Improved Budan-Fourier Count for Root Finding

André Galligo

Improved Budan-Fourier Count for Root Finding

André Galligo*
Laboratoire de Mathématiques
Universite de Nice-Sophia Antipolis (France)

December 13, 2011

Abstract

Given a degree n univariate polynomial $f(x)$, the Budan-Fourier function $V_f(x)$ counts the sign changes in the sequence of derivatives of f evaluated at x. The values at which this function jumps are called the virtual roots of f, these include the real roots of f and any multiple root of its derivatives. This concept was introduced (by an equivalent property) by Gonzales-Vega, Lombardi, Mahé in [17], and then studied by Coste, Lajous, Lombardi, Roy in [8]. The set of virtual roots provide a good real substitute to the set of complex roots; it depends continuously on the coefficients of f. We will describe a root isolation method by a subdivision process based on a generalized Budan-Fourier count, fast evaluation and Newton like approximations. Our algorithm will provide isolating intervals for all augmented virtual roots of f. For a polynomials with integer coefficients of length size $\tau = \tilde{O}(n)$, its bit cost is in $\tilde{O}(n^5)$. We rely on a new connexity property of the Budan table of f which collects the signs of the iterated derivatives of f.

keywords: real univariate polynomial; real root isolation; refinement; Budan-Fourier theorem; Descartes rule; virtual roots; Budan table; Newton process; multiple roots; discretization; separation bound.

1 Introduction

Real or complex root finding of a univariate real polynomial is one of the most classical problem. It re-appears periodically since the 19th century and there is an extensive bibliography on that subject see [18], and [21]. During the last decade, in relation with the applications in Computer Aided Design, the attention focused on the subdivision methods inspired by Vincent's classical algorithm, see e.g. [19], the use of Descartes's rule through homographies (Moebius transforms) and the corresponding representation of real numbers by continued fractions see e.g. [26]. Another novelty is the use of representation of real numbers by dynamic long dyadic approximations called bitstream see [23], and of a

*and INRIA Mediterrannée, Galaad project team.

secant-like method for accelerating the subdivision process see [1]. While completing this article, we have seen on arXiv a paper not yet published by Sagraloff [24] which, with these tools, makes important progresses. Sagraloff presents a subdivision algorithms with the same order of bit complexity bounds than the "considered complicated" almost optimal ones proposed by Schoenague and Pan [21]; although the "complicated" algorithms compute all the complex roots of f.

Our approach is not only conceptually simple and "visual", is also promising and we believe that it has the potential to also meet the quasi optimal complexity. We suggest to avoid a systematic use of Moebius transforms (in the way Descarte's splitting condition is applied), to replace complex roots by "augmented virtual" roots (see below), to apply Newton-Raphson approximation schemes when the derivatives are available, to rely on fast Taylor shifts and evaluations for univariate polynomials. Indeed this last family of algorithms is nowadays well understood and is available as basic commands on long arithmetic packages of some computer algebra systems, see e.g. [6].

In the 19th century the Budan-Fourier theorem, which counts signs variations of a sequence and was followed by the invention of Sturm sequences, was considered as a breakthrough. Subdivision methods, which exploits the ordered structure of the real numbers, are widely applied for calculating good approximations of solutions of polynomial equations or intersections of surfaces in many applied sciences. However the analysis of their complexity, hence efficiency, relies on the algebraic nature of the inputs. The geometric dictionary in complex algebraic geometry between invariants readable on equations and features of varieties is ultimately based on the fact that a polynomial of degree n admits n roots. This is not the case for real roots, and makes real algebraic geometry more complicated. A natural strategy for studying properties of real algebraic varieties is to consider simultaneously roots of iterated derivatives of the input. A first conceptual progress was achieved by Gonzales-Vega, Lombardi, Mahé in [17] when they introduced the concept of the n virtual roots (counted with multiplicities) of a degree n polynomial f, to provide a good "real" substitute to complex roots: The ordered sequence of virtual roots depend continuously on the coefficient of f.

The table containing the signs of all the derivatives of a polynomial f is called, in this paper, its Budan table. It is called after Budan de Boilorant [7], who competed with the famous J. Fourier [12] to provide a proof for the so-called Budan-Fourier theorem. Budan's approach was developed further in a recent work of D. Bembe [3].

As in [14] (see also [4]) we identify the table with a rectangle formed by positive and negative blocks and consider it as a 2D object. The idea developed in this article is to consider successive approximations of the shape of a Budan table, or of portions of this table, defined by their intersection with grids. This will be done in two steps, the first grid is defined as preprocessing taking into account the expected complexity bound of the whole process, then the grid is adapted following a Newton like procedure. We present simple conditions which, when they are satisfied, define what we call a "valid" discretization. However

in our subdivision algorithm, the grid is not statically defined but is the result of a dynamic divide and conquer process, based on an improved Budan-Fourier count.

We adopt a 2D point of view and re-interpret the classical Budan-Fourier bisection method, in such a way that in the "valid" intervals, we get an exact count of real roots (like with Sturm sequences). The presentation of our approach is conceptually simpler when we restrict to the generic case where the roots of all derivatives are two by two distinct, but can form clusters. We are able to analyze and compute isolating intervals for these complicated situations. Then, a slight generalization of our techniques allow to deal with the most general case where either the input polynomial or any of its derivative can have multiple roots; the key tool is a priori given separation bounds.

The paper is organized as follows. Section 2 gives the definitions and the properties of Budan tables and (augmented) virtual roots of a ($\mathcal{P}$) polynomial. and illustrate them with some examples, Section 3 provides conditions to be satisfied for a good discretization of a Budan table (possibly truncated) by a grid; i.e such that an improved Budan-Fourier count gives the exact number of real roots of f in an interval. Section 4 presents the Newton approximation schemes to refine the grid in order to satisfy the previous criteria. Section 5 presents our root finding algorithms of all virtual roots of f and addresses complexity issues. Section 6 reports and comments some experiments.

Notations, separation bounds and arithmetics

$\mathbb{R}$ denotes the field of real numbers, $\mathbb{R}[x]$ denotes the ring of real univariate polynomial, and f a monic polynomial of degree n. $\mathbb{E}_n$ denotes the set of all monic polynomial of degree n, which is identified to $\mathbb{R}^n$. We will also use the notation $f^{(0)}$ for f.

In most part (but not all part) of this paper, f and all its derivatives are assumed square-free. We need two separation bounds to be able to distinguish, after subdivisions, by evaluations on the border of an interval where f' has a single root between three elementary situations: "no root of f", "a double root" or "two distinct roots". The first separation bound, denoted by $s := 2^{-N}$, minors the distance between any two roots of $g(x)$ when g denotes f or any of its derivatives; it allows to certify that an interval with length smaller than s does not contain two distinct roots of such a g. With these notations, it remains to be able to certify that there is no tangent point or equivalently to be able to minor the distance between the graph of g in this interval and the x axis. Assuming that the graph of g (or $-g$) is convex and taking the intersection of the two tangents to the graph at the border of the interval, the graph is contained in a triangle. We want to certify that the x axis does not touch this triangle. The second separation bound, denoted by $t := 2^{-N'}$, minors the distance between two points of the interval, such that the triangle touch the x axis, in case when there is no intersection and no tangency with the x axis.

