Improving the Use of Equational Constraints in Cylindrical Algebraic Decomposition

Matthew England (Coventry University)

Joint work with:
Russell Bradford and James Davenport (University of Bath)

40th International Symposium on Symbolic and Algebraic Computation
University of Bath, Bath, UK.
6–9 July 2015

Supported by EPSRC Grant EP/J003247/1.

Outline

1. Introduction
   - Cylindrical Algebraic Decomposition
   - Equational Constraints
2. Improving the Use of ECs in CAD
   - Reductions in the Lifting Phase
   - Algorithm
3. Evaluating the New Algorithm
   - Worked Example
   - Complexity Analysis

Outline

1. Introduction
   - Cylindrical Algebraic Decomposition
   - Equational Constraints
2. Improving the Use of ECs in CAD
   - Reductions in the Lifting Phase
   - Algorithm
3. Evaluating the New Algorithm
   - Worked Example
   - Complexity Analysis

What is a CAD?

A Cylindrical Algebraic Decomposition (CAD) is:

- a decomposition meaning a partition of $\mathbb{R}^n$ into connected subsets called cells;
- (semi)-algebraic meaning that each cell can be defined by a sequence of polynomial equations and inequations.
- cylindrical meaning the cells are arranged in a useful manner - their projections are either equal or disjoint.

Traditionally a CAD is produced from a set of polynomials such that each polynomial has constant sign (positive, zero or negative) in each cell. Such a CAD is said to be sign-invariant.

Sign-invariance means we need only test one sample point per cell to determine behaviour of the polynomials

What is a CAD?

A Cylindrical Algebraic Decomposition (CAD) is:

- a decomposition meaning a partition of $\mathbb{R}^n$ into connected subsets called cells;
- (semi)-algebraic meaning that each cell can be defined by a sequence of polynomial equations and inequations.
- cylindrical meaning the cells are arranged in a useful manner - their projections are either equal or disjoint.

Traditionally a CAD is produced from a set of polynomials such that each polynomial has constant sign (positive, zero or negative) in each cell. Such a CAD is said to be sign-invariant.

Sign-invariance means we need only test one sample point per cell to determine behaviour of the polynomials

Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells.

$$
\begin{aligned}
x < -1 \{ &\quad [x < -1, y = y], \\
x = -1 \{ &\quad [x = -1, y < 0], [x = -1, y = 0], [x = -1, y > 0], \\
\{ &\quad [-1 < x < 1, y^2 + x^2 - 1 > 0, y > 0], \\
\{ &\quad [-1 < x < 1, y^2 + x^2 - 1 = 0, y > 0], \\
-1 < x < 1 \{ &\quad [-1 < x < 1, y^2 + x^2 - 1 < 0], \\
\{ &\quad [-1 < x < 1, y^2 + x^2 - 1 = 0, y < 0], \\
\{ &\quad [-1 < x < 1, y^2 + x^2 - 1 < 0, y < 0], \\
x = 1 \{ &\quad [x = 1, y < 0], [x = 1, y = 0], [x = 1, y > 0], \\
x > 1 \{ &\quad [x > 1, y = y]
\end{aligned}
$$

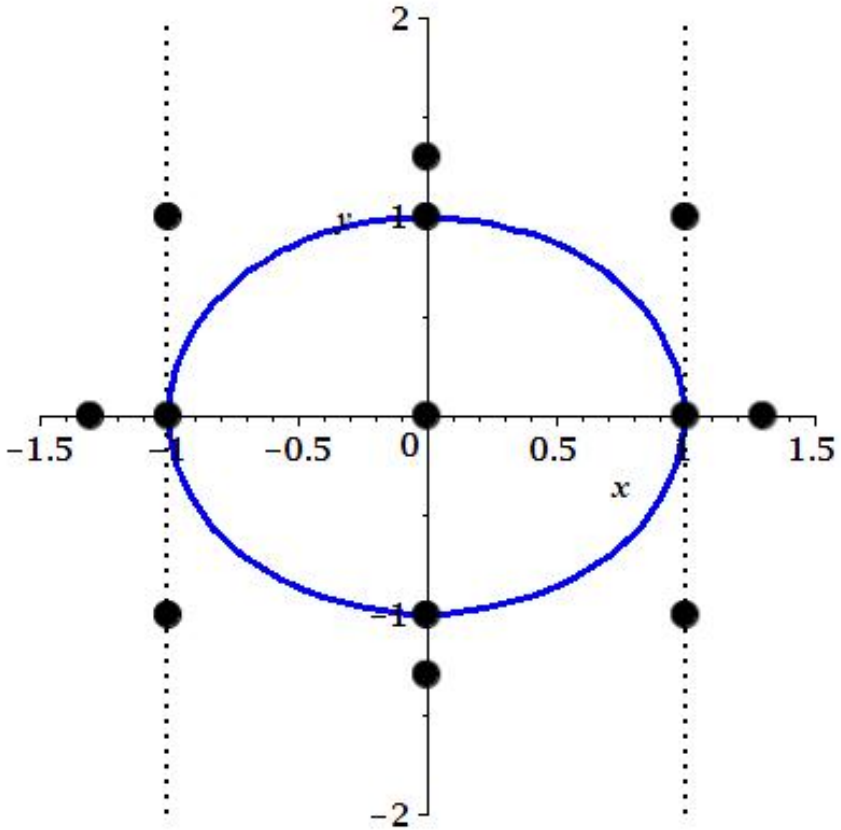
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.

$$
\begin{array}{rll}
x < -1 \{ & [x < -1, y = y], \\
x = -1 \{ & [x = -1, y < 0], [x = -1, y = 0], [x = -1, y > 0], \\
\{ & [-1 < x < 1, y^2 + x^2 - 1 > 0, y > 0], \\
\{ & [-1 < x < 1, y^2 + x^2 - 1 = 0, y > 0], \\
-1 < x < 1 \{ & [-1 < x < 1, y^2 + x^2 - 1 < 0], \\
\{ & [-1 < x < 1, y^2 + x^2 - 1 = 0, y < 0], \\
\{ & [-1 < x < 1, y^2 + x^2 - 1 < 0, y < 0], \\
x = 1 \{ & [x = 1, y < 0], [x = 1, y = 0], [x = 1, y > 0], \\
x > 1 \{ & [x > 1, y = y]
\end{array}
$$

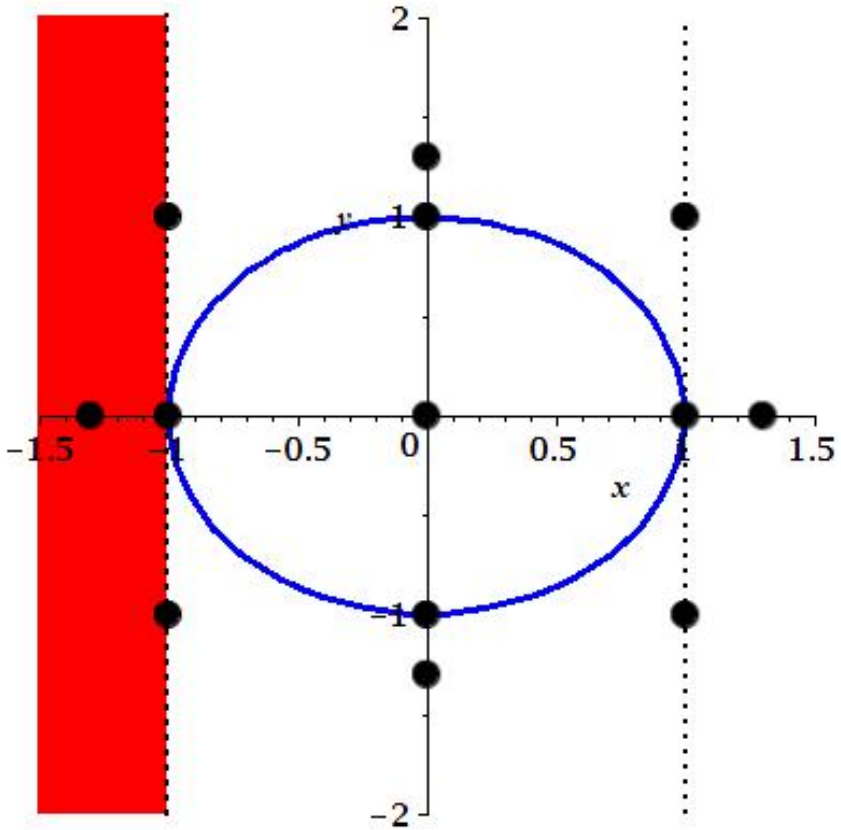
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



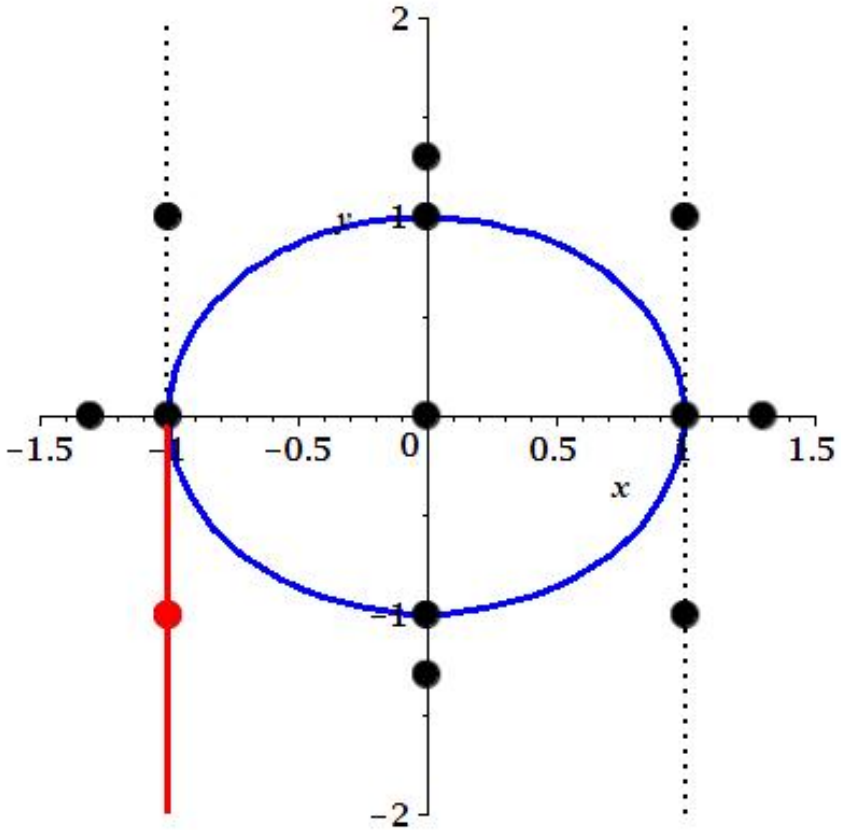
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



