Journal of Universal Computer Science, vol. 15, no. 3 (2009), 523-537
submitted: 5/9/08, accepted: 28/1/09, appeared: 1/2/09 © J.UCS

Linear and Quadratic Complexity Bounds on the Values of the Positive Roots of Polynomials

Alkiviadis G. Akritas
(Department of Computer and Communication Engineering
University of Thessaly
Volos, Greece
akritas@uth.gr)

Dedicated to Professor Doru Ştefănescu[1]

Abstract: In this paper we review the existing *linear* and *quadratic* complexity (upper) bounds on the values of the positive roots of polynomials and their impact on the performance of the Vincent-Akritas-Strzeboński (`VAS`) continued fractions method for the isolation of real roots of polynomials. We first present the following four linear complexity bounds (two "old" and two "*new*" ones, respectively): *C*auchy's, (C), *K*ioustelidis', (K), *F*irst-*L*ambda, (FL) and *L*ocal-*M*ax, (LM); we then state the quadratic complexity extensions of these four bounds, namely: CQ, KQ, FLQ, and LMQ — the second, (KQ), having being presented by Hong back in 1998. *All* eight bounds are derived from Theorem 5 below. The estimates computed by the quadratic complexity bounds are less than or equal to those computed by their linear complexity counterparts. Moreover, it turns out that `VAS(lmq)` — the `VAS` method implementing LMQ — is 40% faster than the original version `VAS(cauchy)`.
Key Words: upper bounds, positive roots, real root isolation, Vincent's theorem
Category: F.2.1, G.1.5

1 Introduction

Computing (lower) bounds on the values of the positive roots of polynomials is a crucial operation in the `VAS` continued fractions method for the isolation of the real roots of polynomials. Therefore, we begin by reviewing some basic facts about this method, which is based on Vincent's theorem, [Vincent, 1836]:

Theorem 1. *(Vincent's original theorem — continued fractions version, 1836) If in a polynomial, $p(x)$, of degree n, with rational coefficients and without multiple roots we perform sequentially replacements of the form*

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

where $\alpha_1 \geq 0$ is an arbitrary non negative integer and $\alpha_2, \alpha_3, \dots$ are arbitrary positive integers, $\alpha_i > 0$, $i > 1$, then the resulting polynomial either has no sign

[1] For his inspiring work on bounds of positive roots of polynomials.

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

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

whereas in the first case there are no positive roots.

[Alesina & Galuzzi, 1998], [Alesina & Galuzzi, 1999], [Alesina & Galuzzi, 2000], are a *must* for a complete historical survey of the subject, whereas for implementation details of this theorem in the process of real root isolation see the papers by Akritas, Strzeboński and Vigklas [Akritas 1978], [Akritas 1980], [Akritas & Strzeboński, 2005], [Akritas, Strzeboński & Vigklas, 2008] and Chapter 7 in [Akritas 1989]. The thing to note is that the quantities α_i (the partial quotients of the continued fraction) are computed by *repeated* application of a method for estimating *lower* bounds on the values of the positive roots of a polynomial.

Therefore, the efficiency of the `VAS` continued fractions method depends on how good these estimates are.

*C*auchy's, (C), (linear complexity) bound on the values of the positive roots of a polynomial, was used until recently in the `VAS` continued fractions real root isolation method, [Akritas & Strzeboński, 2005]. In the SYNAPS implementation of the `VAS`, Tsigaridas and Emiris, [Tsigaridas & Emiris, 2006], used *K*ioustelidis', (K), (linear complexity) bound, [Kioustelidis, 1986], and independently verified the results obtained in [Akritas & Strzeboński, 2005][2].

In 2006 a new theorem, Theorem 5, was discovered, [Akritas & Vigklas, 2007], [Akritas, Strzeboński & Vigklas, 2006], which extended and generalized a theorem by Ştefănescu of 2005, [Ştefănescu, 2005], in such a way that both Cauchy's and Kioustelidis' linear complexity bounds on the values of the positive roots of a polynomial are derived from it; moreover, based on Theorem 5, two new linear complexity bounds were developed: *F*irst-*L*ambda, (FL) and *L*ocal-*M*ax, (LM).

