^{e_{2m-1,t(q_{2m})}-e_{2m,t(q_{2m})}} - b_{2m,t(q_{2m})}\right) + g(x)
\end{aligned}$$

which is strictly positive for

$$x > \max_{\{i=1,2,\ldots,m\}} \left\{ \left(\frac{b_{2i,1}}{c_{2i-1,1}}\right)^{\frac{1}{e_{2i-1,1}-e_{2i,1}}}, \ldots, \left(\frac{b_{2i,t(q_{2i})}}{c_{2i-1,t(q_{2i})}}\right)^{\frac{1}{e_{2i-1,t(q_{2i})}-e_{2i,t(q_{2i})}}} \right\}.$$

□

Please note that the partial extension of Theorem 2 presented in [4] does not treat the case $t(q_{2i-1}) < t(q_{2i})$.

Remark 1: *Pairing up positive with negative coefficients and breaking up a positive coefficient into the required number of parts – to match the corresponding number of negative coefficients – are the key ideas of this theorem. In general, formulae analogous to (8) hold for the cases where: (a) we pair coefficients from the non-adjacent polynomials* $q_{2l-1}(x)$ *and* $q_{2i}(x)$, *for* $1 \leq l < i$, *and (b) we break up one or more positive coefficients into several parts to be paired with the negative coefficients of lower-order terms.*

In the following section, we present several implementations of Theorem 3.

3. Algorithmic Implementations of the New Theorem

Theorem 3 is stated in such a way, that it is amenable to several implementations; to wit, the positive-negative coefficient pairing is not unique and can be done in several ways[2].

Moreover, we have quite a latitude in choosing the positive coefficient to be broken up; and once that choice has been made, we can break it up in equal or unequal parts. We explore some of these choices below.

We begin with the most straightforward approach for implementing Theorem 3, which is to first take care of all the cases where $t(q_{2i-1}) < t(q_{2i})$, and then, for all $i = 1, 2, \dots, m$, to pair a positive coefficient of $q_{2i-1}(x)$ with a negative coefficient of $q_{2i}(x)$ – starting with the coefficients $c_{2i-1,1}$ and $b_{2i,1}$ and moving to the right (in non-increasing order of exponents), until the negative coefficients have been exhausted.

Example 1: *Consider the polynomial*

$$p_1(x) = x^9 + 3x^8 + 2x^7 + x^6 - 4x^4 + x^3 - 4x^2 - 3$$

for which we have

$$\begin{aligned} q_1(x) &= x^9 + 3x^8 + 2x^7 + x^6 \\ -q_2(x) &= -4x^4 \\ q_3(x) &= x^3 \\ -q_4(x) &= -4x^2 - 3. \end{aligned}$$

A direct application of Theorem 3 pairs the terms $\{x^9, -4x^4\}$ *of* $q_1(x)$ *and* $q_2(x)$*, and* ignores *the last three terms of* $q_1(x)$*. It then splits the coefficient of* x^3 *into two, say equal parts to account for the two negative terms of* $q_4(x)$ *and forms the pairs* $\{\frac{x^3}{2}, -4x^2\}$ *and* $\{\frac{x^3}{2}, -3\}$*. The resulting upper bound is* 8.

Another way of applying Theorem 3 would be to pair each of the terms of $q_1(x)$ with $-4x^4$ of $q_2(x)$, and pick the minimum; that is, we pick the minimum of the terms $\{x^9, -4x^4\}$, $\{3x^8, -4x^4\}$, $\{2x^7, -4x^4\}$ and $\{x^6, -4x^4\}$, which is $\sqrt[4]{4/3} = 1.07457$. Then, we pair each of the negative terms of $q_4(x)$ with all of the unmatched positive terms of $q_1(x)$ and $q_3(x)$ and pick the minimum. That is, for

---

[2] An example of the worst possible pairing strategy is the rule by Lagrange and MacLaurin, [4], which states: Suppose α_{n-k}, $k \geq 1$, is the first of the negative coefficients of a polynomial $p(x)$, as in (3), then an upper bound on the values of the positive roots of $p(x)$ is given by

$$ub = 1 + \sqrt[k]{\frac{B}{\alpha_n}},$$

where B is the largest absolute value of the negative coefficients of the polynomial $p(x)$.

