Записки научных
семинаров ПОМИ
Том 373, 2009 г.

A. G. Akritas

VINCENT'S THEOREM OF 1836: OVERVIEW AND FUTURE RESEARCH

Abstract. In this paper, we present the two different versions of Vincent's theorem of 1836 and discuss the various real root isolation methods derived from them: one using continued fractions and two using bisections – the former being the fastest real root isolation method. Regarding the Continued Fractions method we first show how — using a recently developed quadratic complexity bound on the values of the positive roots of polynomials – its performance has been improved by an average of 40%, over its initial implementation, and then we indicate directions for future research.

1. Introduction

Isolation of the real roots of a polynomial is the process of finding real disjoint intervals such that each contains one real root and every real root is contained in some interval.

Since the beginning of the 19th century, and according to the French "school" of mathematics, isolation has been considered the first step in finding the real roots of a polynomial equation – the second step being the approximation of the roots to any degree of accuracy.

Sturm was the first mathematician to present a theorem, in 1829, for isolating the real roots of a polynomial using bisection, [6]. His theorem has been widely used until 1980, when it was replaced – in the major computer algebra systems – by versions of Vincent's theorem.

Vincent's theorem of 1836, [44], has a very interesting and exciting history [11, 12, 20–22] and [31]. It was almost totally forgotten until it was rediscovered, in 1976, by the author and formed the basis of his Ph.D. Thesis, [1]. Subsequently scientists from all over the world made their own

Key words and phrases. root isolation, continuous fractions, complexity, Vincent's theorem.

This paper is based on the *plenary talk* the author gave at ACA 2008, the International Conference on Applications of Computer Algebra, held at RISC-Linz, Hagenberg, Austria (July 27-30, 2008).

contributions on various aspects of it, so that we can today claim that we have a very good understanding of it.

A short biography of Vincent (in French) can be found in p. 1026, vol 31 of "La Grande Encyclopédie", see
`http://gallica.bnf.fr/ark:/12148/bpt6k24666x`, whereas in
`http://www.allposters.fr/-st/Lasnier-Affiches_c25893_s88165_.htm` a copy of his portrait can be seen, [25].

Vincent's theorem depends on Descartes' rule of signs, which gives us an *upper bound* on the number of the positive roots of a polynomial, [24]. Specifically, consider the polynomial $p(x) \in \mathbb{R}[x]$, $p(x) = a_n x^n + \cdots + a_1 x + a_0$ and let $\mathrm{var}\,(p)$ represent the number of sign *variations* or *changes* (positive to negative and vice-versa) in the sequence of coefficients $a_n, a_{n-1}, \ldots, a_0$.

Descartes' rule of signs. The number $\varrho_+(p)$ of real roots – multiplicities counted – of the polynomial $p(x) \in \mathbb{R}[x]$ in the open interval $]0, +\infty[$ is bounded above by $var(p)$; that is, we have $\mathrm{var}\,(p) \geq \varrho_+(p)$.

According to Descartes' rule of signs if $var(p) = 0$ it follows that $\varrho_+(p) = 0$.

Additionally, according to Descartes' rule of signs, the mean value theorem and the fact that the polynomial functions are continuous, if $var(p) = 1$ it follows that $\varrho_+(p) = 1$, [24].

Therefore, Descartes' rule of signs yields the *exact* number of positive roots *only* in the two special cases mentioned above[1].

These two special cases of Descartes' rule are used in both versions of Vincent's theorem of 1836, which are described below.

The rest of the paper is structured as follows.

In Sec. 2, we present the two versions of Vincent's theorem and provide a sketch of one of its proofs.

In Sec. 3, we explain how Vincent's theorem can be used to isolate the real roots of polynomials and describe the continued fractions method and two bisections methods derived from it.

In Sec. 4, we present recently developed, by us, linear and quadratic complexity bounds on the values of the positive roots of polynomials and elaborate on their impact on the performance of the continued fractions method.

[1] These two special cases were known to Cardano; in other words, what Descartes did was to generalize "Cardano's *special* rule of signs." This detail is mentioned in [6.]

Finally there is the conclusion, where we indicate directions for future research.

2. The two different versions of Vincent's theorem

We begin with Vincent's original version, which was published in the first issue (1836) of the French Journal on *Pure and Applied Mathematics*.

Theorem 1 (Vincent's 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 nonnegative integer and $\alpha_2, \alpha_3, \dots$ are arbitrary 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.

The negative roots are treated in the same way – 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.

This theorem was kept "alive" by J. V. Uspensky in his book of 1948, [42], where it was rediscovered by the author and formed the subject of his Ph.D. Thesis, [1]. For a detailed discussion of the theorem, its extension, the geometrical interpretation of the transformations involved and three different proofs see [20–22]; a fourth proof is presented by Ostrowski [33], who rediscovered a special case of a previously stated theorem by Obreschkoff, (32, p. 81). We will sketch an outline of one of the proofs after we present the second version of the theorem.

In 2000, Alesina and Galuzzi published the so called *bisection version* of Vinncent's theorem, [20–22]. This modern version is stated as follows.

Theorem 2 (Vincent's theorem – "bisection" version, 2000). *Let $p(x)$, be a real polynomial of degree n, which has only simple roots. It is possible to determine a positive quantity δ so that for every pair of positive real numbers a, b with $|b-a| < \delta$, every transformed polynomial of the form*

$$f(x) = (1+x)^n p\Big(\frac{a+bx}{1+x}\Big)$$

has exactly 0 or 1 variations. The second case is possible if and only if $p(x)$ has a simple root within $]a, b[$.

Sketch of a proof. The proof by Alesina and Galuzzi is the most recent and — according to the author – the most elegant of all existing proofs of Vincent's theorem.

To prove the theorem Alesina and Galuzzi show that after a series of transformations (bisections) mentioned in Theorem 1 (Theorem 2) a polynomial with one positive root will eventually have one sign variation, see Fig. 1. To show it, they use the following theorem by Obreschkoff, of 1920–1923, that gives the necessary conditions under which a polynomial with one positive root presents exactly one sign variation in the sequence of its coefficients, [32][2].

Theorem 3 (Obreschkoff's Cone or Sector theorem, 1920-1923). *If a real polynomial has one simple root x_0, and all the other (possibly multiple) roots lie in the sector:*

$$S_{\sqrt{3}} = \{\ x = -\alpha + i\beta \mid \alpha > 0 \ \ \textit{and} \ \ \beta^2 \le 3\alpha^2\},$$

then the sequence of its coefficients has exactly one sign variation.

