Topics in Cylindrical Algebraic Decomposition

PhD Confirmation Report

Corin Lee

Department of Mathematical Sciences
University of Bath

Overview

Cylindrical Algebraic Decompositions

A *cylindrical algebraic decomposition* (abbreviated CAD) is a method in real semi-algebraic geometry to decompose $\mathbb{R}^n$ into a finite number of connected semi-algebraic subsets known as *cells*, each homeomorphic to $\mathbb{R}^k$.

CAD has applications in algebraic simplification technology [Bradford Davenport, 2002] and robot motion planning, one example of which is the Piano Movers Problem [Davenport, 1986, Wilson et al, 2013, Schwartz Sharir, 1983]. However, trying to compute even a simple case for this problem can be computationally expensive and impractical.

It is known that the worst-case complexity is doubly-exponential in the number of variables [Davenport Heintz 1987]. Nevertheless, the algorithms are practicable in important cases, and small efficiency gains become important.

There are also competing definitions of CAD and different types of CAD.

Definitions

Let $\mathbb{R}[\mathbf{x}]$ be the polynomial ring over $\mathbb{R}$ with ordered variables $\mathbf{x} = x_1 < \cdots < x_n$.

Definition (region, cylinder, section, sector)

For n-dimensional real space $\mathbb{R}^n$,

- A nonempty connected subset R of $\mathbb{R}^n$ is known as a *region*.
- For a region R, the *cylinder* over R, written $\mathcal{Z}(R)$, is the set $R \times \mathbb{R} = \{(\alpha, x) \mid \alpha \in R, x \in \mathbb{R}\}$.
- For f a continuous, real valued function of R, an f*-section* of $\mathcal{Z}(R)$ is the set $\{\alpha, f(\alpha) \mid \alpha \in R\}$
- For f_1, f_2 continuous, real valued functions of R, an (f_1, f_2)*-sector* of $\mathcal{Z}(R)$ is the set $\{(\alpha, \beta) \mid \alpha \in R, f_1(\alpha) < \beta < f_2(\alpha)\}$. The functions $f_1 = -\infty$ and $f_2 = +\infty$ are allowed.

Definitions

Definition (Decomposition, stack)

For any subset X of $\mathbb{R}^n$, a *decomposition* of X is a finite collection of disjoint regions whose union is X.

Continuous, real-valued functions $f_1 < f_2 < \cdots < f_k,\ k \geq 0$, defined on R, naturally determine a decomposition of $\mathcal{Z}(R)$ consisting of the following regions:

1. the (f_i, f_{i+1})-sectors of $\mathcal{Z}(R)$ for $0 \leq i \leq k$, where $f_0 = -\infty$ and $f_{k+1} = +\infty$, and
2. the f_i-sections of $\mathcal{Z}(R)$ for $1 \leq i \leq k$.

We call such a decomposition a *stack* over R (determined by $f_1, \ldots, f_k$).

Definitions

Definition (Cylindrical decomposition of $\mathbb{R}^n$)

A decomposition $\mathcal{D}$ of $\mathbb{R}^n$ is *cylindrical* if either

1. $n = 1$ and $\mathcal{D}$ is a stack over $\mathbb{R}^0$, or
2. $n > 1$, and there is a unique cylindrical decomposition $\mathcal{D}'$ of $\mathbb{R}^{n-1}$ such that for each region R of $\mathcal{D}'$, some subset of $\mathcal{D}$ is a stack over R.

As $\mathcal{D}'$ is unique, any cylindrical decomposition $\mathcal{D}$ of $\mathbb{R}^n$ will have unique *induced* cylindrical decompositions of $\mathbb{R}^j$ for $j = n-1, n-2, \ldots, 1$. Conversely, given a CAD $\mathcal{D}'$ of $\mathbb{R}^j, j < n$, a CAD $\mathcal{D}$ of $\mathbb{R}^n$ is an *extension* of $\mathcal{D}'$ if $\mathcal{D}$ induces $\mathcal{D}'$.