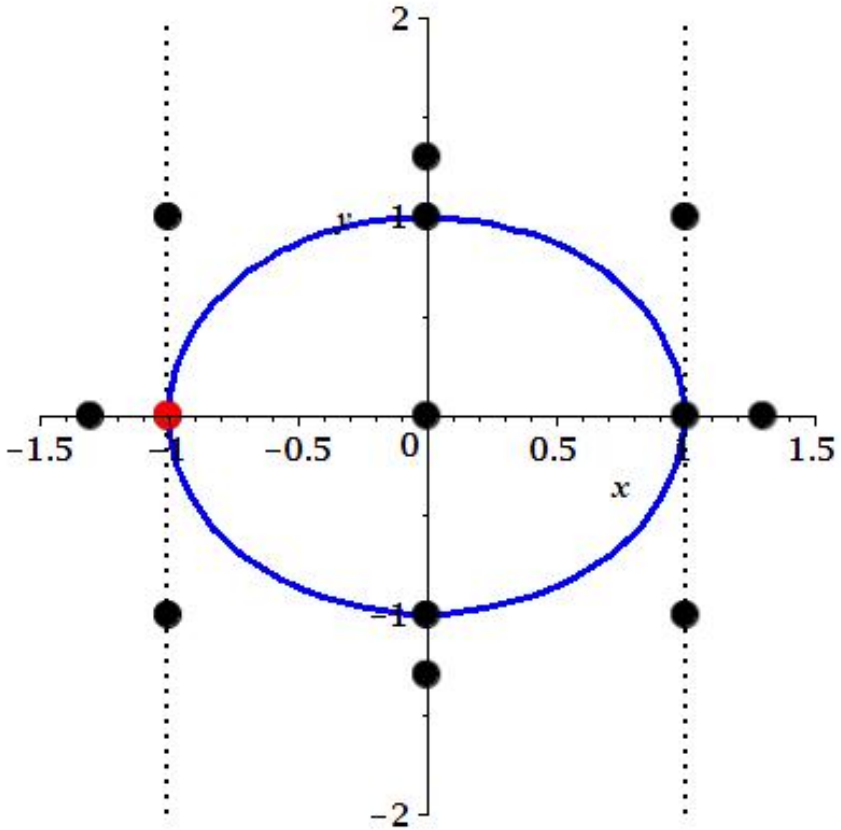
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



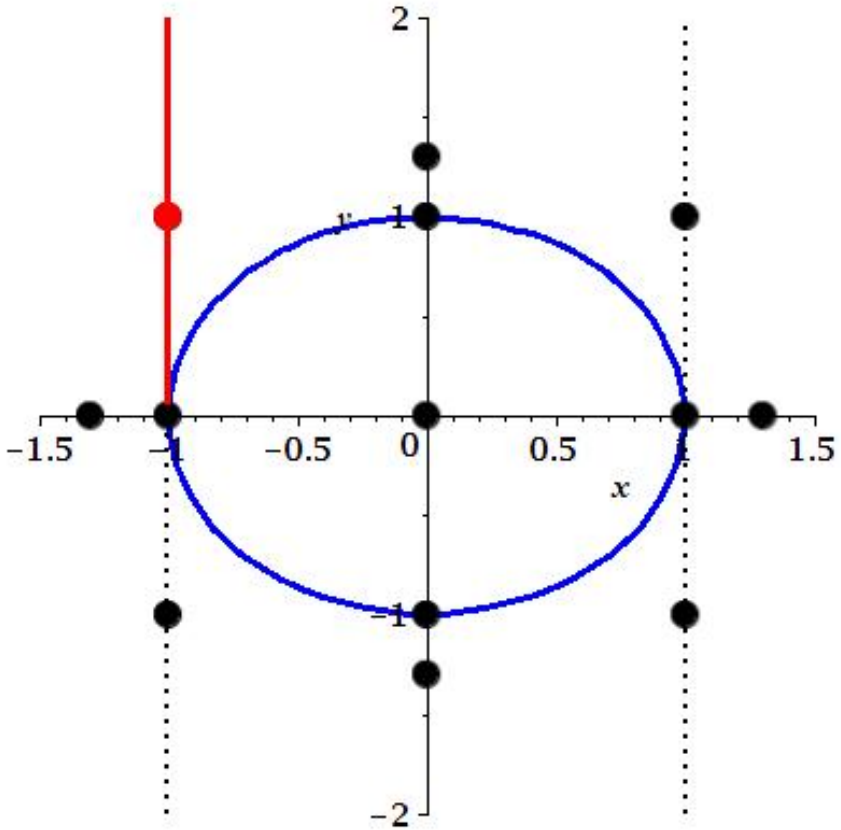
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



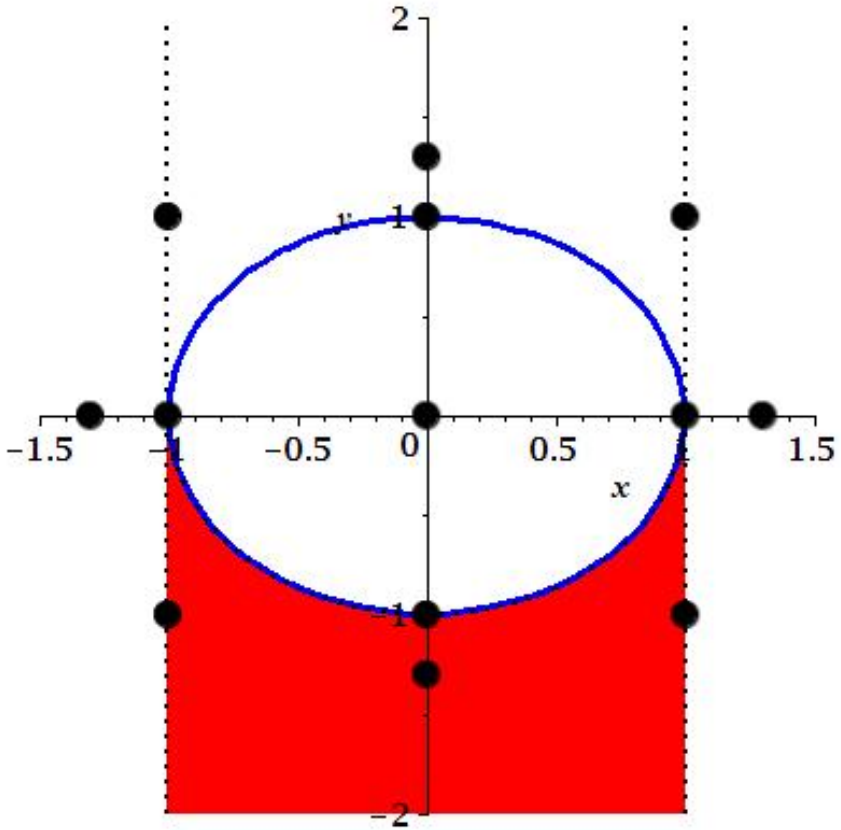
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



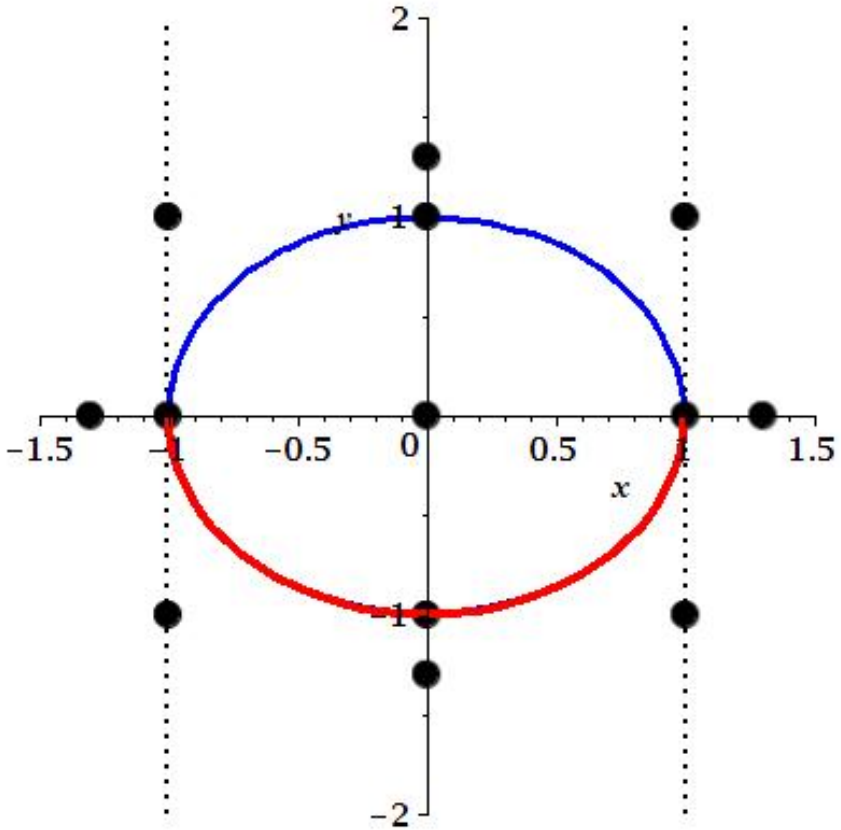
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