Recently, the linear complexity bounds were extended and their corresponding *new* quadratic complexity bounds on the values of the positive roots of polynomials were developed: *C*auchy's *Q*uadratic, (CQ), *K*ioustelidis' *Q*uadratic, (KQ, [Hoon, 1998]), *F*irst-*L*ambda *Q*uadratic, (FLQ), and *L*ocal-*M*ax *Q*uadratic (LMQ). All four quadratic complexity bounds are also derived from Theorem 5.

The rest of the paper is structured as follows:

In section 2 we present Ştefănescu's theorem of 2005, and Theorem 5, from which *all* bounds mentioned above are derived.

[2] See also Sharma's work, [Sharma, 2007a] and [Sharma, 2007b], where he used the worst possible positive lower bound to prove that the `VAS` method is still polynomial in time!

In section 3 we present the four linear complexity bounds: *C*auchy's, *K*ioustelidis', *F*irst-*L*ambda, and *L*ocal-*M*ax. These can be found elsewhere, see the paper by Akritas, Strzeboński & Vigklas [Akritas, Strzeboński & Vigklas, 2006], but are included here for completion. As it is impossible for any single bound to *always* compute the best estimates, taking the minimum of the last two, *FL+LM*, results in the best linear complexity bound.

In section 4 we present the four quadratic complexity bounds: *CQ*, *KQ*, *FLQ*, and *LMQ*. Here, it is impossible to tell which of *FLQ*, *LMQ* and *FLQ+LMQ* is the best; for theoretical reasons *LMQ* was chosen.

And finally there is the conclusions section.

2 Theoretical Background

As was pointed out in the introduction, the efficiency of the `VAS` continued fractions method depends heavily on how good are the estimates of the lower bounds on the values of the positive roots.

A *lower* bound, ℓb, on the values of the positive roots of a polynomial $p(x)$, of degree n, is found by first computing an *upper* bound, ub, on the values of the positive roots of $x^n p(\frac{1}{x})$ and then setting $\ell b = \frac{1}{ub}$.

So, clearly, what is needed is an efficient method for computing upper bounds on the values of (just) the positive roots of polynomial equations.

In the initial implementation of `VAS`, in 1978, the lower bounds were computed using a theorem by Cauchy, [Obreschkoff, 1963].

Theorem 2. *(Cauchy's theorem) Let*

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

be a polynomial with real coefficients $\alpha_n, \alpha_{n-1}, \ldots, \alpha_0$*, of degree* $n > 0$*, and with* $\alpha_{n-k} < 0$ *for at least one* k*,* $1 \leq k \leq n$*. 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*

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

Note that if $\lambda = 0$ *there are* no *positive roots.*

Subsequently, Kioustelidis' bound appeared, [Kioustelidis, 1986], and was used in the SYNAPS implementation of `VAS` by Tsigaridas and Emiris in 2006, [Tsigaridas & Emiris, 2006]. Kioustelidis' theorem is closely related to the one by Cauchy:

Theorem 3. *(Kioustelidis' theorem, 1986) Let $p(x)$ be a polynomial as in Eq. (1), of degree $n > 0$, with $\alpha_{n-k} < 0$ for at least one k, $1 \leq k \leq n$. Then an upper bound on the values of the positive roots of $p(x)$ is given by*

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

However, both implementations of the `VAS` continued fractions method — that is, using either Cauchy's, [Akritas & Strzeboński, 2005], or Kioustelidis' bound, [Tsigaridas & Emiris, 2006] — showed that its "Achilles' heel" was the case of very many very large rational roots. In this case — as can be seen from Table 1 presented below — the `VAS` method was up to 4 times slower than `VCA(rel)` — the fastest implementation of the `VCA` bisection method developed by Rouillier and Zimmermann, [Rouillier and Zimmermann, 2004]. (Table 1 corresponds to the last table (Table 4), found in [Akritas & Strzeboński, 2005].)

Table 1: Products of factors (x-randomly generated integer root). All computations were done on a 850 MHz Athlon PC with 256 MB RAM; (s) stands for time in seconds and (MB) for the amount of memory used, in MBytes.