Alternatively, a decomposition is *cylindrical* if for all $1 \leq j < n$, the projections on the first j variables of any two cells are either equal or disjoint.

Definitions

Definition (Semi-algebraic set)

A subset of $\mathbb{R}^n$ is *semi-algebraic* if it can be constructed by finitely many applications of the union, intersection, and complementation operations on sets of the form

$$\{x \in \mathbb{R}^n \mid f(x) \geq 0\},$$

where $f \in \mathbb{R}[\mathbf{x}]$.

Definition (Cylindrical Algebraic Decomposition)

A decomposition is *algebraic* if each of its regions is a semi-algebraic set. *A cylindrical algebraic decomposition*, or *CAD*, of $\mathbb{R}^n$ is a decomposition which is both cylindrical and semi-algebraic.

The components of a CAD are called *cells*, and for $0 \leq j \leq n$, a *j-cell* in $\mathbb{R}^n$ is a subset of $\mathbb{R}^n$ which is homeomorphic to $\mathbb{R}^j$.

Definitions

We often want the decomposition to respect some collection of polynomials:

Definition ($\mathcal{F}$-invariant)

Let $\mathcal{F} = \{f_i \in \mathbf{k}[x_1, \ldots, x_n], 1 \leq i \leq n\}$ be a set of polynomials in $\mathbf{k}[x_1, \ldots, x_n]$ for $\mathbf{k} = \mathbb{C}$ or $\mathbb{R}$ and $X \subseteq \mathbf{k}^n$. We say X is *$\mathcal{F}$-invariant* (and $\mathcal{F}$ is *invariant* on X) if each $f_i(x)$ has constant sign for every $x \in X$, that is,

$$\forall x \in X : f_i(x) \ \diamond \ 0,$$

where for $\mathbf{k} = \mathbb{R}, \diamond \in \{>, =, <\}$, and for $\mathbf{k} = \mathbb{C}, \diamond \in \{=, \neq\}$. In other words, f_i is either identically zero or never zero; a polynomial $f_i(x)$ with constant sign in this sense is called *sign-invariant*.

Definitions

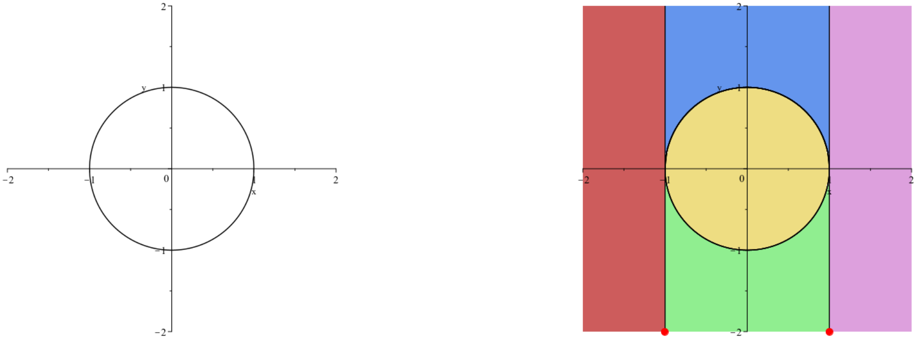

Figure: The graph of $x^2 + y^2 - 1$ and the $\{x^2 + y^2 - 1\}$-invariant CAD of $\mathbb{R}^2$

Approaches and Implementations

There have been several implementations of Collins' original projection and lifting CAD algorithm (PL-CAD) in various software, along with variants and improvements, which usually aim to produce better performance under certain conditions.

Another algorithm, based on triangular sets and regular chains (hereby referred to as RC-CAD), has been implemented in Maple as part of the `RegularChains` package [RegularChains]. Despite this, little appears to have been done in terms of complexity analysis of this algorithm.