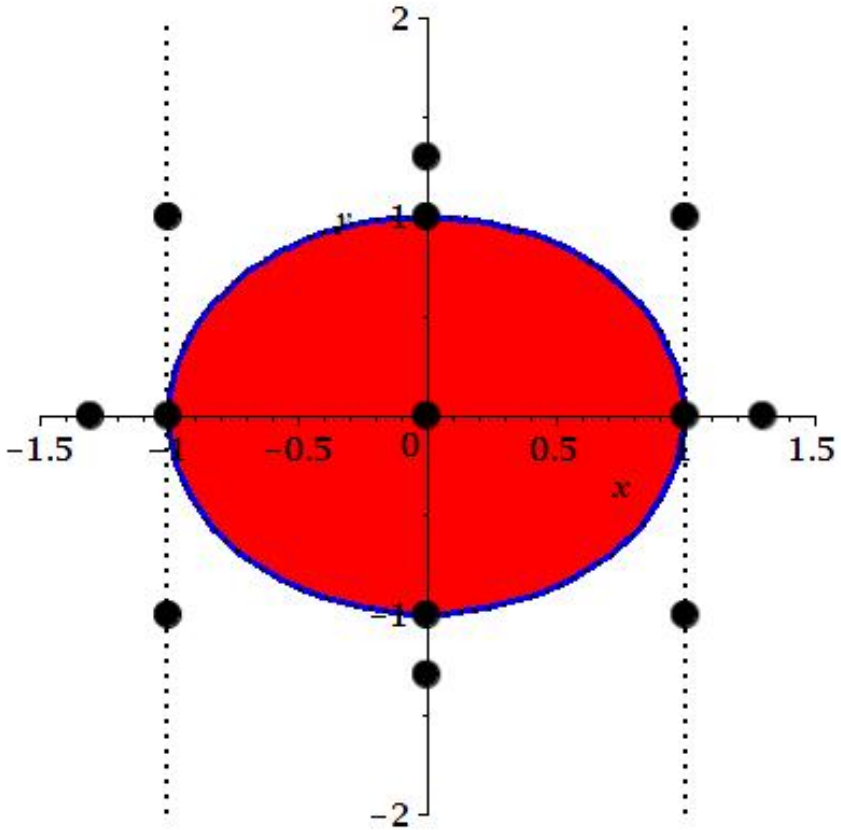
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



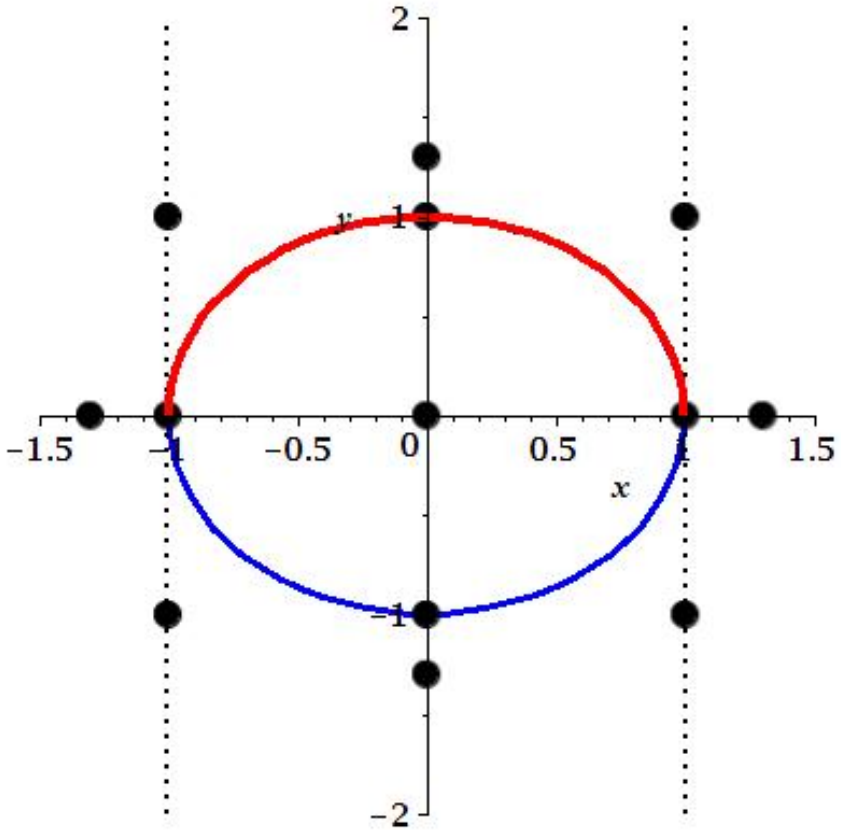
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



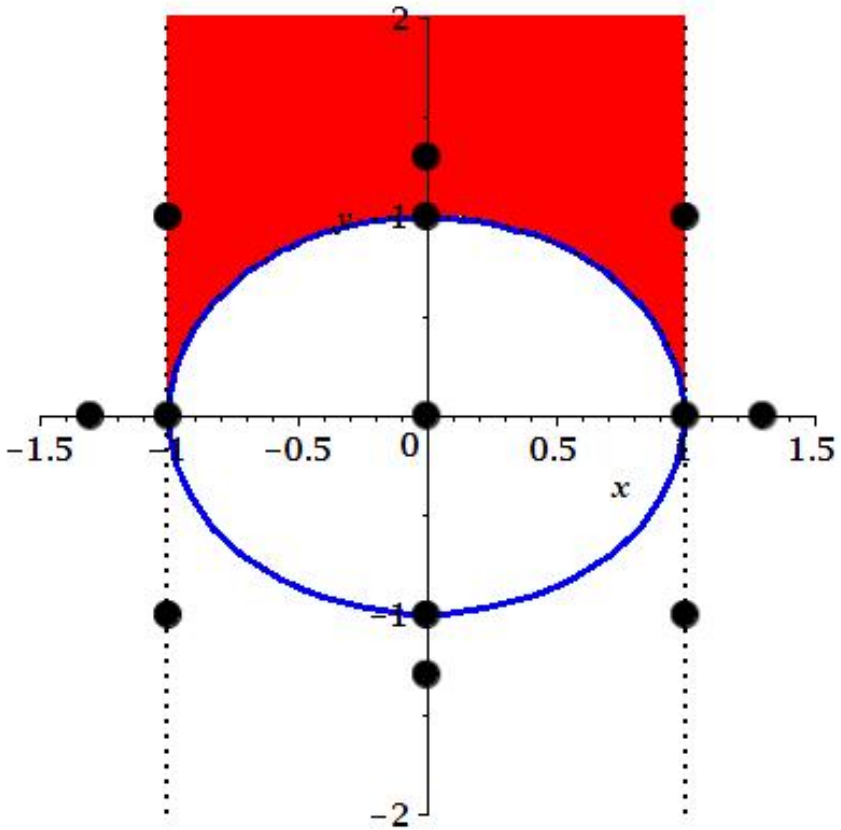
Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.



Example

A CAD of $\mathbb{R}^2$ sign invariant with respect to $f = x^2 + y^2 - 1$ can be given by 13 cells. The cylindricity is with projections onto x.

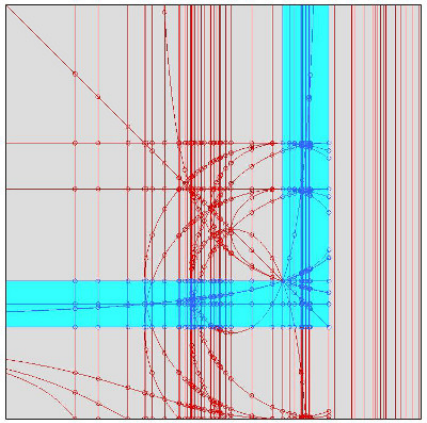


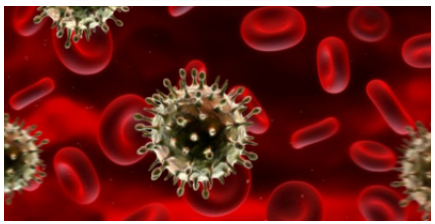
Motivation

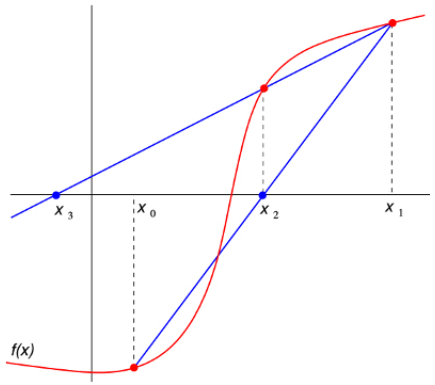
Original motivation is Quantifier Elimination, leading to many applications:

- derivation of optimal numerical schemes;
- parametric optimisation;
- epidemic modelling;
- control theory;
- theorem proving.

Other applications in semi-algebraic geometry include motion planning and programming with complex-valued functions.







CAD Terminology

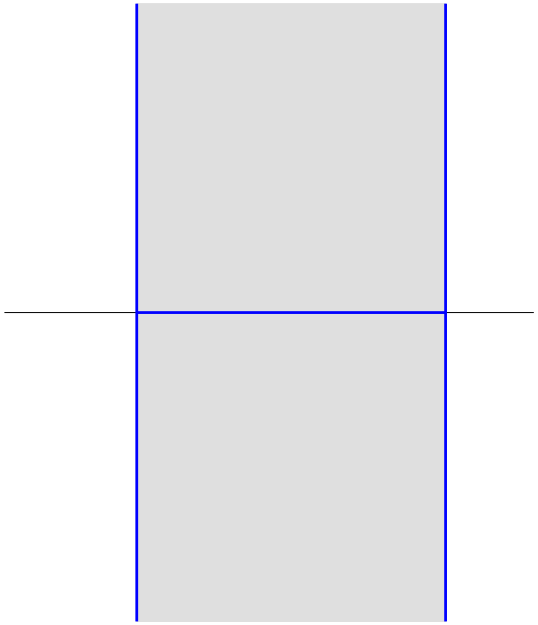
The cylindricity property means that all cells in a CAD of $\mathbb{R}^d$ lie in the cylinder above a cell, $c \in \mathbb{R}^{d-1}$.

I.e. in $c \times \mathbb{R}$.

We call the decomposition of the cylinder a stack. It consists of:

- sections of polynomials (cells where a polynomial vanishes);
- sectors cells in-between (or above / below) sections.

E.g. This stack has 3 sections and 4 sectors.



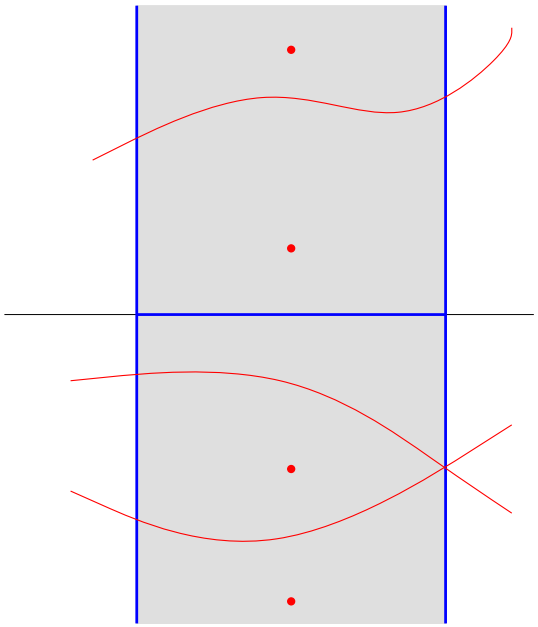
CAD Terminology

The cylindricity property means that all cells in a CAD of $\mathbb{R}^d$ lie in the cylinder above a cell, $c \in \mathbb{R}^{d-1}$.