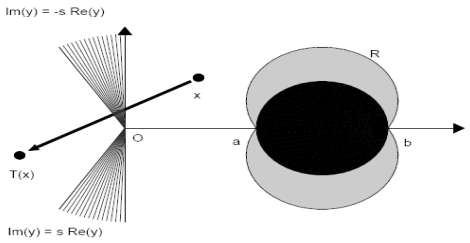

Fig. 1. Obreschkoff's cone or sector theorem of 1920-23, [22].

[2]This is actually Obreschkoff's theorem for the special case where the number of sign variations equals one.

3. Polynomial real root isolation with Vincent's theorem

By cleverly utilizing the two special cases of Descartes' rule – the case of 0 or 1 sign variation – both versions of Vincent's theorem can be used to isolate the positive roots of a given polynomial $p(x)$. To see this, note that if we represent by the Möbius transformation $M(x) = \frac{ax+b}{cx+d}$ the continued fraction that leads to a transformed polynomial

$$f(x) = (cx+d)^n p\Big(\frac{ax+b}{cx+d}\Big) \tag{1}$$

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 correspond to $M(0)$ and $M(\infty)$, respectively[3].

Therefore, to isolate the positive roots of a polynomial, all we have to do is compute – for *each* root – the variables a, b, c, d of the corresponding Möbius transformation $M(x) = \frac{ax+b}{cx+d}$ that leads to a transformed polynomial $f(x) = (cx+d)^n p(\frac{ax+b}{cx+d})$, with one sign variation.

Crucial observation 1. As we will see in the sequel, the variables a, b, c, d of a Möbius transformation $M(x) = \frac{ax+b}{cx+d}$ (in Vincent's theorem) leading to a transformed polynomial with one sign variation can be computed:

- — *either* by *continued fractions*, leading to the continued fractions method developed by Akritas and Strzeboński — which is called the Vincent–Akritas–Strzeboński `(VAS)` *continued fractions* method[4],
- — *or*, by *bisection*, leading to 2 *bisection* methods: the first one was developed by Collins and Akritas and is called the Vincent–Collins–Akritas `(VCA)` *bisection* method[5], whereas the second one was developed by Alesina and Galuzzi and is called the Vincent–Alesina–Galuzzi *bisection* method[6].

[3] As can be seen elsewhere, [7], the endpoints may also be computed from $M(0)$ and $M(1)$, if we work in the interval $]0, 1[$; in that case, Descartes' rule of signs does not apply and we use Uspensky's test instead.

[4] To distinguish it from other continued fraction methods such as [13, 26, 40] cited in ([45, pp. 470–478]). In [37], `(VAS)` is referred to as the "Akritas' continued fractions method". See also [20] and [23].

[5] Erroneously referred to in the literature either as "modified Uspensky's method" (1976–1986), [5], or as "Descartes' method" (1986–2006), [7].

[6] To distinguish them from Sturm's bisection method [45].

It should be noted that the VCA-bisection method was developed first – in 1976 by Collins and Akritas, [27] — and that its fastest implementation was developed in 2004 by Rouillier and Zimmermann, [34]. The VAS-continued fractions method was developed later – in 1978 by Akritas, [2, 3], and in 1994 by Akritas, Botcharov and Strzeboński, [10]. Its fastest implementation was developed in 2008 by Akritas, Strzeboński and Vigklas, [19].

At this point the inquisitive reader might ask why not use numeric methods instead of the symbolic ones mentioned above. The answer is twofold:

— numeric methods *cannot* isolate just the positive roots; they isolate *all* roots, both real and complex, and

— numeric methods can give wrong answers as the following example demonstrates.

Example 1. Consider the polynomial

$$10^{999}(x-1)^{50}-1,$$

whose real roots we want to isolate (it has only two, both positive and $\neq 1$). We solve this example using different versions of *Mathematica*:

— *Mathematica* 5 or 6: In this case,
- a numeric method using 1010 digits takes $56ms$ and fails — finds all 50 roots equal to 1.
- a numeric method using 1020 digits sucsessfully isolates *all fifty* roots, but takes $18000ms$.
- The `VAS` continued fractions method discussed below takes $4ms$ to isolate the two real roots.

— *Mathematica* 7: In this case,
- As can be seen in Fig. 2 the improved numeric method used in Mma 7 takes 12.933 *seconds* to find the two positive roots with 30 digits of accuracy.
- On the other hand, as can be seen in Fig. 3 the function `RootIntervals`, i.e., the *VAS continued fractions method*, isolates the two positive roots in 0.015 *seconds*
- ... and the function `FindRoot` approximates them to 30 digits of accuracy in practically no time at all!

Having made clear the need for the above mentioned symbolic real root isolation methods, we proceed to their description.

```
f = 10^999 (x - 1)^50 - 1;

Select[NRoots[f == 0, x, 30], Im[#[[2]]] == 0 &] // Timing

{12.262, x == 0.99999999999999999999989528714519 || x == 1.00000000000000000000001047128548}
```

Fig. 2. Using the function NRoots in Mma 7, to numerically compute the two roots.

```
ints = RootIntervals[f][[1]] // Timing

{0.015, {{0, 1}, {1, 2}}}

ints = Last[ints]

{{0, 1}, {1, 2}}

FindRoot[f, {x, #[[1]], #[[2]]}, Method → Brent,
    WorkingPrecision → 30, MaxIterations → 200] & /@ ints // Timing

{5.42101 × 10^-19,
 {{x → 0.99999999999999999999989528714519}, {x → 1.00000000000000000000001047128548}}}
```

Fig. 3. Using the functions RootIntervals -- the VAS continued fractions method --- to isolate the roots and the function FindRoot to approximate them, in Mma 7.

3.1. The continued fractions method derived from Vincent's theorem

The `VAS` continued fractions method is a *direct* implementation of Vincent's theorem. It was originally presented by Vincent in 1836 [44] in an "exponential" form; namely, Vincent computed each partial quotient a_i by a series of *unit* increments $a_i \leftarrow a_i + 1$, which are equivalent to substitutions of the form $x \leftarrow x + 1$.

In 1978, the method was converted into its "polynomial" form by Akritas, who in his Ph.D Thesis [1] computed each partial quotient a_i as the lower bound, ℓb, on the values of the positive roots of a polynomial – the so called "*ideal*" positive lower root bound, which computes the integer part of the smallest positive root [16]; that is, we now set $a_i \leftarrow \ell b$ or, equivalently, we perform the substitution $x \leftarrow x + \ell b$, which takes about the same time as the substitution $x \leftarrow x + 1$. For details see also [2–4] and [6, Chapter 7].