PL-CAD

The PL-CAD algorithm can be split into three phases:

- **Projection:** Repeatedly apply a projection operator *Proj*:

$$\mathcal{F} = \mathcal{F}_n(x_1, \ldots, x_n) \xrightarrow{Proj} \mathcal{F}_{n-1}(x_1, \ldots, x_{n-1}) \xrightarrow{Proj} \ldots \xrightarrow{Proj} \mathcal{F}_1(x_1)$$

  The zero sets of the polynomials produced by each step are the projections of "significant points" of the previous set of polynomials.
- **Base phase:** Real root isolation on $\mathcal{F}_1(x_1)$, roots and open intervals between form an $\mathcal{F}_1$-invariant CAD of $\mathbb{R}^1$.
- **Lifting phase:** For each cell C of the $\mathcal{F}_{k-1}$-invariant CAD in $\mathbb{R}^{k-1}$, construct a sample point s and isolate the real roots of the polynomials of $\mathcal{F}_k$ at s. The sectors and sections of these polynomials form a stack, and these stacks make the cells of the $\mathcal{F}_k$-invariant CAD of $\mathbb{R}^k$ above C.

It is enough to determine the signs of $\mathcal{F}$ in these sample points as each cell is $\mathcal{F}$-invariant by construction.

RC-CAD

Incremental algorithm for computing CADs using triangular systems and regular chains (RC-CAD), by constructing a cylindrical decomposition of the complex space (CCD), and refining it into a real CAD. This had the advantage that other PL-CAD methods did not have, which is that it did not have problems with curtains (regions where a polynomial nullifies over a set). This was improved in [Chen Moreno Maza, 2012] by computing CCDs in an incremental way to avoid redundant computations.

It has been implemented into *Maple* via the `RegularChains` package, but little analysis has been done.

We are interested in trying to understand how the algorithm works in sufficient detail to give estimates and boundaries for its complexity, and in learning when and where the algorithm is more, or less, efficient.

The Incremental RC-CAD Algorithm

The CCD construction in RC-CAD can be seen as an enhanced projection phase of PL-CAD [Chen Moreno Maza, 2012], with the benefit that its "case discussion" scheme avoids unnecessary computations that the projection operator performs on unrelated branches, and avoids curtains.

The incremental algorithm of [Chen Moreno Maza, 2012] involves refining the branches of a tree via GCD computation. The CCD algorithm produces a decomposition into triangular sets, say $\mathcal{D}$ such that the zero sets of the output regular chains are disjoint. Such a tree is encoded by a tree data structure, and the decomposition computed is both disjoint and cylindrically arranged.

The complexity of this algorithm can also not be better than doubly exponential in the number of variables [Bradford Davenport, 2002], but the benchmarking of [Chen Moreno Maza, 2012] shows PL-CAD outperforming QEPCAD and `Mathematica` for several well-known examples.

Despite this, no theoretical complexity analysis seems to have been done, despite both these results and the availability of the algorithm in the `RegularChains` Maple package.

Definitions

Let $p \in \mathbb{R}[\mathbf{x}]$ be a non-constant polynomial and $x \in \mathbf{x}$ be a variable.

Definition

1. We denote by $\deg(p, x)$ and $\mathrm{lc}(p, x)$ the *degree* and the *leading coefficient* of p w.r.t. x.
2. The greatest variable appearing in p is called the *main variable*, denoted by $\mathrm{mvar}(p)$.
3. The *separant* $\mathrm{sep}(p)$ of p is $\partial p / \partial\, \mathrm{mvar}(p)$.
4. The leading coefficient, the degree, and the reductum (p minus its leading term) of p regarded as a univariate polynomial in $\mathrm{mvar}(p)$ are called the *initial*, the *main degree*, the *tail* of p; they are denoted by $\mathrm{init}(p), \mathrm{mdeg}(p), \mathrm{tail}(p)$ respectively.
5. The integer k such that $x_k = \mathrm{mvar}(p)$ is called the *level* of the polynomial p.

Triangular Sets and Regular Chains

Definition (Triangular set)

Let $\mathcal{T} \subset \mathbb{R}[\mathbf{x}]$ be a *triangular set*, that is, a set of nonconstant polynomials with pairwise distinct main variables, that is, for all $t, t' \in \mathcal{T}, \mathrm{mvar}(t) \neq \mathrm{mvar}(t')$.