the term $-4x^2$ we pick the minimum of $\{x^9, -4x^2\}$, $\{2x^7, -4x^2\}$, $\{x^6, -4x^2\}$ and $\{x^3, -4x^2\}$ which is $\sqrt[5]{2} = 1.1487$, whereas for the term -3 we pick the minimum of $\{x^9, -3\}$, $\{x^6, -3\}$ and $\{x^3, -3\}$ which is $\sqrt[9]{3} = 1.12983$. Finally, the bound is the $max\{\sqrt[4]{4/3}, \sqrt[5]{2}, \sqrt[9]{3}\} = 1.1487$.

This last approach is also encountered in [6] and [10]. Although the computed bound is close to the optimal value, the computing time of the method becomes exponential for increasing polynomial degrees and hence it will not be examined by us in the sequel. The implementation methods of Theorem 3 that will be presented here are linear in time and the computed bounds are also close to the optimal value.

In general, we can obtain better bounds if we pair coefficients from non-adjacent polynomials $q_{2l-1}(x)$ and $q_{2i}(x)$, for $1 \le l < i$. The earliest known implementation of this type is Cauchy's rule, which states that

If $p(x)$ is a polynomial as in (3), of degree $n > 0$, with $\alpha_{n-k} < 0$ for at least one k, $1 \le k \le n$, and if λ is the number of negative coefficients, then an upper bound on the values of the positive roots of $p(x)$ is given by

$$ub_1 = \max_{\{1 \le k \le n : \alpha_{n-k} < 0\}} \sqrt[k]{-\frac{\lambda \alpha_{n-k}}{\alpha_n}}.$$

Note that if $\lambda = 0$ there are *no* positive roots. From this we obtain the following:

Definition 1 (Cauchy's "*leading-coefficient*" implementation of Theorem 3): *For a polynomial $p(x)$, as in (3), with λ negative coefficients, Cauchy first breaks up its leading coefficient, α_n, into λ* equal *parts and then pairs each part with the first unmatched negative coefficient.*

So, for *Example 1*, we form the pairs $\{\frac{x^9}{3}, -4x^4\}$, $\{\frac{x^9}{3}, -4x^2\}$ and $\{\frac{x^9}{3}, -3\}$, and obtain as upper bound the value 1.64375. This improvement in the estimation of the bound is due to the fact that the radicals that come into play, namely $\sqrt[5]{12}$, $\sqrt[7]{12}$, and $\sqrt[9]{9}$, (obtained from the pairs mentioned above) are of higher order and hence the numbers computed are smaller.

Closely related to Cauchy's rule is Kioustelidis' bound [7] given by

$$ub_2 = 2 \max_{\{1 \le k \le n : \alpha_{n-k} < 0\}} \sqrt[k]{-\frac{\alpha_{n-k}}{\alpha_n}}.$$

This leads to the following:

Definition 2 (Kioustelidis' "*leading-coefficient*" implementation of Theorem 3): *For a polynomial $p(x)$, as in (3), Kioustelidis matches the coefficient $-\alpha_{n-k}$ of the term $-\alpha_{n-k}x^{n-k}$ in $p(x)$ with $\frac{\alpha_n}{2^k}$, the leading coefficient divided by 2^k.*

Kioustelidis' "*leading-coefficient*" implementation of Theorem 3, differs from that of Cauchy's only in that the leading coefficient is now broken up in *unequal* parts, by dividing it with different powers of 2.

So, for *Example 1* with Kioustelidis' method we form the pairs $\{\frac{x^9}{2^5}, -4x^4\}$, $\{\frac{x^9}{2^7}, -4x^2\}$ and $\{\frac{x^9}{2^9}, -3\}$, and obtain as upper bound the value 2.63902.