I.e. in $c \times \mathbb{R}$.
We call the decomposition of the cylinder a stack. It consists of:

- sections of polynomials (cells where a polynomial vanishes);
- sectors cells in-between (or above / below) sections.

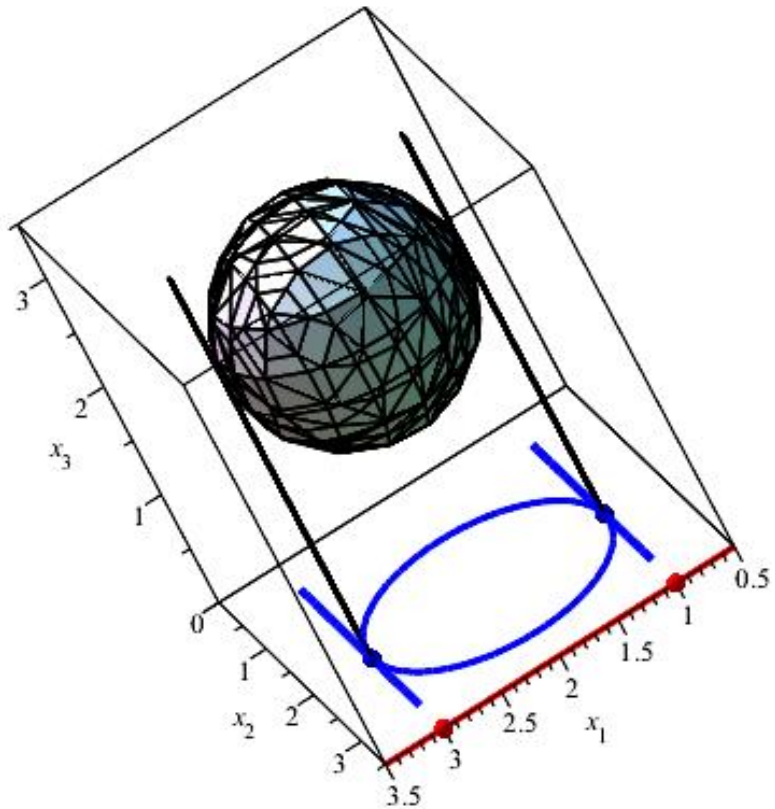
E.g. This stack has 3 sections and 4 sectors.



Projection and lifting

Traditionally CAD algorithms work by a process of:

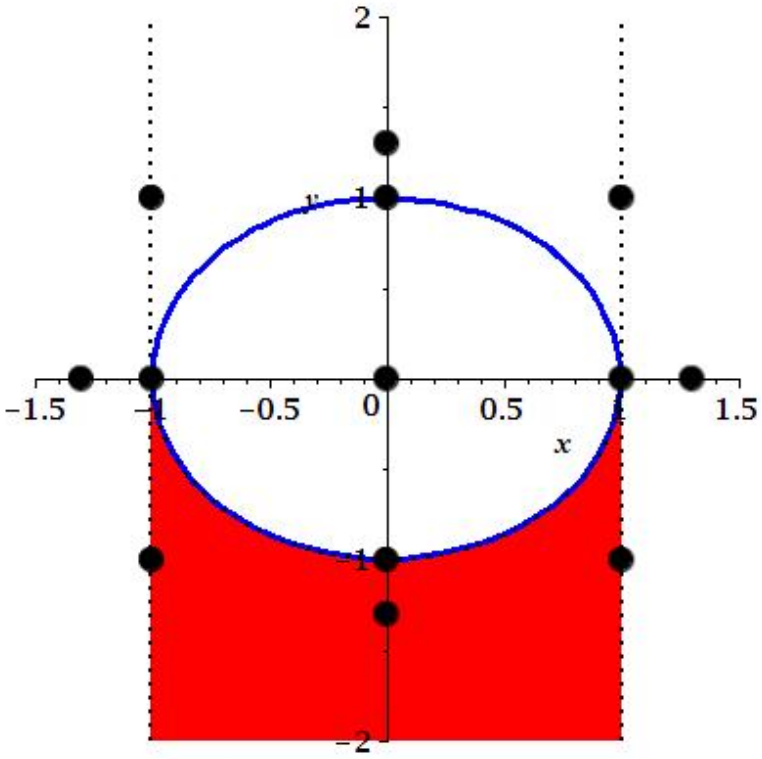
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting: to incrementally build CADs by dimension.



Projection and lifting

CAD algorithms usually work by a process of:

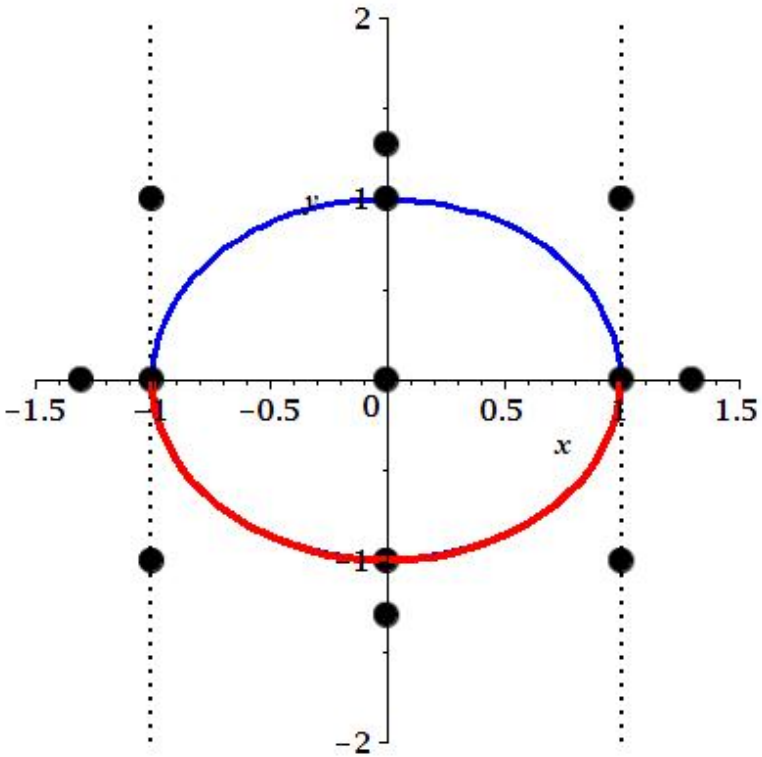
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.



Projection and lifting

Traditionally CAD algorithms work by a process of:

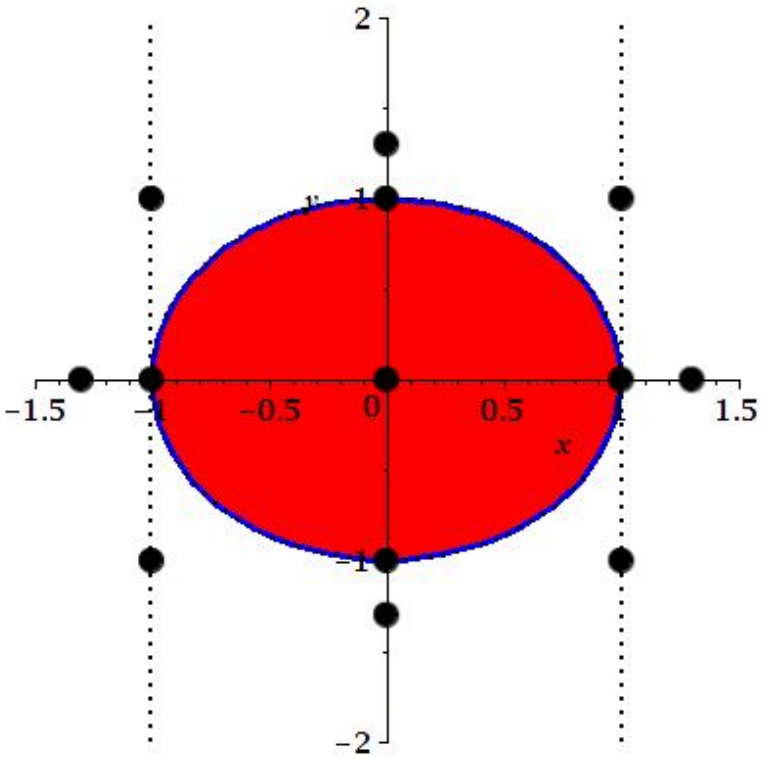
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.



Projection and lifting

Traditionally CAD algorithms work by a process of:

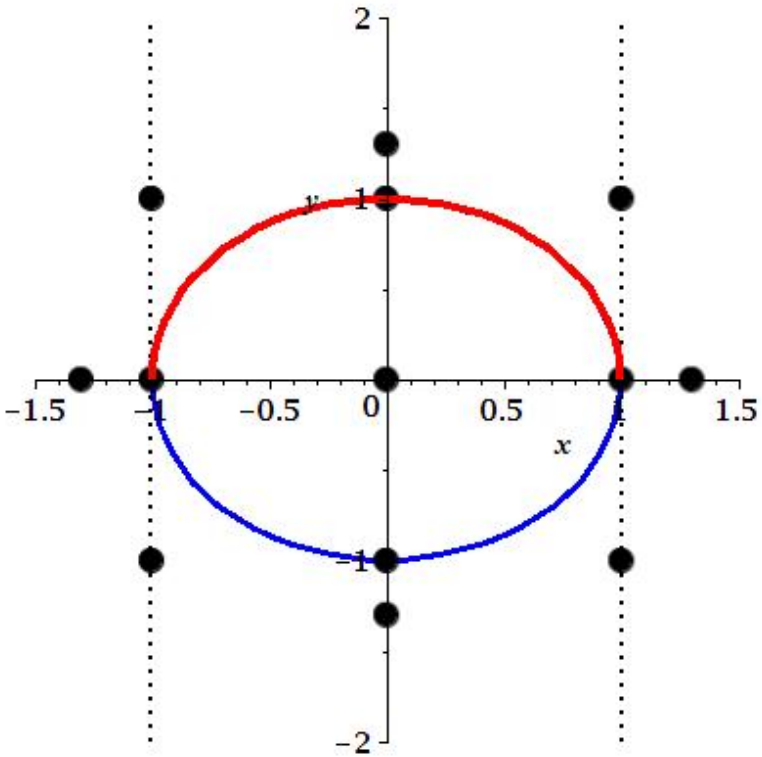
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.



