Infinitesimals and real closure

Renaud Rioboo

Infinitesimals and Real Closure

Renaud Rioboo
Laboratoire D'Informatique de Paris 6
Boite 168
Universtité Paris VI
4 Place Jussieu
F-75252 Paris CEDEX 05
rioboo@calfor.lip6.fr

November 20, 2001

Abstract

The purpose of this paper is to offer an alternative to Thom's coding ([2]) for real algebraic numbers when working over fields that contain infinitesimals. For this main case of non-archimedean ordered fields we modify Newton-Puiseux method ([4], [11]) to separate and work with the distinct real roots of a polynomial. An *Axiom* implementation describing these methods is then presented.

Authors note

This paper is an unedited version of a submission at the ISSAC98 conference. Early versions of this paper were presented but not published at the MEGA94 conference. This work has already been presented at

- séminaire de l'équipe de calcul formel du LIP6,
- séminaire de l'IRMAR,
- grupo de algebra computacional de la universidad de Santander.

Introduction

The real closure of an ordered field is uniquely determined and can be constructively computed using different techniques (see [7], [6], [2] for instance). However the only generic technique that enables to start with any computable ordered field is the one known as Thom's coding ([2]). We want in this paper offer another method based on approximation ideas. Current implementations seem to show that, for the particular case of archimedean fields, approximation methods are somewhat faster than generic methods.

This paper is an effort to take advantage of this fact for the case of fields containing "infinitesimal quantities" which often appear in practice and are the main example of non-archimedean fields. However techniques described here are not interval-based but use Puiseux series.

Using the general scheme of [8] we can reduce the construction of the real closure of an ordered field $\mathbb{K}$ to few basic operations. We only need to give a data structure to code one particular root α of a polynomial $P \in \mathbb{K}[X]$, and a method to compute the sign of $Q(\alpha)$ where $Q \in \mathbb{K}[X]$ and thus $Q(\alpha) \in \widetilde{\mathbb{K}}$ where $\widetilde{\mathbb{K}}$ is the real closure of $\mathbb{K}$.

After some basic recalls, we will describe the data structure we need to compute with algebraic infinitesimals. This is mainly the singular part of the Puiseux expansion of the algebraic viewed as an algebraic function of the infinitesimal. Sign related operations will sometimes need to produce "finer" approximations, we produce those computing the regular part of the serie using Newton method.

Knowing how to do basic arithmetics, we then adapt Newton-Puiseux method for our case where we only need to compute some of all the possible expansions. We restrict ourselves to the case of computing one edge of the *Newton polygon* of a polynomial. Knowing how to work in a real closed field containing one infinitesimal, we then explain how the concept of *towers* can be used to compute with several ones.

1 Preliminaries

1.1 Fields with infinitesimals

Let $\mathbb{K}$ be an ordered field, we want to deal with quantities that depend on a new variable ϵ. It is a well known property of real fields that $\mathbb{K}(\epsilon)$ (the rational fractions in ϵ field) is orderable, and that if we choose the rule :

$$\mathcal{O}r \ : \ \begin{cases} \epsilon > 0 \\ \forall x \in \mathbb{K} \ , \ x > 0 \Rightarrow \epsilon < x \end{cases}$$

then $\mathbf{K}(\epsilon)$ is ordered. Up to a $\mathbb{K}$-linear change of variables any rule for ϵ can be deduced from this one. The rule $\mathcal{O}r$ also enables to order the Laurent and Puiseux series in ϵ. The latter is real closed whenever $\mathbf{K}$ is real closed as shown in [9].

Let $\mathbb{K}$ be an ordered field an $\mathbb{K}_\epsilon$ a field containing $\mathbb{K}$. We say that $\mathbb{K}_\epsilon$ is an *infinitesimal extension* of $\mathbb{K}$ if it is a transcendental extension of $\mathbb{K}$ with a *limit* functionality. Let a_ϵ be in $\mathbb{K}_\epsilon$, we interpret $\text{limit}(a_\epsilon)$ as the usual limit of $a(\epsilon)$ as a continous function of ϵ when ϵ goes to 0^+. We model the fact $\mathbf{K}_\epsilon$ contains only infinitesimals over $\mathbf{K}$ by saying that the target of the limit function is an element of $\mathbf{K} \cup \{\infty\}$. An element a_ϵ of $\mathbb{K}_\epsilon$ is then said to be *finite* if the limit of a is in $\mathbb{K}$; a is said to be *vanishing* if the limit of a is zero. We model this functionality in an *Axiom* category *InfinitesimalExtension*($\mathbb{K}$) where $\mathbb{K}$ is an ordered field.