We can yet improve the estimation of the upper bound, if we use *Remark 1* and we pair the two negative terms of $q_4(x)$ with the first two (of the three) ignored positive terms of $q_1(x)$. In this way, we obtain an upper bound of 1.31951, which is very close to 1.06815, the maximum positive root of $p_1(x)$. This new improvement is explained by the fact that the radicals $\sqrt[5]{4}$, $\sqrt[6]{4/3}$, and $\sqrt[7]{3/2}$, obtained from the pairs $\{x^9, -4x^4\}$, $\{3x^8, -4x^2\}$ and $\{2x^7, -3\}$, yield even smaller numbers.

Moreover, extensive experimentation confirmed that by pairing coefficients from the non-adjacent polynomials $q_{2l-1}(x)$ and $q_{2i}(x)$ of $p(x)$, where $1 \leq l < i$, we obtain bounds which are the same as, or better than, the bounds obtained by direct implementation of Theorem 3, and in *most* cases better than those obtained by Cauchy's and Kioustelidis' rules.

Therefore, we are lead to the "*first*–λ" implementation of Theorem 3, which is defined as follows:

Definition 3 ("*first*–λ" implementation of Theorem 3): *For a polynomial $p(x)$, as in (4), with λ negative coefficients we first take care of all cases for which $t(q_{2i}) > t(q_{2i-1})$, by breaking up the last coefficient $c_{2i-1,t(q_{2i})}$, of $q_{2i-1}(x)$, into $t(q_{2i}) - t(q_{2i-1}) + 1$* equal *parts. We then pair each of the first λ positive coefficients of $p(x)$, encountered as we move in non-increasing order of exponents, with the first unmatched negative coefficient.*

However, even this approach can lead, in some cases, to an overestimation of the upper bound, as seen in the following example, which highlights the importance of suitable pairing of negative and positive coefficients.

Example 2: *Consider the polynomial*

$$p(x) = x^3 + 10^{100}x^2 - 10^{100}x - 1.$$

Cauchy's "leading-coefficient" *implementation of Theorem 3 forms the pairs* $\{\frac{x^3}{2}, -10^{100}x\}$ *and* $\{\frac{x^3}{2}, -1\}$ *yielding an upper bound of* 1.41421×10^{50}*; Kioustelidis'* "leading-coefficient" *implementation of Theorem 3 forms the pairs* $\{\frac{x^3}{2^2}, -10^{100}x\}$ *and* $\{\frac{x^3}{2^3}, -1\}$ *yielding an upper bound of* 2×10^{50}*; and finally our* "first–λ" *implementation pairs the terms* $\{x^3, -10^{100}x\}$ *and* $\{10^{100}x^2, -1\}$ *yielding an upper bound of* 10^{50}.

A "possible solution" to this problem could be to also scan the positive coefficients backwards (in non-decreasing order of exponents) in which case the pairs $\{10^{100}x^2, -10^{100}x\}$ and $\{x^3, -1\}$ are formed, yielding an upper bound of 1.

From the above discussion, it becomes obvious that in addition to the already presented implementations of Theorem 3 we also need another, different pairing strategy to take care of cases in which these three approaches perform poorly.

However, the "possible solution" outlined above, may well take care of *Example 2*, but it picks coefficients from the adjacent polynomials $q_{2i-1}(x)$ and $q_{2i}(x)$ of $p(x)$, with all the associated weaknesses, mentioned above.

Therefore, we did not pick this "possible solution" as our fourth implementation of Theorem 3. Instead, we chose the "*local-max*" pairing strategy, which is defined as follows:

Definition 4 ("*local-max*" implementation of Theorem 3): *For a polynomial* $p(x)$*, as in* (3)*, the coefficient* $-\alpha_k$ *of the term* $-\alpha_k x^k$ *in* $p(x)$ *– as given in Eq.* (3) *– is paired with the coefficient* $\frac{\alpha_m}{2^t}$*, of the term* $\alpha_m x^m$*, where* α_m *is the largest positive coefficient with* $n \geq m > k$ *and* t *indicates the number of times the coefficient* α_m *has been used.*

Note that our "*local-max*" strategy can pair coefficients of $p(x)$ from the non-adjacent polynomials $q_{2l-1}(x)$ and $q_{2i}(x)$ of $p(x)$, where $1 \leq l < i$, and breaks up positive coefficients also in *unequal* parts. Moreover, binary fractions of *only* the coefficient α_m get paired with each negative coefficient; this process continues until we encounter a greater positive coefficient.