Finally, since the ideal positive lower root bound does not exist, Strzeboński [14] introduced the substitution $x \leftarrow \ell b_{\text{computed}} \cdot x$, whenever $\ell b_{\text{computed}} > 16$, where, in general, $\ell b > \ell b_{\text{computed}}$ and the value 16 was determined experimentally.

In [14], it was also shown that the VAS continued fractions method is faster than the fastest implementation of the VCA bisection method [34], a result which was independently confirmed by Tsigaridas and Emiris [41]; see also [17]. In 2007, Sharma removed the hypothesis of the ideal positive lower bound and proved that VAS is still polynomial in time, [36, 37]!

In Algorithm 1 below we present a recursive description of the VAS continued fractions method. We follow [24], which pedagogically seems to be the most appropriate style of presentation:

The VAS continued fractions method

Input: A univariate, square-free polynomial $p(x) \in \mathbb{Z}[x], p(0) \neq 0$,
and the Möbius transformation $M(x) = \frac{ax+b}{cx+d} = x,\ a, b, c, d \in \mathbb{Z}$
Output: A list of isolating intervals of the *positive* roots of $p(x)$;
1 $var \longleftarrow$ the number of sign changes of $p(x)$;
2 if $var = 0$ **then RETURN** $\emptyset$;
3 if $var = 1$ **then RETURN** $\{]a, b[\}$//$a = \min(M(0), M(\infty))$,
$b = \max(M(0), M(\infty))$;
4 $b \longleftarrow$ a lower bound on the positive roots of $p(x)$;
5 if $\ell b > 1\{p \longleftarrow p(x + \ell b), M \longleftarrow M(x + \ell b)\}$;
6 $p_{01} \longleftarrow (x+1)^{deg(p)} p(\frac{1}{x+1}),\ M_{01} \longleftarrow M(\frac{1}{x+1})$//
Look for real roots in $]0, 1[$;
7 $m \longleftarrow M(1)$//Is 1 a root?;
8 $p_{1\infty} \longleftarrow p(x+1),\ M_{1\infty} \longleftarrow M(x+1)$//Look for real roots in $]1, +\infty[$;
9 if $p(1) \neq 0$ then
10 | **RETURN** VAS$(p_{01}, M_{01}) \bigcup$ VAS$(p_{1\infty}, M_{1\infty})$
11 else
12 | **RETURN** VAS$(p_{01}, M_{01}) \bigcup \{[m, m]\} \bigcup$ VAS$(p_{1\infty}, M_{1\infty})$
13 end

Algorithm 1: The VAS (p, M) ''*continued fractions*'' algorithm, where the second argument is the Möbius transformation $M(x)$ associated with $p(x)$. For simplicity, Strzeboński's contribution is not included.

The VAS continued fractions method has been implemented in the computer algebra system *Mathematica*, and, as we will see in the sequel, for the Mignotte polynomials it is several *thousand* times *faster* than the fastest implementation of the the VCA bisections method (see also Fig. 4).

Please note that the VAS continued fractions method uses Descartes'

```
f := x^300 - 2 (5 x - 1)^2;

ints = RootIntervals[f][[1]] // Timing

{0.031, {{-2, 0}, {0, 1/5}, {1/5, 1/4}, {1, 3}}}

ints = Last[ints];
FindRoot[f, {x, #[[1]], #[[2]]}, Method → Brent, WorkingPrecision → 150, MaxIterations → 200] & /@ ints //
 Timing

{0.015, {{x →
    -1.0144385320669281488172557391616077462987206190052230818125849684107751217279585133642868834
      44136979952554984497171389899557015837576228214262248419 8}, {x →
    0.19999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
     99999999999999999999999999999999999999999958661793771948 }, {x →
    0.20000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
     00000000000000000000000000000000000000000041138206229655 }, {x →
    1.01171750912910732155315472587887158814555248053397586417725341251718006156540397630467150535
     40148223065465258510769575765146865990995434417439919722 }}}

 170
-----
0.045
3777.78
```

Fig. 4. For the Mignotte polynomials the VAS continued fractions method is several *thousand* times faster than the VCA bisections method (see also Fig. 7).

rule of signs as termination test – lines 1–3 — and that it relies heavily on the *repeated* estimation of lower bounds on the values of the positive roots of polynomials – line 4. It should be emphasized that the efficiency of `VAS` depends on how good these estimates are – a fact that is exploited in the sequel (Sec. 4). Moreover, note the following:

— If we remove lines 4 and 5 from `VAS`(p, M) we are left with an exponential algorithm.

— Any substitution performed on the polynomial $p(x)$ is also performed on its associated Möbius transformation $M(x)$ — lines 5, 6 and 8.

— To isolate the real roots of $p(x)$ in the open interval $]0, +\infty[$ we proceed as follows:
  - we first isolate the real roots in the interval $]0, 1[$ – lines 6 and 10 (or 12),
  - we then deal with the case where 1 is a root of $p(x)$ – lines 7, 9 and 12,
  - and, finally, we isolate the real roots in the interval $]1, +\infty[$ – lines 8 and 10 (or 12).

— The isolating intervals are computed from the Möbius transformations in line 3 — except for the integer roots which are computed in lines 7 and 12.

Using the plausible hypothesis of the "*ideal*" lower bound on the values of the positive roots along with the fast translation algorithm by von zur Gathen and Gerhard, [29], the computing time of the `VAS` continued

fractions method is

$$O(n^4\tau^2),$$

where n is the degree of the polynomial and τ bounds the coefficient bitsize, [1, 4].

However, the use of the plausible hypothesis mentioned above did *not* go unchallenged, and it was a widely held belief that if it were removed then the computing time of the `VAS` continued fractions method would be exponential! This resulted in people shying away from `VAS` ignoring that:

— in the case of random polynomials the `VAS` continued fractions method was several thousand times faster than (the fastest implementation, [34], of) the `VCA` bisection method (described below), [14, 41],

— in the case of Mignotte polynomials the `VAS` continued fractions method was about 50,000 times faster than (the fastest implementation, [34], of) the `VCA` bisection method, [14, 41],

— only in the case of very many (≥ 20) very large ($\simeq 10^{300}$) roots was the `VAS` continued fractions method 4 times slower than (the fastest implementation, [34], of) the `VCA` bisection method, [14, 41]; see Table 1 in

And all that using Cauchy's bound (the only one available then, but also one of the worst as we will see in Sec. 4) to compute the lower bounds on the values of the positive roots!

