A Root Isolation Algorithm for Sparse Univariate Polynomials

Maria Emilia Alonso, André Galligo

A Root Isolation Algorithm for Sparse Univariate Polynomials

Maria Emilia Alonso Garcia*
Departamento de Algebra, Instituto de Matemática Interdisciplina,
Universidad Complutense de Madrid, Spain.
alonso.mariemi@gmail.com

André Galligo†
Laboratoire de Mathematiques, Universite de Nice-Sophia Antipolis,
Parc Valrose 06108 Nice cedex 02, France
galligo@unice.fr

ABSTRACT

We consider a univariate polynomial f with real coefficients having a high degree N but a rather small number $d+1$ of monomials, with $d << N$. Such a sparse polynomial has a number of real root smaller or equal to d. Our target is to find for each real root of f an interval isolating this root from the others. The usual subdivision methods, relying either on Sturm sequences or Moebius transform followed by Descartes's rule of sign, destruct the sparse structure. Our approach relies on the generalized Budan-Fourier theorem of Coste, Lajous, Lombardi, Roy [8] and the techniques developed in Galligo [12]. To such a f is associated a set of $d+1$ $\mathbb{F}$-derivatives. The Budan-Fourier function $V_f(x)$ counts the sign changes in the sequence of $\mathbb{F}$-derivatives of the f evaluated at x. The values at which this function jumps are called the $\mathbb{F}$-virtual roots of f, these include the real roots of f. We also consider the augmented $\mathbb{F}$-virtual roots of f and introduce a genericity property which eases our study. We present a real root isolation method and an algorithm which has been implemented in Maple. We rely on an improved generalized Budan-Fourier count applied to both the input polynomial and its reciprocal, together with Newton like approximation steps. The paper is illustrated with examples and pictures.

Categories and Subject Descriptors

J.2 [**Mathematics**]; I.1.2 [**Computing methodologies**]: Symbolic and Algebraic Manipulation—*Algebraic Algorithms*

General Terms

Algorithms,Theory

*Partially supported by Spanish organizations: IMI(UCM), UCM (Grupo 910444) and MEC (MTM2011-22435).
†and INRIA Mediterrannée, Galaad project team.

Issac '12, Grenoble, France

Keywords

real univariate polynomial; fewnomial ; real root isolation; Generalized Budan-Fourier theorem; $\mathbb{F}$ virtual roots; $\mathbb{F}$ Budan table; Newton process; discretization; separation bounds

1. INTRODUCTION

On the one hand the concept of sparse representations and more particularly of fewnomial play a crucial role in modern real algebraic geometry and in complexity theory, see [4, 2]. On the other hand, during the last two decades, algorithmic and theoretic progresses have been made on real or complex root finding of a univariate real polynomial; see e.g [15] and [17] and the references therein. See also [9, 10, 25, 27, 16, 22, 23].

In this paper we address, with new tools presented initially in [8] and [13], the root finding problem for sparse polynomials. We consider a univariate polynomial f with real coefficients having a high degree N but a rather small number $d+1$ of monomials, with $d << N$. Such a polynomial (often called sparse) has a number of real root smaller or equal to d. Our target is to find for each real root of f an interval isolating this root from the others. The usual subdivision methods relying either on Sturm sequences or Moebius transform followed by Descartes's rule of sign destruct the sparse structure, hence can hardly be used for very large N.

In the 19th century the Budan-Fourier theorem, which counts signs variations of a sequence of derivatives was considered as an important progress but it only provided a bound on the number of real roots. Then Sturm introduced the Sturm sequences, defined via polynomial Euclidean divisions, to provide an exact count of the real roots in an interval. This breakthrough was followed by many algorithmic progresses, emphasized with the use of computer algebra systems and their applications in applied sciences.

However in some applications (e.g. cryptography) the polynomials have high degrees but are sparse, therefore the efficient determination of their real roots is a natural problem. Our approach relies on the generalized Budan-Fourier theorem of Coste, Lajous, Lombardi, Roy [8] and the techniques developed in [3] and [12]. See also [24, 5, 7] and [3]. To such a f is associated a set of $\mathbb{F}$-derivatives. The Budan-Fourier function $V_f(x)$ counts the sign changes in the sequence of $\mathbb{F}$-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. We also define the augmented $\mathbb{F}$-virtual roots of f. The table containing the signs of all the $\mathbb{F}$-derivatives of a polynomial f is called, in this

paper its $\mathbb{F}$-Budan table, in honor of Budan de Boilorant [5]. Once the coefficients are bounded it is also the case for the real roots of f and all its $\mathbb{F}$-derivatives. So the table is a rectangle decomposed into positive and negative blocks

In [8] it is proved that the ordered sequence of $\mathbb{F}$-virtual roots depend continuously on the coefficient of f. We will also consider a generic condition on f, we call $(\mathcal{FP})$, which imposes that the $\mathbb{F}$-derivatives have pairwise distinct simple roots, but we allow clusters of such roots. This condition eases the analysis and the presentation; since we also consider clusters the general case can be seen as a limit situation. To isolate the real (or $\mathbb{F}$-virtual) roots of f we need a separation bound (which will depend on the size of the coefficients and on d and N), it will serve to stop a subdivision process. Intuitively in the case of integer coefficients, such a separation bound expresses the fact that the input belongs to a finite set of data, consequently the event that two roots collide is discrete and must correspond to a jump. It is well known that a minimum separation bound s satisfies $log(s)$ is $\tilde{O}(Nt)$, where t is the maximum bit size of the integer coefficients of f.