Applying our "*local-max*" approach to *Example* 2 we form the pairs $\{\frac{10^{100}}{2}x^2, -10^{100}x\}$ and $\{\frac{10^{100}}{2^2}x^2, -1\}$, from which we obtain an upper bound of 2. Therefore, we return the value $2 = min\{10^{50}, 2\}$, which is the minimum of our "*first–λ*" and "*local-max*" implementations.

3.1. The Pseudocode

Below we present the pseudocode for the four different implementations of Theorem 3. Cauchy's "*leading-coefficient*" implementation is described in Algorithm 1, lines 1–14, and the output is ub_1. Kioustelidis' "*leading-coefficient*" implementation

```
Input: A univariate polynomial $p(x) = \alpha_n x^n + \alpha_{n-1}x^{n-1} + \ldots + \alpha_0$,   $(\alpha_n > 0)$
Output: An upper bound, $ub_1$, on the values of the positive roots of the polynomial
 1  initializations;
 2  $cl \longleftarrow \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n\}$;
 3  $\lambda \longleftarrow$ the number of negative elements of $cl$;
 4  if $n + 1 <= 1$ or $\lambda = 0$ then return $ub_1 = 0$;
 5  $j = n + 1$;
 7  for $i = 1$ to $n$ do
 9      if $cl(i) < 0$ then
10          $tempub = (\lambda(-cl(i)/cl(j)))^{1/(j-i)}$;
11          if $tempub > ub$ then $ub = tempub$;
12      end
13  end
14  $ub_1 = ub$
```

Algorithm 1: Cauchy's "*leading-coefficient*" implementation of Theorem 3.

```
Input: A univariate polynomial $p(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \ldots + \alpha_0, \quad (\alpha_n > 0)$
Output: An upper bound, $ub_2$, on the values of the positive roots of the polynomial
initializations;
$cl \longleftarrow \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n\}$;
$\lambda \longleftarrow$ the number of negative elements of $cl$;
if $n + 1 <= 1$ or $\lambda = 0$ then return $ub_2 = 0$;
$j = n + 1$;
for $i = 1$ to $n$ do
    if $cl(i) < 0$ then
        $tempub = 2((-cl(i)/cl(j)))^{1/(j-i)}$;
        if $tempub > ub$ then $ub = tempub$;
    end
end
$ub_2 = ub$
```

Algorithm 2: Kioustelidis' "*leading-coefficient*" implementation of Theorem 3.

```
Input: A univariate polynomial $p(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \ldots + \alpha_0, \quad (\alpha_n > 0)$
Output: An upper bound, $ub_3$, on the values of the positive roots of the polynomial
initializations;
$cl \longleftarrow \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n\}$;
$\lambda \longleftarrow$ the number of negative elements of $cl$;
if $n + 1 <= 1$ or $\lambda = 0$ then return $ub_3 = 0$;
$j = n + 1$;
while $j > 1$ do // make sure $t(q_{2i-1}) \geq t(q_{2i})$ holds for all $i$
    while $j > 1$ and $(cl(j) = 0$ or $cl(j) > 0)$ do // compute $t(q_{2i-1})$
        $flag = 0$;
        while $j > 1$ and $cl(j) > 0$ do
            $flag = 1$;
            $posCounter++$;
            $j--$
        end
        if $flag = 1$ then $LastPstvCoef = j + 1$;
        while $j > 1$ and $cl(j) = 0$ do
            $j--$
        end
    end
    if $j = 1$ and $cl(j) > 0$ then $posCounter++$;
    while $j > 1$ and $(cl(j) = 0$ or $cl(j) < 0)$ do // compute $t(q_{2i})$
        while $j > 1$ and $cl(j) < 0$ do
            $negCounter++$;
            $j--$
        end
        while $j > 1$ and $cl(j) = 0$ do
            $j--$
        end
    end
    if $j = 1$ and $cl(j) < 0$ then $negCounter++$;
    if $negCounter > posCounter$ then // replace last coefficient by a list
        $cl(LastPstvCoef) = \{\underbrace{\frac{cl(LastPstvCoef)}{negCounter - posCounter + 1}, \ldots}_{negCounter - posCounter + 1}\}$
    end
    $negCounter = 0$;
    $posCounter = 0$;
end
```