Projection and lifting

Traditionally CAD algorithms work by a process of:

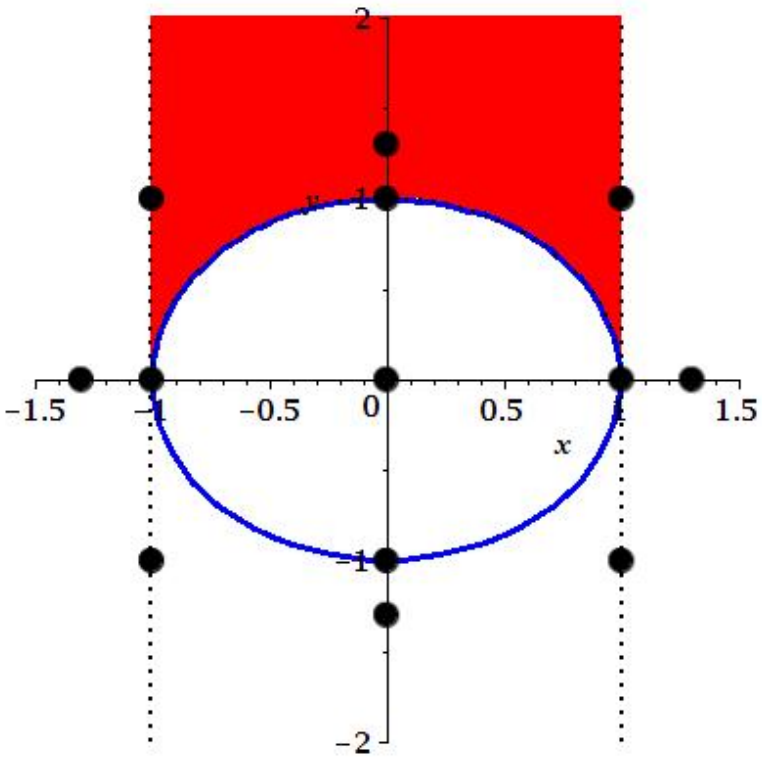
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.



Projection and lifting

Traditionally CAD algorithms work by a process of:

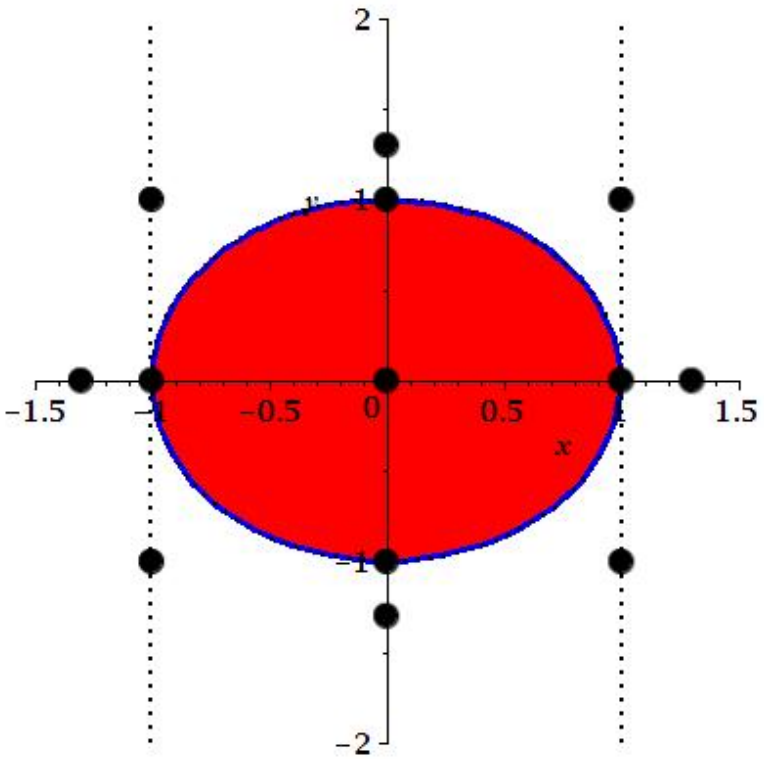
- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.

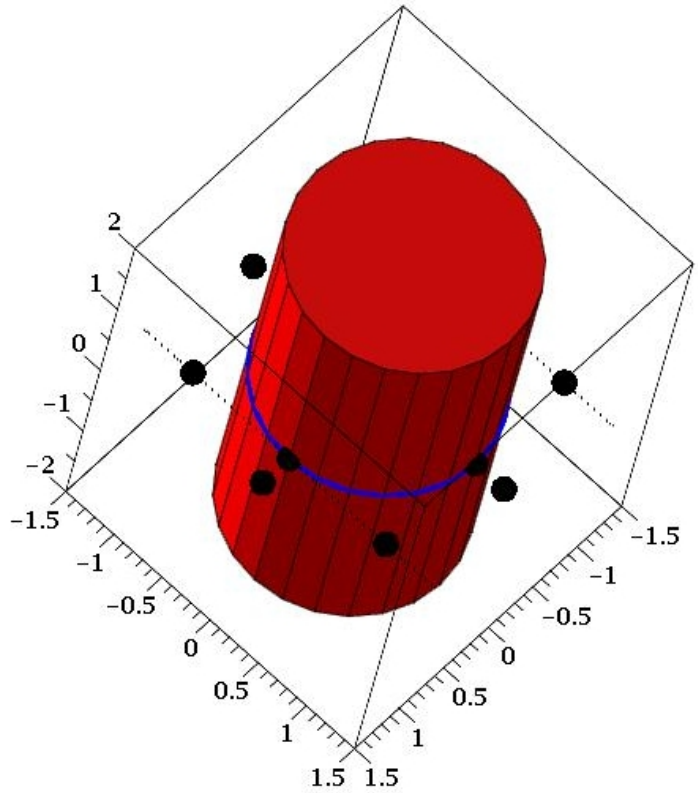


Projection and lifting

Traditionally CAD algorithms work by a process of:

- Projection: to derive a set of polynomials from the input which can define the decomposition
- Lifting to incrementally build CADs by dimension.





Delineability

Polynomials are delineable in a cell if the portion of their zero set in the cell consists of discrete sections. A set of polynomials is delineable when also the sections of different polynomials are identical or disjoint.

The projection operator must be chosen so that when lifting over a cell the polynomials are delineable. This allows us to draw conclusions about the whole cell when working at a sample point.

Truth Invariance

Sign-invariant CAD is more than needed for most applications.
More appropriate is a CAD truth-invariant for formulae.

- Implied by sign-invariance of polynomials in formulae;
- But can usually be achieved with far less cells.

Approaches to achieve truth-invariance:

- Refine sign-invariant CAD once produced;
- Terminate lifting once truth determined;
- Adapt projection according to formula structure.

Equational Constraints

An Equational Constraint (EC) is a polynomial equation logically implied by a Quantifier Free Tarski Formula (QFF). Either:

Explicit when an atom of the formula; as $f = 0$ is in

$$\phi = (f = 0) \wedge \varphi.$$

Implicit otherwise; as $f_1 f_2 = 0$ is in

$$\psi = (f_1 = 0 \wedge \varphi_1) \vee (f_2 = 0 \wedge \varphi_2).$$

In [Collins98] Collins observed that truth invariance of a formula with equational constraint $f = 0$ is implied by:

- Sign-invariant for f; and
- Sign-invariant for all other polynomials g_i when $f = 0$.

Equational Constraints

An Equational Constraint (EC) is a polynomial equation logically implied by a Quantifier Free Tarski Formula (QFF). Either:

Explicit when an atom of the formula; as $f = 0$ is in

$$\phi = (f = 0) \wedge \varphi.$$

Implicit otherwise; as $f_1 f_2 = 0$ is in

$$\psi = (f_1 = 0 \wedge \varphi_1) \vee (f_2 = 0 \wedge \varphi_2).$$

In [Collins98] Collins observed that truth invariance of a formula with equational constraint $f = 0$ is implied by:

- Sign-invariant for f; and
- Sign-invariant for all other polynomials g_i when $f = 0$.

McCallum Projection operators

McCallum1998 $P(B) = \text{coeff}(B) \cup \text{disc}(B) \cup \text{res}(B)$

Theorem 1: If $P(B)$ order-invariant on S then: B delineable on S; and on the sections of B not nullified B is order-invariant.

McCallum1999 $P_F(B) = P(F) \cup \{\text{res}(f, g) \mid f \in F, g \in B \setminus F\}$

Theorem 2: Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order-invariant on S. Then g is sign-invariant in every section of f over S.

McCallum2001 $P_F^*(B) := P_F(B) \cup \text{disc}(B \setminus F)$

Theorem 3: Suppose $r = \text{res}(f, g)$, $d = \text{disc}(g)$, $r, d \neq 0$; f delineable on S, g not nullified and r, d order-invariant on S. Then g is order-invariant in every section of f over S.

McCallum Projection operators

McCallum1998 $P(B) = \text{coeff}(B) \cup \text{disc}(B) \cup \text{res}(B)$

Theorem 1: If $P(B)$ order-invariant on S then: B delineable on S; and on the sections of B not nullified B is order-invariant.

McCallum1999 $P_F(B) = P(F) \cup \{\text{res}(f, g) \mid f \in F, g \in B \setminus F\}$

Theorem 2: Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order-invariant on S. Then g is sign-invariant in every section of f over S.

McCallum2001 $P^*_F(B) := P_F(B) \cup \text{disc}(B \setminus F)$

Theorem 3: Suppose $r = \text{res}(f, g)$, $d = \text{disc}(g)$, $r, d \neq 0$; f delineable on S, g not nullified and r, d order-invariant on S. Then g is order-invariant in every section of f over S.

McCallum Projection operators

McCallum1998 $P(B) = \text{coeff}(B) \cup \text{disc}(B) \cup \text{res}(B)$

Theorem 1: If $P(B)$ order -invariant on S then: B delineable on S; and on the sections of B not nullified B is order -invariant.