Our strategy of computation is based on simple ideas. First, we borrowed from [8], the sequence of derivatives naturally adapted to sparsity and constructed the $\mathbb{F}$-Budan table which has the same features than the "usual" one studied in [13]. Second, we consider successive approximations of the shape of the $\mathbb{F}$-Budan table, or of portions of this table, defined by evaluations of the $\mathbb{F}$-derivatives at some points determined by an exclusion/inclusion process, which draws a discretized picture of the table. This process can be viewed a revisited version of the classical work of Collins and Loos [6], which was recently reconsidered and improved in [19], see also [18]. Third, we noticed that although the positive roots of $F(x) = x^N f(1/x)$ are the reciprocal of the positive roots of $f(x)$, this is generally not the case for the other $\mathbb{F}$-virtual roots. Hence isolating simultaneously the virtual roots of these two sparse polynomials allows to focus on the real roots and get rid of the other $\mathbb{F}$-virtual roots.

We do not provide a complexity analysis of our algorithm. However, let us say that its complexity is bounded by the complexity of computing the roots of all $\mathbb{F}$-derivatives, adapting [6] and [19]. As above, we denote by t the maximum bit size of the integer coefficients of f. We observe that the evaluation cost of all $d+1$ $\mathbb{F}$-derivatives of f at a dyadic number of bit size $\tilde{O}(Nt)$ is bounded by $\tilde{O}(N^2t^2d^2)$, while for a non sparse polynomial the corresponding cost of a (fast) Taylor shift is at least $\tilde{O}(N^3t)$, hence greater when $N > tildeO(td^2)$. Adapting [19] one can compute (recursively on increasing i) the roots of g_i, the $\mathbb{F}$-derivatives of f, in intervals where they are already isolated and where all the g_j with $j < i$ keep a constant sign (hence the generalized Budan-Fourier count indicates 1). We expect, but did not prove yet, that in that situation a Newton like process will converge quadratically. Since there are less than d^2 such points, we would arrive at a bound of $\tilde{O}(N^2t^2d^4)$. So, we expect our algorithm to be competitive with the best non sparse ones, at least when $N > tildeO(td^4)$.

The paper is organized as follows. Section 2 introduces definitions, concepts and properties of $\mathbb{F}$-Budan tables and (augmented) $\mathbb{F}$-virtual roots of a sparse polynomial. Section 3 illustrates them on examples. Section 4 presents our certified root finding algorithms of real roots of a sparse polynomial f, then reports some experiments. Section 5 consider more general fewnomials, recalling the results of [8].

Notations and illustrative example

$\mathbf{R}$ denotes the field of real numbers, and $\mathbf{R}_+$ the set of positive real numbers, A sparse polynomial f is given by the list of its $(d+1)$ terms, we denote the (non zero) coefficients by a_i and the strictly decreasing degrees by r_i, $f := \sum_{i=0}^{i=d} a_i x^{r_i}$. The degree $N = r_d$ is thought much greater than d.

We use as illustrative example the following sparse polynomial with 50 monomials, degree 1881, and important difference in the sizes of its integer coefficients.

$$
\begin{aligned}
f :=\ & \\
& -4\,x^{1881} - 1205\,x^{1868} - 4950\,x^{1851} - 73411\,x^{1782} - 93098\,x^{1764} \\
& -2574643\,x^{1741} + 2315895\,x^{1679} - 317558621\,x^{1645} + 201491989\,x^{1628} \\
& +9380148787\,x^{1627} + 12412675420\,x^{1618} - 85722418140\,x^{1573} \\
& +10128783780\,x^{1532} + 555687384600\,x^{1421} - 974667164900\,x^{1375} \\
& +2085133349000\,x^{1324} + 3739638336000\,x^{1306} - 12663981830000\,x^{1297} \\
& -16198272240000\,x^{1210} + 46232084120000\,x^{1203} + 21208886110000\,x^{1198} \\
& -22131119870000\,x^{1197} - 4024314069000\,x^{1194} + 40855117930000\,x^{1185} \\
& +76664746130000\,x^{1100} + 40671761780000\,x^{1073} - 66898437710000\,x^{1062} \\
& +6858607911000\,x^{1048} - 61678707000000\,x^{954} - 12048642020000\,x^{939} \\
& -24439027840000\,x^{896} + 10427578370000\,x^{860} + 6972638849000\,x^{762} \\
& +1379195646000\,x^{750} + 1201948057000\,x^{743} - 256631147500\,x^{601} \\
& +121240580200\,x^{582} + 52629237440\,x^{476} + 24106459460\,x^{453} \\
& -8681841885\,x^{450} - 937471907\,x^{392} - 303457638\,x^{378} + 3929228\,x^{361} \\
& +3421959\,x^{311} + 758695\,x^{299} - 221139\,x^{196} + 5255\,x^{99} \\
& -945\,x^{16} + 36\,x^{7} - x^{5}.
\end{aligned}
$$

2. DEFINITIONS AND CONCEPTS

2.1 $\mathbb{F}$-derivatives

The idea of $\mathbb{F}$-derivative of a sparse polynomial $f(x) := \sum_{i=0}^{i=d} a_i x^{r_i}$, is quite natural. ([8] reports that it was already considered by Sturm in a special case). It is based on the simple observation that if $f(x)$ admits a monomial factor x^m then for any $x_0 \in \mathbf{R}_+$, $f(x_0)$ and $f(x_0)/x_0^m$ have the same sign. One first forms the polynomial $g_d(x) = f(x)/x^{r_0}$ which has a non zero constant term, degree $r_d - r_0$, and the same distribution of signs on $\mathbf{R}_+$ than f. Then consider the usual derivative $g_d'(x)$, since $r_1 > r_0$ it admits a factor $x^{r_1-r_0-1}$, then define the polynomial $g_{d-1}(x) := g_d'(x)/x^{r_1-r_0-1}$; it has d terms (one less than g_d), degree $r_d - r_1$. So we can iterate this construction.