Roots (bit length)	Degree	No. of roots	`VAS` t(s)/M(MB)	`VCA(rel)` t(s)/M(MB)
10	100	100	0.8/1.82	0.61/1.92
10	200	200	2.45/2.07	10.1/2.64
10	500	500	33.9/3.34	878/8.4
1000	20	20	0.12/1.88	0.044/1.83
1000	50	50	16.7/3.18	4.27/2.86
1000	100	100	550/8.9	133/6.49

The last three lines of Table 1 demonstrate the weaker performance of `VAS` in the case of very many very large rational roots.

To see if the performance of `VAS` could be improved, we needed to better understand the nature of these bounds, and this was achieved with the help of Ştefănescu's theorem of 2005, [Ştefănescu, 2005].

Theorem 4. *(Ştefănescu's theorem, 2005) Let $p(x) \in R[x]$ be such that the number of variations of signs of its coefficients is* ***even****. If*

$$p(x) = c_1 x^{d_1} - b_1 x^{m_1} + c_2 x^{d_2} - b_2 x^{m_2} + \ldots + c_k x^{d_k} - b_k x^{m_k} + g(x),$$

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

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

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

Ştefănescu's theorem introduces the concept of *matching* or *pairing* a positive coefficient with an unmatched negative coefficient of a lower order term; however, Ştefănescu's theorem worked *only* for polynomials with an even number of sign variations.

Note: More precisely, it is the *term* with the positive coefficient that is being matched to the *term* with the negative coefficient.

Ştefănescu's theorem was generalized in the sense that Theorem 5 below applies to polynomials with any number of sign variations, [Akritas & Vigklas, 2007]. To accomplish this, the concept was introduced of *breaking up* a positive coefficient into several parts to be paired with negative coefficients of lower order terms[3], [Akritas, Strzeboński & Vigklas, 2006].

Theorem 5. *(Akritas-Strzeboński-Vigklas, 2006) Let* $p(x)$

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

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*

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

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*

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

and

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

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 have*

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

[3] After the extension by Akritas, Strzeboński and Vigklas, [Akritas, Strzeboński & Vigklas, 2006], Ştefănescu also extended his Theorem 4, [Ştefănescu, 2007].

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

$$ub = \max_{\{i=1,2,\ldots,m\}} \left\{ \left(\frac{b_{2i,1}}{c_{2i-1,1}}\right)^{\frac{1}{e_{2i-1,1}-e_{2i,1}}}, \ldots, \left(\frac{b_{2i,t(q_{2i})}}{c_{2i-1,t(q_{2i})}}\right)^{\frac{1}{e_{2i-1,t(q_{2i})}-e_{2i,t(q_{2i})}}} \right\}, \quad (3)$$

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

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

*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 (3) given above.*

For a proof of this theorem see the paper by Akritas, Strzeboński & Vigklas [Akritas, Strzeboński & Vigklas, 2006]. Please note that the partial extension of Theorem 4 presented in [Akritas & Vigklas, 2007] does not treat the case $t(q_{2i-1}) < t(q_{2i})$.

Crucial Observation. 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 (3) 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.

3 Linear Complexity Bounds Derived from Theorem 5

The bounds in the literature, such as Cauchy's and Kioustelidis', are of *linear* complexity.

The General Idea of the Linear Complexity Bounds: These bounds are computed as follows:

- *each* negative coefficient of the polynomial is paired with *one* of the preceding *unmatched* positive coefficients;
- the maximum of all the computed radicals is taken as the estimate of the bound.

Using Theorem 5 we obtain the following interpretation of Cauchy's and Kioustelidis' theorems:

C. Cauchy's "leading-coefficient" implementation of Theorem 5. For a polynomial $p(x)$, as in Eq. (1), with λ negative coefficients, Cauchy's method first breaks up its leading coefficient, α_n, into λ *equal* parts and then pairs each part with the first unmatched negative coefficient. That is, we have:

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

or, equivalently,

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

K. Kioustelidis' "leading-coefficient" implementation of Theorem 5. For a polynomial $p(x)$, as in Eq. (1), Kioustelidis' method 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.

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

or, equivalently,

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

Kioustelidis' "leading-coefficient" implementation of Theorem 5, 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, [Kioustelidis, 1986].