McCallum1999 $P_F(B) = P(F) \cup \{\text{res}(f, g) \mid f \in F, g \in B \setminus F\}$

Theorem 2: Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order -invariant on S. Then g is sign -invariant in every section of f over S.

McCallum2001 $P_F^*(B) := P_F(B) \cup \text{disc}(B \setminus F)$

Theorem 3: Suppose $r = \text{res}(f, g)$, $d = \text{disc}(g)$, $r, d \neq 0$; f delineable on S, g not nullified and r, d order -invariant on S. Then g is order -invariant in every section of f over S.

McCallum Projection operators

McCallum1998 $P(B) = \text{coeff}(B) \cup \text{disc}(B) \cup \text{res}(B)$

Theorem 1: If $P(B)$ order -invariant on S then: B delineable on S; and on the sections of B not nullified B is order -invariant.

McCallum1999 $P_F(B) = P(F) \cup \{\text{res}(f, g) \mid f \in F, g \in B \setminus F\}$

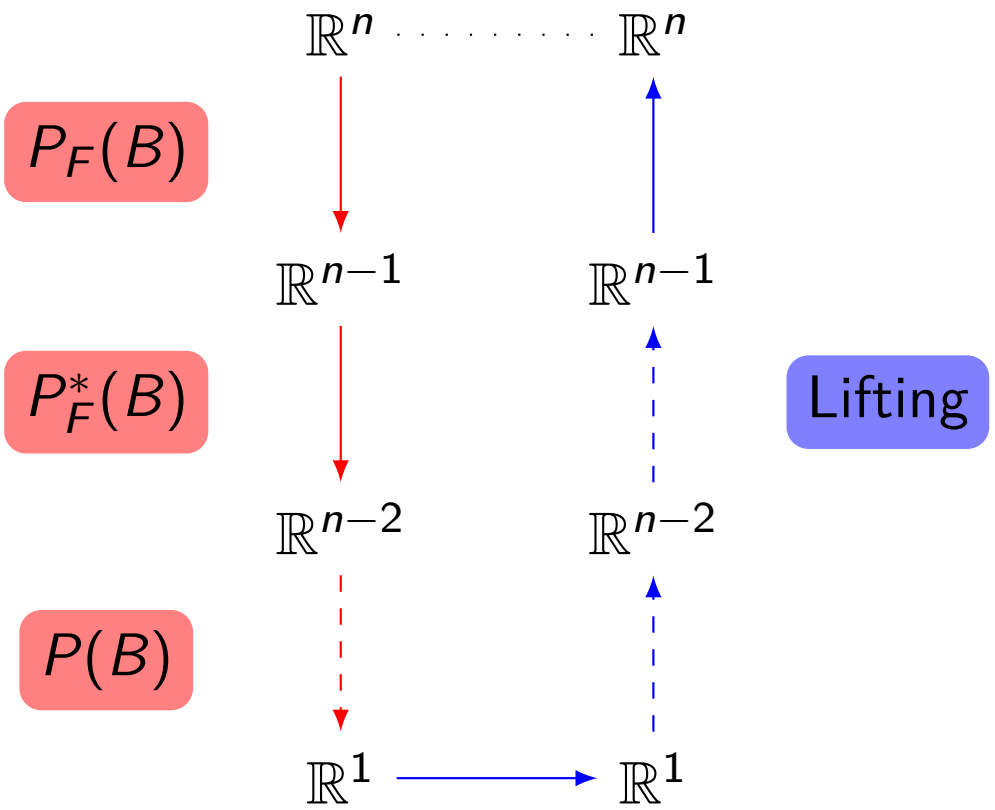
Theorem 2: Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order -invariant on S. Then g is sign -invariant in every section of f over S.

McCallum2001 $P_F^*(B) := P_F(B) \cup \text{disc}(B \setminus F)$

Theorem 3: Suppose $r = \text{res}(f, g)$, $d = \text{disc}(g)$, $r, d \neq 0$; f delineable on S, g not nullified and r, d order -invariant on S. Then g is order -invariant in every section of f over S.

McCallum's propagation of ECs

Reduced projection for first step. Then include extra discriminants for subsequent projection with EC; normal projection otherwise.

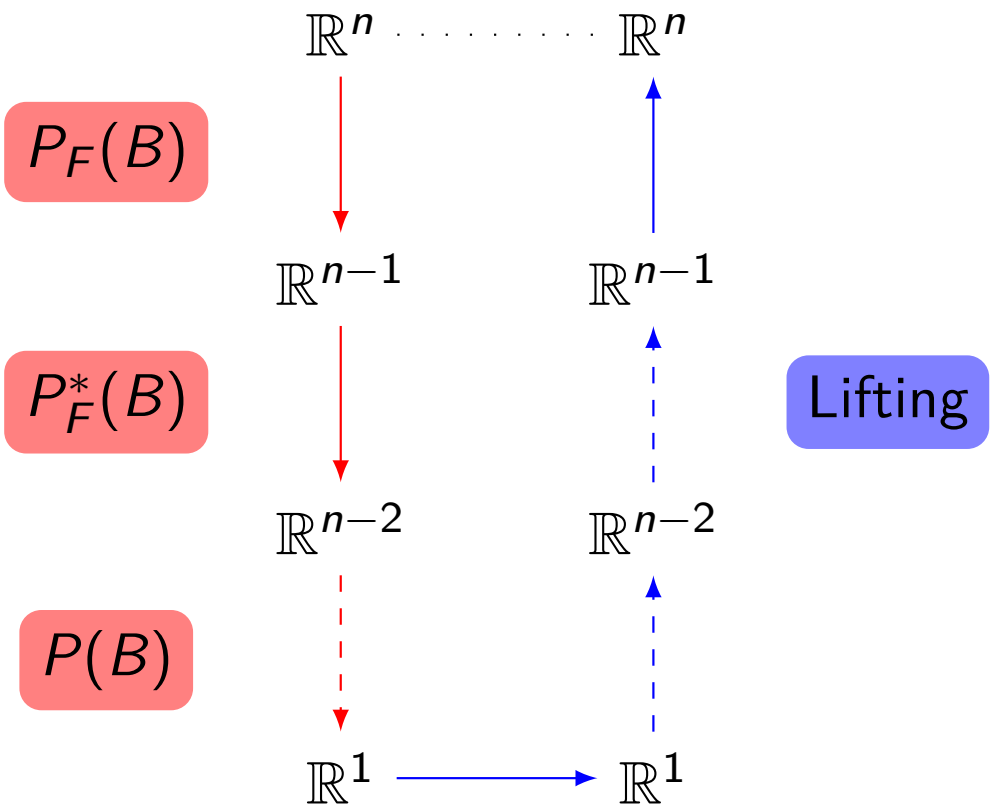


Q) What if two ECs at top level?

Then propagate by considering their resultant in main variable: an EC not in the main variable.

McCallum's propagation of ECs

Reduced projection for first step. Then include extra discriminants for subsequent projection with EC; normal projection otherwise.



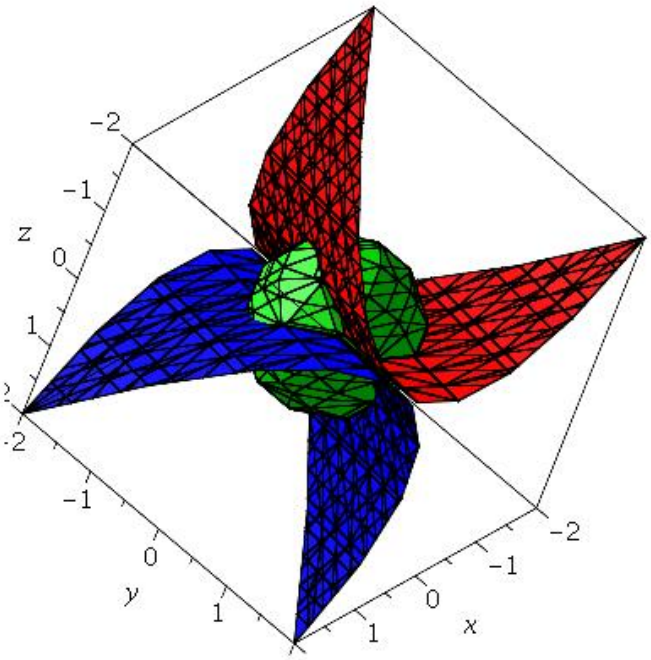
Q) What if two ECs at top level?

Then propagate by considering their resultant in main variable: an EC not in the main variable.

Example

Define $z \succ y \succ x$ and $\phi = f_1 = 0 \wedge f_2 = 0 \wedge g \geq 0$ where

$$f_1 = x + y^2 + z, \quad f_2 = x - y^2 + z, \quad g = x^2 + y^2 + z^2 - 1.$$



$f_1 = f_2$ when $y = 0$
$\Rightarrow |x| \geq \sqrt{2}/2, z = -x.$

How To find this with CAD?

Sign-invariant: 1487 cells

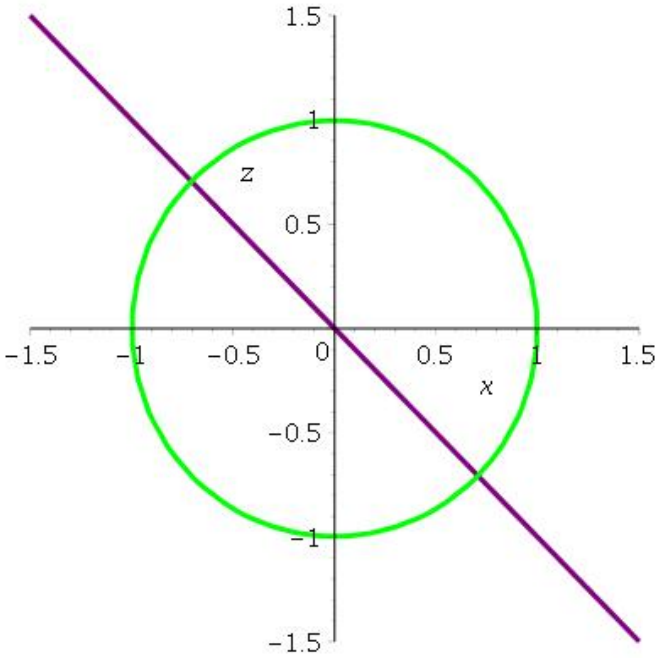
Single EC (either): 289 cells. Solution has 8 cells splitting at $x = \frac{1}{2}(1 \pm \sqrt{6})$.