DEFINITION 1. *With the previous notations, given a sparse polynomial f, we associate to it the following sequence of its $d+1$ $\mathbb{F}$-derivatives constructed by induction as follows. $g_d := f/x^{r_0}$, for i from 1 to d,*

$$g_{d-i} := (g_{d-i+1})'/x^{r_i-r_{i-1}-1}.$$

So g_{d-i} is a sparse polynomial with $d-i$ monomials and degree $r_d - r_i$ and g_0 is a constant.

REMARK 1. *When we restrict to the half line $x > 0$ as we noticed above, g_{d-i} and the derivative of g_{d-i+1} have the same sign. Hence if $g_{d-i+1}(x_0) = 0$ and g_{d-i} is positive for $x > x_0$ then g_{d-i+1} is also positive for $x > x_0$, but if g_{d-i} is positive for $x < x_0$ then g_{d-i+1} is negative for $x < x_0$. (Respectively when we exchange the words positive and negative).*

This is a simple but key remark for understanding the structure of the $\mathbb{F}$-Budan table of f, which collects the signs of all the polynomials g_{d-i}, for $i = 0..d$.

2.2 $\mathbb{F}$-Budan tables and $\mathbb{F}$-virtual roots

DEFINITION 2. *With the previous notations, let f be a monic sparse polynomial and $(g_{d-i})_i$ the sequence of its $\mathbb{F}$-derivatives. The $\mathbb{F}$-Budan table of f is a subset of the real plane equal to the union of $d+1$ infinite rectangles of height one $L_i := \mathbf{R}_+ \times [i-1/2, i+1/2[$ for i from 0 to d, called rows.*
For i from 0 to d, 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 g_{d-i}, and in gray the rectangles corresponding to positive values.

Illustrative examples are provided in the next section.

REMARK 2.
1. *Once we know the coefficients of f, the positive real roots of all its derivatives are contained in an interval $[0, 2^M]$ 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 g_0 is a positive constant, the row L_0 is a gray infinite rectangle.*
4. *The first rectangle of each row L_i is gray (resp. black), if a_i is positive (resp. negative).*
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 "descriptor" attached to a $\mathbb{F}$-Budan table is the function $V_f(x)$ of the real positive 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 $\mathbb{F}$-derivatives evaluated at x.

DEFINITION 3. *For a sequence $(b_0, \ldots, b_n) \in (\mathbf{R}\backslash\{0\})^{n+1}$ the* number of sign changes $\mathbf{V}(b_0, \ldots, b_n)$ *is defined inductively in the following way:*

$$\mathbf{V}(b_0) := 0;$$

$$\mathbf{V}(b_0, \ldots, b_i) := \begin{cases} \mathbf{V}(b_0, \ldots, b_{i-1}) & \text{if } b_{i-1}b_i > 0, \\ \mathbf{V}(b_0, \ldots, b_{i-1}) + 1 & \text{if } b_{i-1}b_i < 0. \end{cases}$$

To determine the number of sign changes of a sequence $(b_0, \ldots, b_n) \in \mathbf{R}^{n+1}$, delete the zeros in $(b_0, \ldots, b_n)$ and apply the previous rule. ($\mathbf{V}$ of the empty sequence equals 0).

The following proposition contains the Generalized Budan-Fourier theorem of [8].

PROPOSITION 1. *With the previous notations, let f be a monic sparse polynomial and $(g_{d-i})_i$ the sequence of its $\mathbb{F}$-derivatives. Then,*

- *$V_f(0)$ equals the number of sign changes in the sequence of coefficients of f, while $V_f(\infty) = 0$.*
- *Near a real root c of multiplicity k of f, which is not a root of another $\mathbb{F}$-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 $c > 0$ of multiplicity k of g_{d-m}, (or equivalently of k successive $\mathbb{F}$-derivatives), which is not a root of another non successive $\mathbb{F}$-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 g_{d-m+1} and g_{d-m-k}.
- *Near $c > 0$, a real root of several non successive $\mathbb{F}$-derivatives of f, V_f decreases by the sum of the quantities corresponding to each of them.*
- *Near the other points of $\mathbf{R}_+$, V_f is constant.*
  The function V_f is decreasing (but not strictly) on $\mathbf{R}_+$.
- *For $a, b \in \mathbf{R}$ with $0 < a < b$, the number m of real roots of f in the interval $]a, b]$ counted with multiplicities is at most $V_f(a) - V_f(b)$. Moreover the defect $V_f(a) - V_f(b) - m$ is an even integer.*

DEFINITION 4. *With the previous notations, let f be a monic sparse polynomial and (g_{d-i}) the sequence of its $\mathbb{F}$-derivatives. The x value of the rightmost upper edge of a connected component (either gray or black) of the $\mathbb{F}$-Budan table of f is called a $\mathbb{F}$-virtual root of f. Any real root (in the usual sense) of f is a $\mathbb{F}$-virtual root of f. Any multiple real root (in the usual sense) of any $\mathbb{F}$-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 one,*
- *the multiplicity of a simple $\mathbb{F}$-virtual non real root (i.e. not a multiple root of a $\mathbb{F}$-derivative of f) counts two,*
- *the multiplicity of a root of f of order k counts k,*
- *the multiplicity of a multiple $\mathbb{F}$-virtual non real root c which is a 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 g_{d-m+1} and g_{d-m-k}.*

The Generalized Budan-Fourier theorem implies that f admits d $\mathbb{F}$-virtual roots counted with multiplicities. Moreover the following result holds.