Algorithm 3: The first part of the "*first–λ*" implementation of Theorem 3.

```
$i = j = n + 1$;
while $i > 0$ and $j > 0$ and $\lambda > 0$ do // pair coefficients and process pairs
    while $cl(j) \leq 0$ do
        $j - -$
    end
    if $cl(j)$ is a list element then // $cl(j)$ is a list element
        while ($cl(i) \geq 0$ or $cl(i)$ is a list) and $i > 1$ do
            $i - -$
        end
        $tempub = (-cl(i)/cl(j))^{1/(j-i)}$;
        $\lambda - -$;
        if $tempub > ub$ then $ub = tempub$;
        $i - -$;
        $j - -$;
    end
end
if $cl(j)$ is a list then // $cl(j)$ is a list
    $k$ = the number of elements of $cl(j)$;
    $temp = cl(j, 1)$;
    if $k > \lambda$ then
        $k = \lambda$
    end
    for $\nu = 1$ to $k$ do
        while ($cl(i) \geq 0$ or $cl(i)$ is a list) and $i > 1$ do
            $i - -$
        end
        $tempub = (-cl(i)/temp)^{1/(j-i)}$;
        $\lambda - -$;
        if $tempub > ub$ then $ub = tempub$;
        $i - -$;
    end
    $j - -$;
end
$ub_3 = ub$
```

Algorithm 4: The second part of the "*first–λ*" implementation of Theorem 3.

```
Input: A univariate polynomial $p(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \ldots + \alpha_0, \quad (\alpha_n > 0)$
Output: An upper bound, $ub_4$, on the values of the positive roots of the polynomial
initializations;
$cl \longleftarrow \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n\}$;
if $n + 1 <= 1$ then return $ub_4 = 0$;
$j = n + 1$;
$t = 1$;
for $i = n$ to $1$ step $-1$ do
    if $cl(i) < 0$ then
        $tempub = (2^t(-cl(i)/cl(j)))^{1/(j-i)}$;
        if $tempub > ub$ then $ub = tempub$;
        $t + +$;
    else
        if $cl(i) > cl(j)$ then
            $j = i$;
            $t = 1$
        end
    end
end
$ub_4 = ub$
```

Algorithm 5: The "*local-max*" implementation of of Theorem 3.

Table 1. Bounds for positive roots of various types of polynomials. *MPR* stands for the maximum positive root, computed numerically