To simplify the analysis and its presentation. in the case of integer arithmetic, we will assume that the length of the coefficients is bounded by $\tau = \tilde{O}(n)$ and that $N = \tilde{O}(n^2)$. We will also assume that $N' = \tilde{O}(n^2)$. These assump-

tions are pessimistic and will be discussed in the conclusion. A process with a quadratic convergence stopping at the separation bound will have a number of steps bounded by $\tilde{O}(log(n))$, use long integer of length bounded by $\tilde{O}(n^2)$ for representing the abscissas, and long integer of length bounded by $\tilde{O}(n^3)$ to represent their evaluation by the polynomials, hence will have a bit cost of $\tilde{O}(n^4)$. Since we aim to locate by Newton-like processes the n virtual roots of f, the arithmetic worst cost cannot be lower than $\tilde{O}(n^4)$. This is usually considered as the target bound for the root isolation problem(see previous references). Therefore we will make free use of pre-processing having a lower cost. Of course in an optimized implementation one should be more careful.

Finally let us notice that our approach (by evaluation) is also well adapted for bitstream or approximate computations with big-floats. So one can use successively both approximate and exact computations: approximate computations with big-floats is often very efficient providing an approximate result, but since the set of virtual roots depend continuously on the input coefficients, this can be later refined and certified with exact computations. Therefore this work is also a contribution to SNC (Symbolic Numerical Computation).

2 Definitions and results

In this section, we first recall classical facts (see e.g. [22]) or in [17] and [8], then in the second subsection we report results from [15], while in the third subsection we present a useful ingredient for the developments in the next sections.

2.1 Facts on Budan tables and virtual roots

Definition 2.1 Let f be a monic univariate polynomial of degree n. The Budan table of f is the union of $n+1$ infinite rectangles of height one $L_i := \mathbb{R} \times [i-1/2, i+1/2[$ for i from 0 to n, called rows.
For i from 0 to n, each row L_i is the union of a set of open rectangles (possibly infinite), separated by vertical segments. We color in black the rectangles corresponding to negative values of the $(n-i)$-th derivative $f^{(n-i)}$ of f, and we color in gray the rectangles corresponding to positive values of $f^{(n-i)}$.

Remark 2.2

1. Once we know the coefficients of f, the real roots of all its derivatives are contained in an interval $[-2^M, 2^M]$ for some integer M. So the table is in fact finite, and when we say ∞ we mean 2^M.
2. Since f is assumed monic, every infinite right rectangle of each row is gray.
3. Since $f^{(n)}$ is a positive constant, the row L_0 is a gray infinite rectangle.
4. The first (infinite) rectangle of each row L_i is alternatively gray or black, depending on the parity of i: it is gray if $n-i$ is even.
5. We are interested by the connected components of the union of the closures of the gray rectangles; and respectively for the black rectangles.

It is clear that there is a gray connected component containing the infinite right rectangles of all rows. The other connected components (gray or black) are said bounded on the right.

A classical descriptor attached to a Budan table is the function $V_f(x)$ of the real indeterminate x with values in the set of integers $\mathbf{N}$, it counts the number of sign changes in the sequence formed by f and its derivatives evaluated at x.

Definition 2.3 For a sequence $(a_0, \ldots, a_n) \in (\mathbb{R} \setminus \{0\})^{n+1}$ the *number of sign changes* $\mathbf{V}(a_0, \ldots, a_n)$ is defined inductively in the following way:

$$\mathbf{V}(a_0) := 0;$$
$$\mathbf{V}(a_0, \ldots, a_i) := \begin{cases} \mathbf{V}(a_0, \ldots, a_{i-1}) & \text{if } a_{i-1}a_i > 0, \\ \mathbf{V}(a_0, \ldots, a_{i-1}) + 1 & \text{if } a_{i-1}a_i < 0. \end{cases}$$

To determine the number of sign changes of a sequence $(a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$, delete the zeros in $(a_0, \ldots, a_n)$ and apply the previous rule. ($\mathbf{V}$ of the empty sequence equals 0).

We notice that the function V_f is computed from the Budan table of f, but two different tables (of two polynomials f and g) may have the same function $V_f = V_g$. Therefore the Budan table is a finer invariant than V_f attached to the polynomial f.

Proposition 2.4 (Budan-Fourier theorem) *Let $f \in \mathbb{R}[X]$ be monic of degree n. Then,*

- *$V_f(-\infty) = n$, $V_f(\infty) = 0$.*
- *Near a real root c of multiplicity k of f, which is not a root of another derivative of f, V_f decreases by k when x moves from $c - h$ to $c + h$, for sufficiently small positive h .*
- *Near a real root c of multiplicity k of $f^{(m)}$, which is not a root of another non successive derivative of f, the following happens:*
  If k is even, V_f decreases by k.
  If k is odd, V_f decreases by the even integer $k + s_1 s_2$, where s_1 and s_2 are the signs at c of $f^{(m-1)}$ and $f^{(m+k)}$.
- *Near c, a real root of several non successive derivative of f, V_f decreases by the sum of the quantities corresponding to each of them.*
- *Near the other points of $\mathbb{R}$, V_f is constant.*
  The function V_f is decreasing (but not strictly) on $\mathbb{R}$.
- *For $a, b \in \mathbb{R}$ with $a < b$, the number of real roots of f in the interval $]a, b]$ counted with multiplicities is at most $V_f(b) - V_f(a)$. Moreover the defect is an even integer.*

Definition 2.5 Let f be a monic real polynomial of degree n. The x value of the rightmost upper segment of a connected component (either gray or black) of the Budan table of f is called a virtual root of f. Any real root (in the usual sense) of f is a virtual root of f. Any multiple real root (in the usual sense) of any derivative of f is also a virtual root of f. The virtual multiplicities are counted as follows:

- the multiplicities of events appearing along a same $x-$value are added,
- the multiplicity of a simple root of f counts 1,
- the multiplicity of a simple virtual non real root (i.e. it is not a multiple root of a derivative of f) counts 2,
- the multiplicity of a multiple root of f of order k counts k,
- the multiplicity of a multiple virtual non real root which is a multiple root of order k of a derivative of f counts k if k is even, and otherwise $k + s_1 s_2$ where s_1 and s_2 are the signs at c of $f^{(m-1)}$ and $f^{(m+k)}$.

Budan-Fourier theorem (Proposition 2.4) implies that f admits n virtual roots counted with multiplicities. Moreover the following result holds.

Proposition 2.6 *Let $f \in \mathbb{R}[X]$ be monic of degree n. Let $y_1 \leq \cdots \leq y_n$ the ordered virtual roots of f, repeated according to their multiplicities, and $y_0 = -\infty$, $y_{n+1} = \infty$. Then we have for $1 \leq r \leq n+1$, with $y_{r-1} \neq y_r$,*

$$x \in [y_{r-1}, y_r[\Longleftrightarrow$$
$$\mathbf{V}(f(x), f'(x), \ldots, f^{(n)}(x)) = n + 1 - k$$

(resp. for $r = 1$ the interval $x \in]-\infty, y_1[$).

Theorem 2.7 ([17], see also [8]) *The ordered sequence of virtual roots of a monic polynomial f depend continuously on the coefficients of f.*

The virtual roots of f and f' satisfy the "classical" interlacing property.

2.2 Generic case

In this subsection we assume a condition $(\mathcal{P})$, generically satisfied. This will ease the presentation of our new tools. Then we will consider the general case. We introduce and study several data, attached to f and its Budan table.

Definition 2.8 A polynomial g in $\mathbb{R}[x]$ satisfies condition $(\mathcal{P})$ if and only if: each derivative of g has simple roots, and all these roots are two by two distinct. A monic polynomial satisfying this condition will be called a $(\mathcal{P})$-polynomial.

Obviously, the parities of the number of real roots and degree of a $(\mathcal{P})$-polynomial are equal. The property $(\mathcal{P})$ is generically satisfied. Moreover, the set of monic polynomials in $\mathbb{E}_n$, identified with $\mathbb{R}^n$, satisfying $(\mathcal{P})$ form a semi-algebraic set of $\mathbb{E}_n$.