- We denote by $\mathrm{mvar}(\mathcal{T})$ the set of the main variables of the polynomials in $\mathcal{T}$.
- A variable in $\mathbf{x}$ is *algebraic* w.r.t. $\mathcal{T}$ if it belongs to $\mathrm{mvar}(\mathcal{T})$, else it is *free* w.r.t. $\mathcal{T}$.
- For $v \in \mathbf{x}$, we denote by $\mathcal{T}_{<v}$ the set of the polynomials $t \in \mathcal{T}$ such that $\mathrm{mvar}(t) < v$ holds.

Definition (Iterated resultant)

Let $h \in \mathbb{R}[\mathbf{x}]$. The *iterated resultant* of h w.r.t. $\mathcal{T}$, denoted by $\mathrm{ires}(h, \mathcal{T})$, is defined as follows:

1. if $h \in \mathbb{R}$ or all variables in h are free w.r.t. $\mathcal{T}$, then $\mathrm{ires}(h, \mathcal{T}) = h$;
2. Otherwise, if v is the largest variable of h which is algebraic w.r.t. $\mathcal{T}$, then $\mathrm{ires}(h, \mathcal{T}) = \mathrm{ires}(r, \mathcal{T}_{<v})$ where r is the resultant w.r.t. v of h and the polynomial in $\mathcal{T}$ whose main variable is v.

Triangular Sets and Regular Chains

Definition (Regular chain, regular system)

Let $h_{\mathcal{T}}$ be the product of the initials of the polynomials in $\mathcal{T}$. A triangular set T is called a *regular chain* if either $\mathcal{T} = \emptyset$ or $\text{ires}(h_{\mathcal{T}}, \mathcal{T}) \neq 0$. The pair $[\mathcal{T}, h]$ is called a *regular system* if $\mathcal{T}$ is a regular chain, and $\text{ires}(h, \mathcal{T}) \neq 0$.

Definition (Squarefree)

Denote by $\text{sep}(\mathcal{T})$ the product of all $\text{sep}(p)$, for $p \in \mathcal{T}$. Then $\mathcal{T}$ is said to be *squarefree if* $\text{ires}(\text{sep}(\mathcal{T}), \mathcal{T}) \neq 0$. A regular system $rs = [T, h]$ is said to be *squarefree* if $\mathcal{T}$ is squarefree.

Complex Cylindrical Trees

Definition (Separation)

Let C be a subset of $\mathbb{C}^{n-1}$ and $P \subset \mathbb{R}[x_1, \ldots, x_{n-1}, x_n]$ be a finite set of level n polynomials. We say that P *separates above* C if for each $\alpha \in C$:

- for each $p \in P$, the polynomial $\mathrm{init}(p)$ does not vanish at α,
- the polynomials $p(\alpha, x_n) \in \mathbb{C}[x_n]$, for all $p \in P$, are squarefree and coprime.

Note that this definition allows C to be a semi-algebraic set. Effectively, this says that the main variable of p is the same whether you look everywhere or only above C, so C is in "general position" relative to P.

Complex Cylindrical Trees

Definition (Cylindrical decomposition of $\mathbb{C}^n$ and associated tree)

By induction on n, we define the notion of a *cylindrical decomposition (CD) of* $\mathbb{C}^n$ together with that of the *tree associated with a cylindrical decomposition of* $\mathbb{C}^n$.

n = 1: a CD of $\mathbb{C}$ is $\mathcal{D} = \{D_1, \ldots, D_{r+1}\}$, where either $r = 0$ and $D_1 = \mathbb{C}$, or $r > 0$ and there exist r nonconstant coprime squarefree polynomials $p_1, \ldots, p_r$ of $\mathbb{R}[x_1]$ such that for $1 \leq i \leq r$ we have

$$D_i = \{x_1 \in \mathbb{C} \mid p_i(x_1) = 0\}, \qquad \text{and} \qquad D_{r+1} = \{x_1 \in \mathbb{C} \mid p_1(x_1) \cdots p_r(x_1) \neq 0\}.$$

These form a partition of $\mathbb{C}$.

