Real Root Isolation of Regular Chains.

François Boulier[1], Changbo Chen[2],
François Lemaire[1], Marc Moreno Maza[2]

[1]University of Lille I (France)

[2]University of London, Ontario (Canada)

ASCM 2009

Plan

Plan

Real Root Isolation

Goal

Isolate the real roots of a zero-dimensional algebraic variety $V \subset \mathbb{C}^n$ given by a **regular chain** C.

Example

$C = \{x^2 - 2, xy^2 - 1\}$

EXACT REAL ROOTS:

$\{x_1 = \sqrt{2}, y_1 = \sqrt{\frac{\sqrt{2}}{2}}\}$ and $\{x_2 = \sqrt{2}, y_2 = -\sqrt{\frac{\sqrt{2}}{2}}\}$

ISOLATED ROOTS:

$\{x_1 \in [1.41, 1.42], y_1 \in [0.84, 0.85]\}$ and
$\{x_2 \in [1.41, 1.42], y_2 \in [-0.85, -0.84]\}$

ENCODED ROOTS:

$\mathrm{box}_1 = [C, [1.41, 1.42], [0.84, 0.85]]$ and
$\mathrm{box}_2 = [C, [1.41, 1.42], [-0.84, -0.85]]$.

Root Isolation

Box

A *n-box* is of the form $B = I_1 \times \cdots \times I_n$ where each I_i is

- either $]a, b[$ for some $a, b \in \mathbb{Q}$ with $a < b$; then $|I_i| := b - a$,
- or $\{a\}$ for some $a \in \mathbb{Q}$; then $|I_i| = 0$.

The *width* of B, denoted by $|B|$, is the max of the $|I_i|$.

Isolation

Let $V \subset \mathbb{C}^n$ be a zero-dimensional variety. A list $B_1, \ldots, B_t$ of n-boxes is a *box-decomposition* of $V \cap \mathbb{R}^n$ if

- each point of $V \cap \mathbb{R}^n$ lies in exactly one B_i,
- $B_i \cap B_j = \emptyset$ whenever $i \neq j$,
- $|B_i|$ can be made arbitrary small for all i.

Zero-dimensional Regular Chains

Definition

$T \subset \mathbb{Q}[x_1 < \cdots < x_n] \setminus \mathbb{Q}$ is a *zero-dimensional regular chain* if

- $T = \{T_1(x_1), T_2(x_1, x_2), \ldots, T_n(x_1, \ldots, x_n)\}$,
- $\mathrm{lc}(T_i, x_i)$ is invertible modulo $\langle T_1, \ldots, T_{i-1} \rangle$ for $1 < i \leq n$,

Additional Properties

1. *Reduced*: $\deg(T_i, x_j) < \deg(T_j, x_j)$ for $1 \leq j < i \leq n$.
2. *Squarefree*: T_i and $\frac{\partial T_i}{\partial x_i}$ are relatively prime modulo $\langle T_1, \ldots, T_{i-1} \rangle$ for $1 \leq i \leq n$,
3. *Normalized*: $\mathrm{lc}(T_i, x_i) = 1$ for $1 \leq i \leq n$.

Comment

We require squarefreeness to ensure termination of our isolation process and speed-up sub-algorithms.

Three Fundamental Theorems

Descartes

Let $f = a_d x^d + \cdots + a_0 \in R[x]$ with $a_d \neq 0$. Let v be the number of sign changes in $a_d, \ldots, a_0$ and let r be the number of positive real roots of f. Then, there exists $m \geq 0$ such that we have $r = v - 2m$.

Sturm

Let $f \in \mathbb{R}[x]$ be a square-free polynomial and let $a, b \in \mathbb{R}$ s.t. $a < b$ and $f(a)f(b) \neq 0$. Let $f_0 = f, f_1, \ldots, f_s$ be a Sturm sequence for f on $[a, b]$. Then, the number of distinct roots of f in $[a, b]$ is given by $V(a) - V(b)$.

Yang, Hou and Zeng

Let $f = a_d x^d + \cdots + a_0 \in R[x]$ with $a_d \neq 0$. Let $D = D_1, D_2, \cdots, D_d$ be the discriminant sequence of f and L its revised sign list. Let ν be the number of sign changes in L and ℓ that of non-zero entries in L. Then, the number of distinct real roots of f equals $\ell - 2\nu$.