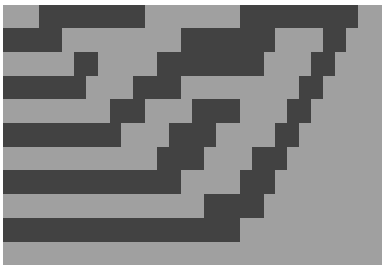

Figure 1: A Budan table of degree 10

Figure 2: A General Budan table

2.2.1 Generic Budan tables

Now, we determine the features of the Budan table B of a generic monic polynomial f with degree n.

Definition 2.9 We say that a table B with $(n+1)$ rows L_i formed by rectangles of alternating colors, gray and black, separated by vertical segments, is a $\mathcal{GB}$ table of degree n if it satisfies the following properties.

- The row L_0 is a gray infinite rectangle. The infinite rightmost rectangle of each row is gray. The first (infinite) rectangle of each row L_i is alternatively gray or black, depending on the parity of i: it is gray if $n-i$ is even.
- If i is even (resp. odd) the number of rectangles on the row L_i is even (resp. odd).
- Let $(l+1)$ be the number of rectangles of the top row L_n, then $l \leq n$ and $n-l$ is an even number $2p$. There are $l+p+1$ same-color-connected components of B. Each non first rectangle of L_i, $i > 0$ is connected on the left to a rectangle of the same color of the row L_{i-1}.
- The previous item is true, replacing n by any m, $0 < m < n$, and B by the table formed by the lower $m+1$ rows.

Theorem 2.10 *Let f be a $(\mathcal{P})$-polynomial of degree n, and let $m \leq n$ be the number or real (simple) roots of f. Then m and n have the same parity, $n = m + 2p$ and the Budan table of f is a $\mathcal{GB}$ table of degree n.*

Example 1 The following Figure 1 shows a $\mathcal{GB}$ table of degree 10 with $m = 4, p = 3$.

Lemma 2.11 Let f be a $(\mathcal{P})$-polynomial of degree n. The x value of the rightmost upper segment of a connected component (either gray or black) of the Budan table of f is a virtual root of f.
Any real root (in the usual sense) of f is a virtual root. Let $m \leq n$ be the number or real (simple) roots of f, and let $n - m = 2p$.
There are p virtual non real roots of f, they are of multiplicity two; each of them is a root of some derivative of f of positive order.

We recover that f admits n virtual roots, counted with multiplicities.

Definition 2.12 We call augmented virtual root of f the pair (y, k) formed by a virtual root of f and the order of the derivative of f which vanishes at y, i.e. $f^{(k)}(y) = 0$.

The augmented virtual roots of f only depend on the Budan table B of f. Virtual roots of a $\mathcal{GB}$ table are well defined.

Proposition 2.13 *Let f be a $(\mathcal{P})$-polynomial of degree n.*
By Rolle's theorem between two successive roots $a < b$ of some derivative $f^{(m)}$ with $0 \leq m \leq n-2$, (or in $\mathbb{R}$ if $f^{(m)}$ has no root), there is an odd number $2r+1$ of roots $(X_1 < ... < X_{2r+1})$ of the next derivative $f^{(m+1)}$. Then the r roots with an even index $(X_2, ...X_{2r})$ are virtual non real roots of f.
Similarly if a is the smallest (resp. the largest) root of $f^{(m)}$, in the infinite interval $]-\infty, a[$ (resp. $]a, \infty[$) there is an even number of roots $2r$ of roots $(X_1 < ... < X_{2r})$ of the next derivative $f^{(m+1)}$. Then the r roots with an odd index $(X_1, ...X_{2r-1})$, (resp. the k roots with an even index $(X_2, ...X_{2r})$) are virtual non real roots of f.
For each augmented virtual non real root (y, k) of f, we have

$$f^{(k-1)}(y)f^{(k+1)}(y) > 0.$$

2.3 General case

In the general case where the condition $(\mathcal{P})$ is not necessarily satisfied, the definitions of (augmented) virtual roots can be generalized and moreover a continuity result holds.

Definition 2.14 We call augmented virtual root of f a triple (y, k, r) formed by a virtual root of f, an order of derivation k 0 and a multiplicity r 1 such that we have:$f^{(k)}(y) = 0$, ..., $f^{(k+r-1)}(y) = 0$ but $f^{(k+r)}(y) \neq 0$; and if $k >$ 0,$f^{(k-1)}(y) \neq 0$. Different augmented virtual root of f can correspond to the same virtual root of f, moreover they only depend on the Budan table of f.

Note that if f is a $(\mathcal{P})$ polynomial then the last coordinates of each triple is 1, so it can be identified with a pair, as we did in the previous subsection.

Each augmented virtual root (y_i, k_i, r_i) may have a multiplicity ≥ 3. See Figure 2 the Budan table of a degree 7 polynomial f with a simple real root, a

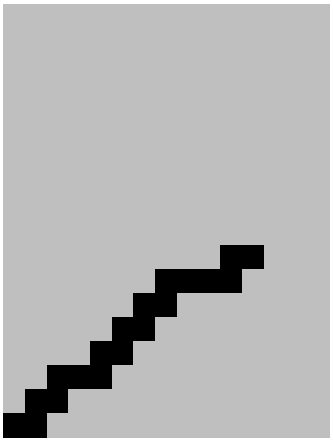

Figure 3: a TWO-truncated Table

real root of multiplicity 4, and a virtual non real root of multiplicity 2 according to the rule of Proposition 2.4 (but is a root of multiplicity 3 of f').

Theorem 2.10 which was proved when condition $(\mathcal{P})$ holds, generalizes to the case when this conditions is violated. The only difference is that the end points of some same-color-connected components have the same first coordinate.

Notice that Figure 2 resembles Figure 1, except that the end points of some same-color-connected components have the same first coordinate.

2.4 Truncated Budan table

Let f be a monic polynomial of degree n, we analyze the properties of a sub table $P := P(f, a, b, u, v)$ of its Budan table BT. P is delimited on the x axis by two real numbers a and b which are not root of a derivative of f, $a < b$, and on the second coordinate by two integers u and v such that $0 \leq u < v \leq n$.

Let us denote by $W(x) := W(f, u, v)(x)$ the function giving the number of sign changes in the sequence formed by the derivatives $f^{(n-k)}$, with $u \leq k \leq v$ evaluated at x. Let $m_2 := W(b) - W(a)$

Among the $p_1 + l_1$ real roots of $f^{(n-u)}$ between a and b, p_1 are virtual roots of f and l_1 are not.

Theorem 2.15 *With the previous notations, let $m := l_1 + m_2$. Then the sub table P has $l+p$ same-color-connected components, with $m = l+2p$; the top row of P has $l+1$ rectangles (their l right segments indicate the l roots of $f^{(n-v)}$*

between a and b); and the p other ends of the same-color-connected components indicate the virtual non real roots of $f^{(n-v)}$ in P (hence virtual non real roots of f).

The proof of this theorem uses the same reasonning and is a very similar to the proof of Theorem 2.10. Note that if a rectangle corner of the lower row of P (which corresponds to a root of $f^{(n-v)}$ between a and b) is surrounded above and on the right by 2 rectangles of opposite color, then in the Budan table of f it is also surrounded below by a rectangle of opposite color; hence it corresponds to a virtual non real root of f.

For an illustration, consider any portion of Figures 1 and 2, or Figure 3 which represents a truncated Budan table analyzed in subsection 5.1.

3 Discretized Budan table

In this section we consider a $(\mathcal{P})$ polynomial f and approximations of its truncated Budan table via intersections with grids. Then sketch on an example a strategy of computation of augmented virtual roots, hence of real roots of f.