THEOREM 2.1 ([8]). *The ordered sequence of $\mathbb{F}$-virtual roots of a sparse polynomial f depend continuously on the coefficients of f.*

2.2.1 Generic case

In this subsection we assume a condition $(\mathcal{FP})$, generically satisfied.

DEFINITION 5. *With the previous notations, let f be a monic sparse polynomial. It satisfies condition $(\mathcal{FP})$ if and only if:*
each of its $\mathbb{F}$-derivatives has simple roots, and all these roots are pairwise distinct. A monic sparse polynomial satisfying this condition will be called a $(\mathcal{FP})$-polynomial.

It is easy to see that the $\mathbb{F}$-Budan table B of a $(\mathcal{FP})$-polynomial f also has the following two features.

- For $0 \leq i \leq d$, if a_i is positive (resp. negative) then the number of rectangles on the row L_i is odd (resp. even).
- Let $(l+1)$ be the number of rectangles of the top row L_d, then $l \leq d$ and $d-l$ 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}.

DEFINITION 6. *We call augmented $\mathbb{F}$-virtual root of f the pair (y,k) formed by a $\mathbb{F}$-virtual root of f and the integer k such that g_{d-k} vanishes at y.*

2.2.2 Multiple roots

In the general case of a sparse polynomial the condition $(\mathcal{FP})$ is not necessarily satisfied, because the g_{d-i} may have multiple roots. To keep the previous two nice features we proceed as follows. We now decompose the row L_{d-i} by a possibly smaller number of rectangles by replacing two adjacent rectangles with the same color by their union (i.e. forgetting the multiple positive roots of g_{d-i} with even order). Then the $\mathbb{F}$-Budan table B of f looks like the Budan table of a $(\mathcal{FP})$-polynomial and we can define similarly the augmented $\mathbb{F}$-virtual root of f to be the pair (y,k) formed by a $\mathbb{F}$-virtual root of f and the integer k such that y is a positive roots of g_{d-k} with odd order of multiplicity; such that (y,k) is the rightmost edge of a same-color-connected component of B.

2.3 Truncated $\mathbb{F}$-Budan table

With the previous notations, let f be a monic sparse polynomial. We analyze the properties of a sub-table $P := P(f,a,b,u,v)$ of its $\mathbb{F}$-Budan table B. P is delimited on the x axis by two real numbers a and b which are not root of a $\mathbb{F}$-derivative of f, $a < b$, and on the second coordinate by two integers u and v such that $0 \leq u < v \leq d$.

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 $\mathbb{F}$-derivatives g_k, with $u \leq k \leq v$ evaluated at x. Let l_1 (resp. l_2) be the number of real roots of g_u (resp. g_v) between a and b.

Let $h_1 := W(a) + l_1$ and similarly $h_2 := W(b) + l_2$. Notice that h_1 counts the number of sign changes along the left-lower corners of the rectangle; similarly h_2 counts the number of sign changes along the right-upper corners of the rectangle.

PROPOSITION 2. *With the previous notations, $h_2 - h_1$ is an even number, we denote it by $2p$. Then the sub-table P has l_2+p same-color-connected components (not bounded on the right), the top row of P has l_2+1 rectangles (their l_2 right ends indicate the l_2 real roots of g_v between a and b); and the p other ends of the same-color-connected components indicate the $\mathbb{F}$-virtual non real roots of g_v in P (hence $\mathbb{F}$-virtual non real roots of f).*

The proof is purely combinatorics and is exactly the same as the one given in [12] for the usual derivatives and the usual Budan tables.

Figure 1: A truncated table with multiple roots

Figure 2: A $\mathbb{F}$-Budan table of a generic polynomial

2.3.1 Discretized Budan table

We can discretize a truncated $\mathbb{F}$-Budan table via intersections with grids. In the illustrations, to be clearer, we replace the sign $+$ by a gray solid box and the sign $-$ by a black empty box.

3. ILLUSTRATIONS AND EXAMPLES

We first provide a picture (Figure 1) of the (truncated) $\mathbb{F}$-Budan table of a low degree polynomial f_1 to illustrate the concept: we consider a polynomial of degree 7 with a multiple real root and another multiple virtual root. We truncated in order to only keep the degrees between $u = 3$ and $v = 7$; and the interval $[0, \infty]$ Notice that the rightmost edges of some rectangles are aligned. We see one gray connected component and two black connected components, (plus a gray component not bounded to the right). We observe 6 sign changes on the leftmost column and 0 sign changes on the rightmost column, 0 roots for g_v in the interval and 3 roots with sign changes for g_u; hence $h_1 = 6$ and $h_2 = 0$. As predicted by the theorem the variation (here 6) equals twice the number of connected components not arriving to the top row, here 3 (two black and one gray).