We now need to model the notion of an extension that contains a single infinitesimal, and following [11] we use a notion of *order* of an element. To make a distinction between infinitesimal extensions where non trivial algebraic relations hold we will consider two maintargets for the order function :

- the integers and the extension will be called *rational*,
- the set of rational numbers and the extension will be called *algebraic*.

As of *Axiom* implementation we have found usefull to abstract the target domain of the function and we defined a category

$$\textit{SingleInfinitesimalExtension}(\mathbb{K} \ , \mathcal{R})$$

where $\mathbb{K}$ is the base ordered field $\mathbb{K}$ and the target $\mathcal{R}$ is a subring of the rational numbers. It then exports the following operations :

```
generator :  R -> %
order :  % -> R
expandOrders :  (%,PositiveInteger) -> %
mainCoefficient :  % -> K
```

together with the operations of an infinitesimal extension of $\mathbb{K}$. Here the `expandOrders` function computes the result of the change of variables $\epsilon \rightarrow \epsilon^n$. We remark that the expandOrders operation leaves $\mathbb{K}$ stable.

Two sample and straightforward implementations have been provided for rational infinitesimal extensions. These are the two *Axiom* domain constructors :

- *ZeroPlusFractions*($\mathbf{K}$) which represents elements as rational fractions in one variable and
- *ZeroPlusSeries*($\mathbb{K}$) which represents elements as Laurent series in one variable.

These serve as a base for real infinitesimal algebraic manipulation that we now describe.

1.2 Towers of extensions

The generic real closure construction of [8] is a tower manipulating program with external data structure operations. The real closure $\widetilde{\mathbb{K}}$ of an ordered field $\mathbb{K}$ is the union of all the possible real finite degree extensions of $\mathbb{K}$. However each member x of $\widetilde{\mathbb{K}}$ belongs to some (or several) finite degree extension of $\mathbf{K}$, and the representation of x will include the necessary informations to describe and manipulate this finite extension, also called *tower*.

Basically x will have three parts in its representation :

- a member of some external data structure representing some *real algebraic variable* χ. This data structure is also responsible of :
  - creating the representations for all the roots $\chi_1, \ldots \chi_n$ of any univariate polynomial P,
  - computing the sign of any univariate polynomial expression Q at an algebraic variable χ.

- A univariate polynomial *value* expressing x in terms of χ.
- A integral *age* identifying χ, used to manipulate the towers of real algebraic variables.

The only requirements needed for the coefficients of the univariate polynomials involved in this scheme is that their coefficients should be real algebraic with lower ages (ie already defined, or "older"). The program then works manipulating towers, comparing ages, and calling either polynomial operations (for arithmetics), either external (for sign related operations) functions; see [8] for details.

This contruction has the advantage to reduce the degrees of the polynomials involved, as an example, if one wants to work with one root of $X^2+\sqrt{2}X-\sqrt{3}$, the heighth of the tower would be 3, each defining polynomial being of degree two, while the degree of the necessary simple extension would be 8 (2 for $\sqrt{2}$, 4 for an extension containing both $\sqrt{2}$ and $\sqrt{3}$, and 8 for the resulting root). Moreover, it is much easier to produce new real algebraic algorithms, since the implementation task is reduced to a minimal one.

An *Axiom* implementation is available for these towers, we will use this simple scheme here, thus only decsribing the data structure, algorithms for sign related operations, and a method to create roots of a polynomial using this data structure. We thus need to describe real infinitesimal algebraic variables manipulation and we found usefull to model these through an *Axiom* category :

$$SingleInfinistesimalRootCodingCategory(\mathbb{K}, \mathbb{K}_\epsilon, \mathbb{K}_\epsilon[X])$$

where $\mathbb{K}$ is the base real closed field, $\mathbb{K}_\epsilon$ is a single infinitesimal extension of $\mathbb{K}$ and $\mathbb{K}_\epsilon[X]$ a domain of univariate polynomials over $\mathbb{K}_\epsilon$. This category inherits $RootCodingCategory(\mathbb{K}_\epsilon, \mathbb{K}_\epsilon[X])$ of [8] together with the operations :

```
expand :  (%, PositiveInteger) -> %
```

```
allRootsNear :  (K_ε[X], K) -> List %
```

The purpose of the `expand` operation is to reflect the change of variables $\epsilon \to \epsilon^k$ at a real infinitesimal variable. The purpose of the `allRootsNear` operation is to provide a functionality to describe all the infinitesimal real algebraic variables whose limit is a given point of $\mathbb{K}$. Other operations are like in [8].