The situation changed in 2007 when Sharma proved that without any hypotheses the computing time of the `VAS` continued fractions method is

$$O(n^8\tau^3),$$

where again n is the degree of the polynomial and τ bounds the coefficient bitsize, [36, 37]. However, this indicates that there must still be something we do not understand about this algorithm, since the gap between the two computing time bounds is quite big, and the latter does *not* match the performance of `VAS` – the fastest real root isolation method.

3.2. The Two bisection methods derived from Vincent's theorem

The two bisection methods derived from Vincent's theorem differ mainly in: (a) the termination criterion they employ, and, (b) the interval they bisect.

The *first* bisection method, `VCA`, derived from Vincent's theorem was developed in 1976 by Collins and Akritas, [27], in an attempt to improve

the exponential behavior of Vincent's original continued fractions algorithm. It uses Uspensky's termination criterion (explained below) and bisects the interval $]0,1[$.

Let $p(x)$ be the polynomial whose roots we want to isolate and let ub be an upper bound on the values of its positive roots. Then all the positive roots of $p(ub \cdot x)$ lie in the interval $]0,1[$ and the `VCA` method isolates them by repeatedly bisecting the interval $]0,1[$, while using in the process an appropriate "*criterion*" to make inferences about the number of positive roots certain transformed polynomials have in the interval $]0,1[$. Finally, the isolating intervals of the roots of $p(x)$ are easily computed from the bijection:

$$\alpha_{]0,ub[} = a + \alpha_{]0,1[}(b-a), \tag{2}$$

that exists between the roots $\alpha_{]0,1[} \in]0,1[$ of the *transformed* polynomial $p(ub \cdot x)$ and the roots $\alpha_{]0,ub[} \in]a,b[=]0,ub[$ of the *original* polynomial $p(x)$.

The appropriate criterion mentioned above is a "*test*" that determines an *upper bound* on the number of positive roots in the interval $]0,1[$.

Please observe that Descartes' rule of signs *cannot* be used in the interval $]0,1[$, as it applies *only* to positive roots in the interval $]0,+\infty[$. Therefore, we have to resort to a different "*rule*" if we want to avoid reinventing Sturm's method for isolating the real roots; recall that Sturm's theorem gives us the *exact* number of positive roots in any interval $]a,b[$, [20].

Here is the test for determining an *upper bound* on the number of positive roots in the interval $]0,1[$; as explained below, it is named after Uspensky, who was the *first* to use it.

Uspensky's test. The number $\varrho_{01}(p)$ of real roots in the open interval $]0,1[$ – multiplicities counted – of the polynomial $p(x) \in \mathbb{R}[x]$ is bounded above by $var_{01}(p)$, where

$$var_{01}(p) = var\Big(\big(x+1\big)^{deg(p)} p\Big(\frac{1}{x+1}\Big)\Big), \tag{3}$$

and we have $var_{01}(p) \geq \varrho_{01}(p)$[7].

As in the case of Descartes' rule of signs if $var_{01}(p) = 0$ it follows that $\varrho_{01}(p) = 0$ and if $var_{01}(p) = 1$ it follows that $\varrho_{01}(p) = 1$.

Therefore, Uspensky's test yields the *exact* number of positive roots *only* in the two special cases mentioned above; to wit, whenever $var_{01}(p) = 0$ or $var_{01}(p) = 1$.

[7] Uspensky's test is a special instance of the powerful "Vincent's test", which is described below.

Please note in Eq. (3) that, *after* the substitution $x \longleftarrow \frac{1}{x+1}$, the positive roots of $p(x)$ that were in the interval $]0,1[$ are now in $]0,+\infty[$, in which case Descartes' rule of signs *can* be applied.

Uspensky's test is associated with Budan's theorem [4] according to which for a given polynomial $p(x) \in \mathbb{Z}[x]$ the following two special cases hold:

— if $var(p(x)) = var(p(x+1))$, then we can conclude that there are no positive real roots of $p(x)$ in the interval $]0,1[$,

and

— if $var(p(x)) - var(p(x+1)) = 1$, then we can conclude that there is one positive real root of $p(x)$ in the interval $]0,1[$.

Vincent was fully aware of Budan's theorem and, consequently, the substitution $x \longleftarrow \frac{1}{x+1}$ is *never* used as a test in the `VAS` method – line 6 of Algorithm 1; it is performed *only* whenever $var(p(x)) - var(p(x+1)) \geq 2$, in which case the existence of positive roots in $]0,1[$ *has* to be investigated. In other words, Vincent approaches a root as indicated in Fig. 5.

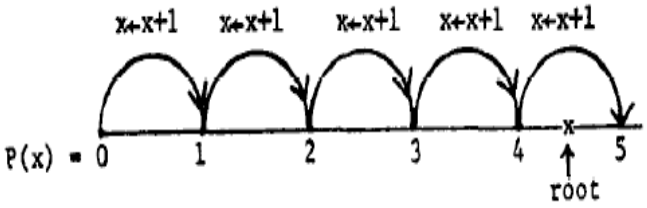

Fig. 5. Being aware of Budan's theorem, Vincent proceeded accordingly.

That, however, was not the case with Uspensky. Whenever he encountered $var(p(x)) = var(p(x+1))$ – not being aware of Budan's theorem – he could not conclude that there are no positive roots of $p(x)$ in the interval $]0,1[$; he would reach that conclusion *only* if $var_{01}(p) = 0$[8]. So, ([42, p. 128]) the transformation $x \leftarrow \frac{1}{1+x}$ was performed *before* the transformation $x \leftarrow x+1$. In other words, Uspensky proceeded as indicated in Fig. 6.

[8] On the other hand, Uspensky used correctly and to his advantage the other special case, $var_{01}(p) = 1$, as well as the case $var_{01}(p) \geq 2$.

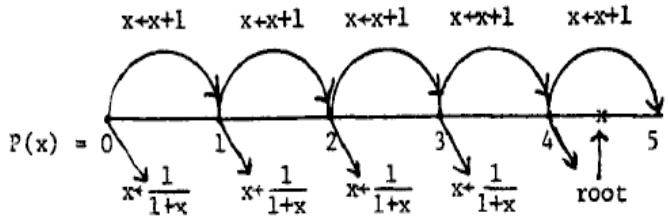

Fig. 6. Not being aware of Budan's theorem, Uspensky proceeded making unnecessary transformations.