We then consider the Budan table (Figure 2) of a generic polynomial, notice that the rectangle are not aligned. We see two black connected components, (plus a gray component not bounded to the right). So there are two virtual non real root. We count 4 sign changes on the leftmost column and 0 sign changes on the rightmost column. Moreover the lower and upper polynomials have no real roots in the interval. As predicted by the theorem the variation (here 4) equals twice the number of connected components not arriving to the top row.

Then we consider the reciprocal polynomial of the sparse

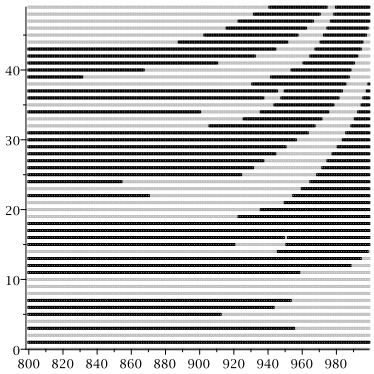

Figure 3: a discretized truncated table

polynomial given as an illustrative example in section 1. Figure 3 shows a discretization of its $\mathbb{F}$-Budan table truncated for $0.8 < x < 1$; we chose an equidistributed grid with 200 values between 0.8 and 1. It allows to "guess" the truncated $\mathbb{F}$-Budan table. (This can be certified by a computation). We can count 19 sign changes on the leftmost column and 4 sign changes on the rightmost column. Since there are no sign change on the first row, we expect $19 - 4 = 15$ contributions of the same-color-connected components. It is what we get. Indeed we see 3 real roots and 6 $\mathbb{F}$-virtual non real roots (each contributes 2), hence $3 + 2 \times 6 = 15$. The grid coordinates of the 6 $\mathbb{F}$-virtual non real roots are: $[870, 22], [900, 34], [949, 37], [951, 16], [953, 7], [955, 3]$.

4. ROOT FINDING ALGORITHMS AND EXPERIMENTS

Here we present the main feature of our root finding algorithm, we made a prototype implementation in Maple.

4.1 Strategy of computation

We will use a subdivision method with inclusion/exclusion tests relying on generalized Budan Fourier counts for the two polynomials f and its reciprocal Rf. Let us first compare the positive virtual roots of f and Rf.

$Rf(x) = x^N f(1/x)$ hence the real roots of Rf are the reciprocal of the real roots of f with the same multiplicity. However if a is a real root of f' but not of f, we have $(Rf)'(1/a) = Na^{-N+1}f(a) - a^{-N+2}f'(a) =$
$= Na^{-N+1}f(a) \neq 0$.

If a is a real root of f'' but not of f, we have $(Rf)''(1/a) = N(N-1)a^{-N+2}f(a) + (2-2N)a^{-N+3}f'(a) + a^{-N+4}f''(a)$, so $(Rf)''(1/a) = a^{-N+2}(N-1)(Nf(a) - 2af'(a))$. Therefore $(Rf)''(1/a)$ may vanish but it is not likely. And similarly for other higher derivatives. This is also applies to $\mathbb{F}$-derivatives.

Hence we can expect that if some a is a $\mathbb{F}$-virtual root of f and $1/a$ is a $\mathbb{F}$-virtual root of Rf then a will be a real root of f. In any case, if we look for the real roots of f we can exclude the reciprocal of the intervals which do not contain any $\mathbb{F}$-virtual root of Rf. Then we will have to check that the remaining intervals does not contain any $\mathbb{F}$-virtual non real root.

For a later step of the algorithm, we will collect in a set B the "small" intervals such that generalized Budan Fourier counts for the two polynomials f and its reciprocal Rf returns 2. This indicates either two close real roots or a virtual root for each of the two polynomials. We will use a Newton like process to narrow the intervals and make a decision. That purpose requires ultimately a separation bound and a control of the quadratic convergence.

We decompose the algorithm into a preprocessing relying on a "small" number of bissection steps, a processing which mix bisections and Newton like steps and a post processing which deals with the intervals in B (i.e. returning 2).

4.2 Subroutines

4.2.1 FDeriv

Our approach uses the $\mathbb{F}$-derivatives of a input sparse polynomial f with $d+1$ monomials. The function FDeriv(f, d) computes the (ordered) sequence g formed by the $d+1$ $\mathbb{F}$-derivatives of f.

4.2.2 sv

The function sv(f, u, v) counts the number of sign variations in the terms (numbered between u and v) of the ordered list of coefficients of f. u and v are integers with $0 \leq u \leq v \leq d$. It also returns the corresponding list of signs.

4.2.3 Count

Our algorithmic method is based on partial Budan Fourier counts. The function Count(g, a, u, v) counts the number of sign variations in the terms (numbered between u and v) of a ordered list of polynomials g, evaluated at a. So a is a positive real number, u and v are integers with $0 \leq u \leq v \leq d$. It also returns the corresponding list of signs.

Notice that Count$(g, 0, u, v)$ is not valid, it is replaced by sv(f, u, v) which plays the same role.

4.2.4 Trunc

Our algorithm is based on the location of the augmented virtual roots in $\mathbf{R} \times [0, d]$, by a kind of quad-tree method which aims to diminish both the real interval and the integer one. Given two lists of signs, the function Trunc(L,M) computes the numbers of sign changes and the index when the difference of numbers of sign changes between the two lists becomes greater than 1, starting from below.

4.2.5 Reciprocal

Our algorithm uses the reciprocal of f to separate the real from the non real $\mathbb{F}$-virtual roots. The function Reciprocal(f) computes the reciprocal polynomial of f, which has the same number of monomials.