The tree associated with $\mathcal{D}$ is a rooted tree whose nodes, other than the root, are $D_1, \ldots, D_r, D_{r+1}$ which all are leaves and children of the root.

Complex Cylindrical Trees

Definition (Cylindrical decomposition of $\mathbb{C}^n$ and associated tree)

n > 1: Let $\mathcal{D}' = \{D_1, \ldots, D_s\}$ be any cylindrical decomposition of $\mathbb{C}^{n-1}$. For each D_i , let r_i be a non-negative integer and let $\{p_{i,1}, \ldots, p_{i,r_i}\}$ be a set of polynomials which separates above D_i. If $r_i = 0$, set $D_{i,1} = D_i \times \mathbb{C}$. If $r_i > 0$, set

$$D_{i,j} = \{(\alpha, x_n) \in \mathbb{C}^n \mid \alpha \in D_i \text{ and } p_{i,j}(\alpha, x_n) = 0\},$$

for $1 \leq j \leq r_i$ and set

$$D_{i,r_i+1} = \{(\alpha, x_n) \in \mathbb{C}^n \mid \alpha \in D_i \text{ and } \prod_{j=1}^{r_i} p_{i,j}(\alpha, x_n) \neq 0\}.$$

The collection $\mathcal{D} = \{D_{i,j} \mid 1 \leq i \leq s, \quad 1 \leq j \leq r_i + 1\}$ is called a *cylindrical decomposition* of $\mathbb{C}^n$. The sets $D_{i,j}$ are called the *cells* of $\mathcal{D}$.

Complex Cylindrical Trees

Definition (Cylindrical decomposition of $\mathbb{C}^n$ and associated tree)

If T' is the tree associated with $\mathcal{D}'$ then the tree T associated with $\mathcal{D}$ is defined as follows.

For each $1 \leq i \leq s$, the set D_i is a leaf in T' which has all $D_{i,j}$ for children in T; thus the $D_{i,j}$ are the leaves of T.

Note that each node N of T is either associated with no constraints, or associated with a polynomial constraint, which itself is either an equation or an inequation. Note also that, if the level of the polynomial defining the constraint at N is ℓ, then ℓ is the length of a path from N to the root.

Moreover, the polynomial constraints along a path from the root to a leaf form a polynomial system called a *cylindrical system of* $\mathbb{R}[x_1, \ldots, x_{n-1}]$ *induced by* T. Let S be such a cylindrical system. We denote by $Z(S)$ the *zero set* of S. Therefore, each cell of $\mathcal{D}$ is the zero set of a cylindrical system induced by T.

Complex Cylindrical Trees

Definition

Let $\hat{T}$ be a sub-tree of T such that the root of $\hat{T}$ is that of T. Then, we call $\hat{T}$ a *cylindrical tree of* $\mathbb{R}[x_1, \ldots, x_{n-1}]$ *induced by* T. This cylindrical tree $\hat{T}$ is said to be *partial* if it admits a non-leaf node N such that the zero set of the constraint of N is not equal to the union of the zero sets of the constraints of the children of N. If $\hat{T}$ is not partial, then it is called *complete*.

Let $\mathcal{F} = \{f_1, \ldots, f_s\}$ be a finite set of polynomials of $\mathbb{R}[\mathbf{x}]$. A cylindrical decomposition $\mathcal{D}$ of $\mathbb{C}^n$ is called $\mathcal{F}$*-invariant* if for any given cell D of $\mathcal{D}$ and any given polynomial $f \in \mathcal{F}$, either f vanishes at all points of D or f vanishes at no points of D, and f is *sign invariant* if it is either identically zero or invertible. Note that this is consistent with the earlier definition.