Therefore, Uspensky was the *first* to use $var_{01}(p) = 0$ *exclusively* as a test, in order to verify that there are no positive roots in the interval $]0, 1[$; hence, naming the test after him seems to be very appropriate. That test was used by Uspensky in his unsuccessful attempt to develop a new procedure for the isolation of the real roots of polynomials [5, 20][9].

Below is a recursive description of the VCA bisection method:

The VCA bisection method has been implemented in the computer algebra system *maple*, where it can be used *after* loading the appropriate package – as can be seen in Fig. 7.

Please note that the VCA bisection method uses Uspensky's test as termination criterion – lines 1–3 – and that the upper bound on the values of the positive roots is computed *only* once. An excellent discussion of this algorithm can be found in [24]; another version of the same algorithm as well as additional information can be found elsewhere, [7].

Moreover, note the following:

— The substitutions in lines 4 and 6 are performed only on the polynomial $p(x)$, whereas at the same time – in line 5 — the interval $]a, b[$

[9] According to Professor Alexei Uteshev [43], of St.-Petersburg's State University, the reason for Uspensky's unsuccessful attempt was the fact that he *never* saw Vincent's actual paper of 1836, where Budan's theorem is stated right at the beginning. Instead, Uspensky relied on the Russian translation of J.-A. Serret's *Cours d'Algèbre Supérieur* [35]. Indeed, in Sec. 167, p. 315 of И. А.Серре, Курс высшей алгебры. Вольф, Б. г. 573 с. we read:
"В одном из мемуаров, составляющих часть первого тома Journal de Mathématiques pures at appliques, Винсент изложил прекрасное свойство непрерывных дробей и вывел из него для вычисления вещественных корней уравнения способ, вытекающий одновременно и из способа Ньютона, и из способа Лагранжа..."
Please note that Serret presents *Fourier*'s theorem under the name "Budan".

The VCA bisection method -- original version

Input: A univariate, square-free polynomial $p(ub \cdot x) \in \mathbb{Z}[x]$, $p(0) \neq 0$, and the open interval $]a, b[=]0, ub[$, where ub is an upper bound on the values of the positive roots of $p(x)$. (The positive roots of $p(ub \cdot x)$ are all in the open interval $]0, 1[$.)
Output: A list of isolating intervals of the *positive* roots of $p(x)$

1 $var \longleftarrow$ the number of sign changes of $(x+1)^{deg(p)} p(\frac{1}{x+1})$;
2 if $var = 0$ **then RETURN** $\emptyset$;
3 if $var = 1$ **then RETURN** $\{]a, b[\}$;
4 $p_{0\frac{1}{2}} \longleftarrow 2^{deg(p)} p(\frac{x}{2})$//Look for real roots in $]0, \frac{1}{2}[$;
5 $m \longleftarrow \frac{a+b}{2}$//Is $\frac{1}{2}$ a root?;
6 $p_{\frac{1}{2}1} \longleftarrow 2^{deg(p)} p(\frac{x+1}{2})$//Look for real roots in $]\frac{1}{2}, 1[$;
7 if $p(\frac{1}{2}) \neq 0$ **then**
8 | **RETURN** VCA$(p_{0\frac{1}{2}},]a, m[) \bigcup$ VCA$(p_{\frac{1}{2}1},]m, b[)$
9 else
10 | **RETURN** VCA$(p_{0\frac{1}{2}},]a, m[) \bigcup \{[m, m]\} \bigcup$ VCA$(p_{\frac{1}{2}1},]m, b[)$
11 end

Algorithm 2: The *original* version of the VCA$(p,]a, b[)$ ''*bisection*'' algorithm, where the second argument is the open interval $]a, b[$ associated with $p(x)$. The isolating intervals of the roots of $p(x)$ are computed directly, without using bijection (2).

```
> with(RootFinding):
> f := x^300 - 2(5x - 1)^2;
                              f := x^300 - 2 (5 x - 1)^2
> st := time( ): Isolate(f, digits = 250): time( ) - st;
                                    170.431
>
```

Fig. 7. For the Mignotte polynomials the VCA bisection method is several thousand times slower than the VAS continued fractions method (see also Fig. 4).

is divided into two equal parts $]a, m[$ and $]m, b[$, to be used in line 8 or 10.

— To isolate the real roots of $p(x)$ in the open interval $]0, 1[$ we proceed as follows:

- we first isolate the real roots in the interval $]0, \frac{1}{2}[$ – lines 4 and 8 or 10,
- we then deal with the case where $\frac{1}{2}$ is a root of $p(x)$ – lines 5, 7 and 10,
- and, finally, we isolate the real roots in the interval $]\frac{1}{2}, 1[$ – lines 6 and 8 or 10.

— The isolating intervals are directly obtained from line 3 – except for those roots that happen to coincide with the midpoint of an interval that gets bisected, in which case they are computed in lines 5 and 10.

The computing time of the `VCA` bisection method is

$$O(n^4\tau^2),$$

where n is the degree of the polynomial and τ bounds the coefficient bitsize, [37].

The *second* bisection method was developed by Alesina and Galuzzi in 2000, [20, 22]. It uses Vincent's powerful test as the termination criterion and bisects the interval $]a, b[=]0, ub[$, where ub is an upper bound on the values of the positive roots. Therefore, this method is a direct implementation of Theorem 2.

Vincent's test. If $a \geq 0$ and $b > a$ then the number $\varrho_{ab}(p)$ of real roots in the open interval $]a, b[$, – multiplicities counted – of the polynomial $p(x) \in \mathbb{R}[x]$ is bounded above by $var_{ab}(p)$, where

$$var_{ab}(p) = var\Big(\big(1+x\big)^{deg(p)} p\Big(\frac{a+bx}{1+x}\Big)\Big), \tag{4}$$

and we have $var_{ab}(p) = var_{ba}(p) \geq \varrho_{ab}(p)$.

Note that this test can be applied also in the case $]a, b[=]1, 0[$, from which we obtain Uspensky's test[10].

As in the case of Descartes' rule of signs if $var_{ab}(p) = 0$ it follows that $\varrho_{ab}(p) = 0$ and if $var_{ab}(p) = 1$ it follows that $\varrho_{ab}(p) = 1$.

Therefore, Vincent's test yields the *exact* number of positive roots *only* in the two special cases mentioned above; to wit, whenever $var_{ab}(p) = 0$ or $var_{ab}(p) = 1$.