2 Coding infinitesimal real algebraic variables

Let $\mathbb{K}$ be a real closed field, we will in this section be concerned with the problem of manipulating algebraic infinitesimal variables using their Puiseux serie expansion.

If we view a infinitesimal real algebraic variable as a number x_ϵ of the real closure of $\mathbb{K}_\epsilon$, it has a Puiseux expansion over $\mathbb{K}_\epsilon$ ($\mathbb{K}$ is real closed) :

$$x_\epsilon = \sum_{i \in I} x_i \epsilon^{i/c} \tag{1}$$

where I is a subset of the integers with $\min(I)$ finite, and c is a positive integer. Note in this definition that $\epsilon^{1/c}$ is perfectly defined and unique in terms of real closed fields. It is another infinitesimal η bigger than ϵ verifying the relation : $\eta^c = \epsilon$. Working in $\mathbb{K}_\eta$ is done the same way than working over $\mathbb{K}_\epsilon$ since the canonical injection of $\mathbb{K}_\epsilon$ in $\mathbb{K}_\eta$ is order preserving. We use the expandOrders operation to explicit this morphism while remaining in the same domain of computations. A sufficient approximation for our purposes consists in the singular part of a Puiseux expansion around some point of the curve of $\mathbb{K}^2$ defined by $P(\epsilon, X) = 0$.

2.1 Representation

We will use "lazy approximation" techniques to represent an algebraic infinitesimal, roughly speaking an algebraic infinitesimal will be the begining of its Puiseux serie together with a way to compute more terms. More precisely we will call $\mathcal{I}$ the *state* of a real algebraic infinitesimal χ and we will encode it as :

- a *contraction factor* $c_\mathcal{I} \in \mathbb{N}^*$,
- an *offset* $m_\mathcal{I} \in \mathbb{Z}$,
- an *approximation* $A_\mathcal{I} \in \mathbb{K}[\eta_\mathcal{I}]$, where $\eta_\mathcal{I}$ is interpreted as $\epsilon^{1/c_\mathcal{I}}$.
- To compute more terms we will use a polynomial $P_\mathcal{I}$ of $\mathbb{K}_{\eta_\mathcal{I}}[X]$ with only one root (real or complex) "nearby" its leading coefficient $a_\mathcal{I}$, more precisely :

$$\text{(I)} \ : \ \left\{ \begin{array}{lcl} P_{\mathcal{I}(\eta_\mathcal{I}=0)}(a_\mathcal{I}) & = & 0 \\ P'_{\mathcal{I}(\eta_\mathcal{I}=0)}(a_\mathcal{I}) & \neq & 0 \text{ or } P_\mathcal{I} = 0 \end{array} \right. .$$

Thus, again viewing $P_{\mathcal{I}}$ as a curve of $\mathbb{K}^2$, the point ($\eta_{\mathcal{I}} = 0$, $X = a_{\mathcal{I}}$) should verify the conditions of implicit functions theorem.

We add a *defining polynomial* P_χ for χ to the representation of an infinitesimal real algebraic variable. This information is redundant but usefull in practice.

Let χ be an infinitesimal real algebraic variable and $\mathcal{I}$ be a state of χ. The *level* of $\mathcal{I}$ will be the integer $\mid P_{\mathcal{I}} \mid + m_{\mathcal{I}}$ and the level of χ will be the level of its current state divided by $c_{\mathcal{I}}$ (a rational number)

2.2 Approximations

As in any approximation technique, it is sometimes needed to produce a "finer" approximations, let us call *refine* this operation. We will produce a new state of *chi* since the defining polynomial does not change We will classically use Newton's approximations to compute it :

- a new contraction factor $c_{\mathcal{I}'}$,
- a new offset $m_{\mathcal{I}'}$
- a new approximation $A_{\mathcal{I}'}$ of higher level than $A_{\mathcal{I}}$,
- a new $P_{\mathcal{I}'} \in \mathbb{K}_{\eta_{\mathcal{I}}}[X]$ verifying (I) as above.