We will consider two kinds of grids, the first one is either formed by an arithmetic progression on the x-axis times an interval of integer (degrees of derivatives of f) or by a simple bisection method; while the second one is obtained via a Newton like refining process.

In the illustrations, to be clearer, we present only small portions of the grid; we replace the sign + by a grey solid box and the sign − by a black empty box.

3.1 A simple example

We consider a polynomial of degree $n = 256$ given by $f = \sum_{i=0}^{n} rand()\sqrt{\binom{n}{i}}x^i$; $rand()$ is a function which returns a signed random integer with a uniform distribution between $-N$ and N, where N is a large integer, such a polynomial is often called a random $SO(2)$ polynomial. It is almost surely a generic polynomial which satisfies condition $(\mathcal{P})$. We choose 50 points forming an arithmetic progression between -0.1 and 0.1 and consider f and its first 15 derivatives. So we get a discretized 2D picture with 800 pixels shown in Figure 4. Then we proceed similarly replacing -0.1 and 0.1 by -1 and 1 to get Figure 5.

Let us notice that the Budan-Fourier count for these two intervals gives:

$$V_f(0.1) - V_f(-0.1) = 220\ ,\ V_f(1) - V_f(-1) = 246.$$

Therefore for this example a great part of the virtual roots are between -0.1 and 0.1.

It turns out that the first discretization gives a faithful representation of the truncated Budan table (see below the definition of a "valid" discretization), in contrast with the coarser second one.

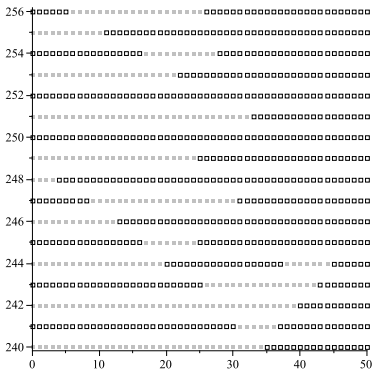

Figure 4: A valid discretized table

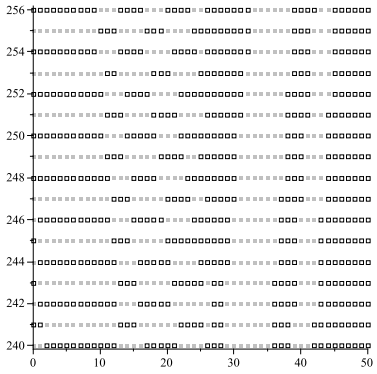

Figure 5: A coarse discretization of a table

We assume that we have already certified (by performing inductively on the lower degrees the same construction) that $f^{(15)}$ has only 1 root between -0.1 and 0.1 with the location shown on Figure 4, i.e with the integer grid coordinates $[34, 240]$. Then we apply Theorem 2.15 to this truncated Budan table.

We count the signs changes and get

$$W(f, 240, 256)(-0.1) = 15\,,\ W(f, 240, 256)(0.1) = 0$$

hence with the notations of Theorem 2.15, m:=15+1=16.

Now we see on the Figure 4 that we have only 7 candidate virtual non real roots, namely with the integer grid coordinates:

$$[24, 245], [24, 249], [27, 254], [30, 247], [32, 251], [44, 244], [30, 241].$$

With the help of some technical tool (Newton approximations until the separation bound or via interval arithmetic), we certify the signs above and below, hence that they are indeed virtual roots of f.

Finally Theorem 2.15 asserts that there are at most 16-2*7=2 real roots of f between -0.1 and 0.1. We see on Figure 4 that there are exactly 2 such roots located in two intervals:
$]-0.1+5*0.2/50, -0.1+6*0.2/50[$ and $]-0.1+25*0.2/50, -0.1+26*0.2/50[$.

Now, let us comment the information given by the grid on Figure 5. It is too coarse and should be refined to become a faithful representation of the corresponding Budan table. Indeed we expect that all derivatives of f are square-free, but this property is not well represented on Figure 5 since in some places the same small interval serves for delimiting the root of some $f^{(k)}$ and of its derivative. E.g. the interval starting at the integer grid coordinates: $[38, 240], [38, 241]$. A refinement or a partition will solve this point, but partition are less expansive from a computational point of view.
In this case, first assume (by induction) that the last row is correctly discretized.

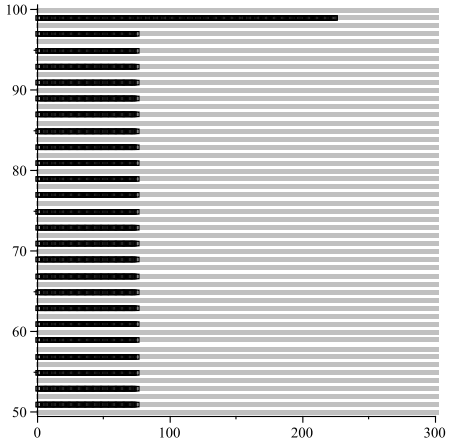

Figure 6: a Mignotte polynomial

A partial Budan-Fourier count, the $W(f, u, v)$ function of the previous section, shows that there are no non real virtual roots between x_{27} and x_{50}, similarly between x_0 and x_{23}. Then there are 7 candidates virtual roots in $[x_{23}, x_{27}]$. This interval is included in $[-0.1, 0.1]$ and it now requires the refinement performed in Figure 4 to separate the augmented virtual roots.

3.2 A Mignotte polynomial

Now let us describes what happens with the very different case of a Mignotte polynomial: $n = 100$, $f := x^n + 2(5x - 1)$. It is well known that this polynomial has only 2 very near real roots. It has also clearly a virtual non real root of multiplicity $n - 2$ at $x = 0$, since its second derivative is $n(n - 1)x^{n-2}$. An arithmetic grid with 300 elements between -0.1 and 0.3 will guess the virtual non real root of multiplicity $n - 2$ but will not distinguish between two near by real roots and virtual root near 0.2 as shown on Figure 6.

4 Valid discretization

4.1 Conditions

In order to express conditions for a good discretization of the Budan table of a $\mathcal{P}$ polynomial f, we consider truncated tables of height 1 (resp. 2), i.e. $P(f, a, b, n, n - 1)$ (resp. $P(f, a, b, n, n - 2)$) and their 2 points discretizations, they give a 2 by 2 grid (resp. 2 by 3). We first assume that we know by induction that if the signs $f'(a)$ and $f'(b)$ are equal (resp. different) there are no root (resp. 1 root) between a and b. Then we classify the configurations of

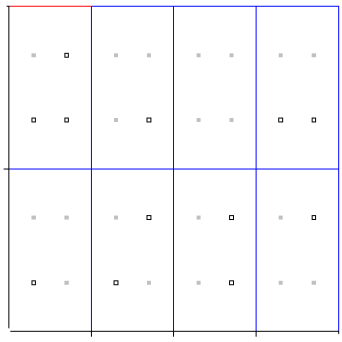

Figure 7: Distributions of signs

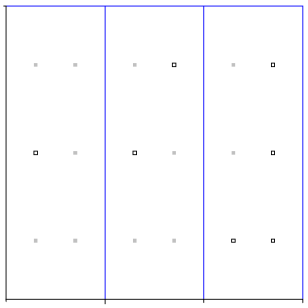

Figure 8: Signs for K_1 or K_2

roots for f in each of the 16 possible distributions of signs on the 2 by 2 grid. Since the roles of + and - are symmetric, we fix to + the sign in the upper left corner, and reduce to 8 the possible distributions of signs.

In the upper row in Figure 7, the 4 cases correspond to valid configurations of signs. In the lower row, the first case that we call K_1 may correspond to a virtual root (possibly of higher multiplicity) or two roots, the second and third cases that we call K_2 should be refined since they locate a root of a polynomial and of its derivative in a same interval, while the last case is impossible.