Using Theorem 5 a new linear complexity method, *first*–λ, was developed for computing upper bounds on the values of the positive roots of polynomials.

FL. "**first**–λ" implementation of Theorem 5. For a polynomial $p(x)$, as in (2), 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.

This is an improvement over the other two bounds by Cauchy and Kioustelidis, but as the following Example demonstrates, all three methods can fail miserably.

Example: Consider the polynomial

$$x^3 + 10^{100}x^2 - 10^{100}x - 1,$$

which has one sign variation and, hence, only one positive root = 1.

- For Cauchy's theorem we pair the terms $\{\frac{x^3}{2}, -10^{100}x\}$ and $\{\frac{x^3}{2}, -1\}$, and taking the maximum of the radicals computed (as in Ştefănescu's theorem, Theorem 4) we obtain a bound estimate of $1.41421 * 10^{50}$.
- Likewise, for Kioustelidis' theorem we pair the terms $\{\frac{x^3}{2^2}, -10^{100}x\}$ and $\{\frac{x^3}{2^3}, -1\}$, and obtain a bound estimate of $2 * 10^{50}$.
- Similarly, for *first*–λ we pair the terms $\{x^3, -10^{100}x\}$ and $\{10^{100}x^2, -1\}$, and obtain a bound estimate of 10^{50}.

To correct this inadequacy yet another new linear complexity method, *local-max*, was developed for computing an upper bound on the values of the positive roots of polynomials:

LM. "**local-max**" implementation of Theorem 5. For a polynomial $p(x)$, as in (1), the coefficient $-\alpha_k$ of the term $-\alpha_k x^k$ in $p(x)$ — as given in Eq. (1) — 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.

Example, continued: For *local-max* we pair the terms $\{\frac{10^{100}x^2}{2}, -10^{100}x\}$ and $\{\frac{10^{100}x^2}{2^2}, -1\}$, and taking the maximum of the radicals computed we obtain a bound estimate of 2.

All four linear complexity bounds mentioned above have been tested extensively — on various classes of specific and random polynomials — and the following is a summary of the findings, [Akritas, Strzeboński & Vigklas, 2006], [Akritas & Vigklas, 2007]:

- Kioustelidis' bound is, in general, better (or much better) than Cauchy's; this happens because the former breaks up the leading coefficient in *unequal* parts, whereas the latter breaks it up in *equal* parts.
- the *first*-λ bound, 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 bounds. In the few cases where Kioustelidis' bound is better than *first*-λ, the *local-max* bound takes again the lead.

Therefore, given their linear cost of execution, $min(FL, LM)$ or $FL + LM$ is the best among the linear complexity bounds on values of the positive roots of a polynomial, [Akritas, Strzeboński & Vigklas, 2006].

In Table 2 below we recalculate the results of Table 1, and compare the timings in seconds, $t(s)$, for: (a) `VAS(cauchy)`, the `VAS` continued fractions method using Cauchy's rule (the "*old*" method), (b) `VAS(fl+lm)`, the `VAS` continued fractions method using $min(FL, LM)$ or $FL + LM$ (the "*new*" method), and (c) `VCA(rel)`, the fastest implementation of the `VCA` bisection method. (Table 2 corresponds to the last table (Table 2), found in the paper by Akritas, Strzeboński & Vigklas [Akritas, Strzeboński & Vigklas, 2007].)

Table 2: Products of terms $x - r$ with random integer r. The tests were run on a laptop computer with 1.8 Ghz Pentium M processor, running a Linux virtual machine with 1.78 GB of RAM.

Roots (bit length)	Deg	`VAS(cauchy)` t(s) Average (Min/Max)	`VAS(fl+lm)` t(s) Average (Min/Max)	`VCA(rel)` t(s) Average (Min/Max)
10	100	0.314 (0.248/0.392)	0.253 (0.228/0.280)	0.346 (0.308/0.384)
10	200	1.74 (1.42/2.33)	1.51 (1.34/1.66)	3.90 (3.72/4.05)
10	500	17.6 (16.9/18/7)	17.4 (16.3/18.1)	129 (122/140)
1000	20	0.066 (0.040/0.084)	0.031 (0.024/0.040)	0.038 (0.028/0.044)
1000	50	1.96 (1.45/2.44)	0.633 (0.512/0.840)	1.03 (0.916/1.27)
1000	100	52.3 (36.7/81.3)	12.7 (11.3/14.6)	17.2 (16.1/18.7)

Due to the different computational environment the times $t(s)$ differ substantially, but they confirm the fact that `VAS(fl+lm)` is now *always* faster than `VCA(rel)`.