We write the Taylor expansion in X around $a_{\mathcal{I}}$:

$$P_{\mathcal{I}}(\eta_{\mathcal{I}}, X + a_{\mathcal{I}}) = P_{\mathcal{I}}(\eta_{\mathcal{I}}, a_{\mathcal{I}}) + X\frac{\partial P_{\mathcal{I}}}{\partial X}(\eta_{\mathcal{I}}, a_{\mathcal{I}}) + XF(\eta_{\mathcal{I}}, X) \tag{2}$$

with $F(\eta_{\mathcal{I}}, 0) = 0$. We assume $P_{\mathcal{I}}$ is not null (otherwise the approximation is exact and we can return a dummy result), we have :

$$P_{\mathcal{I}}(\eta_{\mathcal{I}}, a_{\mathcal{I}}) = {\eta_{\mathcal{I}}}^q(a + f_q(\eta_{\mathcal{I}})) \tag{3}$$

with $f_q(0) = 0$ and where $q = n/d$ is the order (in $\mathbb{K}_{\eta_{\mathcal{I}}}[X]$) of $P_{\mathcal{I}}(\eta_{\mathcal{I}}, a_{\mathcal{I}})$ and $a \neq 0$. We now write the expansion :

$$\frac{\partial P_{\mathcal{I}}}{\partial X}(\eta_{\mathcal{I}}, a_{\mathcal{I}}) = b + g_q(\eta_{\mathcal{I}})$$

with $b \neq 0$ (in $\mathbb{K}$) and with the order of $g_q(\eta_{\mathcal{I}})$ strictly positive. We will report the latter together with (3) in (2) where X becomes ${\eta_{\mathcal{I}}}^n X$ and $\eta_{\mathcal{I}}$ becomes ${\eta_{\mathcal{I}}}^d$ obtaining :

$$\begin{aligned} P_{\mathcal{I}}({\eta_{\mathcal{I}}}^d, {\eta_{\mathcal{I}}}^n X + a_{\mathcal{I}}) &= P_{\mathcal{I}}(0, a_{\mathcal{I}}) \\ &+ {\eta_{\mathcal{I}}}^n \quad a + Xb \\ &+ {\eta_{\mathcal{I}}}^n \quad G(\eta_{\mathcal{I}}, X). \end{aligned}$$

with $G(0, 0) = 0$. We can now state :

- $m_{\mathcal{I}'} = m_{\mathcal{I}} d$
- $A_{\mathcal{I}'} = \dfrac{-a\ {\eta_{\mathcal{I}}}^{|A'_{\mathcal{I}}|+1}}{b} + A'_{\mathcal{I}}$ where $A'_{\mathcal{I}}$ is obtained by multiplying all the degrees of the terms of $A_{\mathcal{I}}$ by a factor d and
- $P_{\mathcal{I}'}(\eta_{\mathcal{I}}, X) = \dfrac{P_{\mathcal{I}}({\eta_{\mathcal{I}}}^d, X + a_{\mathcal{I}})}{{\eta_{\mathcal{I}}}^n}$ this involves again *expandOrders* on the coefficients of $P_{\mathcal{I}}(\eta_{\mathcal{I}}, X)$ viewed as a polynomial in X.

We clearly have the conditions (I).

The
sl expand operation is performed in a similar way multiplying the degrees in the approximation recalling the expandOrders function on the terms of $P_{\mathcal{I}}$.

2.3 Sign computations

Now that we can approximate an infinitesimal algebraic up to any precision we need, we can go on with sign computations. These operations mainly consist of a zero check and a sign computation. As in [10], computations remain ordered and the zero check depends on the particular root of a polynomial. We will have two basic approaches when computing the nullity or the sign of $Q_\epsilon(\chi)$ where χ is a infinitesimal real algebraic variable that encodes some root of $P_\epsilon(X)$ coded as above:

- use the defining polynomial P_ϵ first, and use the state $\mathcal{I}$ of χ after; or
- use the state $\mathcal{I}$ first, and use $P_\mathcal{I}$ after.

The first approach is faster when using as base field rational fractions in ϵ and the latter seems faster when using Laurent series as base field.

2.3.1 Zero check

The zero check is computed using a *polynomial first* approach:

- compute the gcd $R = \gcd_X(P_\epsilon, Q_\epsilon)$, this is a polynomial of $\mathbb{K}_\epsilon[X]$;
- if R is 1 we return **false**;
- otherwise, evaluate R over $(\epsilon = {\eta_\mathcal{I}}^{c_\chi})$, giving R_1 a polynomial of $\mathbb{K}_{\eta_\mathcal{I}}[X]$;
- evaluate R_1 over $(X = A_\mathcal{I})$, this is done interpreting $A_\mathcal{I}$ in $\mathbb{K}_\epsilon[X]$. This gives a result v which is a number of $\mathbb{K}_{\eta_\mathcal{I}}$;
- if the order (in $\mathbb{K}_{\eta_\mathcal{I}}$) of v is greater than the level of the state $\mathcal{I}$ of χ or if v is null, we return **true**, and **false** otherwise.