Complex Cylindrical Trees

The definition of $Z(S)$ can be extended by replacing S with a subtree $\hat{T}$ as above:

Definition

$Z(\hat{T}) = \{w \in \mathbb{C}^n \mid f(w) = 0 \ \forall f \text{ labelling edges of } \hat{T}\}$

If $p \in \mathbb{R}[\mathbf{x}]$, we denote by V_p the variety of p, and we say

1. p is *invertible modulo* $\hat{T}$ if $V_p \cap Z(\hat{T}) = \emptyset$.
2. p is *zero modulo* $\hat{T}$ if $V_p \cap Z(\hat{T}) = Z(\hat{T})$.
3. We say p is *sign invariant above* $\hat{T}$ if it is either invertible or zero modulo $\hat{T}$.

Thus, if $q \in \mathbb{R}[\mathbf{x}]$, then $p = q \mod \hat{T}$ if $V_p \cap Z(\hat{T}) = V_q \cap Z(\hat{T})$, that is, you cannot distinguish them just by looking at $\hat{T}$.

In the special case where $\hat{T} = S$ is a cylindrical system, we make the following definition:

Definition (GCD mod S)

$g \in \mathbb{R}[\mathbf{x}]$ is a GCD of p and f mod S if $g(\alpha)$ is a GCD of $p(\alpha)$ and $f(\alpha) \in \mathbb{R}[z_n]$, for any $\alpha \in Z(S)$.

Complex Cylindrical Trees

If p_1 and p_2 are univariate polynomials over an algebraically closed field, then they are coprime if and only if they do not ever both vanish at the same place, i.e. the set of zeroes of $p_1(\alpha, x_n)$ and $p_2(\alpha, x_n)$ are disjoint.

In the base case $n = 1$, each D_i for $1 \leq i \leq r$ is the vanishing locus of p_i and D_{r+1} is everything else, and the p_i are coprime and squarefree. Thus, $p_i = \prod_{w_j \in D_i}(z - w_j)$, and these are squarefree by construction. The p_i being coprime means D_1 through D_r are disjoint. The main constraint is that the p_i have to be real polynomials.

For $n > 1$, inductively, we have a cylindrical decomposition of $\mathbb{C}^{n-1}$. For each of the D_i we choose some polynomials that separate above D_1. The line above α in D_i contains points where these polynomials vanish, because they have been chosen to separate above D_1, the polynomials are squarefree and coprime, and we repeat the process on the fibre.

We continue making the tree in the same way, adding leaves $D_{1,1}$ to D_{1,r_1+1}. Each of these is associated with either a constraint (a polynomial p_i or $p_{i,j}$ and so on) or with no constraints. Each cell $D_{i,1}, \ldots, D_{i,\ell}$ is the zero set of a cylindrical system induced by T.

Construction of a cylindrical tree

A brief overview is shown below.

Algorithm 1 `CylindricalDecompose`($\mathcal{F}$) Meta-algorithm

Input: $\mathcal{F}$ a set of non-constant polynomials in $\mathbb{R}[\mathbf{x}]$
Output: An $\mathcal{F}$-invariant cylindrical decomposition of $\mathbb{C}^n$

$\texttt{CylindricalDecompose}(\mathcal{F})$
- $\texttt{Intersect}_n(p, T)$
  - $\texttt{IntersectPath}_n(p, \Gamma, T)$
    - $\texttt{IntersectMain}_k(p, \Gamma_k, T_k)$
      - $\texttt{Squarefree}_k(p, \Gamma_{k-1}, T_{k-1})$
        - $\texttt{MakeLeadingCoefficientInvertible}_k(p, p, \Gamma_{k-1}, T_{k-1})$
          - $\texttt{IntersectPath}_{k-2}(\mathrm{lc}(\bar{p}, x_{k-1}), \Gamma_{k-1}, T_{k-1})$
          - $\ldots$
        - $\texttt{Gcd}_k(f, \mathrm{sep}(f), C_{k-1}, T_{k-1})$
          - $\texttt{Gcd}_k(f, \mathrm{sep}(f), S, d, 0, C_{k-1}, T_{k-1})$
            - $\texttt{IntersectPath}_{k-1}(s_i, C_{k-1}, T_{k-1})$
            - $\ldots$
      - $\texttt{Gcd}_k(sp, f, C_{k-1}, T_{k-1})$
        - $\texttt{Gcd}_k(sp, f, S, d, 0, C_{k-1}, T_{k-1})$
          - $\texttt{IntersectPath}_{k-1}(s_i, C_{k-1}, T_{k-1})$
      - $\texttt{Cofactor}(sp, L.Gcd[sp, f], f)$

Building a CAD from a Complex Cylindrical Tree

The final step is to compute a CAD of $\mathbb{R}^n$ from a cylindrical decomposition of $\mathbb{C}^n$.