Again, of interest are the last three lines of Table 2, where as in Table 1 the performance of `VAS(cauchy)` is worse than `VCA(rel)` — at worst 3 times slower, as the last entry indicates. However, from these same lines of Table 2 we observe that `VAS(fl+lm)` is now always faster than `VCA(rel)` — at best twice as fast, as seen in the 5-th line.

When the times of `VAS(cauchy)` were compared with those of `VAS(fl+lm)` — on various classes of specific and random polynomials — not only was an overall speed-up of 15% observed, [Akritas, Strzeboński & Vigklas, 2008], but `VAS(fl+lm)` also became always faster than the Vincent-Collins-Akritas[4] bisection method (`VCA`), see the papers [Akritas, Strzeboński & Vigklas, 2007], [Akritas, Strzeboński & Vigklas, 2008], [Boulier, 2007], [Collins & Akritas, 1976].

[4] Misleadingly referred to in the literature as "modified Uspensky's" or "Descartes"' method.

4 Quadratic Complexity Bounds Derived from Theorem 5

To further improve the performance of the VAS continued fractions method it was decided to use quadratic complexity bounds on the values of the positive real roots hoping that their improved estimates *should* compensate for the extra time needed to compute them. These bounds are based on the following idea:

The General Idea of the Quadratic Complexity Bounds: These bounds are computed as follows:

- *each* negative coefficient of the polynomial is paired with *all* the preceding positive coefficients and the minimum of the computed values is taken;
- the maximum of all those minimums is taken as the estimate of the bound.

In general, the estimates obtained from the quadratic complexity bounds are less than or equal to those obtained from the corresponding linear complexity bounds, as the former are computed after much greater effort and time[5]. The quadratic complexity bounds described below are all extensions of their linear complexity counterparts.

Thus, we have:

CQ. Cauchy's Quadratic complexity implementation of Theorem 5. For a polynomial $p(x)$, as in Eq. (1), each negative coefficient $a_i < 0$ is "paired" with *each* one of the preceding positive coefficients a_j divided by λ_i — that is, *each* positive coefficient a_j is "broken up" into *equal* parts, as is done with *just* the leading coefficient in Cauchy's bound; λ_i is the number of negative coefficients to the right of, and including, a_i — and the minimum is taken over all j; subsequently, the maximum is taken over all i.

That is, we have:

$$ub_{CQ} = \max_{\{a_i<0\}} \min_{\{a_j>0: j>i\}} \sqrt[j-i]{-\frac{a_i}{\frac{a_j}{\lambda_i}}}.$$

Example, continued: For CQ we first compute

- the minimum of the two radicals obtained from the pairs of terms $\{\frac{x^3}{2}, -10^{100}x\}$ and $\{\frac{10^{100}x^2}{2}, -10^{100}x\}$ which is 2,

[5] It should be noted that time is not so importance in our case, since — as can be seen in line 4 of the description of the VAS algorithm, [Akritas, Strzeboński & Vigklas, 2008] — these bounds are estimated *before* a translation of complexity at *least* $O(n^2)$ is executed.

- the minimum of the two radicals obtained from the pairs of terms $\{x^3, -1\}$ and $\{10^{100}x^2, -1\}$ which is $\frac{1}{10^{50}}$,

and we then obtain as a bound estimate the value $max\{2, \frac{1}{10^{50}}\} = 2$.

KQ. Kioustelidis' Quadratic complexity implementation of Theorem 5. For a polynomial $p(x)$, as in Eq. (1), each negative coefficient $a_i < 0$ is "paired" with *each* one of the preceding positive coefficients a_j divided by 2^{j-i} — that is, *each* positive coefficient a_j is "broken up" into *unequal* parts, as is done with *just* the leading coefficient in Kioustelidis' bound — and the minimum is taken over all j; subsequently, the maximum is taken over all i.