[10] By comparison, Uspensky's test is rather weak as it applies *only* in the interval $]a, b[=]0, 1[$.

The Vincent--Alesina--Galuzzi bisection method: B

Input: A univariate, square-free polynomial $p(x) \in \mathbb{Z}[x]$, $p(0) \neq 0$, and the open interval $]a, b[=]0,\ ub[$, where ub is an upper bound on the values of the positive roots of $p(x)$.
Output: A list of isolating intervals of the *positive* roots of $p(x)$

1 $var \longleftarrow$ the number of sign changes of $(1+x)^{deg(p)}p(\frac{a+bx}{1+x})$;
2 if $var = 0$ **then RETURN** $\emptyset$;
3 if $var = 1$ **then RETURN** $\{]a, b[\}$;
4 $m \longleftarrow \frac{a+b}{2}$//Subdivide the interval $]a, b[$ in two equal parts;
5 if $p(m) \neq 0$ then
6 | **RETURN** (B)$(p,]m[) \bigcup$ (B)$(p,]m, b[)$
7 else
8 | **RETURN** (B)$(p,]m[) \bigcup \{[m, m]\} \bigcup$ (B)$(p,]m, b[)$
9 end

Algorithm 3: The B$(p,]a, b[)$ ''*bisection*'' algorithm, proposed by Alesina and Galuzzi [22]; the second argument is the open interval $]a, b[$, whose endpoints a, b are used in Vincent's test in line 1. The isolating intervals of the roots of $p(x)$ are computed directly, without using bijection (2).

Below is a recursive description of the second bisection method derived from Vincent's theorem; its simplicity is unsurpassed, but we pay for it by using a much more complicated test. Obviously, there is a trade off between simplicity of the method and complexity of the termination test.

Please note the following:

— Vincent's test is a *crucial component* of the B$(p,]a, b[)$ bisection algorithm – lines 1–3.

— In line 4 the interval $]a, b[$ is divided into two equal parts $]a, m[$ and $]m, b[$, to be

used in lines 6 or 8. Note that there are no polynomial transformations at all; only polynomial evaluations in line 1.

— To isolate the real roots of $p(x)$ in the open interval $]a, b[$ we proceed as follows:
- we first isolate the real roots in the interval $]a, \frac{a+b}{2}[$ – lines 6 or 8,
- we then deal with the case where $\frac{a+b}{2}$ is a root of $p(x)$ – lines 5, and 8,
- and, finally, we isolate the real roots in the interval $]\frac{a+b}{2}, b[$ – lines 6 or 8.

— The isolating intervals are directly obtained from line 3 – except for those roots that happen to coincide with the midpoint of an interval that gets bisected, in which case they are computed in lines 5 and 8.

A complete discussion and empirical comparison between the two bisection methods can be found elsewhere, [18]. It turns out that the Vincent–Alesina–Galuzzi bisection method, despite its simplicity, is much slower than the `VCA` bisection method and hence of little practical importance.

4. Improving the performance of the VAS continued fractions method

As was pointed out in the discussion of `VAS`, the efficiency of this 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, [32]. To state it, we refer to polynomials of the type

$$p(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \ldots + \alpha_0, \quad (\alpha_n > 0) \tag{5}$$

with real coefficients $\alpha_n, \alpha_{n-1}, \ldots, \alpha_0$ and having at least one sign variation.

Theorem 4 (Cauchy's theorem)**.** *Let $p(x)$ be a polynomial as in Eq. (5), of degree $n > 0$, 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, [30], and was used in the SYNAPS implementation of `VAS` by Tsigaridas and Emiris in 2006, [41]. Kioustelidis' theorem is closely related to the one by Cauchy and is stated below.

Theorem 5 (Kioustelidis' theorem)**.** *Let $p(x)$ be a polynomial as in Eq. (5), 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 or Kioustelidis' bound – showed that its "Achilles' heel" was the case of very many very large rational roots. In this case 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, [34]. Table 1 presented below, corresponds to the last table (Table 4), found in [14].

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.

Therefore, the question was posed: can we improve the performance of `VAS` by discovering new bounds on the values of the positive roots?

In order to answer this question we needed to better understand the nature of these bounds, and this was achieved with the help of Ştefănescu's theorem of 2005, [38].

Theorem 6 (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*

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

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.

We generalized Ştefănescu's theorem in the sense that Theorem 7 below applies to polynomials with any number of sign variations, [15]. 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[11], [16].

Theorem 7 (Akritas–Strzeboński–Vigklas, 2006). *Let $p(x)$*

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

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

$$p(x) = q_1(x) - q_2(x) + q_3(x) - q_4(x) + \cdots + q_{2m-1}(x) - q_{2m}(x) + g(x), \quad (6)$$

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

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

[11] After our work, [16], Ştefănescu also extended his Theorem 6, [39].

and

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

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*

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

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

$$ub = \max_{\{i=1,2,\ldots,m\}} \left\{ \left(\frac{b_{2i,1}}{c_{2i-1,1}}\right)^{\frac{1}{e_{2i-1,1}-e_{2i,1}}} , \ldots , \left(\frac{b_{2i,t(q_{2i})}}{c_{2i-1,t(q_{2i})}}\right)^{\frac{1}{e_{2i-1,t(q_{2i})}-e_{2i,t(q_{2i})}}} \right\}, \quad (7)$$

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* (7) *given above.*

For a proof of this theorem (see [16]). Please note that the partial extension of Theorem 6 presented in [15] does not treat the case $t(q_{2i-1}) < t(q_{2i})$.

Crucial Observation 2. 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 (7) hold for the cases where: (a) we pair coefficients from the nonadjacent 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.

Using Theorem 7, we obtain the following interpretation of Cauchy's and Kioustelidis' theorems:

C Cauchy's "leading-coefficient" implementation of Theorem 7. For a polynomial $p(x)$, as in Eq. (5), with λ negative coefficients, Cauchy method first breaks up its leading coefficient, α_n, into λ *equal* parts and then pairs each part with the first unmatched negative coefficient.

K Kioustelidis' "leading-coefficient" implementation of Theorem 7. For a polynomial $p(x)$, as in Eq. (5), 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.

Kioustelidis' "leading-coefficient" implementation of Theorem 7, 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.

It turns out that all methods for computing upper bounds on the values of the positive roots of a polynomial, are derived from Theorem 7. In the sequel we present linear and quadratic complexity bounds derived from Theorem 7 and elaborate on their impact on the efficiency of `VAS`.