Definition (Delineable)

Suppose $p \in \mathbb{R}[\mathbf{x}]$ and S is a connected semi-algebraic subset of $\mathbb{R}^{n-1}$. We say that p is *delineable* on S if the real zeros of p define continuous semi-algebraic functions $\theta_1, \ldots, \theta_s$ such that, for all $\alpha \in S$ we have $\theta_1(\alpha) < \cdots < \theta_s(\alpha)$. In other words, p is delineable on S if its real zeroes determine a stack over S.

Applying this inductively to our cylindrical tree yeilds an $\mathcal{F}$-invariant CAD of $\mathbb{R}^n$.

Complexity of RC-CAD

RC-CAD is a different algorithm to PL-CAD at a much earlier stage of development. It has been implemented in *Maple*, but not a very great deal has been done on trying to improve its performance. There are some experiments and timings conducted by Chen and Moreno Maza, which suggest it is at least competitive with PL-CAD sometimes. However, no theoretical complexity estimates have been done, and as a consequence we have no idea of when it performs well or badly.

For PL-CAD, on the other hand, there have been serious attempts to understand the complexity in detail and also to understand where the worst cases occur.

Current and Future Research

Using the meta-algorithm described above as the first step, I plan to carry out a detailed complexity analysis of RC-CAD, and once I have complexity estimates, refine them and see if I can determine cases where complexity is better or worse and get some kind of heuristic for when it is better to use RC-CAD or PL-CAD. To help with this I have also been looking at the *Maple* source.

I also intend to doing some experiments on some further test cases (such as the Piano Mover's Problem in [Davenport, 1986], where PL-CAD fails).

Given there are competing versions of CAD, various desirable properties that a CAD might have such as being well-based [Davenport Locatelli Sankaran 2019], we would like to know what kind of CAD RC-CAD produces, and whether we can we "coerce" it to produce the types of CAD we want.

Beyond that, there is another algorithm using Comprehensive Gröbner Systems (CGS-CAD) that is not implemented as far as we know, so there are no timings and very little knowledge of complexity. This is something we could also investigate in the future.

References

Collins, George (1975)
Quantifier elimination for real closed fields by cylindrical algebraic decomposition
Lecture Notes in Computer Science.

Arnon, Dennis and Collins, George and McCallum, Scott (1984)
Cylindrical Algebraic Decomposition I: The Basic Algorithm
SIAM J. Comput. 13(11), 865 – 877.

Jirstrand, Mats (1995)
Cylindrical Algebraic Decomposition - an Introduction

Chen, Changbo and Moreno Maza, Marc (2009)
Computing cylindrical algebraic decomposition via triangular decomposition
Proceedings of the 2009 international symposium on Symbolic and algebraic computation, 95 – 102.

Chen, Changbo and Moreno Maza, Marc (2012)
An Incremental Algorithm for Computing Cylindrical Algebraic Decompositions

Davenport, James (1986)
A "piano movers" problem

References

- Bradford, Russell and Davenport, James H. (2002)
  Towards Better Simplification of Elementary Functions
- Schwartz, Jacob T. and Sharir, Micha (1983)
  On the "piano movers" problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers
- Wilson, David and Davenport, James H. and England, Matthew and Bradford, Russell (2013)
  A "Piano Movers" Problem Reformulated
- Davenport, J. H. and Locatelli, A. F. and Sankaran, G. K. (2019)
  Regular cylindrical algebraic decomposition
- Davenport, James H. and Heintz, Joos (1987)
  Real quantifier elimination is doubly exponential
- Chen, Changbo and Moreno Maza, Marc
  The RegularChains Library `https://www.regularchains.org/`