Definition 4.1 A discretization of a truncated Budan table P by a grid $[x_1, ..., x_N] \times [u, ..., v]$ is called valid, if and only if for each 2 by 2 square formed by the signs $[B[i,k], B[i+1,k], B[i,k+1], B[i+1,k+1]]$, the last 3 cases in the lower row of Figure 7 (or the cases obtained by inverting the signs + and -), never happen.

In this section we consider a first computational strategy which amounts to check inductively by increasing degrees when either of the cases K_1 or K_2 appear, and refine the grid near these points until they are replaced by a case of the upper row of Figure 7 until it is certified that the considered case K_1 corresponds to a virtual root. We are aware that this strategy performs unnecessary computations, but it clearly describe our approach; in the next section we will present a fast algorithm relying on exclusion tests.

The logical condition for case K_2 is

$$B[i,k] * B[i+1,k] = -1 \text{ and } B[i,k+1] * B[i+1,k+1] = -1.$$

The logical condition for case K_1 is for $e = 1$ or $e = -1$,

$$B[i,k] = -e,\ B[i+1,k] = e \text{ and } B[i,k+1] = B[i+1,k+1] = e.$$

Moreover, we have the following simple but useful result.

Lemma 4.2 *For each of the 3 cases of type K_1 or K_2, only one configuration of a 2 by 3 rectangular grid is allowed, if we assume (by induction) that the lower part of the discretization is valid. These configurations have equal signs on the lower third row, they are shown on Figure 8.*

4.1.1 Approximation theorem

Let f be a monic $(\mathcal{P})$-polynomial of degree n, consider a sub table $P := P(f, a, b, u, v)$ of its Budan table BT. P is delimited on the x axis by two real numbers a and b which are not root of a derivative of f, $a < b$, and on the second coordinate by two integers u and v such that $0 \leq u < v \leq n$.

Denote by $W(x) := W(f, u, v)(x)$ the function giving the number of sign changes in the sequence formed by the derivatives $f^{(n-k)}$, with $u \leq k \leq v$ evaluated at x. Let $m_2 := W(a) - W(b)$.

Assume that $f^{(n-u)}$ has r_1 real roots between a and b.

Theorem 4.3 *With the previous notations and definitions, consider the discretization of the table $P := P(f, a, b, u, v)$ by a grid of points. Assume that the cases K_2 never happen and that all the squares of P verifying K_1 are certified to correspond to virtual roots. Assume that the discretization shows r_1 changes of signs on the lower row (which corresponds to the real roots of $f^{(n-u)}$) and p_1 of them verify K_1. Let $m_1 := r_1 - p_1$ and $m := m_1 + m_2$.*
Then the sub table P has $l+p$ same-color-connected components, with $m = l+2p$; the top row of P has l change of signs (they locate the l roots of $f^{(n-v)}$ between a and b); and there are p same-color-connected components ends indicting the virtual non real roots of $f^{(n-v)}$ in P (hence virtual non real roots of f).

Proof: It is a direct consequence of the definitions and of the previous results.

4.2 Refinements

Here, we present for each of the two cases K_1 and K_2 described in the previous section, a refinement scheme relying on Newton-Raphson algorithm.

4.2.1 For K_2

The two sub cases corresponding to K_2 have symmetric shapes, so it suffices to consider the one with positive convexity. Take a derivative g of f corresponding to case K_2 on an interval $[a, b]$, and assume (by induction) that the lower grid is valid. Consider the points A and B of coordinates $A := [a, g(a)]$, $B := [b, g(b)]$ and the intersection point C of the two tangent lines to the graph of g at A and B. By positive convexity the graph of g is contained in the triangle ACB.

The hypothesis K_2 says: $g(a) > 0$, $g'(a) < 0$, $g(b) < 0$, $g'(b) > 0$, and $g''_{[a,b]} > 0$. We look for γ such that $a < \gamma < b$ and $g(\gamma) \leq 0$, $g'(\gamma) < 0$.

We apply the classical Newton-Raphson algorithm to $g(x) - g(b)$ starting from a, and stop when the root of g is isolated. More precisely:

$c := a - \frac{g(a)-g(b)}{g'(a)}$

if $g(c) \leq 0$, then RETURN($\gamma := c$), else update $a := c$ end if.

Note that by positive convexity we have always $g'(c) > 0$. Since the algorithm has quadratic convergence and the distance between a root of g and a root of g' is assumed greater than the separation bound s.

4.2.2 For K_1

Take a derivative g of f corresponding to case K_2 on an interval $[a, b]$, and assume (by induction) that the lower grid is valid. Consider the points A and B of coordinates $A := [a, g(a)]$, $B := [b, g(b)]$ and the intersection point C of the two tangent lines to the graph of g at A and B. By positive convexity the graph of g is contained in the triangle ACB.

The hypothesis K_1 says: $g(a) > 0$, $g'(a) < 0$, $g(b) > 0$, $g'(b) > 0$, and $g''_{[a,b]} > 0$.

We want to determine if the graph of g cuts the x axis between a and b. This will not be the case if the second coordinate of the intersection point C is positive. While if the graph cuts the x axis, we detect it via a γ such that $a < \gamma < b$ and $g(\gamma) \leq 0$, and more likely when the decision is tough $g'(\gamma)$ will be "near" 0.

We will test the two possibilities with an iterative algorithm which alternates two computations: a Newton step for g' starting at a (or b) which updates a (or b) and a step computing the coordinates $[c, L]$ of the intersection C of the two tangents and proceeds as follows.

If $g(c) < 0$ then we return that the interval corresponds to two distinct roots. If $L > 0$ then we return that the interval corresponds to a virtual root, moreover without tangency; else if $g'(c) \leq 0$ then update $a := c$, else update $b := c$ end if end if.

If the size of the interval becomes lower than the two separation bounds then there is a virtual root with tangency.

An alternative test that g remains positive on a small interval $[a, b]$ when we do not know that $g'' > 0$ but when $b - a < 2^{-l}$ is much smaller than 1 is to test if

$$g(a) - (b - a)g'(a) > \sum_{k=2}^{k=n} \frac{g^{(k)}(a)}{k!} 2^{-lk}.$$

The last possibility to consider is when we already had a virtual root which is a multiple root of f or of a derivative of f. Then we rely on the separation bounds.

5 Algorithms for a $(\mathcal{P})$ polynomial

In this section, we describe a fast algorithm for virtual roots isolation for a $(\mathcal{P})$ polynomial, it already contains the main difficulties since we also consider the possibility of clusters. The general case addressed in the next section will follow. We present a (pessimistic) estimation of the worst case bit complexity of our algorithm for a polynomial f with integer coefficients of length $\tau = \tilde{O}(n)$, then

it is known that the roots of f and all its derivatives are in $[2^{-M}, 2^M]$ with $\tau = \tilde{O}(n)$. Moreover, we assume given a separation bound between each pair of roots of each derivative of f, equal to $s = 2^{-N}$ with $N = \tilde{O}(n^2)$. (That is one of the point that we find pessimistic and would like to improve in a future work).