The method gives the right result since to be null a term must have a non trivial gcd with the defining polynomial, and when evaluated over an approximation of the root, it must have an order bigger than that of the approximation.

2.3.2 Proper sign computation

For sign computation we must refine the approximation until it has a correct order. Current implementation uses the state first approach and lazy evaluation for the order of an element:

- evaluate Q over $(\epsilon = {\eta_\mathcal{I}}^{c_\chi})$, giving Q_1 a polynomial of $\mathbb{K}_{\eta_\mathcal{I}}[X]$;
- $Q_1 \leftarrow Q_1(X = {\eta_\mathcal{I}}^{m_\chi} X)$
- $n \leftarrow |\, A_\chi \,|$;
- let l be the list of non null terms of A_χ. This list is of the form $a_d {\eta_\mathcal{I}}^d$ with $a_d \neq 0$ (in $\mathbb{K}$) sorted by increasing order of degrees, let i be 0;
- foreach term of l do:
  - $i \leftarrow d - i$,
  - $Q_1 \leftarrow Q_1(X = {\eta_\mathcal{I}}^i X + a_d)$
  - if the order of $(Q_1(0))$ is less than the level od the state $\mathcal{I}$ of χ and if $Q_1(0)$ is not null we return its sign.
- Now $Q(\mathcal{I})$ is "negligible" up to the level of χ, we compute $R = \gcd_X(P_\mathcal{I}, Q_1)$ a polynomial of $\mathbb{K}_{\eta_\mathcal{I}}[X]$;
- if R non trivial then we return 0; now we must refine the approximation untill the order of $Q_1(0)$ is smaller than the level of the state:
- while the order of $(Q_1(0))$ is bigger than the level of (χ) repeat
  - let $\mathcal{I}$ be the state of χ;
  - refine χ; let $\mathcal{I}'$ be the new state of χ
  - $n \leftarrow |\, A_{\mathcal{I}'} \,|$ and $a_n \leftarrow \text{leadingCoefficient}(A_{\mathcal{I}'})$
  - $Q_1 \leftarrow Q_1({\eta_{\mathcal{I}'}} = {\eta_\mathcal{I}}^{c_{\mathcal{I}'}/c_\mathcal{I}}, X = \eta_\mathcal{I} X + a_n)$; here $c_{\mathcal{I}'}/c_\mathcal{I}$ remains a positive integer.
- return $\text{sign}(Q_1(0))$

The first loop lazily computes the same transformation that is done in P_ϵ to obtain $P_{\mathcal{I}}$ with an exit when the sign is "clear", that is when the term is significant. The method terminates since when we reach the second loop the polynomials are relatively prime thus the roots of each polynomial have different Puiseux expansions.

We can now compute sign related operations over our data structure, and leave to the generic manipulation program the other arithmetic operations. We are left with the problem of finding an approximation verifying the conditions (I).

3 Producing infinitesimal real algebraic variables

Let P be a polynomial of $\mathbb{K}_\epsilon[X]$, we consider the problem of finding a an encoding for each root of P.

3.1 Presentation

For a polynomial P of $\mathbb{K}_\epsilon[X]$ we call its *non-vanishing part* of P the polynomial of $\mathbb{K}_\epsilon[X]$ obtained by selecting in P those terms with a non negative order coefficient. The *main part* of P will then be obtained by taking the limit (in $\mathbb{K}$) of the coefficient in the terms in its non-vanishing part. This is a polynomial of $\mathbb{K}[X]$.

We will describe our functionalities allRootsNear and allRootsOf by means of two other functions which work only with a given state $\mathcal{I}$. These internal operations will forget the offset factor and will thus need to be called properly. These are :

- a *findFiniteRoots* operation. It takes as input a square free polynomial P of $\mathbb{K}_\epsilon[X]$. The main part of P should have a non null (in $\mathbb{K}$) constant term and another term of order 0. It returns the necessary information to describe the state $\mathcal{I}$ of an infinitesimal real algebraic variable but without its offset part.
- a *findRootsNear* operation. It takes as input a square free polynomial P of $\mathbb{K}_\epsilon[X]$ and a number x of $\mathbb{K}$ which is a root of the main polynomial P of $\mathbb{K}[X]$. P should have a non constant term whose coefficient is of order 0. In this way the main polynomial of P is non constant, it return the same kind of information than findFiniteRoots.