That is, we have:

$$ub_{KQ} = 2 \max_{\{a_i<0\}} \min_{\{a_j>0:j>i\}} \sqrt[j-i]{-\frac{a_i}{a_j}},$$

or, equivalently,

$$ub_{KQ} = \max_{\{a_i<0\}} \min_{\{a_j>0:j>i\}} \sqrt[j-i]{-\frac{a_i}{\frac{a_j}{2^{j-i}}}}.$$

Example, continued: For KQ we first compute

- the minimum of the two radicals obtained from the pairs of terms $\{\frac{x^3}{2^2}, -10^{100}x\}$ and $\{\frac{10^{100}x^2}{2}, -10^{100}x\}$ which is 2,
- the minimum of the two radicals obtained from the pairs of terms $\{\frac{x^3}{2^3}, -1\}$ and $\{\frac{10^{100}x^2}{2^2}, -1\}$ which is $\frac{2}{10^{50}}$,

and we then obtain as a bound estimate the value $max\{2, \frac{2}{10^{50}}\} = 2$.

FLQ. "First-Lambda" Quadratic complexity implementation of Theorem 5. For a polynomial $p(x)$, as in (2), with λ negative coefficients we first take care of all cases for which $t(q_{2\ell}) > t(q_{2\ell-1})$, by breaking up the last coefficient $c_{2\ell-1,t(q_{2\ell})}$, of $q_{2\ell-1}(x)$, into $d_{2\ell-1,t(q_{2\ell})} = t(q_{2\ell}) - t(q_{2\ell-1}) + 1$ *equal* parts. Then each negative coefficient $a_i < 0$ is "paired" with *each* one of the preceding $min(i, \lambda)$ positive coefficients a_j divided by d_j — that is, *each* of the preceding $min(i, \lambda)$ positive coefficient a_j is "broken up" into d_j *equal* parts, where d_j is initially set to 1 and its value changes *only* if the positive coefficient a_j is broken up into equal parts, as stated in Theorem 5; $u(j)$ indicates the number of times a_j can be used to calculate the minimum, it is originally set equal to d_j and its value decreases each time a_j is used in the computation of the minimum — and the minimum is taken over all j; subsequently, the maximum is taken over all i.

That is, we have:

$$ub_{FLQ} = \max_{\{a_i<0\}} \min_{\{a_j>0: j>min(i,\lambda): u(j)\neq 0\}} \sqrt[j-i]{-\frac{a_i}{\frac{a_j}{d_j}}}.$$

From the above descriptions it is clear that FLQ tests *just* the first $min(\iota, \lambda)$ positive coefficients, whereas *all* the other quadratic complexity bounds test *every* preceding positive coefficient. Hence, FLQ is faster (or quite faster) than all of them.

Example, continued: For FLQ we first compute

– the minimum of the two radicals obtained from the pairs of terms $\{x^3, -10^{100}x\}$ and $\{10^{100}x^2, -10^{100}x\}$ which is 1 — evaluated from the second pair of terms,

– the radical obtained from the pair of terms $\{x^3, -1\}$ which is 1,

and we then obtain as a bound estimate the value $max\{1, 1\} = 1$. Note that once a term with a positive coefficient has been used in obtaining the minimum, it cannot be used again!

LMQ. "Local-Max" Quadratic complexity implementation of Theorem 5. For a polynomial $p(x)$, as in (1), each negative coefficient $a_i < 0$ is "paired" with *each* one of the preceding positive coefficients a_j divided by 2^{t_j} — that is, *each* positive coefficient a_j is "broken up" into *unequal* parts, as is done with *just* the locally maximum coefficient in the local max bound; t_j is initially set to 1 and is incremented each time the positive coefficient a_j is used — and the minimum is taken over all j; subsequently, the maximum is taken over all i.