5.1 Subroutine TWO

We first consider the basic case where we have an interval $I = [a, b]$, such that the Budan-Fourier count $BF(f, I)$, (the difference between the two signs variations) is equal to 2. We want to determine if there are 2 distinct real roots of f, else compute the augmented virtual root i.e determine a degree $k > 0$ such that for an interval $I' = [a', b']$ included in I, $f^{(n-k+1)}$ keeps a constant sign on I' and $f^{(n-k)}$ has one simple root in I' with $BF(f^{(n-k)}, I) = 1$.
Assume that neither a nor b is a root of a derivative of f. Let k_1 be the greatest integer such that $BF(f^{(n-k)}, I) < 2$. Necessarily $k \geq k_1$. Then for any $k' > k_1$, $f^{(n-k')}$ has either 0 or 2 roots in I, hence $f^{(n-k')}(a)$ and $f^{(n-k')}(b)$ have the same sign. Moreover by monotony, $f^{(n-k'-1)}(a) < f^{(n-k'-1)}(b)$. These remarks allow to restrict the interval $]k_1, k_2]$ where k should be searched. See Figure 3.
We proceed by a binary search so in at most $log(k_2 - k_1)$ steps. At each step, we test with an integer k', let $g := f^{(n-k')}(x)$, and apply to g Newton steps starting from b to search a value $c < b$ such that $g(c) < 0$. If c is found in $]a, b[$ then we update $k_1 : k'$ and $a := c$. Else we update $k_2 : k'$. Notice that g decreases from b to c, (or to a if there is no c) and that we have $g'' > 0$ on this interval $[c, b]$, except for $k' = k + 1$, detected at the end of the loop and where we proceed as in subsection 4.4.2.

5.2 Preprocessing

We perform $3M = \tilde{O}(n)$ steps of the following bisection, construct a binary tree of segments $I = [a, b]$, and update three sets A, B and BB. When we start $B$and BB are empty and A contains $[2^{-M}, 2^M]$.
Pick $I = [a, b]$ from A, and delete it from A, by fast Taylor shift expand $f(x + a)$ and $f(x + b)$ then compute the Budan-Fourier count $S := BF(I)$, i.e the difference between the two signs variations.
If $S = 0$ then discard I.
If $S = 1$ then put I in BB.
If $S = 2$ then put I in B.
Else divide I by its middle $m(I)$ then add $[a, m(I)]$ and $[m(I), b]$ to A.

At the end of the preprocessing, the set A contains u intervals $I_i = [a_i, b_i]$ such that $b_i - a_i < 2^{-2M}$ and $\sum_{1 \leq i \leq u} BF(I_i) \leq n$. Moreover following [21], we assume that these intervals are separated by $\tilde{O}(n)$ times their size and will call them "clusters" of virtual roots.

The intervals in BB will correspond to interval isolating a root of f. We will later apply Subroutine TWO to the intervals in B.

The total bit cost the preprocessing is of order $\tilde{O}(n^4)$.

5.3 Processing

Now instead of performing bisection of $I \in A$ by the middle $m(I)$, we perform the 2 following steps which aims bounding the clusters of augmented virtual roots in rectangles of type $I \times [k_1, k_2]$. Now in A and B we collect not only the intervals I but indeed the products $I \times [k_1, k_2]$. Just after the preprocessing we start with $k_1 = 0$ and $k_2 = n$.

1) Cutting the bottom and refining:

For a chosen $I \times [k_1, k_2]$ in A, we first compute the lower degree $k + 1$ such that the Budan Fourier count $BF(f^{(n-k-1)}(I)$ becomes greater or equal to 2. Therefore $f^{(n-k)}$ admits a simple root on I and all its derivatives have one or zero (simple) root on I. We propose to perform two Newton steps (or as usual in numerical recipes, mix it with some bisection to avoid to encounter a cycle, since we cannot certify convexity, but still achieve quadratic convergence) from a and b to compute a' and b' with $a \leq a' \leq b' \leq b$. Then update the sets A and B as explained above.

2) Cutting the top:

If for some $I \times [k_1, k_2]$ in A the total multiplicities of the cluster of virtual roots in I, detected by the changes in the signs variations, is greater than $k_2 - k_1$, it means that in the cluster should be divided at least in two parts. Starting from the top, a probable "weakest link" is the the value k_3 where the partial difference of signs variations $W(f, u, v)$ on I (see section 3) pauses. So we perform one of the same sign test K_1 on I. If it succeeds, we delete $I \times [k_1, k_2]$ from A, then add $I \times [k_1, k_3]$ and $I \times [k_3 + 1, k_2]$ in A. Note that $f^{(n-k_3-1)}$ admits a simple root on I.

We stop either when A is empty or if the sizes of all remaining intervals I are smaller than the separation bound.

5.4 Illustration with Figure 4

We consider the example shown in Figure 4, assume that at the end of the preprocessing $I = [-0.1, 0.1]$ and $I \times [240, 256] \in A$; and $I \times [0, 240]$ has already been processed, in particular the real root shown in the last row is simple. We start with a cluster of $S(I, 240, 256) = 16$ virtual roots counted with multiplicities, and cut it into smaller clusters. In this illustration the values obtained by Newton steps are denoted using the letter $\mathcal{N}$.

Step 1: we perform two Newton steps, $\mathcal{N}(f^{(15)}, x_0) \to x_{20}$, $\mathcal{N}(f^{(15)}, x_{50}) \to x_{40}$. so we easily decompose $I \times [240, 256]$ in 3 parts, the left one and the right one are added to B so let us concentrate on the new cluster. We update A adding $I_1 := [x_{20}, x_{40}]$ and $S([x_{20}, x_{40}], 240, 256) = 13$.

Step 2: we perform two Newton steps, $\mathcal{N}(f^{(15)}, x_{20}) \to x_{26}$, $\mathcal{N}(f^{(15)}, x_{40}) \to x_{35}$. Then the rectangle $[x_{20}, x_{40}] \times [240, 256]$ is discarded;

we set $I_2 := [x_{20}, x_{26}]$, $I_3 := [x_{26}, x_{35}]$, add $I_2 \times [245, 256]$ and $I_3 \times [240, 256]$ to A, with $S([x_{26}, x_{35}], 240, 256) = 8$ and $S([x_{20}, x_{26}], 245, 256) = 5$.

Now since $256 - 245 = 11 > 5$ we try to cut $I_2 \times [245, 256]$ from the top: since

the jumps of W are $5, 4, 4, 4, 4, 4, 3, 2, 2, 1$, we test if f' keeps the same sign (here positive) on I_2.
Since it is so, we delete $I_2 \times [245, 256]$ from A, then add $I_2 \times [256, 256]$ to B and $I_2 \times [245, 255]$ to A. etc ...

Post processing:

We consider all elements $I \times [k_1, k_2]$ in B, which have multiplicity 2, and apply Subroutine TWO.

Finally check that all the augmented virtual roots of f has been well separated and certified.

Output the list of the augmented virtual roots of f. Output the list of the number of real roots (and their multiplicities) of each derivative of f.

5.5 Worst case complexity, bit costs

The depth of a Newton iteration tree is $O(log(n))$, i.e. $\tilde{O}(1)$. Hence the size of the total subdivision tree is in $\tilde{O}(n)$. However the last evaluations involve smaller intervals so are more costly, assuming pessimistically that all process go till the separation bound and that each separation bound is in 2^{-N} with $N = \tilde{O}(n^2)$; we arrive at a total bit cost of $\tilde{O}(n^5)$ bits. This lags by a factor n behind the fastest (complex) root finding algorithms of Schoenague and Pan [21].

Remark 5.1 Notice that for the same order of computational bit cost, i.e $\tilde{O}(n^5)$, one can get a set of isolating intervals of the roots for all derivative of f, and the shape of the Budan table of f.

Our bit costs also lags by a factor n behind the new subdivision algorithm for square-free polynomials presented by Sagraloff in his very recent arXiv preprint [24]. However this paper relies on a more subtle separation bound which is the product of all the separation bounds of all the complex roots of f; he uses it for his fine interpretation of Obrechkoff theorems.

We do not have yet in our setting any estimation of a similar smart separation bound for the augmented virtual roots of f.