This information can be viewed as a state with a 0 offset part, we will call it a *non-vanishing state*. Let P be a polynomial verifying the conditions for the input of findFiniteRoots and let P_0 be its main part.. If P_0 has no roots in $\mathbb{K}$ then P has no non-vanishing roots in the real closure of $\mathbb{K}_\epsilon$ (since an element of the real closure has a Puiseux expansion). Otherwise let x be a root of P_0 and let m be its multiplicity. We can write :

$$P(\epsilon, X + x) = \sum_{i=0}^{i=m} \epsilon^{q_i}(a_i + f_i(\epsilon))X^i + X^m Q(\epsilon, X)$$

with $Q(\epsilon, 0) = 0$ (in $\mathbb{K}_\epsilon$). The a_i and q_i are the main coefficient and order of the terms of $P(\epsilon, X + x)$. Since P is non-vanishing we have $q_i \geq 0$. Since m is the multiplicity of x in $P_{(\epsilon=0)}$ we know that $q_m = 0$ and $a_m \neq 0$. Since $P_{(\epsilon=0)}$ has non null constant term we know that $q_0 > 0$ and $a_0 \neq 0$. We write $q_0/m = \frac{n}{d}$ where n and d are relatively prime positive integers.

Again following [11] we will make a change of variables as in 2.2 stating $\epsilon = \epsilon^d$ and $X = \epsilon^n X$. We now have :

$$P(\epsilon^d, \epsilon^n X + x) = \epsilon^{md}\left(\sum_{i=0}^{i=m} \epsilon^{(q_i - q_0)i}(a_i + f_i(\epsilon^d))X^i + X^m Q(\epsilon^d, \epsilon^n X)\right)$$

It is now clear that if m is 1 we have finished since our conditions for unique encoding of the state are met. Otherwise, $P(\epsilon^d, \epsilon^n X + x)$ is square free and its main part meets the requirements for the findFiniteRoots operation. We can view that after a finite number of steps the multiplicity m reaches 1. Otherwise there would exist a sequence $x_0 \ldots x_n \ldots$ of $\mathbb{K}$ and a sequence $q_0 \ldots q_n \ldots$ of rational numbers such that, if we let s be the serie $\sum_n x_n \epsilon^{q_n}$ is a root of P. This is impossible since at each step the findFiniteRoots multiplies the contraction factor $c_{\mathcal{I}_n}$ of the state $\mathcal{I}_n$ by a factor bigger than one. We thus conclude that s is not a Puiseux serie (denominators are not stationary) which is impossible since these form a real closed field. Uniqueness of the result is ensured by the fact that if two Puiseux series differ their regular part differ. Finally we find all the possible finite roots since the singular part of a Puiseux serie with coeffients in $\mathbb{K}$ is made of only positive order terms with coefficient in $\mathbb{K}$.

We will remark that our presentation slightly differs from [11] in some important points :

- we return a result that is lazier since we don't necessarily require to compute the whole regular part of the serie.
- Our presentation clearly shows the dependence in terms of the multiplicity of a given root of the main part of a polynomial.

3.2 Methods used

We can now describe our operations. The purpose of the *findRootsNear* operation is to detect the case $m = 1$ and to recall *findFiniteRoots*. To compute *findFiniteRoots*(P, x) we do :

- compute the multiplicity m of x in the main part of P by changing variables $X \to X + x$ and taking the least degree non null term that has order 0. We are sure that m exists since P is assumed to have one order zero non null coefficient in its terms. We now have a polynomial $Q(\epsilon, X)$
- if m is 1 we are done ; return a state (see 2.2) $\mathcal{I}$ such that :
  - $c_{\mathcal{I}} = 1$
  - $m_{\mathcal{I}} = 0$
  - $A_{\mathcal{I}} = x$
  - $P_{\mathcal{I}} = Q$
- otherwise compute n and d as in 3.1 and compute $Q(\epsilon, X) = Q(\epsilon^d, \epsilon^n X + x)$
- let l be the list of roots of P_0 ; and l_r be the empty list
- foreach x in l, let l_x be the list of states returned by *findRootsNear*(P, x)
- for each state $\mathcal{I}$ of l_x we produce a new state $\mathcal{I}'$ such that
  - $c_{\mathcal{I}'} = dc_{\mathcal{I}}$,
  - $m_{\mathcal{I}'} = m_{\mathcal{I}}$,
  - $A_{\mathcal{I}'} = \eta_{\mathcal{I}}^{n+|A'_{\mathcal{I}}|} + A'_{\mathcal{I}}$ where $A'_{\mathcal{I}}$ is as above
  - $P_{\mathcal{I}'} = P_{\mathcal{I}}$

  and append $\mathcal{I}'$ to l_r.
- return l_r.