		Degrees									
Polynomial	Bounds	10	100	200	300	400	500	600	700	800	900
Laguerre	Cauchy(ub_1)	500	5×10^5	4×10^6	1.35×10^7	3.2×10^7	6.25×10^7	1.08×10^8	1.72×10^8	2.56×10^8	3.65×10^8
	K (ub_2)	200	2×10^4	8×10^4	18×10^4	32×10^4	50×10^4	72×10^4	98×10^4	1.28×10^6	1.62×10^6
	$min(ub_3, ub_4)$	100	1×10^4	4×10^4	9×10^4	16×10^4	25×10^4	36×10^4	49×10^4	64×10^4	81×10^4
	MPR	29.92	374.98	767.82	1162.8	1558.81	1955.44	2352.5	2749.87	3147.48	3545.29
ChebyshevI	Cauchy(ub_1)	2.74	25	50	75	100	125	150	175	200	225
	K (ub_2)	3.16	10	14.14	17.32	20	22.36	24.49	26.46	28.28	30
	$min(ub_3, ub_4)$	1.58	5	7.07	8.66	10	11.18	12.25	13.23	14.14	15
	MPR	0.987688	0.999877	0.999969	0.999986	0.999992	0.999995	0.999997	0.999997	0.999998	0.999998
ChebyshevII	Cauchy(ub_1)	2.60	24.87	49.87	74.87	99.87	124.86	149.88	174.88	199.88	224.88
	K (ub_2)	3	9.95	14.11	17.29	19.98	22.34	24.47	26.44	28.27	29.98
	$min(ub_3, ub_4)$	1.5	4.97	7.05	8.65	9.99	11.17	12.24	13.22	14.13	14.99
	MPR	0.959493	0.999516	0.999878	0.999945	0.999969	0.99998	0.999986	0.99999	0.999992	0.999994
Wilkinson	Cauchy(ub_1)	275	252500	2.01×10^6	6.77×10^6	1.6×10^7	3.13×10^7	5.4×10^7	8.59×10^7	1.28×10^8	1.82×10^8
	K (ub_2)	110	10100	40200	90300	160400	250500	360600	490700	640800	810900
	$min(ub_3, ub_4)$	55	5050	20100	45150	80200	125250	180300	245350	320400	405450
	MPR	10	100	200	300	400	500	600	700	800	900
Mignotte	Cauchy(ub_1)	1.778	1.048	1.024	1.016	1.012	1.009	1.008	1.007	1.006	1.005
	K (ub_2)	3.26	2.081	2.040	2.026	2.020	2.016	2.013	2.011	2.0098	2.0087
	$min(ub_3, ub_4)$	1.63	1.041	1.020	1.013	1.0099	1.0079	1.0066	1.0056	1.0049	1.0044
	MPR	1.5763	1.0362	1.0177	1.0117	1.0088	1.0070	1.0058	1.0050	1.0044	1.0039
uRandom	Cauchy(ub_1)	1892	42535	7.04×10^6	5282.2	9.62×10^7	11801.2	5.25×10^7	17389	17199.7	513.4
	K (ub_2)	1892	1810	135426	2001.73	1.01×10^6	1441.75	373400	1851.05	1746.37	133.8
	$min(ub_3, ub_4)$	946	1810	135426*	4.92*	506494	29.3*	186700	20.4*	3.08*	2.57*
	MPR	944.962	905.528	67721.9	1.40192	506493	13.7921	186698	10.6972	0.998305	1.21821
Random	Cauchy(ub_1)	2.02	52	11.62	156.95	7.15	122.6	258.6	45.8	10.48	993.1
	K (ub_2)	2.23	2.24	2.28	2.04	2.41	2.32	3.49	4.88	2.08	4.54
	$min(ub_3, ub_4)$	3.11*	2.15*	1.4	1.98*	1.68*	2.43*	3.47	2.44	1.82*	4.54*
	MPR	1.1843	1.64514	1.00699	1.22919	1.00248	1.39784	2.69568	1.00576	1.02541	3.39394
usRandom	Cauchy(ub_1)	602.6	17.61	205.1	1.50×10^8	100.4	13574	7.31×10^7	6.28×10^7	2.20×10^8	636.6
	K (ub_2)	602.6	18.61	91.19	2.06×10^6	54.17	1752.4	493872	364264	1.12×10^6	165.9
	$min(ub_3, ub_4)$	1.48*	1.90*	1.73*	1.03163×10^6	1.99*	17.37*	493872*	364264*	557783	1.99*
	MPR	∄(-0.236)	∄(-0.236)	∄(-0.236)	1.03162×10^6	1.20669	9.69017	246938	182136	557782	1.06084
sRandom	Cauchy(ub_1)	13.6	152.5	303.1	458.9	87.2	513	6.03	5.16	18.36	8.65
	K (ub_2)	4.54	5.65	5.56	6.33	2.18	3.95	2.89	2.38	2.00	2.25
	$min(ub_3, ub_4)$	4.54*	5.65*	5.56	3.17	1.61	3.64	1.44	1.67*	1.99*	1.99*
	MPR	2.40372	4.8321	3.5684	2.7936	1.02576	1.01633	1.00183	1.0038	1.01238	1.00061

is described in Algorithm 2, lines 1–14, and the output is ub_2. (These two bounds are presented here for completion.) The "*first–λ*" implementation is described in Algorithms 3 and 4, lines 1–77, and the output is ub_3. The "*local-max*" implementation is described in Algorithm 5, lines 1–20, and the output is ub_4. The final upper bound is $ub = min\{ub_3, ub_4\}$.