Real Root Isolation Algorithms (1/2)

For polynomials in $\mathbb{Q}[x]$

- The (independent) works of Vincent and Uspensky turn Descartes rule of signs into an algorithm for RRI.
- Rouillier and Zimmermann (2003) have designed a memory-efficient of it.
- Akritas et al. (2006, 2007) have further develop Vincent's work.

For univariate with real algebraic number coefficients

- Rioboo (1992) using Sturm Sequences and isolation intervals.
- Collins, Krandick, Johnson (2002, 1887) using Descartes rule of signs and interval arithmetic.

Real Root Isolation Algorithms (2/2)

For zero-dimensional multivariate systems

Various techniques are employed to reduce to the univariate case

1. *RUR*: (Rouillier, AAECC 1999)
2. *Polyhedron Algebra*: (Mourrain & Pavone, Tech. Rep. 2005)
3. *Triangular Sets*:
   - (Cheng, Gao & Yap, ISSAC 2007)
   - (Lu, He, Luo & Pan, SNC 2005)
   - (Xia & Zhang, Comput Math Appl 2006)

   These methods:
   - rely on sleeve of polynomials on an interval,
   - use big floats or interval arithmetic,
   - do **not** use algebraic operations (invertibility test, GCD computation) modulo a regular chain.

Our approach

Main idea

- Adapt Vincent-Collins-Akritas to $(\mathbb{Q}[x_1, \ldots, x_i]/\langle T \rangle)[x_{i+1}]$ where T is a 0-dim. squarefree regular chain.
- Deduce a RealRootIsolate command for 0-dim. regular chains.

Challenge

- $L_i := \mathbb{Q}[x_1, \ldots, x_i]/\langle T \rangle$ may not be a field and
- we need to evaluate signs of elements in L_i

Solution

- Combine interval arithmetic and invertibility test modulo $\langle T \rangle$.
- invertibility test shoots troubles in sign determination.
- (Rioboo 1992) uses a similar technique but in for univariate polynomials and with Sturm sequences.

Plan

VCA algorithm

Algorithm 1 VCA($p,]a, b[$)

Input: $p \in \mathbb{Q}[x]$ squarefree and $a < b$ rational.
Output: an interval decomposition of $V(p) \cap]a, b[$.

$nsv \leftarrow$ RootNumberBound($p,]a, b[$)
if $nsv = 0$ **then return** $\emptyset$
else if $nsv = 1$ **then return** $]a, b[$
else
    $m \leftarrow (a + b)/2 \quad res \leftarrow \emptyset$
    if $p(m) = 0$ **then** $res \leftarrow \{\{m\}\}$
    {Next line ensures the roots are sorted increasingly}
    return VCA($p,]a, m[$) $\cup$ $res \cup$ VCA($p,]m, b[$)

The RootNumberBound Algorithm

Algorithm 2 RootNumberBound($p,]a, b[$)

Input: $p \in \mathbb{Q}[x]$ and $a < b$ rational

Output: a bound on the number of roots of p in the interval $]a, b[$

1: $\bar{p} \leftarrow (x+1)^d \, p\left(\frac{a\,x+b}{x+1}\right)$ where d is the degree of p, and denote $\bar{p} = \sum_{i=0}^{d} a_i x^i$

2: $a'_e, \ldots, a'_0 \leftarrow$ the sequence obtained from $a_d, \ldots, a_0$ by removing zero coefficients

3: **return** the number of sign variations in the sequence $a'_e, \ldots, a'_0$

Plan

Invertibility Test

Algorithm 3 CheckZeroDivisor(p, T)

Input: $T \subset \mathbb{Q}[x_1, \ldots, x_n]$ a 0-dim regular chain and $p \in \mathbb{Q}[x_1, \ldots, x_n]$

Output: If p is invertible modulo T, then the algorithm terminates normally. Otherwise, an exception is thrown exhibiting $t \geq 2$ regular chains $T_1, \ldots, T_t$ such that $\langle T \rangle = \cap \langle T_i \rangle$ and $\sum_{i=1}^{t} \deg(T_i) = \deg(T)$.