And we simply compute *findFiniteRoots*(P) by :

- let P_0 be the main part of P
- append the results of *findFiniteRoots*(P, x)

We can now describe our *allRootsOf* and *allRootsNear* functionalities. These two operations begin by taking the square free part (in $\mathbb{K}_\epsilon[X]$) of their polynomial argument.

To compute *allRootsOf*(P) we first make P monic and then compute the minimum of the orders of $\frac{a_i}{N-i} = n/d$. Here n is an integer relatively prime to p which is a positive integer and where N is the degree of P. The numbers $a_i(\epsilon)$ $(\in \mathbb{K}_\eta)$ are the non null coefficients of degree i in the reductum of P. We then call *findRootsNear*$(Q, 0)$ where Q is $\frac{P(\epsilon^d, \epsilon^n X)}{\epsilon^{nN}}$. We simply have to adapt the state $\mathcal{I}$ returned as above.

To compute *allRootsNear*(P, x) we use the same method described in the *findRootsNear* function, thus recalling *findFiniteRoots* when the multiplicty is bigger than one.

3.3 Implementation

We have put these functionalities together in an *Axiom* domain contructor *PuiseuxRealRootCoding*$(\mathbb{K}, \mathbb{K}_\epsilon, \mathbb{K}_\epsilon[X])$ where $\mathbb{K}$ is any implementation of a real closed field, $\mathbb{K}_\epsilon$ is a single infinitesimal algebraic extension of $\mathbb{K}$ and $\mathbb{K}_\epsilon[X]$ is any instance of a univariate polynomial constructor.

We also implemented a functionality *mainPart*(P, χ) that computes both the order and the main coefficient of the expression $P(\chi)$. Here P is a polynomial of $\mathbb{K}_\epsilon[X]$ and χ is an infinitesimal real algebraic variable. The technique we used is analogous than that described for sign computations (2.3.2).

As of other implementations, the techniques described in [4] or [5] compute (in $\mathbb{K}$) roots of smaller degree polynomials by grouping together all the Puiseux developments that have the same arithmetic

part by grouping them together. In ([3]) the situation is adapted to the real case that is expressing both $\eta_{\mathcal{I}}$ and X in terms of real algebraic numbers on the ground field. These have the drawback of requiring dynamic evaluation thus hiding properties to the users of the algorithms. On the contrary we acheive ordered field computations. We offer the same kind of functionalities by having sparsers series (we do not compute the full regular part) and we rely on the tower mechanism to efficiently manage the algebraics. Furthermore [4] or [5] have not been implemented in a recent computer algebra we could not do any comparison. This might be feasible using dynamic evaluation support for *axiomxl*. The techniques of [3] have never been implemented.

4 Real closures with infinitesimals

4.1 Adding one infinitesimal

Now that we can perform our basic manipulations we have to give an implementation of the real closure using these operations. To describe it we use a category

$$SingleInfinitesimalRealClosedField(\mathbb{K})$$

which inherits from the two categories *RealClosedField* and

$$SingleAlgebraicInfinitesimalExtension(\mathbb{K})$$

and that adds an operation $allRootsNear(P, x)$ where P is a polynomial with coefficients over the closure itself and x is an element of $\mathbb{K}$. It returns a list of algebraic numbers and can be usefull for applications such as curve analysis.

We have an *Axiom* domain constructor

$$SingleInfinitesimalRealClosure(\mathbb{K}, \mathbb{K}_\epsilon)$$

where $\mathbb{K}$ a real closed field and $\mathbb{K}_\epsilon$ is a single rational infinitesimal extension $\mathbb{K}_\epsilon$ of $\mathbb{K}$ (see 1.1). We provide the real closure described in [10] as sample implementation for $\mathbb{K}$, and the implementations mentioned in 1.1 as single rational infinitesimal extensions of $\mathbb{K}$.

Our implementation for $\widetilde{\mathbb{K}_\epsilon}$ uses an infinitesimal real algebraic variable domain which is :

$$PuiseuxRootCoding(\mathbb{K}, \widetilde{\mathbb{K}_\epsilon}, \widetilde{\mathbb{K}_\epsilon}[X])$$

Here $\widetilde{\mathbb{K}_\epsilon}[X]$ is the domain of sparse univariate polynomials with coefficients over $\widetilde{\mathbb{K}_\epsilon}$. Elements of $\widetilde{\mathbb{K}_\epsilon}$ are then represented as either an element of $\mathbb{K}_\epsilon$ either as structures as in 1.2 and most operations proceed like in [8].