However, if we restrict to the problem of separating simple real roots of a square-free polynomial, we can implement an early detection subroutine explained below, and discard most branches of the subdivision tree.

In that case we believe (but we did not prove yet) that our approach could also take advantage of a smart separation bound, drop the extra factor n and also meets quasi optimal complexity bounds.

5.6 Early detection

Assuming that f is square-free, after the pre-processing we can concentrate on finding only the real roots of f.

We proceed as in the previous subsection but we discard from the set A all rectangles $[a, b] \times [k_1, k_2]$ such that $k_2 < n$.

With the assumptions and $b - a < 2^{(-L)}$ with $L = \tilde{O}(n)$, in "many" cases the same-sign test will work and certify in the early stage of the subdivision process that $k_2 < n$.

Remark 5.2 If we suspect that $k < n$ an alternate procedure could be to apply a Moebius transform (i.e. a translation composed with an inversion). A generic inversion will not lower the multiplicity of a multiple root (or of a compact cluster) of f, however it will spread out and separate the multiple root (or of a compact cluster) of a derivative of f which are not root of f. One Moebius transform by cluster is enough. Notice that other differentiable bijections, e.g polynomial transform, will do the same effect but they will increase the degree of f.

6 General case

We follow essentially the same algorithm and analysis as in the previous subsection.

The only difference is that the multiplicity of a virtual root can be greater than 2. In the Budan table of f this means as in Figure 3, that some ends of same-sign-connected components instead of being surrounded by points with the opposite sign can have just above them a zero. In other words the corresponding graph admits a tangency.

This can give rise to singular Newton like approximations, but in the previous section we already took it into account with the clusters, and proposed a safe bottom-up process.

Therefore the only difference will appear at the very last steps. at this point we rely on the second separation bound (we assumed a lower bound $2^{-N'}$ with $N' = \tilde{O}(n^2)$) which allow to distinguish the tangency hence the equality of the x values of the augmented virtual roots. At the same order bit complexity cost: $\tilde{O}(n^5)$.

The algorithm similarly output all virtual roots of f but also their multiplicities.

7 Experiments

We have implemented a prototype of our algorithm which needs to be tuned and optimized, nevertheless it produces interesting informations.

The usual benchmarks where f is either a classic random polynomial or Laguerre or Wilkinson or Mignotte polynomials are somehow rough since either the separation bounds are not small or one of their first derivative is a polynomial with many well separated real roots, or if they admit clusters there are only one or two of them. Therefore the efficiency of our approach reduces to the efficiency of the used Taylor shifts which computes the $f(x+a)$ in the preprocessing.
We are not aware of other benchmarks, but it would be interesting to develop a great variety of them.

7.1 A composed example

Here, to present and comment our algorithm on a complete example, we consider the following polynomial (composed with the previous ones with the following notations: $f := Wilkison(n) := \prod_{0\leq i\leq n}(x-i)$ $Kac(96)) := \sum_{0\leq i\leq n} a_i x^i$ where a_i are random real numbers following a standard normal distribution.

$$f := Wilkison(32) * Mignotte(64) + round(10^{10} * Kac(96)).$$

It has degree $n = 96$ with integer coefficients of of about 30 digits, hence τ about n. We chose a rather small degree to ease the description. We denote by $V_f(a)$ the Budan Fourier count at the value a (i.e. the number of sign changes in the sequence of derivatives evaluated at a).

After few (less than $log(n)$) checks. We see that $V_f(-1/2) = 96$, $V_f(7/2) = 1$, and $V_f(4) = 0$. So f admits a simple real root between 3.5 and 4 and all the subdivisions will happen in $[-1/2, 7/2]$ an interval of size 4. In order to illustrate the potential of our approach we perform a "short" preprocessing.

7.2 Preprocessing

We construct a bisection tree of depth 7 i.e. $log(128)$, we collect the intervals of size about 1/32 in 3 sets: BB contains the intervals $[a, b]$ such that $V(a) - V(b) = 1$ isolating a simple root of f, B contains the intervals $[a, b]$ such that $V(a) - V(b) = 2$ isolating a virtual non real root of f, or two simple roots of f, (they will be solved later in a post processing), A contains the intervals $[a, b]$ such that $V(a) - V(b) \geq 3$, that we call "clusters".

For this example, BB contains 4 simple roots:
$[[1, 33/32], [63/32, 2], [3, 97/32], [3.5, 4]]$.
At this stage we cannot certify that they are the only ones.
B contains 15 intervals:
$B := [[3/32, 1/8], [1/8, 5/32], [1/4, 9/32], [3/8, 13/32], [9/16, 19/32],$
$[5/8, 21/32], [21/32, 11/16], [11/16, 23/32], [3/4, 25/32], [25/32, 13/16],$
$[29/32, 15/16], [15/16, 31/32], [31/32, 1], [39/32, 5/4], [29/16, 59/32]]$.

So it remains $96-30-4 = 62$ virtual roots counted with multiplicities. They are included in the 7 following intervals of A, for each of them we indicate the

value $V(a) - V(b)$.

$A := [([-1/16, -1/32], 4), ([-1/32, 0], 36), ([1/16, 3/32], 4), ([3/16, 7/32], 6), ([5/16, 11/32], 4), ([7/16, 15/32], 4), ([17/32, 9/16], 4)]$.

7.3 Processing

For this example, the small clusters can be solved and the new intervals put in the set B. So let us concentrate our description on the more compact cluster $([-1/32, 0], 36)$.

We compare the two lists of signs at $a = -1/32$ and $b = 0$, at their ends we read: $[..., +, -, +]$ and $[..., +, +, +]$ the last derivatives. Hence the degree 1 polynomial $f^{(95)}$ vanishes on that interval, we compute a rough decimal approximation of its root,-0.0098. Then evaluate $V(-0.01) = 80$, $V(-0.009) = 78$, and recall that $V(-1/32) = 92$, $V(0) = 56$ were computed previously.
Therefore after this Newton step, the first compact cluster is replaced in A by 2 new clusters $([-1/32, -0.01], 12)$ and $([-0.009, 0], 22)$; $[-0.01, -0.009]$ is put in B.

Again let us e.g. concentrate our description on $([-0.009, 0], 22)$.

We compare the two lists of signs at $a = -0.009$ and $b = 0$. They are rather similar and differ only between the degrees 32 and 53: at $b = 0$ there are only 21 signs + while at $a = -0.009$ the signs $-$ and $+$ alternate (as described in section 2 for a multiple virtual root of multiplicity 22). We apply two Newton steps to $f^{(64)}$ which is of degree 32, and compute two approximations, of the simple root of that polynomial in the interval, with a doubled precision of 10^{-4}, we found $a' = -10^{-4}$ and $b' = 0$. Moreover $V(-10^{-4}) = 64$. Hence we get two new smaller clusters: $([-0.009, -10^{-4}], 14)$ and $([-10^{-4}, 0], 8)$.

Again we perform Newton steps, the first one with a precision of 10^{-4} and the second one with a doubled precision of 10^{-8}. etc...

When the interval is small enough, here 10^{-8}, we check that we found the correct augmented virtual root. It is not a real root of f.

And so on, until A is empty and B contains 46 elements. For this cluster we computed the Newton steps with a precision up to 2^{-50}, i.e. $\tilde{O}(n)$.

7.4 Post processing

Here we consider each element $[a, b] \times [k_1, k_2]$ of the set B. We look for $k \in [k_1, k_2]$ the greatest integer such that $f^{(n-k)}$ has only one root in $[a, b]$. We assume wlog that there is a safe value k' such that $W(f, k', k_1, a) = W(f, k', k_1, b) + 1$ and $W(f, k', k_1 + 1, a) = W(f, k', k_1 + 1, b) + 2$. Then we apply our subroutine TWO. Its cost depends on the separation distance between the roots of the derivatives.