That is, we have:

$$ub_{LMQ} = \max_{\{a_i<0\}} \min_{\{a_j>0: j>i\}} \sqrt[j-i]{-\frac{a_i}{\frac{a_j}{2^{t_j}}}}.$$

Since $2^{t_j} \leq 2^{j-i}$ — where i and j are the indices realizing the max of min; equality holds when there are *no* missing terms in the polynomial — it is clear that the estimates computed by LMQ are sharper by the factor $2^{\frac{j-i-t_j}{j-i}}$ than those computed by Kioustelidis' KQ.

Example, continued: For LMQ we first compute

– the minimum of the two radicals obtained from the pairs of terms $\{\frac{x^3}{2}, -10^{100}x\}$ and $\{\frac{10^{100}x^2}{2}, -10^{100}x\}$ which is 2,

– the minimum of the two radicals obtained from the pairs of terms $\{\frac{x^3}{2^2}, -1\}$ and $\{\frac{10^{100}x^2}{2^2}, -1\}$ which is $\frac{2}{10^{50}}$,

and we then obtain as a bound estimate the value $max\{2, \frac{2}{10^{50}}\} = 2$.

Notice how the estimates of all quadratic complexity bounds are much better than those of their linear complexity counterparts. Extensive experimentation revealed that the bounds FLQ, LMQ and $min(FLQ, LMQ)$ behave the same when implemented in the `VAS` continued fractions method for the isolation of real roots of polynomials, [Akritas, Argyris & Strzeboński, 2008]. Therefore, for theoretical reasons, it was decided to use LMQ. It turns out that `VAS(lmq)` — the `VAS` method implementing LMQ — is 40% faster than the original version `VAS(cauchy)`, [Akritas, Strzeboński & Vigklas, 2008].

We finally present Table 3 — corresponding to Table 8 in the paper by Akritas, Strzeboński & Vigklas [Akritas, Strzeboński & Vigklas, 2008] — where we demonstrate the performance of `VAS` using quadratic complexity bounds. This is actually the only case where the best linear complexity bound $FL + LM$ is slightly better than LMQ.

Table 3: Products of terms $x - r$ with random integer r. The average speed-up for this table is about 35%.

Bit-length of roots	Degree	`VAS(cauchy)` t(s) Avg(Min/Max)	`VAS(fl+lm)` t(s) Avg(Min/Max)	`VAS(lmq)` t(s) Avg(Min/Max)
10	100	0.46 (0.28/0.94)	0.24 (0.18/0.28)	0.34 (0.30/0.41)
10	200	1.46 (1.24/1.85)	1.40 (1.28/1.69)	1.40 (1.20/1.69)
10	500	18.1 (16.5/18.9)	18.1 (16.6/18.8)	22.1 (18.7/24.2)
1000	20	0.07 (0.04/0.14)	0.02 (0.02/0.03)	0.03 (0.02/0.04)
1000	50	3.69 (2.38/6.26)	0.81 (0.60/1.28)	0.81 (0.52/1.11)
1000	100	47.8 (37.6/56.9)	13.8 (10.3/19.2)	15.8 (11.3/21.3)

5 Conclusions

Linear complexity bounds are in general inferior to the quadratic complexity ones, both in the computed estimate and when implemented in the `VAS` continued fractions method for real root isolation method.

Quadratic complexity bounds, when implemented in the `VAS` real root isolation method, speed up its performance by an average overall factor of 40%.

References

[Akritas 1978] Akritas, A. G.: *Vincent's Theorem in Algebraic Manipulation*. Ph.D. Thesis, Department of Operations Research, North Carolina State University, Raleigh, NC, (1978).

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

[Akritas 1989] Akritas, A. G.: *Elements of Computer Algebra with Applications*. John Wiley Interscience, New York, (1989).

[Akritas, Argyris & Strzeboński, 2008] Akritas, A. G., Argyris, A. I. Strzeboński, A. W.: "FLQ, the Fastest Quadratic Complexity Bound on the Values of Positive Roots of Polynomials."; Serdica Journal of Computing **2**, (2008), 145–162.

[Akritas & Strzeboński, 2005] Akritas, A. G., Strzeboński, A. W.: "A comparative study of two real root isolation methods". *Nonlinear Analysis: Modelling and Control*, **10**(4), (2005), 297–304.

[Akritas, Strzeboński & Vigklas, 2006] Akritas, A. G., Strzeboński, A. W., Vigklas, P.: "Implementations of a New Theorem for Computing Bounds for Positive Roots of Polynomials"; *Computing*, **78**, (2006), 355–367.