4.1. Improving `VAS` with linear complexity bounds derived from Theorem 7

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 7 we developed *first-*λ, a new linear complexity method for computing an upper bound on the values of the positive roots of polynomials.

FL "**first-**λ" implementation of Theorem 7. For a polynomial $p(x)$, as in (6), 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 nonincreasing 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 2. 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 obtain a bound estimate of $1.41421 * 10^{50}$.

— 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 $2 * 10^{50}$.

— 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, we developed *local-max*, yet another new linear complexity method for computing an upper bound on the values of the positive roots of polynomials.

LM "**local-max**" implementation of Theorem 7. For a polynomial $p(x)$, as in (5), the coefficient $-\alpha_k$ of the term $-\alpha_k x^k$ in $p(x)$ – as given in Eq. (5) – 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 2, continued. For *local-max* we pair the terms

$$\left\{\frac{10^{100}x^2}{2}, -10^{100}x\right\} \quad \text{and} \quad \left\{\frac{10^{100}x^2}{2^2}, -1\right\},$$

and obtain a bound estimate of 2.

We have tested extensively – on various classes of specific and random polynomials — all four linear complexity bounds mentioned above and the following is a summary of our findings, [16]:

— 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.

— Our *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, our *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, [16].

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 [17].)

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 5th line.

When we compare – on various classes of specific and random polynomials – the times of `VAS(Cauchy)` with those of `VAS(fl+lm)` we observe an overall speed-up of 15%, [19].

4.2. Improving `VAS` with quadratic complexity bounds derived from Theorem 7

To further improve the the performance of the `VAS` continued fractions method we 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.

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.

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	Deg	VAS(Cauchy) $t(s)$ Average (Min/Max)	VAS(fl+lm) $t(s)$ Average (Min/Max)
10	100	0.314(0.248/0.392)	0.253(0.228/0.280)
10	200	1.74(1.42/2.33)	1.51(1.34/1.66)
10	500	17.6(16.9/18/7)	17.4(16.3/18.1)
000	20	0.066(0.040/0.084)	0.031(0.024/0.040)
1000	50	1.96(1.45/2.44)	0.633(0.512/0.840)
1000	100	52.3(36.7/81.3)	12.7(11.3/14.6)
Roots	Deg	VCA(rel) $t(s)$ Average (Min/Max)	Memory (MB) old/new/rel
10	110	0.346(0.308/0.384)	4.46/4.48/4.56
10	200	3.90(3.72/4.05)	4.73/4.77/5.35
10	500	129(122/140)	6.28/6.54/11.8
000	20	0.038(0.028/0.044)	4.57/4.62/4.51
1000	50	1.03(0.916/1.27)	5.87/6.50/5.55
1000	100	17.2(16.1/18.7)	10.4/11.7/9.17

In general, the estimates obtained from the quadratic complexity bounds are better than those obtained from the corresponding linear complexity bounds, as the former are computed after much greater effort and time[12].

We developed several new quadratic complexity bounds by extending three (of the four) linear complexity ones, [8]; by contrast, Kioustelidis' linear complexity bound was extended by Hong in 1998, [28]. We were able to demonstrate that one of them, FLQ, the "first-λ" quadratic complexity bound is not only one of the best to be used in VAS, but also the fastest, [9]. However, we decided that the following quadratic complexity bound should be used in VAS instead, [19].

[12]It should be noted that time is not so importance in our case, since – as can be seen in the description of the VAS algorithm, line 4 – these bounds are estimated *before* a translation of complexity at *least* $O(n^2)$ is executed, [29].

LMQ "Local-Max" Quadratic complexity implementation of Theorem 7. For a polynomial $p(x)$, as in (5), 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:

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

Today, LMQ, is theoretically the sharpest bound on the values of the positive roots of a polynomial, and using it in `VAS` has resulted in an overall speed-up of 40% over its initial implementation.

We finally present Table 3 – corresponding to Table 8 in [19] – 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. Conclusion

We have presented Vincent's theorem of 1836, along with the various methods derived from it for isolating the real roots of polynomials with

rational coefficients. It was shown that the VAS continued fractions method has always been the fastest, and recently – with the development of the quadratic complexity bounds – its performance has increased by an overall factor of 40%.

We end our presentation with a list of topics for future research:

— Sharma's bound on the computing time of the VAS continued fractions method is *greatly overestimated*. Hence, theoretical research is needed to see if we can bring it down.

— The VAS continued fractions method works for *integer* or *rational* coefficients. Hence, we need to discover new ways to deal with coefficients that are *algebraic* numbers or *approximate* reals.

— The VAS continued fractions method is the fastest real root isolation method when the polynomials are *not* sparse and their degree is less than a few thousand. However, *Mathematica runs out of memory* when we try to isolate the roots of a sparse polynomial of degree 100000 or greater. Hence, we need to discover new ways to deal with sparse polynomials of extremely high degrees.

— Last, but not least, we need to investigate the performance of the VAS continued fractions method in a multiprocessor environment.

References

1. A. G. Akritas, *Vincent's theorem in algebraic manipulation.* Ph.D. Thesis, Operations Research Program, North Carolina State University, Raleigh, NC, (1978).
2. A. G. Akritas, *An implementation of Vincent's Theorem.* — Numerische Mathematik **36**, (1980), 53–62.
3. A. G. Akritas, *The fastest exact algorithms for the isolation of the real roots of a polynomial equation.* — Computing **24** (1980), 299–313.
4. A. G. Akritas, *Reflections on a pair of theorems by Budan and Fourier.* — Mathematics Magazine **55**, 5 (1982), 292–298.
5. A. G. Akritas, *There is no "Uspensky's method".* — In: Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation, Waterloo, Ontario, Canada, (1986), pp. 88–90.
6. A. G. Akritas, *Elements of Computer Algebra with Applications.* John Wiley Interscience, New York, 1989.
7. A. G. Akritas, *There is no "Descartes' method".* — In M. J. Wester and M. Beaudin (Eds), Computer Algebra in Education, AullonaPress, USA, (2008), pp. 19–35.
8. A. G. Akritas, *Linear and quadratic complexity bounds on the values of the positive roots of polynomials.* — (Submitted).
9. A. G. Akritas, A. I, Argyris, A. W. Strzeboński, *FLQ, the Fastest Quadratic Complexity Bound on the Values of Positive Roots of Polynomials.* — Serdica Journal of Computing **2**, (2008), 145–162.