4. Experimental Results

In this section, we present some examples using the same classes of polynomials, as in [3] in order to evaluate our new combined implementation, min{"*first–λ*", "*local-max*"}, of Theorem 3 and to compare it with Cauchy's and Kioustelidis' "*leading-coefficient*" implementations.

In Table 1, "uRandom" indicates a random polynomial whose leading coefficient is one, whereas "sRandom" indicates a random polynomial obtained with the randomly chosen seed 1001; the average size of the coefficients ranges from -2^{20} to 2^{20}. Additionally, Kioustelidis' name was shortened to "K" and a "star" indicates that the bound obtained by "*local-max*" was the minimum of the two.

5. Conclusion

From Table 1, we see that Kioustelidis' method is, in general, better (or much better) than that of Cauchy's. This is not surprising given the fact that Kioustelidis breaks up the leading coefficient in *unequal* parts, whereas Cauchy breaks it up in *equal* parts.

Our "*first–λ*" implementation, as the name indicates, uses additional coefficients and, therefore, it is not surprising that it is, in general, better (or much better) than both previous methods. In the few cases where Kioustelidis' method is better than ours, our "*local-max*" method takes again the lead.

Therefore, given their linear cost of execution, we propose that one could safely use only the last two implementations of Theorem 3 in order to obtain the best bounds possible. Certainly, this is worth trying in the continued fractions real root isolation method in order to further improve its performance.

Acknowledgements

We wish to thank the referees for their constructive criticism.

References

[1] Akritas, A.: Vincent's theorem in algebraic manipulation. Ph.D. Thesis, North Carolina State University, Raleigh, NC, 1978.

[2] Akritas, A.: The fastest exact algorithms for the isolation of the real roots of a polynomial equation. Computing *24*, 299–313 (1980).

[3] Akritas, A., Strzebonski, A.: A comparative study of two real root isolation methods. Nonlinear Analysis: Modelling Control *10*(4), 297–304 (2005).

[4] Akritas, A., Vigklas, P.: Comparison of various methods for computing bounds for positive roots of polynomials. Abstracts of the 12th Int. Conf. on Applications of Computer Algebra (Vladimir Gerdt, Margarita Spiridonova and Maria Nisheva-Pavlova, eds.). Varna, Bulgaria 2006.

[5] Demidovich, P., Maron, A.: Computational mathematics. Moscow: Mir Publishers 1987.

[6] Hong, H.: Bounds for absolute positiveness of multivariate polynomials. J. Symb. Comput. *25*(5), 571–585 (1998).

[7] Kioustelidis, B.: Bounds for positive roots of polynomials. J. Comput. Appl. Math. *16*(2), 241–244 (1986).

[8] Mignotte, M.: Mathematics for computer algebra. New York: Springer 1992.

[9] Obreschkoff, N.: Verteilung und Berechnung der Nullstellen reeller Polynome. Berlin: VEB Deutscher Verlag der Wissenschaften 1963.

[10] Stefanescu, D.: New bounds for positive roots of polynomials. J. Univ. Comput. Sci. *11*(12), 2132–2141 (2005).

[11] Vincent A. J. H.: Sur la resolution des équations numériques. J. Math. Pures Appl. *1*, 341–372 (1836).

[12] Yap, C.-K.: Fundamental problems of algorithmic algebra. Oxford University Press 2000.

A. Akritas and P. Vigklas
Department of Computer and
Communication Engineering
37 Glavani and 28th October
38221 Volos
Greece
e-mails: {akritas, pviglas}@uth.gr

A. W. Strzeboński
Wolfram Research, Inc.
100 Trade Center Drive
Champaign, IL 61820-7237
USA
e-mail: adams@wolfram.com