Both ECs (with resultant propagated): 133 cells. Solution given by 4 (the minimum).

Example

Define $z \succ y \succ x$ and $\phi = f_1 = 0 \wedge f_2 = 0 \wedge g \geq 0$ where

$$f_1 = x + y^2 + z, \quad f_2 = x - y^2 + z, \quad g = x^2 + y^2 + z^2 - 1.$$



$f_1 = f_2$ when $y = 0$
$\Rightarrow |x| \geq \sqrt{2}/2, z = -x.$

How To find this with CAD?

Sign-invariant: 1487 cells

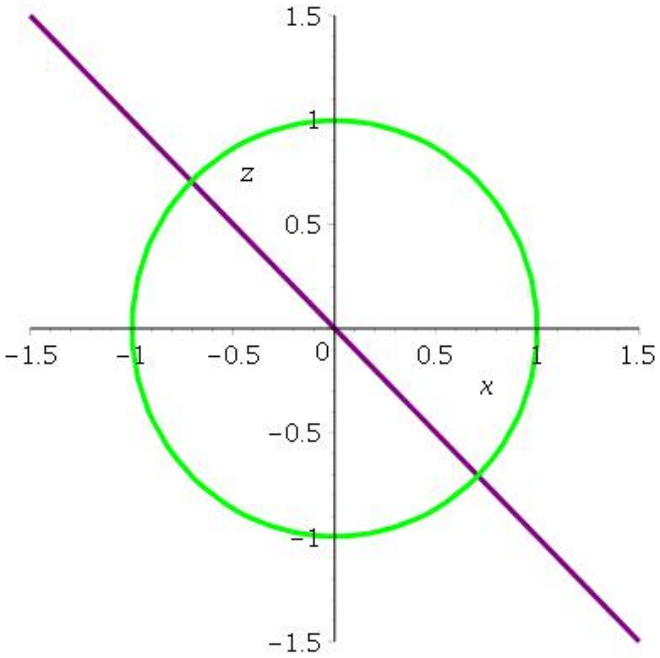
Single EC (either): 289 cells. Solution has 8 cells splitting at $x = \frac{1}{2}(1 \pm \sqrt{6})$.

Both ECs (with resultant propagated): 133 cells. Solution given by 4 (the minimum).

Example

Define $z \succ y \succ x$ and $\phi = f_1 = 0 \wedge f_2 = 0 \wedge g \geq 0$ where

$$f_1 = x + y^2 + z, \quad f_2 = x - y^2 + z, \quad g = x^2 + y^2 + z^2 - 1.$$



$f_1 = f_2$ when $y = 0$
$\Rightarrow |x| \geq \sqrt{2}/2, z = -x.$

How To find this with CAD?

Sign-invariant: 1487 cells

Single EC (either): 289 cells. Solution has 8 cells splitting at $x = \frac{1}{2}(1 \pm \sqrt{6})$.

Both ECs (with resultant propagated): 133 cells. Solution given by 4 (the minimum).

Outline

1. Introduction
   - Cylindrical Algebraic Decomposition
   - Equational Constraints
2. Improving the Use of ECs in CAD
   - Reductions in the Lifting Phase
   - Algorithm
3. Evaluating the New Algorithm
   - Worked Example
   - Complexity Analysis

Reductions in the lifting phase

Our contribution is to describe how the theory of reduced projection operators allows for savings when lifting also.
To achieve this we have to discard two embedded principles:

1. **That the projection polynomials are a fixed set.**
   Also rejected in the `RegularChains` approach to CAD of Chen, Maza, Xia, Yang.
2. **That the invariance structure of the final CAD can be expressed as sign-invariance of certain polynomials.**
   Also rejected in work on CAD for varieties of Brown and McCallum.

Reducing Lifting Polynomials

Theorem 2

Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order-invariant on S. Then g sign-invariant in every section of f over S.

Sufficient to lift only with respect to f (assuming f forms EC).

- Lifting will impose sign-invariance for f, while Theorem 2 guarantees it for g.
- In previous example:
  - Single EC use would have 141 cells instead of 289.
  - Both EC use would have 45 cells instead of 133.
- Proof follows directly from Theorem in [McCallum99] but not realised until work on TTICAD in ISSAC 2013.
- We can no longer talk about a single unified set of projection polynomials.

Reducing Lifting Polynomials

Theorem 2

Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order-invariant on S. Then g sign-invariant in every section of f over S.

Sufficient to lift only with respect to f (assuming f forms EC).

- Lifting will impose sign-invariance for f, while Theorem 2 guarantees it for g.
- In previous example:
  - Single EC use would have 141 cells instead of 289.
  - Both EC use would have 45 cells instead of 133.
- Proof follows directly from Theorem in [McCallum99] but not realised until work on TTICAD in ISSAC 2013.
- We can no longer talk about a single unified set of projection polynomials.

Reducing Lifting Polynomials

Theorem 2

Suppose $r = \text{res}(f, g), r \neq 0$; f delineable on S and r order-invariant on S. Then g sign-invariant in every section of f over S.

Sufficient to lift only with respect to f (assuming f forms EC).

- Lifting will impose sign-invariance for f, while Theorem 2 guarantees it for g.
- In previous example:
  - Single EC use would have 141 cells instead of 289.
  - Both EC use would have 45 cells instead of 133.
- Proof follows directly from Theorem in [McCallum99] but not realised until work on TTICAD in ISSAC 2013.
- We can no longer talk about a single unified set of projection polynomials.

Reducing Lifting Cells

When there exists an EC (not in the main variable) we use the operator $P_F^*(B)$ and lift with F only. The stacks consist of:

sections: where an element of F is zero;

sectors: where EC is not satisfied and the formula thus false.

Then: **lift over the sectors trivially** (extend to full cylinder); split stacks according to zeros of polynomials **over the sections only**.

- Easy to identify the sections (parity of cell index).
- In previous example cell count of 45 is reduced further to 25.
- Proof: induction making use of earlier theorems and restricting to truth-invariance.
- The final CAD will be truth-invariant for the formula, but may not be sign invariant for any individual polynomial.

Reducing Lifting Cells

When there exists an EC (not in the main variable) we use the operator $P_F^*(B)$ and lift with F only. The stacks consist of:

sections: where an element of F is zero;

sectors: where EC is not satisfied and the formula thus false.

Then: **lift over the sectors trivially** (extend to full cylinder); split stacks according to zeros of polynomials **over the sections only**.

- Easy to identify the sections (parity of cell index).
- In previous example cell count of 45 is reduced further to 25.
- Proof: induction making use of earlier theorems and restricting to truth-invariance.
- The final CAD will be truth-invariant for the formula, but may not be sign invariant for any individual polynomial.

Reducing Lifting Cells

When there exists an EC (not in the main variable) we use the operator $P_F^*(B)$ and lift with F only. The stacks consist of:

sections: where an element of F is zero;

sectors: where EC is not satisfied and the formula thus false.

Then: **lift over the sectors trivially** (extend to full cylinder); split stacks according to zeros of polynomials **over the sections only**.

- Easy to identify the sections (parity of cell index).
- In previous example cell count of 45 is reduced further to 25.
- Proof: induction making use of earlier theorems and restricting to truth-invariance.
- The final CAD will be truth-invariant for the formula, but may not be sign invariant for any individual polynomial.

(Sketch of) Algorithm for multiple ECs

1: Identify and propagate ECs.
2: Set n as number of variables
3: **for** v in $v_n, \ldots, v_2$ **do**
4: **if** no EC with mvar v **then**
5: Apply $P(B)$
6: **else**
7: **if** v is v_n or v_2 **then**
8: Apply $P_F(B)$
9: **else**
10: Apply $P_F^*(B)$
11: **end if**
12: **end if**
13: **end for**
14: Build CAD of $\mathbb{R}^1$
15: **for** k from 2 to n **do**
16: **if** EC with mvar v_k **then**
17: Set L to be EC
18: **else**
19: Set L to all polys
20: **end if**
21: **if** EC with mvar v_{k-1} **then**
22: Lift wrt L over sections (last index even)
23: Extend rest to cylinders
24: **else**
25: Lift wrt L over all cells
26: **end if**
27: **end for**

Outline

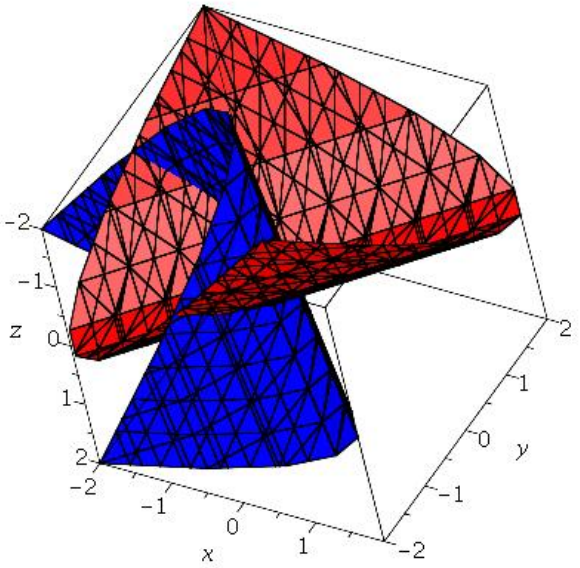
1. Introduction
   - Cylindrical Algebraic Decomposition
   - Equational Constraints
2. Improving the Use of ECs in CAD
   - Reductions in the Lifting Phase
   - Algorithm
3. Evaluating the New Algorithm
   - Worked Example
   - Complexity Analysis

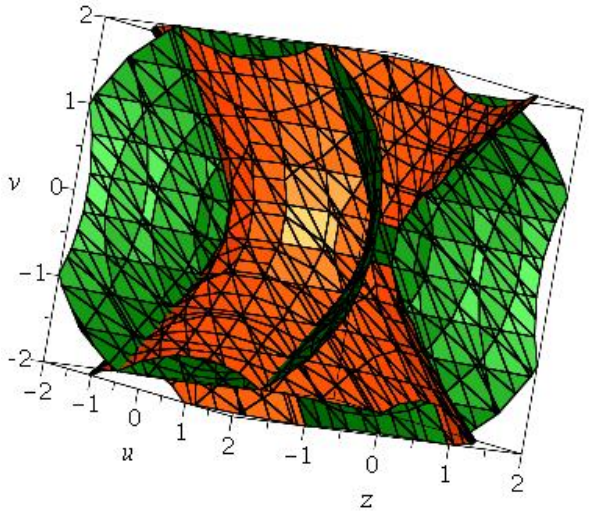
Worked Example

Let $z \succ y \succ x \succ u \succ v$ and define

$$\phi = f_1 = 0 \wedge f_2 = 0 \wedge f_3 = 0 \wedge f_4 = 0 \wedge g \geq 0 \wedge h \geq 0,$$

$$f_1 := x - y + z^2, \quad f_2 := z^2 - u^2 + v^2 - 1, \quad g := x^2 - 1,$$
$$f_3 := x + y + z^2, \quad f_4 := z^2 + u^2 - v^2 - 1, \quad h := z.$$