10. A. G. Akritas, A. Bocharov, A. W. Strzeboński, *Implementation of real root isolation algorithms in Mathematica.* — Abstracts of the International Conference on Interval and Computer-Algebraic Methods in Science and Engineering (Interval '94), St. Petersburg, Russia, (March 7-10), (1994), 23–27.
11. A. G. Akritas, S. D. Danielopoulos, *On the forgotten theorem of Mr. Vincent.* — Historia Mathematica **5** (1978), 427–435.
12. A. G. Akritas, S. D. Danielopoulos, *An unknown theorem for the isolation of the roots of polynomials.* — Ganita-Bharati (Bulletin of the Indian Society for History of Mathematics) **2** (1980), 41–49.
13. A. G. Akritas, K. H. Ng, *Exact algorithms for polynomial real root approximation using continued fractions.* — Computing **30** (1983), 63–76.
14. A. G. Akritas, A. Strzeboński, *A comparative study of two real root isolation methods.* — Nonlinear Analysis: Modelling and Control **10**, 4 (2005), 297–304.
15. A. G. Akritas, P. Vigklas, *A Comparison of Various Methods for Computing Bounds for Positive Roots of Polynomials.* — Journal of Universal Computer Science **13**, 4 (2007), 455–467.
16. A. G. Akritas, A. Strzeboński , P. Vigklas, *Implementations of a New Theorem for Computing Bounds for Positive Roots of Polynomials.* — Computing **78** (2006), 355–367.
17. A. G. Akritas, A. Strzeboński, P. Vigklas, *Advances on the Continued Fractions Method Using Better Estimations of Positive Root Bounds.* In: 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. Edited by V. G. Ganzha, E. W. Mayr and E. V. Vorozhtsov.
18. A. G. Akritas, A. Strzeboński, P. Vigklas, *On the Various Bisection Methods Derived from Vincent's Theorem.* — Serdica Journal of Computing **2** (2008), 89–104.
19. A. G. Akritas, A. Strzeboński, P. Vigklas, *Improving the Performance of the Continued Fractions Method Using new Bounds of Positive Roots.* — Nonlinear Analysis: Modelling and Control **13** (3) (2008), 265–279.
20. A. Alesina, M. Galuzzi, *A new proof of Vincent's theorem.* — L'Enseignement Mathemathique **44** (1998), 219–256.
21. A. Alesina, M. Galuzzi, *Addentum to the paper "A new proof of Vincent's theorem".* — L'Enseignement Mathemathique **45** (1999), 379–380.
22. A. Alesina, M. Galuzzi, *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, **64** (2000), 179–191.
23. E. Bombieri, A. J. van der Poorten, *Continued fractions of algebraic numbers.* — In: Computational Algebra and Number Theory, (Sydney, 1992), Math. Appl. 325, Kluwer Academic Publishers, Dordrecht, 1995, pp. 137–152.
24. F. Boulier, *Systèmes polynomiaux: que signifie "résoudre" ?.* — Lect. Notes, Université Lille 1, 8 janvier 2007.
`http://www2.lifl.fr/~ boulier/RESOUDRE/SHARED/support.pdf` or
`http://www.fil.univ-lille1.fr/portail/ls4/resoudre`
25. F. Boulier, *Private Communication.* October 2007.

26. D. G. Cantor, P. H. Galyean, H. G. Zimmer, *A Continued Fraction Algorithm for Real Algebraic Numbers.* — Mathematics of Computation **26** (119) (1972), 785–791.
27. G. E. Collins, A. G. Akritas, *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.
28. H. Hong, *Bounds for absolute positiveness of multivariate polynomials.* — J. Symb. Comput. **25** (5) (1998), 571–585.
29. J. von zur Gathen, J. Gerhard, *Fast Algorithms for Taylor Shifts and Certain Difference Equations.* — In: Proceedings of ISSAC'97, Maui, Hawaii, U.S.A., (1997), pp. 40–47.
30. B. Kioustelidis, *Bounds for positive roots of polynomials.* — J. Comput. Appl. Math. bf16, 2 (1986), 241–244.
31. E. K. Lloyd, *On the forgotten Mr, Vincent.* — Historia Mathematica **6** (1979), 448–450.
32. N. Obreschkoff, *Verteilung und Berechnung der Nullstellen reeller Polynome.* — VEB Deutscher Verlag der Wissenschaften, Berlin, (1963)[13].
33. A. M. Ostrowski, *Note on Vincent's Theorem.* — The Annals of Mathematics, 2nd Series **52**, 3 (1950), 702–707.
34. F. Rouillier, P. Zimmermann, *Efficient isolation of polynomial's real roots.* — Journal of Computational and Applied Mathematics **162** (2004), 33–50.
35. J.-A. Serret, *Cours d'Algèbre Supérieur.* Vol. 1, 2. Paris: Gauthier–Villars (1866). Copies of these volumes can be downloaded from `http://www.archive.org/details/coursdalgebsuper01serrrich`.
36. V. Sharma, *Complexity of Real Root Isolation Using Continued Fractions.* — ISAAC07 preprint (2007).
37 V. Sharma, *Complexity Analysis of Algorithms in Algebraic Computation.* Ph.D. Thesis, Department of Computer Sciences, Courant Institute of Mathematical Sciences, New York University, 2007.
38. D. Ştefănescu, *New bounds for positive roots of polynomials.* — Journal of Universal Computer Science **11** (12) (2005), 2132–2141.
39. D. Ştefănescu, *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.
40. K. Thull, *Approximation by Continued Fraction of a Polynomial Real Root.* — Proceedings of the 1984 ACM Symposium on Symbolic and Algebraic Computations, LNCS **174** (1984), 367–377.
41. E. P. Tsigaridas, I. Z. Emiris, *Univariate polynomial real root isolation: Continued fractions revisited.* — (Y. Azar and T. Erlebach (Eds.)), ESA 2006, LNCS **4168** (2006), 817–828.
42. J. V. Uspensky, *Theory of Equations.* McGraw-Hill, New York, 1948.
43. A. Yu. Uteshev, *Private Communication.* September, 2007.

[13]For an English translation of a book with similar content see: Obreschkoff, N.: "Zeros of Polynomials", Bulgarian Academic Monographs (7), Sofia, 2003.

44. A. J. H. Vincent, *Sur la resolution des équations numériques.* — Journal de Mathématiques Pures et Appliquées **1** (1836), 341–372.
45 C. K. Yap, *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, 2000.

Department of Computer and Communication Engineering University of Thessaly Greece
E-mail: akritas@uth.gr

Поступило 14 сентября 2009 г.