The implementation of the *allRootsOf* and *allRootsNear* operations are straightforward interfaces with the corresponding operations in the infinitesimal real algebraic variable domain. The two functions *mainCoefficient* and order are easilly implemented using the *mainPart* functionality :

The *order* of an expression is computed either in $\mathbb{K}_\epsilon$. Otherwise the expression is of the form $Q(\chi)$ at an infinitesimal real algebraic variable and its order is computed in the coding domain. The limit and mainCoefficient functions also use this functionality.

To perform the *expandOrders* operation we again proceed recursively, by either calling the *expandOrders* function in $\mathbb{K}_\epsilon$, either calling the *expand* operation in the coding domain.

4.2 Using various infinitesimals

These techniques enable us so far to add one infinitesimal to a real closed field producing another real closed field. We are now able to work in the real closure $\widetilde{\mathbb{K}_\epsilon}$ of $\mathbb{K}_\epsilon$ if $\mathbb{K}_\epsilon$ is a rational infinitesimal extension of $\mathbb{K}$ which is a real closed field. This possibility enables to go further and we can sequentially add infinitesimals one above the other. We can first add an ϵ_1, and then an $\epsilon_2 << \epsilon_1$ meaning that ϵ_2 is smaller than all the rational powers of ϵ_1. We would thus work in $(\widetilde{\mathbb{K}_{\epsilon_1}})_{\epsilon_2}$. More generally, we could use use a tower structure similar than that of 1.2 and work in $\mathbb{K}_\infty$ that is a real closed field containing the real closed field $\mathbb{K}$ and all the necessary infinitesimals we need. We could chose a particular implementation for $\mathbb{K}_\epsilon$ (*ZeroPlusFractions* as in 1.2) and represent elements of the infinitesimal real algebraic closure $\mathbb{K}_\infty$ of $\mathbb{K}$ as :

- either trivial, that is an element of the base real closed field $\mathbb{K}$.
- either a structure containing two parts :
  - a value giving the expression in terms of an infinitesimal real closed variable. This is a element of $ZeroPlusFractions(\mathbb{K}_\infty)$.
  - tower manipulating informations similar than that of 1.2.

The tower manipulating information being dynamically managed as it is done in [8].

This possibility could however not be tested, and future work should include this feature as well as netter mangement of algebraics than that outlined here. One important cost is the cost of the gcd operation involved in zero check and sign computations. We hope that in a very near future we will be able to have an implementation including these enhancements which are still being debugged.

References

[1] J. Bochnak, M. Coste, M.F. Roy: *Géométrie algébrique réelle*
Springer-Verlag, New-York 1988.

[2] M. Coste, M. F. Roy, *Thom's Lemma, the coding of Real Algebraic Numbers and the Computation of the Topology of Semi-Algebraic Sets*
In *Journal Of Symbolic Computation*, 1988 vol 5, pp. 121–129.

[3] F. Cucker, L. M. Pardo, M. Raimondo, T. Recio, M. F. Roy: *Computation of the local and global analytic structure of a real algebraic curve*
In *proceedings of the AAECC-5 Conference* Berlin 1988, pp 161-181.

[4] D. Duval: *Diverses questions relatives au calcul formel avec des nombres algébriques*
Thèse d'Etat à l'Université de Grenoble, Grenoble 1987.

[5] D. Duval: *Rational Puiseux expansions*
Compositio Mathematica 70, 1989 pp 119-154.

[6] A. Frohlich, J. C. Shepherdson: *Effective Procedures In Field Theory*
In *Phil. Trans. Roy. Soc.* London 1956, pp. 407–432.

[7] A. Hollkott: *Finite Konstruktion geordneter algebraischer Erweterungen von geordneten Grundkorpern*
Dissertation, Hamburg 1941.

[8] Z. Ligatsikas, R. Rioboo, M. F. Roy: *Generic computation of the real closure of an ordered field.*
In *Mathematics and Computers in Simulation Volume 42, Issue 4-6* November 1996

[9] Z. Ligatsikas, M. F. Roy: *Séries de puiseux sur un corps réel clos.*
C.R Acad. Sci. Paris 1990 Tome 311 Série I pp. 625-628.

[10] R. Rioboo: *Real Algebraic Closure of an ordered Field: Implementation in Axiom*
In *proceedings of the ISSAC'92 Conference*, Berkeley 1991 pp. 206-215.

[11] R. J. Walker: *Algebraic curves*
Dover publications, 1950.