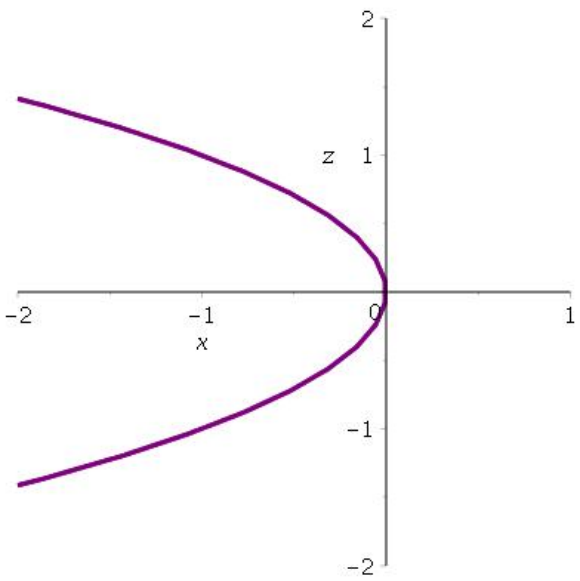
$$\left\{ \begin{array}{l} u = \pm v, \\ x = -1, \\ y = 0, \\ z = 1 \end{array} \right\}$$

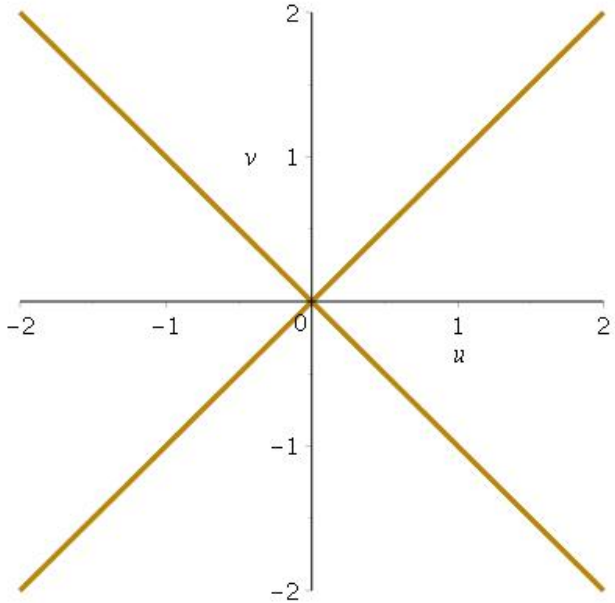
Worked Example

Let $z \succ y \succ x \succ u \succ v$ and define

$$\phi = f_1 = 0 \wedge f_2 = 0 \wedge f_3 = 0 \wedge f_4 = 0 \wedge g \geq 0 \wedge h \geq 0,$$

$$f_1 := x - y + z^2, \quad f_2 := z^2 - u^2 + v^2 - 1, \quad g := x^2 - 1,$$
$$f_3 := x + y + z^2, \quad f_4 := z^2 + u^2 - v^2 - 1, \quad h := z.$$





$$\left\{ \begin{array}{l} u = \pm v, \\ x = -1, \\ y = 0, \\ z = 1 \end{array} \right\}$$

Propagation

There are 4 explicit ECs all with main variable z. However, taking repeated resultants discovers many additional implicit ECs:

mvar z	4
mvar y	5
mvar x	3
mvar u	1
mvar v	0

We can only make use of one EC for each projection. Hence 60 different possibilities for EC designation.

Three different cell counts for truth invariant CADs using the new algorithm: 113, 103 or 93.

Comparison

Three different cell counts for truth invariant CADs using the new algorithm: 113, 103 or 93.

- **Sign-invariant CAD** for all 6 polynomials has 1,118,205 cells.
- **Using a single EC**:
  - 11,961, 30,233, 158,475 or 158,451 (depending on EC choice); or
  - 3023, 10,935 or 48,299 (twice) when lifting only with that EC.
- **Propagating ECs but regular lifting**: 21,079 (default in `Qepcad`) but can get as little as 5633 cells by making the *correct* designation.

The `RegularChains` approach to CAD also makes use of multiple ECs. Development version can achieve 137 cell output.

Comparison

Three different cell counts for truth invariant CADs using the new algorithm: 113, 103 or 93.

- **Sign-invariant CAD** for all 6 polynomials has 1,118,205 cells.
- **Using a single EC**:
  - 11,961, 30,233, 158,475 or 158,451 (depending on EC choice); or
  - 3023, 10,935 or 48,299 (twice) when lifting only with that EC.
- **Propagating ECs but regular lifting**: 21,079 (default in `Qepcad`) but can get as little as 5633 cells by making the *correct* designation.

The `RegularChains` approach to CAD also makes use of multiple ECs. Development version can achieve 137 cell output.

Complexity Analysis

Ideas used in analysis:

- Compare the bounds on the number of cells produced in a CAD (closely correlated to computation time);
- Then measure the dominant terms in these.
- Start with m polynomials of maximum degree d in any one of n variables;
- Assume EC declared for first ℓ projections;

Complexity Analysis

Ideas used in analysis:

- Compare the bounds on the number of cells produced in a CAD (closely correlated to computation time);
- Then measure the dominant terms in these.
- Start with m polynomials of maximum degree d in any one of n variables;
- Assume EC declared for first ℓ projections;

Complexity Results

The dominant term in the cell count bound:

McCallum1998	$(2d)^{2^n-1}m^{2^n-1}2^{2^{n-1}-1}$
Reduced projection only	$(2d)^{2^n-1}m^{2^{n-\ell}+\ell-1}2^{\ell 2^{n-\ell}+\ell(\ell-3)/2}$
Reduced lifting also	$(2d)^{2^n-1}m^{2^{n-\ell}-2}2^{\ell 2^{n-\ell}-3\ell}$

- CAD is inherently double exponential and the factor that varies with degree is still doubly exponential in number of variables. But:
- Taking advantage of ECs in projection reduced double exponents by ℓ (number of ECs).
- Improved lifting leads to further reduction in single exponents.

Complexity Results

The dominant term in the cell count bound:

McCallum1998 $(2d)^{2^n-1} m^{2^n-1} 2^{2^{n-1}-1}$

Reduced projection only $(2d)^{2^n-1} m^{2^{n-\ell}+\ell-1} 2^{\ell 2^{n-\ell}+\ell(\ell-3)/2}$

Reduced lifting also $(2d)^{2^n-1} m^{2^{n-\ell}-2} 2^{\ell 2^{n-\ell}-3\ell}$

- CAD is inherently double exponential and the factor that varies with degree is still doubly exponential in number of variables. But:
- Taking advantage of ECs in projection reduced double exponents by ℓ (number of ECs).
- Improved lifting leads to further reduction in single exponents;

Complexity Results

The dominant term in the cell count bound:

McCallum1998 $(2d)^{2^n-1} m^{2^n-1} 2^{2^{n-1}-1}$

Reduced projection only $(2d)^{2^n-1} m^{2^{n-\ell}+\ell-1} 2^{\ell 2^{n-\ell}+\ell(\ell-3)/2}$

Reduced lifting also $(2d)^{2^n-1} m^{2^{n-\ell}-2} 2^{\ell 2^{n-\ell}-3\ell}$

- CAD is inherently double exponential and the factor that varies with degree is still doubly exponential in number of variables. But:
- Taking advantage of ECs in projection reduced double exponents by ℓ (number of ECs).
- Improved lifting leads to further reduction in single exponents;

Complexity Results

The dominant term in the cell count bound:

McCallum1998 $(2d)^{2^n-1}m^{2^n-1}2^{2^{n-1}-1}$

Reduced projection only $(2d)^{2^n-1}m^{2^{n-\ell}+\ell-1}2^{\ell 2^{n-\ell}+\ell(\ell-3)/2}$

Reduced lifting also $(2d)^{2^n-1}m^{2^{n-\ell}-2}2^{\ell 2^{n-\ell}-3\ell}$

- CAD is inherently double exponential and the factor that varies with degree is still doubly exponential in number of variables. But:
- Taking advantage of ECs in projection reduced double exponents by ℓ (number of ECs).
- Improved lifting leads to further reduction in single exponents;

Future work

Results presented so far relate only to primitive ECs. How to take advantage of non-primitive ECs?

Example

Consider $\phi := zy = 0 \wedge \varphi$. Under ordering $\cdots \succ z \succ y \succ \ldots$ the EC $zy = 0$ is not primitive.

We could take $\{z\}$ as the primitive part; project with reduced operator and include content y in projection. CAD of $(y, \ldots)$-space would be sign-invariant for y and thus CAD of $(z, y, \ldots)$-space truth invariant for $zy = 0$. However, it is **no longer that only above sections can ϕ be true**.

Future work (continued)

How to proceed?

- Simply lift over all cells.
- Instead rewrite ϕ as $\phi := (z = 0 \wedge \varphi) \vee (y = 0 \wedge \varphi)$, so each clause has its own EC and use the theory of TTICAD.

Also

- Extension of TTICAD to multiple levels;
- Investigation of how best to propagate ECs (which EC designations).

McCallum's Papers

S. McCallum

An improved projection operator for cylindrical algebraic decomposition.

Quantifier Elimination and Cylindrical Algebraic Decomposition, pages 242–268, Springer, 1998.

S. McCallum

On projection in CAD-based quantifier elimination with equational constraints.

Proc. ISSAC '99, pages 145–149, ACM, 1999.

S. McCallum

On propagation of equational constraints in CAD-based quantifier elimination .

Proc. ISSAC '01, pages 223–231, ACM, 2001.

Further Information

M. England, R. Bradford and J.H.Davenport
Improving the use of equational constraints in cylindrical algebraic decomposition.
Proc. ISSAC '15, pages 165–172, ACM, 2015.

Contact Details

Matthew.England@coventry.ac.uk

http://computing.coventry.ac.uk/~mengland/index.html

Thanks for listening!