4.2.6 Bound

In some cases we need to certify that a polynomial h does not vanish on a small interval $[a, b]$, this is done by the function Bound(f), relying on interval arithmetic.

4.2.7 Halley

We will need a function computing a Newton like step. Since with sparse polynomials, usual Newton approximations "near" zero can be unstable, we propose to use Halley approximation process, which uses the second derivative and

gives an approximation of order 3. More precisely for a polynomial h at a point a, letting $b := h(a), c := h'(a), e := h''(a)$ the increment is given by:

$$\frac{b}{c - \frac{be}{2c}}.$$

4.3 Preprocessing

The first step of the preprocessing, computes $sv(f, 0, d)$ then $Count(g, i, 0, d$ for the integers $i < M$, for some bound M defined by the context, till we find 0 or 1 or 2, call i_0 this maximal integer, and similarly for Rf, and j, call j_0 this maximal integer. The second step proceed for each $1 \leq i \leq i_0$ to bisections with exclusions (resp. selection in a set A) of intervals which returns 0 to one of the two tests for f and Rf. And similarly for j.

At the end of this preprocessing, we have three sets: A which contains isolating intervals for some positive real roots of f; B which contains intervals returning 2 to the generalized Budan Fourier counts for the two polynomials f and Rf; E which contains intervals returning at least 2 and more than 2 to the two generalized Budan Fourier counts (they will be subdivided during the processing step).

4.4 Processing

During this step we will not only subdivide the intervals of E but also the range of integers $[0, d]$, so we replace each intervals I in E by a product $I \times [k_1, k_2]$ and initialize $k_1 := 0, k_2 := d$.

Instead of performing a bisection of I by the middle, we perform the 2 following steps which aims bounding the clusters of augmented $\mathbb{F}$-virtual roots.

1) Cutting the bottom and refining:

For a chosen $I \times [k_1, k_2]$ in E, we first compute the lower degree $k + 1$ such that the generalized Budan Fourier count $BF(g_{d-k+1}(I)$ becomes greater or equal to 2. Therefore g_{d-k} admits a simple root on I and all its $\mathbb{F}$-derivatives have one or zero (simple) root on I. To achieve a good convergence, we propose to perform two Newton like steps from each edge and a bissection (as usual in numerical recipes, or follow [1], to avoid to leave the interval or encounter a cycle, since we cannot certify convexity). Then update the sets A, B and E as explained above.

Notice that an approximation provides a decimal number α , then for any derivative h, the sign of $h(\alpha)$ can be exactly computed. Compare with [27].

2) Cutting the top:

If for some $I \times [k_1, k_2]$ in E the total multiplicities of the cluster of $\mathbb{F}$-virtual roots in I, detected by the changes in the signs variations, is greater than $k_2 - k_1$, it means that the cluster should be divided at least in two parts. Starting from the top, a probable good cutting integer is the the value k_3 where the partial difference of signs variations $W(g, u, v)$ on I (see Section 2) pauses. So we perform a same sign test; if it succeeds, we delete $I \times [k_1, k_2]$ from E, then add $I \times [k_1, k_3]$ and $I \times [k_3 + 1, k_2]$ in E.

We stop either when E is empty or if the sizes of all remaining intervals I are smaller than a separation bound (to be given together with the input).

4.5 Post processing

We consider all elements $I \times [k_1, k_2]$ in B, which returns 2 to both generalized Budan Fourier counts. Then the picture of both $\mathbb{F}$-Budan tables either look like the one pictured in Figure 4, or there are two (close) real roots.

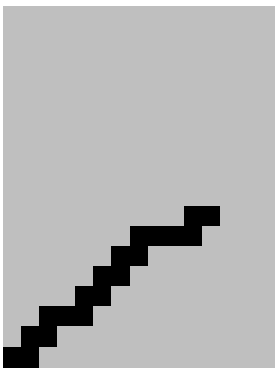

Figure 4: a Two truncated table

We determine a degree $k > 0$ such that for an interval $I' = [a', b']$ included in I, g_{d-k+1} keeps a constant sign on I' and g_{d-k} has one simple root in I', then check if the corresponding generalized Budan Fourier count returns 1. We use Newton like procedures to compute I' from I.

Finally check that all the augmented virtual roots of f (or Rf) have been well processed.

4.6 Experiments

4.6.1 The illustrative example

Consider as input the sparse polynomial f given as illustrative example in section 1, with $d = 49$. We denote by g the sequence of 50 $\mathbb{F}$-derivatives of f and by Rg the sequence corresponding to the reciprocal Rf of f.

A very fast computation gives:

$sv(f, 0, d) = 24; Count(g, 1, 0, d) = 3; Count(Rg, 1, 0, d) = 5; Count(g, 2, 0, d) = 0; Count(Rg, 2, 0, d) = 1$.

This means that f admits in $\mathbf{R}_+$, 24 $\mathbb{F}$-virtual roots counted with multiplicities (it is of course the same number for Rf but they need not be the same real numbers). Among them 21 are between 0 and 1, and three are between 1 and 2. Respectively, 19 $\mathbb{F}$-virtual roots counted with multiplicities of Rf are between 0 and 1, four are between 1 and 2, and one is greater than 2.

Considering the bijection between the real roots defined by the reciprocal, this implies that f has one real root between 0 and 0.5 and at most four real roots between 0.5 and 1.

After only 6 bissection steps, and without Newton steps, we isolate the three real roots of f between 1 and 2 in $[1.015625, 1.0234375], [1.0234375, 1.03125], [1.0625, 1.125]$.