[Akritas, Strzeboński & Vigklas, 2007] Akritas, A. G., Strzeboński, A. W., Vigklas, P.: "Advances on the Continued Fractions Method Using Better Estimations of Positive Root Bounds". In: V. G. Ganzha, E. W. Mayr and E. V. Vorozhtsov (Editors): *Proceedings of the 10th International Workshop on Computer Algebra in Scientific Computing*, CASC 2007, pp. 24 – 30, Bonn, Germany, September 16-20, 2007. LNCS 4770, Springer Verlag, Berlin, Heidelberg.

[Akritas, Strzeboński & Vigklas, 2008] Akritas, A. G., Strzeboński, A. W., Vigklas, P.: "Improving the Performance of the Continued Fractions Method Using New Bounds of Positive Roots."; Nonlinear Analysis: Modelling and Control, **13**, (2008), 265–279.

[Akritas & Vigklas, 2007] Akritas, A. G., Vigklas, P.: "A Comparison of Various Methods for Computing Bounds for Positive Roots of Polynomials"; *Journal of Universal Computer Science*, **13**(4), (2007), 455–467

[Alesina & Galuzzi, 1998] Alesina, A., Galuzzi, M.: "A new proof of Vincent's theorem"; *L'Enseignement Mathemathique*, **44**, (1998), 219–256.

[Alesina & Galuzzi, 1999] Alesina, A., Galuzzi, M.: Addentum to the paper "A new proof of Vincent's theorem"; *L'Enseignement Mathemathique*, 45, (1999), 379–380.

[Alesina & Galuzzi, 2000] Alesina, A., Galuzzi, M.: "Vincent's Theorem from a Modern Point of View"; (Betti, R. and Lawvere W.F. (eds.)), Categorical Studies in Italy 2000, Rendiconti del Circolo Matematico di Palermo, Serie II, n. 64, (2000), 179–191.

[Boulier, 2007] Boulier, F.: *Systèmes polynomiaux : que signifie "résoudre"?*. Université Lille 1, UE INFO 251, (2007)

[Collins & Akritas, 1976] Collins, G. E. & Akritas, A. G.: "Polynomial real root isolation using Descartes' rule of signs". In: *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computations*, Yorktown Heights, N.Y., (1976), pp. 272–275.

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

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

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

[Rouillier and Zimmermann, 2004] Rouillier, F. & Zimmermann, P.: "Efficient isolation of polynomial's real roots"; *Journal of Computational and Applied Mathematics*, **162**, (2004), pp. 33–50.

[Sharma, 2007a] Sharma, V.: "Complexity of Real Root Isolation Using Continued Fractions"; In: *ISAAC07* preprint, 2007.

[Sharma, 2007b] Sharma, V.: *Complexity Analysis of Algorithms in Algebraic Computation*; Ph.D. Thesis, Department of Computer Sciences, Courant Institute of Mathematical Sciences, New York University, 2007.

[Ştefănescu, 2005] Ştefănescu, D.:"New bounds for positive roots of polynomials"; *Journal of Universal Computer Science*, **11**(12), (2005), 2132–2141.

[Ştefănescu, 2007] Ştefănescu, D.:"Bounds for Real Roots and Applications to Orthogonal Polynomials". In: V. G. Ganzha, E. W. Mayr and E. V. Vorozhtsov (Editors): *Proceedings of the 10th International Workshop on Computer Algebra in Scientific Computing*, CASC 2007, pp. 377 – 391, Bonn, Germany, September 16-20, 2007. LNCS 4770, Springer Verlag, Berlin, Heidelberg.

[Tsigaridas & Emiris, 2006] Tsigaridas, E. P., Emiris, I. Z.: "Univariate polynomial real root isolation: Continued fractions revisited". In: Y. Azar and T. Erlebach (Eds.), ESA 2006, LNCS 4168, (2006), 817–828.

[Vincent, 1836] Vincent, A. J. H.:"Sur la resolution des équations numériques"; *Journal de Mathématiques Pures et Appliquées*, **1**, (1836), 341–372.