1: $T_1, \ldots, T_t \leftarrow$ Regularize(p, T)

2: **if** p belongs to at least one $\langle T_i \rangle$ **then throw exception**($T_1, \ldots, T_t$)

DC Condition and Task

DC Condition

Let $B = I_1 \times \cdots \times I_t$ be an s-box and $T = \{p_1, \ldots, p_s\} \subset \mathbb{Q}[x_1, \ldots, x_s]$ be a 0-dim reg. chain. (B, T) satisfies the *Dichotomy Condition* (**DC**) if

- one and only one real root of T lies in B
- if $I_1 =]a, b[$ then $p_1(x_1 = a)p_1(x_1 = b) < 0$ holds
- if $I_k =]a, b[$, then EvalBox$(p_k(x_k = a), B)$, EvalBox$(p_k(x_k = b), B)$ do not meet 0 and have opposite signs, for all $2 \leq k \leq s$.

Task

Let T and B be as before such that (B, T) satisfies **DC**. Let $p \in \mathbb{Q}[x_1, \ldots, x_{s+1}]$ such that $T \cup p$ is a regular chain. Let $a < b$ be in $\mathbb{Q}$. Then $\mathcal{M} = \textsc{Task}(p,]a, b[, B, T)$ is called a task.
The solution of $\mathcal{M}$ denoted by $V_t(\mathcal{M})$ is defined as $V(T \cup \{p\}) \cap (B \times]a, b[)$ (i.e. the real solutions of $T \cup \{p\}$ which prolong the real root in B and whose x_{s+1}-component lies in $]a, b[$).

VCA Algorithm Modulo a Regular Chain

Algorithm 4 SolveTask($\mathcal{M}$)

Input: a task $\mathcal{M} = \text{TASK}(p,]a, b[, B, T)$ where T is a 0-dim squarefree regular chain of $\mathbb{Q}[x_1, \ldots, x_s]$.

$nsv, B' \leftarrow$ RootNumberBound($\mathcal{M}$)
if $nsv = 0$ **then return** $\emptyset$
else if $nsv = 1$ **then**
    $B'' \leftarrow B' \times]a, b[$
    refine B'' until $(B'', T \cup \{p\})$ satisfies **DC**
    return $\{B''\}$
else
    $m \leftarrow (a + b)/2 \quad res \leftarrow \emptyset \quad p' \leftarrow p(x_{s+1} = m)$
    if $p' \in \langle T \rangle$ **then** $res \leftarrow \{B' \times \{m\}\}$ **else** CheckZeroDivisor(p', T)
    return $res \cup \{\text{TASK}(p,]a, m[, B', T), \text{TASK}(p,]m, b[, B', T)\}$

The RootNumberBound Algorithm

Algorithm 5 RootNumberBound($\mathcal{M}$)

Input: a task $\mathcal{M} = \text{TASK}(p,]a, b[, B, T)$ where $T \subset \mathbb{Q}[x_1, \ldots, x_s]$.

Output: (nsv, B') such that $B' \subset B$, (B', T) satisfies **DC**, and nsv is a bound on the cardinal of $V_t(\mathcal{M})$. The bound is exact if $nsv \in \{0, 1\}$.

1: $\bar{p} \leftarrow (x_{s+1} + 1)^d \, p\left(x_{s+1} = \frac{a\, x_{s+1} + b}{x_{s+1} + 1}\right)$ with $d = \mathrm{mdeg}(p)$

2: denote $\bar{p} = \sum_{i=0}^{d} a_i x_{s+1}^i$

3: $a'_e, \ldots, a'_0 \leftarrow$ the sequence obtained from $a_d, \ldots, a_0$ by removing the a_i belonging to $\langle T \rangle$

4: **for all** a'_i **do** CheckZeroDivisor(a'_i, T)

5: $B' \leftarrow B$

6: **while** there is an a'_i such that $0 \in$ EvalBox(a'_i, B') **do** $B' =$ RefineBox(B', T)

7: **return** the number of sign variations of the sequence

$$\text{EvalBox}(a'_e, B'), \text{EvalBox}(a'_{e-1}, B'), \ldots, \text{EvalBox}(a'_0, B')$$

Plan

Implementation Issues

Tricks currently used

- Fast Taylor Shift (von zur Gathen & Gerhard, ISSAC 2007)
- Horner's rule for evaluating a polynomial on a box

Work in progress

- fast arithmetic techniques for CheckZeroDivisor(p, T) and testing $p \in \langle T \rangle$.
- Subproduct tree techniques for multiple calls to CheckZeroDivisor
- Greedy algorithms for optimizing Horner's rule
- Using floating-point number arithmetic (MPFR library) for interval arithmetic.