After only 5 bissection steps, and without Newton steps, we isolate the four real roots of f between 0.5 and 1 in $[0.5, .75], [.9375, .96875], [.984375, .9921875], [.9921875, 1]$.

Together with the root of f between 0 and 0.5 (reciprocal of the one of Rf greater than 2), we can now certify that f has only 8 positive real roots.
We use the subroutine Halley applied 10 times (starting from left and right edges) to get the following approximations:
$[0.1666668837, 0.6909172229, 0.9618369266, 0.9897489318, 0.9977283470, 1.020705838, 1.026238398, 1.063606662]$.

4.6.2 Other examples

We made other experiments with higher degrees and number of monomials.

Example 1: $d = 100, N = 10000$ the integer coefficients are chosen randomly with mean 0 and variance $\binom{d}{i}$. We got:
$sv(f,0,d) = 49; Count(g,1,0,d) = 7; Count(Rg,1,0,d) = 4; Count(g,2,0,d) = 0; Count(Rg,2,0,d) = 0$.
After 13 bissection steps on $[1,2]$, we got tree isolating intervals for real roots of f:
$[1.00390625, 1.0078125], [1.000976562, 1.001953125], [1.000488281, 1.000976562]$
After 10 bissection steps on $[0.5,1]$, we got tree isolating intervals for real roots of f:
$[0.96875, 0.984375], [0.998046875, 0.9990234375], [0.9995117188, 0.9997558594]$.
So in total 6 positive real roots for f.

Example 1bis: $d = 100, N = 100000$ the integer coefficients are chosen randomly with mean 0 and variance $\binom{d}{i}$. We got:
$sv(f,0,d) = 49; Count(g,1,0,d) = 3; Count(Rg,1,0,d) = 4; Count(g,2,0,d) = 0; Count(Rg,2,0,d) = 0$.
After 9 bissection steps on $[1,2]$, 8 bissection steps on $[0.5,1]$, we found 5 positive real roots approximately
0.99831299795; 0.999564942443394; 1.00088846858; 1.00131900658 and 1.00433128542457.

Example 2: $d = 100, N = 10000$ the integer coefficients are chosen randomly with mean 0 and variance 1000. We got:
two positive real roots one in each interval $[0.5,1]$ and $[1,2]$, approximately 0.9995297658 and 1.000014911.

Example 2bis: $d = 100, N = 10000$ the integer coefficients are chosen randomly with mean 0 and variance 1000. We got:
After 13 bissection steps on $[1,2]$, three positive real roots approximately 0.9745463; 1.00005 and 1.0001103.

Example 3: $d = 5, N = 100$, we consider the Mignotte like polynomial: $x^{100} - 625\,x^4 + 1500\,x^3 - 1350\,x^2 + 540\,x - 81$.

For the approximations we used big-floats with 50 digits.
$sv(f,0,d) = 5; Count(g,1,0,d) = 1; Count(Rg,1,0,d) = 4; Count(g,2,0,d) = 0; Count(Rg,2,0,d) = 0$.
After 1 bissection steps on $[1,2]$, we obtain one positive real roots approximated by $x_1 := 1.0312$.
After 10 bissection steps on $[0.5,1]$ the preprocessing diminish the interval but fails to isolate any other root and the processing starts with $a := 0.599609375$, $b := 0.60009765625$, such that $Count(g,a,0,d) = 5$; $Count(Rg,a^{-1},0,d) = 0$; $Count(g,b,0,d) = 1$; $Count(Rg,b^{-1},0,d) = 4$.
This means that in $[a,b]$ there are between 0 and 4 real roots of f. Moreover the list of signs of g evaluated at a, resp. b are $[-1,1,-1,1,-1,1]$, resp $[-1,-1,-1,-1,-1,1]$.
As we can see, there is a root of $g[2]$ in $[a,b]$ it is approximated with one step of Newton-Halley. We get $bb :=$ 0.60000000000000000019563263224;
then $Count(g,bb,0,d) = 2; Count(Rg,bb^{-1},0,d) = 3$.
This means that there is a real root of f in $[bb,b]$ and up to 3 real roots in $[a,bb]$. The root in $[bb,b]$ is approximated, using 10 steps of Newton-Halley, by $x_2 := 0.6000005$. Moreover the list of signs of $g[5], g[4], g[3]$ evaluated at bb is $[1,1,1]$, so there is a root of $g[3]$ in $[a,bb]$, it is approximated with 10 steps of Newton-Halley.
We get $aa := 0.59999999998452259$; then $Count(g,aa,0,d) = 4; Count(Rg,aa^{-1},0,d) = 1$.
This means that there is one real root of f in $[a,aa]$, and up to 2 real roots in $[aa,bb]$ or a virtual non real root.
The root in $[a,aa]$ is approximated, using Newton-Halley, by $x_3 := 0.59999943$.
Since $bb - aa$ is small (about 10^{-11}), an early detection is checked using the subroutine Bound (interval arithmetic), it did not work neither with f nor with Rf (too many cancellations). Then we check the signs of $g[4]$ (and the other derivatives of f, resp Rf) around aa (resp aa^{-1}), letting $aa1 := aa - 10^{-20}$ and $aa2 := aa + 10^{-20}$;
we get: $Count(g,aa1,0,d) = 4; Count(Rg,aa1^{-1},0,d) = 1$, $Count(g,aa2,0,d) = 2; Count(Rg,aa2^{-1},0,d) = 1$. This means that there is a $\mathbb{F}$-virtual non real root of f in $[aa1,aa2]$.