7.5 Early detection

If we are only interested by computing the real roots, there is a test to get rid of a cluster of virtual roots which is not a multiple root of f. In this example consider the "compact" cluster $([-1/32, 0], 36)$.

We have $f(0)$ 5.10^{35}, but then the signs of the last 30 coefficients of f alternate, therefore on $[-1/32, 0]$ their contribution can only increase positively the approximation of $f(x)$ given by the Taylor expansion. To estimate a lower bound of $f(x)$ we only need to find an upper bound of the "remainder". Since $|x| \leq 2^{-5}$ a bound is $\sum_{i=31}^{i=n} |\text{coeff}(f, x, i)|2^{-5i}$, but this sum is expected to be very small, it is indeed about 10^{-35}. Therefore after the preprocessing we can forget this cluster and subtract 36 from the cumulated multiplicities to be controlled.
More generally, we can take advantage of our knowledge of the signs to bound the remainder in a Taylor expansion.

8 Conclusion

Although there have been many scientific works (and implementations) on real root finding algorithm via subdivision methods, the Budan-Fourier count which historically initiated the subject was not considered as an efficient tool. The reason is that, in contrast with Sturm count or Descartes rule of signs associated to Moebius transforms, it did not provide a termination criterium. In this article we presented a new subdivision based on an improved Budan Fourier count, extended to the derivatives of the input polynomial f, which now provides such a termination criterium.

Instead of just finding the real root of f, our approach allows to also find all the "augmented virtual roots" of f, and eventually the Budan tree and the Budan table of f. Those are new concepts that we introduced in a previous work and that we consider important abstract data associated to f. We view the proposed algorithm as successive approximations of these data. More precisely we have considered a quad-tree like approximations of truncated rectangles of the Budan table of f, in the spirit of the classical and improved Weyl algorithms as explained in [21] for complex root finding, and a bottom-up (with increasing degrees) processing.

The semantic of our approach is geometric and quite classical in singular Newton processes: the multiplicities or compact clusters of roots of g are controlled by a derivative of g having simple well separated roots. The depth and size of the subdivision tree are controlled by separations bounds.

The subject of virtual roots is still new and we do not have yet the fine estimates separation bounds known for the complex roots of a polynomial. We obtained a satisfactory bit complexity cost of $\tilde{O}(n^5)$ for our all procedure.

But if we compare it with the best real root finding algorithms, it lags behind by a factor n, due to the lack of a good global estimate of all the separation

bounds.

This will be a subject for future researches. Another direction of research is to study finely the effect of bijective transforms such as Moebius or Graeffe transforms on clusters of augmented virtual roots.

From another point of view, following the philosophy of [16] if the coeficients of the input polynomial f are approximate real numbers known with some precision (or given by some oracle), one can only expect to compute within some precision the virtual roots (or clusters of virtual roots). Our approach allows to achieve this goal.

In a joint paper in preparation, with Mariemi Alonso Garcia, we are applying the approach presented in this article to the important case of fewnomials.

We hope that our approach will be adopted and further developed, even in higher dimensions, by other researchers.

Acknowledgments

The author thanks Henri Lombardi and Mariemi Alonso Garcia for valuable discussions. The paper was written the author visited the department of Algebra of the University Complutense in Madrid (Spain). This work was partially supported by the contract MathAmSud (11MATH-04-Complexity- Deterministic and probabilistic complexity of algorithms for solving equations) and by the European Marie Curie network SAGA.

References

[1] Abbott,J: Quadratic interval refinement for real roots. Poster presented at the 2006 Int. Symp. on Symb. and Alg. Comp. (ISSAC 2006).

[2] Akritas Alkiviadis G., *Reflections on an pair of theorems by Budan and Fourier*, University of Cansas **22**.

[3] Bembé, D: An algebraic certificate for Budan?s theorem. Journal of Pure and Applied Algebra 215 (2011) 1360 ? 1370.

[4] Bembé, D and Galligo, A: Virtual roots of real polynomials and fractional derivatives. Proceedings of Issac'2011 pp 27-34, ACM, (2011).

[5] Bochnack, J. and Coste, M. and Roy, M-F.: Real Algebraic Geometry. Springer (1998).

[6] Bostan, A and Schost, E: Polynomial evaluation and interpolation on special sets of points. Journal of Complexity (Festschrift for the 70th birthday of Arnold Schnhage) Volume 21 Issue 4, August 2005.

Bostan

[7] Budan de Boislaurent, *Nouvelle méthode pour la résolution des équations numériques d'un degré quelconque*. Paris (1822). Contains in the appendix a proof of Budan's theorem submitted to the Académie des Sciences (1811).

[8] Coste, M and Lajous, T and Lombardi, H and Roy, M-F : Generalized Budan-Fourier theorem and virtual roots. Journal of Complexity, 21, 478-486 (2005).

[9] Eigenwillig,A and Sharma,V and Yap, C: Almost tight complexity bounds for the Descartes method. In ISSAC'06, pages 7178, 2006.

[10] Emiris, I and Galligo, A and Tsigaridas, E: Random polynomials and expected complexity of bissection methods for real solving. *Proceedings of the ISSAC'2010 conference,* pp 235-242, ACM NY, (2010).

[11] Farahmand,K: Topics in random polynomials. Pitman research notes in mathematics series 393, Addison Wesley, (1998).

[12] Fourier, J: Analyse des équations déterminées, F. Didot, Paris (1831).

[13] Galligo, A: Deformation of Roots of Polynomials via Fractional Derivatives Submitted J. Symb. Comp. (Oct.2011).

[14] Galligo, A: Roots of the Derivatives of some Random Polynomials. Proc. SNC, ACM (2011).

[15] Galligo, A: Budan Tables of Real Univariate Polynomials. Submitted J. Symb. Comp. (Oct. 2011).

[16] Labhalla, S and Lombardi, H and Moutai, E: Espace métrique rationnellement présentés et complexité. T.C.S. 250 pp 265-332,(2001).

[17] Gonzales-Vega, L and Lombardi, H and Mahé, L: Virtual roots of real polynomials. J. Pure Appl. Algebra,124, pp 147-166,(1998).

[18] McNamee,J: A bibliography on roots of polynomials. J. of Computing and Applied Math. 47:391-394(1993).

[19] Mourrain,B and Rouillier,F and Roy,M.-F.: The Bernstein basis and real root isolation. In Combinatorial and Computational Geometry, pages 459478. 2005.

[20] Mourrain, B and Vrahatis,M and Yakoubshon, J.C: On the complexity of isolating real roots ofand computing with certainity the topological degree. J. of Complexity, 18:612-640, 2002.

[21] Pan, V: Solving a polynomial equation: some history and recent progress. SIAM Review, 39(2):187220, 1997.

[22] Rahman, Q.I and Schmeisser,G: Analytic theory of polynomials, Oxford Univ. press. (2002).

[23] Rouillier,F and Zimmermann,F: Efficient isolation of polynomials real roots. J. Computational and Applied Mathematics, 162:3350, 2004.

[24] Sagraloff,M: When Newton meets Descartes: A simple and fast algorithm to isolate the real roots of a polynomial. ArXiv [cs.SC], Sept 2011.

[25] Sagraloff,M and Yap,C.-K: A simple but exact and efficient algorithm for complex root isolation. In ISSAC, pages 353360, 2011.

[26] Tsigaridas,E and I. Z. Emiris,I: On the complexity of real root isolation using continued fractions. Theor. Comput. Sci., 392(1-3):158173, 2008.

[27] Vincent M,. *Sur la résolution des équations numéricques*, Journal de mathématicques pures et appliquées **44** (1836) 235–372.