Plan

Special examples

nql-n-d examples

- Suggested by Fabrice Rouillier
- $x_1^d - 2 = 0$, $x_i^d + x_i^{d/2} - x_{i-1} = 0$ for $2 \leq i \leq n$ for some even degree d.
- This is a zero-dimensional regular chain.
- The algorithm RealRootIsolate solves it easily since the degrees are distributed evenly among the equations.
- A similar example is simple-nql-n-d defined by $x_1^d - 2 = 0$, $x_i^d - x_{i-1} = 0$ for $2 \leq i \leq n$. The degree of the rational univariate representation is also roughly d^n. For the example simple-nql-20-30, d^n is around 10^{29}.

nld-d-n examples

- n equations of the form
  $x_1 + \cdots + x_{i-1} + x_i^d + x_{i+1} + \cdots + x_n - 1 = 0$ for $1 \leq i \leq n$.
- Triangularize tend to split it into many branches, even though the equiprojectable decomposition consists of a few components (generally 2 or 3).
- For System nld-9-3, which has degree 729, the command Triangularize produces 16 components where the largest coefficient has size 20 digits.
- Whereas there are 3 equiprojectable components where most coefficients have more than 1,000 digits.

Comparison with RootFinding[Isolate]

	Sys	v/e/s	Rf-1	Rf-2	Tr	Is/10
1	4-body-homog	3/3/7	0.31	0.32	1.6	11
2	5-body-homog	3/3/11	0.31	0.36	3.1	32
3	Caprasse	4/4/18	0.13	0.12	1.2	2.9
4	circles	2/2/22	0.89	0.9	0.55	26
5	cyclic-5	5/5/10	0.4	0.4	2.4	4.6
6	neural-network	4/4/22	1	1	0.81	18
7	nld-9-3	3/3/7	1785	1777	39	43
8	nld-10-3	3/3/8	$>$2000	$>$2000	26	148
9	nql-10-4	10/10/2	$>$2000	$>$2000	0.33	3.2
10	nql-15-2	15/15/2	$>$2000	$>$2000	0.36	5.8
11	p3p-special	5/5/24	0.41	0.46	0.23	23
12	r-5	5/5/1	1.6	1.6	0.43	$<$0.1
13	r-6	6/6/1	$>$2000	$>$2000	0.96	$<$0.1
14	Rose	3/3/18	0.63	0.67	0.72	39
15	simple-nql-20-30	20/20/2	$>$2000	$>$2000	0.57	28

Different Strategies

	Strategy 1		Strategy 2		Strategy 3			
Sys	Tr	Is/10	Tr/No	Is/10	Tr	Is/∞	∞/5	5/10
1	1.6	11	6.2	11	1.5	3.4	4	4.1
2	3.1	32	38	43	3.2	9.4	11	12
3	1.2	2.9	1.5	2	1.2	0.52	1.6	1.4
4	0.55	26	1.1	26	0.59	16	4.6	4.5
5	2.4	4.6	3.6	1.4	2.5	0.67	3.9	1.8
6	0.81	18	1.2	15	0.87	4.5	7.7	7
7	39	43	121	70	40	45	0.34	0.29
8	26	148	370	308	25	148	8.1	8.1
9	0.33	3.2	0.61	3.3	0.34	0.92	0.62	0.83
10	0.36	5.8	0.65	5.7	0.33	3.1	1.3	1.9
11	0.23	23	0.69	31	0.24	6.4	8.2	9
12	0.43	$<$0.1	0.49	$<$0.1	0.37	$<$0.1	$<$0.1	$<$0.1
13	0.96	$<$0.1	1.2	$<$0.1	0.98	$<$0.1	$<$0.1	$<$0.1
14	0.72	39	1.1	59	0.71	5	22	20
15	0.57	28	0.88	28	0.63	65	2.8	0.33

Plan

Conclusion

- We have adapted the Vincent-Collins-Akritas Algorithm to work modulo a zero-dimensional regular chain
- This provides a way for isolating the real roots of zero-dimensional systems.
- In our context, it is easy to prescribe the values of some variables and take it into account during the isolation process.
- We have realized a preliminary, non-optimized implementation in Maple interpreted code.
- For certain degree configurations (non Shape Lemma systems) it can outperform optimized implementation written in C.
- There is a large room for optimizing our VCA algorithm and its implementation.