Therefore f admits only 3 positive real roots, approximated by $x_1 := 1.0312$, $x_2 := 0.6000005$ and $x_3 := 0.59999943$ and a $\mathbb{F}$-virtual non real root (it is a root of the second derivative of f) approximated by $aa := 0.59999999998452259$.

5. GENERAL FEWNOMIALS

In [8], the concept of f-derivatives is generalized to functions which are not polynomials but which behave similarly, called fewnomials. For instance, a finite sum of monomials and exponentials multiplied by monomials.

EXAMPLE 1. $f := -2\,x + 3\,x^2 - x^3 + x^5 + \mathrm{e}^x + 5\,\mathrm{e}^x x - 4\,x^2\mathrm{e}^x + 3\,\mathrm{e}^x x^3 - 2\,\mathrm{e}^x x^4$

One can attach to such a function f, a sequence of positive functions over $\mathbf{R}_+$ denoted by f_i and a finite sequence of $\mathbb{F}$-derivatives g_i of f such that the last one is 0. More precisely the i-th differentiation operator D_i satisfies

$$D_i(h) := (h/f_i)'.$$

Therefore the sparse polynomials and the $\mathbb{F}$-derivatives considered in the previous sections are a special case of fewnomials and their $\mathbb{F}$-derivatives.

In the previous example, the sequence of f_i is $[1,1,...,1,\mathrm{e}^x,1,1,...]$. We get $g_{10} := f$ then deriving as usual $g_9 := f'$, ...,

$g_4 := -449\,\mathrm{e}^x - 733\,\mathrm{e}^x x - 310\,x^2\mathrm{e}^x - 45\,\mathrm{e}^x x^3 - 2\,\mathrm{e}^x x^4$

then we divide by e^x, then we derive as usual to get eventually $g_0 := -48$.

Note that in this case the number of $\mathbb{F}$-derivatives (here 10) is not equal to the number of terms (here 9).

All our constructions ($\mathbb{F}$-Budan tables and trees, augmented $\mathbb{F}$-virtual roots, discretizations, etc.) and even the strategy

of the root isolation algorithm could be generalized to fewnomials. This will be the subject of a future work.

6. CONCLUSION

In this paper we described an algorithmic realization of the improved generalized Budan-Fourier count (relying on the concept of $\mathbb{F}$-derivatives) for the case of sparse polynomials. We illustrated it with a step by step description on examples, and emphasized that sparsity was always preserved. The example were computed with a prototype implementation which needs to be developed further.

Our description and illustrative examples suggest that for very sparse polynomials, our new approach can become competitive with Descartes or Sturm-based solvers. A tentative asymptotic complexity analysis indicates that this could be already the case when $N > d^4t$, t being the maximal bit-size of the coefficients. In a future work, we plan to study the influence of the complex non real roots, near the real axis, on the generalized Budan-Fourier count.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. The first author was partially supported by Spanish organizations: IMI(UCM), UCM (Grupo 910444) and MEC (MTM2011-22435). The second author 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.

7. 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] Basu,S and Pollack,R and Roy,M.-F.: Algorithms in real algebraic geometry, Springer (2003).

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

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

[5] 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 edited by the Académie des Sciences (1811).

[6] Collins,G.E and Loos,R: Polynomial Real Root Isolation by Differentiation, Proc. 1976 ACM Symposium on Symbolic and Algebraic Calculation, (1976).

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

[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 conference, pages 71-78, ACM NY, (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] Galligo, A: Deformation of Roots of Polynomials via Fractional Derivatives. Submitted J. Symb. Comp. (Oct.2011).

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

[13] Galligo, A: Improved Budan-Fourier count for Root Finding. In preparation

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

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

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

[17] Pan, V: Solving a polynomial equation: some history and recent progress. SIAM Review, 39(2):187-220, (1997).

[18] Pan, V: Univariate polynomials: Nearly optimal algorithms for numerical factorization and rootfinding. J. Symbolic Computation, 33(5):701-733, (2002).

[19] Pan,V and Brian Murphy,B and Rosholt,R.E and Guoliang Qian and Yuqing Tang: Real root-finding. Proceedings SNC 2007: 161-169, ACM (2007).

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

[21] Rouillier,F and Zimmermann,F: Efficient isolation of polynomialÂŠs real roots. J. Computational and Applied Mathematics, 162:33-50, (2004).

[22] Sagraloff,M: When Newton meets Descartes: A simple and fast algorithm to isolate the real roots of a polynomial. ArXiv [cs.SC], Sept 2011, to appear in Proceedings of Issac'2012.

[23] Sagraloff,M and Yap,C.-K: A simple but exact and efficient algorithm for complex root isolation. In Proceedings of Issac'2011 pages 353-360, ACM, (2011).

[24] Sturm,C: Mémoire sur la résolution des équations num'eriques, presented (but lost) at the Académie des Sciences, (1829).

[25] Tsigaridas,E and I. Z. Emiris,I: On the complexity of real root isolation using continued fractions. Theor. Comput. Sci., 392(1-3):158-173, (2008).

[26] Vincent, M: *Sur la résolution des équations numériques*, Journal de mathématicques pures et appliquées **44** 235-372 (1836).

[27] Zhang,T and Xia,B: A New Method for Real Root Isolation of Univariate Polynomials, Mathematics in Computer Science, Volume 1, Number 2, 305-320,(2007).